

DESCRIPTION DU SYSTEME DE PILOTAGE DE RESEAU FERROVIAIRE UTS "ULYSSE TRAIN SYSTEM" v2.0

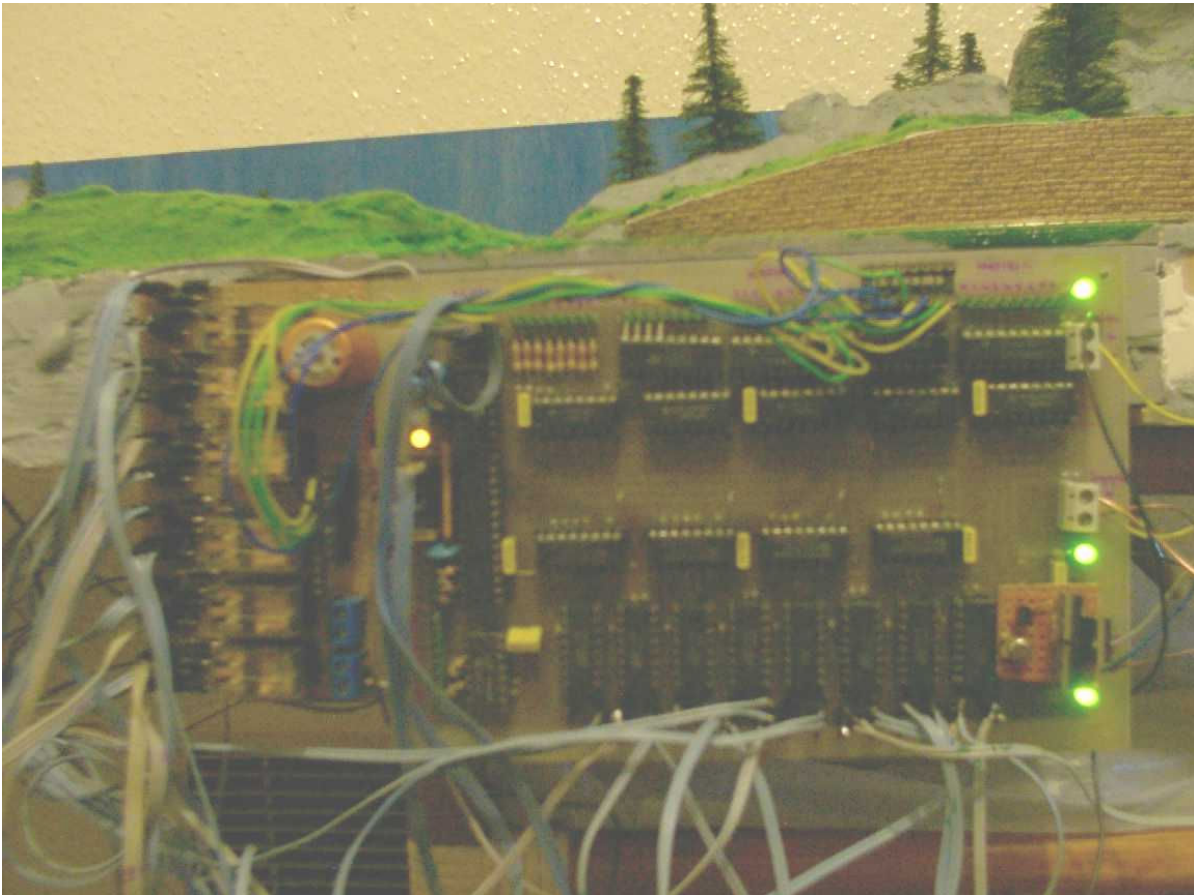


Photo de la carte électronique en opération. Elle est accompagnée d'une carte de commande capacitive à relais pour les aiguillages. Le câblage n'est pas encore définitif surtout pour la liaison entre les 2 cartes ;-)

Je dédie cette réalisation à tous les modélistes qui procurent du rêve aux petits comme aux grands ...

Table des matières :

1. Introduction
2. Présentation
3. Electronique
4. Informatique
5. Conclusion

Ulysse Delmas_Begue (Dernière mise à jour de cette doc : 3 juillet 2005 - Version correspondante d'UTS : 2.0)
Email : udelmas@chez.com (ou ulyссе.delmas@laposte.net si la première ne fonctionne pas bien)
SiteWeb : <http://www.chez.com/udelmas>

1. Introduction

UTS est un système de contrôle de réseau ferroviaire piloté par ordinateur et reposant sur le principe du cantonnement. J'ai créé ce système car je ne trouvais pas ce que je recherchais sur le marché ou sur Internet. Cette documentation a pour but de décrire mon système afin que vous puissiez le réaliser, l'adapter ou vous en inspirer pour vos réalisations. J'ai décidé de livrer ce système gratuitement aux particuliers et associations (pour réalisation commerciale : me contacter) dans l'esprit GNU, c'est à dire que vous pouvez faire ce que vous voulez à condition de redistribuer toutes les informations et les sources des codes si vous décidez de redistribuer le système. Et oui quelques réalisations gratuites existent encore dans un monde où tout se vend et se brevète mais c'est une autre histoire ... Enfin j'ai essayé de mettre ce document à la portée du plus grand nombre en expliquant le maximum de choses afin que vous compreniez ce que vous faites, même si vous ne connaissez rien en électronique et informatique. En effet il est ridicule d'acquiescer un système qu'on ne comprend pas car en cas de pannes ou problèmes vous seriez vite très embêtés (et de toutes façons vous ne pourriez pas le construire). Bref vous l'avez compris, le partage d'information est très important pour moi. N'hésitez pas à me contacter pour toutes questions mais merci de lire la documentation jusqu'au bout avant. Bien entendu une photo de votre réalisation me ferait le plus grand plaisir. Et bien entendu, la célèbre phrase s'applique à mon projet : Je ne pourrais être tenu pour responsable de quelque manière que ce soit des dommages relatifs à l'utilisation des infos de cette documentation. (Il ne manquerait plus que ça pour un système gratuit). Enfin sachez que j'ai pris du plaisir à réaliser ce système et j'espère que cette documentation vous servira à quelque chose. Bonne lecture.

2. Présentation

Ce système s'appelle UTS pour "Ulysse Train System". C'est la mode de donner des noms Anglais aux logiciels. En Français, cela se traduit par : Système de train d'Ulysse. Ulysse étant mon prénom. Ce n'est pas le culte de la personnalité, c'est juste que je ne trouvais rien d'autre...

Mais avant de présenter le système, examinons ce qui existe.

2.1. Digital et cantonnement

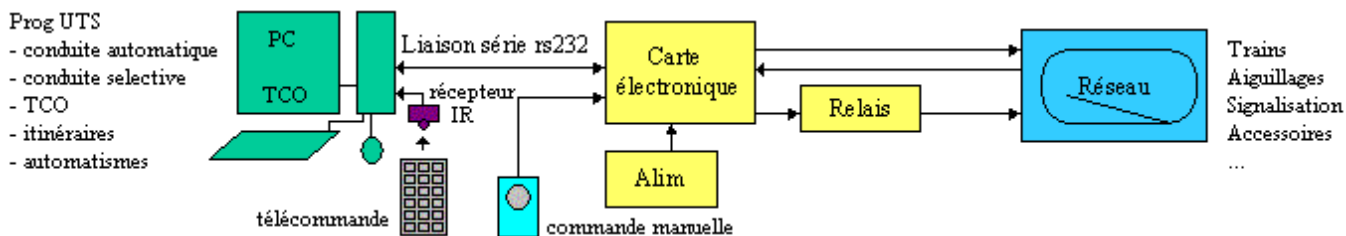
Il existe aujourd'hui 2 grandes méthodes pour piloter un réseau avec plusieurs trains:

- **Le système digital** : Cette technique permet de ne pas couper le réseau. Elle utilise des décodeurs électroniques que l'on place dans les locomotives et un système informatique qui envoie des commandes à ces décodeurs à travers la voie. Le digital est de plus en plus utilisé car il est très simple à mettre en œuvre et autorise de faire rouler plusieurs locomotives sur la même voie. Et bien entendu, il s'interface très bien avec un système informatique. Pour ceux qui souhaitent en savoir plus sur le système digital, je conseille d'aller faire un tour sur le site de la NMRA (<http://www.nmra.org>). La NMRA est chargée de normaliser le système digital. Mais le digital a malheureusement quelques points négatifs, en effet, il faut placer un décodeur dans chaque locomotive et l'apparente simplicité de câblage du système digital n'est que de la poudre aux yeux car pour repérer les trains il faut mettre des détecteurs (sur chaque canton) ! N'allez pas croire que je crache sur le système digital, je trouve que c'est une belle évolution pour les réseaux totalement manuels ou la conduite de plusieurs locomotives sur la même voie est un réel plaisir.
- **Le cantonnement** : Avec cette méthode, le réseau est coupé en plusieurs parties appelées "cantons". Il ne peut y avoir qu'un train au maximum à un moment donné sur un canton. Plusieurs implémentations de ce principe existent. La première est d'avoir autant d'alimentations que de trains en mouvement et d'utiliser des relais pour relier les alimentations aux cantons correspondants. En général, chaque canton peut être connecté sur une des alimentations. Un système informatique commute donc ces relais en fonction des détections des trains, des destinations ... Inutile de préciser que cette méthode nécessite une multitude de relais. La seconde méthode consiste à utiliser une alimentation par canton ce qui supprime les relais, mais nécessitait jusqu'à présent plus d'électronique. Il est vrai que jusqu'à présent, les systèmes de cantonnement étaient assez impressionnants (et donc rebutants) car il n'était pas rare d'avoir besoin d'une carte par canton.

J'ai choisi le principe de cantonnement car je ne souhaitais pas mettre des décodeurs dans mes locomotives. Je suis à l'échelle N et je ne vois pas comment mettre un décodeur dans une 63000 Roco. De plus l'électronique moderne associée à l'informatique permet de proposer un système simple.

2.1. UTS

Le système UTS utilise le principe de cantonnement où chaque canton possède sa propre alimentation. Grâce aux composants modernes couplés à un ordinateur, il est possible de faire tenir 16 cantons sur une carte assez petite. Donc les inconvénients habituels des systèmes à base de cantonnement sont balayés (pas besoin de relais, ni de dizaine de cartes ...).



UTS se compose d'une carte électronique reliée à un ordinateur. Il y aura donc de l'électronique et de l'informatique au programme. L'électronique sera détaillée car vous aurez à la comprendre et à la réaliser. Pour la partie informatique, je montrerai juste comment utiliser et configurer le logiciel. En effet il serait trop long de commenter les quelques 10000 lignes de code (mais rien ne vous empêche d'aller voir les sources ...). Bien entendu le système aura besoin d'une alimentation. Quelques relais pourront être nécessaires suivant ce que vous souhaitez faire. Toute l'électronique est regroupée sur une seule carte. En manuel (en fait en conduite sélective), il est possible d'utiliser un potentiomètre (commande manuelle), une télécommande de TV ou le clavier. La télécommande facultative peut être utilisée pour conduire les trains et commander les aiguillages. Si les 16 cantons de la carte ne suffisent pas alors il est possible d'utiliser 4 cartes qui se connectent au même port série. L'alimentation délivre une tension fixe pour la logique de la carte, une tension fixe pour les voies (c'est la carte qui génère les tensions variables en fait PWM pour les voies des cantons).

Maintenant que vous connaissez l'architecture du système, voici ce qu'il peut faire:

- Conduite automatique de train : Il permet de conduire automatiquement des trains sur des parcours prédéfinis en partant aux heures indiquées, aux vitesses fixées ...
- Conduite manuelle sélective : Vous pouvez conduire vos trains manuellement soit au clavier, soit avec un bon vieux potentiomètre (ce qui reste tout de même agréable) ou encore avec une télécommande de PC. Le mode sélectif permet de faire suivre les consignes du potentiomètre sur les cantons qu'emprunte le train. Ainsi vous pouvez conduire en manuel vos trains sur n'importe quel canton (tant qu'il n'y en a qu'un au maximum par canton). On rejoint presque le système digital avec ce mode car 1 potentiomètre = 1 train ! De plus le système gère correctement les boucles et retournements divers.
- Les modes automatiques et sélectifs peuvent être mixés, ce qui ouvre des fantastiques possibilités de jeu. Vous pouvez par exemple manœuvrer en sélectif des wagons amenés par une machine en automatique qui passera les reprendre en fin de journée. Et pendant vos passionnantes manœuvres en sélectif dans la cours de débord le services voyageurs continue à être assuré en automatique.
- Pour plus de réalisme, il est possible de simuler l'inertie en accélération, décélération et freinage.
- Vous pouvez créer simplement et rapidement des TCO complets et performants. Avec la représentation des voies, des voyants et des interrupteurs. Vous pourrez ainsi contrôler votre réseau comme dans la réalité.
- Tout type d'aiguillage peut être utilisé comme des aiguillages à moteur lent ou à solénoïdes. Les aiguillages sont commandés un par un pour ne pas surcharger l'alimentation. Toutes les durées sont configurables afin de ne pas cramer les aiguillages sans contacts de fin de course. Il est également possible d'utiliser une commande capacitive. Le système mémorise la position des aiguillages, donc il n'y a pas besoin d'utiliser des modèles avec un interrupteur indiquant la position.
- Des itinéraires d'aiguillage peuvent être définis, ainsi en appuyant sur un bouton, tous les aiguillages affectés à cet itinéraire se mettent dans les positions prédéfinis. Un mode plus proche de la réalité permet d'activer un itinéraire (qu'il faudra désactiver en fin), les aiguillages utilisés ne peuvent plus être utilisés jusqu'à la désactivation de l'itinéraire. Les autres demandes d'itinéraires utilisant ces aiguillages sont mis en attente jusqu'à la libération des aiguillages.
- La carte dispose d'entrées pour connecter des capteurs comme des pédales de voie, des ILS, des interrupteurs... Elle dispose également de sorties à usage générales (lampes, moteurs ...). Grâce au support d'équations et de tables de vérités, il est possible de réaliser des automatismes, gérer la signalisation ... Il est également tout à fait possible de réaliser un vrai TCO avec des LED et interrupteurs. Toutes les variables (détections des trains, aiguillages, entrées, sorties ...) sont accessibles au niveau du logiciel donc il est possible de faire à peu près tout ce que vous voulez ...
- Le système est évolutif, et donc il est possible par logiciel d'ajouter des fonctionnalités.
- Un système maison permet d'oser des bizarreries comme le déclenchement d'un vibreur en cas de détection d'un mauvais contact.
- Un horodateur permet de déclencher les déplacements automatiques ou toute autre action comme l'activation d'une sortie. Pourquoi ne pas allumer les lumières à la tombée de la nuit et éteindre celle de la pièce ...

Cet aperçu des possibilités devrait vous mettre l'eau à la bouche et explique pourquoi je ne trouvais rien sur le marché.

Il ne serait pas honnête de quitter cette présentation sans dire ce que le système ne sait pas faire (d'autant plus que je ne le vends pas). Ainsi, pour simplifier la réalisation et ne pas saucissonner le réseau, je n'utilise pas de zone d'arrêt dans les cantons. Il n'est donc pas possible d'utiliser le block système qui permet habituellement de faire circuler les trains sur une boucle sans qu'ils ne se rattrapent. Cela ne me gêne pas car je ne vois pas l'intérêt de faire tourner des trains en rond surtout que l'on peut faire bien mieux. L'absence de zone d'arrêt oblige à utiliser des temporisations pour l'arrêt en automatique, ce qui laisse une incertitude d'arrêt de l'ordre de 10cm. Mais ceci peut être corrigé en ajoutant une détection comme une pédale de voie.

Les chapitres suivants montreront comment construire ce système du point de vue électronique et comment utiliser le logiciel.

3. Electronique

Cette partie explique le fonctionnement et la réalisation de la partie électronique.

3.1. Notions d'électronique

Afin de comprendre les explications électriques sur la réalisation, il est nécessaire d'avoir un minimum de connaissances en électronique. Et comme un modélisme ferroviaire n'est pas forcément un électronicien je vais rappeler quelques notions et essayer de faire sentir quelques règles. Si vous avez déjà des notions, vous pouvez sauter ce chapitre. Je n'utiliserai pas forcément des définitions rigoureuses, le but étant de faire comprendre et sentir en un minimum de temps.

Tension, Courant, Puissance, Energie :

L'électricité traite des charges électriques, mais malheureusement ces charges sont invisibles c'est pourquoi il est intéressant de faire l'analogie avec l'hydraulique qui est l'étude des liquides ou des gouttes d'eau dans notre cas. Il était une fois une charge et une goutte d'eau ... Pour déplacer de l'eau par exemple pour faire tourner la roue d'un moulin, il faut une différence de hauteur. Et bien c'est pareil en électricité, il faut une différence de potentiel encore appelé voltage dont l'unité est le volt V.

Une autre grandeur importante en hydraulique est le débit qui est la quantité de liquide qui passe à un endroit en un temps donné. Ce débit s'exprime par exemple en mètre cube par seconde ou litre par minute ... Ce débit, encore appelé courant peut facilement être transposé dans le domaine électrique, il représente alors le nombre de charge qui passent dans une section dans un temps donné. Son unité est l'ampère A. En électronique, on parle souvent de mA car l'ampère est une grosse unité. Il faut 1000mA pour faire un ampère. $1\text{mA}=0.001\text{A}$; $1\text{A}=1000\text{mA}$.

La puissance est liée à la hauteur et au débit, en effet plus quelque chose tombe de haut et plus ça fait mal, et plus il en tombe également. Transposé à l'électricité, on en déduit la formule de la puissance : $P=UI$. La puissance s'exprime en Watt W. Par exemple une locomotive qui consomme 200mA sous 12V consomme $P=UI=12*0.200=2.4\text{W}$

Tant qu'on y est, parlons de l'énergie. L'énergie permet de faire quelque chose. On voit de suite qu'elle est liée à la puissance consommée et au temps d'utilisation. D'où : $E=Pt$. Elle s'exprime en Watt Heure Wh. Attention pas Watt par heure mais Watt heure. Par exemple une batterie de 12V de chargée à sa capacité max de 6Wh pourra alimenter un moteur consommant 1A sous 12V pendant : $E=Pt$; $t=E/P$; $t=E/(UI)$; $t=6/(12*1)=3$ heures.

Une petite application pratique au fils. Plus la tension est grande et plus il faut d'isolant. Pour vous en convaincre, regardez l'épaisseur des isolateurs des supports de caténaire 1500V et 25000V. Plus l'intensité est grande et plus le diamètre doit être important. Regardez la lourde caténaire 1500V et la légère caténaire 25000V. Calculons le courant qui y passe lorsqu'une machine type BB22200 de 5900kW (8000ch) circule à pleine puissance $P=UI$; $I=P/U$; sous 1500V : $I=5900000/1500=3933\text{A}$; sous 25000V : $I=5900000/25000=236\text{A}$.

Résistance :

Une résistance est un composant à 2 pattes qui lie le courant et la tension avec la loi suivante : $U=RI$. R est sa résistance et son unité est l'ohm et son symbole omega. Une résistance transforme la puissance en chaleur. A notre niveau, nous verrons les moteurs et lampes comme des résistances mais transformant respectivement l'énergie en mouvement mécanique et lumière. En analogie électrique, une résistance peut se voir comme un tuyau, plus le tuyau est gros (et donc sa résistance au passage de l'eau faible) et plus le débit est important pour une différence de hauteur donnée.

Par exemple une lampe qui consomme 100mA sous 12V à une résistance interne de $U=RI$; $R=U/I=12/0.100=120$ Ohms.

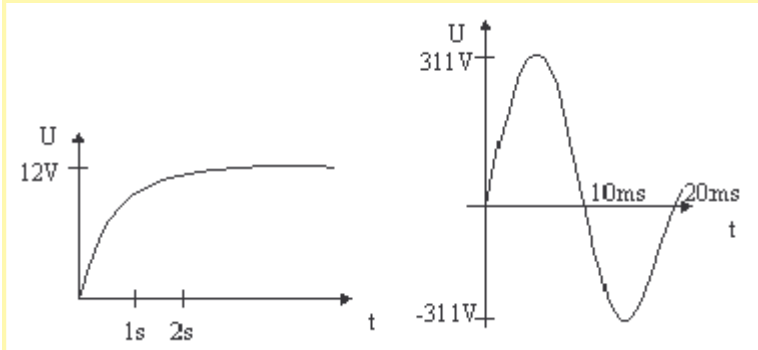
Le corps humain est aussi assimilable à une résistance d'environ 1000Ohms lorsqu'on est mouillé, sachant que ça devient mortel lorsqu'un courant de plus de 30mA traverse le corps, il vaut mieux ne pas s'exposer à plus de $U=RI=1000*0.03=30\text{V}$. (48V en alternatif). Donc pas de problème en modélisme ferroviaire où le 12V est inoffensif.

Maintenant calculons, le courant dans un fil de 0.1Ohms que l'on utilise pour court-circuiter une batterie de 12V. $U=RI$; $I=U/R=12/0.1=120\text{A}$ Le fil crame et souvent la voiture qui va avec aussi d'où l'intérêt des fusibles.

Calculons la résistance à mettre en série avec une led qui doit être alimentée en 2V sous 10mA si on dispose d'une alimentation de 12V. La résistance doit donc avoir $12-2=10V$ a ses bornes et est également traversée par 10mA, d'où $R=U/I=10/0.010=1000\text{Ohms}$. La puissance dissipée dans la résistance est de $P=UI=10*0.010=0.1W$ donc une résistance standard d'un quart de watt 0.250 suffit.

Diagrammes temporels :

Un diagramme temporel est la représentation d'une grandeur en fonction du temps. Le diagramme suivant montre l'évolution de la tension aux bornes d'un condensateur lorsqu'on le charge à travers une résistance et un signal alternatif comme celui que l'on voit sur le secteur.



On voit par exemple que ce condensateur est chargé à 12V au bout de 2s. La tension du secteur passe à 0V, puis à 311V au bout de 5ms puis à 0V puis à -311V puis à 0V et le cycle recommence. Ce type de signal est périodique car il se reproduit à l'infinie. Cette forme de signal en sinusoïde est représentative du courant alternatif. La sinusoïde se reproduit toute les 20ms, on parle de période de $T=20\text{ms}$ et donc de fréquence $F=1/T=1/0.020=50\text{Hz}$. 1 hertz Hz correspond à une fois par seconde. On retrouve bien le 50Hz du réseau EDF. Et le alors, où est le 220V? Et bien c'est la tension en continue qui produirait la même énergie. Du point de vue énergétique un radiateur alimenté en alternatif chaufferait de la même manière que s'il était alimenté en continue avec 220V.

Etats logiques :

En électronique numérique, on préfère travailler avec des états, par exemple en alimentant les circuit en 5V on parle d'état bas ou 0 sous 2.5V et d'état haut ou 1 au dessus de 2.5V. On parle de logique booléenne ou binaire car il n'y a que 2 états.

Quelques Composants :

Enfin nous terminerons ces notions par quelques composants :

- **La diode** est un composant à 2 pattes qui ne laisse passer le courant que dans un seul sens. La tension à ses bornes est de 0.6V lorsque le courant passe. L'entrée du courant s'appelle l'anode et la sortie la cathode. La cathode est repérée par un trait.
- **Le pont de diode** est un ensemble de 4 diodes qui permet de redresser un signal, c'est à dire de transformer les tensions négatives en positives (donc le courant circule dans le même sens).
- **La LED ou DEL** en Français pour diode électro-luminescente. C'est une diode qui s'allume lorsque le courant la traverse. La tension de service est de 2V environ et le courant de 10mA mais ca dépend des modèles. La plus grande patte est le + (anode) et le chanfrein le - (cathode).
- **Le condensateur** est un réservoir d'énergie. Il est utilisé pour stocker de l'énergie temporairement dans les alimentations, faire des oscillateurs, éliminer les parasites. Il est recommandé d'en mettre un de 100nF sur chaque circuit intégré actif. La capacité d'un condensateur s'exprime en Farad F. Cette valeur est énorme c'est pourquoi on parle souvent de microFarad μF ou nano Farad nF ($1\text{F}=1000000\mu\text{F}=1000000000\text{nF}$). Les gros condensateurs sont ronds et chimiques et il faut les brancher à l'endroit sous peine d'explosion. (Une bande de - indique la patte -, l'autre patte étant le +). Les petits condensateurs que nous utiliserons seront dit plastiques et ne sont pas polarisés. Enfin lors de l'achat il faut préciser la tension maximale de fonctionnement du condensateur. Par exemple : chimique 1000 μF 16V, plastique 100nF 63V...
- **Le transistor** est un composant à 3 pattes qui permet d'amplifier un courant qui passe entre la base et l'émetteur. Le transistor est capable de faire passer un courant maximal entre le collecteur et l'émetteur égale au courant de base (ou de commande) multiplié par le gain du transistor.
- **Le quartz** est un composant qui permet de produire des vibrations très précises.
- **L'oscillateur céramique** : Un quartz un peu moins précis.

- **Le circuit intégré** est un composant avec plein de pattes avec un mélange de composants en interne. Les pattes sont numérotées dans le sens des aiguilles d'une montre de 1 jusqu'au nombre de broches. La patte une est souvent indiquée par un petit point. Quand on lit l'inscription horizontalement, elle se trouve en bas à gauche. Généralement ces circuits intégrés ou CI s'alimente sous 5V du moins ce sera notre cas.
- **Le régulateur** est un circuit qui permet de fournir une tension fixée en sortie quelque soit la tension d'entrée (Elle doit tout de même être supérieure de 2V pour la plupart des régulateurs).
- **Le relais électromagnétique** est un ensemble d'interrupteurs qui sont commandés par une bobine. Lorsque la bobine est alimentée et alors par effet magnétique des interrupteurs se ferment et d'autres s'ouvrent. Cela permet de commander de grosses charges, en isolant la source et la commande.

Vous voilà fin prêt pour comprendre la suite du document. Il est à noter que des ouvrages existent sur ces sujets.

3.2. Les alimentations

Tout montage électronique à besoin d'une alimentation pour fonctionner correctement, nous allons voir dans cette partie comment alimenter l'électronique et le réseau avec ce dont vous disposez. Avant ceci regardons ce qu'il y a dans une alimentation linéaire. (Il existe un autre type d'alimentation : celles à découpage mais elles sortent du cadre de cette documentation).

Eléments d'une alimentation linéaire

Une alimentation ou alim comprend les éléments suivants :

- Fusible : Pour éviter à l'alimentation de prendre feu en cas de problème
- Transformateur : Pour abaisser la tension par exemple de 220V à 15V. Il est possible de s'arrêter ici pour une tension alternative.
- Pont de diode : Pour redresser le courant.
- Condensateur de filtrage : Pour s'approcher du continu. Le condensateur se charge pendant les impulsions et restitue une partie de son énergie lors des trous. Il est possible de s'arrêter ici pour une tension continue non régulé.
- Régulateur : Permet de réguler la tension à une valeur fixée. Le régulateur peut être fixe ou ajustable. Sur la plupart des régulateurs, la tension d'entrée devra être plus grande d'au moins 2V que la tension de sortie afin que le régulateur régule convenablement. Le régulateur est souvent protégé contre les courts circuits. (Ce n'est pas le fusible qui grille mais le régulateur qui coupe la sortie). C'est un composant qui chauffe beaucoup car il doit convertir en chaleur la différence de tension entre l'entrée et la sortie multipliée par le courant. C'est pourquoi, on les monte souvent sur des radiateurs.

Les aiguillages et les lampes à solénoïde fonctionnent en alternatif ou continue non régulé ou régulé.

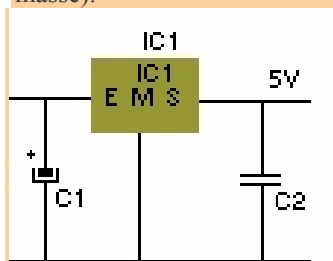
Les moteurs sont souvent des moteurs à courant continu et fonctionnent alors en continue non régulé et régulé.

L'électronique plus sensible n'accepte quant à elle que le continu régulé.

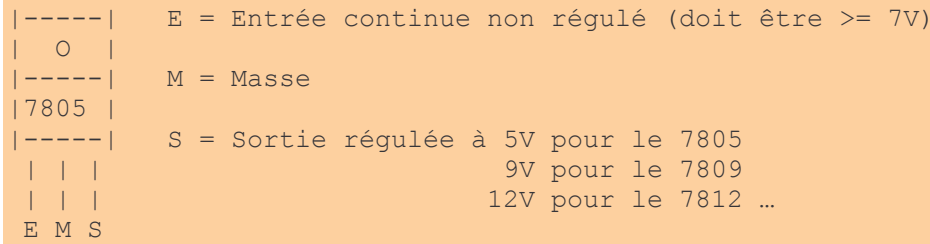
Notre alimentation

Il vous faut du 5V continue fixe régulé pour l'électronique. Entre 9V et 15V continue fixe régulé ou non pour les voies. Plus une tension susceptible de faire fonctionner vos accessoires. Le plus simple pour un modélisme ferroviaire est de récupérer un transfo ferroviaire qui fournit une sortie voie et accessoire qui fonctionne bien avec votre réseau et de rajouter un régulateur 7805 pour fournir du 5V régulé à l'électronique. sinon vous pouvez aussi vous faire votre propre alim, utiliser des blocs secteurs, une alim de PC ... Veuillez juste à disposer d'une puissance suffisante.

La figure suivante montre comment obtenir du 5V régulé à partir d'une tension continue d'au moins 7V en entrée. On peut prendre C1=1000uF en 16V, C2=100nF, IC1=7805. Attention à ne pas brancher le condensateur de filtrage C1 à l'envers car sinon il explose et ça pue pendant des jours. Si vous avez peur de brancher le montage à l'envers alors rajoutez une diode en entrée. Pensez à mettre le 7805 sur un radiateur pour qu'il puisse dissiper. Notez que la languette de refroidissement du 7805 (la partie où il y a le trou qui sert à fixer le 7805 sur un radiateur est reliée électriquement à la patte du milieu, c'est à dire la masse).

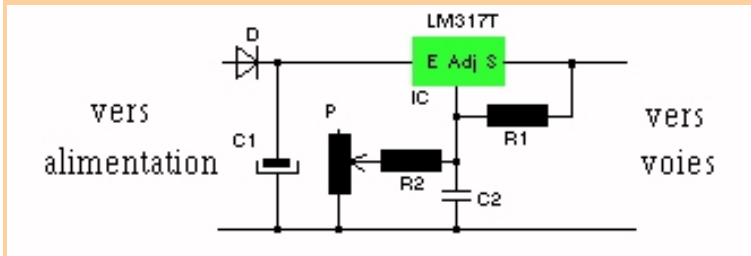


Le 7805 à la forme suivante :



Le courant maximal de sortie est de 1.5A. Le 7805 est protégé contre les court circuit et contre la surchauffe. Si on note vin la tension d'entrée, Vout la tension de sortie et Is le courant de sortie, alors la dissipation thermique du 7805 en Watt est de $P=(V_{in}-V_{out})\cdot I_s$.

Tant qu'on parle des régulateurs, notez qu'il est également possible d'utiliser des régulateurs réglables comme le LM317T



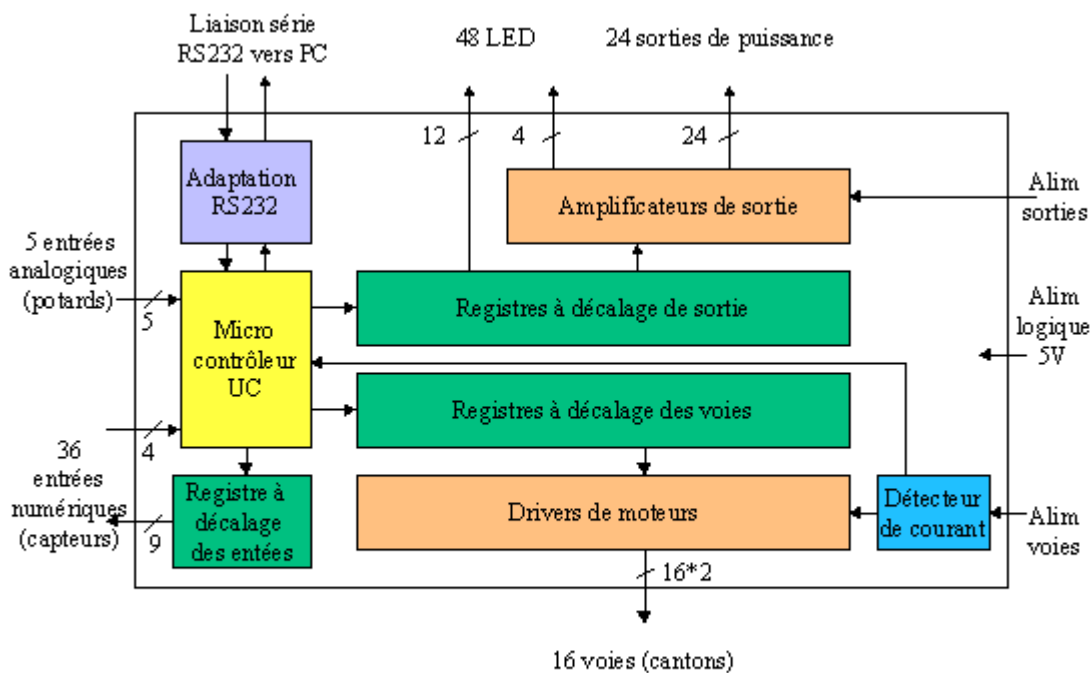
$$V_{outmax}=1.25+1.25\cdot((R2+P)/R1)$$

$$V_{outmin}=1.25+1.25\cdot(R2/R1)$$

P est un potentiomètre, c'est à dire une résistance de valeur P entre les 2 pattes extrêmes avec une 3eme patte relié au curseur. Ainsi entre une patte et le curseur, la résistance peut être réglée entre 0 et P Ohms. Il existe des potentiomètres linéaires ou rotatifs.

3.3. Synoptique de la carte électronique

Maintenant que nous avons une alimentation, il est tant de passer à l'étude de la carte électronique. La figure suivante montre le synoptique de la carte, c'est à dire une représentation par blocks fonctionnels:



Synoptique de la carte électronique du système UTS

On y trouve les groupes suivants :

- **Le microcontrôleur ou UC** est le cœur du montage, en effet c'est lui qui contrôle tout ce qui se passe sur la carte. Un microcontrôleur est un véritable petit ordinateur qui tient sur une puce. Le UC exécute un programme stocké dans sa mémoire interne et peut venir lire l'état de ses pattes et le modifier. Sans lui il aurait fallu mettre sur la carte une centaine de composants supplémentaires pour faire la même chose. Dans notre cas, le programme du UC, dialogue avec le PC, et lui indique la position des potentiomètres (potards), la détection sur les cantons, l'état des capteurs et prend attend les consignes pour fixer les vitesses et sens sur les différents cantons ainsi que l'état des sorties.
- **La partie interface RS232** permet d'interfacer les niveaux RS232 qui sont de -12V pour l'état 1 et +12V pour l'état 0 en niveaux corrects pour le UC qui sont de 0V pour l'état 0 et 5V pour l'état 1.
- **Les registres à décalage** permettent d'étendre les nombres de sorties disponibles sur le UC. En effet les différents groupes de registres à décalages apportent 81 sorties en utilisant seulement 8 sorties au niveau du UC.
- **Le groupe amplificateur de sortie** autorise de piloter directement des charges jusqu'à 500mA, ce qui est suffisant pour y brancher directement des relais, des moteurs et des petits aiguillages.
- **Les drivers de moteurs** permettent de mettre 0V, ALIM voies ou - ALIM voies sur chaque canton sous une intensité maximale de 700mA.
- **Le détecteur de courant** permet de voir s'il y a consommation de courant sur les voies, ce qui permet de repérer les trains sans matériel supplémentaire.

La carte possède les connexions suivantes :

- **L'alimentation logique 5V** est nécessaire pour alimenter l'électronique de la carte. Il est nécessaire de lui fournir du 5V continu régulé (avec un 7805 par exemple).
- **L'alimentation voies** servira pour alimenter les voies. On peut par exemple lui fournir du 12V continue régulé ou non.
- **L'alimentation des sorties** est juste redirigée vers les connecteurs de sortie afin de pouvoir être utilisée pour alimenter les charges de sortie. Cette tension doit être continue.
- **5 entrées analogiques** permettent de connecter des potentiomètres ou capteurs divers comme un capteur pour connaître l'éclairage de la pièce et déclencher les lumières des habitations ...
- **36 entrées numériques matricées** (9 sorties qui rebouclent à travers les capteurs sur 4 entrées) permettent d'utiliser des capteurs numériques comme des pédales de voie, des ILS, des boutons poussoirs, des inverseurs ...
- **24 sorties de puissances** permettent de connecter des relais, moteurs, aiguillages, lampes ...
- **48 LED** peuvent être gérées afin de réaliser des feux de signalisations par exemple. Le matricage est utilisé afin de piloter 48 LED avec seulement 16 sorties. Si vous ne souhaitez pas utiliser les LED, il est possible de les récupérer 16 sorties.
- **16 cantons** peuvent être connectés. Il suffit de relier chaque voie par 2 fils sur la carte. Sur chaque canton le UC est capable de fixer un sens, une vitesse et de détecter la présence d'une locomotive.
- Enfin **l'interface RS232** permet au PC de commander la carte en dialoguant avec le UC.

3.4. Les CI de la carte électronique

Cette partie décrit un par un les 5 types de circuits intégrés (CI) utilisés sur la carte :

- le **UC 16F873** pour gérer la carte
- le **MAX232** pour l'adaptation des signaux RS232
- les registres à décalage **4094** pour étendre les sorties du UC.
- Les **ULN2803** pour fournir des sorties de puissances jusqu'à 500mA
- Les drivers de moteurs **L293D** pour piloter les moteurs des locomotives.

A. Le microcontrôleur PIC 16F873 :

La famille PIC

Le UC est un véritable ordinateur qui tient sur une puce. Il exécute un programme contenu dans sa mémoire interne et utilise ses entrées et sorties pour contrôler la carte. Le UC possède également des périphériques internes comme un convertisseur analogique numérique qui servira à convertir les tensions en sortie des potentiomètres en valeurs numériques, un contrôleur rs232 utilisé pour communiquer avec le PC... Ce UC fait partie de la famille PIC fabriquée par Microchip. Toute la documentation sur ces UC peut être trouvée sur le site web du fabricant <http://www.microchip.com>. Les PIC sont des UC très populaires et faciles à se procurer et à programmer. Aussi beaucoup de sites web leurs sont dédiés et trouver des informations et de l'aide est toujours possible. Je conseille par exemple de lire les excellents cours de BIGONOFF sur le site : <http://www.abcelectronique.com/bigonoff>. La famille des PIC compte de nombreux composants dont les plus connus sont sans doutes les PIC16F84 et PIC16F876.

Programmation des PIC

Comme je l'ai déjà mentionné le UC exécute un programme qui est stocké dans sa mémoire interne, il faut donc que vous mettiez ce programme dans cette mémoire. C'est la phase de programmation du UC. Pour ce faire, je conseille de réaliser le programmeur JDM qui est une toute petite carte à brancher sur un port série. Je conseille d'utiliser le célèbre logiciel de programmation IC_PROG. Vous pourrez trouver des informations sur JDM + ICPROG n'importe où sur Internet (ou sur mon site). Je recommande ces 2 produits car le soft est simple d'utilisation et d'installation et la carte facile à réaliser et fiable. Une fois le JDM fabriqué et ICPROG installé, il suffit de placer le PIC sur le support, configurer ICPROG, sélectionner le fichier à charger (873rs12_0.hex disponible dans l'archive ZIP que je propose.), fixer les fusibles et lancer la programmation:

Si vous voulez voir à quoi ressemble le programme en assembleur ou si vous voulez le modifier, éditez le fichier 873rs11.asm.

Si vous utilisez plusieurs cartes, alors flasher le bon fichier en fonction de l'adresse de la carte

- carte0 : 873rs12_0
- carte1 : 873rs12_1
- carte2 : 873rs12_2
- carte3 : 873rs12_3

Si vous utilisez la version précédente 873rs11.hex, il n'est pas nécessaire de reflasher la carte, mais vous ne pourrez utiliser qu'une seule carte.

icprog

icprog est un freeware créé et maintenu gratuitement par Bonny Gijzen. Il qui permet de programmer facilement entre autre les PICs. Vous pouvez télécharger la dernière version et trouver des informations sur <http://www.ic-prog.com>.

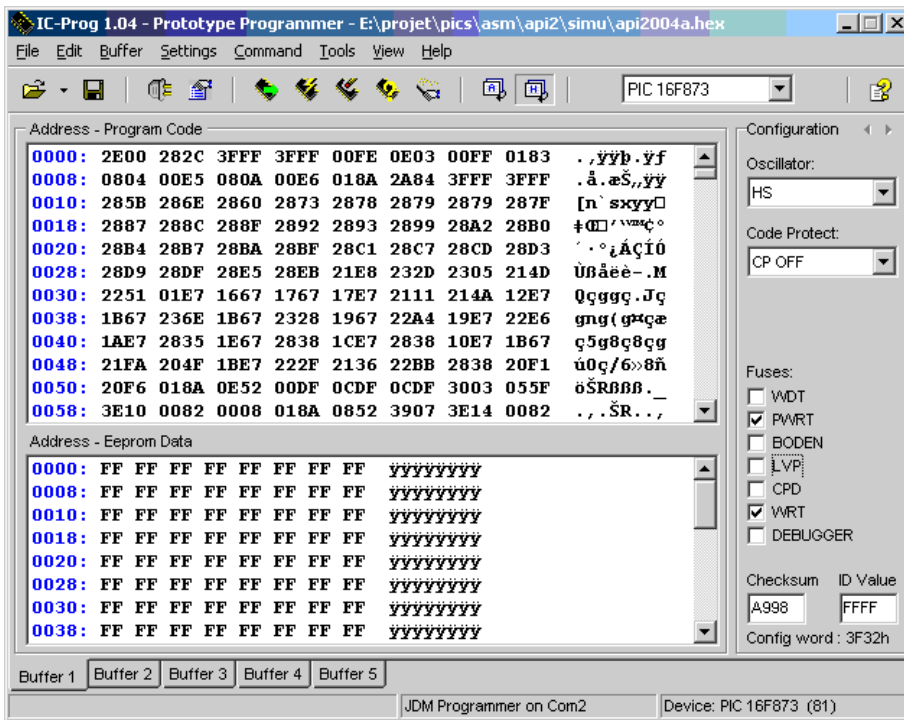
Pensez à configurer correctement le soft :

- Device = PIC 16F873 (convient aussi pour les autres PIC et programme uniquement les 4 premiers kw de flash ce qui divise par 2 le temps des UC avec 8kw).
- Hardware = JDM (sans inversions)
- Connexion : Indiquez votre port, souvent COM1 ou COM2 et choisissez Windows API pour que le programmeur fonctionne sans problèmes sous Windows 98, 2000 et XP car les registres ne sont pas directement accessibles sous ces OS.
- Laissez le délai à 10.
- Vérification : Durant la programmation afin de détecter un problème le plus rapidement possible.

Pour programmer un firmware (fichier .hex) :

- Choisir le UC
- Ouvrir le fichier
- Sélectionner les options de configuration du PIC comme suit :
 - oscillateur : HS (High Speed : Quartz > 4MHz)
 - code protect : CP OFF (Pas de protection de code)
 - WDT : 0 (pas de chien de garde)
 - PWRT : 1 (timer de mise en marche)
 - BODEN : 0 (pas de remise à zéro si la tension d'alimentation baisse)
 - LVP : 0 (pas de programmation basse tension)
 - CPD : 0 (pas de protection des données)
 - WRT : 0 (interdiction pour le programme d'écrire dans sa propre mémoire)
 - DBUGGER :0 (pas de debugger)

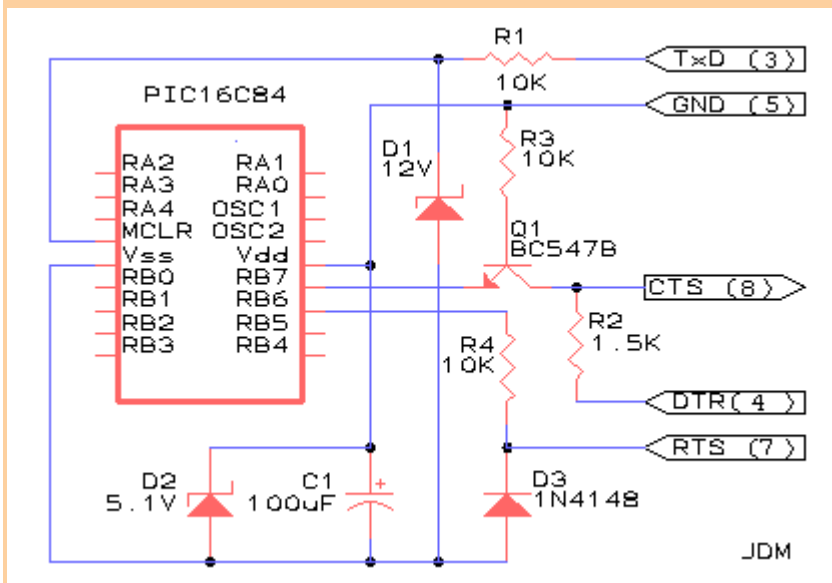
La copie d'écran suivant montre icprog



JDM

Cette carte de programmation se branche sur un port série et ne connaît pas les problèmes des autres programmeurs sur port série car la masse du microcontrôleur est mise a -5V afin de garantir plus de 12V sur la broche reset (qui permet la programmation).

La figure suivante montre la carte JDM pour un PIC16F84 avec un connecteur série DB9



Pour utiliser un vieux connecteur DB25, changer TXD 3-->2, GND 5-->7, CTS 8-->5, DTR 4-->20, RTS 7-->4.

Pour info, les numéros des broches sont gravées sur les connecteurs. Les ports séries sont des connecteurs DB9 ou DB25 mâles a l'arrière des PC.

Pour programmer les UC qui nous intéressent, il suffit juste d'adapter le support.

Par exemple pour les microcontrôleurs flash a 28 broches (16F873 16F876) ou à 40 broches, adapter le connecteur au brochage suivant (et mettre une résistance de 270Ohms entre RB3 et la masse pour ne pas avoir de problème avec la programmation basse tension).

PIC 18 broches PIC 16F84 16F628	PIC 28 broches PIC 16F873 876	PIC 40 broches PIC 16F874 877
RA2 -## #- RA1	==>MCLR -## #- RB7<===	==>MCLR -#### #- RB7<===
RA3 -#####- RA0	RA0 -#####- RB6<===	RA0 -#####- RB6<===
RA4 -#####- OSC1	RA1 -#####- RB5	RA1 -#####- RB5
==>MCLR -#####- OSC2	RA2 -#####- RB4	RA2 -#####- RB4
===>VSS -#####- VDD<===	RA3 -#####- RB3<===	RA3 -#####- RB3<===
RB0 -#####- RB7<===	RA4 -#####- RB2	RA4 -#####- RB2
RB1 -#####- RB6<===	RA5 -#####- RB1	RA5 -#####- RB1
RB2 -#####- RB5	VSS -#####- RB0	RE0 -#####- RB0
RB3 -#####- RB4	OSC1 -#####- VDD<===	RE1 -#####- VDD<===
	OSC2 -#####- VSS<===	RE2 -#####- VSS<===
	RC0 -#####- RC7	VDD -#####- RD7
	RC1 -#####- RC6	VSS -#####- RD6
VSS=GND	RC2 -#####- RC5	OSC1 -#####- RD5
	RC3 -#####- RC4	OSC2 -#####- RD4
		RC0 -#####- RC7
		RC1 -#####- RC6
		RC2 -#####- RC5
		RC3 -#####- RC4
		RD0 -#####- RD3
		RD1 -#####- RD2

Le PIC 16F873

J'ai choisit le PIC 16F873 car :

- Il fallait un UC avec suffisamment d'entrées/sorties. Ce UC pour 28pattes dispose de 22 E/S.
- Il fallait une mémoire FLASH afin de pouvoir le programmer plusieurs fois. En effet la mémoire FLASH peut être reprogrammée et ne s'efface pas lorsque l'on coupe l'électricité. Ainsi s'il y a besoin de rajouter des fonctions, il suffira de reprogrammer le UC, on parle aussi de flasher le UC.
- Il fallait un convertisseur analogique numérique
- Il fallait un contrôleur RS232

Avec toutes ces exigences seuls les PIC16F873,876,874et877 convenaient. J'ai retenu les 873 et 876 car je n'avais pas besoin des 33 E/S (40 pattes) des 874 et 877. Le 876 est une version avec plus de mémoire que le 873 et peut donc être indifféremment utilisé à la place du 873. Pour obtenir un UC, vous pouvez soit en acheter un chez un revendeur d'électronique, soit demander des échantillons sur le site de microchip. Le Uc doit fonctionner à 8 MHz (En fait les instructions sont exécutées 4 fois moins vites), alors éviter de choisir la verion qui supporte 4 MHz max (même si j'en ai overclocké sans probleme à 8MHz).

Brochage du 16F873

```

          + G
B B B B B B B B 5 N C C C C   GND=masse
7 6 5 4 3 2 1 0 V D 7 6 5 4   OSC=oscillateur
| | | | | | | | | | | | | |   /RAZ=RESET au niveau bas
#####                       A0...A5,B0...B7,C0...C7 = E/S Entrées/Sorties
#                               #
#           16F873             #   résonateur           quartz #####
#o                               #   #####           #####
#####                       #####
| | | | | | | | | | | | | |   | | |
/ A A A A A A G O O C C C C   O G O           GND-- | |--+   +-- | |--GND
R 0 1 2 3 4 5 N S S 0 1 2 3   S N S           O O
A           D C C             C D C           S S
Z                               C C

```

Pour fonctionner un Pic à besoin d'un oscillateur pour faire battre son cœur. Afin d'être précis, nous utiliserons soit un quartz, soit un résonateur céramique. Le résonateur céramique prend moins de place est est moins précis (mais sa précision est suffisante pour notre application). Pour un quartz, il faut rajouter 2 condensateurs céramiques entre 15 et 33pF.

Enfin il faut relier sa patte reset (raz) au +5V par l'intermédiaire d'une résistance de 1kOhms. Ceci initialisera automatiquement le UC à la mise sous tension.

La figure suivante montre le brochage du UC, une fois programmé :

C	L	D	D	D	L	C	+	G		C	D	CKO : clock (horloge) pour les reg à dec des sorties		
K	D	T	T	E	D	K	N	5	N	R	T	K	T	LDO : load (chargement) pour les reg à dec des sorties
O	O	O	V	T	V	V	A	V	D	X	X	I	I	DT0 : data (données) pour les reg à dec des sorties
														CKV/LDV/DTV : idem pour les reg à dec des voies
#####													CKI/DTI : idem pour le reg à decalage des entrées.	
#												#	IN0...3 : Entrées numériques matricées	
#	16F873											#	POT0...4 : Entrées analogiques (pour les potentiomètres)	
#	O											#	CLI : sortie clignotante (pour voir si le UC marche)	
#####													RX : réception RS232	
														TX : transmission RS232
/	P	P	P	P	C	P	G	O	O	I	I	I	I	OSC : oscillateur
R	O	O	O	O	L	O	N	S	S	N	N	N	N	GND : masse
A	T	T	T	T	I	T	D	C	C	0	1	2	3	+5V : alimentation du UC
Z	0	1	2	3	4								/RAZ : Reset : pour initialiser le UC	

Pour tester le UC, il suffit de l'alimenter en 5V, brancher son oscillateur et la résistance de reset, mettre un condensateur de découplage entre la masse et le 5V et brancher une LED en série avec une résistance de 470Oms entre le 5V et la patte CLI.

Voilà vous en savez assez sur le UC

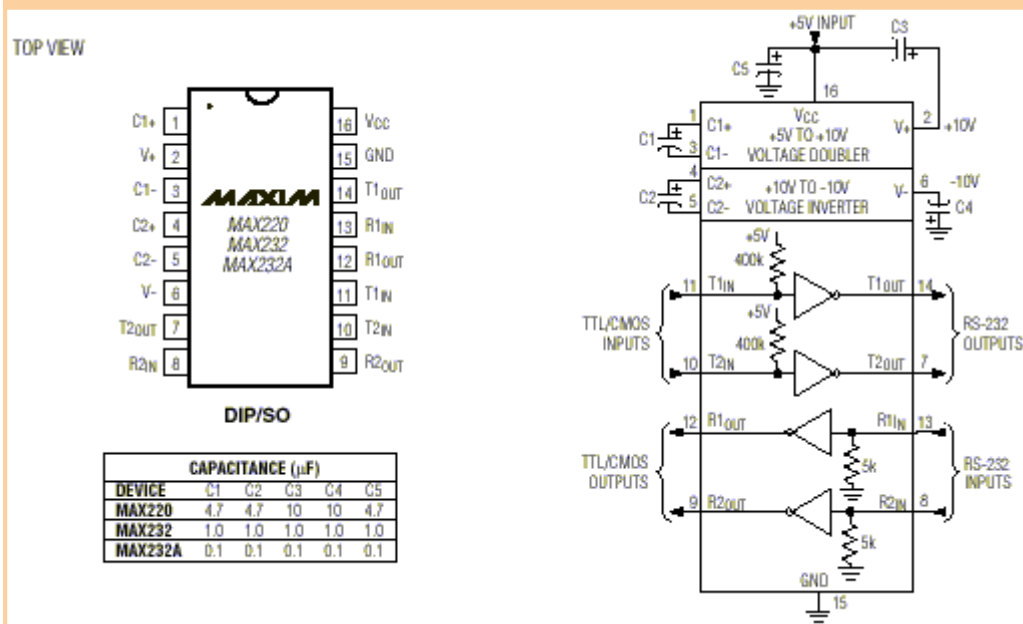
B. Le MAX232

La RS232 est une liaison standard qui remonte à l'origine du PC. Elle permet de transférer des informations entre un PC et tout type d'appareil. Sur cette liaison, les informations transitent en série c'est à dire les unes après les autres. Pour en savoir plus regardez sur Internet. Nous utiliserons cette liaison pour commander notre carte à l'aide du PC. Les PC en possèdent généralement 2 accessibles chacune grâce à un connecteur à 9 points mâle (DB9) à l'arrière du PC. Les vieux ordinateurs possèdent un connecteur à 9 points mâle et un connecteur à 25 points mâle (Ne pas confondre avec le port parallèle qui lui est à 25 points femelle). Les numéros des pattes sont gravés sur les connecteurs DB et on à :

- TX PC DB9: patte3 DB25:patte2
- RX PC DB9: patte2 DB25:patte3
- GND DB9: patte5 DB25:patte7

Les autres pattes ne seront pas utilisées.

Le MAX232 permet d'adapter les signaux RS232 12/-12V aux signaux 0/5V du UC. Ce composant fabrique le +12V et -12V (en fait du +9 et -9) à partir du 5V de la carte en utilisant une pompe de charge qui nécessite 4 condensateurs. Le MAX232 possède 2 émetteurs et 2 récepteurs. La figure suivante présente ce composant :



Nous n'aurons besoin que d'un émetteur et d'un récepteur, par exemple ceux qui sont placés entre les pattes 10,7 et 8,9. Les connexions suivantes devront être réalisées entre le UC et un connecteur DB9 à raccorder au PC. Les numéros des pattes sont marqués sur les connecteurs 9 points.

```

UC TX -- > (10)MAX232 (7) -- > DB9 (2) PC RX
UC RX < -- ( 9)MAX232 (8) < -- DB9 (3) PC RX
Masse de la carte ----- DB9 (5) GND

```

Pour relier plusieurs cartes (au maximum 4) au même port série, il faut juste utiliser une diode supplémentaire par carte. Chaque carte est identifiée par une adresse 0 à 3, il faut donc flasher chaque carte avec le programme assembleur qui contient la bonne adresse. Si le montage ci-dessous ne fonctionne pas, alors essayer de rajouter une résistance de 2200 ohms en parallèle avec une des diodes.

```

UC1 TX --> (10)MAX232 (7) --|>|---+----> DB9 (2) PC RX
                                     1N4148 |
UC2 TX --> (10)MAX232 (7) --|>|---+
                                     1N4148

UC1 RX <-- ( 9)MAX232 (8) <-----+-----< DB9 (3) PC RX
                                     |
UC2 RX <-- ( 9)MAX232 (8) <-----+

Masse de la carte 1 -----+----- DB9 (5) GND
                             |
Masse de la carte 2 -----+

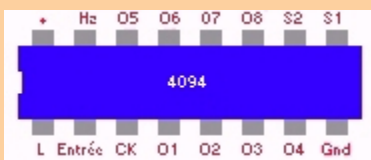
```

C. Les registres à décalage 4094

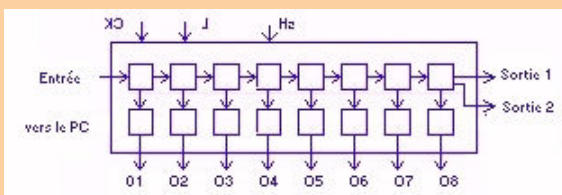
Les registres à décalage nous permettent d'obtenir autant de sortie que l'on veut en utilisant seulement 3 sorties du UC. La carte utilise 3 groupes de registres à décalage :

- Le 1^{er} pour les sorties (dont les LED)
- Le 2^{eme} pour les voies (cantons)
- Le 3^{eme} pour le matricage des entrées numériques

Le circuit utilisé pour réaliser ces registres à décalage est le 4094 dont le brochage est le suivant :



Et le diagramme fonctionnel :



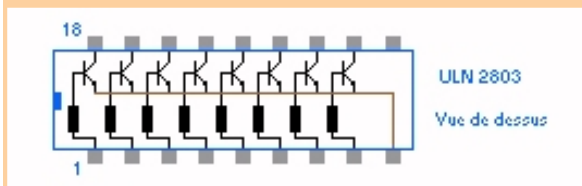
A chaque front positif (passage du niveau bas au niveau haut) d'horloge sur CK ; la donnée présente en entrée est mise dans la première bascule (carré en haut à gauche) alors que l'ancienne donnée de la 1^{ere} va en 2 ... Les données se décalent donc. Lorsque la patte L pour Load (chargement) est à l'état haut alors les données des bascules de décalages sont copiées dans les bascules de sorties qui se retrouvent sur les pattes O1 à O8. Au niveau bas, les sorties sont mémorisées et indépendantes de l'état des bascules de décalages. Hz permet de d'activer les sorties lorsqu'il est à l'état haut, nous le mettrons donc à l'état haut. La pattes S2 permet de chaîner les registres à décalage en la connectant à l'entrée du registre suivant. Nous n'utiliserons pas S1. nous utiliserons donc les 4094 comme suit :

- Décalage des données présentées sur l'entrée avec des impulsions sur l'horloge
- Transferts des données sur les sorties avec une impulsion sur la patte Load.

Pour le registre des entrées, nous laisserons Load à l'état haut afin d'économiser une patte car ce n'est pas gênant de voire les données se déplacer en sortie alors que c'est inacceptable pour les 2 autres groupes de registres.

D. Les ULN 2803

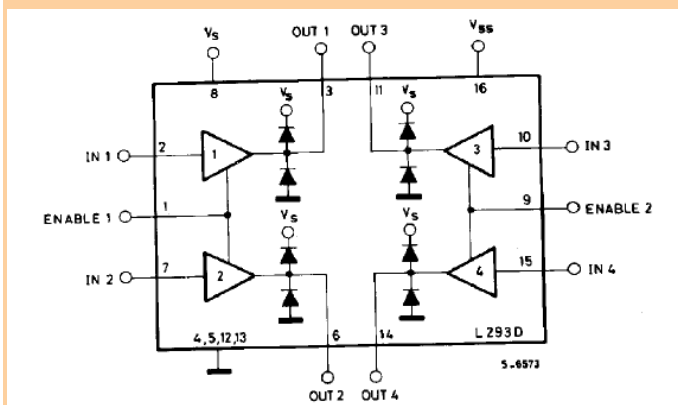
Ces réseaux de transistors darlington permettent de commander à partir d'un faible courant disponible en sortie des 4094 des charges jusqu'à 500mA. Lorsque l'on met à l'état haut une des entrées, alors la sortie correspondante est reliée à la masse grâce à un transistor de puissance. Lorsqu'il n'y a pas de commande ou qu'elle est à l'état bas, alors le transistor ne conduit pas et la sortie est en l'air. Il faut donc voir chaque sortie comme un interrupteur relié à la masse qui se commande par l'entrée correspondante. La figure suivante détaille un ULN2803 :



Vous pouvez ainsi connecter des moteurs, lampes, petits aiguillages, relais entre le + d'une alimentation et une des sorties de l'ULN. Ne connectez pas une charge qui consomme plus de 500mA. Utilisez un relais ou un transistor plus puissant dans ce cas. L'ULN ne peut pas commuter des tensions alternatives car les transistors fonctionnent que dans un seul sens, utilisez un relais dans ce cas. Enfin pensez à mettre une diode de roue libre DRL aux bornes de chaque charge inductive (qui contient une bobine comme un relais ou un moteur ou un aiguillage). Une DRL est une diode montée en inverse qui va éliminer la surtension énorme qui se crée lorsque l'on coupe l'alimentation d'une charge inductive. L'anode de la diode (+) est branché sur la sortie et la cathode (-) sur l'alimentation.

E. Les drivers de moteur L293D

Le L293D est un circuit qui permet de faire tourner ou pas 2 moteurs dans un sens ou dans l'autre. Il peut contrôler des moteurs jusqu'à 700mA. Pour des moteurs plus puissants il faudra utiliser des L298 mais le brochage n'est vraiment pas le même. 700mA devraient suffire pour une locomotive en HO ou 2 en N sur chaque canton. Il est indispensable de se procurer la version D car elle contient les diodes de roue libre. La figure suivante détaille ce composant.



Les amplificateurs 1 à 4 mettent Vs ou la masse sur les sorties OUT1 à OUT4 suivant les niveaux présents sur les broches IN1 à IN2. Enable 1 et 2 permettent d'activer respectivement les amplificateur 1,2 et 3,4. La partie logique du circuit s'alimente par Vss que l'on mettra à 5V pour être compatible avec les autres CI de la carte. Toutes les entrées (IN1 à 4 et enable 1 et 2) doivent donc avoir des tension entre 0 et 5V. La partie puissance est alimentée par Vs (patte 8). C'est sur cette patte qu'il faudra mettre le + de l'alimentation des voies. On activera toujours les sorties, donc enable1=enable2=5V. On branchera une voie entre les sorties 1 et 2 et une autre entre 3 et 4. On peut donc alimenter 2 cantons avec un composant. Ainsi une locomotive branchée entre les sorties 1 et 2 aura le comportement suivant :

- Locomotive arrêtée si IN1=IN2
- Locomotive dans un sens si IN1=0V et IN2=5V (OUT1=0V et OUT2=+V_ALIM_VOIES)
- Locomotive dans le sens inverse si IN1=5V et IN2=0V (OUT1=+V_ALIM_VOIES, OUT2=0V)

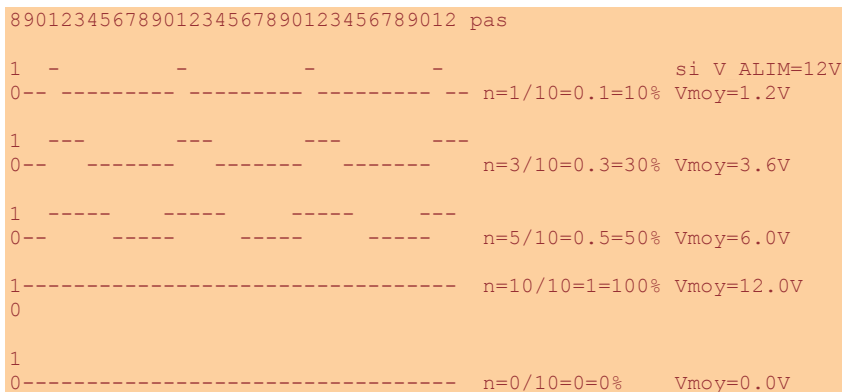
Bon très bien, on sait arrêter un moteur, le faire tourner dans un sens ou un autre, mais où est la gestion de la vitesse ? Et bien c'est le UC qui va produire une commande spéciale en utilisant la méthode de la PWM décrite au chapitre suivant. Pour l'arrêt, les 2 IN seront à la masse. Pour un sens, une des entrées sera à la masse (suivant le sens choisit tandis que l'autre recevra cette fameuse PWM.

Enfin pour terminer la description de ce double pont en H, il faut signaler que les 4 broches centrales permettent d'évacuer la chaleur du composant, il faudra donc veiller à faire un gros pâté de soudure sous ces pattes. Ce composant risque de chauffer un peu, c'est normal. Il ne risque pas de partir en fumée car il est protégé contre la surchauffe (Les sorties sont alors désactivées le temps qu'il refroidisse). Ce composant est également protégé contre les court-circuit ce qui tombe bien car il y en aura forcément lorsqu'une locomotive change de canton.

La PWM

PWM pour Pulse Width Modulation signifie modulation de largeur d'impulsion (MLI en Français). Au lieu de faire varier la tension du moteur pour faire varier la vitesse, la PWM alimente le moteur avec la tension maximale pendant un temps plus ou

moins long tandis qu'on ne l'alimente plus pendant le reste du temps et ceci périodiquement. On comprend bien qu'en alimentant un moteur avec 12V pendant 1s et en ne l'alimentant plus pendant 1s puis en le ré alimentant pendant 1s ... le moteur tournera moins vite que s'il était alimenté en permanence. Le signal du petit exemple précédent se répète toute les 2s, on parle de période T. La fréquence est le nombre de fois que la période se produit en 1s $F=1/T=0.5\text{Hz}$ dans ce cas. L'impulsion dure 1s sur une période de 2s, on note ce rapport : rapport cyclique N : $N=Ton/T=0.5=50\%$ dans l'exemple. La tension moyenne vaut : $V_{moy}=N*V_ALIM_VOIES$. La figure suivante montre différents rapports cycliques :



La fréquence est importante, car trop lente le moteur aura des a coups. J'ai choisit une fréquence de 50Hz, c'est à dire que 50 fois par seconde le UC envoie une impulsion qui dure entre 0 et 20ms suivant la vitesse souhaitée. Cette fréquence est suffisante pour ne pas avoir d'a coups. Le 50Hz s'entend dans les bobines des moteurs mais ce n'est pas bien gênant car le bruit de roulement le couvre et ca donne même plus de réalisme aux locomotives électriques. De toute façon je ne pouvais pas hacher plus rapidement pour ne pas dépasser la puissance de calcul du UC (N'oublions pas qu'il doit fabriquer entre autre 16 signaux de PWM après des registres à décalage !). Pour éliminer tout bruit il faudrait hacher vers 20000Hz, ce qui serait impossible pour nous et pose un autre problème : A chaque commutation, les transistors de puissances chauffent beaucoup et donc plus ils commutent souvent et plus ils chauffent. Donc le UC fabrique de la PWM qu'il envoie sur une des 2 pattes d'entrée (suivant le sens souhaité) de chaque voie du L293D (l'autre patte étant à la masse). La PWM se retrouve ensuite sur la voie et alimente la locomotive qui s'y trouve. Voilà on sait piloter chaque canton en sens et vitesse. Une autre donnée importante en PWM numérique est le nombre de pas par période. Sur la figure il y en a 10 (0 à 9). Le programme du UC utilise 64 pas ce qui est très précis. Le UC fait donc un calcul pour chaque canton $50*64=3200$ fois par seconde.

Sur le montage, le pas 0 est spécial, en effet, il est utilisé pour détecter les locomotives sur tous les cantons avec un seul capteur de courant ! Le capteur de courant est branché entre l'alimentation des voies et la broche puissance Vs des L293D. Ainsi s'il y a du courant consommé sur la partie puissance d'un des L293D (par une locomotive) alors le UC en est informé. Comme j'utilise un seul capteur de courant pour chaque canton, le UC active à chaque période tour à tour un seul des 16 cantons pour voir s'il y a une locomotive. Ceci peut se voir comme de la PWM dans la PWM ! D'ailleurs si vous approchez l'oreille d'une locomotive avec une consigne de vitesse à 0, vous entendrez un léger bruit d'horloge (tic.tic) qui provient de cette impulsion de contrôle. Rassurez vous, elle est bien incapable de faire bouger une locomotive car elle provient sur chaque canton uniquement pendant le pas 0 toute les 16 période ($1/(64pas*16periode)=0.001=0.1\%$ de puissance ou encore donne une tension moyenne de 0.012V si $V_ALIM_VOIE=12V$). L'utilisation d'un seul détecteur simplifie grandement le montage mais rallonge un petit peu le temps de détection qui est au pire de $16*20ms=320ms$, mais en pratique ca ne gêne pas. Pour la petite histoire, je pensais utiliser uniquement des ILS pour détecter les trains, donc sans capteur de courant, mais je me suis vite rendu compte (uns fois la carte terminée que ce n'était pas viable, et qu'il fallait impérativement pouvoir détecter les locomotives sur les cantons (comme en réalité d'ailleurs), c'est pourquoi j'ai du trouver cette astuce car il était impensable de rajouter 16 capteurs de courant.

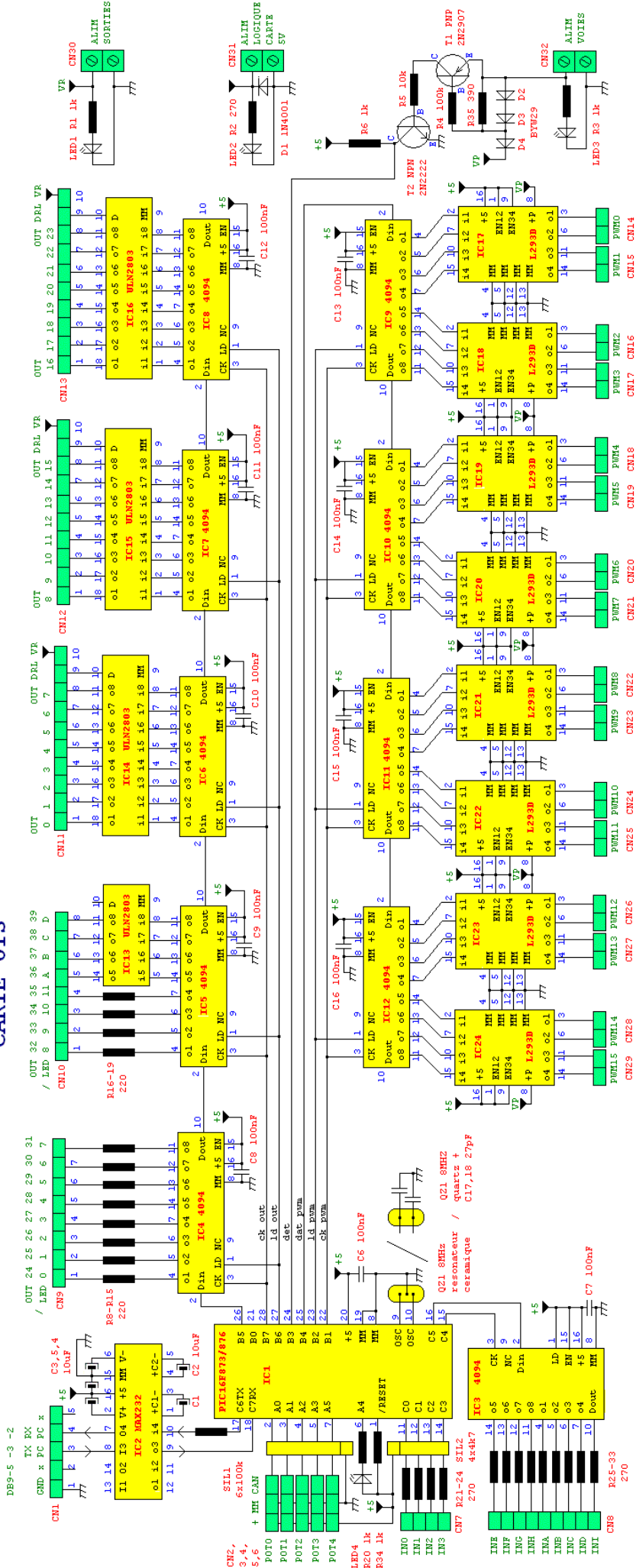
En plus de sa simplicité d'utilisation, la PWM offre de précieux avantages pour les modélistes par rapport à la traditionnelle tension variable :

- Puisque les moteurs sont alimentés avec la tension maximale (V_ALIM_VOIE), alors le couple est toujours important, et ce même à petite vitesse. Il est ainsi possible de manœuvrer à très petite vitesse (1mm/s) ce qui est fort apprécié et réaliste.
- Les voies se salissent beaucoup moins et les problèmes de captage sont quasiment inexistant! Avec la tension variable donc continue, je devais nettoyer mes voies plusieurs fois par jours et les problèmes de captages étaient nombreux, avec la PWM un petit coup chaque semaine pour presque plus aucun problème ! Bon pour l'instant ca à l'air magique mais que se passe t'il vraiment ? Et bien 2 choses salissent les voies : La poussière (et nous n'y pouvons rien) et les étincelles (peu de gens le savent). Une étincelle se produit lorsque l'alimentation du moteur vient juste d'être coupée (par exemple la locomotive passe sur un aiguillage ou sur une poussière ...), à ce moment là, la bobine produit une forte tension pour essayer de continuer à faire passer le courant ce qui crée un arc électrique : c'est à dire l'étincelle. Avec la PWM, comme le courant est coupé régulièrement, l'arc n'a pas le temps de se former (ou du moins est très court).

3.5. Schéma électrique de la carte

Maintenant que vous connaissez les CI (comme un expert), il est temps de dévoiler et commenter le schéma électrique de la carte. Ne partez pas en courant, il n'a rien de compliqué, ce sont juste plusieurs CI décrits qui sont mis bout à bout.

CARTE UTS



Les 24 CI :

On retrouve donc le UC (IC1) qui peut dialoguer avec le PC avec la liaison RS232 qui est mise à niveau par le MAX232 (IC2). Le UC pilote 10 registres à décalages (IC3 à IC12) afin d'augmenter ses sorties. Les signaux des registres à décalage sont amplifiés soit par les drivers de moteurs L293D (IC17 à IC24) ou par les darlington ULN2803 (IC13 à IC16).

Les alimentations :

Les alimentations se connectent sur les borniers à vis CN30 à CN32. Notez que les masses sont reliées entre elles. Les alimentations suivantes sont disponibles :

- **L'alimentation des sorties VR** : Pour les sorties : Cette alimentation n'est pas utilisée par la carte, elle est juste redirigée vers les connecteurs de sorties. Ainsi les charges de sorties peuvent se brancher entre VR et une des sorties d'un ULN qui rappelle le fait la liaison à la masse si l'entrée correspondante est active. La charge se trouve ainsi alimentée par VR. Si une autre alimentation est nécessaire, penser à relier sa masse à la masse de la carte. Si les charges sont inductives, pensez à mettre des DRL. Notez que les alimentations doivent impérativement être continues.
- **L'alimentation logique +5V** : C'est cette source qui est en charge d'alimenter tous les circuits logiques de la carte. Elle doit donc être régulée. Un condensateur de 100nF est chargé de découpler chaque circuit logique en nettoyant la tension des parasites haute fréquence qui se promènent sur les pistes d'alimentations et qui sont générés par les circuits. Les circuits de puissance n'en ont pas besoin. Le MAX 232 a besoin d'un plus gros condensateur car sa pompe de charge qui transforme du 5V en 9V génère beaucoup de parasites. Enfin une diode branchée à l'envers est sensée lutter contre les parasites en tension inverse. Inutile de préciser que brancher cette alimentation à l'envers détruit irrémédiablement tous les circuits de la carte. Il est préférable que l'alimentation logique soit toujours allumée lorsque la tension voie est allumée.
- **L'alimentation des voies** : Cette source sert à alimenter les cantons et donc les locomotives. Cette tension passe à travers le détecteur de courant et les drivers de moteurs, donc comptez une perte de 2.5V environ.

Le détecteur de courant :

Le détecteur de courant informe le UC, lorsqu'il y a une consommation de courant sur les cantons alimentés par les drivers et donc de la présence des locomotives. La double PWM permet en utilisant un seul détecteur de courant de savoir si une locomotive est présente sur chaque canton. Le fonctionnement du détecteur est le suivant :

- Le courant de traction (s'il y en a un) traverse les 3 diodes de puissance D2, D3 et D4.
- D4 empêche que les moteurs ne perturbent le détecteur en remontant du courant.
- Lorsqu'il n'y a aucun courant il n'y a aucune tension aux bornes de D2 et D3, T1 est bloqué, T2 aussi et le UC voit 5V par l'intermédiaire de la résistance de pull-up R6.
- Pour les non-électroniciens et pour faire simple un transistor est bloqué lorsqu'il y a moins de 0.6V entre sa base et son émetteur. Lorsque cette tension est atteinte alors le transistor est passant et le collecteur et l'émetteur sont reliés. Il existe 2 sortes de transistors : Les NPN ou le courant sort par l'émetteur et les PNP ou le courant rentre.
- Lorsque les cantons consomment alors un courant circule à travers D2 et D3 et il apparaît une tension de 1.2V aux bornes de ces 2 diodes (car une diode dans le sens passant fait perdre 0.6V). Cette tension est suffisante pour faire conduire le transistor T1 car la tension base-émetteur est supérieure à 0.6V. La tension des voies se retrouve donc sur la base de T2, ce qui fait une tension base-émetteur supérieure à 0.6V et qui le fait conduire. En conduisant le collecteur est relié à l'émetteur de T2 et donc la résistance R6 est court-circuitée à la masse et le UC voit 0V.
- Pour résumer lorsqu'un courant voie est consommé, alors le UC voit 0V (état 0) sinon il voit 5V (état 1). Cette tension d'information ne dépend pas de la tension voie.
- Les résistances R4 et R5 protègent les bases car la tension maximale entre la base et l'émetteur est de 0.6V. Le reste va dans les résistances. Lorsque je parlais de tension supérieure à 0.6V je voulais dire entre la résistance de base et l'émetteur. Elles limitent également le gain en courant.
- La résistance R34 sert à produire une tension pour les faibles courants lorsque les diodes ne conduisent pas encore. Ceci permet d'améliorer la sensibilité du détecteur. La sensibilité est telle qu'il détecte vos doigts mouillés si vous les placez entre 2 rails. Donc aucun problème pour détecter le moteur d'une locomotive ou même ses phares.

D'après les mails de retour, le détecteur de courant est assez difficile à mettre au point car le gain des transistors n'est pas constant entre 2 modèles (même de même référence). S'il ne marche pas, vous pouvez essayer de modifier les valeurs des résistances R4 et R5. A ce titre il est judicieux de les monter sur des picots tulipes à récupérer sur des supports DIL tulipes. Si le détecteur ne détecte aucune présence alors diminuer les valeurs (sans passer sous 1KOhms). Si la détection est permanente, alors augmenter les valeurs (sans dépasser 470KOhms).

Voilà, il n'y a rien d'autre à ajouter que vous ne saviez sur le schéma de cette carte.

Modularité :

Bien entendu, vous pouvez vous passer de certains circuits suivant vos besoins.

- Si vous n'avez pas besoin des entrées (ou si 4 entrées sont suffisantes) alors IC3 est inutile.
- Si 12 cantons vous suffisent alors passez-vous de IC12, IC23 et IC24
- Si 8 cantons suffisent alors retirez aussi IC11, IC21 et IC22
- Si 4 cantons suffisent alors retirez aussi IC10, IC19 et IC20
- Si seulement 16 sorties suffisent alors retirer IC8 et IC16
- Si seulement 8 sorties suffisent alors retirer IC7 et IC15
- Si le contrôle des LEDs ne vous intéresse pas, vous pouvez les remplacer par des sorties.
- Si les sorties ne demandent pas d'amplificateurs, vous pouvez supprimer les ULN (mais vous aurez alors 10mA sous 5V au lieu de 500mA sur la tension que vous voulez (VR)).
- Si une carte ne suffit pas, alors il est possible d'en utiliser 4

Capacités en fonction du nombre de cartes :

Nombre de cartes:1234
Cantons16324864
Entrées numériques3672	...108	...144
Entrées analogiques5101520
Sorties/LEDS	.24/48	.48/96	72/144	96/192
	.40/00	.64/48	.88/96	112/144
		.80/00	104/48	128/96
			120/00	144/48
				160/00

Si le nombre de canton n'est pas suffisent, il est possible d'utiliser des garages. Un garage est simplement une voie qui se connecte ou déconnecte électriquement d'un canton grâce à un relais. UTS gère simplement les garages. Il est possible d'utiliser jusqu'à 16 garages. Un garage est connecté en fonctionnement normal. Il peut être déconnecté pour garer un train. Le canton associé peut donc être utilisé par un autre train. Il ne peut donc y avoir qu'un seul train actif sur un canton et ses garages. Les garages sont une notion du logiciel, ils utilisent juste une sortie par garage pour piloter le relais. Donc si le nombre ne suffit pas, il suffit juste de changer un peu le logiciel.

Le nombre élevé d'aiguillages sur certains réseaux ne pose pas de PB. Avec les informations présentées au chapitre des aiguillages, il est possible d'en commander 64 avec seulement 7 sorties. La limitation de 64 vient aussi du soft et peut être modifiée sans gros efforts.

Pour être complet, voici les capacités du soft (je peux les modifier sur demande)

Nombre de TCO	8
Nombre de trains	16
Nombre d'aiguillages	64
Nombre d'itinéraires	50
Nombre de déplacements	100

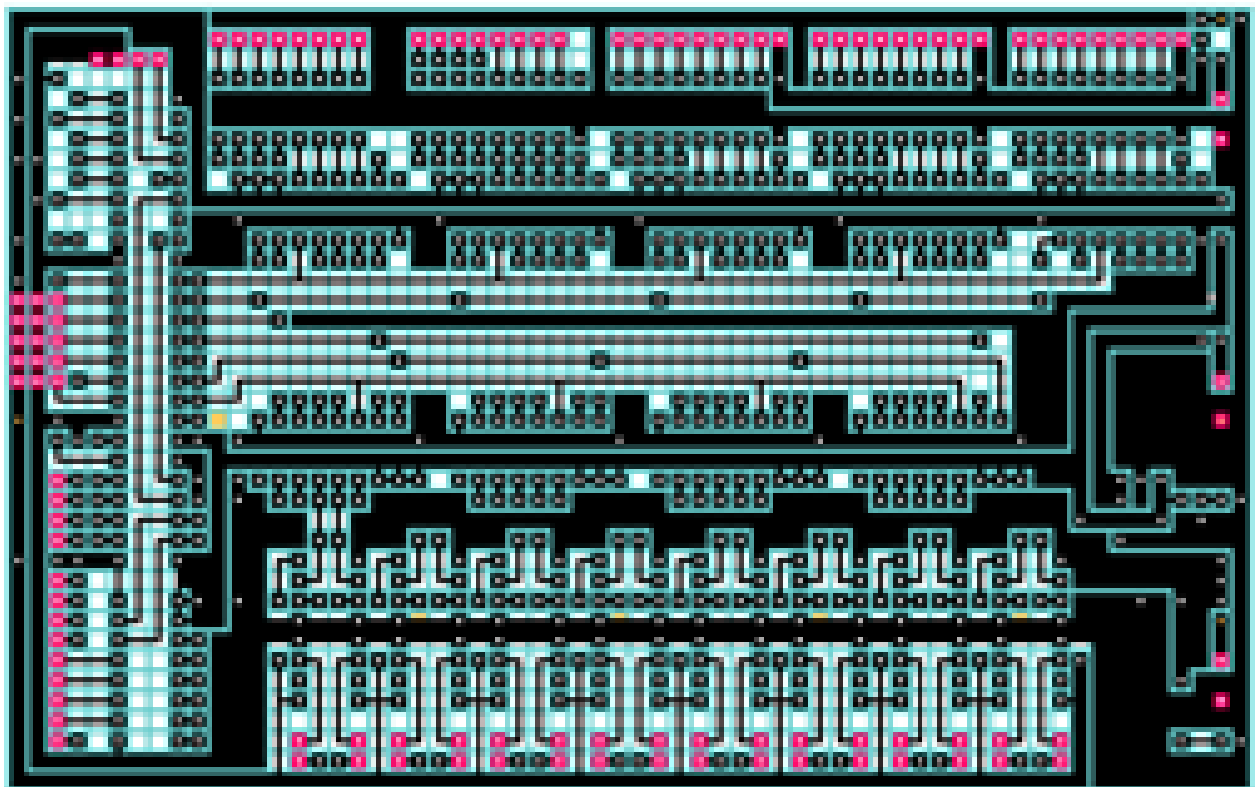
3.6. Réalisation et tests

Pour réaliser la carte, les 3 solutions suivantes peuvent être utilisées :

- Les plaques vero-boards sont des plaques percées avec des bandes ou des pastilles de cuivre ou l'on soude les composants. C'est la façon la plus directe mais il faudra sans doute utiliser beaucoup de fils.
- Il est possible de se graver un circuit imprimé avec la méthode photographique (insolable, graveuse) à partir de plaques époxy photosensible. Cette méthode nécessite du matériel mais donne de bons résultats.
- Enfin il est possible d'acheter une plaque époxy non pré sensibilisée et de graver le contour des pistes avec une mini perceuse équipée d'une fraise sphérique (Ces fraises sont utilisées pour graver le verre). La gravure manuelle est longue mais il n'y a qu'une plaque à réaliser.

Bref, si vous avez jamais fait de plaques le vero-board semble être la solution la plus facile, la perceuse donne un plus beau résultat car il n'y a pas de trou de partout et le soudage est plus rapide car il n'y a pas besoin de fils étant donné que les pistes sont gravées. Enfin la méthode photographique est la méthode des pros. Pour ma part, j'ai utilisé la méthode de la perceuse.

Afin de vous aidez, je vous fournis le typon suivant. Attention. Il n'est pas à l'échelle ; vous devrez donc le mettre en utilisant un zoom logiciel avant impression ou alors le zoom d'une photocopieuse. Pour vérifier la taille, essayer de positionner le plus gros circuit. C'est un typon simple face, c'est à dire que les pistes ne sont que sur une seule face. C'est beaucoup plus facile à réaliser que du double face. Revers de la médaille, il faudra rajouter des fils que l'on appelle des straps. Je vous laisse trouver les emplacements grâce au schéma électrique. L'emplacement des composants est identique au schéma électrique. Les IC9 à 12 et 17 à 18 devront être retournés (patte 1 au Nord Est pour les 4094 et au Sud Est pour les L293D. Aidez-vous du schéma électrique pour placer les composants. Le détecteur de courant n'est pas présent sur ce typon. La patte 15 du MAX232 n'est pas reliée à la masse. La figure est donnée coté composant. Il faut faire un miroir pour l'avoir du coté des pistes.



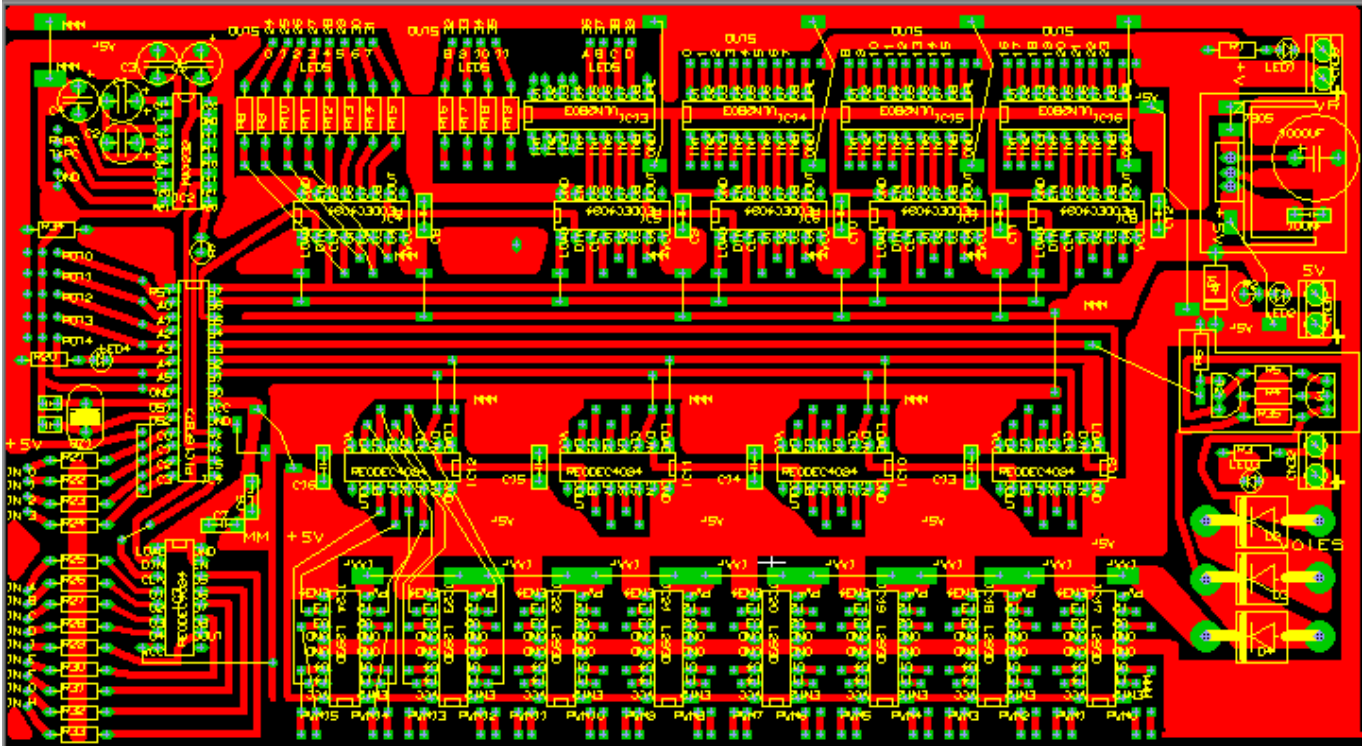
Plaque 160 * 100 mm (62*39 trous).

Le détecteur de courant n'est pas inclus sur cette carte.

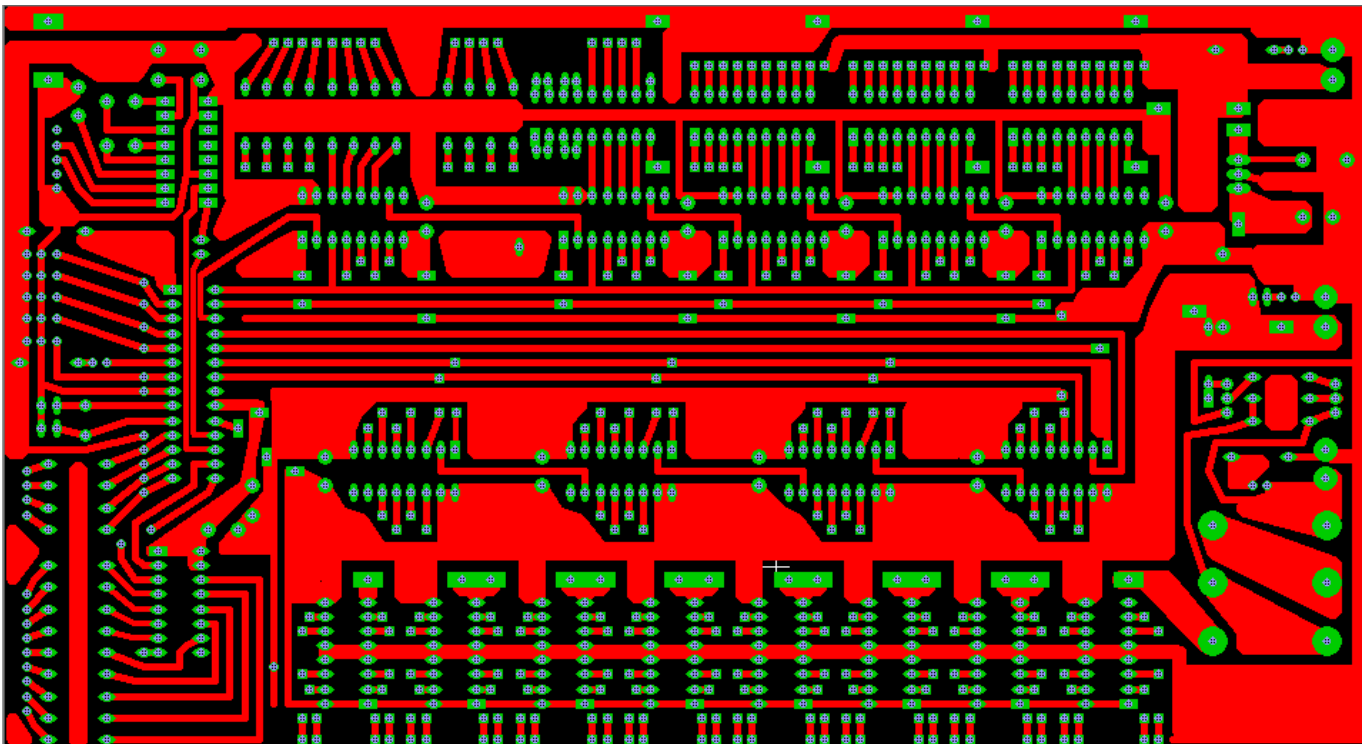
Vous pouvez voir cette carte sur la première page de ce manuel.

Laurent (un utilisateur) m'a donné un bon typon au format tci. Ce fichier est disponible dans l'archive. C'est l'excellent freeware TCI de Bruno Urbani (<http://b.urban.free.fr>) qui a servi pour le générer. Vous pouvez l'utiliser pour modifier le typon ou en faire un autre. Laurent, si tu me lis, j'ai juste rajouter des connecteurs aux sorties PWM. Attention, tout de même la sérigraphie inverse le +5V et la masse des 4094. Attention également, le typon n'est pas à l'échelle, il faudra donc utiliser un zoom (sur PC avant impression ou sur un photocopieur). Vérifier que le plus gros de CI rentre bien pour éviter les PB. Vous remarquerez que Laurent a rajouter un 7805 afin de fournir le 5V nécessaire à la carte à partir de l'alim des sorties. Il n'y a donc besoin que de 2 tensions pour alimenter la carte ou seulement une si les voies et sorties utilisent la même tension.

Avec la sérigraphie (coté composants)



Sans la sérigraphie (coté composants)



La nomenclature suivante liste les composants utiles.

IC1	1x	microcontrôleur PIC 16F873 (ou 16F876)
IC2	1x	MAX232
IC3-IC12	10x	registres a décalage 4094
IC13-IC16	4x	réseaux de transistors darlington ULN2803
IC17-IC24	8x	drivers de moteurs L293D (ou L293DNE) (Attention, il faut le D: L293 ou L293NE ne conviennent pas)
C1-C5	4x	condensateurs chimique 10uF 16V
C6-C16	11x	condensateurs plastiques 100nF 63V
T1	1x	transistor PNP 2N2907
T2	1x	transistor NPN 2N2222
R8-R19	12x	resistances 220 Ohms 5% 1/4W
R7,R21-R33 ...	14x	resistances 270 Ohms 5% 1/4W
R35	1x	resistance 390 Ohms 5% 1/4W
R2	1x	resistance 470 Ohms 5% 1/4W
R1,R3,R6,R20,R34 ..	5x	resistances 1k Ohms 5% 1/4W
R5	1x	resistance 10K Ohms 5% 1/4W
R4	1x	resistance 100K Ohms 5% 1/4W
SIL1	1x	réseau SIL de 6 résistances de 100K Ohms (Peut être remplacé par des R 100KOhms)
SIL2	1x	réseau SIL de 4 résistances de 4.7K Ohms (Peut être remplacé par des R 4.7KOhms)
QZ1	1x	oscillateur céramique 8MHz (ressemble a une bulle bleu a 3 pattes) Peut être remplacé par un quartz avec + 2 condensateurs 27pF
LED1-3	3x	LEDs 3mm vertes
LED4	1x	LED 3mm jaune
D1	1x	diode 1N4001
D2-D4	3x	diodes de puissance BYW29 (ou équivalentes qui supportent 5A)
CN30-CN32	3x	borniers a vis de 2 elements
CN1-CN29	29x	a fabriquer a partir de barette autosecable male (il faut 108 points)
Support DIL28	1x	support tulipe DIL28 étroit (pour 16f873)
Support DIL16	19x	supports tulipe DIL16 pour MAX232, 4094, L293D
Support DIL18	4x	supports tulipe DIL18 pour ULN2803

Pour les composants, il existe des magasins d'électronique dans les grandes villes, sinon il vous faudra soit vous déplacer soit les commander à distance. Pour vous aidez, voici la liste des magasins que je connais :

- Paris (Nation): ECE (Espace Composant Electronique)
- Paris (place de la Nation) : Selectronic (Marche aussi en correspondance).
- Nice (rue Tonduti de l'Escarène) : composant diffusion
- Toulouse (rue d'Alsace Lorraine ou zone industrielle de Montaudran) : Comptoir du languedoc

Je vous conseille de suivre les points suivants :

- Approvisionnement des composants
- Trouvez la place des composants et des straps sur le typon
- Soudez les alimentations le UC (La led jaune L4 doit clignoter)
- Soudez le MAX232 et faire un câble série pour relier la carte au PC
- (A ce niveau, vous pouvez utiliser le soft UTS qui doit voir la carte. Le logiciel dispose d'une fenêtre de test de la carte. Vous pouvez déjà tester les potentiomètres et les 4 entrées)
- Souder Le 4094 des entrées (testez les 36 entrées multiplexées avec le logiciel)
- Souder les 2 4094 et l'ULN 2803 des leds (testez les 48 leds multiplexées avec le logiciel)
- Souder les 3 4094 et les 3 ULN2803 des sorties (testez les 24 sorties avec le logiciel)
- Soudez les 4094 des voies et les L293D sans utiliser le détecteur (relier +VVI à l'alimentation des voies) (testez la PWM avec le logiciel)
- Rajoutez le détecteur (testez avec le logiciel)
- (Voilà la carte est opérationnelle)

3.7. Connexions

Cette section montre comment connecter votre carte.

Les alimentations :

Il suffit simplement de les relier aux borniers à vis CN30 à CN32 en prenant garde de les brancher dans le bon sens ! Les LEDs vertes LED1 à LED3 doivent s'allumer et la LED jaune du UC doit clignoter.

Le PC :

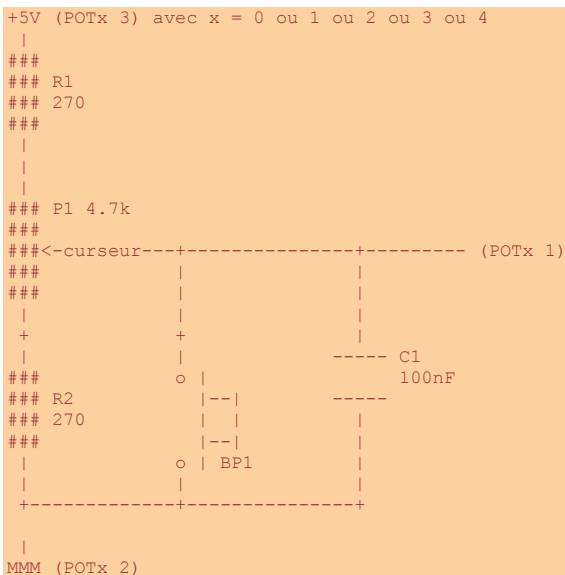
Il suffit de fabriquer un câble avec les informations cachées dans ce documents (Elles y sont ...)

Les cantons :

Il suffit simplement de relier chaque canton par 2 fils (un sur chaque rail) aux connecteurs CN14 à CN29. Si le sens de la locomotive sur ce canton ne vous convient pas alors branchez à l'envers.

Les potentiomètres :

La carte permet la mesure de 5 tensions. Il est possible de raccorder tout type de capteur analogique comme des LDR (pour mesurer la lumière) ... Mais le plus naturel en modélisme est bien entendu d'y connecter des potentiomètres afin de conduire les locomotives avec plaisir car il n'est pas très agréable de conduire un train au clavier ou à la souris. Je conseille le montage suivant :

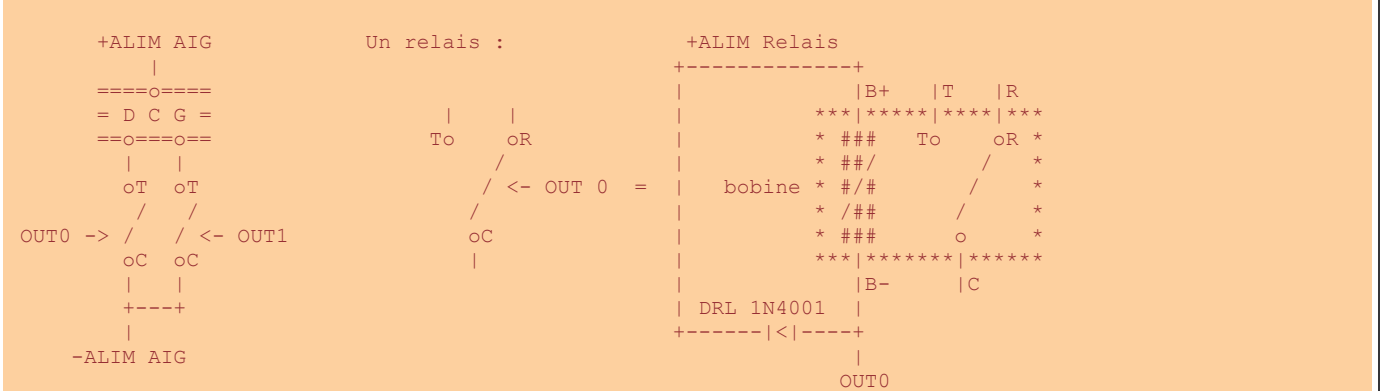


Le cœur du montage est bien entendu le potentiomètre. Vous pouvez en prendre de rotatifs ou des droits. Mais choisissez un modèle linéaire et non logarithmique pour avoir une résistance qui varie linéairement avec la position du curseur. Souvent un linéaire se reconnaît par un A à côté de sa valeur, alors qu'un logarithmique présente la lettre B. R1 évite de cramer le potentiomètre lorsque la résistance entre le curseur et la patte connectée à R1 devient trop faible et qu'un courant circule à travers le curseur. Ce qui veut dire que la tension mesurée n'atteindra jamais 5V ce qui n'est pas gênant. R2 a une tout autre mission, Lorsque le bouton poussoir BP1 n'est pas pressé, alors R2 empêche le potentiomètre de présenter la masse au UC et donc la seule façon de voir la masse au niveau du UC est de presser le bouton poussoir. En effet le 5V est appliqué à la résistance totale $R1+P+R2=270+4700+270=5240\text{Ohms}$, donc il apparaît aux bornes de R2 : $(270/5240)*5=0.25\text{V}$ et lorsque le curseur est en bas sur le schéma, alors le UC voit 0.25V. Pour info, lorsqu'il est en haut, la tension mesurée est de $5-0.25=4.75\text{V}$. Bon très bien, mais à quoi sert ce bouton ? ... Des suggestions ? Et bien, il peut être utilisé pour freiner. En effet le logiciel peut simuler l'inertie d'un train en accélération, décélération et freinage. Et enfin le condensateur C1 garde la tension, lorsque le curseur quitte un court instant sa piste. Ceci évite d'avoir des mauvaises mesures lorsque l'on tourne le potentiomètre trop vite. S'il y a quand même des problèmes, alors il faudra augmenter le condensateur. Voilà, il ne vous reste plus qu'à fabriquer une jolie boîte pour chaque potentiomètre et de choisir un beau capuchon à fixer sur l'axe de 6mm. Bien entendu, si votre réseau est grand, vous pouvez mettre des prises pour les potentiomètres à des points stratégiques, plutôt que d'utiliser un très long câble. Lorsqu'un potentiomètre est débranché, le UC voit 0V par l'intermédiaire des résistances de pull_down du SIL 1. Pour voir si vos potentiomètres fonctionnent, il suffit de regarder les barre graphes dans le logiciel.

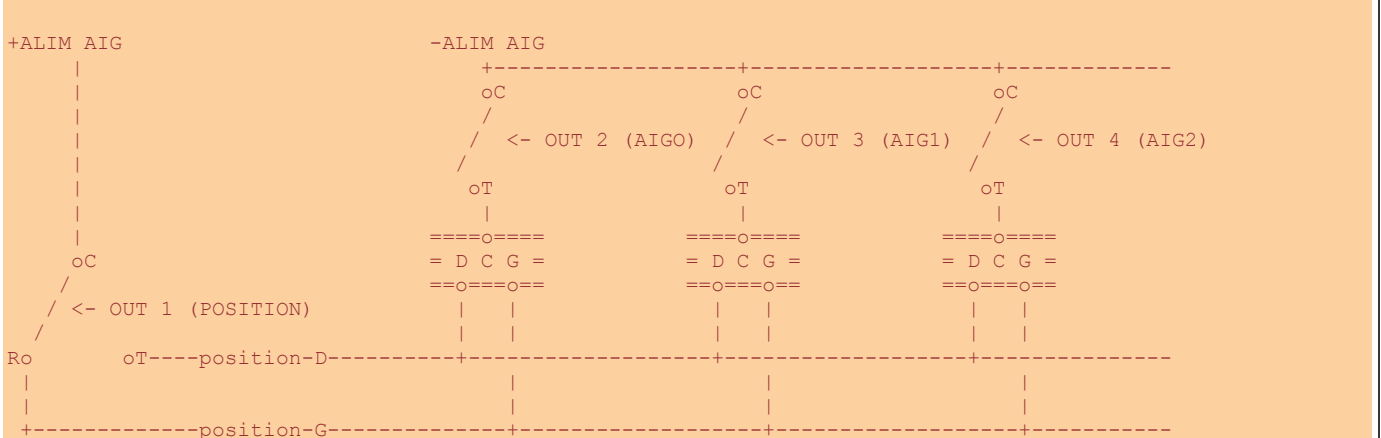
Les entrées numériques :

Afin de pouvoir utiliser un grand nombre de capteur (36) en minimisant les broches du UC (4 en entrées + 2 en sorties pour piloter le 4094 qui dispose de 9 sorties), les entrées sont multiplexées. Le multiplexage ou matricage, utilise une matrice, donc des lignes et des colonnes. Un capteur peut être mis à chaque intersection entre une ligne et une colonne. Chaque ligne est mise

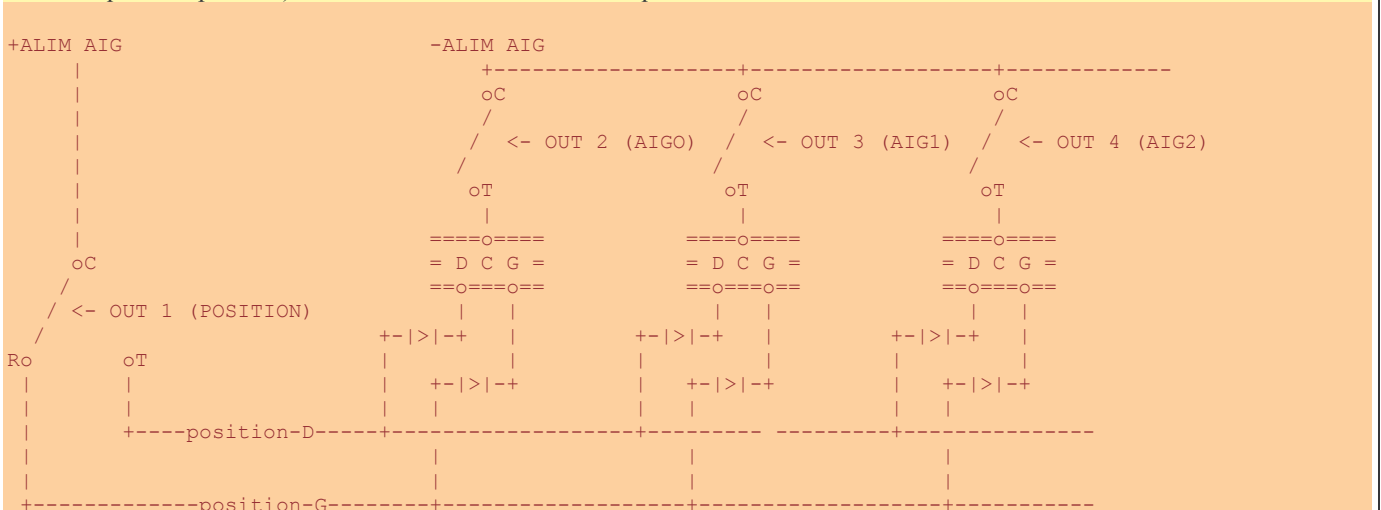
La plupart des aiguillages (sauf certains en N) consomment plus de 500mA et donc il est bon, d'utiliser des relais afin de ne pas détruire nos pauvres ULN2803. Un relais se compose d'une bobine qui lorsqu'elle est alimentée manœuvre un interrupteur par un champ magnétique. Sur les schémas, je ne représenterai que ces contacts que l'on appelle des RT. Au repos lorsque la bobine n'est pas alimentée, il y a contacte entre les pattes R(repos) et C(commun) des interrupteurs. Au travail lorsque la bobine est alimentée, il y a contacte entre les pattes T(travail) et C(commun). Les bobines sont donc reliées aux sorties. La tension peut être continue ou alternative.



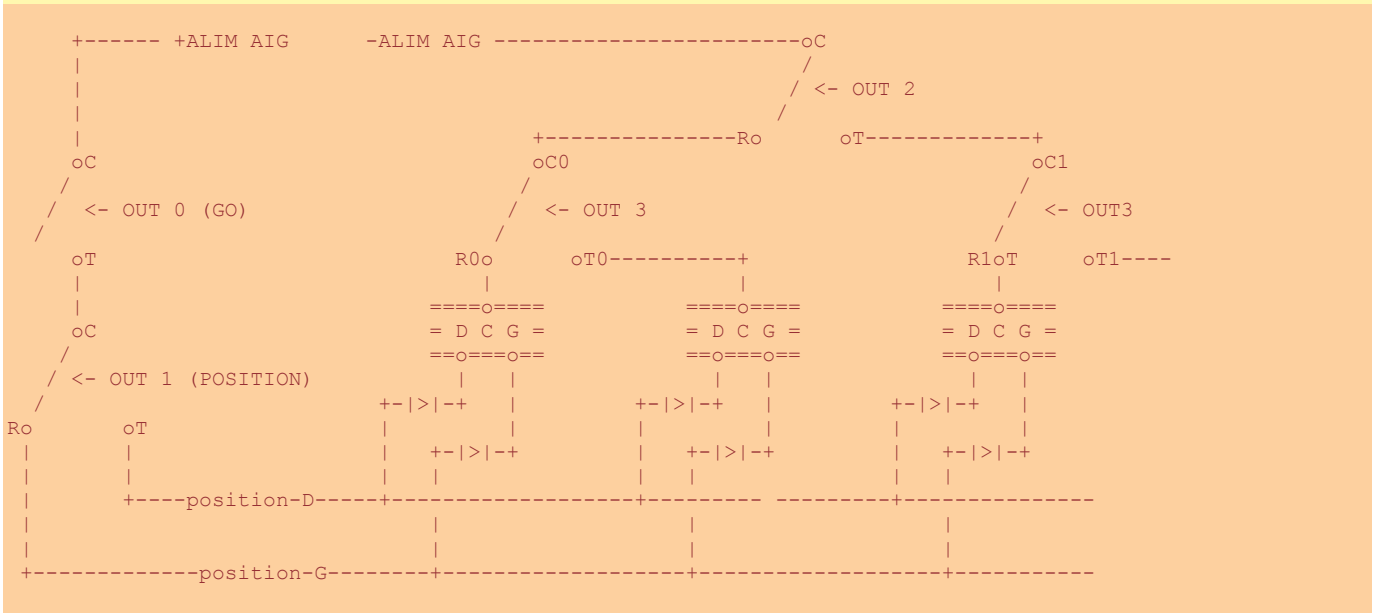
Très bien, mais il faut tout de même 2 relais et 2 sorties par aiguillage, ce qui est un peu beaucoup ! Quelle méthode peut on utiliser pour réduire ces nombre ? Et oui bravo ! Le multiplexage, encore lui ! On se servira d'un relais par aiguillage pour sélectionner le dit aiguillage et d'un autre commun à tous les aiguillages pour choisir la position).



L'amélioration est flagrante. Mais on ne peut commander qu'un aiguillage à la fois car si on en commande plusieurs en même temps alors il se mettront tous dans la même position. En fait ceci n'est pas un problème car le logiciel supporte ce fonctionnement séquentiel. Et cela offre même un avantage non négligeable car comme on ne pilote qu'un aiguillage à la fois, alors l'alimentation des aiguillages n'a pas besoin d'être très puissante. Il suffit juste qu'elle sache faire bouger un aiguillage. Mais le schéma tel qu'il est présente un gros problème qui fait que les aiguillages font n'importe quoi ! En effet regardez ce qu'il se passe lorsque l'on commande l'aiguillage 0 à droite. La bobine D0 est bien alimentée, mais également la G0 par l'intermédiaire des bobines des autres aiguillages ! Et oui comme la ligne position D est alimenté, un courant rentre dans D1 (D aiguillage1) et ressort par G1, ce qui alimente la ligne position G et donc la bobine G0 ! Pour résoudre ce problème, nous allons rajouter une diode 1N4001 (dans le bon sens) qui autorisera le courant à seulement rentrer dans les bobines (Il sortira alors uniquement par le C). Vérifiez sur le schéma suivant que cela fonctionne.

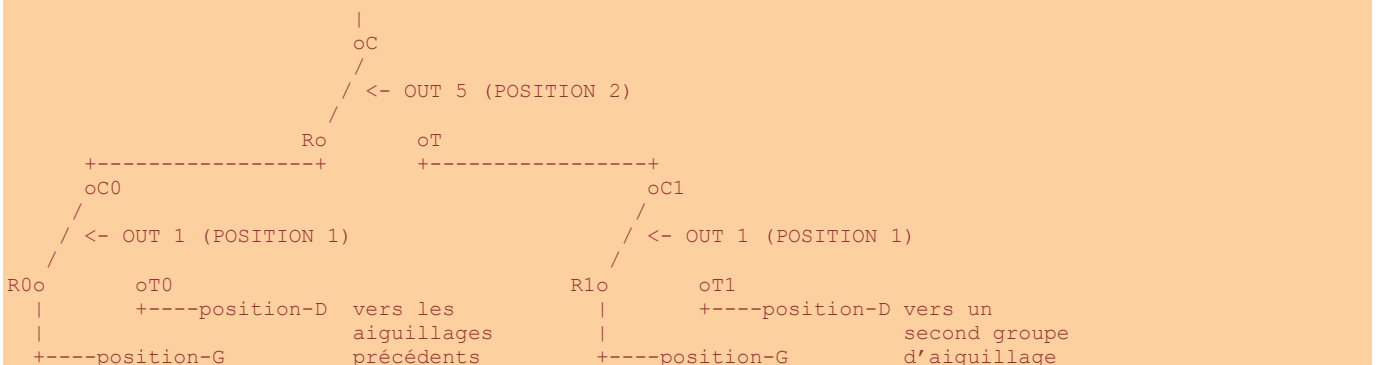


Bien entendu, lorsque l'on utilise les diodes, il n'est plus question d'utiliser de l'alternatif. Seule l'alimentation continue convient. On peut encore aller plus loin dans le multiplexage en utilisant des relais à plusieurs interrupteurs (plusieurs RT). Dans ce cas, il faut rajouter un relais pour produire l'impulsion de manœuvre.



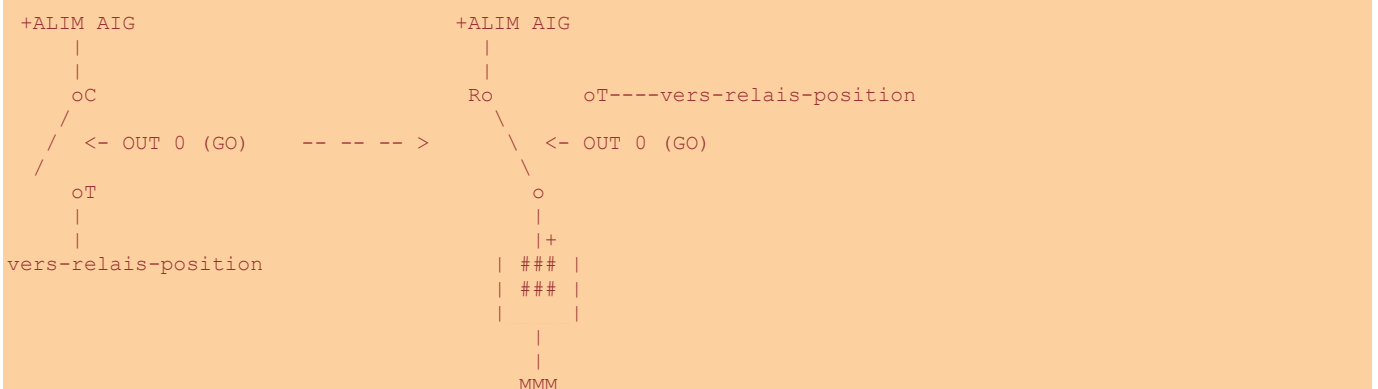
Ca commence à être bien, en effet avec 4 relais (dont un 2RT : celui commandé par OUT3, on arrive à commander 4 aiguillages). Avec 5 relais (dont un 2RT et un 4RT), on peut en piloter 8 ... Bien entendu, si on ne trouve pas de 4RT, on peut utiliser 2 relais 2RT que l'on peut même brancher sur la même sortie.

Il est encore possible de faire mieux, en multiplexant au niveau de la position, en effet, si on rajoute un relais 2RT à ce niveau, alors on double le nombre d'aiguillage. Le second groupe d'aiguillage à bien entendu, les communs branchés aux mêmes communs que le 1^{er} groupe.



Il est maintenant possible avec 6 relais (3x 1RT + 2x 2RT + 1x 4RT) de piloter 16 aiguillages. Je vous laisse trouver la variante à 6 relais pour 12 aiguillages (3x 1RT + 3x 2RT).

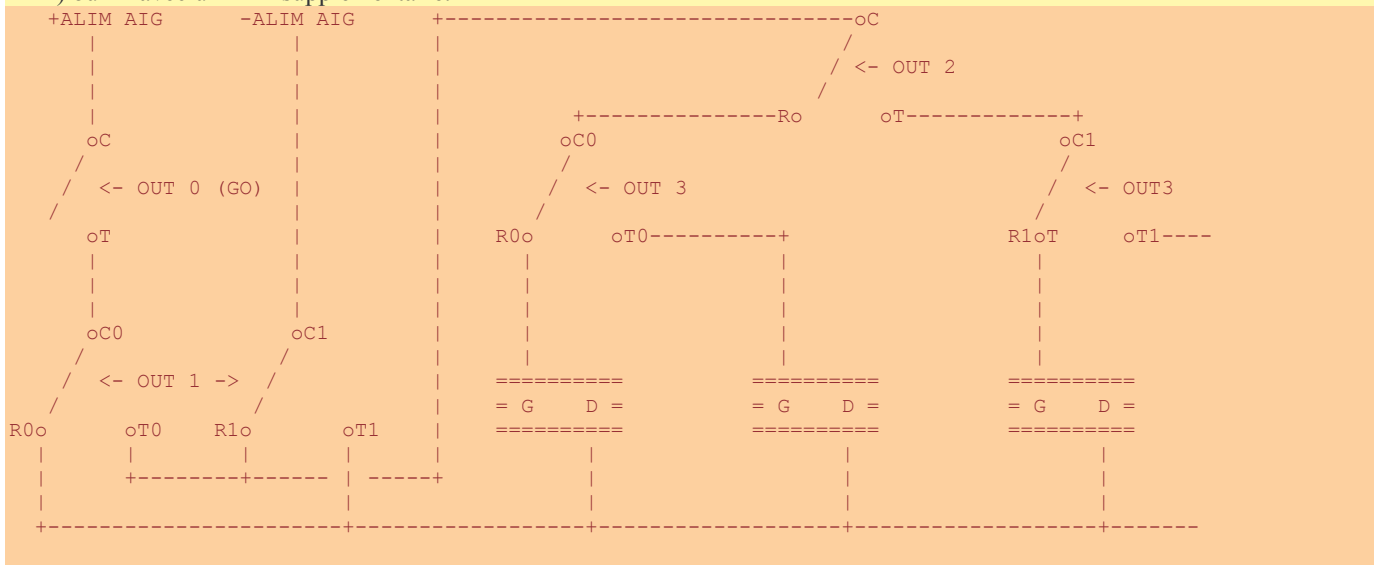
Bon, on est presque au must, si ce n'est que pour une raison ou une autre, quelque chose plante ou la liaison série est débranchée alors les relais gardent leurs positions, et s'ils n'ont pas d'interrupteurs intégrés qui coupent l'alimentation des bobines (comme pour les aiguillages Roco en N), alors une bobine risque d'être alimentée et donc de cramer. Pour cette raison, je conseille d'utiliser le système capacitif, qui consiste à charger un condensateur, puis à le décharger à travers la bobine sélectionnée. Cela garantit la vie de vos aiguillages. Il convient juste de remplacer le relais d'impulsion par le montage suivant :



Au repos le condensateur se charge, et il se décharge dans la bobine sélectionnée lorsque le relais est activé. Il faudra prendre un gros condensateur chimique. Attention au sens de branchement sinon c'est l'explosion et ça pue pendant des jours. Je conseille d'essayer plusieurs valeurs et lorsque l'aiguillage commence à marcher à tous les coups, doubler cette valeur par sécurité. Par exemple, j'utilise 2200uF pour mes aiguillages N Roco. Quelque fois, ça marche mieux en neutralisant les interrupteurs intégrés de coupure des bobines lorsqu'ils existent (C'était mon cas). Vous savez maintenant tout sur la commande des aiguillages à solénoïdes.

Je vais maintenant m'intéresser aux aiguillages à moteurs. Souvent ces moteurs sont placés sous la table et sont lents. 2 fils permettent de les mettre dans une position ou une autre, en les alimentant dans un sens ou dans l'autre. Généralement un interrupteur coupe le courant lorsque le moteur a atteint sa position terminale. Le présent montage est conçu pour fonctionner avec les moteurs qui ont cet interrupteur. On utilisera 1 relais (1RT) pour autoriser le déplacement, 1 autre (2RT) pour fixer le sens de rotation du moteur (en l'alimentant dans un sens ou dans un autre) et un groupe de relais pour sélectionner l'aiguillage. Il n'y aura pas besoin de rajouter des diodes puisque seulement 2 fils sont utilisés et on en coupe 1 avec le 3^{ème} groupe de relais lorsque l'aiguillage n'est pas sélectionné.

Avec 4 relais (2x 1RT + 2x 2RT) il est possible de commander 4 aiguillages. 8 si on rajoute un 4RT, 16 avec un 8RT (ou 2x 4RT) ou 12 avec un 4RT supplémentaire.



Coté logiciel, il permet de :

- spécifier un état de repos pour chaque sorties des aiguillages (qui est appliqué lorsqu'aucune opération n'est demandée).
- rentrer pour chaque aiguillage, l'état des sorties pour chacune des 6 phases suivantes (préparation à la manœuvre gauche, manœuvre gauche, libération de la manœuvre gauche, préparation à la manœuvre droite, manœuvre droite, libération de la manœuvre droite)
- indiquer la durée de chacune des phases.
- connaître la position des aiguillages sans interrupteurs de retours. Au démarrage, il met tous les aiguillages dans l'état de démarrage spécifié et donc une fois cette opération réalisé il connaît la position de chaque aiguillage. Il la met ensuite à jour, chaque fois qu'il modifie la position d'un aiguillage.
- bouger les aiguillages les uns après les autres.
- faire des itinéraires ...

Pour la réalisation de la carte à relais, je conseille d'utiliser le vero board.

3.7. Communication avec le PC

Cette section décrit le protocole de communication entre la carte et le PC. Ceci afin de satisfaire votre curiosité ou d'utiliser la carte avec un logiciel de votre cru ou de faire une carte qui se pilote par mon logiciel. Ou encore pour ajouter des fonctions à la carte ... La carte communique à 19200 bauds 8 bits sans parité. Les erreurs ne sont pas prises en compte car UTS les renvoie et les demande toute les 250ms. C'est un petit moniteur codé en assembleur dans le Pic qui permet de communiquer avec le PC. Il peut aussi servir à déboguer la carte ou la mettre dans des modes particuliers. UTS permet de dialoguer avec le moniteur et voire et modifier tous les registres du UC.

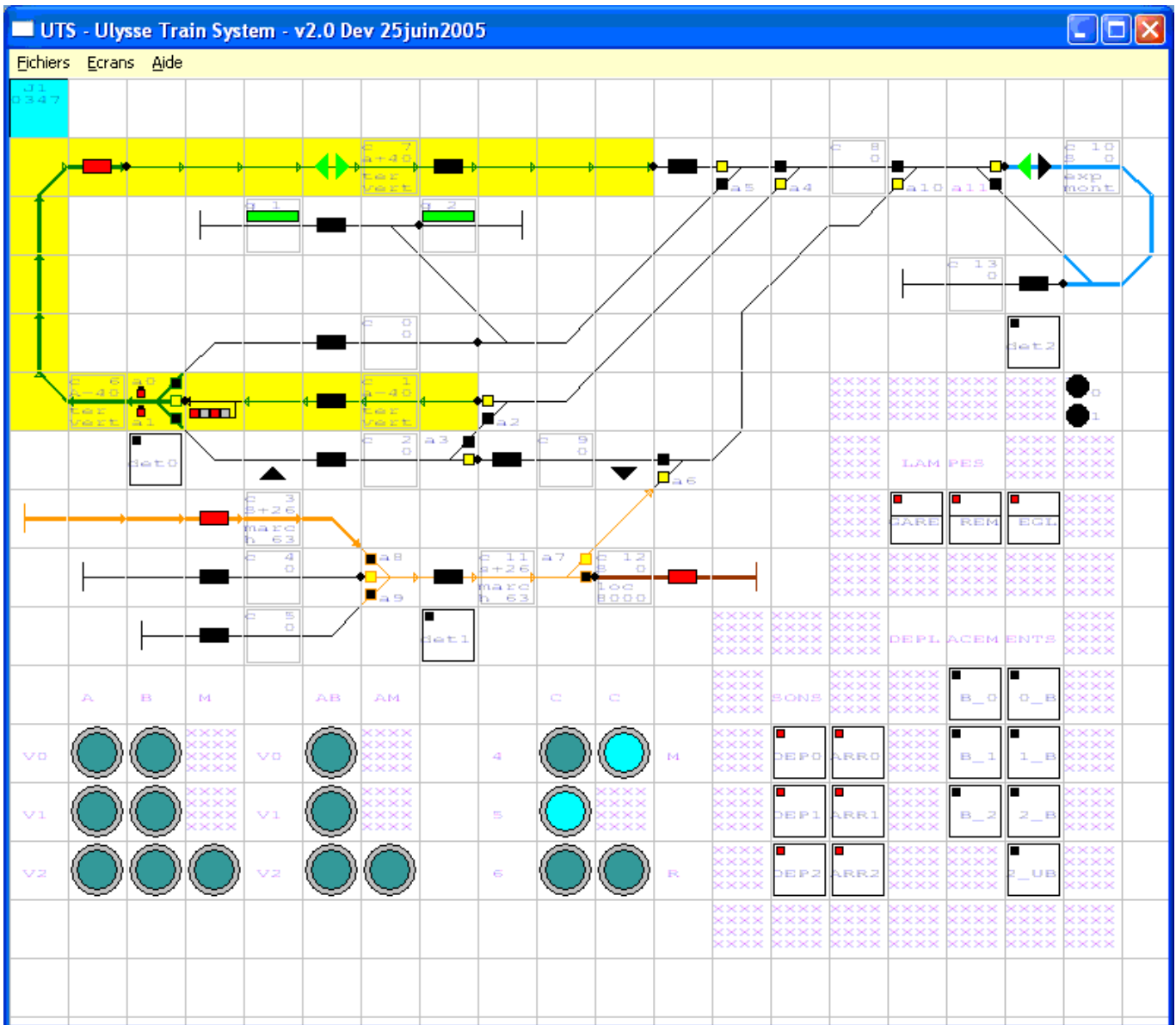
Les lignes suivantes décrivent le protocole utilisé :

- 0 : synchronisation : en envoyant la commande 0, le UC va se synchroniser, c'est utile lorsqu'on se perd dans le protocole. Un envoi de quelques 0 devrait permettre de récupérer toutes les erreurs de protocole. Il est utile d'en envoyer avant d'utiliser toute autre commande et régulièrement afin de venir corriger un éventuel problème.
- 1 : Test : en envoyant la commande 1, le UC devrait répondre 0x83. Cette commande peut servir à vérifier que le moniteur est bien actif et qu'il est possible de communiquer avec le UC.
- 2 : Adresse : en envoyant la commande 2 puis l'adresse du registre, on spécifie l'adresse du registre à utiliser pour les opérations de lecture et d'écriture. L'adresse étant sur 8 bits, il est possible d'accéder aux banques 0 et 1. Pour infos, Les registres les plus importants des PIC sont dans ces banques et MRC utilise uniquement la banque 0, donc ce n'est pas nécessaire d'accéder aux banques 2 et 3.
- 3 : Ecrire : en envoyant la commande 3 puis la donnée à écrire, cette donnée est mise dans le registre sélectionné.
- 4 : Lecture : en envoyant la commande 4, le moniteur renvoie la valeur courante du registre sélectionné.
- 5 : Ecriture incrémentale : La commande 5 suivit d'une donnée, permet de passer au registre suivant, en incrémentant l'adresse courante, puis d'y écrire la donnée.
- 6 : Lecture incrémentale : La commande 6 suivit permet de passer au registre suivant en incrémentant l'adresse courante, puis de renvoyer la valeur de ce registre.
- 7 : Lecture avec effacement: en envoyant la commande 7, le moniteur renvoie la valeur courante du registre sélectionné et le met à 0.
- 8 : Déclenche un transfert en lecture et écriture de tous les registres couramment utilisés par UTS. Ce mode fait gagné beaucoup de temps car il n'y a pratiquement que de la donnée qui circule.
- 8x (hex) : sélectionne une carte. (Les cartes avec l'ancienne version 873_rs11 flashée ne tiennent pas compte de cette commande. Le soft sait aussi reconnaître la version de la carte. Si une ancienne carte est utilisée, elle sera considérée comme la carte 0. Rappelons qu'il n'est pas possible d'utiliser des anciennes et nouvelles cartes en même temps. Soit une ancienne carte, soit de 1 à 4 nouvelles cartes.

Vous savez maintenant tout sur votre carte électronique !

J'espère avoir été assez clair ;-)

4. Logiciel



Copie d'écran d'un TCO de mon réseau. Vous pouvez voir 4 trains dont 1 en automatique (fond jaune) et un en manuel sélectif en orange. Les ronds sont les interrupteurs de commande d'itinéraires. Les boutons sur la droite permettent divers actions comme des lampes, sons et déplacement. Le fait de cliquer sur un aiguillage modifie sa position.

4.1. Présentation du logiciel

Le logiciel UTS permet de piloter la carte électronique (ou les cartes) et donc le réseau. Il fonctionne sous Windows (à partir de Windows 95 : 95, 98, Me, 2000, NT, XP ...).

Vous pouvez télécharger l'exécutable (uts0200.exe) ainsi que les sources si vous souhaitez le modifier ou voir comment il est codé (10000 lignes de code en C/C++). Il a été compilé avec GCC sous l'environnement Dev C++. Ces outils sont des logiciels GNU donc vous pouvez vous les procurer gratuitement. UTS est également livré aux particuliers dans l'esprit GNU, c'est à dire que vous en faite ce que vous voulez, tant que vous livrez les sources. Bien entendu, ayant fait ce logiciel gratuitement, je ne peux être tenu responsable des problèmes découlant de l'utilisation de ce soft (et système). Développer soit même son logiciel, permet d'y introduire des fonctions novatrices comme l'utilisation d'un vibreur en cas de perte de contact.

Pour fonctionner UTS utilise 2 fichiers de configuration :

- Config.txt : Placé dans le même répertoire que l'exécutable, ce fichier décrit toutes la configuration (aiguillages, itinéraires, déplacements, automatismes ...). Tout doit être indiqué dans le fichier de configuration. Le logiciel ne peut pas modifier ce fichier. Cette approche permet d'être plus efficace sans perdre de temps dans de nombreuses fenêtres de configurations. Le format de ce fichier sera décrit pour chaque module.
- Configuration_tco.txt : Ce fichier permet de stocker les TCO (dessins, couleurs ...). Le logiciel le modifie directement car il serait trop difficile à l'utilisateur de le manipuler. Si ce fichier n'existe pas, il est automatiquement crée. Il est mis à jour à chaque fermeture de programme ou lorsque vous souhaitez le sauvegarder.

Ces 2 fichiers décrivent complètement votre réseau.

Les différents modules du logiciel permettent :

- la conduite sélective
- la conduite automatique
- les itinéraires
- l'utilisation des TCO
- l'automatisation des animations
- la signalisation
- ...

L'interface graphique du logiciel permet d'afficher 8 TCO ainsi que des informations sur les différents modules.

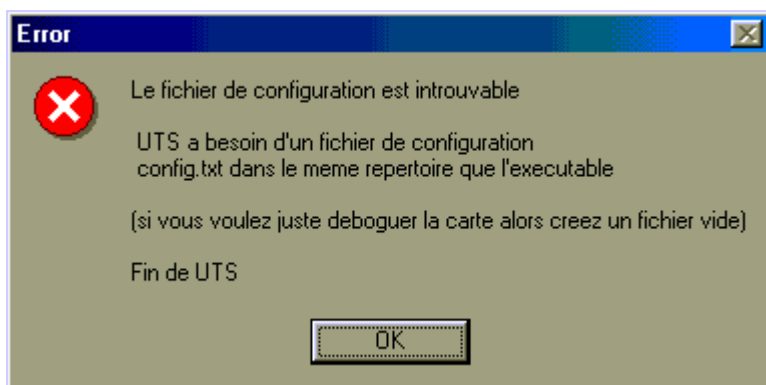
Bien entendu, il y aura toujours des choses à améliorer, d'autres à corriger ... Je compte sur vous pour me donner vos commentaires, signaler les bogues (bugs) (problèmes), ainsi que vos suggestions. La version 0.4 peut encore être améliorée, mais elle est suffisamment complète et stable pour une utilisation satisfaisante. De plus les amis modélistes commencent à s'impatier.

4.2. Téléchargement, installation et premier lancement

Le moment est venu d'installer le logiciel et de le lancer pour la première fois. Ce n'est pas encore la peine de connecter la carte.

- Créez un répertoire pour le système UTS
- Télécharger l'archive zip du site web,
- La décompresser dans ce répertoire
- Le répertoire dans contenir l'exécutable uts0200.exe
- Vous pouvez créer un raccourci afin de mettre une icône sur le bureau de Windows.
- Lancer UTS.

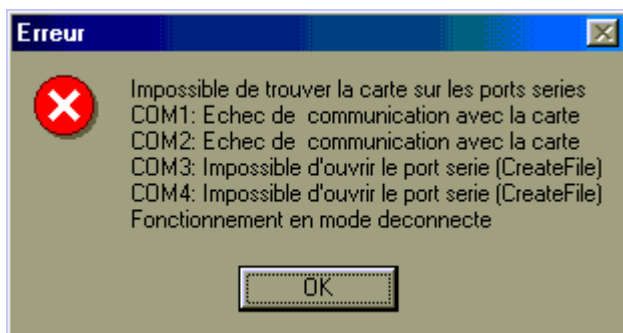
Vous verrez sûrement gratifier de l'erreur suivante :



Pas très brillant pour un premier lancement ... En fait UTS a besoin d'un fichier de configuration pour fonctionner. (Il est même possible de le faire fonctionner avec un fichier vide)

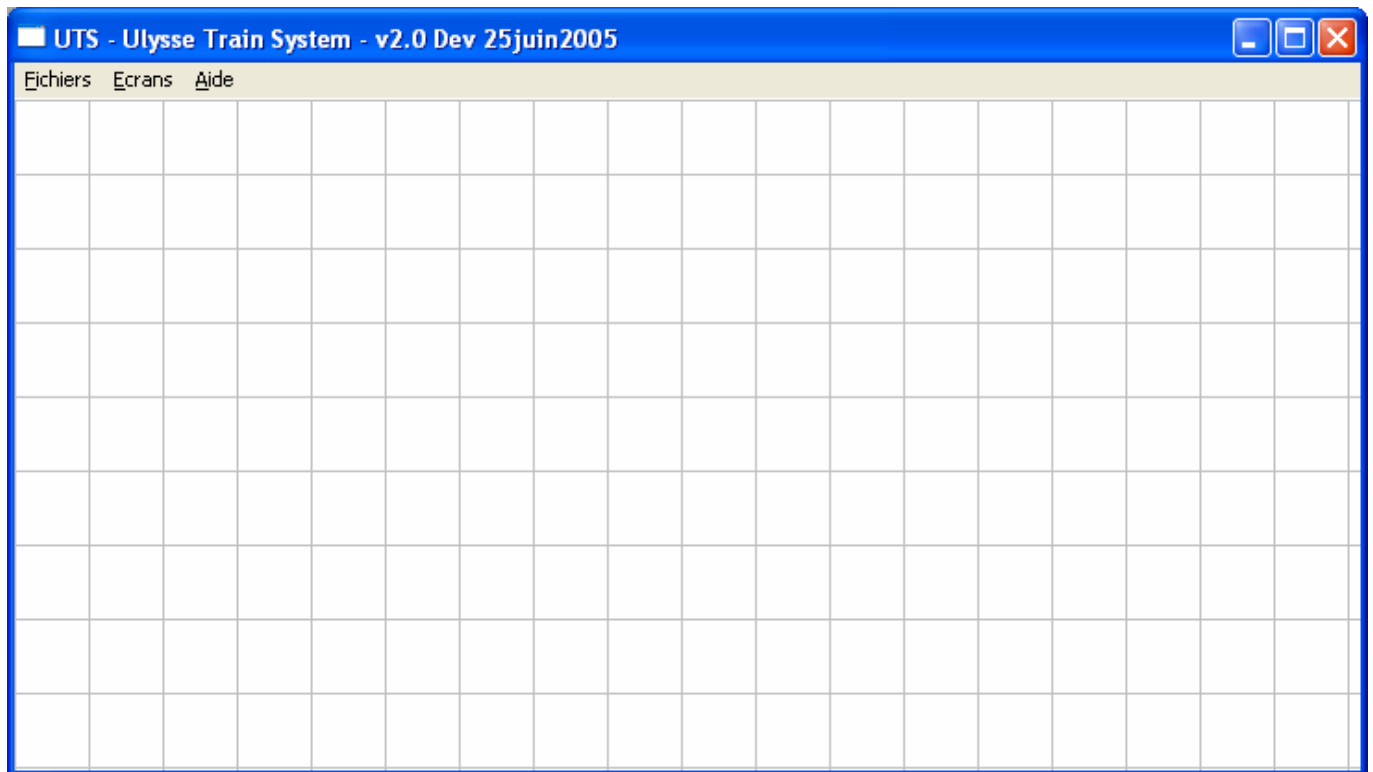
- Créez un fichier config.txt dans le même répertoire qu'UTS ou mieux copier config_vider.txt et renommer le. Ce fichier est intéressant car il contient toutes les sections et de l'aide.
- Relancez UTS

Une nouvelle fenêtre d'erreur s'ouvre :



Cette fenêtre indique que la carte n'a pas été trouvée. Ce qui est normal car elle n'est pas connectée. UTS recherche la carte sur les ports séries COM1 à COM4. La fenêtre indique que COM3 et COM4 n'existent pas sur cet ordinateur. En effet, la plupart des ordinateurs n'ont que 2 ports séries COM1 et COM2. Elle indique également que les ports COM1 et COM2 existent et ne sont pas utilisés par une autre application, mais la carte n'a pas été trouvée sur ces ports.

- Faites OK



Vous êtes enfin arrivé à lancer UTS ! Le résultat n'est pas spectaculaire car vous ne communiquez pas avec la carte et vous avez une magnifique TCO vide !
Cet état va néanmoins nous permettre de découvrir quelques points.

La barre des titres contient outre UTS la version (important pour me signaler un bogue) et la date de compilation

Les menus suivants sont disponibles :

- Fichier | Sortir : pour quitter l'application.
- Fichier | Editer config : ouvre le fichier de configuration config.txt
- Fichier | Recharger config : permet de recharger le fichier de configuration. Seuls certains modules peuvent être rechargés lorsque UTS est déjà lancé. Pour les autres il faut relancer UTS.

- Ecrans | TCO0 : affiche le TCO numéro 0.
- Ecrans | TCO1 : affiche le TCO numéro 1.
- Ecrans | TCO2 : affiche le TCO numéro 2.
- Ecrans | TCO3 : affiche le TCO numéro 3.
- Ecrans | TCO4 : affiche le TCO numéro 4.
- Ecrans | TCO5 : affiche le TCO numéro 5.
- Ecrans | TCO6 : affiche le TCO numéro 6.
- Ecrans | TCO7 : affiche le TCO numéro 7.
- Ecrans | TCO : ouvre la fenêtre d'édition des TCO
- Ecrans | TCO : ouvre la fenêtre de sélection des couleurs des éléments des TCO
- Ecrans | TCO sauver maintenant : permet de sauver les TCO (sinon ils sont sauvegardés à la fermeture du soft)
- Ecrans | Trains : permet de modifier les caractéristiques des trains (commande, positionnement, acc/dec/freinage ...)
- Ecrans | Déplacements : permet de déclencher, visualiser et tester les déplacements.
- Ecrans | Vibreur : pour indiquer le seuil de déclenchement du vibreur et visualiser son état.
- Ecrans | Horloge : affiche l'écran de l'horloge de simulation.
- Ecrans | Cantons : permet de choisir un mode et de visualiser les informations pour chaque canton.
- Ecrans | Garages : permet de tester les garages
- Ecrans | Aiguillages : permet de commander et tester les aiguillages.
- Ecrans | Itinéraires : permet de déclencher, visualiser et tester les itinéraires.
- Ecrans | RS232 : affiche l'écran de bidouillage de la carte. Cet écran permet d'essayer les sorties, voire les entrées ...

- Help | Help : Ouvre ce manuel s'il est disponible dans le répertoire d'UTS et si vous pouvez lire du Word.
- Help | About : Indique la version d'UTS et donne l'adresse du site web.

La zone centrale affiche les TCO ou des grilles d'information.

Vous savez maintenant, installer UTS, le lancer, changer d'écran et sortir. Il va falloir apprendre à écrire ce fameux fichier de configuration et connecter la carte.

4.3. Connexion, détection et test de la carte

Le moment est maintenant venu de connecter la (ou les) carte électronique afin de vérifier qu'elle fonctionne et communique bien avec le PC. Le test se fait facilement à travers une interface graphique.

- Branchez la carte à un des ports séries libres du PC.
- Alimentez la carte
- Vérifiez que le UC de la carte fonctionne grâce à la led jaune qui doit clignoter.
- Lancer UTS.

Bien entendu, il faut lancer UTS après la connexion et l'alimentation de la carte car sinon, il ne la trouvera pas. Si tout se passe bien, UTS devrait se lancer sans erreurs. Sinon, soit une autre application utilise le port, soit le câble est mal branché ou mal câblé, soit la carte a un problème. Il faut alors résoudre le problème et refaire la procédure.

- Allez dans le menu Ecran | RS232 pour afficher l'écran de bidouillage de la carte.

The screenshot shows the UTS software interface with the following components:

- Card Selection:** Buttons for card0 (green), card1, card2, and card3.
- Register Table:**

adr dec	adr hex	dat dec	dat hex	dat bin	actions		
0	0	0	0	00000000	W	R	A
0	0	0	0	00000000	W	R	A
0	0	0	0	00000000	W	R	A
0	0	0	0	00000000	W	R	A
- LED Grid:** A grid of LEDs labeled A through I, each with 4 sub-LEDs. The status of each LED is indicated by a number (0-7) or a color (red for off, green for on, blue for flashing).
- System Status:**
 - Err RS232: 0
 - En leds: On
 - En servo0: Off
 - En servol: Off
 - Tic nb : 918

L'écran RS232 se compose des zones suivantes :

- 4 boutons pour choisir la carte (vert=carte sélectionnée, rouge=carte disponible mais non sélectionnée, gris=carte non disponible, en fait non détectée au démarrage)
- 36 voyants pour indiquer et fixer l'état des leds (rouge=off, ver=on, jaune=clignote, bleu=clignote mais en opposition au jaune) Bien entendu les leds sont disponibles si En leds est à On.
- 24 voyants pour indiquer et fixer l'état des sorties. Si les leds ne sont pas utilisées, alors 16 sorties supplémentaires remplacent les leds.
- Un tableau pour examiner les registres du UC. Pour cela il suffit de cliquer sur une cellule, puis taper l'adresse ou la donner et valider. Et appuyer sur W pour écrire la donnée à l'adresse spécifiée ou R pour lire la valeur du registre dont

l'adresse est spécifiée ou A pour passer en lecture continue ou X pour stopper la lecture continue. Bien entendu lors d'une écriture il faut savoir ce que l'on fait ... En principe vous n'avez pas à vous servir de ce tableau pour une utilisation standard.

- Des voyants pour indiquer l'état des entrées (rouge = capteur inactif, vert = capteur actif). Dans l'exemple, le capteur in15 relié entre D et 3 est actif.
- Des boutons pour activer les leds ou les servos-moteurs. Je ne conseille pas l'utilisation des servos-moteurs car cela perturbe légèrement les sorties PWM. Cette partie affiche aussi les erreurs de liaison (qui devrait rester à 0) ainsi que le nombre de tic (repassé à 0 lorsqu'il atteint 1000). UTS communique avec la carte 4 fois par seconde, soit toutes les 250ms. 1 tic correspond à 250ms. Cette référence de temps sera très utilisée par la suite.
- Des barres graphes bleues pour indiquer la mesure des tensions analogiques et donc la position des potentiomètres. Les valeurs varient entre 0 et 255.
- Des voyants pour indiquer les détections des locomotives. Ici, les locomotives sont sur les cantons 3,8,10 et 12.
- Des potentiomètres pour indiquer et fixer la vitesse et le sens des locomotives sur chaque canton.
- Des boutons pour inverser les potentiomètres ou les cantons s'ils sont branchés à l'envers.
- Enfin la dernière ligne permet de mettre en relation un potentiomètre avec des cantons afin de réaliser de petits tests. Ainsi lorsque le potentiomètre bouge, alors les sorties pwm des cantons associées suivent. Pour cela sélectionnez un potentiomètre (le I doit se transformer en L), puis sélectionner les cantons. Bien entendu, il faut désactiver le contrôle de ces cantons par UTS (en enlevant la croix).

Les commandes sous le contrôle d'UTS sont barrées d'une croix. Pour en prendre le contrôle manuel à la place d'UTS, il suffit de cliquer avec le bouton droit de la souris. La croix s'efface, autorisant alors le contrôle manuel de la led, le potentiomètre ou la sortie. Il suffit alors de faire un click gauche pour passer les commandes.

Afin de s'adapter aux utilisateurs, les 2 registres à décalages des sorties les plus proches du UC peuvent piloter 48 leds (led0 à led47) par multiplexage ou gérer 16 sorties (out24 à out39). Il est possible de les désactiver les leds au profit des sorties supplémentaires.

Le sens positif ou « + », qui sera très utilisé par la suite est celui qui s'applique lorsque les potentiomètres des cantons sont dans la partie haute. Si les locomotives n'avancent pas dans le bon sens, alors inverser le branchement ou utiliser le bouton i.

Vous pouvez ainsi tester complètement votre carte. Par la suite, ce sera UTS qui remplacera vos actions manuelles. Lorsque UTS a le contrôle (avec la croix), vous pourrez voir l'état des sorties, leds et potentiomètres changer.

Comme UTS ne sauve rien dans le fichier de configuration alors si vous avez eu recours aux inversions ou sélection des leds, ils faudra mettre à jour le module général du fichier de config

```
#general
[
  @ off desactive les leds (out24 .. 39 sont utilisables)
  @ on active les leds (led0 .. 47 sont utilisables) (default)
  leds_card_0 on

  @ off desactive le servo moteur (default)
  @ on active le servo moteur
  servo_000_card_0 off
  servo_001_card_0 off

  @ off desactive l'affichage des cartes trouvees au lancement (default)
  @ on active l'affichage des cartes trouvees au lancement
  show_cards_found off

  @ off n'inverse pas le sens du canton ou du potentiometre (default)
  @ on inverse le sens du canton ou du potentiometre reactive
  inv_canton_0 off
  inv_canton_15 off
  inv_pot_0 off
  inv_pot_4 off
]
```

4.4. Le fichier de configuration

L'heure est enfin venue de nous intéresser au fichier de configuration.

Le fichier de configuration doit s'appeler config.txt et doit se trouver dans le même répertoire que l'exécutable.

Il est bien entendu possible d'avoir plusieurs fichiers de configuration et de les copier dans config.txt avant de lancer UTS.

J'ai choisi d'utiliser un fichier de configuration texte afin de faciliter l'efficacité, la compréhension et la simplicité par rapport à l'utilisation d'innombrables fenêtres de configurations. Pour éditer ce fichier, vous pouvez par exemple utiliser notepad. Ce fichier est un simple fichier texte. Il vaut mieux utiliser uniquement les minuscules. Sauf si besoin, pour les noms à afficher.

La syntaxe du fichier est la suivante :

```
#nom_de_la_section_en_minuscules  
[  
  commandes propres à la section  
  @ commentaires  
]
```

ce que vous voulez au dehors des sections.

Il y a souvent de l'aide

Le fichier est divisé en sections à l'intérieur desquelles les commandes propres à cette section sont rentrées. Il est possible d'utiliser @ pour faire un commentaire à l'intérieur d'une section. A l'extérieur, vous pouvez mettre les commentaires que vous voulez à condition qu'il n'y ait pas #nom d'une section. Pour désactiver une section, il suffit d'y mettre un nom non reconnu par UTS. S'il y a une erreur dans les commandes d'une section UTS signalera la section incriminée. Mais vaut mieux faire attention lorsque l'on tape les commandes car les vérifications ne sont pas très poussées. Si une section ou une commande n'est pas définie, alors UTS prendra la configuration par défaut.

Les sections suivantes sont actuellement utilisées.

```
- general : configuration d'ordre général de la carte ou des cartes  
- aig : configuration (connexions et commandes) des aiguillages  
- iti : Y itinéraires  
- selectif : définition du suivi des sur les cantons trains en manuel  
- train : définition des trains(nom, vitesses, commande ...)  
          + position au démarrage d'UTS  
- garage : configuration des voies de garage  
- horo : Y paramétrage de l'horloge + horodateur pour déclencher des équations  
        (ex les déplacements automatiques)  
- depl : Y définition des déplacements automatique  
- tdv : Y opération logiques (table de vérité, équations, impulsions)  
- exe : Y fichier à exécuter  
- vib : Y configuration du vibreur
```

Toutes les sections sont chargées par UTS au démarrage.

Pour un rechargement en cours d'utilisation, pour la stabilité d'UTS, seules les sections Y seront rechargées

Il est possible d'ouvrir le fichier depuis UTS et de le recharger à partir du menu.

Enfin comme indiqué plus haut, il existe un autre fichier de configuration « config_tco.txt ». Ce fichier permet de stocker les TCO (dessins, couleurs ...). Le logiciel le modifie directement car il serait trop difficile à l'utilisateur de le manipuler. Si ce fichier n'existe pas, il est automatiquement créé. Il est mis à jour à chaque fermeture de programme ou lorsque vous souhaitez le sauvegarder.

4.5. Les variables

Pour vous aider à contrôler tout ce que vous voulez sur votre réseau, UTS met à votre disposition plusieurs variables. Ces variables peuvent être utilisées dans le fichier de config et les TCO pour réaliser tout contrôle ou automatismes. Par exemple avec le module table de vérité, il est possible de gérer la signalisation en contrôlant les leds en fonction des détections et positions des aiguillages.

Nom	base	plage	écriture	valeurs	Description
det_	0	4x16		0-1	Détections locomotives (ex det_0, det_15, det_315)
can_	800	4x5		0-255	Valeurs des entrées analogiques
in_	400	4x36		0-1	Entrées matricées
out_	1200	4x40	Oui	0-1	Sorties
led_	1600	4x48	Oui	0-1	Allumage leds
led_cli_	2000	4x48	Oui	0-1	Mode clignotement
led_phase_	2400	4x48	Oui	0-1	Inversion de la phase du clignotement
aig_cmd	5000	64	Oui	0-1	Commandes des aiguillages
pos0_aig_	5100	64		0-1	Aiguillages en position 0 et verrouillés (pas en mouvement)

pos1_aig_	5200	64		0-1	Aiguillages en position 1 et verrouillés (pas en mouvement)
var_	5400	512	oui	0-1023	Libre pour l'utilisateur
Iti_start	6000	50	Oui	0-1 Z	Activation d'un itinéraire
Iti_stop	6100	50	Oui	0-1 Z	Désactivation d'un itinéraire
Iti_wait	6200	50		0-1	Itinéraire en attente
Iti_ok	6300	50		0-1	Itinéraire établi
depl_start_run	6500	100	Oui	0-1 Z2	Non utilisé
depl_stop	6600	100	Oui	0-1 Z	Arrêt prématuré d'un déplacement
garage_lock_cmd	6800	16	Oui	0-1	Commandes de verrouillage des garages
garage_lock_status	6900	16		0-1	Etats de verrouillage des garages
vib_on	7000		Oui	0-1	Activation du vibreur
vib_state	7002			0-1	Etat du vibreur
exe_	8400	100	Oui	0-1 Z	Lance un programme (surtout utile pour des sons)
pulse cmd	8500	32	Oui	0-1 Z	Declenchement d'une impulsion
pb capt c	8600	4x16		0-1	Problème de captage sur les cantons
pb capt t	9000	4x16		0-1	Probleme de captage sur les trains

Z : automatiquement remis a zéro après prise en compte

Z2 : automatiquement remis a zéro en fin de déplacement

La base sert à calculer le numéro de variable (util pour les tco). Par exemple la led 15 de la carte 2 a comme numéro 1600+200+15=1815

La plage indique le nombre de variable. 4x signifie qu'il faut remplacer le chiffre des centaines par le numéro de carte.

4.6. Les TCO

Les TCO (Tableau de Contrôle Optique) permettent de visualiser le réseau (ou une partie) à l'aide d'un plan sur lequel sont disposés des voyants. Des interrupteurs permettent également d'agir sur les équipements. UTS dispose de 8 TCO pour permettre de bien découper le réseau si nécessaire. Ces TCO sont composés de 32x25 cellules au maximum suivant les possibilités d'affichage de votre système.

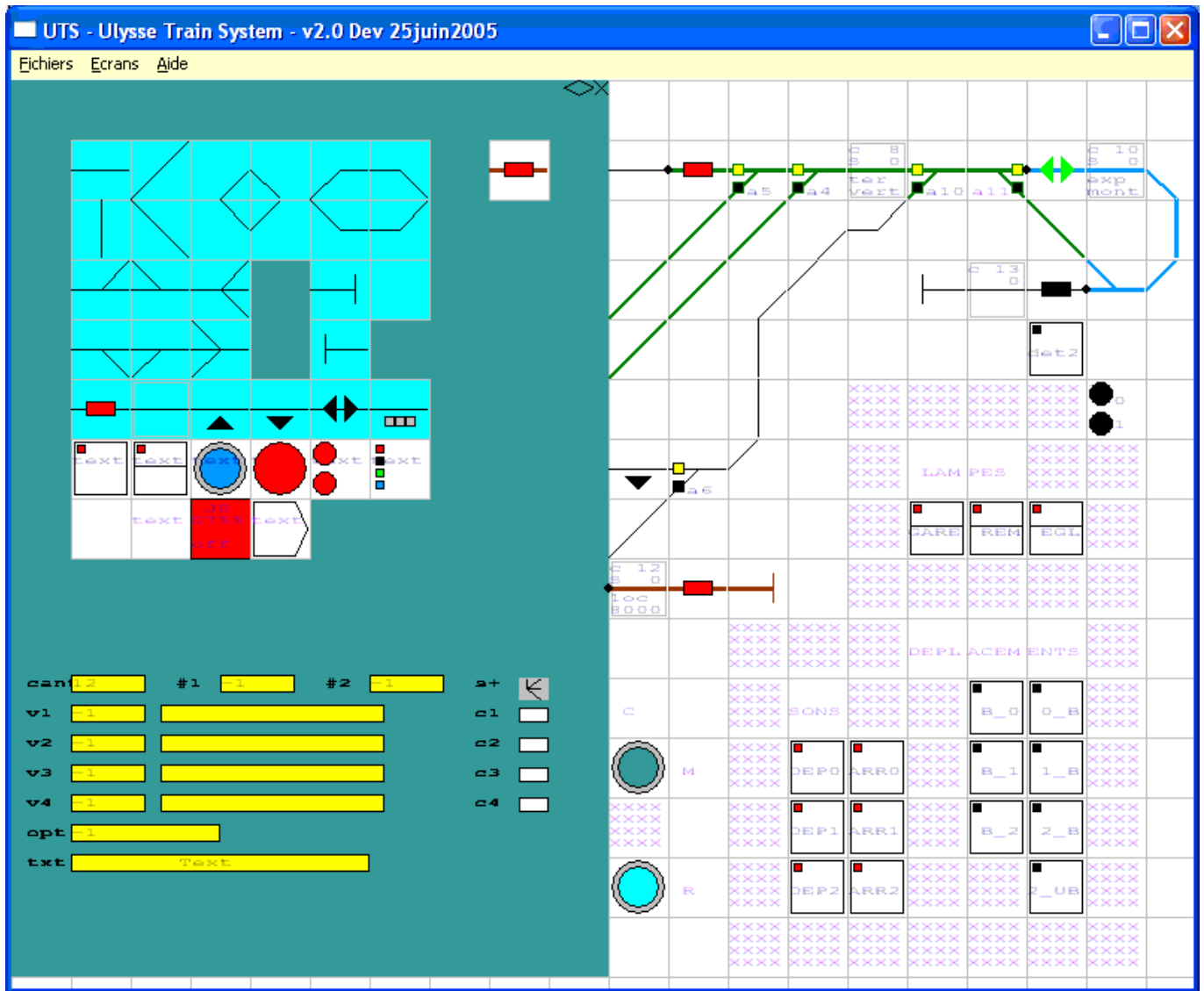
Toutes les informations sur les TCO sont sauvegardées dans le fichier « config_tco.txt ». C'est UTS qui gère directement ce fichier car il serait trop compliquer à l'utilisateur de le modifier directement. Le fichier est chargé à chaque lancement d'UTS et mis à jours à la sortie ou avec le menu « TCO sauver maintenant ».

Pour dessiner un TCO, il faut simplement utiliser la fenêtre d'édition accessible à partir du menu « TCO Edition ». Si la fenêtre cache la partie à modifier, alors cliquez sur les symboles < ou >. Pour fermer la fenêtre, utiliser le X. Pour mettre un symbole dans une cellule du TCO, choisir un symbole dans la fenêtre d'édition en cliquant dessus, régler ces paramètres, puis cliquer sur la cellule du TCO qui doit le recevoir. Lorsqu'un symbole est sélectionné, alors il apparaît dans le rectangle de prévisualisation (en haut à droite dans la fenêtre d'édition). Les paramètres à utiliser dépendent du symbole. Ils ne sont pas tous nécessaire à tous les symboles. Par exemple premier élément (la voie horizontal) ou l'indicateur de présence sélectionné ont besoin du numéro de canton (paramètre cant) et du sens positif (paramètre s+). Les autres ne servent à rien. Le numéro de canton sert à colorier et le sens à dessiner des flèches dans le bon sens. Il est également possible de charger un élément du TCO dans la fenêtre d'édition avec un click droit. Une fois dans la fenêtre d'édition, il est possible de modifier les paramètres puis de le remettre à sa place ou ailleurs avec un click normal. Cette méthode permet de réaliser des copier/coller.

Les paramètres sont les suivants :

- cant : numéro du canton de l'élément (tous les éléments cyans sur l'image en ont besoin à par la case vide). Pour un garage, indiquez le numéro du garage plus 1000. (pour g3, mettre 1003 dans cant).
- #1 et #2 sont des numéros surtout utilisés pour les numéros des aiguillages. Il y en a 2 car les aiguillages triples sont constitués de 2 moteurs.
- v1-4 sont des numéros de variable (le nom de la variable s'affiche à droite). Elles sont surtout utilisées pour les voyants et interrupteurs
- opt est un numéro.
- txt est le texte à afficher
- s+ est le sens positif. Sur l'exemple le sens positif est à droite (est) car l'indicateur indique (nord ou nord/est ou est ou sud/est). Il est utilisé pour les flèches. Il suffit de cliquer dessus pour le changer.
- Les couleurs peuvent rarement servir comme pour les feux ou l'horloge. La majorité des couleurs utiles se définissent dans la fenêtre de rendu.

Pour entrer une valeur, cliquez sur la case, appuyer sur espace pour l'initialiser puis rentrez la valeur.



Il est maintenant temps de décrire les symboles.

Les liaisons :

Ce sont des rails qui permettent de former le réseau. Il est souhaitable d'y associer un numéro canton ainsi qu'un sens positif afin de profiter des facilités d'affichage des TCO. Un petit point noir sépare automatiquement 2 cantons différents, mais pour l'instant par manque de temps il n'est rajouté que sur les section horizontales..

Paramètres : cant et s+

Les aiguillages :

En plus de cant et s+, il faut renseigner le numéro de l'aiguillage. La commande 0 correspond à la voie directe représentée, alors que la commande 1 à la voie déviée représentée. J'insiste sur représentée, car il peut arriver pour des raisons pratiques de représenter un aiguillage sur TCO différemment de celui disponible sur le réseau.

Pour un aiguillage triple, renseigner les 2 numéros. Le numéro 1 correspond à la direction du haut.

En mode d'exploitation (fenêtres d'édition et de rendu fermées), un simple click sur le symbole d'un aiguillage le fait changer de position. (Pour un aiguillage triple : a1=0 et a2=0 puis a1=1 et a2=0 puis a1=0 et a2=1 puis rebouclage ...)

Un petit cadenas rouge se dessine lorsque l'aiguillage est utilisé en automatique ou par un itinéraire. Il est alors impossible de le changer de position.

Paramètres : cant, s+, #1, [#2]

Les voyants d'occupations :

Se colorient avec la couleur det on lorsqu'une locomotive est détectée ou avec det off dans le cas contraire. Si le voyant appartient à un garage, alors il suit le même comportement que ceux du canton associé lorsque le garage est libre et indique une présence lorsque le garage est verrouillé.

Paramètres : cant et s+

Symbole de canton (le carré après le voyant d'occupation) :

Affiche des information sur la canton comme le nom du train présent, la vitesse ... Clignote rouge lorsqu'une perte de contact survient.

Paramètres : cant et s+

Capteur de position

Indique l'état d'un capteur de position. Par exemple ILS ou pédale de voie. De tel capteurs peuvent être utilisées pour améliorer la précision des arrêts car UTS ne dispose pas de zone d'arrêt dans les cantons. Il faut indiquer dans v1 le numéro de la variable qui gère ce capteur (sûrement une entrée). Le symbole affiche juste l'état du capteur, il ne fait rien d'autre

Paramètres : cant, s+ , v1

Pédale de voie virtuelle

Ce symbole permettra de commander une variable lorsqu'un train est détecté sur le canton. Par exemple pour les passages à niveau. Ce symbole ne fonctionne pas actuellement. Il faut utiliser une équation ou table de vérité pour le remplacer.

Paramètres : cant, s+ , v1

Sélection de sens

Ce symbole permet de sélectionner un sens pour un canton. Ainsi, les trains automatiques ou sélectifs voulant emprunter ce canton n'y seront pas autorisés. Il est possible de bloquer les 2 sens. Un click sur la partie droite gère le sens droit et inversement.

Paramètres : cant, s+

Feux

Représente un feu. Placez dans la zone de texte des 1 pour les ampoules de la variable 1 et choisir la couleur. Idem pour les variables 2 à 4. L'option permet de mettre un pied au feu. Il est ainsi possible de représenter une grande variété de feux.

Paramètres : cant, s+, texte, opt, v1-v4, c1-c4

Bouton poussoir

Permet de changer les variables d'états (par exemple des sorties)

Si V1 utilisée : mise à 1 de v1 au click et à 0 au relâchement

Si V2 utilisée : mise à 1 de v2 au click

Si V3 utilisée : mise à 0 de v3 au click

Paramètres : v1-v3

Interrupteur

Permet de changer les variables d'états (par exemple des sorties)

Changement d'état de V1 à chaque appui.

Paramètres : v1

Bouton à appui/traction

Il met à 1 la variable v1 lorsque l'on appuie sur le bouton (click).

Il met à 1 la variable v2 lorsque l'on tire sur le bouton (simulé par un click droit).

Le voyant s'allume lorsque la variable v3 est à 1 ou clignote lorsque la variable v4 est à 1.

Ce type de bouton au fonctionnement bizarre est surtout utilisé pour les itinéraires. Il faut lier v1 à une variable iti_start, v2 à iti_stop, v3 à iti_ok et v4 à iti_wait. Donc lorsque l'on appuie, l'itinéraire s'enclenche. Le bouton clignote jusqu'à ce que l'itinéraire soit opérationnel. En fin d'itinéraire, il n'y a plus qu'à tirer sur ce bouton pour le désactiver (J'ai vu ce type de fonctionnement dans un des postes d'aiguillage de la gare de Toulouse Matabiau).

Afin d'interdire des itinéraires, il est possible en réalité de rendre le bouton inactif en le bloquant physiquement au moyen d'un capuchon ou d'une rondelle. Ce fonctionnement est reproduit lorsque l'on clique sur le bord gauche du bouton.

Paramètres : v1,v2,v3,v4

Les voyants

Pour représenter l'activité des variables, il est possible d'utiliser 1, 2 ou 4 voyants.

1 voyant : v1 variable qui allume, v2 variable qui fait clignoter (v1 est prioritaire)

2 voyants : idem pour le voyant du haut. V3 et v4 pour celui du bas

4 voyants : Pas de clignotement possible v1 pour celui du haut jusqu'à v4 pour celui du bas.

Bien entendu pour ne pas faire clignoter un voyant, laisser sa variable de clignotement à -1.

Paramètres : v1,v2,v3,v4

La section voie permet de choisir les couleurs des rails et du fond des cellules qui représentent des voies en fonction de l'état du canton (vide, occupé par erreur, en mode automatique avec ou sans train, en mode manuel sélectif ou en prévision de mode manuel sélectif). Il est possible de choisir une couleur fixe ou une couleur pour chaque train (couleur T) ou canton (couleur C). Les couleurs T et C sont définies dans la partie cantons/trains. Il est également possible en fonction de ces modes de choisir l'épaisseur des rails et la présence d'une flèche de sens du train.

Ainsi cette puissante configuration convient tant au puriste souhaitant un minimum, qu'au joueur qui aime bien repérer rapidement ses trains. Par exemple, le puriste choisira des rails noirs sur fond blancs pour tous les modes et utilisera uniquement les voyants (rouge pour une détection et jaune pour une absence de détection). Alors qu'un autre, comme moi, préférera avoir des gros traits lorsqu'un train est détecté de la couleur du train avec les flèches. Ma configuration montre également que j'ai choisi de rajouter un fond jaune aux trains en automatique ainsi qu'un fond rouge pour une détection d'erreur lorsque le système détecte une locomotive ou il ne devrait pas y en avoir.

La section général permet de fixer les couleurs de fond / grilles / voyants et interrupteurs. « Ipp » correspond au bouton « push / pull » pousser/tirer décrit précédemment. Vous noterez qu'il est possible de choisir la couleur de la rondelle de blocage.

Pour changer une couleur ou une option, il suffit simplement de cliquer sur le rectangle associé. C'est vrai qu'il est un peu long de faire défiler les 16 couleurs, mais l'interface est super simple.

Voilà, vous savez tout sur les TCO. Vous pouvez faire tout ce que vous voulez et les rendre aussi réalistes que vous le voulez. Il est également possible de réaliser un vrai TCO en utilisant les leds et les entrées matricées. Afin de rendre les TCO encore plus réalistes, si vous travaillez à la SNCF et avez des descriptions ou images, je suis intéressé.

4.7. Les aiguillages

La section aig permet d'indiquer les sorties à activer pour piloter les aiguillages. Vous pouvez vous servir du menu Ecran | Aiguillages pour les tester. Et enfin, vous pouvez les commander et visualiser les positions à partir des TCO. UTS permet de gérer un maximum de 64 aiguillages. Tout type d'aiguillage peut être commandé (se référer à la section électronique correspondante). Les aiguillages sont commandés les uns après les autres afin de ne pas charger l'alimentation.

```
#aig
[
  @ aig0:  prep0:      sorties ...
  @      go0:
  @      lib0:
  @      prep1:
  @      go1:
  @      lib1:
  @
  default tic:2,2,0,2,2,0
  repos:      out15=0 out14=0 out13=0 out12=0 out11=0 out10=0

  @c0 c4 d0 d4
  aig2: prep0:      out15=1 out14=1 out13=0 out12=1 out11=0 out10=0 tic=10
      go0:      out15=1 out14=1 out13=0 out12=1 out11=0 out10=1
      prep1:    out15=1 out14=1 out13=0 out12=0 out11=0 out10=0
      go1:      out15=1 out14=1 out13=0 out12=0 out11=0 out10=1

  @c8 cc d8 dc
  aig1: prep0:      out15=1 out14=1 out13=0 out12=1 out11=1 out10=0
      go0:      out15=1 out14=1 out13=0 out12=1 out11=1 out10=1
      prep1:    out15=1 out14=1 out13=0 out12=0 out11=1 out10=0
      go1:      out15=1 out14=1 out13=0 out12=0 out11=1 out10=1

  ...
]
```

Cet exemple de section aig permet de voir la structure. Afin d'adapter UTS à tout type d'aiguillage.

La commande se passe en 3 phases :

- La préparation,
- Le déplacement
- La libération

Il est possible pour chacune de ces phases de définir l'état de 8 sorties max, ainsi que le temps à passer dans chacune des phases. Pourquoi 3 phases me diriez vous, car pour commander un solénoïde, il suffit d'une phase GO pour activer une bobine et une phase LIB pour remettre la sortie à 0 (ou alors une phase si la remise à 0 est faite dans la phase de repos).

En fait la phase de préparation est très utile pour les commande capacitive lorsqu'il faut charger un condensateur, ou encore dans les commandes multiplexées ou il vaut mieux mettre en position les relais permettant de choisir l'aiguillage avant de délivrer l'impulsion de commande.

Les temps s'expriment en tic (250ms), ainsi tic=4 signifie, rester 1 seconde dans cet état.

Entre chaque commande UTS insère une phase de repos d'au minimum 1 tic. Il est possible de définir des états à appliquer à 40 sorties max.

En général les temps des mêmes phases sont identiques pour tous les aiguillages. Pour ce faire, la directive default_tic permet de définir des temps par défaut (prep0/go0/lib0/rep0/rep1/go1/lib1). Mais il est toujours possible d'utiliser une directive tic. Pour ne pas griller les aiguillages, je vous conseil de tester avec des leds ou vous tenir prêt à couper l'alimentation lors de la mise au point.

L'exemple montre une commande de type capacitive. Il faut donc une phase de préparation pour charger le condensateur et sélectionner l'aiguillage, puis une phase de déplacement pour vider le condensateur dans la bobine sélectionnée de l'aiguillage sélectionné. Pour une commande directe, il n'y a pas besoin de préparation.

L'exemple suivant montre la commande directe d'aiguillages avec des impulsions de 500ms. Les 2 premiers aiguillages utilisent le repos pour éteindre les bobines alors que le 3eme se sert d'une phase de libération. Bien que cette phase dure 0 tic, cela suffit pour mettre à 0 les variables.

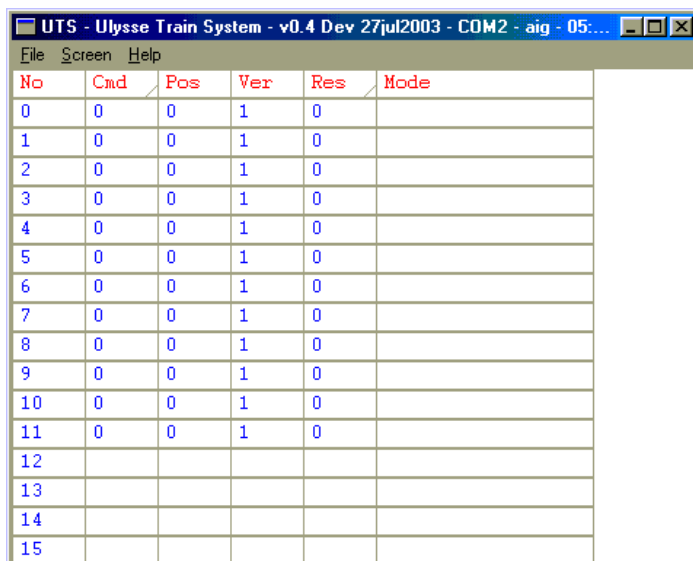
```
#aig
[
  default tic:0,2,0,0,2,0
  repos:      out10=0 out11=0 out12=0 out13=0

@aiguillage 0 (bobines sur out10 pour position 0 et out11 pour position 1)
  aig0: go0:      out10=1 out11=0
        go1:      out10=0 out11=1

@aiguillage 1 (bobines sur out12 pour position 0 et out13 pour position 1)
  aig0: go0:      out12=1 out13=0
        go1:      out12=0 out13=1

@aiguillage 2 (bobines sur out14 pour position 0 et out15 pour position 1)
  aig0: go0:      out14=1 out15=0
        lib0:      out14=0 out15=0
        go1:      out14=0 out15=1
        lib1:      out14=0 out15=0
  ...
]
```

L'écran de debug des aiguillages permet de les tester si par exemple le TCO n'est pas pret. Il est accessible par le menu « Ecran/Aiguillages ».



No	Cmd	Pos	Ver	Res	Mode
0	0	0	1	0	
1	0	0	1	0	
2	0	0	1	0	
3	0	0	1	0	
4	0	0	1	0	
5	0	0	1	0	
6	0	0	1	0	
7	0	0	1	0	
8	0	0	1	0	
9	0	0	1	0	
10	0	0	1	0	
11	0	0	1	0	
12					
13					
14					
15					

Comme pour tous les tableaux d'UTS, un triangle inférieur indique qu'il est possible de cliquer sur les cellules de la colonne. Les touches - et + permettent de faire défiler les lignes.

- Les cellules cmd permettent commander chaque aiguillage par la colonne cmd.
- Les cellules Res permettent de voir la réservation des aiguillages (pour un itinéraire ou un déplacement). Il est également possible de modifier cette cellule.
- Les cellules Pos indiquent la position actuelle des aiguillages
- Les cellules ver indiquent si l'aiguillage est verrouillé.

Mais cet écran sert surtout pour la mise au point car le but est quand même de commander les aiguillages avec le TCO ou les itinéraires. Vous pouvez aussi vous servir des variables des aiguillages.

Les variables io_aig_cmd_0..31 commandent les aiguillages

Les variables io_aig_pos_0..31 indiquent la position les aiguillages

Les variables io_aig_ver_0..31 indiquent le verrouillage des aiguillages mais elles ne sont pas vraiment utiles.

4.8. Les itinéraires

Un itinéraire définit la position de plusieurs aiguillages. La section iti définit les itinéraires. Vous pouvez vous servir du menu Screen | Itinéraires pour les tester. Et enfin, vous pouvez les commander et les visualiser à partir des TCO. UTS permet de gérer un maximum de 50 itinéraires.

UTS définit 2 types d'itinéraires :

- Les directes qui mettent à jours les commandes des aiguillages lorsqu'on les active. Ils se désactivent aussi tôt après.
- Les onoff qui réservent les aiguillages lorsqu'on les active et demande une désactivation volontaire. Ils peuvent également être déclenchés manuellement puis attendre une locomotive sur un canton et désactiver l'itinéraire lorsque la locomotive est sur un autre canton spécifié.

Section aig

```
#iti
[
  iti0:  onoff  a0.1                               name: A 0 @ blabla
  iti1:  onoff  a0.0 a1.0                           name: A_1
  iti2:  onoff  a0.0 a1.1                           name: A_2
  iti3:  onoff  a5.1 a4.0                           name: 0 B
  iti4:  onoff  a2.0 a4.1                           name: 1 B
  iti5:  onoff  a3.1 a2.1 a4.1                      name: 2 B
  iti6:  onoff  a3.0 a6.0                           name: 2 M
  iti7:  onoff  a0.1 a5.1 a4.0                      name: A_0 B
  iti8:  onoff  a0.0 a1.0 a2.0 a4.1                name: A_1_B
  iti9:  onoff  a0.0 a1.1 a3.1 a2.1 a4.1          name: A_2 B
  iti10: onoff  a0.0 a1.1 a3.0 a6.0                name: A_2 M

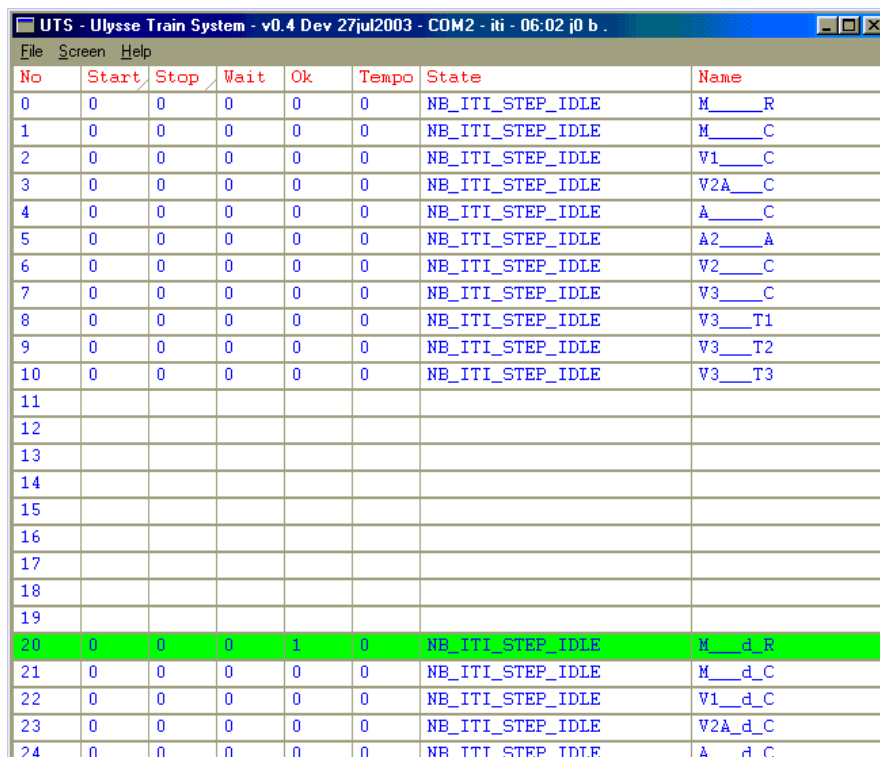
  iti20: direct a8.1 a9.0                           name: 4 C @ blabla
  iti21: direct a8.0 a9.0                           name: 5_C
  iti22: direct a8.0 a9.1                           name: 6_C
  iti23: direct a7.1 a6.1                           name: C M
  iti24: direct a7.0 a6.0                           name: C R

  @ iti31: onoff  a9.1 a4.0 a5.1      start c2 stop c4 @ M <--> U
]
```

Cet exemple de section iti montre la structure. AX.Y = met l'aiguillage X en position Y. cZ = attend une locomotive sur le canton Z. Il est également possible de nommer les itinéraires afin de s'y retrouver plus facilement.

Ecran iti

L'écran des itinéraires permet de visualiser, déclencher, stopper et tester les itinéraires. (utilisez les touches + et - pour vous déplacer dans le tableau)



No	Start	Stop	Wait	Ok	Tempo	State	Name
0	0	0	0	0	0	NB_ITI_STEP_IDLE	M___R
1	0	0	0	0	0	NB_ITI_STEP_IDLE	M___C
2	0	0	0	0	0	NB_ITI_STEP_IDLE	V1___C
3	0	0	0	0	0	NB_ITI_STEP_IDLE	V2A___C
4	0	0	0	0	0	NB_ITI_STEP_IDLE	A___C
5	0	0	0	0	0	NB_ITI_STEP_IDLE	A2___A
6	0	0	0	0	0	NB_ITI_STEP_IDLE	V2___C
7	0	0	0	0	0	NB_ITI_STEP_IDLE	V3___C
8	0	0	0	0	0	NB_ITI_STEP_IDLE	V3___T1
9	0	0	0	0	0	NB_ITI_STEP_IDLE	V3___T2
10	0	0	0	0	0	NB_ITI_STEP_IDLE	V3___T3
11							
12							
13							
14							
15							
16							
17							
18							
19							
20	0	0	0	1	0	NB_ITI_STEP_IDLE	M___d_R
21	0	0	0	0	0	NB_ITI_STEP_IDLE	M___d_C
22	0	0	0	0	0	NB_ITI_STEP_IDLE	V1___d_C
23	0	0	0	0	0	NB_ITI_STEP_IDLE	V2A___d_C
24	0	0	0	0	0	NB_ITI_STEP_IDLE	A___d_C

Variables

Les variables `iti_start_0..49` déclenchent les itinéraires.

Les variables `iti_stop_0..49` stoppent les itinéraires.

Les variables `iti_wait_0..49` indiquent que l'itinéraire est en cours d'établissement.

Les variables `iti_ok_0..49` indiquent que l'itinéraire est établi.

Il est possible de reproduire le comportement réel des postes d'aiguillages en mettant un bouton pousser/tirer qui clignote lorsque l'itinéraire est en cours d'établissement et reste allumé sans clignoter lorsque l'itinéraire est établi. Pour établir l'itinéraire il suffit d'appuyer. Pour le désactiver, il faut le tirer (click droit). Voir la description de ce bouton dans la section TCO.

4.9. Les trains

UTS permet de gérer 16 trains. Cette limitation est purement logicielle et si vous le demandez, il est tout à fait possible d'en rajouter. Chaque train est défini par un numéro mais il est possible de lui ajouter les attributs suivants :

- un nom (vaut mieux utiliser un nom de 8 lettres sans espaces pour qu'il s'affiche sur le TCO)
- une commande (un des 5 potentiomètres par carte (20max) ou un des 5 potentiomètres virtuels gérés par le clavier ou par la télécommande).
- une autorisation de fonctionner en automatique
- des caractéristiques dynamique (vitesse min et max, accélération, ralentissement, freinage)

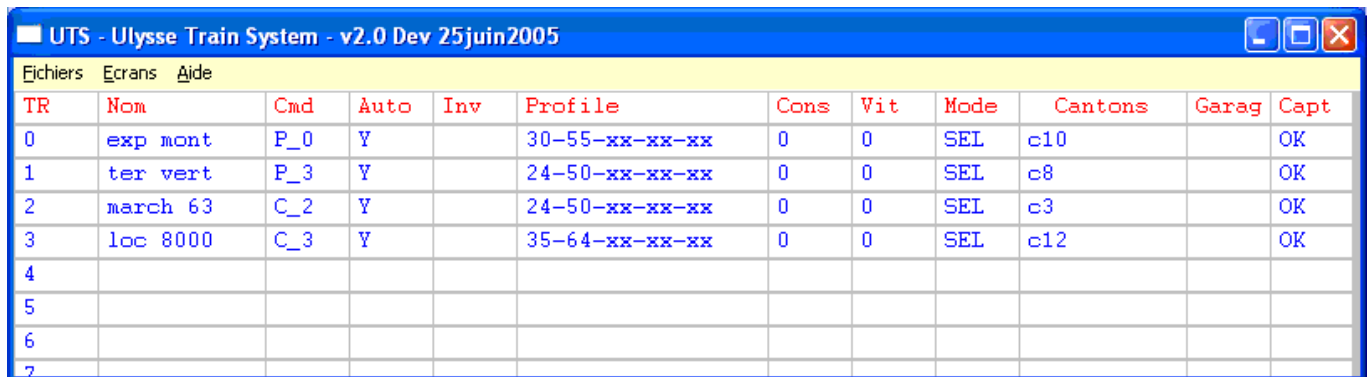
Bien entendu, il faut aussi spécifier à UTS où se trouve le train.

Ces caractéristiques peuvent être rentrées dans la section train et mise à jour (sans sauvegarde) dans l'écran train.

```
#train
[
  train0 : exp_mont c10 pot0 30-55-xx-xx-xx en_auto
  train1 : ter_vert c8 pot3 24-50-02-01-03 en_auto
  train2 : march_63 c3 clav2 24-50-xx-xx-xx en_auto
  train3 : loc_8000 c12 clav3 35-64-xx-xx-xx

  clav_conf_0 : +a !q -w
  clav_conf_1 : !s -x +z
  clav_conf_2 : +e !d -c
  clav_conf_3 : +r !f -v
  clav_conf_4 : +t !g -b
]
```

On retrouve les paramètres + les touches du clavier pour les potentiomètres virtuels.



TR	Nom	Cmd	Auto	Inv	Profile	Cons	Vit	Mode	Cantons	Garag	Capt
0	exp mont	P_0	Y		30-55-xx-xx-xx	0	0	SEL	c10		OK
1	ter vert	P_3	Y		24-50-xx-xx-xx	0	0	SEL	c8		OK
2	march 63	C_2	Y		24-50-xx-xx-xx	0	0	SEL	c3		OK
3	loc 8000	C_3	Y		35-64-xx-xx-xx	0	0	SEL	c12		OK
4											
5											
6											
7											

Pour ajouter ou enlever un train, cliquez dans la colonne TR. Pour modifier les autres colonnes, cliquez dessus.

Le potentiomètre de commande permet de faire varier la vitesse entre 0 (au centre), +64 (à une extrémité) et -64 (à l'autre extrémité). Il est également possible de définir une vitesse minimal (correspondant à la valeur PWM qui suffit à démarrer le moteur) et une vitesse maximal (pour éviter qu'une draine se transforme en TGV). Les potentiomètres virtuels ne possèdent que 3 vitesses dans chaque sens (min, moitié (min, max) et max). Sans paramètres acc/dec/frein, le potentiomètre donne la consigne qui est directement appliquées à la locomotive. Inv peut servir à inverser les ordres de commandes.

Les paramètres acc/dec/frein permettent de simuler l'inertie des rames. Pour ceux qui n'ont jamais conduit un train avec de l'inertie, les premières minutes seront déroutantes et vous défoncerez à coup sûr quelque heurtoir ! L'inertie amène un grand pas vers plus de réalisme et est un vrai plaisir. Ces paramètres s'expriment en évolution de vitesse par tic. Ainsi une configuration passe partout de 2/1/3 permet d'accélérer de 2 pas toutes les 250ms soit 8 pas par seconde, ralentir de 4 pas par seconde et freiner à 12 pas par seconde. UTS sait qu'il doit freiner lorsque vous appuyez sur le bouton frein, accélérer lorsque la consigne est supérieure à la vitesse et ralentir dans le cas contraire. Il est vrai que ces valeurs sont assez timides, mais elles permettent déjà d'entrer dans un nouveau monde. Pour faire un pas supplémentaire, essayer avec 1/0/1, pour vous obliger à freiner ... Si vous avez besoin de plus d'inertie (surtout pour les grand réseaux), dites le moi. Car il est vrai qu'avec l'inertie maximal de 1 la pleine échelle est parcouru en 16 secondes (essayez avec le paramètre de décélération à 0).

Il est à noter que le mode automatique utilise aussi un profile dynamique afin d'avoir des changements de vitesse sans à-coups. Mais pour une plus grande homogénéité, il utilise un profile 5/5/7 pour tous les trains.

A ce stade votre train ne pourra pas quitter son canton car le mode sélectif n'est pas encore défini.

4.10. Le mode sélectif

Le mode sélectif permet de conduire un train avec un seul potentiomètre. Dans ce mode UTS fait suivre les consignes du potentiomètre sur les cantons qu'empreinte le train. Ce mode ressemble un peu à la commande numérique mais ici, il ne peut y avoir qu'un seul train par canton. Le mode sélectif permet de gérer les retournements, boucles ... Bien entendu, il peut y avoir plusieurs trains dont chacun est commandé par un potentiomètre spécifique.

La section sélective permet d'entrer la topologie du réseau qui sera utile à l'algorithme du mode sélectif.

L'algorithme est le suivant :

Pour chaque canton libre : S'il y a un train sur un canton sélectif adjacent qui vient dans la direction du canton libre avec une vitesse ≥ 5 alors ce canton devient une destination possible du train et les consignes du trains sont appliqués.

```
#selectif
[
    c0:+ c6+   c0:- c8-
    c1:+ c6+   c1:- c8-
    c2:+ c6+   c2:- c8-   c2:- c9-
    c3:- c11-
    c4:- c11-
    c5:- c11-
    c6:+ c7-   c6:- c0-   c6:- c1-   c6:- c2-
    c7:+ c6-   c7:- c8-
    c8:+ c0+   c8:+ c1+   c8:+ c2+   c8:+ c7+   c8:+ c9+   c8:- c10+
    c9:+ c2+   c9:+ c11+  c9:- c8-
    c10:+ c8+  c10:+ c13+
    c11:+ c3+  c11:+ c4+   c11:+ c5+  c11:- c9-  c11:- c12-
    c12:+ c11+
    c13:- c10-
]
```

La section indique pour chaque canton libre dans le sens spécifié (+ correspond aux vitesses positives et - aux négatives), les cantons et les sens d'où le train peut surgir. Par exemple pour la première ligne, le canton 0 attend un train sur le canton 6 dans le sens + et s'il en trouve un alors il copiera les consignes. Idem pour le canton 8 mais dans le sens -. Si les sens sont différents, par exemple suite à une raquette ou triangle ou diagonale de retournement alors il y aura comme à la dernière 7eme ligne $c6:+ c7-$ une inversion de sens automatique sur le canton qui reçoit. Cet exemple est celui de mon réseau dont une copie d'écran du TCO est disponible dans la rubrique TCO. Vous pouvez ainsi faire le lien et voir si vous avez compris.

Si plusieurs trains se dirigent vers le même canton alors il y a contention et UTS favorise le train qui porte le plus petit numéro. Mais afin d'éviter les contentions, lorsque plusieurs choix sont possibles, il est possible d'utiliser les aiguillages.

```
#selectif
[
    c0:+ c6+ a0.1   c0:- c8- a5.1 a4.0
    c1:+ c6+ a0.0 a1.0   c1:- c8- a2.0 a4.1
    c2:+ c6+ a0.0 a1.1   c2:- c8- a2.1 a4.1 a3.1   c2:- c9- a3.0 a6.0
    c3:- c11- a8.1 a9.0
    c4:- c11- a8.0 a9.0
    c5:- c11- a8.0 a9.1
    ...
]
```

Sur ce second exemple tronqué, la première ligne indique que le canton 0 attend un train du canton 6 dans le sens + si l'aiguillage 0 est en position déviée (1).

Si un train arrive de chaque coté, alors UTS choisira celui de plus petit numéro, à moins que vous utilisiez le control d'autorisation de sens disponible sur les TCO.

Je conseil de créer une section avec les aiguillage et une sans et de choisir en fonction de l'humeur du moment en donnant un nom bidon à la section non utilisé. Une section sans aiguillage permet de bouger les aiguillages à la main (éviter alors de faire de l'automatique). Alors qu'une section avec, évite beaucoup de contentions.

Evitez d'arrêter une locomotive à cheval sur 2 cantons. Cela peu mettre à mal l'algo de sélection. Dans ce cas le TCO indiquera sûrement une erreur d'occupation et il faut alors pousser la loco sur le canton suivant (ou UTS s'attend à trouver sa loco).

L'écran canton donne des infos sur ce mode (ainsi que sur le mode automatique). Il ne vous sera pas trop utile, d'autant que les TCO permettent maintenant de suivre la prévision d'évolution des trains en sélectif. mstS correspond aux cantons sélectif avec un train alors que updS aux cantons susceptibles d'accueillir un train en sélectif.

UTS - Ulysse Train System - v2.0 Dev 25juin2005										
Fichiers Ecrans Aide										
Cant	Train	ResA	VitA	mstS	updS	invS	VitS	Vit	Det	
0	-1	0	0	0	0	0	0	0	0	
1	-1	0	0	0	0	0	0	0	0	
2	-1	0	0	0	0	0	0	0	0	
3	2	0	0	1	0	0	0	0	1	
4	-1	0	0	0	0	0	0	0	0	
5	-1	0	0	0	0	0	0	0	0	
6	-1	0	0	0	0	0	0	0	0	
7	-1	0	0	0	0	0	0	0	0	
8	1	0	0	1	0	0	0	0	1	
9	-1	0	0	0	0	0	0	0	0	
10	0	0	0	1	0	0	0	0	1	
11	-1	0	0	0	0	0	0	0	0	
12	3	0	0	1	0	0	0	0	1	
13	-1	0	0	0	0	0	0	0	0	
14	-1	0	0	0	0	0	0	0	0	
15	-1	0	0	0	0	0	0	0	0	
100	-1	0	0	0	0	0	0	0	0	
101	-1	0	0	0	0	0	0	0	0	

4.11. Garages (TBD)

Un garages est une voie (souvent de garage) qui est connectée électriquement à un canton par l'intermédiaire d'un simple relais commandé par UTS. Il ne peut y avoir qu'une seule locomotive active sur un canton et ses garages, mais il peut y avoir autant de locomotives inactives que de cantons. Grâce aux garages, il est possible d'économiser des cantons et donc d'augmenter la taille du réseau simplement.

Actuellement UTS supporte 16 garages, mais cette limite est purement logicielle. Si vous en avez besoin, je peux augmenter cette limite. Pour choisir entre un garage ou un canton, il suffit de se demander si la locomotive sur la voie en question doit pouvoir bouger lorsqu'une locomotive est sur le canton. Si non, il est possible d'utiliser un garage. Bien entendu, s'il vous reste des sorties PWM, vous pouvez utiliser des cantons à la place d'un garage pour économiser un relais. Il est vrai que la plupart des garages correspondront aux vraies voies de garage, c'est-à-dire souvent celles se terminant par un heurtoir. Mais ce n'est pas toujours le cas.

Pour réaliser un garage, il suffit d'isoler électriquement (avec une écluse isolante) la voie en question du canton en coupant un rail. Puis de rajouter un relais pour connecter ou non le rail du canton à celui du garage. Lorsqu'un garage n'est pas utilisé, alors il est disponible et la liaison avec le canton doit être réalisée. A l'inverse lorsque l'on souhaite y immobiliser une locomotive, le garage est utilisé et il faut rompre la liaison avec le canton. Ainsi une autre locomotive pourra utiliser le canton. Si le canton est plus souvent libre qu'occupé, donc souvent connecté, il est préférable que le garage soit connecté lorsque le relais est eu repos (avec les contacts C-R) afin d'alimenter la bobine le moins souvent possible pour économiser l'énergie de commande et donc avoir besoin d'une alimentation moins puissante. Et inversement. La commande polarité indique l'état de la sortie lorsque le garage doit être verrouillé.

Pour un garage sur le TCO, il suffit de rajouter 1000 au numéro du garage pour le paramètre « cant ». Le symbole TCO de canton permet de verrouiller ou déverrouiller le garage lorsque l'on clique dessus. Bien entendu, lorsqu'un garage est verrouillé, UTS s'attend à trouver un train sur ce garage (en fait une absence de détection sur le canton correspondant lorsque le garage est déconnecté). Les cantons peuvent aussi être utilisés pour les déplacements en automatique.

La section garage permet de déclarer les garages à UTS en indiquant le canton rattaché, la sortie commandant le relais ainsi que l'état de la sortie lorsque le garage doit être isolé.

```
#garage
[
  g1: c8 out4 poll
  g2: c8 out5 poll
]
```

L'écran « Ecran | Garage » permet de tester les garages. (locked veut dire verrouillé en anglais)

UTS - Ulysse Train System - v2.0 Dev 25juin2005							
Fichiers	Ecrans	Aide					
Garage	Canton	Out	Pol	Cmd_L	Locked	Res	Train
0	-1	0	1	0	0	0	-1
1	8	4	1	0	0	0	-1
2	8	5	1	0	0	0	-1
3	-1	0	1	0	0	0	-1
4	-1	0	1	0	0	0	-1
5	-1	0	1	0	0	0	-1
6	-1	0	1	0	0	0	-1
7	-1	0	1	0	0	0	-1
8	-1	0	1	0	0	0	-1
9	-1	0	1	0	0	0	-1
10	-1	0	1	0	0	0	-1
11	-1	0	1	0	0	0	-1
12	-1	0	1	0	0	0	-1
13	-1	0	1	0	0	0	-1
14	-1	0	1	0	0	0	-1
15	-1	0	1	0	0	0	-1

4.12. Déplacements

Les déplacements définissent des trajets d'un canton à un autre pour les trains. 100 déplacements sont possibles. Ils peuvent fonctionner en même temps s'ils n'utilisent pas les mêmes cantons et aiguillages. Ils sont en général lancés par l'horloge. Les déplacements associés à l'horloge forment le mode automatique. L'écran des déplacements permet de suivre les déplacements en cours, ainsi que d'en déclencher ou en arrêter manuellement. La section depl définit les déplacements.

L'algorithme simplifié d'un déplacement est le suivant

- attendre start (qui vient souvent de l'horloge)
- attendre que les cantons et les aiguillages utilisés soient libres.
- réserver les cantons et les aiguillages utilisés. (Ils ne peuvent alors pas être utilisés par quelqu'un d'autre)
- mettre les consignes de vitesses et de sens sur les cantons utilisés
- attendre la condition de fin

Section depl

```
#depl
[
d0 c0+30 c8+40 c10+30 a5.1 a4.0 a10.0 fin0:det_10 taf7 tsf0 name:dv0_n start:c0
d1 c10+30 c8-40 c7-40 c6+40 c0+30 a10.0 a4.0 a5.0 a0.1 fin0:det_0 taf8 tsf0 name:c_dv0 start:c10
d2 c1-30 c6-40 c7+40 c8+40 c10+30 a1.0 a0.0 a5.0 a4.0 a10.0 fin0:det_10 taf7 tsf0 name:dv1_c start:c1
d3 c10+30 c8-40 c1-30 a10.0 a4.1 a2.0 fin0:det_1 taf8 tsf0 name:n_dv1 start:c10

d4 c2+35 c9+40 c8+40 a3.0 a6.0 a10.1 fin0:det_8 taf1 tsf0 name:sr_dv2_b start:c2
d5 c8-40 c9-40 c2-35 a10.1 a6.0 a3.0 fin0:det_2 taf6 tsf0 name:sr_b_dv2 start:c8
d6 c8-40 c1-30 a10.0 a4.1 a2.0 fin0:det_1 taf7 tsf0 name:sr_n_dv1 start:c8
d7 c1-30 c6-40 c7+40 a1.0 a0.0 fin0:det_7 taf4 tsf0 name:sr_dv1_fond start:c1
d8 c7-40 c6+40 c2+30 a1.1 a0.0 fin0:det_2 taf2 tsf0 name:sr_fond_dv2 start:c7
d9 c2+30 c8+40 a2.1 a3.1 a4.1 fin0:det_8 taf9 tsf0 name:sr_dv2_n start:c2
d10 c2-30 c6-40 c7+40 a1.1 a0.0 fin0:det_7 taf4 tsf0 name:sr_dv2_fond start:c2

d11 c1-30 c6-40 c7+40 a1.0 a0.0 fin0:det_7 taf2 tsf0 name:dv1_fond start:c1
d12 c7+40 c8+40 c10+30 a5.0 a4.0 a10.0 fin0:det_10 taf7 tsf0 name:fond_c start:c7
]
```

- d + numéro identifie le déplacement
- name permet de lui donner un nom. (vous ne pouvez pas utiliser d'espaces)
- Les c indiquent les cantons à utiliser, ainsi que le sens et la vitesse.
- Les a spécifient les aiguillages à utiliser et dans quelles positions.
- Fin0 spécifie la condition de fin. (Le canton de fin est souvent utilisé)

- Une seconde condition de fin fin1 peut être ajoutée. Dans ce cas, il faut que la première condition ai été satisfaite au moins une fois avant que la seconde le soit. Ce mécanisme sert surtout, lorsque l'on veut utiliser un déplacement qui à le même canton de départ, et d'arrivé. Il suffit de mettre un canton du parcourt en fin0 et le canton de départ/arrivé en fin 1. Sans cette précaution, le déplacement serait aussitôt terminé avant même d'avoir commencé car au départ, la locomotive se trouve sur le canton de départ qui est aussi celui de fin...
- Taf pour tempo avant fin autorise une attente entre la détection de la fin et la mise à 0 des cantons utilisés. Cette temporisation est très utile car la première détection de la locomotive intervient juste quand elle rentre sur le canton et donc le reste du train est encore sur le canton précédent. Cette solution n'est pas géniale, car la durée de la temporisation doit dépendre de la machine, du nombre de wagons ... Mais elle est très simple à mettre en œuvre. Pour plus de précision, il est possible d'utiliser une pédale de voie, ou un ILS, ou un essieu résistif pour le dernier wagon du train.
- Tsf pour tempo suivant fin lance une temporisation entre le moment ou les cantons sont mis à 0 et l'instant ou le déplacement est libéré. Dans ce dernier instant, le déplacement rend disponibles tous ses cantons et aiguillages pour les autres déplacements.
- Il est également possible de jouer avec 4 variables (par ex pour lancer des annonces sdeb : exe_12) avec les directives :
 - sdeb: mettre la variable a 1 au debut du deplacement
 - cdeb: mettre la variable a 0 au debut du deplacement
 - sfin: mettre la variable a 1 au debut du deplacement
 - cfin: mettre la variable a 0 au debut du deplacement
 - sc : mettre la variable a 1 au debut du deplacement et a 0 a la fin

Ecran des déplacements

No	Start	Stop	Tempo	State	Name
0	0	0	0	DEPL_CST_STATE_IDLE	V0est_bcl
1	0	0	0	DEPL_CST_STATE_IDLE	bcl_V0est
2	0	0	0	DEPL_CST_STATE_IDLE	V0ouest_bcl
3	0	0	0	DEPL_CST_STATE_IDLE	bcl_V0ouest
4	0	0	0	DEPL_CST_STATE_IDLE	V1est_bcl
5	0	0	0	DEPL_CST_STATE_IDLE	bcl_V1est
6	0	0	0	DEPL_CST_STATE_IDLE	V1ouest_bcl
7	0	0	0	DEPL_CST_STATE_IDLE	bcl_V1oue
8	0	0	0	DEPL_CST_STATE_IDLE	V2est_bcl
9	0	0	0	DEPL_CST_STATE_IDLE	bcl_V2est
10	0	0	0	DEPL_CST_STATE_IDLE	V2ouest_bcl
11	0	0	0	DEPL_CST_STATE_IDLE	bcl_V2ouest
12	1	0	0	DEPL_CST_STATE_WAIT_FIN0	V2estvu_bcl
13	1	0	0	DEPL_CST_STATE_RESERVATION	bcl_V2estvu
14	0	0	0	DEPL_CST_STATE_IDLE	
15	0	0	0	DEPL_CST_STATE_IDLE	

- Les cellules de la colonne start déclenchent les déplacements
- Celles de la colonne stop les désactivent.
- La colonne Tempo décompte lorsque les temporisations taf et tsf sont utilisées
- La colonne State indique l'état du déplacement. Sur la capture, la plupart des déplacements sont inactifs. Le déplacement 12 avec sa bande verte attend la condition de fin (souvent, l'arrivée de la locomotive sur le dernier canton). Le déplacement 13 avec sa bande jaune attend car il à besoin de cantons et d'aiguillages réservés (par le déplacement 12).
- Enfin, la colonne name, affiche le nom du déplacement.

Variables des déplacements

- Les variables `depl_start_run_0..99` permettent de déclencher les déplacements et indiquent s'ils sont actifs. Ces variables une fois mises à 1 pour le déclenchement, le reste jusqu'à la fin du déplacement, où elles sont automatiquement remises à 0.
- Les variables `depl_stop_0..99` permettent de stopper prématurément des déplacements.

4.13. Horloge

L'horloge gère le temps mais sert aussi à déclencher des actions comme des déplacements. L'écran horloge permet de régler l'horloge alors que la section horo pour horodateur spécifie les actions à effectuer en fonction de la date et de l'heure courante.

Section horo

```
#horo
[
  depart j:0 6:35 tic par min:16

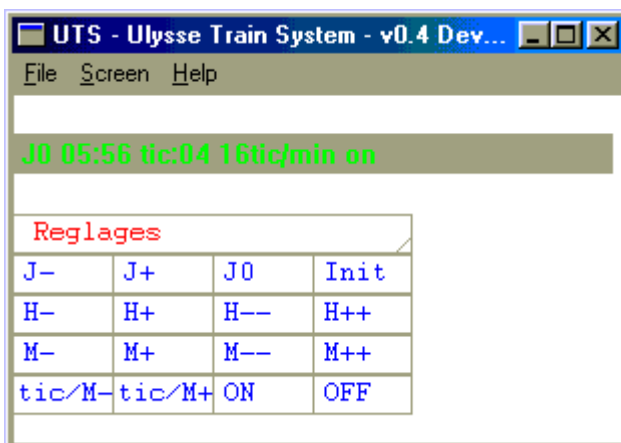
  hd 6:40 hb +0:04 vf:!depl_start_run_5 s:depl_start_run_5 s:exe 2
  hd 6:42 hb +0:03 vf:!depl_start_run_3 s:depl_start_run_3
  hd 6:45 hb +0:10 vf:!depl_start_run_2 s:depl_start_run_2
  hd 7:10 hb +0:04 vf:!depl_start_run_1 s:depl_start_run_1
  hd 7:15 hb +0:05 vf:!depl_start_run_0 s:depl_start_run_0
  hd 7:16 hb +0:05 vf:!depl_start_run_4 s:depl_start_run_4

  hd 8:00 hb +0:05 vf:!depl_start_run_6 s:depl_start_run_6
  hd 8:05 hb +0:05 vf:!depl_start_run_7 s:depl_start_run_7
  hd 8:10 hb +0:05 vf:!depl_start_run_8 s:depl_start_run_8
  hd 8:15 hb +0:05 vf:!depl_start_run_4 s:depl_start_run_4
  hd 8:20 hb +0:05 vf:!depl_start_run_5 s:depl_start_run_5
  hd 8:25 hb +0:05 vf:!depl_start_run_10 s:depl_start_run_10
  hd 8:30 hb +0:05 vf:!depl_start_run_8 s:depl_start_run_8
  hd 8:35 hb +0:05 vf:!depl_start_run_9 s:depl_start_run_9
]
```

- La directive départ indique la date et l'heure de simulation pour le réseau au lancement d'UTS. Elle spécifie également la durée d'une minute de simulation. Pour cet exemple une minute de simulation dure 16 tics de 250 ms soit 4s réelles ou encore, une heure de simulation dure $60*4=240s=4$ minutes réelles ou encore une journée de simulation de 24h dure pendant $4*24=96$ minutes=1h et 36 minutes...
- La directive hd pour heure de départ, définit le moment pour effectuer les actions, c'est à dire, mettre les variables qui ont une directive s pour set à 1 et celles qui ont une directive c pour clear à 0.
- Pour des actions répétitives, il est possible de spécifier un temps intermédiaire hi, qui fixe la récurrence des actions. Il est également possible de définir une heure de fin hf où cette récurrence doit cesser. Par exemple `hd 6:00 hf 16:00 hi 4:00`
- L'horloge peut également être bloquée si après le temps spécifié par + suivant un déclenchement, la variable indiquée par vf n'est pas à 1 (ou à 0 si ! est utilisé). Ce mécanisme est très utile pour les déplacements, où il permet de bloquer l'horloge, si le déplacement ne se termine pas dans les temps (retard, blocage ...). Avec cette méthode il est possible d'assurer toutes les correspondances et de faire l'heure quoi qu'il se passe.

Cet exemple permet de séquencer des déplacements. Il met à 1 les variables de déclenchement des déplacements et bloque l'horloge s'ils ne se terminent pas dans les temps spécifiés. Mais l'horloge peut également servir à déclencher toute autre action comme éteindre la lumière de la pièce et allumer celles de la gare entre 21h00 et 07h00 ...

Ecran de l'horloge



- Les boutons, J, H, M et tic/M permettent d'ajuster respectivement le jour courant, l'heure, les minutes et les tics par minutes réelles.
- Le bouton init initialise l'horloge avec la date et l'heure spécifiées dans la section horo, ou à J0 5 :00 si cette information n'est pas présente dans la section horo.

4.14. Tables de vérité et équations

Les tables de vérités et équations permettent de mettre des relations entre variables très simplement. L'utilisation la plus évidente est celle de la signalisation pour laquelle il peut y avoir une relation directe entre les leds d'un feu et les détections sur les cantons. La section tdv permet d'entrer les tables de vérité et équations.

Section tdv

```
#tdv
[
  tdv:
  init: set led cli 11
        set led cli 22
        set led phase 22

  tdv:
  on: 1
  in: det 9 | out: led 11 led 22
  eq: 0 | 0 0
  eq: 1 | 1 1

  @ou
  @1 det 9 = led 11 = led 22

  tdv:
  in: det 6 det 7 pos0 aig 1 pos0 aig 0 | out: led 28 led 29 led 30 led 31 @ r rh j v
  eq: 1 x x x | 1 1 0 0
  eq: x x 0 x | 1 1 0 0
  eq: x x x 0 | 1 1 0 0
  eq: 0 1 1 1 | 0 0 1 0
  eq: 0 0 1 1 | 0 0 0 1

  @pulse0:12-16-2-8-20 out_22
  @pulse1:4-16-2-8-20 out_20
  @pulse3:4-8-16-8-20 out_22 out_23
  @pulse31:4-8-16-8-20 out_22

  @equ: 1 a o x = ln an on xn =set =clr det_0
  @equ: ln det_1 a det_2 o det_3 x det_4 an det_5 =set out_11 =clr out_12
]
```

Les tables de vérités commencent par tdv

- on :1 indique que la table est active
 - in : identifie les variables en entrée
 - init : initialise les sorties avant le traitement (ici, on les mets à 0). clear = mettre à 0, set = mettre à 1.
 - Les équations possèdent des conditions sur les entrées qui lorsqu'elles sont est vraies, déclenchent les actions.
- Par exemple la première équation allume les leds si un train est détecté sur le canton du passage à niveau. Afin de les faire clignoter en alternance, la directive init est utilisée. Il est à noter que la table de vérité (sauf init), peut être remplacée par l'équation en commentaire.

La seconde table de vérité permet de gérer un feu (voir sa position sur le TCO). Le canton qu'il protège est accessible par un aiguillage triple. C'est la raison pour laquelle les leds du feu dépendent des détections sur le canton protégé et celui qui lui succède, ainsi que la position de cet aiguillage.

```
o rouge(io out 29)
o vert (io out 31)
o rouge(io_out_28)
o jaune(io_out_30)
| feu visible du canton 1

aig0 ====canton0==== det0
====canton7==== X ====canton6==== X =feu====canton1==== det1
det_7 det_6 aig1 ====canton2==== det2
```

Les équations sont un peu plus difficiles à comprendre. Elles reposent sur une variable de test dont la valeur (ou l'état) peut évoluer en fonction d'autres variables et opérations. Son état peut également être assigné à une variable.

- l <var> (pour load) charge test avec l'état de la variable
- a <var> (pour and) effectue un ET logique entre test et var. Le résultat est à 1 si var et test sont à 1
- o <var> (pour or) effectue un OU logique entre test et var. Le résultat est à 1 si var ou test (ou les 2) sont à 1
- x <var> (pour xor) effectue un OU EXCLUSIF entre test et var. Le résultat est à 1 si var ou test (mais pas les 2) sont à 1
- ln, an, on, xn. L'état complémentaire de la variable est utilisé (1 det_0 an det_1 = out_3). La sorties est à 1 s'il y a une locomotive sur le canton 0 mais pas sur le 3
- = <var> affecte test à la variable
- =set <var> met la variable à 1 si état est à 1. Dans le cas contraire, rien ne se passe (contrairement au cas précédent = où la variable aurait été mise à 0 si test était à 0)
- =clr <var> met la variable à 0 si état est à 1. Dans le cas contraire, rien ne se passe.
- = <var>=<val> met var à val si état est à un.

Pour les opérateurs l,a,o,x il est possible de remplacer <var> par <var>[==/</>/</>=<=]<val>. Cela peut servir à faire des tests sur les variables qui peuvent prendre plus de 2 états. Par exemple, si l'on souhaite allumer une lampe lorsque la lumière de la pièce est faible. Dans ce cas, on connecte une LDR à une entrée analogique (ex la 3). Si on considère que la lumière est faible lorsque la valeur mesurée passe en dessous 99, et que les lampes du réseaux sont connectées sur out_4 et out_5 : `1 can_3<99 = out_4 = out_5`

UTS n'utilise pas les priorités, ce qui veut dire que l'équation est traitée de la gauche vers la droite. Les équations utilisent la logique de BOOL, c'est pourquoi ils s'appellent opérateurs booléens. La logique booléenne est à la base de tous les ordinateurs et pourra vous rendre de précieux services.

La section tdv permet également de définir des impulsions qui se commandent par les variables pulse_cmd_<0-31>. Ex : `pulse0:12-16-2-8-20 out_22`. Cela veut dire que lorsque pulse_cmd_0 est à 1, alors elle est remise à 0 puis la sortie 22 est mise à 0 pendant 12tics (3s), puis à 1 durant 16 tics puis à 0 durant 2 tics puis à 1 durant 8 tics puis à 0 durant 20tics. Si une seule impulsion est nécessaire, il suffit de mettre la seconde pulse à 0. Ces impulsions un peu bizarres ont été conçues pour piloter des dételeurs. Mon réseau est en N et souvent un dételeur haut peut bloquer une locomotive, donc la première impulsion bloque la locomotive, puis la relâche puis remonte pour dételeur le convoi. Mais je n'ai jamais trouvé un dételeur fiable en N. (si vous avez des infos, je suis preneur). Dommage car j'aurais bien aimé voir un retournement en automatique... La dernière durée sert à laisser reposer le dételeur car un solénoïde de dételeur à besoin de refroidir. Pour un fonctionnement manuel, vous pouvez directement utiliser un bouton poussoir sur le TCO. Bien entendu les impulsions peuvent servir à bien d'autres choses.

4.15. Vibreur

Le vibreur permet de faire vibrer le réseau lorsqu'une perte de contact est détectée. Il faut bien entendu relier un moteur avec une masse excentrée à une sortie pour produire les vibrations. Construire un système soit même permet ce type d'excentricité.

Section vib

La section vib permet de spécifier la sortie à activer en cas de mauvais captage et d'initialiser quelques paramètres.

```
#vib
[
  on
  time before:5
  time after:7
  @out_pwm:8,32
  @out:out_21
]
```

- La directive `on` active le vibreur au démarrage. (Il peut être désactivé par la suite avec l'écran du vibreur ou avec `off`).
- `Time_before` indique la durée minimal du problème de détection nécessaire afin d'éviter les déclenchements intempestifs.
- `Time_after` indique le temps durant lequel le vibreur doit continuer à fonctionner après le contacte rétablit. L'exemple utilise 7 tics, soit $7*0.250=1.75s$.
- `out` permet d'indiquer la sortie à activer si vous utiliser une sortie.
- Il est aussi possible de remplacer la sortie (ou la compléter) en utilisant une sortie pwm de canton. Dans ce cas il faut utiliser la directive `out_pwm` suivit du numéro de canton et de la vitesse. Pour info, si une sortie pwm est utilisée alors l'écran RS232 ne pourra pas modifier cette sortie.

Variables du vibreur

La variable `vib_on` permet d'activer ou de désactiver le vibreur.

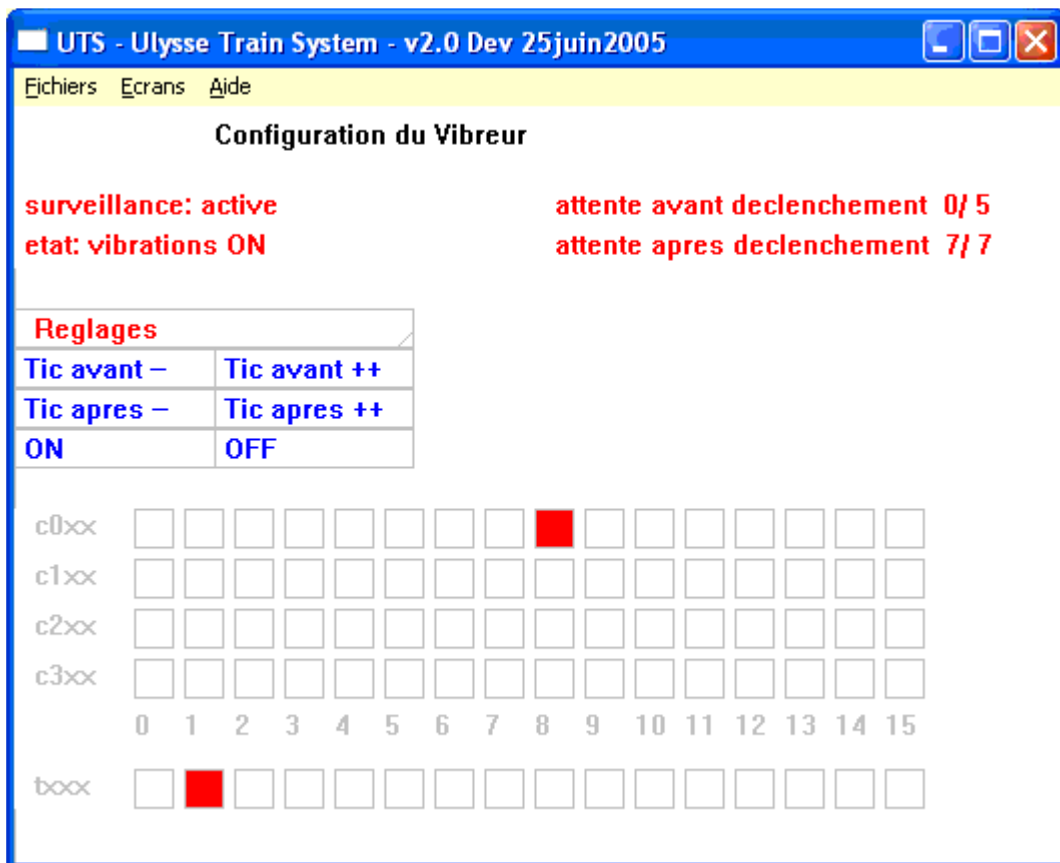
La variable `vib_state` indique l'état du vibreur

Il est également possible de faire vibrer seulement la partie du réseau incriminé en utilisant les variables `pb_capt_c t`
Par exemple pour une section comprenant les cantons 8, 10,13 et un vibreur sur la sortie 14 :

```
1 pb_capt_c_8 o pb_capt_c_10 o pb_capt_c_13 a vib_state = out_14
```

Ecran du vibreur

L'écran vibrateur permet de modifier les paramètres du vibreur pendant l'utilisation. Il indique également l'état du vibreur. La capture suivante indique un problème de captage du train 1 sur le canton 8. Lorsqu'un problème de détection survient, le symbole canton du canton incriminé doit clignoter.



4.16. Télécommande

Il est possible de commander les 5 potentiomètres virtuels (normalement commandés au clavier) par une télécommande de télévision. Elle peut également commander les aiguillages. Bien entendu, l'utilisation de la télécommande est pratique, mais pas indispensable. Sous Windows, tous les programmes peuvent recevoir des messages de Windows lui-même (ex click souris, déplacement fenêtre ...) ou d'autres programmes. UTS accepte donc des messages pour les claviers virtuels et aiguillages. Le soft « girder » associé à un capteur infra rouge simple à réaliser permet d'envoyer de tels messages à un programme. Malheureusement, ce soft gratuit vient juste de devenir payant. Je vous conseil donc juste de chercher sur Internet une ancienne version gratuite.

Ensuite il faut juste envoyer les messages suivants à UTS : du type :

```
MSG_TYPE : Command (WM_COMMAND)
wParam   : 999
lParam   : numéro spécifié
```

aiguillages:

```
20      aig inverse (si non specifie aig change)
21      aig direct  (si non specifie aig change)
0-15    selection aiguillage si cfg_aig_digit=1
0-9 puis 0-9 selection aiguillage si cfg_aig_digit=2
```

trains:

```
50      Tous les pot virtuels a 0
30-34   selection pot virtuel 0 a 4
40      stop (step=0)
41      step++ (max=3)
42      step-- (min=-3)
```

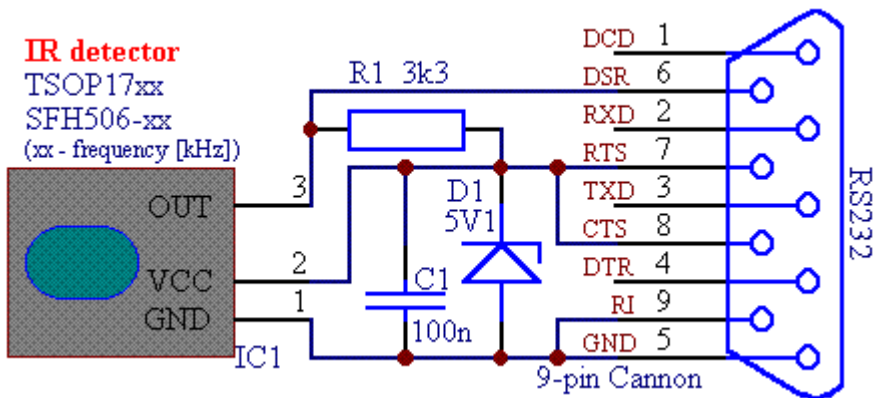
Pour les trains, il suffit de sélectionner le train (en fait un des potentiomètres virtuels qu'il est possible d'attribuer à un train avec l'écran train). Une fois le train sélectionné, utiliser une touche pour arrêter, une autre pour passer d'un cran dans un sens et une autre pour l'autre sens.

Pour les aiguillages, il faut d'abord spécifier si les aiguillages ont besoins d'une ou 2 touches à l'aide de la directive `ir_aig_digit` dans la section général. Pour 16 aiguillages ou moins une touche suffit. Pour plus (jusqu'à 64), il faut utiliser 2 touches. Si vous avez entre 10 et 16 aiguillages, vous pouvez choisir une des 2 méthodes.

- Pour une touche : Il suffit d'appuyer sur la touche correspondante pour que l'aiguillage change de position. (Cet appuie peut être précédé de la touche position directe ou position inverse si nécessaire).
- Pour 2 touches, utilisez la touche de position directe ou inverse puis composer le numéro de l'aiguillage sur 2 chiffres ?

```
#general
[
...
@ nombre de chiffre nécessaire pour commander les aiguillages 1 ou 2
ir_aig_digit 1
]
```

Pour plus de renseignements sur le montage du capteur IR et « girder », voir l'excellent site d' Igor Cesko <http://www.cesko.host.sk/girderplugin.htm> ... Le schéma ci-dessous montre la simplicité de réalisation du capteur. Il est préférable de rajouter un condensateur de 10uF en complément de celui de 100nF.



Vous savez maintenant tout sur le logiciel UTS !

5. Conclusion

Vous connaissez maintenant parfaitement le système UTS. Vous pouvez donc le réaliser, vous en inspirer ... J'espère que cette documentation aura été agréable à lire et vous aura apporté des connaissances. N'hésitez pas pour toute question. Bien entendu, une photo de votre réalisation me fera le plus grand plaisir ...

Bonne utilisation !

Vous pouvez télécharger l'archive zip sur mon site Internet qui contient :

- Le programme à charger dans le PIC de la carte 0 (873rs13_0.hex)
- Le programme à charger dans le PIC de la carte 1 (873rs13_1.hex)
- Le programme à charger dans le PIC de la carte 2 (873rs13_2.hex)
- Le programme à charger dans le PIC de la carte 3 (873rs13_3.hex)
- L'exécutable (uts0200.exe)
- Un exemple de fichier de configuration vide mais avec l'aide (config_vide.txt)
- Un exemple de fichier de configuration d'exemple (celui de mon réseau) (config_ex.txt)
- Un exemple de fichier de configuration de TCO d'exemple (celui de mon réseau) (config_tco_ex.txt)
- Les sources du programme du PIC (873rs13.asm)
- Les sources du logiciel coté PC (main.c, main.h, rsrc.rc, uts0200.dev)

Auteur : Ulysse Delmas-Begue

Dernière mise à jour de cette doc : 6 juillet 2005

Version correspondance de UTS : 2.0

Email : udelmas@chez.com (ou ulyssedelmas@laposte.net si la première ne fonctionne pas bien (pas de réponses ...))

SiteWeb : <http://www.chez.com/udelmas>

Notes :