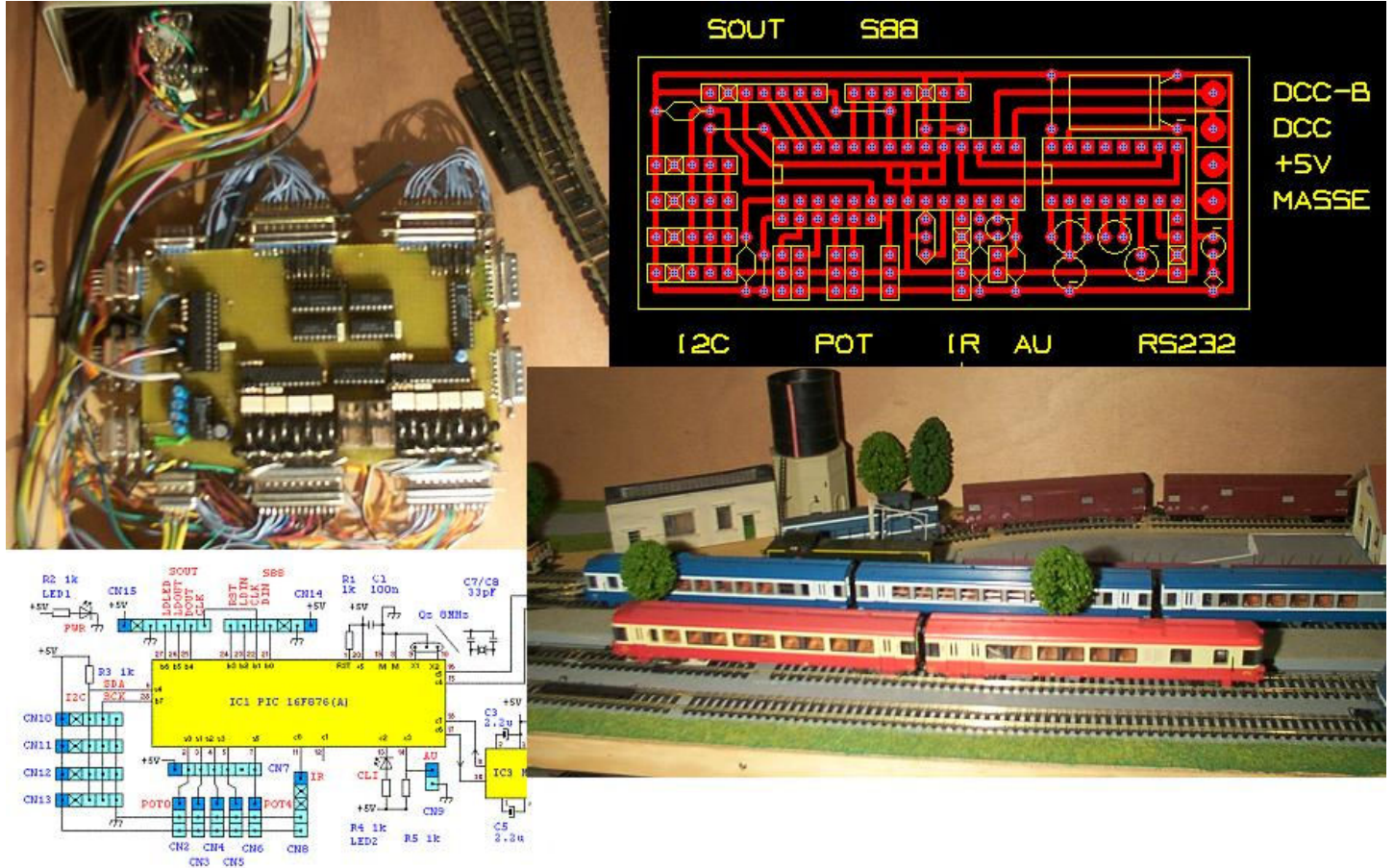


Version préliminaire ! Version préliminaire ! Version préliminaire ! Version préliminaire ! Version préliminaire

Free-DCC est un système numérique DCC de pilotage de réseau ferroviaire miniature à réaliser soit même. Il est complet, performant, compatible, modulable à souhait et peu coûteux.



## Table des matières

1. Introduction
2. Le DCC
3. Présentation de Free-DCC
4. Electronique
5. Utilisation
6. Exemple
7. Conclusion

## Historique

22 août 2007: Première Mise à disposition de la version préliminaire.

11 juin 2007: Création de la documentation correspondant a la version 0.6

Septembre 2006: Début du projet free-dcc

Free-DCC est une centrale numérique à réaliser soi-même pour contrôler un réseau de train miniature digitalisé basé sur la norme DCC. Le fait de réaliser sa propre centrale permet de comprendre son fonctionnement et réduire drastiquement le prix de revient. Bien que ce système soit une réalisation personnelle, il est performant et extrêmement modulable ce qui permettra sans aucun doute de l'adapter à vos besoins.

J'ai réalisé ce projet pour plusieurs raisons. La première étant d'avoir la connaissance profonde du système afin de l'adapter à mes besoins, le dépanner en cas de besoin, le faire évoluer ... Cela m'a également permis de réduire les coûts, de faire tenir toutes l'électronique du réseau sur une seule carte. Mais le principal avantage est d'intégrer toutes sortes de fonctionnalités. Souhaitant partager cette réalisation avec le plus grand nombre de passionnés, le système est entièrement gratuit et documenté. D'ailleurs, son nom « Free-DCC » signifie gratuit ou plutôt libre à la mode des logiciels libres GNU. J'ai choisi un nom Anglais ou plutôt Américain car c'est à la mode dans le domaine de l'informatique.

Le choix du numérique, à la place du traditionnel analogique offre une plus grande souplesse d'utilisation ainsi que de nouvelles fonctionnalités. En effet au prix de l'installation d'un décodeur dans les locomotives il est maintenant possible de conduire indépendamment ces machines sur la même voie, commander les feux et même récemment l'attelage ! Le numérique révolutionne le train miniature et il n'était pas question de s'en passer pour une nouvelle réalisation. Plusieurs systèmes numériques existent à ce jour, certains anciens, d'autres propriétaires, mais le plus intéressant est le standard DCC imaginé par les modélistes américains. Un chapitre dédié au DCC permet de comprendre exactement ce qu'est ce système, ce qu'il est possible de faire avec, courant, utile, nécessaire, superflu ...

Après le chapitre DCC, un chapitre présente Free-DCC vous permettant de vous faire une idée précise de ce système. Vous verrez ainsi que Free-DCC permet de piloter des trains en manuel avec une multitude de souris, en automatiques, contrôler les aiguillages, les itinéraires, gérer la signalisation. Ce système est bien adapté aux petits et moyens réseaux. Le fait qu'il soit modulable vous permet de l'adapter à vos besoins et me permettra de le faire évoluer par la suite.

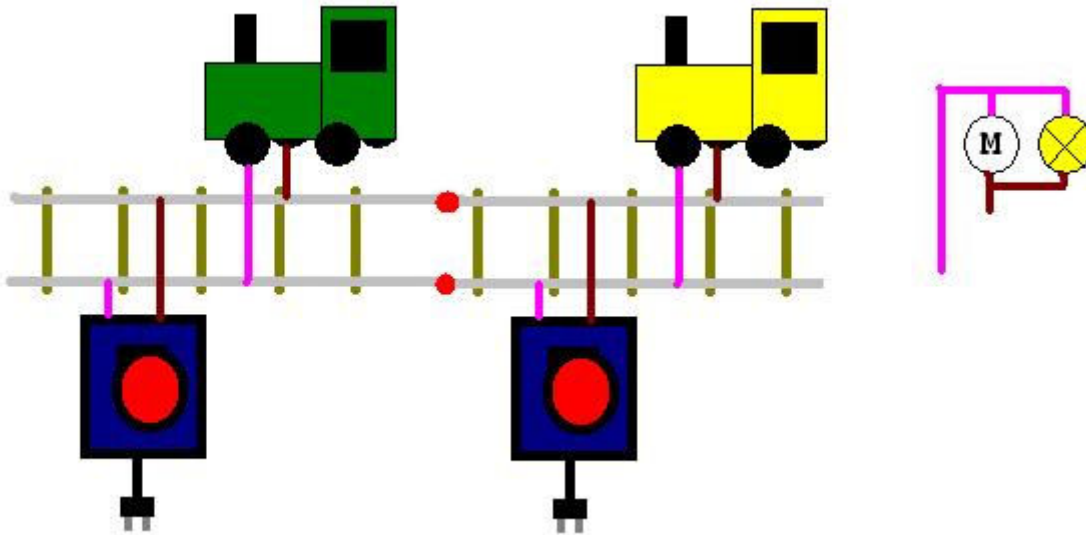
Une fois que vous comprendrez exactement ce que Free-DCC peut faire, un autre chapitre décrit la réalisation électronique de la centrale ainsi que de tout son entourage comme la commande des aiguillages, la signalisation, la détection des trains ... Vous pourrez même mixer ces montages avec des produits du commerce ou les regrouper sur une seule carte afin d'avoir une réalisation extrêmement compacte. Le chapitre suivant est le manuel de Free-DCC qui décrit l'utilisation de la centrale. Aussi bien en mode autonome qu'en mode piloté par un ordinateur. En effet elle peut s'interface avec des logiciels du commerce ou bientôt avec mon propre logiciel UTS. Enfin un dernier chapitre donne l'exemple d'un de mes modules afin de travailler sur un cas concret !

Juridiquement ce système est gratuit pour les particuliers ou clubs et ne peut pas être utilisé pour des produits commerciaux sans mon consentement. Il a été réalisé avec des logiciels libres de droits comme les compilateurs, éditeurs ... Etant gratuit, je ne peux être tenu pour responsable en cas de problèmes. Je ne suis pas tenu non plus d'assurer du support. Ceci dit, j'ai mis le plus grand soin dans cette réalisation et je me ferai bien entendu un plaisir de vous aider en cas de besoin. Ayant un réseau HO, ce système a été testé, mûrement réfléchi et va vivre de nombreuses années. N'hésitez pas à commenter, m'envoyer des photos, des remarques, des suggestions, des critiques constructives ... Je ne compare pas volontairement mon système à ceux du commerce, à vous de le faire, on y trouve le meilleur comme le pire

Vous pouvez utiliser ce document pour réaliser Free-DCC mais également pour en apprendre un peu plus sur le DCC ou l'électronique. Sur ce, bonne lecture

## A. La norme

### A.1. Commande directe et digital

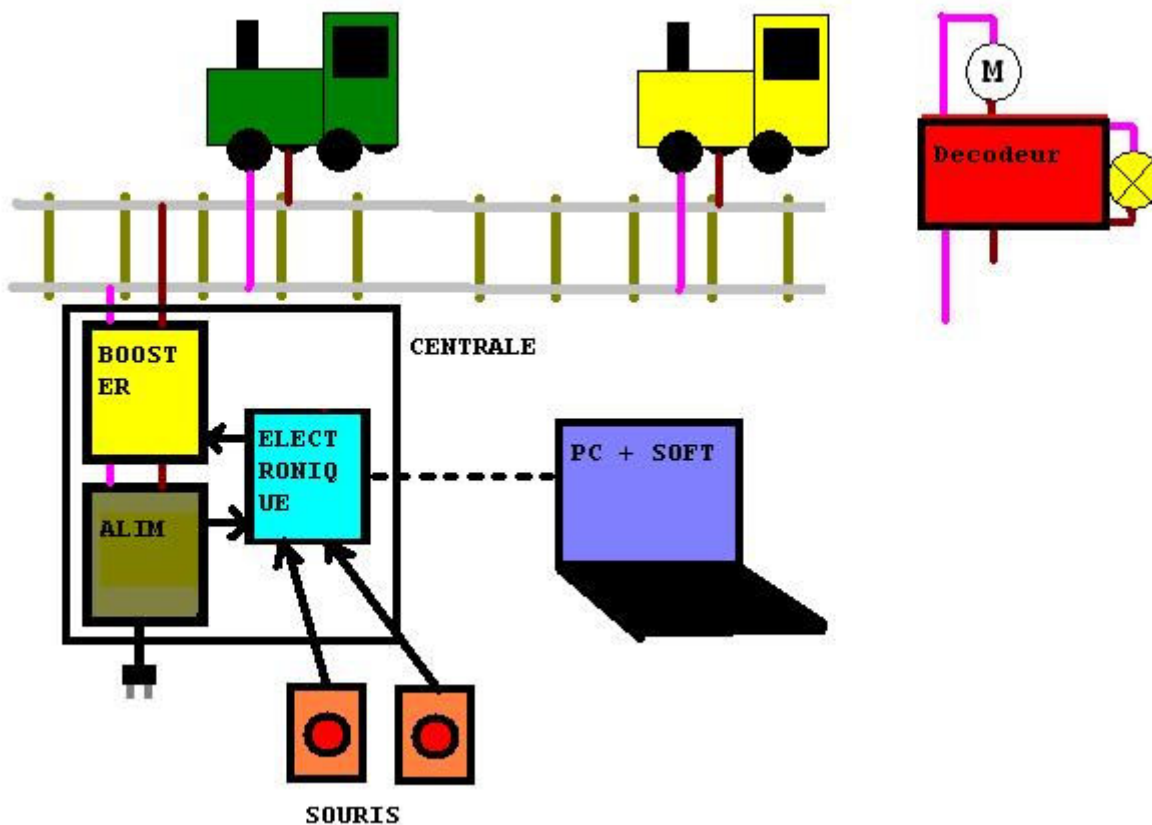


La commande analogique ou directe des trains électriques est très simple. Un transformateur alimente le moteur d'une locomotive directement par la voie. Donc si plusieurs locomotives sont sur la même voie, alors elles feront les mêmes mouvements. Pour résoudre ce problème, les réseaux peuvent être divisés en cantons. Chaque canton est alimenté par un transformateur ou des interrupteurs qui permettent de sélectionner un transformateur pour chaque canton. Il est possible avec des systèmes électroniques plus ou moins complexes de connecter automatiquement les transformateurs aux cantons de sorte que les transformateurs changent de canton avec les locomotives. Ces systèmes sont connus sous le nom de conduite sélective.

On en déduit donc que 2 locomotives ne peuvent pas évoluer de façon indépendante sur le même canton. Une autre contrainte concerne l'éclairage embarqué qui dépend de la tension du canton imposée par la commande du moteur. Pour résoudre ce problème les rames peuvent être éclairées par un système haute fréquence mais c'est un peu compliqué. Les phares ont aussi ce problème d'intensité et ils sont éteints à l'arrêt.

Bref la commande directe est simple, il est possible de l'améliorer mais elle pose de nombreuses limitations. Mais son principal avantage est qu'elle ne nécessite pas la modification du matériel roulant.

Pour résoudre ces limitations, l'idée est de mettre des décodeurs dans les locomotives. Il est donc nécessaire de transmettre aux décodeurs de la puissance et des commandes. La puissance doit nécessairement passer par la voie. Généralement une tension de valeur efficace constante est utilisée. Les commandes peuvent passer par la voie (par superposition HF ou inversion de polarité) ou non. (Ondes radio, infra rouge, câble rayonnant...). Hormis la transmission radio pour les grosses échelles, la transmission des commandes par la voie par inversion de polarité s'est imposée pour sa simplicité de mise en œuvre. De plus elle ne nécessite pas d'analogique ce qui permet un gain de place sur les décodeurs et l'absence de réglages.



Ce principe consiste à mettre sur la voie +15V ou -15V (par exemple). Les durées de ces états permettent de transmettre des informations élémentaires connus sous le nom de bits. (Les fameux 0 et 1 en informatique). Plusieurs bits forment une trame qui est en fait une commande. Au début du numérique chacun avait son protocole (bits et trames) ce qui interdisait toute compatibilité. Ensuite le protocole Motorola a percé à un moment. Puis la NMRA, l'association des modélistes ferroviaire américain, décida de spécifier un standard qu'ils nommèrent DCC. Depuis presque tout le monde utilise ce standard ce qui est très bien pour la compatibilité. Je vous encourage à télécharger ce standard sur le site de la NMRA <http://www.nmra.org>.

Un système DCC se compose de décodeurs (un dans chaque locomotive) et d'une centrale.

La centrale dispose d'une alimentation, (un transformateur) dont la tension est inversée ou non par un booster qui est commandé par une électronique qui lui fournit les trames. Les trames sont formées en fonction des consignes des souris ou de celles d'un système informatique.

Pour différencier les locomotives, chaque décodeur est programmé avec un numéro unique que l'on appelle adresse. Par exemple la première souris pourra contrôler la locomotive d'adresse 25... Les décodeurs redressent la tension +15/-15, ce qui leur permet de disposer d'une tension constante pour l'électronique et pour le moteur. Les décodeurs permettent bien entendu de contrôler le moteur en sens et vitesse mais aussi les feux et des sorties appelées fonctions spéciales dont l'utilisateur pourra se servir à sa convenance (éclairage cabine, dételeur, pantographes, fanions, allumer chaque phare indépendamment ...)

Le digital procure donc des avantages certains par rapport à la commande analogique. Mais son plus gros problème est de nécessiter l'installation d'un décodeur dans chaque locomotive. De plus l'apparente simplicité de connexion que vantent les publicités (juste 2 fils entre la centrale et les voies) s'évapore vite lorsque l'on réalise un réseau un peu sérieux. En effet, tout bon modéliste à besoin de découper son réseau en cantons comme en réalité et poser des détecteurs d'occupation pour gérer la signalisation, visualiser les occupations sur un TCO ...

## A.2. Etude de la norme DCC

### A.2.1. Les bases

Je vous conseille de télécharger la norme DCC sur le site de la NMRA <http://www.nmra.com> .

Je sais, elle est en Anglais, mais les figures parlent d'elles mêmes ... Elle décrit les éléments suivants :

**\*\* Un bit 1** est composé d'une tension dans un sens durant 52 à 64 us (microseconde) (Il y a un million de us dans une seconde) puis d'une tension inverse (donc dans l'autre sens durant le même temps). Cette répétition simplifie les décodeurs car les locomotives peuvent être placés dans les 2 sens.

**\*\* Un bit 0** est composé d'une tension dans un sens durant 90 à 10000 us. Puis d'une tension inverse durant 90 à 10000 us. Ces 2 parties n'ont pas besoin de durer le même temps pour un 0, alors que c'est impératif pour un 1.

**\*\* Une trame** est composée de bits

- Un minimum de 10 bits a 1 pour indiquer le début d'une trame
- Un 0 de séparation
- Une adresse pour sélectionner un décodeur (8 bits)
- Un 0 de séparation
- Une commande (1x8bits ou 2x8bits ...) (un 0 est ajouté entre chaque groupe de 8 bits)
- Un 0 de séparation
- Une vérification (8bits)
- Un 1 final

Par exemple avec un octet de commande (1 octet = 8 bits) :

11111111 0 AAAAAAAAA 0 CCCCCCCC 0 EEEEEEEE 1

Par exemple avec deux octets de commandes :

11111111 0 AAAAAAAAA 0 CCCCCCCC 0 CCCCCCCC 0 EEEEEEEE 1

Les A, C et E peuvent être à 0 ou 1.

Le nombre d'octets de commande dépend de certains bits C du premier octet.

Les bits de vérifications se calculent en faisant un OU EXCLUSIF entre chaque bit A et C. En clair ils valent 1 s'il y a un nombre impair de bits a 1.

Par exemple avec 1 octet de commande

Si AAAAAAAAA = 00001111 et CCCCCCCC = 01000111

Alors EEEEEEEE = 01001000

C'est grâce à ce mécanisme que le décodeur sait ou non si la trame transmise est correcte. Si oui, il effectue l'action demandée. Sinon, il ne fait rien. Essayez de modifier 1 bit A ou C, et vous verrez que le E calculé ne colle pas.

On voit également que grâce aux 0 entre les octets, il n'est pas possible d'avoir plus de 8 bits à 1 hors du préambule. Donc la détection du début de trame est aisée.

Les 0 et 1 sont bien pour des ordres à 2 positions (ex : phares activés ou pas), mais on a besoin de transmettre des nombres comme l'adresse du décodeur ou encore la vitesse. Pour ce faire on utilise le principe suivant :

Avec 1 bit il est possible de coder 2 valeurs (0 ou 1)

Avec 2 bits il est possible de coder 4 valeurs (00=0, 01=1, 10=2, 11=3)

Avec 3 bits il est possible de coder 8 valeurs (000=0, 001=1, 010=2, 011=3, 100=4, ...111=7)

Avec 4 bits il est possible de coder 16 valeurs (0 à 15)

Avec 5 bits il est possible de coder 32 valeurs (0 à 31)

Avec 6 bits il est possible de coder 64 valeurs (0 à 63)

Avec 7 bits il est possible de coder 128 valeurs (0 à 127)

Avec 8 bits il est possible de coder 256 valeurs (0 à 255)

...

Pour cela on donne des poids aux bits. Le plus à droite à le poids le plus faible 1 (sa contribution est de 0 lorsqu'il est à 0 et de 1 lorsqu'il est à 1). Ce bit est connu sous le nom de LSB (Less Significant Bit=Bit le moins significatif). Le bit immédiatement à gauche du LSB a un poids de 2, le précédent de 4, le précédent de 8 ... jusqu'au MSB (Most significant Bit= Bit le plus significatif car il a le plus gros poids).  
Par exemple  $10011011 = 1*128+0*64+0*32+1*16+1*8+0*4+1*2+1*1=155$

Les trames sont transmises en permanence car à chaque perte de contact électrique, les décodeurs doivent recevoir les ordres au plus vite afin que l'effet de la perte de contact passe inaperçu. Par exemple lorsque la locomotive passe sur un rail encrassé, il n'y a plus de contact électrique, donc le décodeur n'est plus alimenté et perd donc ses ordres. L'inertie permet de dépasser la saleté, ce qui réalimente le décodeur, mais celui-ci ne possède plus l'ordre de vitesse qui doit donc arriver au plus vite pour passer inaperçu. Je considère qu'il est nécessaire d'envoyer un ordre sens-vitesse environ 4 fois par seconde. De toute façon, en dessous, la réactivité est plutôt mauvaise.

Comme les 0 durent plus longtemps que les 1, une trame à 1 octet de commande dure le plus longtemps si :  $1111111111\ 0\ 00000000\ 0\ 00000000\ 0\ 00000000\ 1 = 11*1\ \text{et}\ 27*0$ . Dans mon cas la centrale utilise  $2*58\mu\text{s}$  pour les 1 et  $2*120\mu\text{s}$  pour les 0 et 10 bits de préambule, donc une trame a 1 octet dure au maximum :  $11*2*58+27*2*120=7756\mu\text{s}$ . Ce qui fait donc  $1000000\mu\text{s}/7756\mu\text{s}=128$  trames max par seconde et donc 32 locos max si on transmet 4 fois par seconde. Ma centrale limite ce nombre à 16 car elle ne transmet pas que l'ordre sens-vitesse mais aussi des ordres pour les fonctions spéciales. Vous remarquerez qu'il aurait été préférable de compléter les bits d'erreur mais les créateurs de la norme n'ont pas vu cette astuce qui aurait permis de gagner du temps.

La norme prévoyait à l'origine 128 adresses pour les locomotives, mais vous constatez déjà que ce nombre est irréalisable, sans parler de la consommation électrique. 128 décodeurs à l'arrêt mettraient déjà à genou le booster. (ex :  $20\text{mA} \text{ par exemple} * 128 = 2.56\text{A}$  à comparer aux booster de 3A couramment utilisés ...). La norme à quelques incohérences comme celle là. Alors que dire de l'extension à 1024 adresses ...

Les adresses sont codées sur 8 bits donc (0 à 255) et j'indique que la norme en prévoit 128 ! Ce n'est pas une erreur car les autres adresses sont attribuées à d'autres décodeurs comme les décodeurs fixes d'accessoires utilisés notamment pour la commande des aiguillages. Je n'utilise pas ce type de décodeur car le protocole est assez mal fait, en effet il faut envoyer une trame pour changer l'état d'une sortie, ce qui se révèle vite ingérable.

A noter que l'adresse 0 permet d'envoyer un ordre à tous les décodeurs. Ceci peut être utile pour un arrêt d'urgence généralisé (quoi qu'une coupure de courant fasse la même chose). La norme ne prévoit donc non pas 128 mais 127 locomotives.

A propos d'adresse spéciale la 255 (les 8 bits à 1) n'est destinée à aucun décodeur. On dit que c'est une trame de bourrage. Pour ma part, je les utilise s'il y a moins de 16 locomotives.

Vous en savez maintenant assez pour regarder dans le détail les différentes trames

### A.2.2. La trame de sens-vitesse

La norme actuelle permet d'utiliser des décodeurs à 14 ou 28 crans de vitesse. Il est en outre possible d'utiliser en complément une commande à 127 pas sur certains décodeurs. Si le décodeur supporte plus d'un mode alors il est possible de sélectionner le mode à utiliser par programmation.

La première norme prévoyait 14 vitesses et la trame se présentait de la façon suivante :

1111111111 0 0AAAAAAAA 0 01DFSSSS 0 EEEEEEEE 1

Les bits de l'octet de commande se définissent comme suit :

D : Direction (0 arrière, 1 avant)

F : Allumage ou extinction des feux (phares)

S : Pas de Vitesse à atteindre

0000 = 0 = arrêt  
0001 = 1 = arrêt immédiat  
0010 = 2 = pas de vitesse 1

...  
1111 = 15 = pas de vitesse 14

Les pas de vitesse sont définis individuellement par programmation ou alors calculés à partir de la vitesse min et max. La norme prévoit la possibilité d'utiliser des profils pour l'accélération et la décélération. Ainsi lorsqu'un cran de vitesse est demandé, la locomotive accélère jusqu'à l'atteindre. Idem pour la décélération. On comprend ainsi le stop immédiat, sorte d'arrêt d'urgence qui n'utilise pas le profil de décélération mais arrête immédiatement la locomotive.

L'évolution de la norme a permis de rajouter un nouveau mode possédant 28 crans en utilisant le bit des phares pour un cran intermédiaire entre les vitesses. La commande des phares passant alors dans le groupe 1 qui contient aussi 4 sorties auxiliaires. Ainsi l'octet de commande prend cette forme : 0-1-D-S0-S4-S3-S2-S1

00000 ou 10000 = arrêt  
00001 ou 10001 = arrêt immédiat  
00010 = cran 1  
10010 = cran 2

...  
11111 = cran 28

Enfin il est possible d'utiliser 127 crans sur les derniers décodeurs. Ces crans ne sont pas configurables individuellement mais utilisent la vitesse min pour le 1 et max pour 127. La trame comporte 2 octets de commande.

111111111 0 0AAAAAAAA 0 00111111 0 DSSSSSSS 0 EEEEEEEE 1  
SSSSSSS = 0000000 = 0 = arrêt  
SSSSSSS = 0000001 = 1 = cran 1 = vitesse min

...  
SSSSSSS = 0000000 = 0 = cran 127 = vitesse max

Cette trame est différente de la trame habituelle donc il n'y a pas besoin d'indiquer au décodeur d'utiliser ce mode à condition qu'il le supporte. Il est donc possible d'utiliser ce mode en complément des 2 autres. Mais je pense que ce mode n'a pas grand intérêt car 28 pas configurables suffisent amplement ! D'ailleurs peu de centrales utilisent ce mode.

### A.2.3. Les trames des fonctions spéciales (group 1 et 2)

Grâce au digital, il est possible d'activer des fonctions spéciales à utiliser pour ce que bon vous semble comme allumer les phares individuellement, la cabine, les ventilateurs... Il existe même un locotracteur du commerce qui permet d'atteler et déatteler à volonté. Enfin les nouveaux décodeurs sonores utilisent aussi ces fonctions pour simuler le démarrage du moteur, les avertisseurs... Il existe aussi des décodeurs de fonctions (sans moteur) qui proposent uniquement des sorties spéciales. Ils sont destinés à être placés dans les voitures ou remorques pour par exemple gérer l'éclairage...

La commande group 1 contrôle 4 fonctions spéciales F1 à F4, plus la commande des feux en mode 28 vitesses. La commande d'allumage des phares appelée FL ou F0 permet d'allumer ou pas les phares. Le décodeur activant 2 sorties suivant le sens. Mais rien ne vous empêche d'utiliser les autres sorties pour contrôler les feux individuellement...

Voici la trame du groupe 1  
111111111 0 0AAAAAAAA 0 100-FL-F4-F3-F2-F1 0 EEEEEEEE 1

Le groupe 2 propose quand à lui 4 sorties supplémentaires F5 à F8

111111111 0 0AAAAAAAA 0 1011-F8-F7-F6-F5 0 EEEEEEEE 1

Enfin notons une extension future qui permettra de gérer 13 sorties, mais personne ne l'utilise actuellement

11111111 0 0AAAAAAAA 0 110FFFFFF 0 FFFFFFFF 0 EEEEEEEE 1

## A.2.4. La trame de programmation

Afin de correspondre exactement aux souhaits des utilisateurs, les décodeurs sont configurables. Il suffit de programmer ce que l'on appelle des variables de configuration dites CV. Chaque variable est constitué d'un octet. Le CV le plus important est le 1 qui contient l'adresse du décodeur.

Les CV sont numérotés de 1 à 1024, mais rassurer vous, vous n'aurez pas 1024 paramètres à définir car seul un petit nombre est utilisé en réalité. Les fabricants de décodeurs se doivent d'implémenter les indispensables, ne sont pas obligés d'implémenter les optionnels et peuvent même en définir de nouveaux. C'est pourquoi, il est conseillé de bien lire la fiche technique du décodeur avant de le programmer. Bien entendu, les CV sont sauvegardés dans une mémoire EEPROM ou FLASH qui a la bonne idée de garder son contenu même lorsque le décodeur n'est pas alimenté.

Voici les principaux CV :

CV	Nom	Description
1	Adresse	1-127
2	Point de démarrage	0-255 (pour que la loco démarre au cran 1)
5	Vitesse max	0-255
3	Inertie à l'accélération	0-255
4	Inertie à la décélération	0-255
29	Configuration	

La norme n'a vraiment pas fait dans la simplicité quand à la programmation des CV, en effet on retrouve pas moins de 5 méthodes pour programmer les décodeurs, ce qui est complètement idiot car cela complique les décodeurs, la compatibilité en programmation centrale/décodeur et surtout ce n'était pas justifié !

Nous allons tout de suite abandonner la programmation POM « Programmation On Main track » (programmation sur voie principale) car elle ne permet pas de modifier l'adresse de la locomotive, ce qui vous en conviendrez est très embêtant !

Nous nous concentrerons donc sur les programmations POL « Programmation On Learning track » (programmation sur voie d'apprentissage / de programmation). Parmi ces méthodes, la programmation par page est la plus utilisée, c'est donc celle-ci que nous utiliserons. Les méthodes POL ont l'inconvénient de programmer toutes les locomotives présentes sur le réseau ! Pour palier ce PB, les centrales disposent souvent d'une sortie à connecter sur une voie de programmation où vous programmerez vos décodeurs. Une autre solution est d'utiliser un interrupteur qui permet d'isoler le réseau à l'exception d'une voie dévolue à la programmation lors des opérations de programmation mais gare à l'oublie !

La trame de programmation est la suivante:

```
111111111 0 011110RRR 0 DDDDDDDDD 0 EEEEEEEE 1
```

Les bits D forment l'octet à programmer tandis que les bits R indiquent le registre.

Comme vous avez bien suivi vous allez me dire que 3 bits autorisent 8 registres au maximum (R0 à R7) et non 1024. L'astuce vient du registre R5 qui contient le numéro de page. Il suffit donc de choisir sa page avec R5 (256 choix, 00000000=0 pour la page 1, 1 pour la page 2, 255 pour la page 256) puis de programmer les registres R0 à R3. Ainsi pour programmer le CV1 il faut sélectionner la première page (mettre 00000000 dans R5), puis écrire dans R0

CV1 P1/R0 (P1=0 dans R5)

CV2 P1/R1

CV3 P1/R2

CV4 P1/R3

CV5 P2/R0 (P2=1 dans R5)

...



Afin de ne pas programmer le décodeur par erreur les protections suivantes ont été rajoutées :

- Le préambule passe maintenant à 10 bits au minimum
- La même trame de programmation doit être reçue identiquement 5 fois au minimum
- Des trames de reset doivent être judicieusement interposées à des endroits précis

Voici par exemple la succession de trames utilisées par Free-DCC

```
table_dcc_prog
  goto    dcc_mp_idle
  goto    dcc_mp_idle
  goto    dcc_mp_idle
  goto    dcc_mp_idle
  goto    dcc_mp_reset
  goto    dcc_mp_reset
  goto    dcc_mp_reset
  goto    dcc_mp_page
  goto    dcc_mp_page
  goto    dcc_mp_page
  goto    dcc_mp_page
  goto    dcc_mp_page
  goto    dcc_mp_reset
  goto    dcc_mp_reset
  goto    dcc_mp_reset
  goto    dcc_mp_reset
  goto    dcc_mp_reset
  goto    dcc_mp_reset
  goto    dcc_mp_reset
  goto    dcc_mp_reset
  goto    dcc_mp_prog
  goto    dcc_mp_prog
  goto    dcc_mp_prog
  goto    dcc_mp_prog
  goto    dcc_mp_prog
  goto    dcc_mp_reset
  goto    dcc_mp_reset
  goto    dcc_mp_reset
  goto    dcc_mp_reset
  goto    dcc_mp_reset
  goto    dcc_mp_reset
  goto    dcc_mp_reset
  goto    dcc_mp_idle
  goto    dcc_mp_idle
  goto    dcc_mp_idle
  goto    dcc_mp_last_idle
```

Pour indiquer que la programmation s'est bien déroulée, le décodeur doit augmenter sa consommation de courant, par exemple en activant brièvement le moteur. Cette surconsommation peut alors être détectée par la centrale pour vérification, mais ce n'est pas obligation pour la centrale.

### A.2.5. Gestion des sorties

Maintenant que vous savez tout sur la commande des locomotives, voyons comment la norme à décider de piloter les accessoires que sont les aiguillages, la signalisation ...

Afin de contrôler des sorties comme des aiguillages ou feux, la norme prévoit de commander par le signal DCC des décodeurs d'accessoires de 8 sorties.

La trame de commande est la suivante :

11111111 0 10AAAAAA 0 1AADSSS 0 EEEEEEEE 1

AAAAAA-AAA est l'adresse du module (512 modules de 8 sorties)

D est la valeur 0/1 à appliquer à la sortie SSS

(000=première sortie ... 001=seconde sortie ... 111=8ème sortie)

Ces sorties sont souvent couplées en 4 ensembles de 2 sorties, en effet un aiguillage qui possède 2 bobines à besoin de 2 sorties, de même qu'un feu vert/rouge Lorsque les sorties sont couplées

000=aiguillage droit ou feu vert

001=aiguillage dévié ou feu rouge

En mode couplé, le fait d'activer une sortie désactive la sortie complémentaire.

Il est fréquent de couper une section de voie lorsque le signal est rouge afin d'arrêter les trains (violemment) devant le signal.

Les décodeurs d'accessoires disposent également de CV afin de coupler les sorties, définir des temps d'activation.

Ces temps sont utiles pour envoyer des impulsions précises aux aiguillages. Dans ces conditions, il n'est pas utile d'envoyer un ordre de mise à 0 de la sortie car elle revient à 0 après l'impulsion. Ces CV ne se programment pas avec les mêmes trames de programmations que celle des décodeurs de locomotive.

La gestion des accessoires n'est pas géniale car elle nécessite l'envoi d'une trame chaque fois que l'on veut changer une sortie ! Ce qui surcharge la bande passante du DCC. De plus comme les accessoires ne sont pas mobiles et donc il n'est pas vraiment justifié de les commander par le signal DCC.

PS : Pour ces raisons je n'utiliserai pas ce type de commande dans ma réalisation, étant donnée qu'il faudra utiliser un bus pour les entrées, je mettrai également les sorties sur un bus. Je ne détaille donc pas d'avantage cette partie de la norme.

### A.2.6. Gestion des entrées

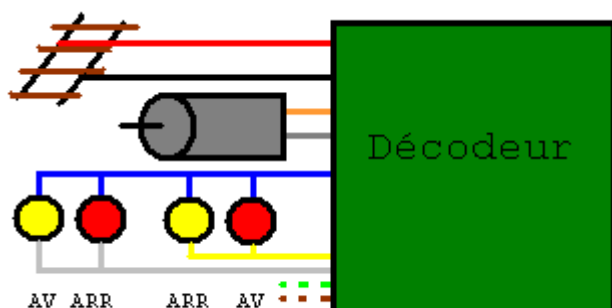
Nous savons maintenant piloter un réseau (locomotives, aiguillages, feux), mais comment connaître son état (occupation des cantons, passage des trains sur des ILS ou pédale de voie ...)?

Ce chapitre va être court car la norme n'a absolument rien prévu ! Conséquence chaque constructeur de centrale propose sa propre solution. Mais la plus populaire est l'utilisation de modules de rétro-signalisations connectés sur un bus S88. Nous reviendrons par la suite sur cette solution car c'est également celle qui a été retenue pour Free-DCC

## B. Les décodeurs

### B.1. Présentation des décodeurs

Vu de l'extérieur, un décodeur est une platine électronique de laquelle sortent des fils. Parfois ces fils vont sur une prise plus ou moins normalisée qui peut même être présente dans les locomotives récentes. Mais le plus souvent il faudra souder directement ces fils aux bons endroits. La couleur des fils est en principe normalisée, mais il vaut mieux lire la notice.



Pour les installations les plus simples il suffit de relier 2 fils (rouge et noir) aux palpeurs de courant et 2 autres (orange ou gris) au moteur. Les autres servants aux phares et fonctions spéciales. Les fils de captages peuvent être connecté au pif bien que la norme indique le contraire. Par contre les fils du moteur influent sur le sens de rotation du moteur. Il est à noter qu'en cas d'erreur, ce sens peut être inverse grâce aux CV. Donc pas d'inquiétude.

Pour les phares et fonctions spéciales un fil (bleu) fournit du 12V. Les autres fils des phares et fonctions spéciales sont reliés à des transistors qui mettent ou non à la masse le fil correspondant. Ainsi lorsque la fonction est activée, le fil est mis à la masse (sinon, il est en l'air) et la lampe entre 12V et la masse s'éclaire. Normalement le fil blanc utilisé pour les phares dans le sens avant, est activé lorsque FL=1 et sens=avant. Idem pour le jaune mais en marche arrière. Les fils des autres couleurs servent aux fonctions spéciales. Sur certains décodeur, il est possible de redéfinir les sorties phares en fonction spéciales.

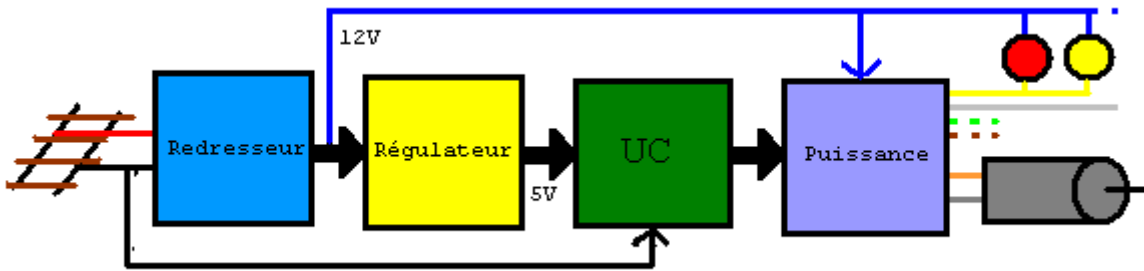
Le choix d'un décodeur se fait suivant les caractéristiques suivantes

- norme (ici DCC)
- taille
- prix
- courant pouvant être délivré au moteur
- existence des sorties phares
- nombre de sorties spéciales (en plus des sorties phares)
- courant pouvant être utilisé pour chaque fonction spéciale.

Il existe dans le commerce toute sorte de décodeur allant de 15 à 120 euros. Les plus simples gèrent uniquement le moteur, Les plus complexes gèrent en plus de multiples sorties spéciales ainsi que le son.

Outre les décodeurs de locomotive, il existe également les décodeurs d'accessoires pour les aiguillages, la signalisation ... Et les décodeurs de fonctions qui sont des décodeurs de locomotive sans la gestion du moteur. Ils peuvent par exemple être utilisés pour contrôler l'éclairage des voitures, les feux de fin de convoie ...

## B.2. Structure des décodeurs

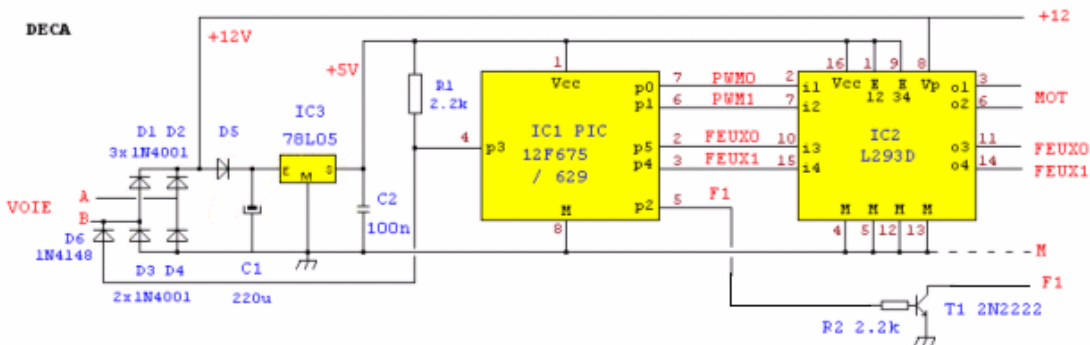


Un décodeur est constitué des éléments suivants :

- Un redresseur qui transforme le + ou - 15V de la voie en une tension continue d'un peu plus de +12V compte tenu des chutes de tension dans les diodes. Ainsi, quelque soit la polarité de la voie, le décodeur est alimentée en 12V.
- Un régulateur qui transforme le +12V en +5V bien propre, ce qui est nécessaire pour alimenter certains circuits électroniques comme le microcontrôleur (UC). Un condensateur peut être rajouté pour garder un peu d'énergie pendant les microcoupures et alimenter le UC.
- Le microcontrôleur commande la partie puissance en fonction des trames qu'il reçoit. Ces trames sont formées de bits récupérés grâce au fil avant le redresseur qui permet de connaître la polarité relative du signal. Le UC est un composant qui contient un petit processeur, de la mémoire et des entrées/sorties. Il exécute un programme qui permet de donner vie au décodeur.
- La partie puissance amplifie les signaux du UC pour piloter le moteur, les phares et les fonctions spéciales. Le moteur est piloté par un composant qui s'appelle pont en H et qui permet de mettre ou pas le 12V aux bornes du moteur et ceci dans un sens ou l'autre.

Il est à noter que le moteur est piloté en PWM pour éviter les échauffements. En PWM, on génère régulièrement des impulsions plus ou moins longues. Le nombre d'impulsion par seconde s'appelle fréquence et s'exprime en hertz (hz). Par exemple a 50Hz, donc 50 impulsions par secondes, la durée de l'impulsion peut varier de 0 à  $1/50=0.02s=20ms$ . La PWM permet d'obtenir des ralentis très lents car le couple instantané est toujours au maximum. Bien entendu, jusqu'à 20kHz, les fréquences s'entendent. Certains prétendent qu'il est préférable d'alimenter les moteur a fer doux à plus de 20kHz pour ne pas les détruire, mais ceci est plus une légende qu'autre chose. Le léger bruit des moteurs n'est pas vraiment gênant. Quelques décodeurs permettent d'ajuster cette fréquence grâce à un CV.

La figure suivante présente le schéma électrique d'un décodeur personnel, mais vu le temps de réalisation et la taille vaut mieux opter pour des modèles du commerce.

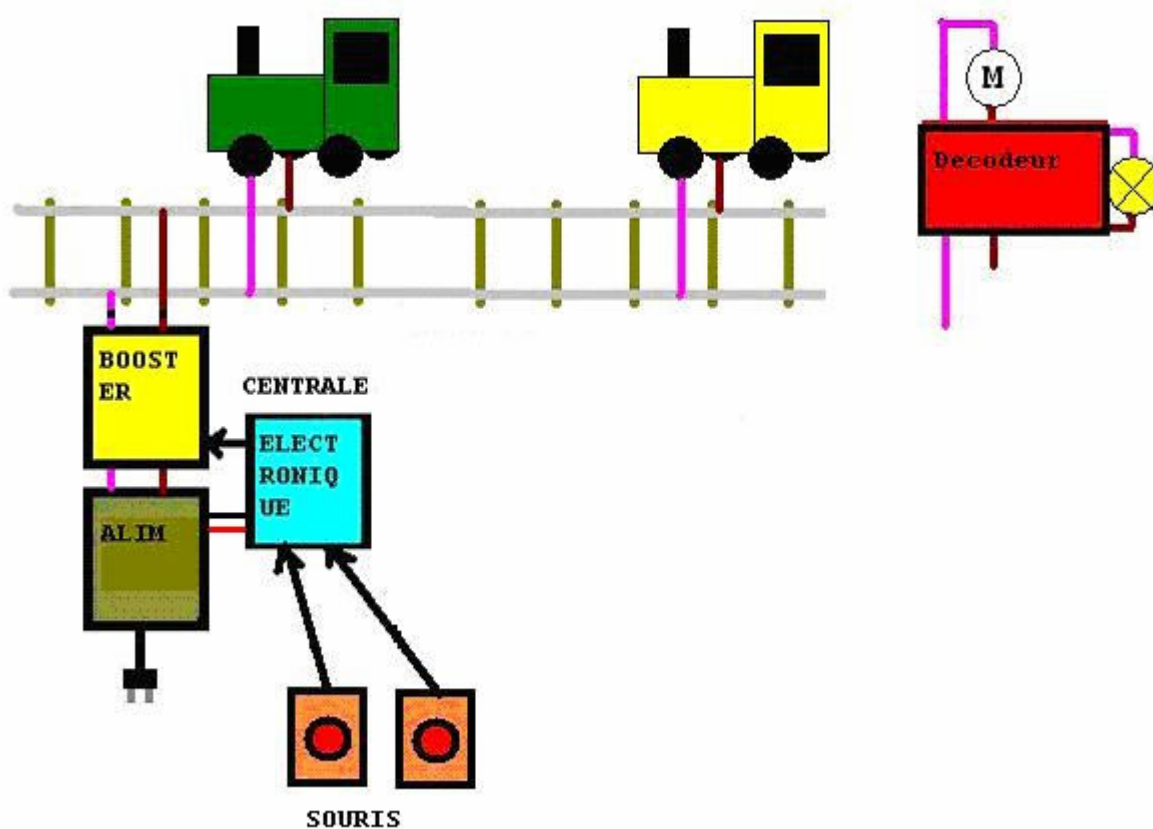


## C. Architecture d'un système DCC

### C.1. La base

La base d'un système DCC a été présentée lors de la comparaison avec le système analogique. Pour mémoire, elle se compose :

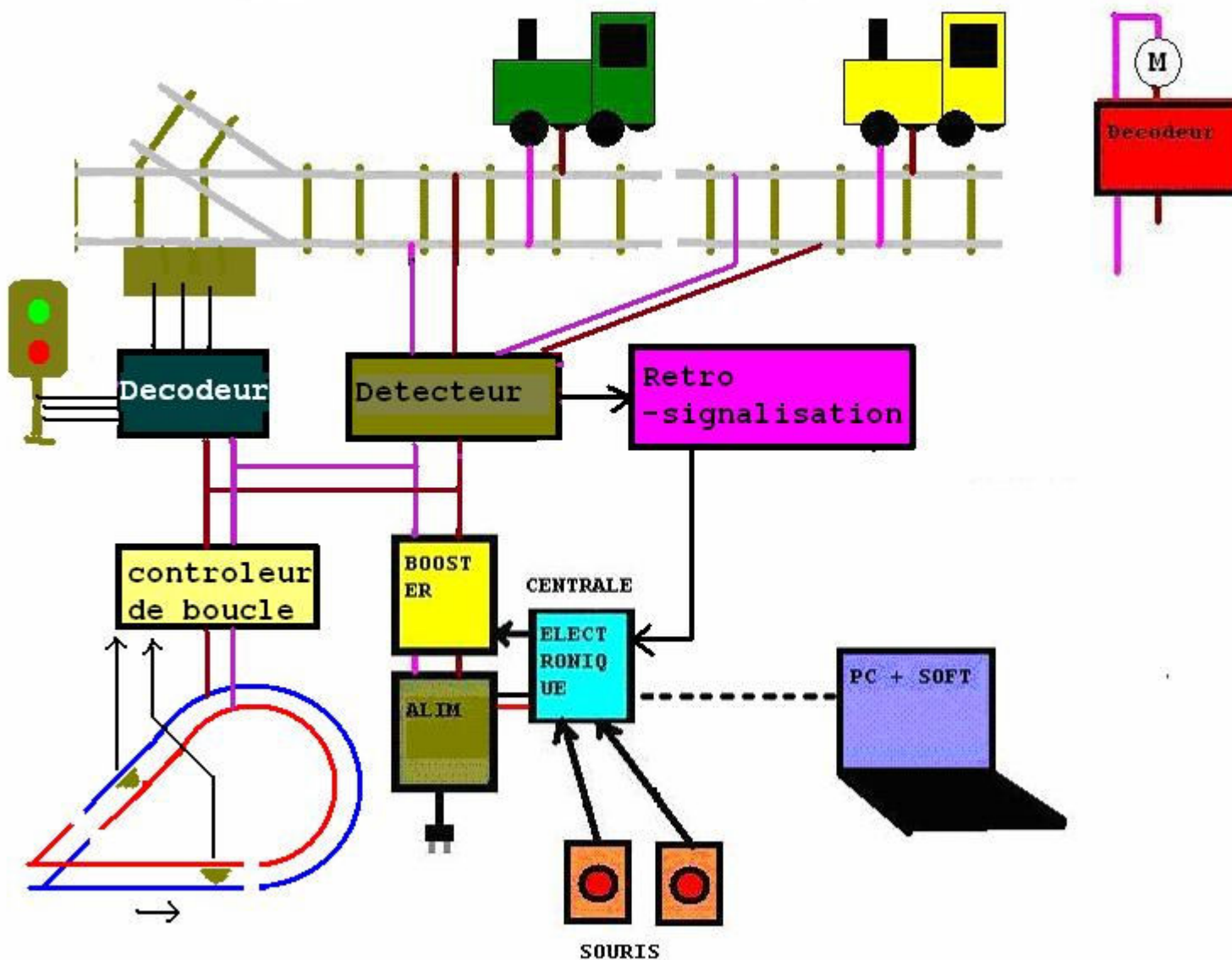
- d'une centrale chargée de produire le signal DCC de commande en fonction de la consigne des souris
- de souris pour que l'utilisateur conduise ses trains
- d'un booster pour fournir le signal DCC de puissance (ex : +/-15V 3A) en fonction du signal de commande (ex : 0/5V 10mA)
- d'une alimentation pour alimenter le tout en courant électrique
- d'un décodeur dans chacune des locomotives



### C.2. Les extensions

L'utilisateur peut rajouter à cette vision de base :

- La commande des aiguillages.
- La commande de la signalisation
- La détection des trains
- La gestion des boucles de retournement
- La connexion à un ordinateur.



Il est tout à fait possible de se passer du DCC pour ces extensions. Par exemple une paire de bouton poussoir permet de commander un aiguillage à distance ...

Mais ce serait se passer d'intéressantes possibilités facilement gérables par un système DCC comme les itinéraires, la conduite automatique, le TCO déporté sur un PC ... La commande des aiguillages et de la signalisation se fait par l'intermédiaire de décodeurs d'accessoires. Pour la détection des trains, c'est un peu plus complexe, car la norme ne prévoit pas de remonter des informations en provenance du réseau. Pour combler ce manque les fabricants de centrales DCC utilisent des bus de retro-signalisation qui permettent de connecter des modules d'entrées auxquels on peut connecter des détecteurs de courant, des pédales de voies ou autre ILS. Connaître la position des trains permet de gérer la signalisation et de conduire automatiquement les trains.

En 2 rails, il est nécessaire d'inverser la polarité des boucles de retournement entre l'entrée et la sortie des trains pour éviter les courts-circuits. À noter que ce PB ne se pose pas en 3 rails.

Enfin comble du raffinement, il est possible de contrôler la centrale et donc le réseau par un PC. Ceci ouvre de nouvelles perspectives comme un TCO, une gestion plus poussée car un ordinateur est beaucoup plus puissant que l'électronique de la centrale ...

### C.3. Coût d'un système DCC

Voici quelques prix du commerce :

Elément	Prix du commerce
centrale DCC (avec alim / booster / 2 souris)	500€
Souris	50€
décodeur 4 aiguillages ou 4 feux	50€
détecteur 8 cantons	50€
module de retro-signalisation a 16 entrées sur bus S88	50€
décodeur de locomotive	30€
boucle de retournement	50€

Ce qui donne par exemple pour un petit réseau comme le mien

(16cantons / 16aiguillages / 6 locomotives / 8 feux (vert ou rouge) / 2 boucles / 4 souris)

$500 \text{ (centrale)} + 2 \times 50 \text{ (souris)} + 6 \times 50 \text{ (dec acc)} + 2 \times 50 \text{ (det)} + 50 \text{ (retro)} + 6 \times 30 \text{ (dec loco)} + 2 \times 50 \text{ (bcl)} = 1320 \text{ €}$

Au vu de ce prix (qui comprend uniquement l'électronique), on peut se demander si le DCC vaut toujours la peine !

Cette prise de conscience permet de tirer plusieurs enseignements. Le premier est qu'il est préférable de réaliser soit même le système. Ainsi FreeDCC divise sans peine cette somme par au moins 4 sans réduire les fonctionnalités. La division est encore plus spectaculaire pour les grands réseaux. Ensuite la norme DCC très bien adaptée aux locomotives l'est beaucoup moins pour les accessoires. Dans ce cas, pourquoi ne pas utiliser un bus accessoires comme on le fait pour les entrées pour réduire les coûts. Car avec 50€ par groupe de 4 aiguillages ou 8 leds de feux, la facture monte vite...

### 3. Presentation de Free-DCC

Le coeur de Free-DCC est sa centrale qui tire sa puissance et sa flexibilité d'un microcontrôleur. Cette centrale possède de nombreuses connections qui lui permet de contrôler le réseau et d'interagir avec les utilisateurs.

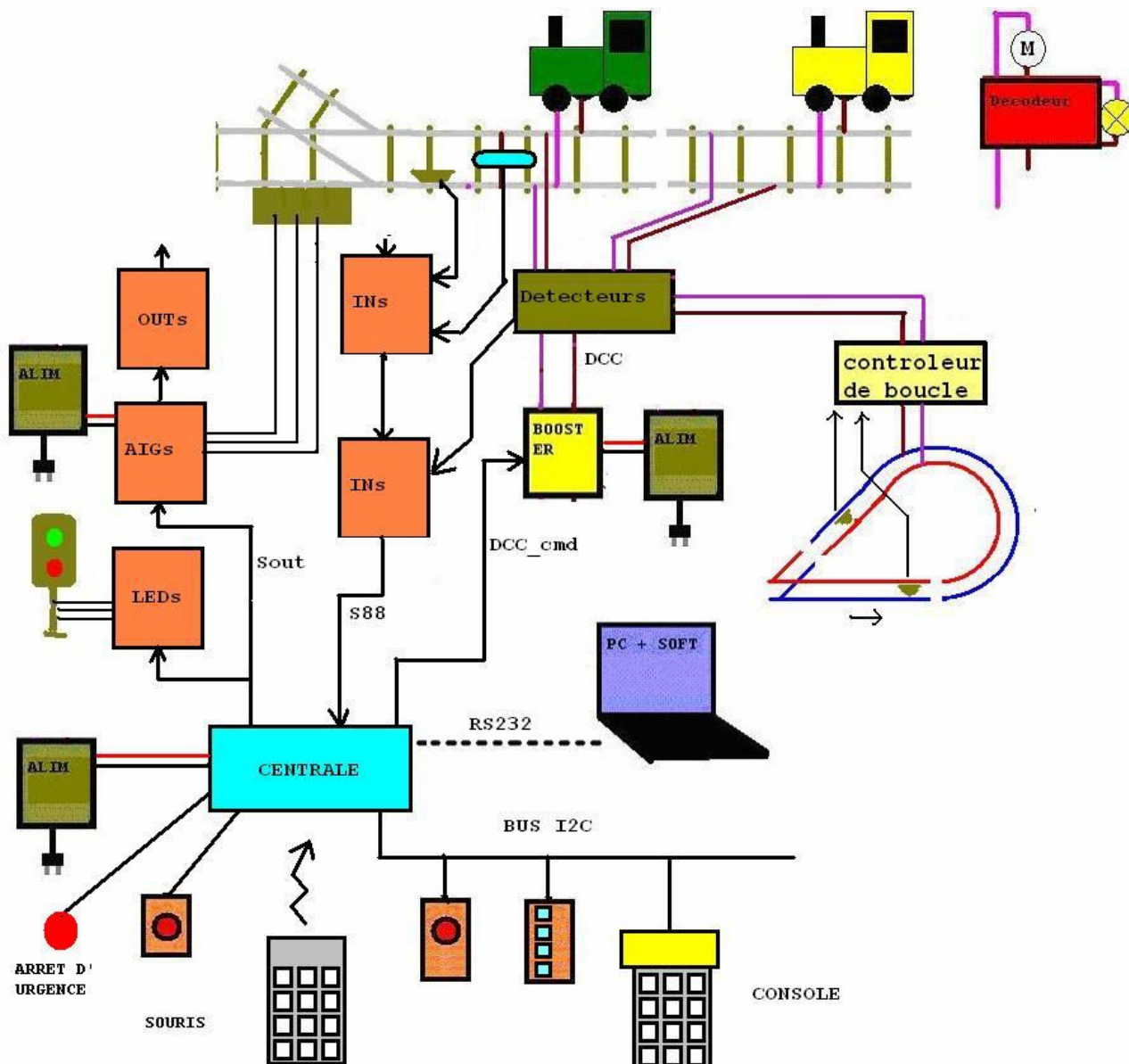
Coté utilisateur on trouve donc diverse souris (potentiomètres a relier directement à la centrale, télécommande de télévision, souris I2C analogiques (avec un potentiomètre) ou numérique (avec des boutons poussoirs). Le bus I2C permet dans notre cas de brancher une souris I2C à n'importe quel endroit du bus. Une console compose d'un écran LCD ainsi que d'un clavier permet de contrôler la centrale. Un bouton d'arrêt d'urgence permet de stopper les trains. Et enfin une liaison RS232 autorise un PC à contrôler la centrale.

Coté réseau, la centrale génère le signal DCC de commande à destination du booster qui alimentera les voies avec le signal DCC de puissance.

Pour les entrées, un classique bus S88 permet d'y connecter des modules d'entrées qui remontent l'état de capteurs (pédales de voies, ILS ...) ou l'occupation des cantons grâce a des détecteurs de courant.

Au niveau des sorties, un second bus dérivé du S88 que nous appellerons Sout permet d'y connecter un module pour piloter les aiguillages, un module pour les leds ainsi que quelques modules de sortie afin d'y connecter divers actionneurs (lampes, moteurs ...)

La figure suivante présente les différents éléments





Ce système étant modulaire, il n'est pas nécessaire d'utiliser tous les modules de la figure précédente et vous trouverez sans doute une configuration qui correspond parfaitement à vos besoins. Il est ainsi possible d'utiliser ma centrale dans des configurations très simples comme avec quelques souris pour piloter uniquement les locomotives et contrôler les aiguillages à la main ou directement avec des boutons poussoirs. De même la centrale s'accommodera de configuration complexes aussi bien en autonome qu'avec l'aide d'un PC ... La modularité autorise également à incorporer des éléments du commerce ou d'autres auteurs, par exemple si vous avez déjà un booster ou des modules S88 pourquoi ne pas les réutiliser. Inversement le booster proposé ou les modules S88 peuvent être utilisés avec des produits du commerce. Pour ceux qui ne souhaitent pas avoir des modules partout sur leur réseau, il est possible de regrouper toute l'électronique sur une seule carte. La figure possède 3 alimentations, mais vous pouvez tout aussi bien réaliser/utiliser une alimentation unique.

Voici les caractéristiques techniques principales du système:

- 16 locomotives peuvent être contrôlées simultanément sur un total de 99 en vitesse (28pas), sens, fonctions FL (feux) et F1 à F4. Les locomotives se contrôlent avec la console de la centrale ou bien avec l'une des nombreuses souris (potentiomètre, numérique/analogique sur bus I2C, télécommande infrarouge ...)
- 6 locomotives peuvent également être contrôlées simultanément en automatique et suivre 50 déplacements prédéfinis. Une horloge de modélisme couplée à un séquenceur permet de lancer les déplacements.
- 32 aiguillages et 32 itinéraires.
- 18 sorties (comme des lampadaires ...)
- 64 leds (pour la signalisation, le TC0). Des équations logiques permettent de contrôler la signalisation.
- 64 entrées (pour détecter les trains)
- La centrale fournit toutes les connections nécessaires au pilotage d'un booster ainsi qu'un mécanisme d'arrêt d'urgence
- Une console à brancher sur bus I2C avec clavier et écran LCD permet de contrôler la centrale en autonome.
- La liaison RS232 permet de contrôler la centrale depuis un PC à l'aide de logiciel du commerce ou personnels afin d'étendre ses fonctionnalités. (Bien entendu la centrale peut aussi fonctionner toute seule.

## 4. Electronique

Ce gros chapitre décrit l'électronique des différents modules de Free-DCC.

Nous traiterons dans l'ordre les modules suivants :

- La partie puissance : Les alimentations et le booster
- La centrale, son microcontrôleur et la liaison rs232
- La console
- Les souris
- Les entrées S88 / les détecteurs de courant / La gestion des boucles de retournement
- Les sorties / aiguillages / leds du bus Sout

Afin de rendre accessible ce chapitre a tout le monde :

- une première partie rappelle quelques notions d'électricité.
- Puis les composants électroniques utilisés sont décrits.
- Enfin chaque module est systématiquement décrit et les schémas largement commentés.

La réalisation est également facilitée grâce à la simplicité de l'électronique et aux typons qui accompagnent systématiquement les différents modules. Ces modules peuvent être regroupés sur une carte électronique afin de réduire les connections. Si la réalisation de circuits imprimés vous intimide vous pouvez toujours utiliser du veroboard (plaque à trou).

## 4.1. 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 il en tombe, plus cela fait mal. 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 500mA sous 12V consomme  $P=UI=12*0.500=6\text{W}$ . La puissance peut aussi s'exprimer en chevaux avec la règle de conversion suivante ( $1\text{kw} = \text{environ } 1.3\text{ch}$ )

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 voiture de 12V chargée possède une énergie de 600Wh. Elle pourrait donc alimenter un moteur consommant 1A sous 12V pendant :  $E=Pt$  ;  $t=E/P$  ;  $t=E/(UI)$  ;  $t=600/(12*1)=50$  heures.

Une petite application pratique au fils.

Plus la tension est grande et plus il faut d'isolant. Pour vous en convaincre, comparez 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 4400kW (5600ch) circule à pleine puissance  $P=UI$  ;  $I=P/U$  :

- sous 1500V :  $I=4400000/1500=2933$  A

- sous 25000V :  $I=4400000/25000=176$ 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 qui s'exprime en ohm et son symbole est omega  $\Omega$ . 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.

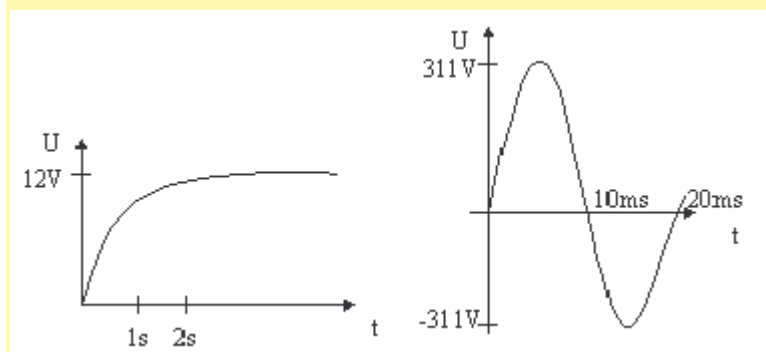
Sachant que le corps humain mouillé est assimilable à une résistance d'environ 1000Ohms et qu'un courant de 30mA le traversant est mortel, il vaut mieux ne pas s'exposer à plus de  $U=RI=1000*0.03=30V$ . (48V en alternatif). Donc pas de problème en modélisme ferroviaire ou les faibles tensions comme 12V sont inoffensives.

Maintenant calculons, le courant dans un fil de 0.1Ohms que l'on utilise pour court-circuiter une batterie de 12V.  $U=RI$  ;  $I=U/R=12/0.1=120A$  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  $R=U/I=10/0.010=1000Ohms$ . 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 a 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  $T=20ms$  et donc de fréquence  $F=1/T=1/0.020=50Hz$ . 1 hertz Hz correspond à une fois par seconde. On retrouve bien le 50Hz du réseau EDF. Et le alors, ou est le 220V? Et bien c'est la tension en continue qui produirait le même effet. 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.

## 4.2. Les composants électroniques utilisés

Cette partie décrit les composants électroniques utilisés. Cela commence avec les composants de base puis se poursuit avec la description des circuits intégrés. Pour plus de détail sur un composant particulier, chercher sa fiche technique ou datasheet en Anglais sur Internet. Beaucoup de fiches sont en Anglais mais l'Anglais technique se comprend facilement et les nombreuses figures sont d'une aide précieuse.

Pour vous procurer les composants vous pouvez :

- Aller au magasin d'électronique de votre ville
- Commander à distance par courrier ou Internet
- Demander des échantillons gratuits sur les sites Web des constructeurs
- Faire de la récupération

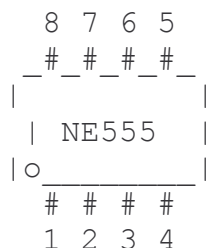
Pour ma part, j'aime bien aller au magasin « Selectronic » à Paris ou demander des échantillons gratuits chez Microchip (<http://www.microchip.com>) (PIC16F876) et Maxim (<http://www.maxim-ic.com>) (MAX232 / MAX7219)

### 4.2.A. Les Composants de base

- **La résistance** déjà présentée précédemment
- **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 en sortie, le courant circule toujours 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 ça 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  $\mu\text{F}$  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 $\mu\text{F}$  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 relais électromagnétique** est un ensemble d'interrupteurs qui sont commandés par une bobine. Lorsque la bobine est alimentée, 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.

## 4.2.B. Les circuits intégrés du système

Les circuits intégrés sont des composants avec de nombreuses pattes reliées à une puce en silicium. Cette puce contient de nombreux composants qui sont souvent des transistors. 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. La figure suivante explicite le principe de numérotation sur le légendaire NE555 qui comporte 8 pattes.



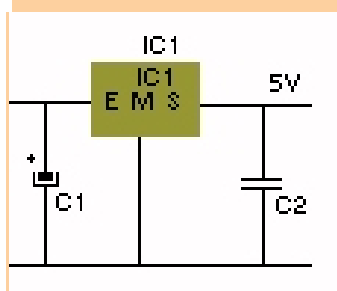
Généralement ces circuits intégrés ou CI s'alimentent avec une tension régulée. Dans notre cas, ce sera du 5V.

== Les régulateurs == (7805 / 7809 / 7812 / 7815 / LM317T)

Le régulateur est un circuit qui permet de fournir en sortie une tension définie quelque soit la tension d'entrée. Dans la plupart des cas la tension d'entrée doit tout de même être supérieure de 2V par rapport à la tension de sortie. Il existe des régulateurs fixes qui fournissent une tension définie ou des régulateurs réglables dont on peut choisir la tension de sortie avec un potentiomètre. Ces régulateurs sont dit linéaires et consomment la différence de tension sous forme thermique. Pour cette raison, il convient de les monter sur des radiateurs.

Les lignes suivantes présentent un régulateur fixe 7805 (05 pour 5V) et le régulateur réglable LM317T.

La figure suivante montre comment obtenir du 5V régulé à partir d'une tension continue d'au moins 7V en entrée. On peut prendre  $C1=1000\mu\text{F}$  en 16V,  $C2=100\text{nF}$ , IC1=7805. Attention à ne pas brancher le condensateur de filtrage C1 à l'envers car sinon il explose et dégage une puanteur tenace. 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).

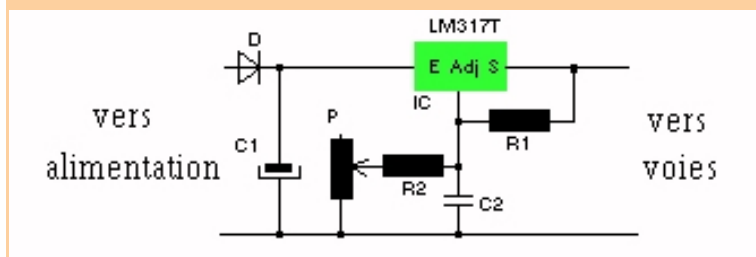


Le 7805 à la forme suivante :

```
|-----|   E = Entrée continue non régulé (doit être >= 7V)
|   O   |
|-----|   M = Masse
|7805 |
|-----|   S = Sortie régulée à 5V pour le 7805
| | |           9V pour le 7809
| | |           12V pour le 7812 ...
E M S       Imax=1.5A
```

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

La figure suivante montre l'utilisation d'un régulateur réglable LM317T



Le LM317T à la forme suivante :

```

|-----|   A = Ajustement
|   O   |
|-----|   S = Sortie réglée
|LM317|
|-----|   E = Entrée continue non régulé (doit être >= sortie+2V)
|   |   |
|   |   |   Imax=1.5A
|   |   |
A S E

```

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

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

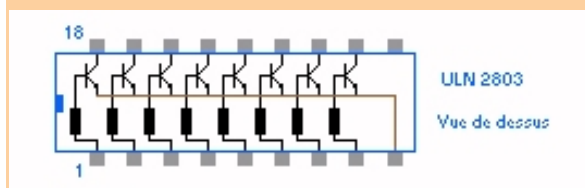
P est un potentiomètre, c'est à dire une résistance de valeur P entre les 2 pattes extrêmes avec une 3eme patte relié au curseur. Ainsi entre une patte et le curseur, la résistance peut être réglée entre 0 et P Ohms. Il existe des potentiomètres linéaires ou rotatifs.

Il est recommandé de choisir R1=220 ohms

Pour plus d'intensité, il est possible d'utiliser des LM317K (3A) ou LM350 (3A), mais les boîtiers sont plus difficiles a utiliser et les régulateurs bien plus chers. Une autre solution est de mettre des LM317T en parallèle. Mais dans ce cas, il faut équilibrer les sorties avec une résistance de 0.2ohm de puissance suffisante.

== Les réseaux de transistors Darlington == (ULN2803)

Ces réseaux de transistors Darlington permettent de commander à partir d'un faible courant disponible en sortie d'un circuit intégré 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



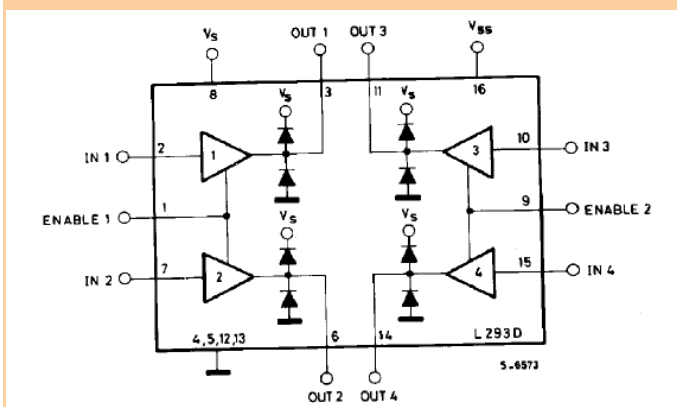


patte à la masse. Pour la logique le circuit dispose de son propre régulateur, mais il faut lui rajouter une petite capacité de 220nF sur la patte Vref. Les sorties sont disponibles sur les pattes OUT1 et 2. Il est conseillé de relier les sorties aux pattes BOOST1 et 2 par l'intermédiaire de condensateur de 15nF afin d'augmenter la vitesse de commutation des transistors.

Les transistors sont des MOS de puissance, c'est-à-dire qu'ils se comportent comme une énorme résistance lorsqu'ils ne conduisent pas et comme une très faible résistance lorsqu'ils conduisent. La résistance à l'état passant est de 0.3ohm ce qui provoque une chute de tension de  $2*3*0.3=1.8V$  à 3A. La tension de sortie variera donc légèrement en fonction de la charge du booster. Ces résistances sont des aubaines pour mettre des boosters en parallèle car elles équilibrent les charges.

Il convient de monter le circuit sur radiateur, car en fonctionnement normal (hors court circuit), il peut dissiper jusqu'à  $P=U*I=1.8*3=5.4W$ . Signalons que le circuit dispose d'une protection contre la surchauffe ce qui l'empêche de partir en fumée en cas de problèmes. Comme le circuit ne possède pas de protection de courant, il est préférable de le précéder d'un régulateur. Cette solution permettra également de régler la tension de sortie aux alentours de 15V en HO et 12V en N

Pour appliquer aux aiguillages la tension accessoires, nous utiliserons les demi ponts (ex : T1/T2 ou T3/T4) comme de simples interrupteurs avec le plus (avec une diode en sortie pour ne pas être embêté par la masse). Ceci simplifiera énormément le montage car contrairement aux interrupteurs avec la masse, il n'existe pas d'équivalent à l'ULN2830 pour les alimentations. Le composant retenu est le L273D de 700mA qui possède 2 ponts soit 4 demi ponts.



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 amplificateurs 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 tensions 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 accessoires.

Enfin pour terminer la description de ce double pont en H, il est à 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. Mais vu le faible temps de manœuvre de s aiguillages, il ne devrait pas chauffer.

De toute façon, 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 courts-circuits.

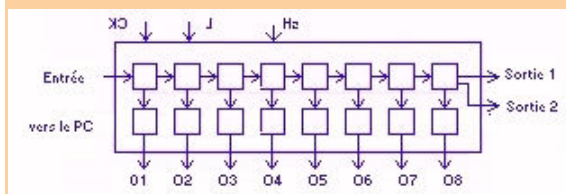
== Le registre à décalage de sortie == (4094)

Les registres à décalage nous permettent d'obtenir autant de sortie que l'on veut en utilisant seulement 3 sorties du UC. Pour commander les sorties et aiguillages nous utiliserons des registres 4094 qui permet de contrôler 8 sorties.

Voici son brochage :



Et le diagramme fonctionnel :



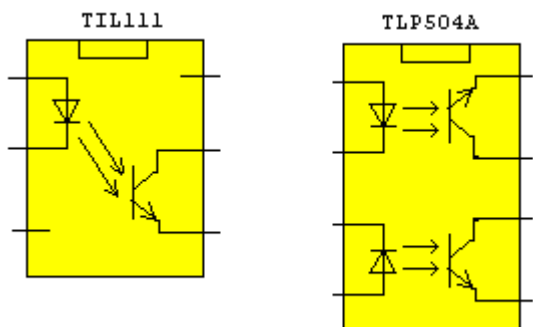
A chaque front positif (passage du niveau bas au niveau haut) d'horloge sur CK ; la donnée présente en entrée est mise dans la première bascule (carré en haut à gauche) alors que l'ancienne donnée de la 1ere va en 2 ... Les données se décalent donc. Lorsque la patte L pour Load (chargement) est à l'état 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 patte 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 voir les données se déplacer en sortie alors que c'est inacceptable pour les 2 autres groupes de registres.

== Les optocoupleurs == (TIL111 / TLP504A)

Un optocoupleur se compose d'une LED qui lorsqu'elle est alimentée, éclaire un phototransistor qui devient passant. Cela permet d'isoler électriquement les 2 parties de l'optocoupleur et donc du montage associé. Dans notre cas, ils seront utilisés dans ce montage pour réaliser des capteurs de courant. On trouve des optocoupleurs simples ou multiples.





== Les circuits d'interface I2C numérique et numérique == (PCF8574 / PCF 8591)

Le bus I2C permet de contrôler un ensemble de circuits compatibles I2C avec seulement 2 fils nommés SDA (Data) et SCK (Horloge). Dans notre cas, ce sera bien entendu la centrale qui pilotera ce bus. Sur un bus I2C chaque composant à une adresse qui se compose d'une partie fixe liée au circuit et d'une partie réglable par la mise à la masse ou au +5V des pattes d'adresses. Ce bus est conçu pour avoir une longueur de quelques mètres tout au plus. Mais par diminution de la résistance de pull-up de SDA, comme dans notre cas, il est possible d'augmenter cette portée à 10m. Ensuite il faudrait modifier le logiciel pour baisser la vitesse. Une caractéristique intéressante du bus I2C est qu'il est possible de brancher et débrancher les circuits du bus sans couper l'alimentation.

Nous utiliserons les circuits suivants :

- PCF8574 : interface d'entrées/sorties qui permet de disposer de 8 entrées/sorties (adresses : 32-39 )
- PCF8574A : Idem mais avec d'autres adresses (adresses : 56-63 )
- PCF8591 : convertisseur analogique/numérique qui permet de mesurer jusqu'à 4 tensions (adresses : 68-75 )

```
+5 SDASCKINTP7 P6 P5 P4
16 15 14 13 12 11 10 09
# # # # # # # #
#####
# PCF8574 # adresse de base 0100+A2A1A0+R/W#
##### adresse de base version A 0111+A2A1A0+R/W#
# # # # # # # #
01 02 03 04 05 06 07 08
A0 A1 A2 P0 P1 P2 P3 MM
```

Ce circuit permet de lire ou écrire sur 8 entrées/sorties P0 à P7. Chaque port dispose d'une pull-up. Les sorties sont à collecteurs ouverts, elles sont donc assimilables à des interrupteurs qui peuvent relier la masse pour l'état 0. L'état 1 étant assuré par les pull-up. Reste ensuite l'alimentation et les signaux I2C SDA et SCK ainsi que les pattes de sélection d'adresse du circuit. La broche INT quand à elle peut être utilisée pour signaler le changement d'état des entrées, mais comme cela rajoute un fil, elle n'est pas utilisée.

```
+5 OUTREFAMMEXTOSCSCKSDA adresse de base 1001+A2A1A0+R/W#
16 15 14 13 12 11 10 09 oscillateur interne EXT=MM OSC=NC
# # # # # # # # plage 5V REF=+5 AMM=MM
#####
# PCF8591 # reg: 0.OUTON.INCONFx2.0.autoinc.INFIRSTx2
##### INCONF: 00=4MC, 01=3Min3, 10=2MC+1DIFF, 11=2DIFF
# # # # # # # # INFIRST: canal a convertir durant le mot après du mot d'adresse
01 02 03 04 05 06 07 08 le mot après le mot d'adresse vient de la conversion précédente
IN0IN1IN2IN3A0 A1 A2 MM a la mise sous tension config a 0.0.0.0.0.0.0
```

Ce circuit permet de mesurer les 4 tensions présentes sur les pattes IN0 à IN3. Il peut également sortir une tension sur la sortie analogique OUT. La plage du convertisseur analogique numérique se règle avec Ref et AMM, nous choisirons (5V et masse). Nous utiliserons l'oscillateur interne pour les conversions (donc EXT à la masse et OS non connecté) Reste ensuite l'alimentation et les signaux I2C SDA et SCK ainsi que les pattes de sélection d'adresse du circuit.

== L'adaptateur de niveau RS232 == (MAX232)

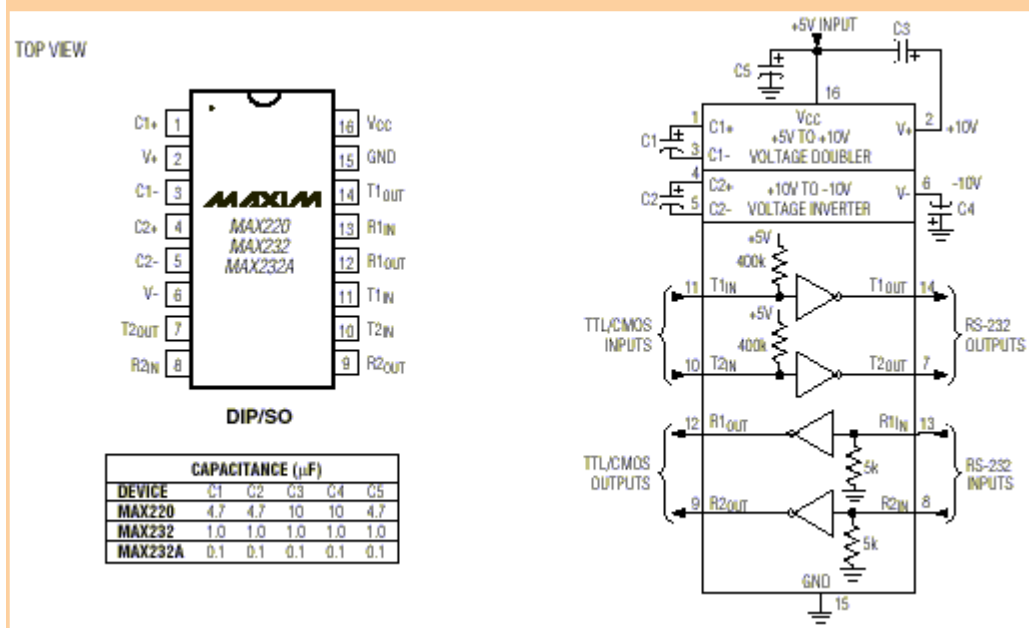
Le MAX232

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 connections 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

== Le microcontrôleur == (PIC16F876(A) / PIC16F873(A) )

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 flash 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 (enc\_it\_5.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 .asm.

## 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>.

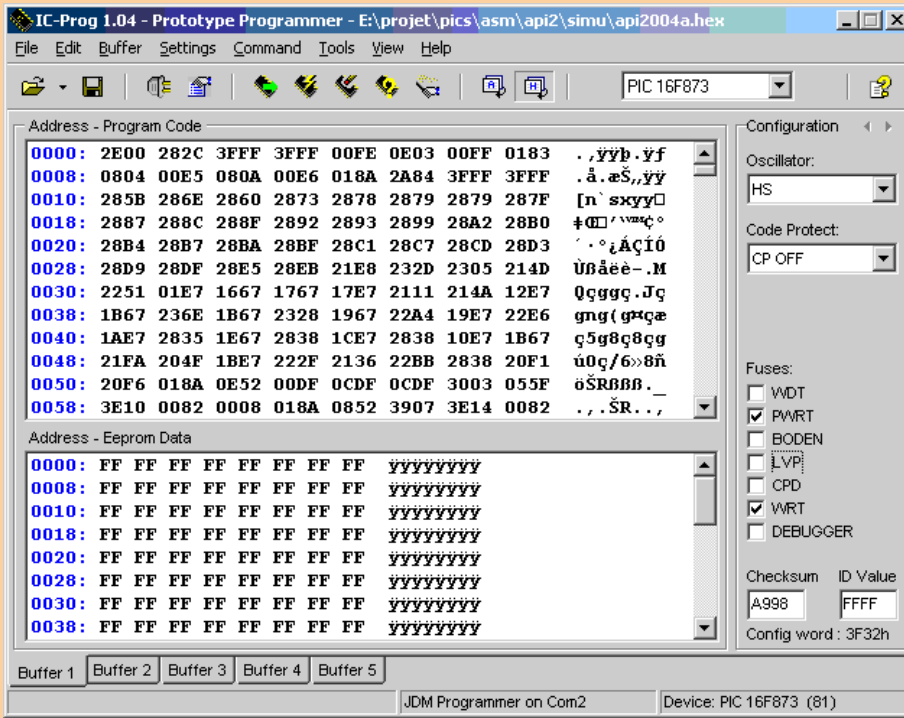
Pensez à configurer correctement le soft :

- Device = PIC 16F876 (ou 876A suivant votre modèle).
- 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 : 1 (autorisation pour le programme d'écrire dans sa propre mémoire)
  - DBUGGER :0 (pas de debugger)

La copie d'écran suivant montre icprog

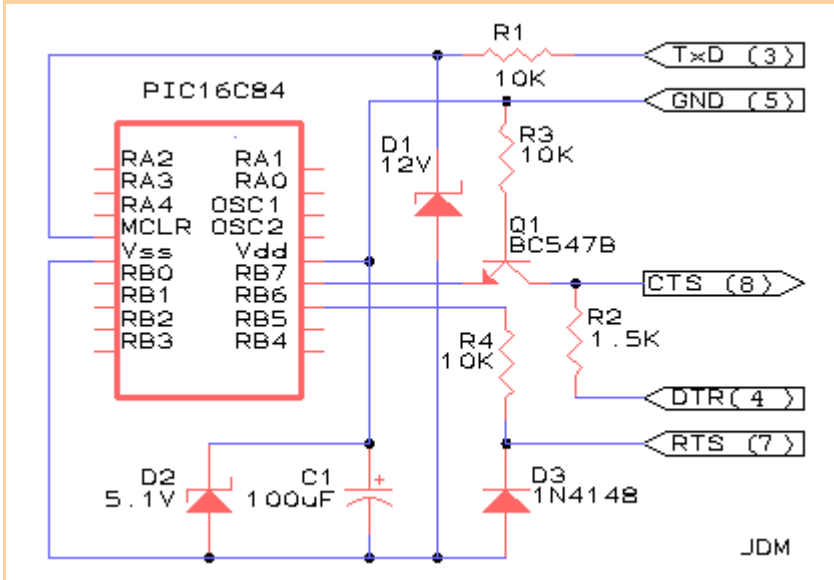


La partie du haut contient les codes du programme. La partie du bas les données pour l'EEPROM. C'est dans cette partie que vous pouvez régler les variables de configuration CC de la centrale. Attention, les données ne sont ni en décimal, ni en binaire mais en hexadécimal. (Utilisez la calculatrice de Windows pour convertir)

### 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 gravés 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).





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 pour la centrale :

```

S L L D R L C D + G      D D      DCC/DC2 : signal DCC et signal DCC complémenté
C D D O S D L I 5 N R T C C      IR : entrée de réception infrarouge
K L O U T I K N V D X X 2 C      AU : entrée d'arrêt d'urgence
| | | | | | | | | | | | | |      DOU/CLK/LDO/LDL : Bus des sorties (AIG+LED+OUT)
#####
#                               #      DIN/CLK/LDI/RST : Bus S88 pour les entrées
#           16F876 (A)         #      SDA/SCK : Données et horloge u bus I2C
#o                               #      POT0...4 : Entrées analogiques (pour les potentiomètres)
#####
| | | | | | | | | | | | | |      CLI : sortie clignotante (pour voir si le UC marche)
/ P P P P S P G O O I x C A      RX : réception RS232
R O O O O D O N S S R      L U      TX : transmission RS232
A T T T T A T D C C      I      OSC : oscillateur
Z 0 1 2 3      4      GND : masse
                                +5V : alimentation du UC
                                /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 1k entre le 5V et la patte CLI.

\* La figure suivante montre le brochage du UC, une fois programmé pour un module S88 :

```

                                + G      I I
I I I I I I I I 5 N I I 1 1      I0-15 : entrées
0 1 2 3 4 5 6 7 V D 8 9 0 1      CLK : donnée suivante
| | | | | | | | | | | | | |      DIN : donnée série en provenance du module précédent
#####
#                               #      DOUT : donnée série pour le module suivant ou la centrale
#           16F873 (A) /16F876 (A) #      LDIN : chargement des entrées dans le registre à décalage
#o                               #      EFF : Effacement des données mémorisées
#####
| | | | | | | | | | | | | |      MEM : sortie de mémorisation C0/C1 pour les boucles
/ D D M E L C G O O I I I I      OSC : oscillateur
R I O E F D L N S S 1 1 1 1      GND : masse
A N U M F I K D C C 5 4 3 2      +5V : alimentation du UC
Z      T      N      /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 1k entre le 5V et la patte CLI.

Vous voilà fin prêt pour comprendre la suite du document..

### 4.3. La réalisation

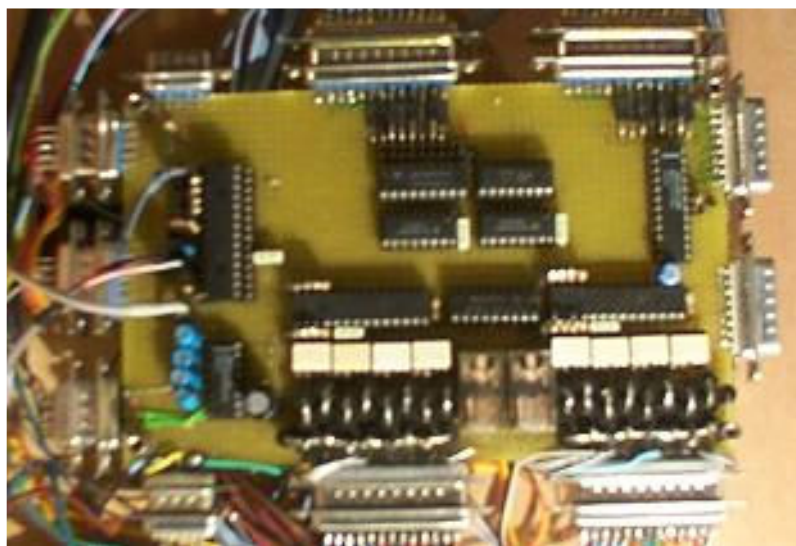
Pour réaliser les modules vous pouvez utiliser

- Soit utiliser des plaques « veroboard » qui sont des plaques à trous avec des pastilles (ne pas prendre les modèles à bandes). Cette solution est facile et ne demande pas de matériel trop spécifique. Pour un résultat correcte, il est conseillé de commencer par souder les supports, puis créer les pistes d'alimentations avec du fil nu ou des queues de résistances, puis souder les composants et essayer de les relier avec leurs queues ou avec des queues de résistances et enfin relier ce qui ne peut pas l'être directement avec du fil de faible diamètre comme du fila à wrapper.
- Soit réaliser des circuits imprimés en vous inspirant des typons fournis. Vous pouvez utiliser la méthode des transferts ou encore avoir recours à une machine à insoler. A noter qu'il existe aussi la possibilité de faire réaliser son circuit imprimé sur Internet. Cette méthode offre un résultat professionnel et un temps de montage rapide.

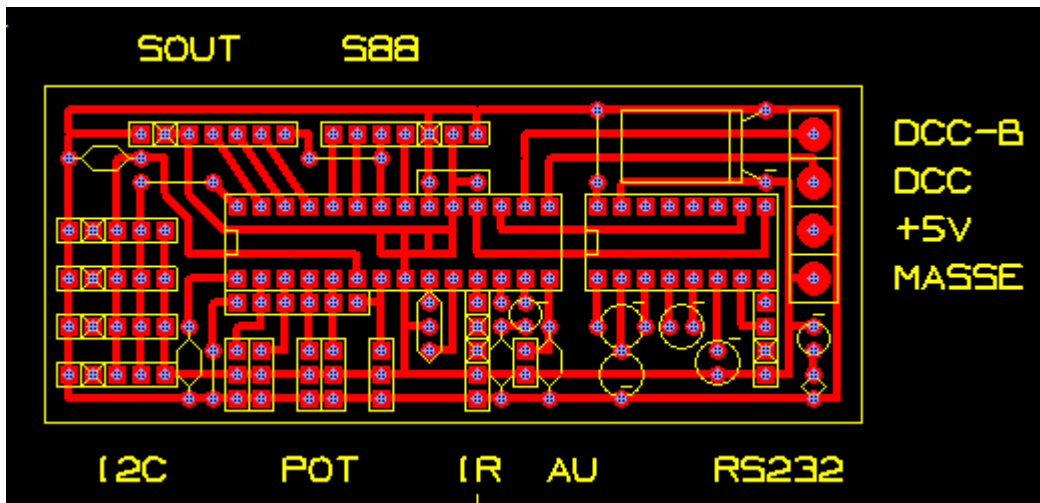
Les typons fournis ont été réalisés à la main avec l'excellent freeware TCI gratuit de Bruno Urbani téléchargeable gratuitement à partir de <http://b.urban.free.fr> . Pour réaliser un typon, lancer le logiciel, charger le fichier qui contient tous les typons « centrale\_free\_dcc.tci », sélectionner uniquement les couches des pistes puis imprimer le typon du module que vous souhaitez réaliser. Après impression, il est vivement conseillé de vérifier avec de vrais composants si les empreintes sont correctes car certaines imprimantes n'impriment pas exactement comme il faut et il serait dommage de recommencer un circuit imprimé pour cela. Si la taille n'est pas bonne, vous pouvez rectifier le tir avec le zoom d'une photocopieuse. N'utilisez pas les typons illustrant ce document car ils ne sont pas à l'échelle et contiennent la couche de sérigraphie.

Au lieu de réaliser une carte par module, vous pouvez également regrouper sur une carte plusieurs modules et pourquoi pas comme moi tous les modules dont vous avez besoin ! Pour ce faire avec l'aide de TCI, collez les modules les uns aux autres puis faites vos circuits imprimés. Il suffira ensuite de ne pas monter les connecteurs inter modules et de les remplacer par des fils. Si vous avez un peu de patience vous pouvez également relier les modules entre eux avec TCI. Vous pouvez également utiliser TCI pour modifier les circuits imprimés par exemple si l'empreinte d'un composant ne correspond pas (comme cela peut être le cas pour les relais), ou encore si vous voulez changer les connecteurs.

La photo suivante montre ma carte qui contient la centrale, la gestion de 16 aiguillages, la gestion des leds, 2 modules S88 ainsi que 16 détecteurs et 2 gestions de boucles. Comme c'était un prototype, je l'ai réalisée avec une plaque à trou. J'ai ensuite créé les typons pour vous rendre la réalisation plus facile et me permettre un jour d'avoir un résultat plus propre. De ce fait, les typons ne sont pas testés et je compte sur vous pour me dire s'ils fonctionnent ! Comme vous le voyez, j'utilise des connecteurs DB9 pour la liaison série et le bus I2C contrairement aux barrettes autosécables proposées sur les typons.

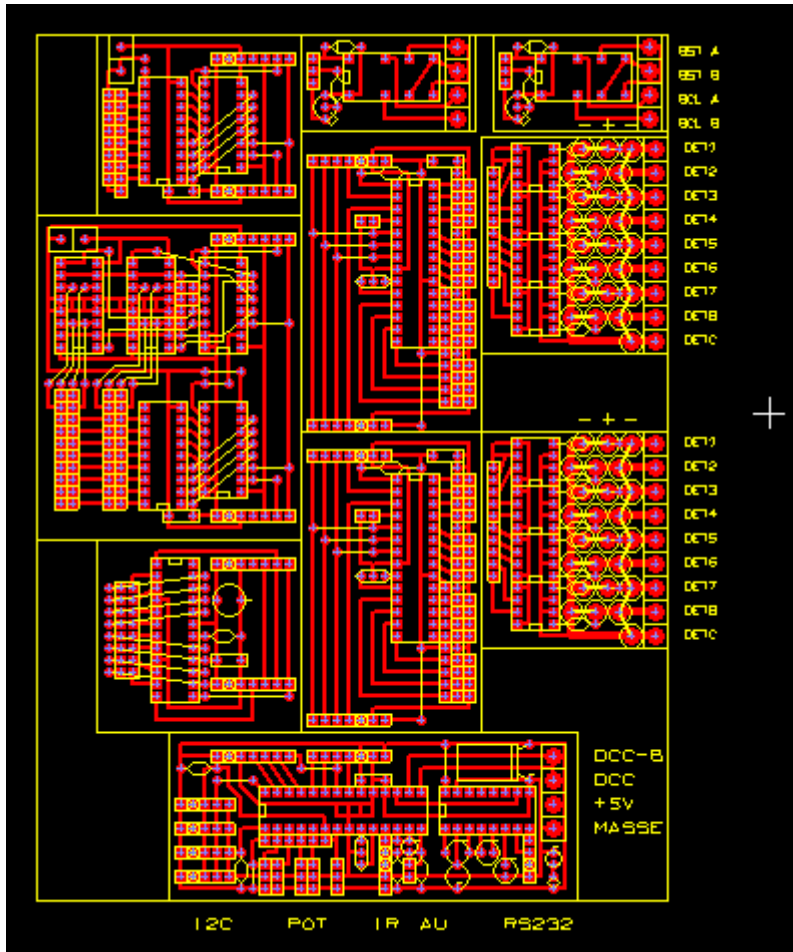


Le typon suivant (pas à l'échelle !) est celui de la centrale



Alors que celui-ci regroupe les modules suivants

- La centrale
- Le module 64 LEDs
- Le module 32 aiguillages
- Le module à 8 sorties
- 2 modules S88 soit 32 entrées
- 16 détecteurs de courant
- 2 boucles de retournement



Bonnes réalisations ! Et envoyez-moi vos photos !!

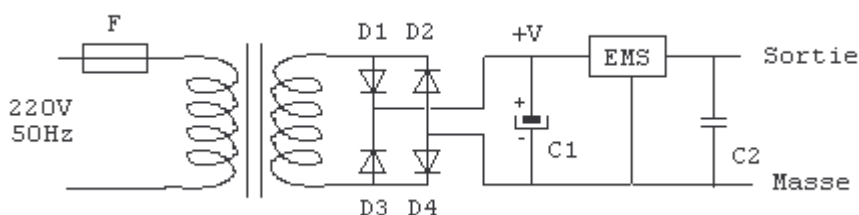
## 4.4. La partie puissance – Les alimentations et le booster

Cette partie traite des alimentations et du booster. Les alimentations ont pour but de fournir du courant électrique continu sous tension constante aux différents équipements. Quand au booster son rôle est d'appliquer le signal DCC de commande aux voies.

### 4.4.A Les alimentations

Tout montage électronique à besoin d'une alimentation pour fonctionner, nous allons voir dans cette partie comment alimenter l'électronique et le réseau. 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. Lors de l'achat d'un transformateur, il faut spécifier, le type, le nombre d'enroulement du secondaire, la tension primaire, la tension secondaire et la puissance. Le type est la technologie par exemple normal, torique ... Un enroulement secondaire est suffisant dans notre cas, mais il peut être intéressant d'en avoir plusieurs pour différentes tensions ou pour utiliser des montages à point milieu. On choisira 220V pour le primaire et une tension secondaire plus grande que la tension de sortie en tenant compte des pertes dans les diodes et le régulateur. Par exemple 15V est un minimum pour une tension de sortie de 12V. Attention, cette tension efficace n'est pas la tension instantanée maximale qui vaut  $V_{max}=1.4V_{eff}$  soit 21V pour 15V<sub>eff</sub>. Il faudra en tenir compte pour le choix des composants. Enfin la puissance efficace du transformateur est donnée en VA. Pour faire simple disons que cela correspond à la puissance disponible au secondaire. Si nous voulons 2A, il faudra donc prendre dans notre exemple un transfo de  $P=UI=2*15=30W$  soit environ 30VA
- Pont de diode : Pour redresser le courant. Pensez à spécifier la tension inverse et le courant max.
- 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. On prend généralement un condensateur chimique capable de supporter la tension maximale du transformateur et de capacité de 1000uF par ampère.
- 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.
- Condensateur de découplage pour éliminer les composantes haute fréquence. Un petit condensateur plastique de 100nF suffit.

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 que le continu régulé.

## Nos alimentations

Pour faire fonctionner Free-DCC, nous aurons besoin des tensions suivantes :

- 5V (500mA) pour alimenter l'électronique
- 15V (1A) pour alimenter les aiguillages
- 18V (3A) pour alimenter le booster

Pour réaliser ces alimentations vous pouvez les faire vous-même en utilisant les explications précédentes ou utiliser en totalité ou en partie des alimentations du commerce. Par exemple pour l'alimentation 5V, il est possible de prendre un bloc secteur 9V 500mA et de lui rajouter un régulateur 7805... Il est également possible d'utiliser un seul transformateur et 3 régulateurs. Mais dans ce cas, il faut mettre le régulateur 5V après celui de 15V car il risque de ne pas supporter la tension d'entrée trop grande. C'est plus simple mais pas nécessairement une bonne idée au niveau rendement car si le transfo délivre 20V pour le DCC, le régulateur 5V doit manger 15V pour en laisser 5 en sortie. Autrement dit pour chaque watt consommé, 3 parties sous forme de chaleur. Pour ma part, j'ai modifié une alimentation de « CB 13.8V » pour me fournir mes 3 tensions. J'y ai même intégré le booster.

### 4.4.B Le booster

Le booster est un élément capable d'inverser la tension sur les voies en fonction du signal DCC de commande. En clair :

- Lorsque le signal de commande est à 0 (0V), il fournit -15V
- Lorsque le signal de commande est à 1 (5V), il fournit +15V

Il n'y a aucun problème si le fonctionnement est inversé.

Un booster se caractérise par son intensité maximale. On trouve couramment des modèles 3A, 5A ou 10A. Je ne conseille pas d'utiliser des boosters de plus de 3A car en cas de surcharge ils pourraient détruire les diodes 3A des détecteurs souvent utilisés pour détecter les trains. Ou plus embêtant faire fondre les fils de captage des locomotives en cas de problème. Si vous avez besoin de plus de 3A, il est préférable d'utiliser plusieurs booster 3A ou alors un gros booster alimentant plusieurs sorties protégées chacune par un disjoncteur 3A.

Pour réaliser cet inverseur, il est possible d'utiliser un montage en pont en H qui grâce à 4 transistors de puissance permet d'inverser cette tension. La figure suivante présente ce principe :

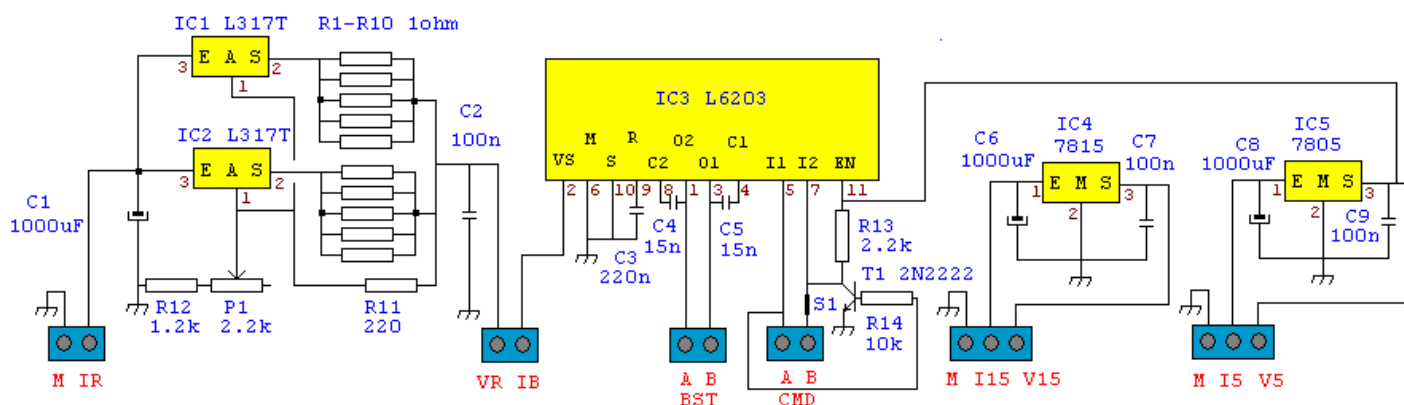
Il est bien plus commode d'utiliser des circuits de pont en H tout fait. En effet, la place occupée est moindre et ils possèdent de nombreuses protections. Les circuits courants pouvant être utilisés sont les suivants:

- LMD 18200 T (3A) particulièrement bien adapté mais horriblement cher (30€)
- L298N (2\*2A) mais ne dispose d'aucune protection
- L293N/D (2\*0.7A) pas assez puissant
- L6203 (3A) convient parfaitement

J'ai donc retenu le L6203 qui permet de délivrer 3A et possède une protection thermique. Il est à noter que grâce à la protection thermique ce composant ne peut pas partir en fumée, par contre il ne comporte pas de protection contre les courts circuits. Il est donc conseillé de le précéder d'un régulateur afin de limiter l'intensité.

#### 4.4.C Réalisation

Le schéma suivant présente une petite carte permettant de mettre en œuvre les régulateurs et le L6203. Il suffit d'y connecter un ou plusieurs transformateurs (avec un redresseur). Bien entendu compte tenu des explications précédentes, vous pouvez utiliser des solutions différentes.



On trouve d'abord une alimentation réglable 3A destiné à alimenter le booster. Afin d'être économique, cette alimentation utilise des LM317T. Comme l'intensité maximale d'un LM317T est de 1.5A, il faut en mettre 2 en parallèle pour atteindre 3A. Les sorties sont équilibrées entre elles grâce à des résistances de 0.2ohms. Afin d'obtenir facilement une résistance de 0.2ohm de puissance suffisante, 5 résistance de 1 ohm en parallèle sont nécessaires. L'ajustable permet de régler la tension de sortie pour par exemple baisser cette tension en N. A ce niveau, 15-18V constitue une bonne tension en HO tandis que 12V convient bien en N.

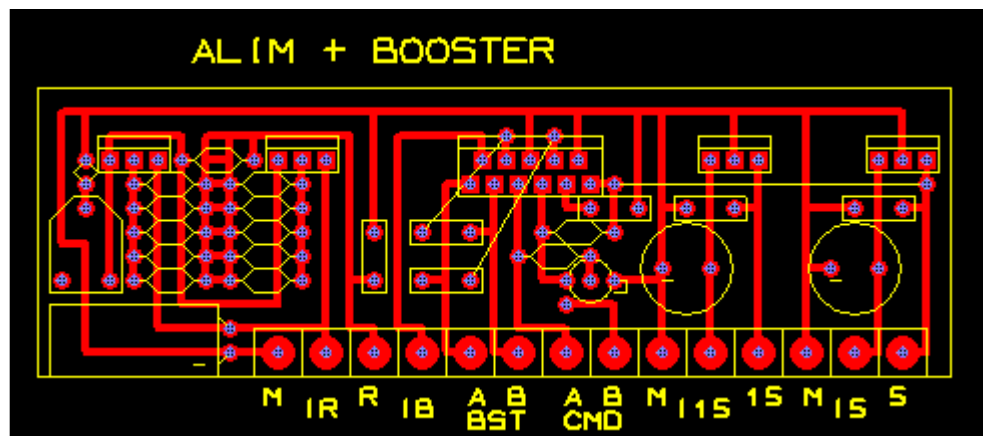
L'alimentation est complétée d'un régulateur 5V et d'un autre 15V.

Le 6203 est utilisé pour la partie booster. Le signal de commande DCC pilote directement le premier demi pont. Tandis que le second est piloté en inversant ce signal à l'aide du montage à transistor. Il est également possible de piloter ce second demi pont directement par le signal cmdDCC2, dans ce cas, ne pas monter le transistor et ses résistances et mettre le strap. Le fait de commander indépendamment les 2 demi ponts ne sert à rien en DCC mais cela permet de générer un signal PWM pour utiliser temporairement une locomotive analogique. Le 6203 est également entouré de 3 condensateurs nécessaires à son fonctionnement optimal.

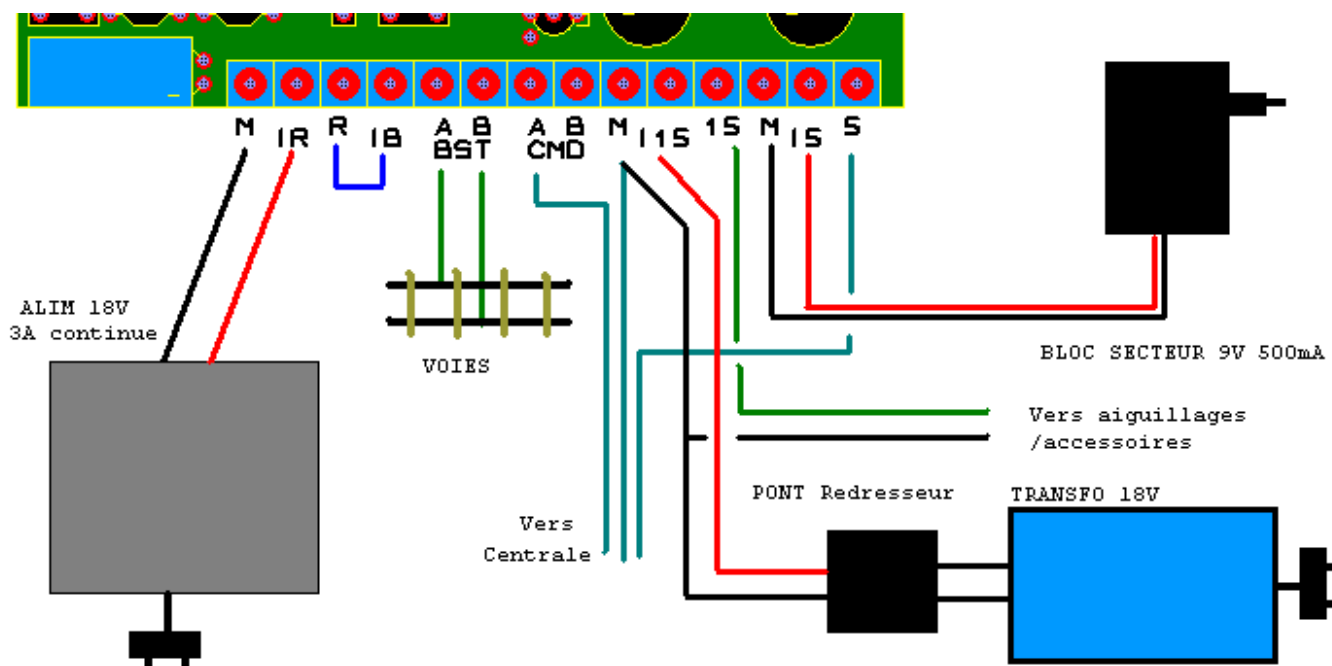
Pour la réalisation, vous pouvez utiliser :

- le typon suivant (Utilisez celui de l'archive)
- une plaque pastillée percée
- des dominos

Pensez à monter les régulateurs et le pont en H sur des radiateurs. Mais attentions les languettes des LM317T sont reliés à la tension d'entrée tandis que celles des 3 autres circuits sont reliés à la masse.

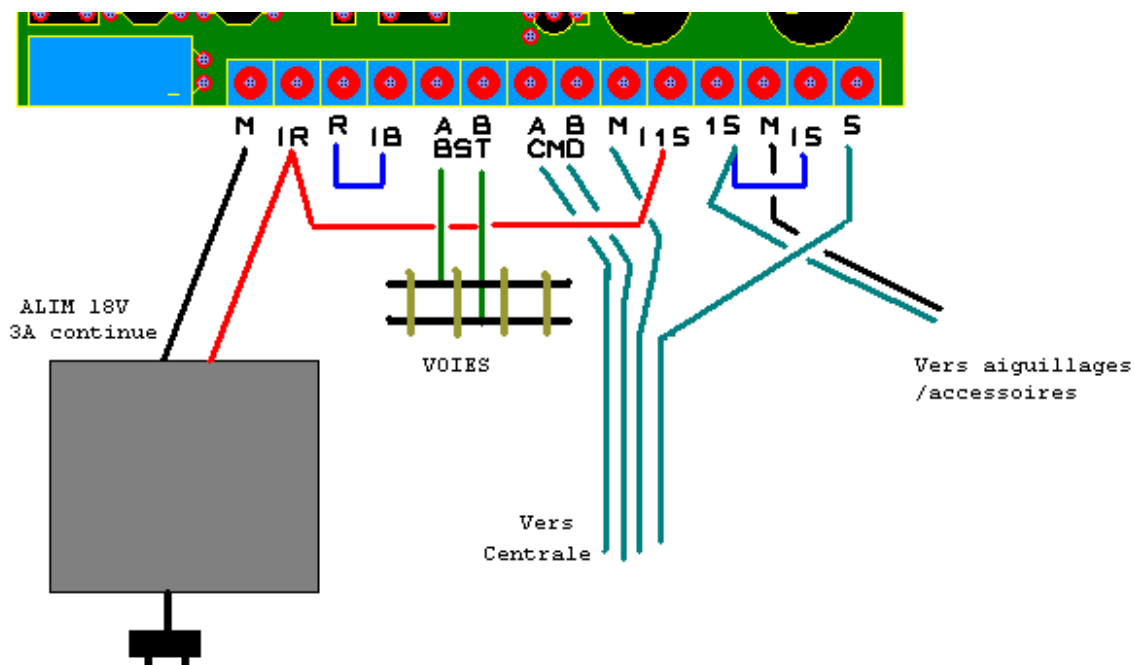


Les figures suivantes présentent quelques configurations possibles.



Dans cette première utilisation :

- Une alimentation continue 18V 3A alimente la partie booster
- Un simple transfo 18V 1A suivi d'un pont redresseur alimente la partie accessoires
- Un bloc secteur 9V 500mA alimente la partie électronique
- La centrale reçoit le 5V et la masse et fournit le signal DCC au 1er demi pont du booster. (Le transistor complète ce signal pour commander le second demi pont)

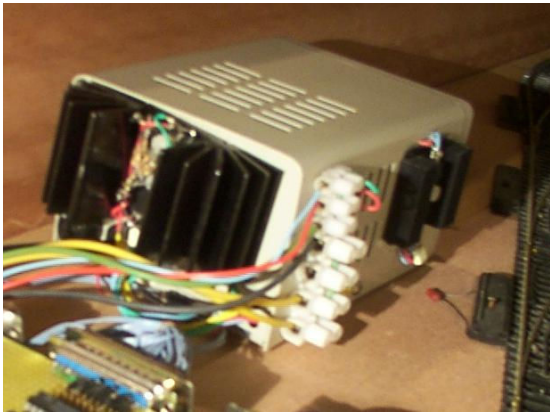


Dans cette seconde utilisation :

- Une alimentation continue 18V 3A alimente la partie booster, la partie accessoire et la partie électronique. Le 5V pour l'électronique est fabriqué à partir du 15V car le 18V est à la limite des caractéristiques du 7805. Cette implémentation est simple mais le 7805 chauffe énormément. Une solution intermédiaire avec un bloc secteur pour le 5V serait plus judicieuse du point de vue de la dissipation.
- Cette fois la centrale pilote directement les 2 demis ponts en H. Il est donc nécessaire de ne pas souder le transistor et ses 2 résistances. Par contre il faut souder le strap S1. Cette solution économise le transistor mais permet surtout d'utiliser le mode PWM de la centrale pour utiliser une locomotive analogique sur le réseau sans rien modifier.

Conseils : Pour ma part, j'ai utilisé une alimentation très bon marché 13.8V 3A de chez Selectronic (SL1760). Cette alimentation est destinée à alimenter des accessoires pour l'automobile (CB ...) à la maison. Elle délivre 13.8V mais la régulation est effectués avec des composants anciens moins performants que les régulateurs (mais moins chers) qui demande une perte de tension importante. Ne nous en plaignons pas car cela réduit le pris de l'alimentation qui est vendu 20 euros ! De plus en retirant toute l'électronique de la carte de l'alimentation mais en gardant le condensateur de filtrage de 3300uF et les diodes du pont redresseurs (il n'y en a que 2 car le transformateur est à point milieu), on obtient une tension continue de 18V au moins. Il ne reste donc plus qu'à raccorder cette tension à notre carte. Les plus astucieux pourront même remplacer le transistor ballast par un LM317K ou encore comme mois monter les 317T et le L6203 sur ce radiateur, et les 2 autres régulateurs sur les flancs mais attention a bien isoler ! Pour le 5V si le 7805 chauffe trop, il est toujours possible d'utiliser un bloc secteur 9V 500mA dédié.

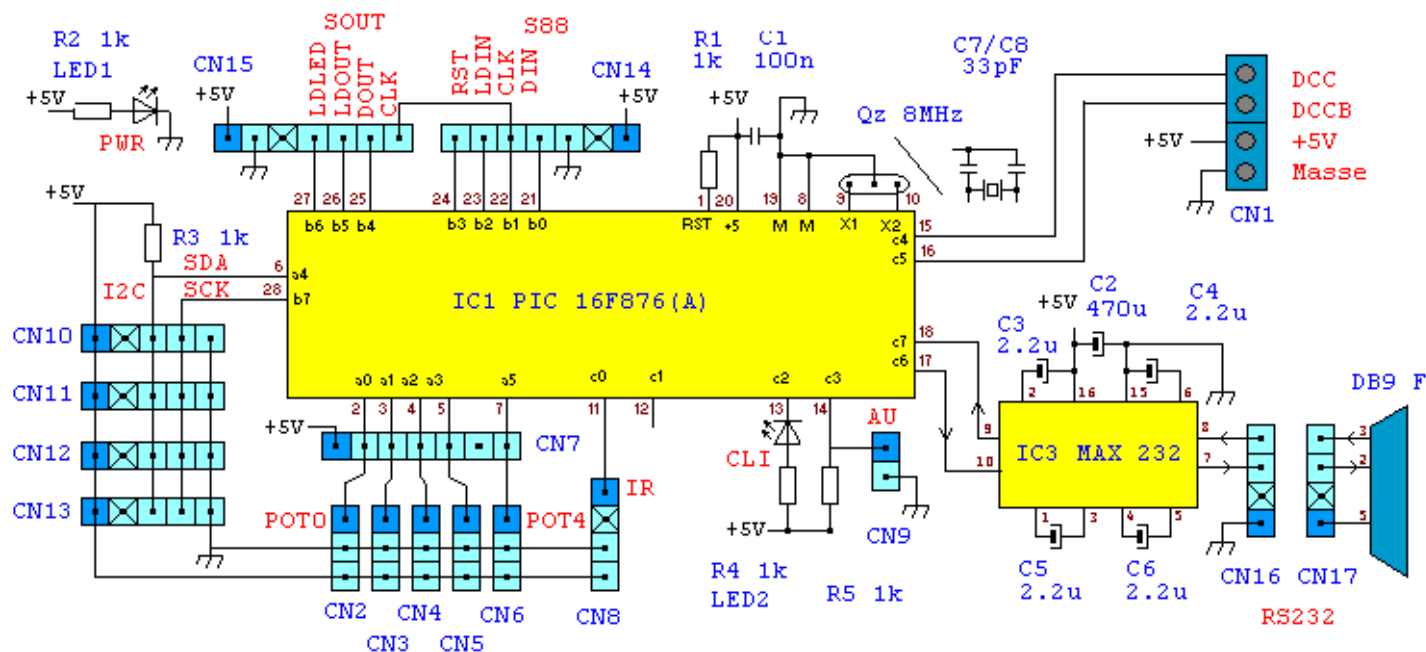
Au niveau du prix 40euros devraient suffire pour cette alimentation + booster (20euros pour l'alimentation 3A, 10euros pour le L6203, 1 euro pour chaque régulateur ...).





## 4.5. La centrale

### 4.5.A Le schéma



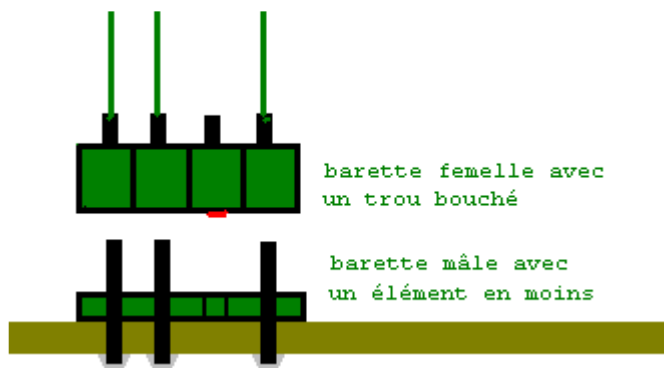
L'électronique de la centrale est composée uniquement de 2 circuits intégrés, Le microcontrôleur PIC16F876 véritable cœur du système et le MAX232 facultatif qui permet d'adapter les signaux série RS232 pour communiquer avec un ordinateur.

Le montage est alimenté en 5V par l'intermédiaire du connecteur CN1. Le condensateur C2 filtre la tension et permet d'absorber les pics de courants tandis que C1 élimine les composantes hautes fréquences. La LED 1 protégée par la résistance R32 indique la présence de cette tension.

Le chapitre suivant décrira en détail ce qu'est un microcontrôleur. Mais sachez a présent que c'est un petit ordinateur sur une puce qui exécute un programme. On retrouve autour du microcontrôleur les composants nécessaires a son fonctionnement comme la résistance de reset ainsi que son horloge constituée au choix par un résonateur céramique 8MHz ou un quartz de même valeur accompagnée de ces 2 condensateurs. Le choix s'effectuera en fonction des disponibilités. Le résonateur céramique est moins précis que le quartz (0.05% contre 0.002%) mais bien suffisant pour notre application. Il faudra impérativement utilisé la valeur prescrite car dans le cas contraire tous les timings seront faussés. Le clignotement de la LED 2 protégée par la résistance R4 indique un fonctionnement correct du microcontrôleur.

Afin de communiquer avec un ordinateur par l'intermédiaire d'une liaison série RS232 (ou d'un câble USB-RS232), il est nécessaire d'utiliser un MAX 232 afin d'adapter le niveau des signaux. En effet le « 0 » vaut 0V pour le UC et +12V en RS232 alors que le « 1 » vaut -12V en RS232. Afin de n'avoir qu'une seule alimentation le MAX232 fabrique le +12 et -12 (en fait +9v/-9v) à partir du 5V grâce a sa pompe de charge intégrée. Le 4 condensateurs entourant le circuit sont nécessaires au fonctionnement de cette pompe. Bien entendu le MAX 232 est facultatif si vous ne comptez pas raccorder votre montage à un ordinateur.

Le reste du montage se compose de nombreux connecteurs nécessaires au raccordement des différents modules et souris. La carte est conçue pour utiliser des connecteurs économiques réalisés avec de la barrette autosécable male. La croix symbolise un élément manquant qui joue le rôle de détrompeur. Vous ne pourrez donc pas vous tromper lors des branchements. Bien entendu libre à vous d'utiliser d'autres connecteurs suivants vos exigences.



Le connecteur CN1 reçoit l'alimentation et fournit le signal DCC de commande pour le booster. Il fournit également un signal complémentaire afin de se passer du transistor d'inversion pour la commande des boosters maison. Mais cela permet également de commander en PWM les 2 demis ponts du booster afin de tester une locomotive analogique (après avoir enlevé les locomotives analogiques du réseau)

Le connecteur CN7 est destiné à recevoir un réseau de résistance SIL de 100k (ou tout simplement 5 résistances à connecter entre les 5 entrées analogiques et le +5V. Le but est de mettre à +5V les entrées analogiques du UC en l'absence de souris potentiométriques. Les connecteurs CN2 à CN6 autorise de raccorder jusqu'à 5 souris potentiométriques à la centrale.

Les connecteurs I2C CN10 à CN13 servent quand à eux à connecter des souris analogique ou numériques ainsi que la console. Le bus I2C inventé par Philips permet au UC de piloter tout un tas de circuit I2C avec seulement 2 fils (en plus de l'alimentation). La résistance R3 maintient au repos un niveau haut sur la ligne des données du bus I2C.

Le connecteur AU CN9 est destiné à recevoir un bouton poussoir d'arrêt d'urgence afin d'éviter des tragédies en cas de besoin. Une pression du bouton met à la masse l'entrée correspondante du UC qui est normalement à l'état haut via la résistance R5. A chaque mise à l'état bas de cette entrée, la centrale passe du mode normale au mode arrêt d'urgence et inversement. L'arrêt d'urgence st également indiqué par le clignotement rapide de la LED 2. Il est également possible de déclencher/arrêter l'arrêt d'urgence avec la centrale, la télécommande ou le PC.

Le connecteur IR CN8 est destiné à recevoir un capteur infrarouge qui permet d'utiliser une télécommande de télévision Philips format RC5 pour piloter jusqu'à 3 locomotives.

Le connecteur CN14 est destiné à recevoir le bus S88 qui remonte vers la centrale l'état des entrés des modules de rétro signalisations S88. Ces modules sont par exemple connectés à des détecteurs d'occupation, des pédales de voies, des ILS ou encore à des boutons poussoirs ...

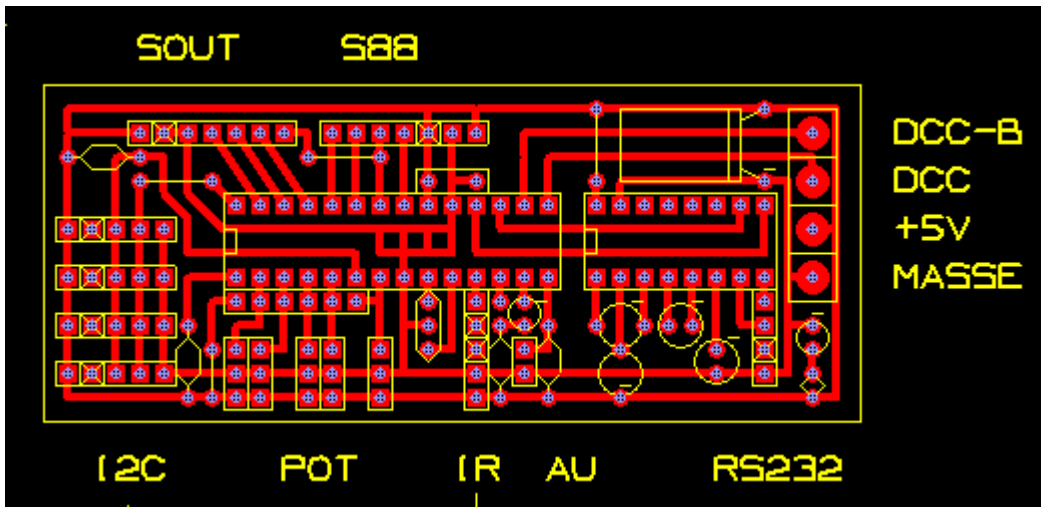
Le connecteur CN15 permet à la centrale de piloter les aiguillages, les LEDs et des sorties.

Enfin le connecteur CN16 permet de communiquer avec un ordinateur par l'interface RS232. Vous devrez réaliser un câble avec un connecteur DB9 d'un coté et un connecteur CN17 à barrette auto sécable femelle de l'autre.

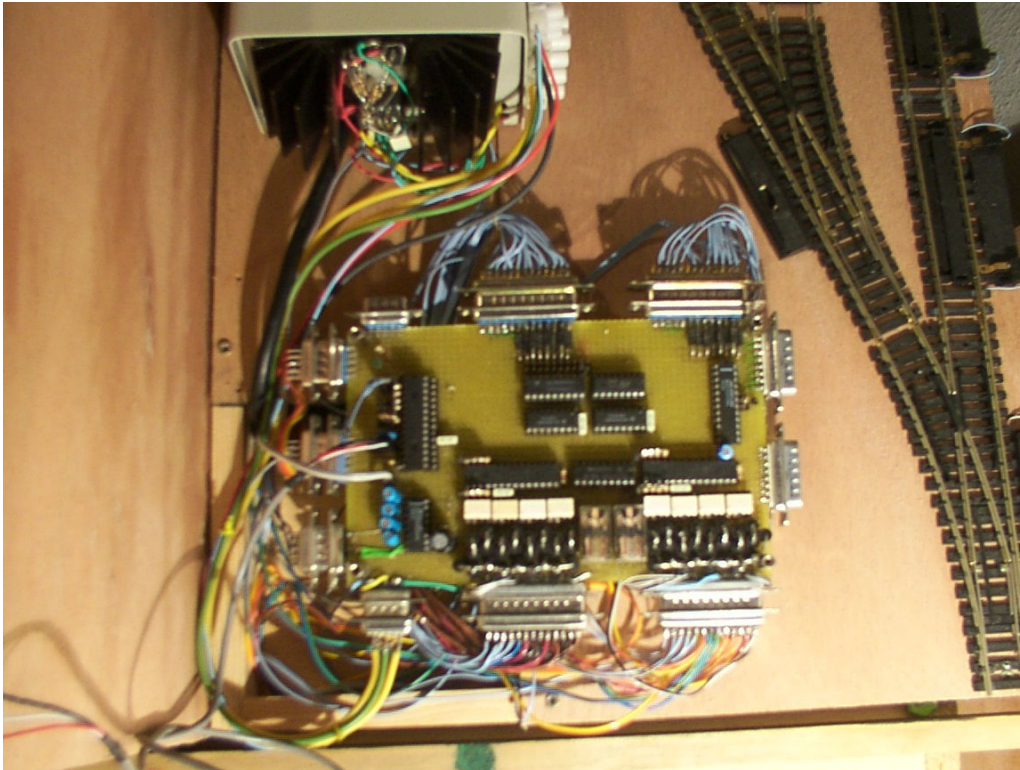
Voilà vous savez tout de l'électronique de la centrale ! Qui ose dore que l'électronique est compliquée ?

#### 4.5.B La réalisation

Pour la réalisation, vous pouvez utiliser le typon fournit (un fichier est disponible dans l'archive) ou alors simplement utiliser de la plaque percée et pastillée.



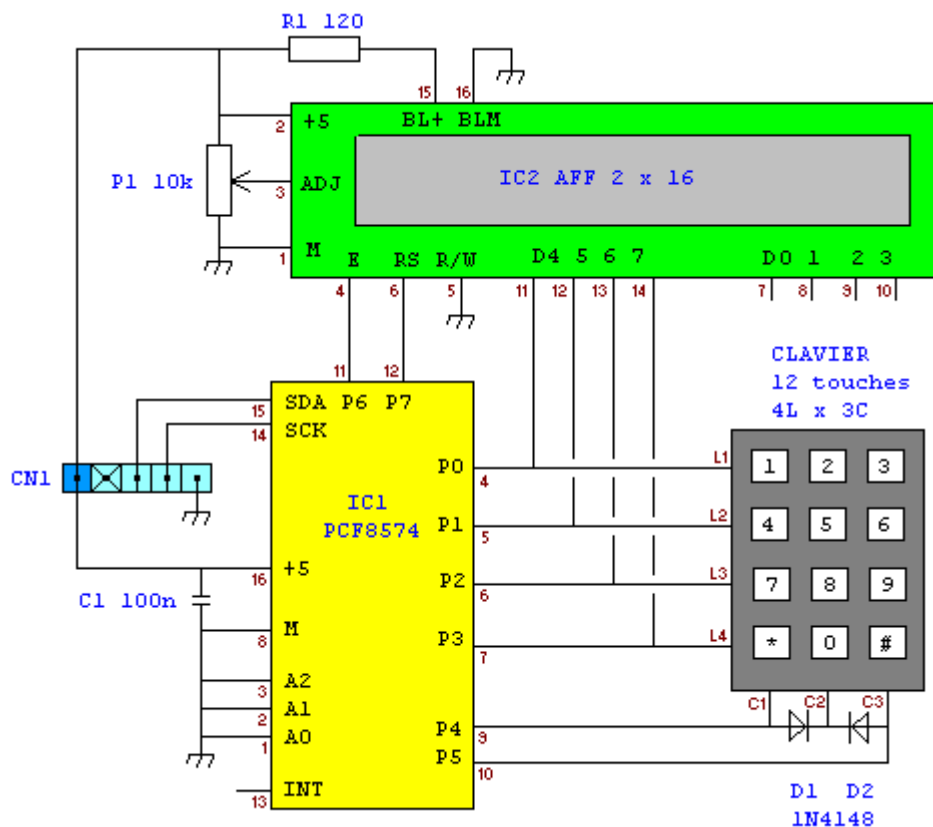
Vous pouvez également regrouper plusieurs modules sur la carte afin de réduire le nombre de module au total. Si les connecteurs ne vous conviennent pas, vous pouvez en changer. Dans mon cas s'agissant d'un prototype, j'ai utilisé de la plaque à trou et j'ai concentré toute l'électronique nécessaire au réseau sur cette carte. Au niveau des connecteurs j'ai opté pour des connecteurs DB9.



## 4.6. La console

La console avec son clavier et son écran permet de configurer et utiliser la centrale manuellement. Elle sert également de souris pour conduire 3 locomotives simultanément. Elle est reliée à la centrale par le bus I2C.

### 4.6.B. Le schéma



L'électronique de la console se compose uniquement d'un écran LCD, un clavier et un circuit I2C d'interfaçage.

Le circuit I2C est un PCF 8574 qui permet à la centrale de commander ou lire ses 8 entrées/sorties. Les ports P0, P1, P2, P3, P6 et P7 sont utilisés en sorties pour piloter l'afficheur et sélectionner les lignes du clavier. Les ports P4 et P5 sont quand a eux utilisés en entrées pour lire l'état des colonnes. L'adresse du circuit est fixée à 0, vous pourrez donc utiliser ce même type de circuit avec les adresses 1 à 7 pour des souris numériques. Une des caractéristiques intéressantes de l'I2C est qu'il est possible de connecter et déconnecter les circuits du bus sans couper l'alimentation. Vous pourrez donc débrancher la console pour la brancher à un autre endroit du bus I2C sans couper l'alimentation. La centrale détectera automatiquement la connexion et réinitialisera le LCD.

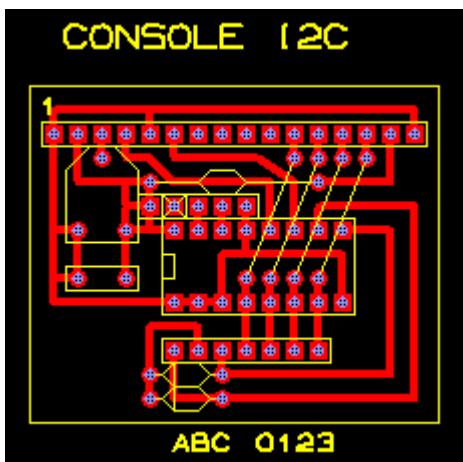
L'écran LCD est un modèle standard à commande parallèle de 2 lignes de 16 caractères utilisé en mode 4 bits pour réduire le nombre de signaux nécessaires. En mode 4 bits la communication se fait par les broches D4-D7 tandis que les broches D0-D3 ne sont pas utilisées. Coté control, La broches RS indique s'il s'agit d'une commande ou d'un caractère à afficher tandis qu'une impulsion sur la broche E valide l'échange. Les commandes et caractères étant codés sur 8 bits, 2 échanges sont nécessaires. L'ajustable permet de régler le contraste de l'écran. Afin de pou voir lire l'écran dans le noir ou sous faible éclairage, il est conseillé d'utiliser un modèle avec rétro-éclairage (backlight). Dans ce cas les LEDs de rétro éclairage sont alimentées en passant par la résistance R1. Vous pouvez donc jouer sur cette résistance pour modifier la puissance du rétro éclairage.

Le clavier est un modèle 12 touches que l'on trouve partout pour quelques euros. Un clavier 16 touches aurait été plus judicieux, mais la difficulté d'approvisionnement et les prix en on eu raison. Après l'utilisation de l'écran, il ne restait plus que 2 lignes sur le PCF8574. Il a donc été décidé d'utiliser aussi les 4 lignes de donnée

du LCD pour sélectionner les lignes du clavier. Ceci ne gêne nullement le LCD. Pour identifier la touche pressée, la centrale met tour à tour à la masse chaque ligne du clavier. Ainsi lorsqu'une touche est pressée, la masse apparaît sur la colonne correspondante. Les ports P4 et P5 étant configurés en entrées et maintenus à +5V par des résistances internes passent alors à la masse. P4 passe à la masse si une touche de la colonne 1 est pressée, P5 fait de même pour la colonne 3. Mais il ne restait alors plus d'entrée pour la colonne 2. Une petite astuce utilisant les 2 diodes à alors était imaginée, elle consiste à faire passer à la masse P4 et P5 simultanément lorsqu'une touche de la colonne 2 est pressée. Et comme une touche est pressée à la fois, cela ne pose pas le moindre problème.

#### 4.6.B. Réalisation

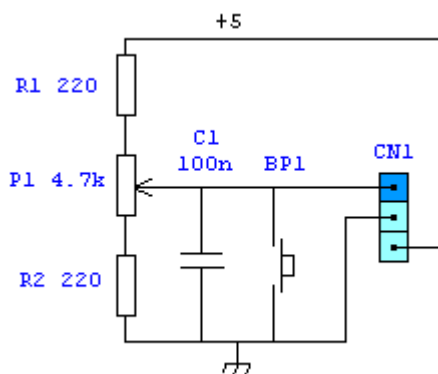
Comme d'habitude, vous pouvez utiliser la plaque à trou ou le typon suivant (utilisez celui de l'archive). Il est conseillé de mettre le montage dans un boîtier isolé, cela vous évitera de tout « cramer » si vous posez la console sur la voie.



## 4.7. Les souris

Free-DCC propose de nombreuses souris qui vont du simple potentiomètre à la télécommande infrarouge en passant par les souris I2C analogiques et numériques. En plus de cela la console peut aussi jouer le rôle de souris.

### 4.7.A. Souris potentiométriques

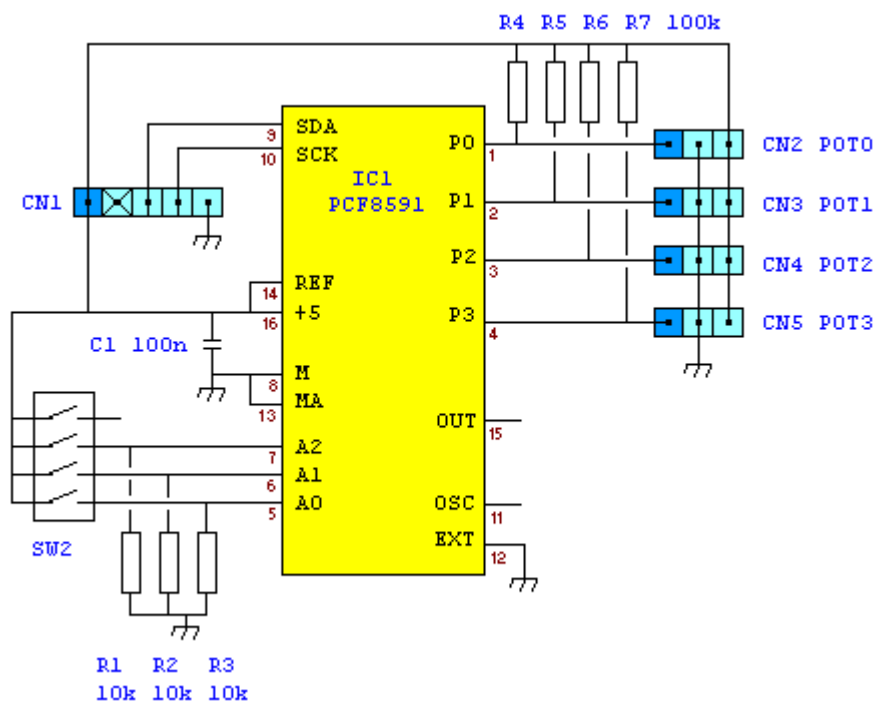


Cette souris très simple permet de faire varier la tension sur la première patte du connecteur CN1 grâce au potentiomètre P1. Le potentiomètre n'est pas directement câblé entre le +5 et la masse afin de pouvoir détecter l'absence de souris et l'appuie sur le bouton poussoir. En effet les entrées des convertisseurs analogiques/numérique de la centrale ou de la souris analogique sont tirées à +5V en l'absence de souris grâce aux résistances de pull-up. Inversement lors de l'appuie sur le bouton poussoir, la souris fournit 0V. Ce bouton est utilisé pour piloter la fonction 0. Le condensateur sert à absorber les sauts du curseur du potentiomètre, mais il ne peut rien contre les potentiomètres de trop mauvaise qualité. Il est primordial de choisir un potentiomètre linéaire et non un logarithmique afin d'avoir une vitesse proportionnelle à sa rotation. Pour information ces potentiomètres portent la lettre A à côté de la valeur, soit « 4.7k A » dans notre cas. L'inversion du sens de marche est automatique lorsque le potentiomètre passe son point milieu : Dans un sens la locomotive avance, tandis que dans l'autre elle recule.

Vu l'extrême simplicité du montage, il n'y a pas besoin de circuit imprimé.

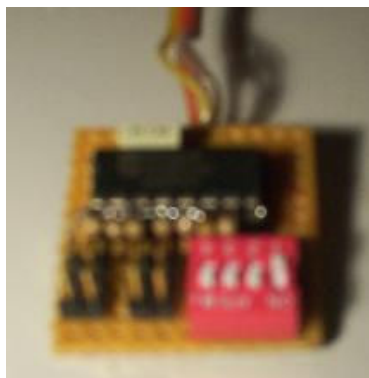
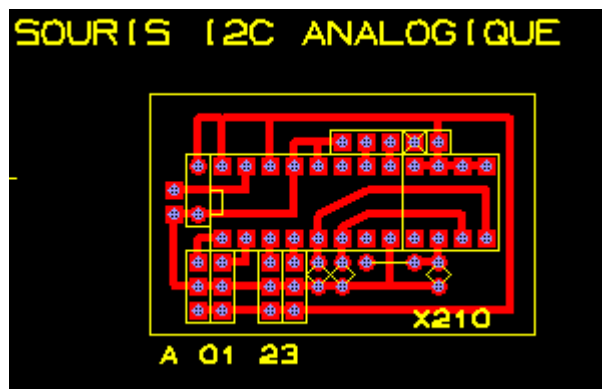


## 4.7.B Souris I2C analogique

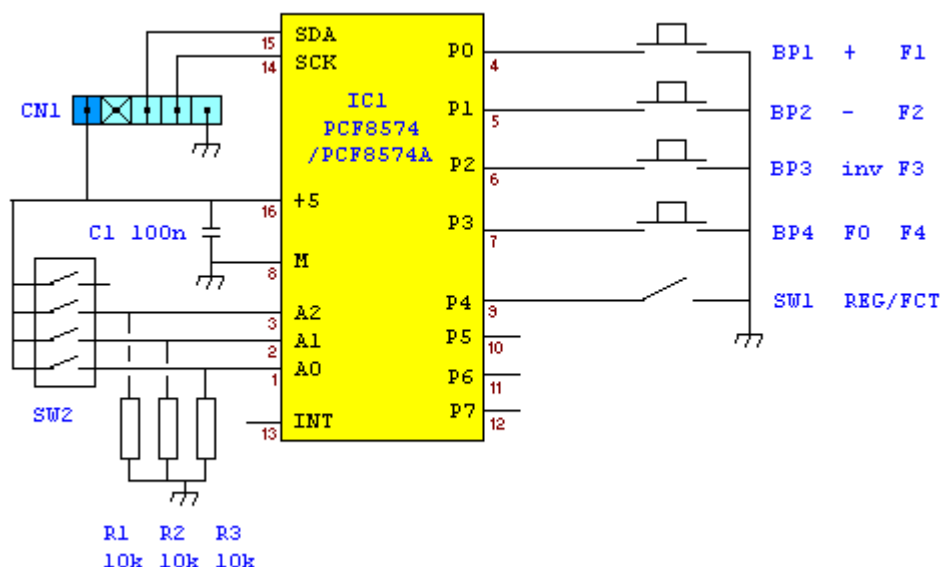


Cette souris ou plutôt cet adaptateur permet de brancher jusqu'à 4 souris potentiométriques. L'adaptateur se connecte ensuite sur le bus I2C. Il peut y avoir 8 sur le bus I2C. Pour cela il suffit de sélectionner une adresse 0-7 avec les minis interrupteurs SW2. Pour réaliser une souris indépendante, il suffit de rajouter cet adaptateur dans le boîtier d'une souris potentiométriques.

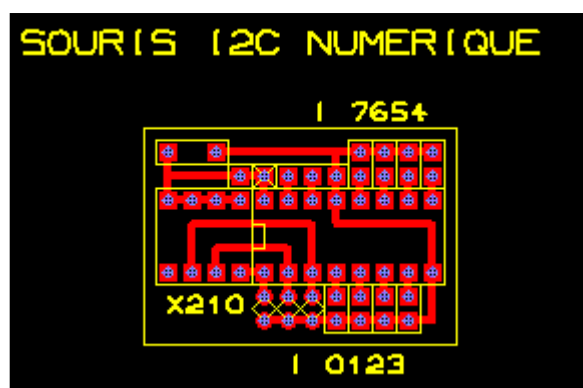
Pour la réalisation, vous pouvez utiliser de la plaque à trous ou le typon. Attention, les résistances de pull-up manquent sur le typon, vous pouvez les rajouter en les soudant côté soudure si vous voulez utiliser le montage comme adaptateur. Dans le cas où une souris est connectée en permanence, il n'y a aucun avantage à souder ces résistances.



## 4.7.C Souris I2C numérique

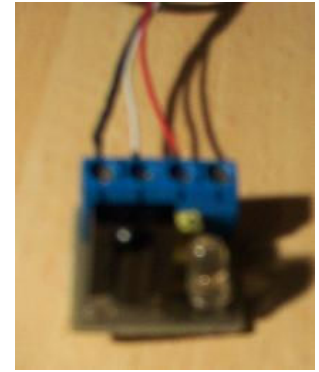
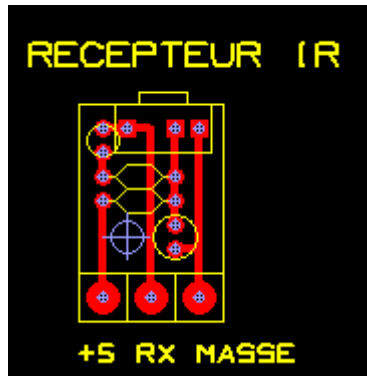
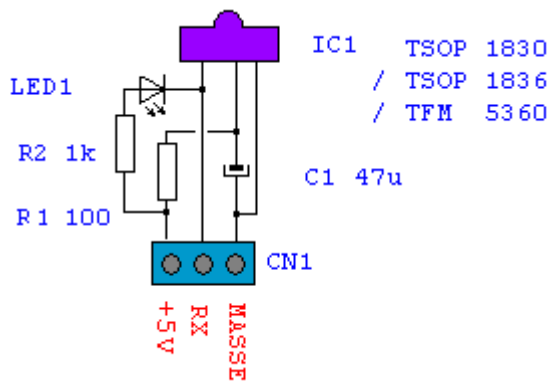


Cette souris est faite pour ceux qui préfèrent conduire leurs locomotives avec des boutons plutôt qu'avec un potentiomètre. On retrouve un circuit I2C d'entrées/sortie PCF8574 ou PCF8574A auquel on connecte 4 boutons poussoirs et un interrupteur. Cet interrupteur permet de choisir le mode des 4 autres boutons. Lorsqu'il est ouvert les boutons poussoirs commandent l'accélération, le freinage, l'inversion du sens et la fonction 0. Dans le cas contraire, ils commandent les fonctions 1 à 4. Au repos, les entrées sont tirées au niveau haut par des résistances de pull-up internes, les boutons quand à eux mettent les entrées à la masse. SW2 et R1-3 permettent de choisir une des 8 adresses possibles. Il est donc possible d'utiliser un maximum de 15 souris numériques. 7 avec le PCF8574 car l'adresse 0 est réservée à la console et 8 pour le PCF8574A. Coté réalisation, vous pouvez utiliser de la plaque à trous ou le typon.





#### 4.7.D. Souris télécommande IR



Cette souris ou plutôt ce récepteur infra rouge permet d'utiliser une télécommande infra-rouge de télévision comme souris. Cette télécommande peut être utilisée pour conduire jusqu'à 3 locomotives. Pour chacune des 3 locomotives, il est possible de choisir :

- 2 touches pour accélérer
- 2 touches pour freiner
- 1 touche pour inverser le sens
- 1 touche pour la fonction 0

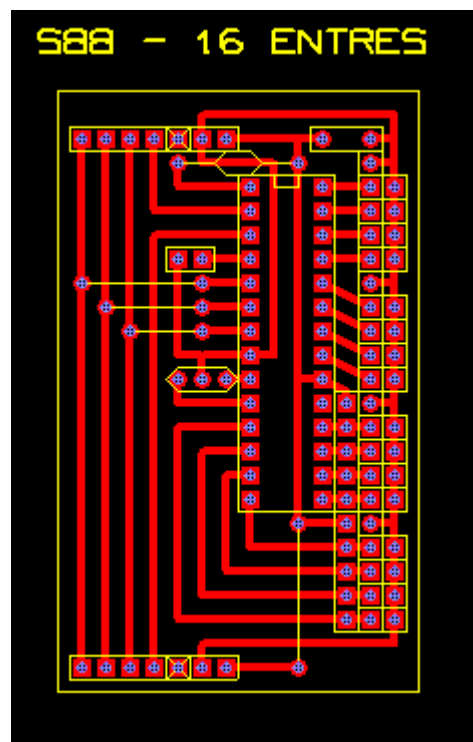
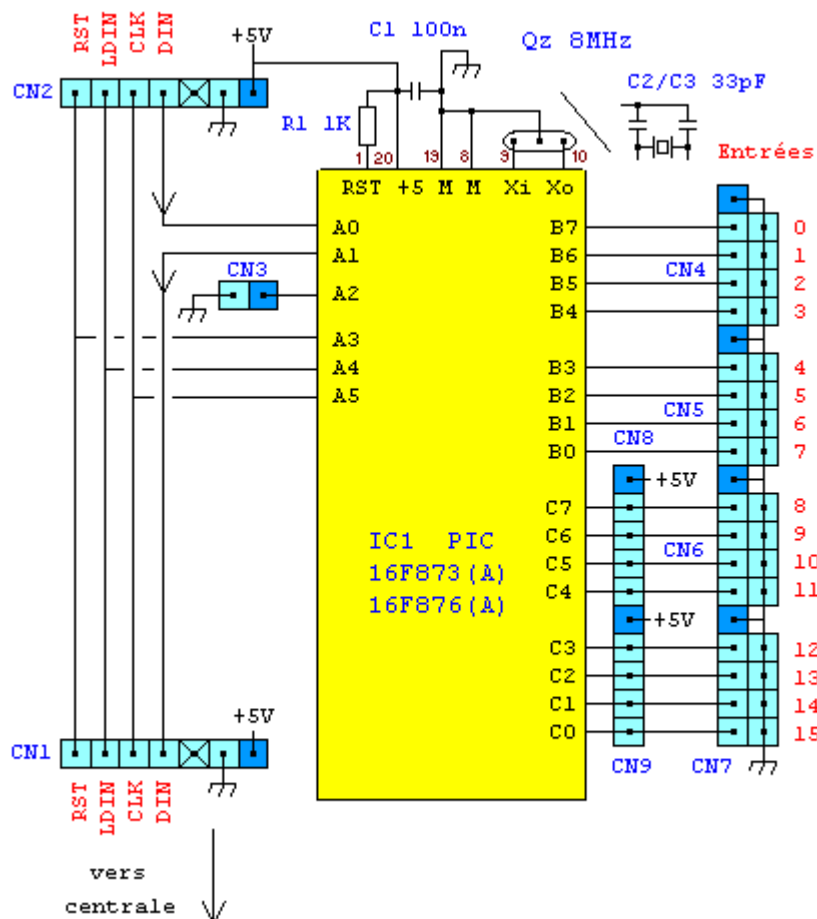
Une touche supplémentaire est utilisable pour l'arrêt d'urgence.

Seules les télécommandes au format RC5 (Philips) sont supportées. Si vous n'en avez pas, on en trouve à un euro dans les bazars.

## 4.8. Les modules d'entrée S88 et les détecteurs de courants

Cette partie décrit les modules S88 à raccorder sur le bus S88. Chaque module accepte 16 entrées qui peuvent être des pédales de voies / ILS / bouton poussoirs / interrupteurs / détecteur de courant. Des détecteur de courant simple à réaliser sont également décrits.

### 4.8.A. Les modules d'entrées S88



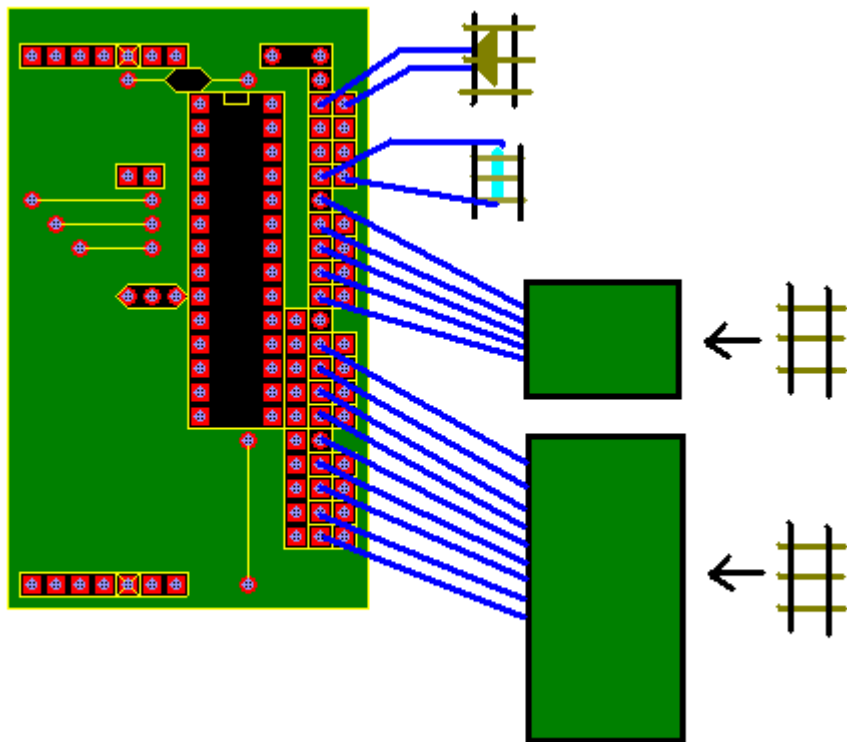
Ce module permet de remonter l'état de 16 entrées vers la centrale. Free-DCC supporte jusqu'à 4 modules ce qui fait 64 entrées au maximum. Le bus S88 étant standard, vous pourrez également utiliser des modules du commerce. Normalement ces modules sont composés de registres à décalage 4014 et mémoire 4044, mais le montage est plus complexe car il faut 6 circuits pour 16 entrées. Le montage à UC est légèrement plus cher mais vraiment plus facile à réaliser. Il faudra néanmoins programmer le UC avec le programme S88\_16.HEX

Les entrées sont maintenues au niveau haut par les résistances de rappel. Elles sont internes pour le PORTB et à connecter sur les connecteurs CN8 et 9 pour le PORTC. Une entrée est considérée comme active lorsqu'elle est reliée à la masse. Le microcontrôleur mémorise les impulsions donc la centrale ne peut pas rater une entrée. Cela est indispensable pour les ILS car la centrale interroge seulement une dizaine de fois par seconde les entrées. La mémorisation elle scrute les entrées 20000 fois par seconde. Lorsque la centrale à chargée l'état des entrées dans le registre à décalage virtuel avec LDIN, elle demande d'effacer la mémorisation avec RST. Ensuite elle récupère les données par DIN à chaque front de l'horloge CLK.

Le diagramme suivant montre les signaux du protocole S88

RST	#						
LDIN	#						
CLK	###	#	#	#	#	#	
Data	0	1	2	15	16	31	

Les connecteurs CN4 à CN7 permettent de connecter les entrées une à une ou 4 à 4 ou 8 par 8 comme l'explique la figure suivante avec une pédale de voie, un ILS, un détecteur de présence à 4 voies et un autre à 8 :



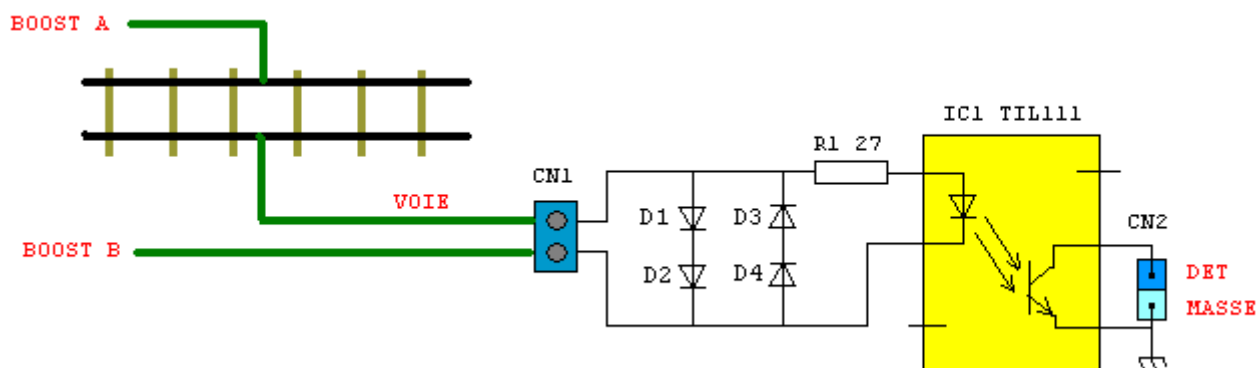
Le montage se résume donc à un UC entourée de quelques connecteurs, du circuit de reset et d'horloge. Pour l'horloge vous pouvez utiliser un résonateur céramique, un quartz avec ses 2 condensateurs. Si la centrale est sur la même carte, vous pouvez connecter la sortie de l'oscillateur de la centrale X2 sur l'entrée Xi du UC S88. S'il y a d'autres modules, chaînez les Xi et Xo.

La sortie A2 passe à 1 lorsque C0 est active et revient à 0 lorsque C1 est active. Cette mémoire peut servir à piloter un relais standard pour gérer une boucle de retournement.

La réalisation est extrêmement aisée avec une plaque à trou ou le typon fournit. Si vous utilisez ces entrées avec des capteurs de courant, vous pouvez monter ces capteurs sur la même carte.

## 4.8.B. Les détecteurs de courant

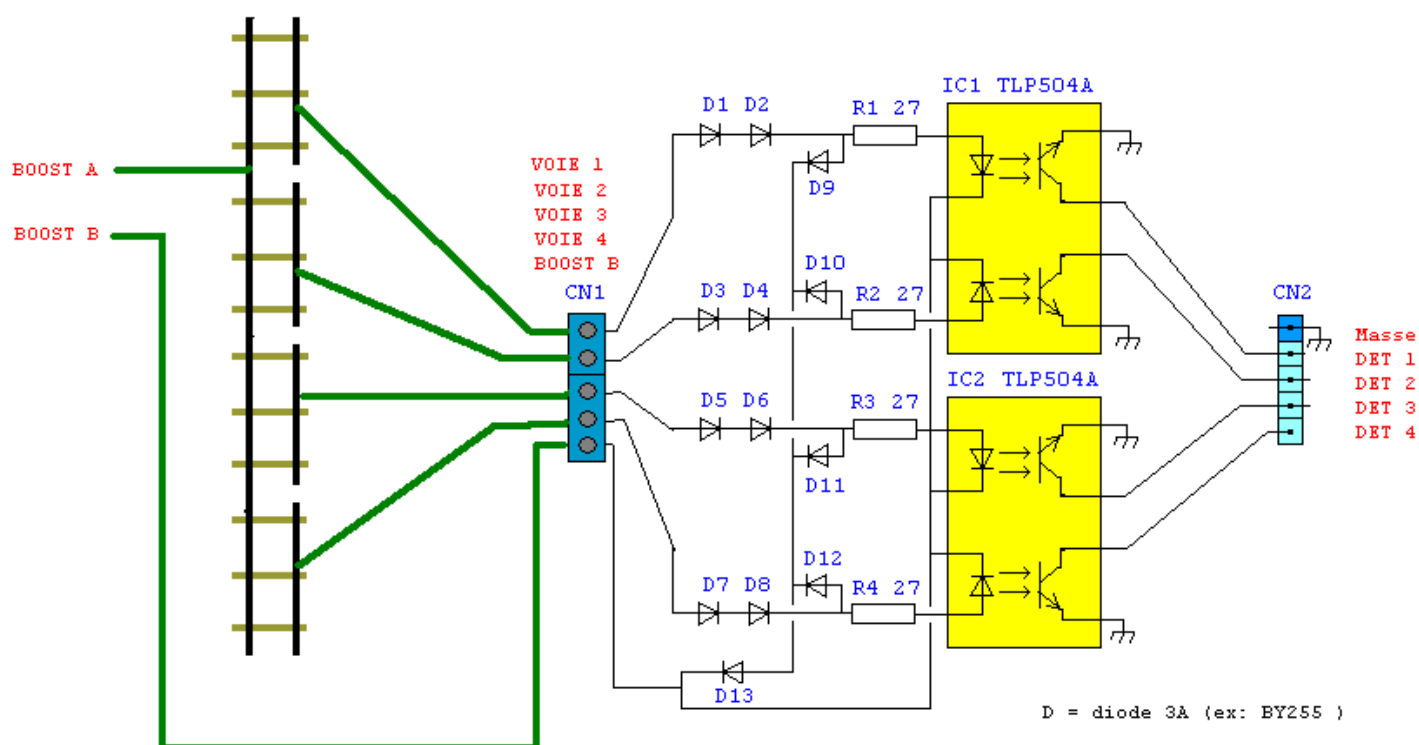
Les détecteur de courant permettent de détecter l'occupation des voies par les trains en toute discrétion contrairement aux pédales de voies ou ILS. En fait ils détectent les décodeurs des locomotives, les moteurs ou les éclairages des voitures. Si vous souhaitez également détecter les wagons, il suffit de rajouter une résistance de 4.7 K sur les essieux ou graphiter ces derniers. Les détecteurs sont particulièrement bien adaptés pour la détection sur des sections de voies ou cantons contrairement aux pédales de voies ou ILS qui ont un rôle de détection plus ponctuelle. Il est également possible d'utiliser de courtes sections pour des détections ponctuelles. La détection de courant est quasiment indispensable pour gérer la signalisation. Avec un prix de revient de 2 euros par section de détection il ne faut pas vous en priver.



Le schéma précédent présente un détecteur pour une voie.

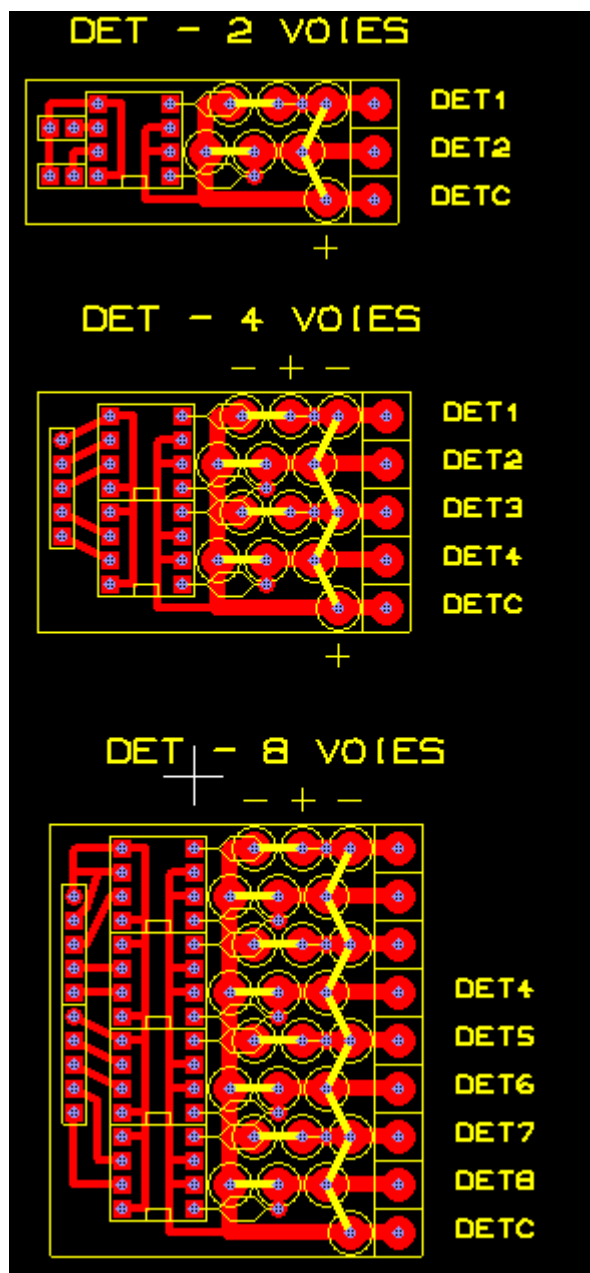
Le booster délivre alternativement +15V et -15V. Ceci se traduit lorsqu'une locomotive est présente par un courant passant par D1 et D2 dans un sens puis D3 et D4 dans l'autre. Dans le cas contraire aucun courant ne passe. Lorsque du courant passe dans D1 et D2, ceci provoque une chute de tension de 1.2V au minimum aux bornes des diodes ce qui alimente la LED de l'optocoupleur par l'intermédiaire de la résistance R1 de faible valeur. La diode éclaire le phototransistor de l'optocoupleur ce qui provoque la mise à la masse de l'entrée de détection. Cette mise à la masse sera ensuite mémorisée et remontée à la centrale par un module S88. Bien entendu les diodes doivent supporter le courant maximal du booster soit 3A dans notre cas comme les BY255.

Le schéma suivant présente un détecteur pour 4 voies. Ceci permet d'économiser une diode par voie et d'utiliser des optocoupleurs multiples. Mais le fonctionnement est identique.



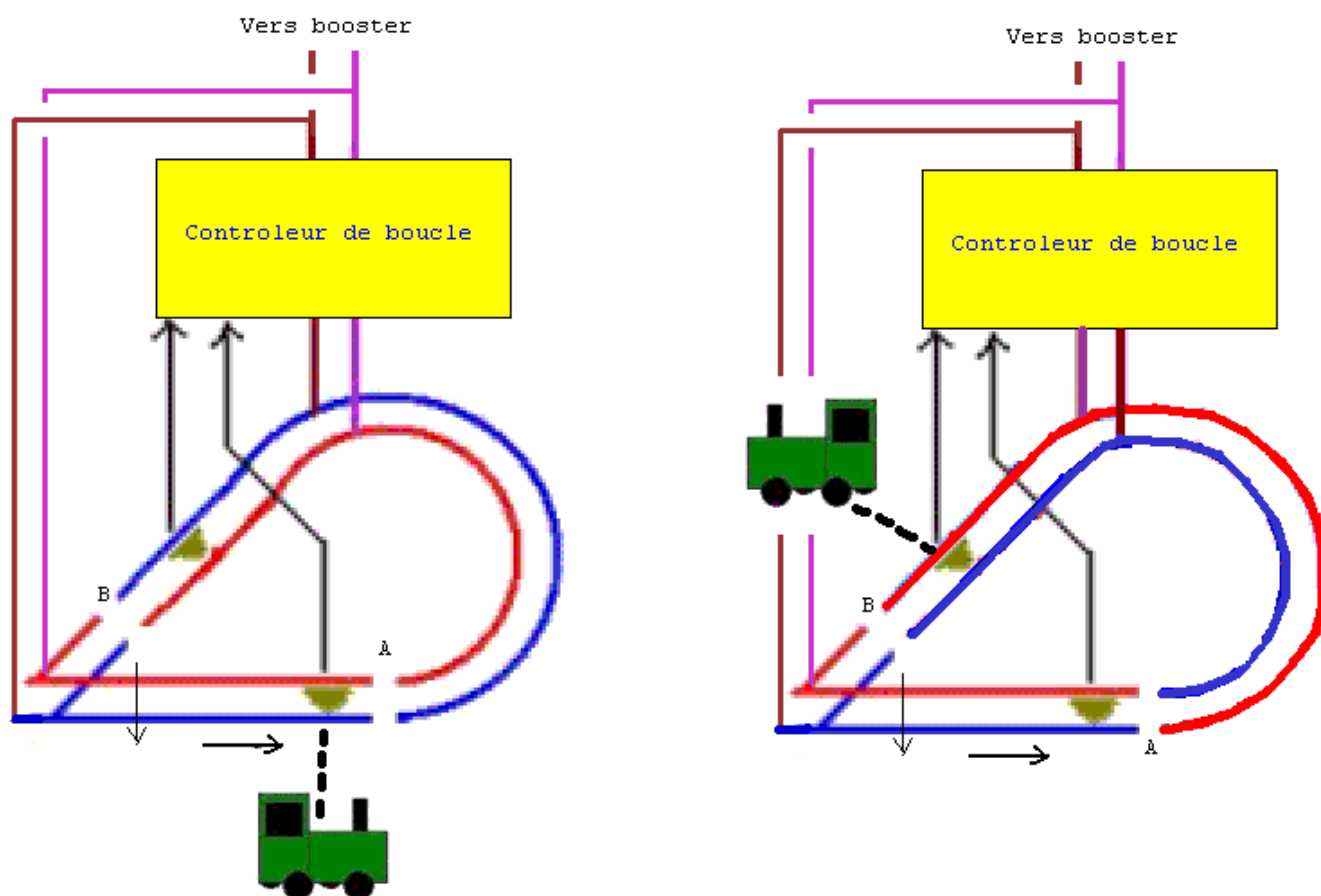
Vous pouvez extrapoler le schéma d'un détecteur à 8 voies ou à un nombre quelconque de voie.

Au niveau de la réalisation, vous pouvez utiliser de la plaque à trou ou les typons suivants.  
Avec les typons, les résistances sont à souder coté cuivre.



#### 4.8.C. Les boucles de retournement

Les boucles de retournement présent sur de nombreux réseaux souvent en coulisse contribuent au réalisme des circulations en retournant les rames. En 3 rails de un tel dispositif ne pose aucun problème car les plots centraux sont à relier à une sortie du booster et les rails à l'autre. Par contre, il n'en va pas de même en 2 rails. En effet à l'aide de la figure suivante en fonction de la polarité de la boucle, on provoque un court circuit soit au point A, soit au point B. La solution est d'appliquer une polarité à la boucle compatible avec le passage d'une locomotive au point A lorsqu'elle actionne la pédale A (première figure) et de l'inverser lorsque elle actionne la pédale B afin d'autoriser son passage en B.



Bien entendu avec cette solution la boucle ne peut être parcourut que dans un sens. Pour un parcourt dans les 2 sens, il suffit d'utiliser 2 pédales positionnées de chaque coté des points de basculement. Si vous jugez les pédales pas assez fiables, vous pouvez toujours les remplacer par des capteurs de courant.

Le contrôleur de boucle se compose d'un simple relais 2RT qui permet d'inverser la polarité de la boucle. Ce relais est soit un modèle bistable qui change de position et la maintient suivant les impulsions venant des pédales ou un modèle standard qu'il faut associer à une mémoire. La mémorisation des modules S88 précédent peut être utilisée à cette fin.

Le schéma suivant présente ces 2 cas.

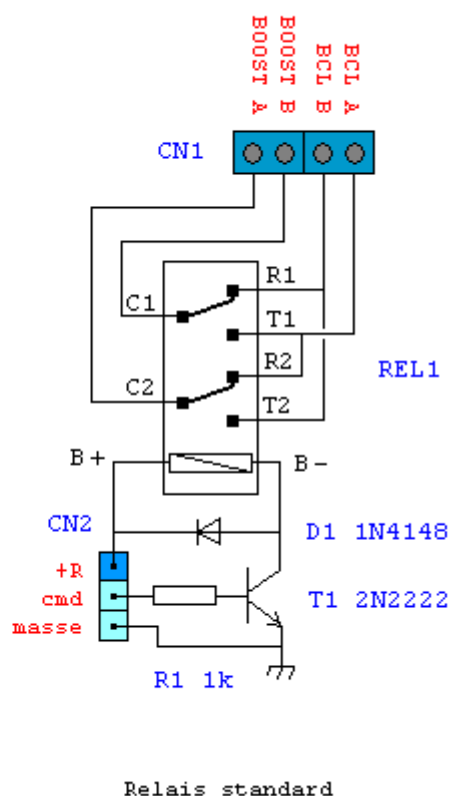
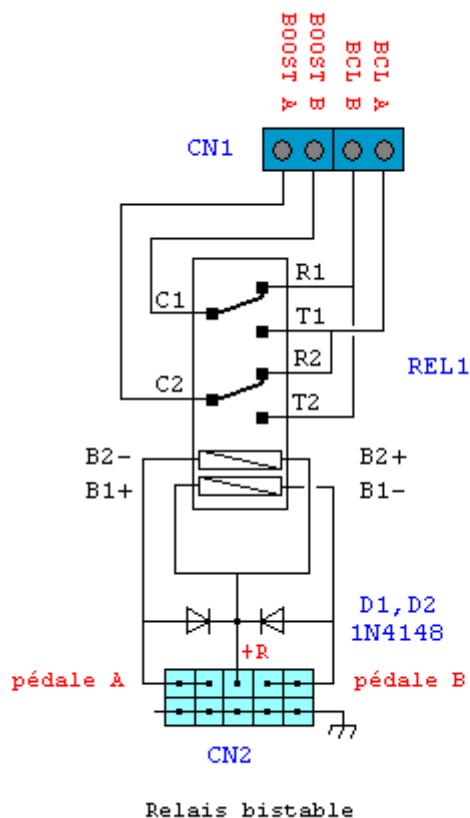
Le premier utilise un relais bistable qui bascule les contacts :

- C et T lorsque la bobine 1 est activée par la ou les pédales B
- C et R lorsque la bobine 2 est activée par la ou les pédales A

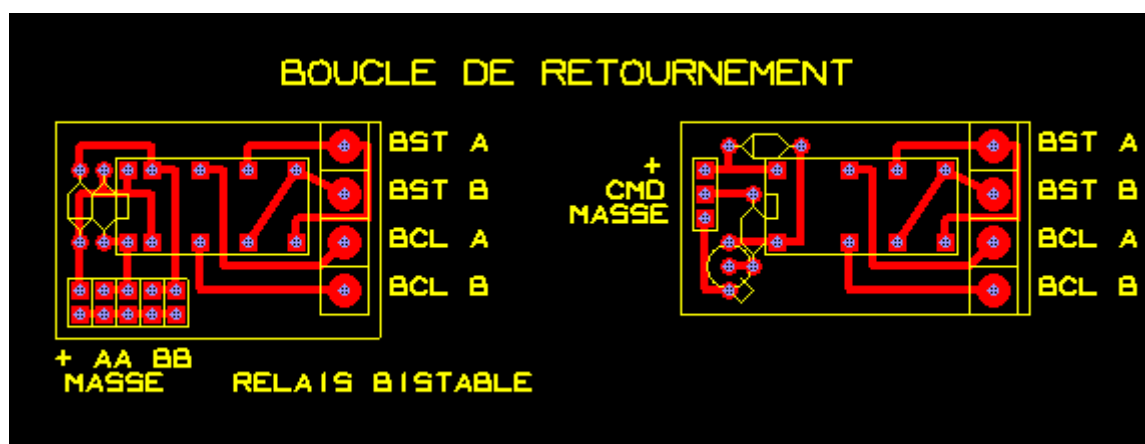
Les diodes éliminent la surtension qui apparaît lorsque l'on coupe l'alimentation d'une bobine.

Le second schéma met en œuvre un relais standard commandé par la broche cmd du connecteur CN2. Le transistor amplifie le signale de commande afin de piloter le relais

Dans les 2 cas la tension +R alimente les bobines des relais. Elle est à raccorder au +5V ou +15V en fonction du relais utilisé. Les contacts des relais devront supporter 3A,mais si vous n'en trouvez pas des modèle 1A peuvent faire l'affaire.



La réalisation peut se faire sans plaque, sur une plaque à trou ou alors en utilisant un circuit imprimé en s'inspirant du typon suivant. Vu la différence de brochage entre les différents relais, il faudra peut être modifier les typons.



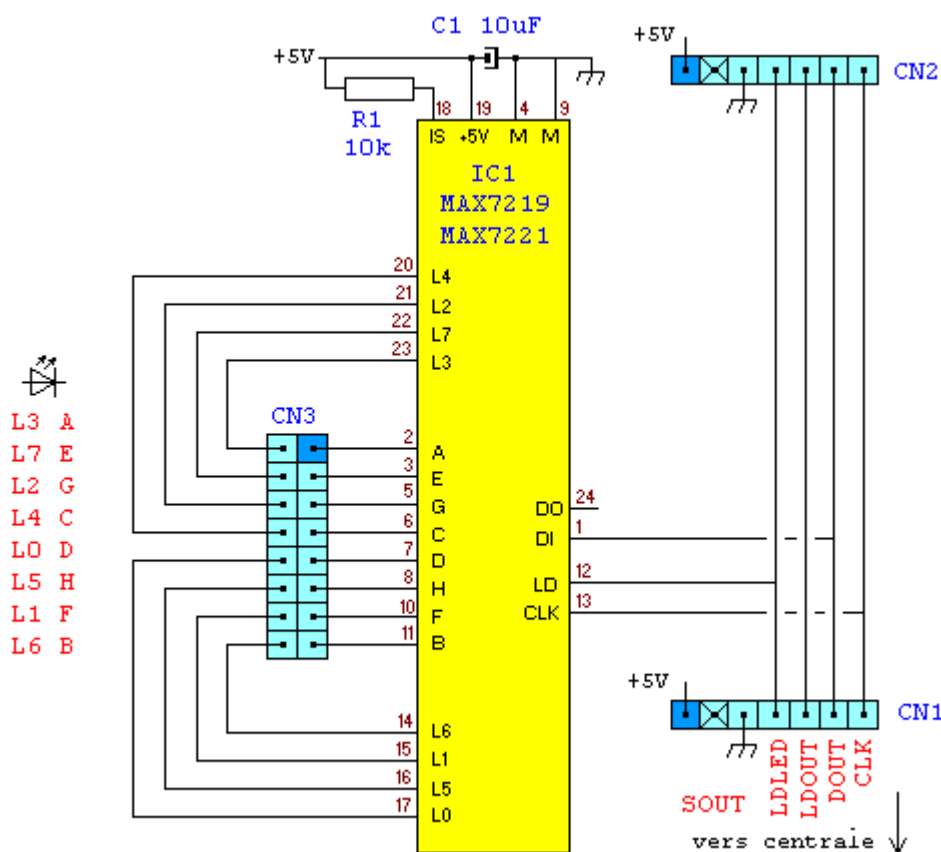
## 4.9. Les modules de sorties / LEDs / aiguillages

Ces modules permettent de contrôler des leds pour la signalisation, les aiguillages ou encore diverses sorties. Il est possible de raccorder sur le bus SOUT :

- 1 module de 64 LEDs
- 1 module de 32 aiguillages
- 2 modules de 8 sorties

L'ordre de connexion à une grande importance, ainsi si vous utilisez le module LED, il est impératif de le raccorder en premier à la centrale, ensuite vient le module d'aiguillage puis les éventuels modules de sortie.

### 4.9.A. Le module 64 LEDs



Ce module permet de contrôler matriciellement 64 LEDs.

Il suffit de les brancher sans aucune résistance à partir du connecteur 3 en mettant

- les anodes (+) des LEDs sur les pattes L0 à L7
- les cathode (-) sur les pattes A-H

Le numéro de la led pour Free-DCC se calcule en additionnant le numéro de son anode et de sa cathode avec les poids suivants :

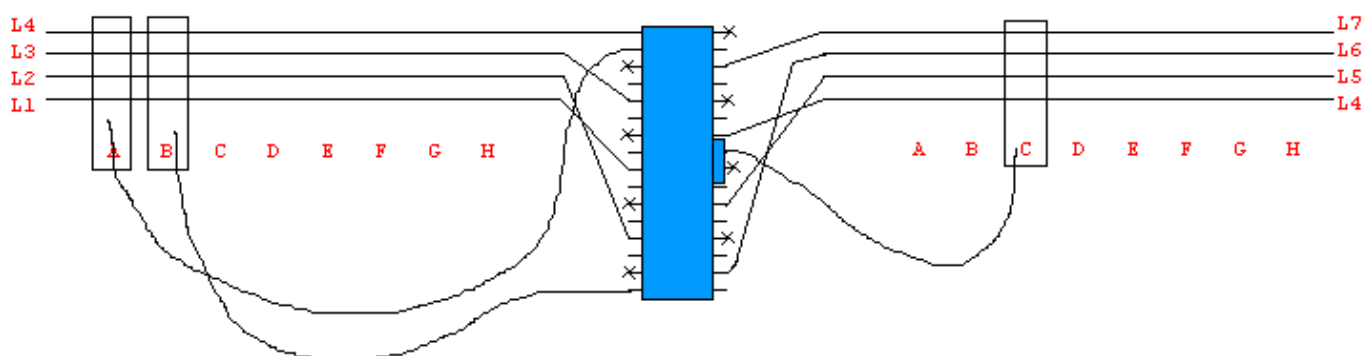
L0 = 0	L1 = 1	L2 = 2	L3 = 3	L4 = 4	L5 = 5	L6 = 6	L7 = 7
A = 0	B = 8	C = 16	D = 24	E = 32	F = 40	G = 48	H = 56

Par exemple une LED connectée entre E et L3 à comme numéro  $32+3=35$

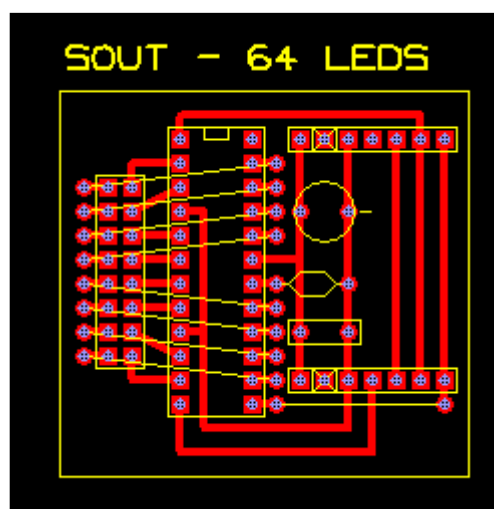


Le schéma électrique se compose principalement du driver de LED IC1 qui peut être un MAX7219 ou 7221. Le circuit est alimenté en 5V par le bus SOUT et cette tension est filtrée par le condensateur C1 beaucoup plus gros que d'habitude car le driver perturbe beaucoup l'alimentation lorsqu'il passe d'un groupe de 8 LEDs à un autre. Le module s'interface à la centrale par le bus SOUT via le connecteurs CN1. Le module suivant qui doit être le module aiguillage s'il est utilisé ou alors un module de sortie s'interface sur le connecteur CN2. Le connecteur CN3 permet quand à lui d'y connecter les LEDs. Il reçoit les commandes et données sur son entrée DI au rythme de l'horloge CLK, puis le transfert est validé avec la broche LD. La résistance R1 permet de fixer le courant maximal dans les diodes. Par la suite la centrale à la possibilité de diminuer ce courant. Ceci peut être utile pour voir la signalisation en plein jour et éviter qu'elle ne se transforme en lampadaire dans l'obscurité. Le courant traversant toutes les LED sera identique, si vous souhaitez le diminuer pour une LED particulière, il faudra lui ajouter une résistance.

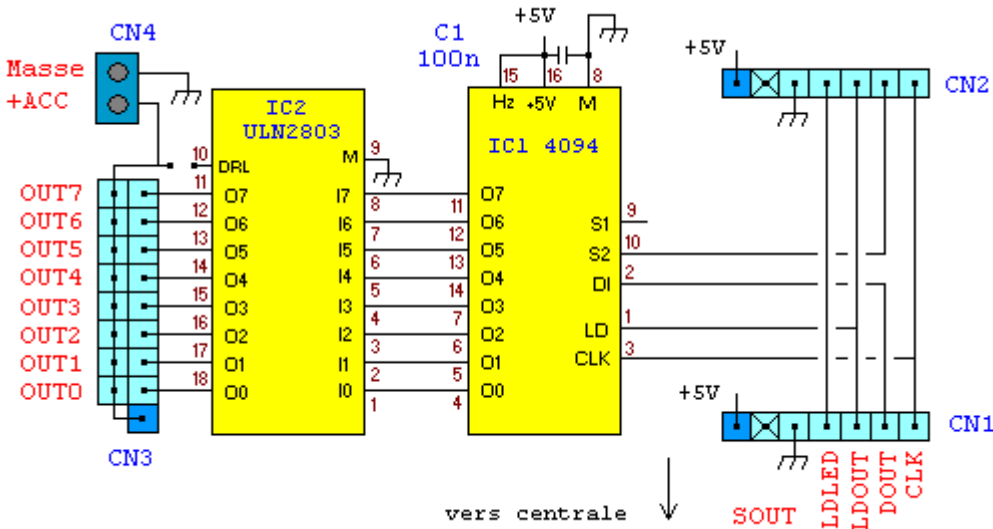
Bien entendu, vous ne pouvez pas relier les 64 LEDs une à une sur le pauvre connecteur CN3 à 16 contacts. Pour ce faire vous pouvez réaliser une carte de répartition et brancher les leds une à une ou alors les souder directement sur des fils. Une solution simple est de sertir par le milieu un câble nappe à 16 conducteur dans un connecteur HE10 2\*8 femelle, puis de désolidariser les fils et enfin créer 2 bus de 4 leds ou vous brancherez les anodes (+) des LEDs avec pour les anodes 8 sélections possible par bus. Ce qui permet par exemple de connecter 16 signaux à 4 leds. La figure suivante explicite cette solution avec uniquement 3 signaux pour la lisibilité



Coté réalisation vous pouvez utiliser de la plaque pastillée percée ou le typon suivant :



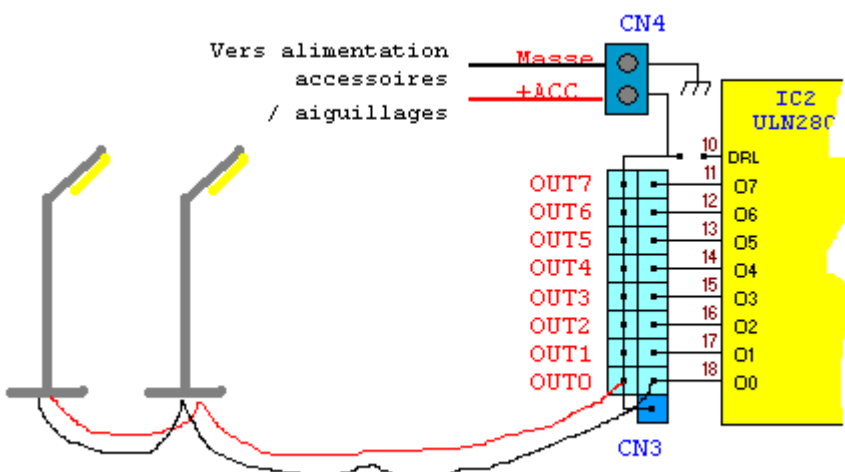
## 4.9.B. Le module 8 sorties



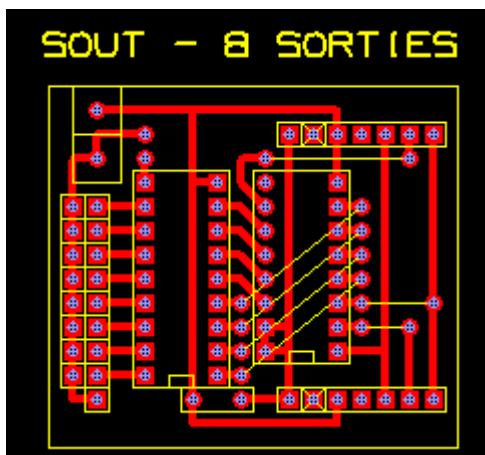
Ce module de sortie permet de contrôler 8 sorties de 500mA chacune. Il est composé du registre à décalage IC1 qui reçoit les données en provenance de la centrale par l'intermédiaire de sa broche DI au rythme de l'horloge CLK. Les données à destination des autres modules ressortent par la broche S2. Lorsque toutes les données sont correctement décalées alors la centrale les charge dans les registres de sorties grâce à la patte LD. A ce moment là les nouvelles données de sortie sont disponibles en sortie d'IC1 et contrôlent le réseau de transistor Darlington qui active ou pas les sorties. Quelques connecteurs ainsi que le condensateur de découplage complètent le montage.

Pour la centrale, le numéro de la sortie est égale à celui indiqué à proximité du connecteur plus 8 fois le nombre de 4094 qui précèdent le module. Par exemple s'il y a sur le bus un module d'aiguillage (qui possède 2 4094) puis un module de sortie (avec un 4094), et enfin notre module de sortie, alors le numéro de la sortie OUT 5 sera  $3 \times 8 + 5 = 29$

Lorsqu'une sortie est activée, l'ULN283 la connecte à la masse alors que dans le cas contraire elle est en l'air. Il faut donc brancher la charge entre le + d'une alimentation et cette sortie (Le - de l'alimentation étant relié à la masse du montage). Dans la plupart des cas, les charges seront alimentées par l'alimentation accessoire de 15V également utilisée par les aiguillages. Dans ce cas, il suffit de relier cette alimentation sur CN4 et brancher les charges sur CN4. Si vous utilisez des charges inductives, penser à rajouter une diode de roue libre en parallèle afin d'éviter les pics de tensions qui ne manquent pas de se produire lorsque l'on coupe la tension à une bobine. Si toutes les charges sont alimentées par la même alimentation alors il est possible d'utiliser les DRL du circuit en connectant la patte 10 à l'alimentation. La figure suivante montre comment relier un ensemble de 2 lampadaires (qui s'allumeront en même temps) sur la sortie OUT0



Coté réalisation vous pouvez utiliser de la plaque pastillée percée ou le typon suivant :

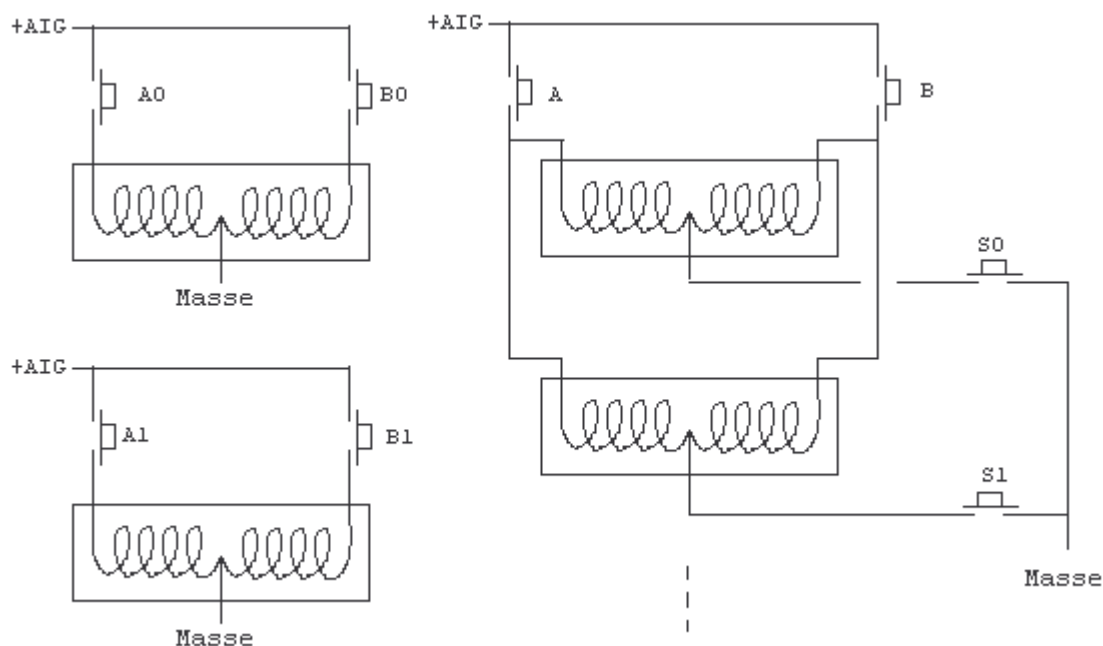


### 4.9.C. Le module 32 aiguillages

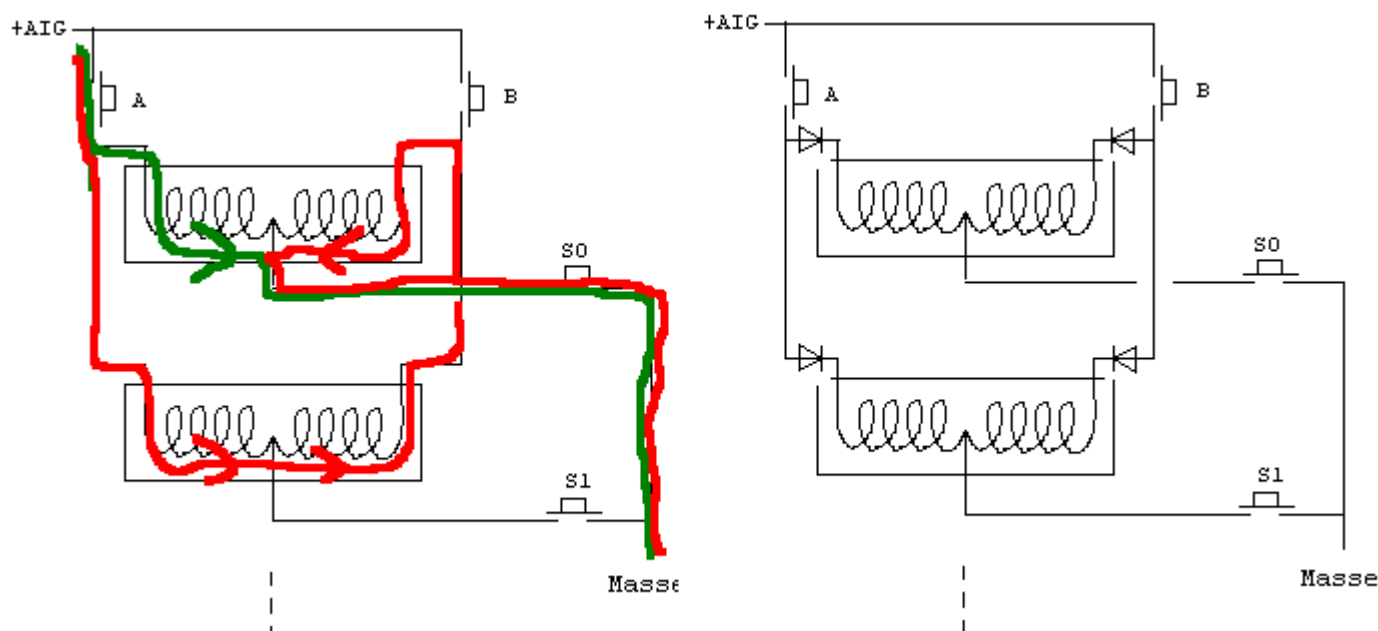
Ce module permet de commander jusqu'à 32 aiguillages. Il a été conçu pour des aiguillages à solénoïdes, mais en rajoutant quelques relais, il est également capable de commander des aiguillages à mouvement lents.

Avant de décrire le schéma revenons sur la commande des aiguillages.

Un aiguillage à solénoïde peut se commander simplement avec 2 interrupteurs ou plutôt boutons poussoirs si on tient à la vie des bobine. Dans ce mode chaque bouton contrôle une bobine ce qui positionne l'aiguillage en direct ou en dévié.



Le problème est qu'il nous faudrait 64 interrupteurs électroniques pour commander nos 32 aiguillages. Une solution intermédiaire serait d'utiliser 2 interrupteurs selon la position et un autre pour sélectionner chaque aiguillage. Ceci nous amènerait donc à 34 interrupteurs (2 avec le + et 32 avec la masse).



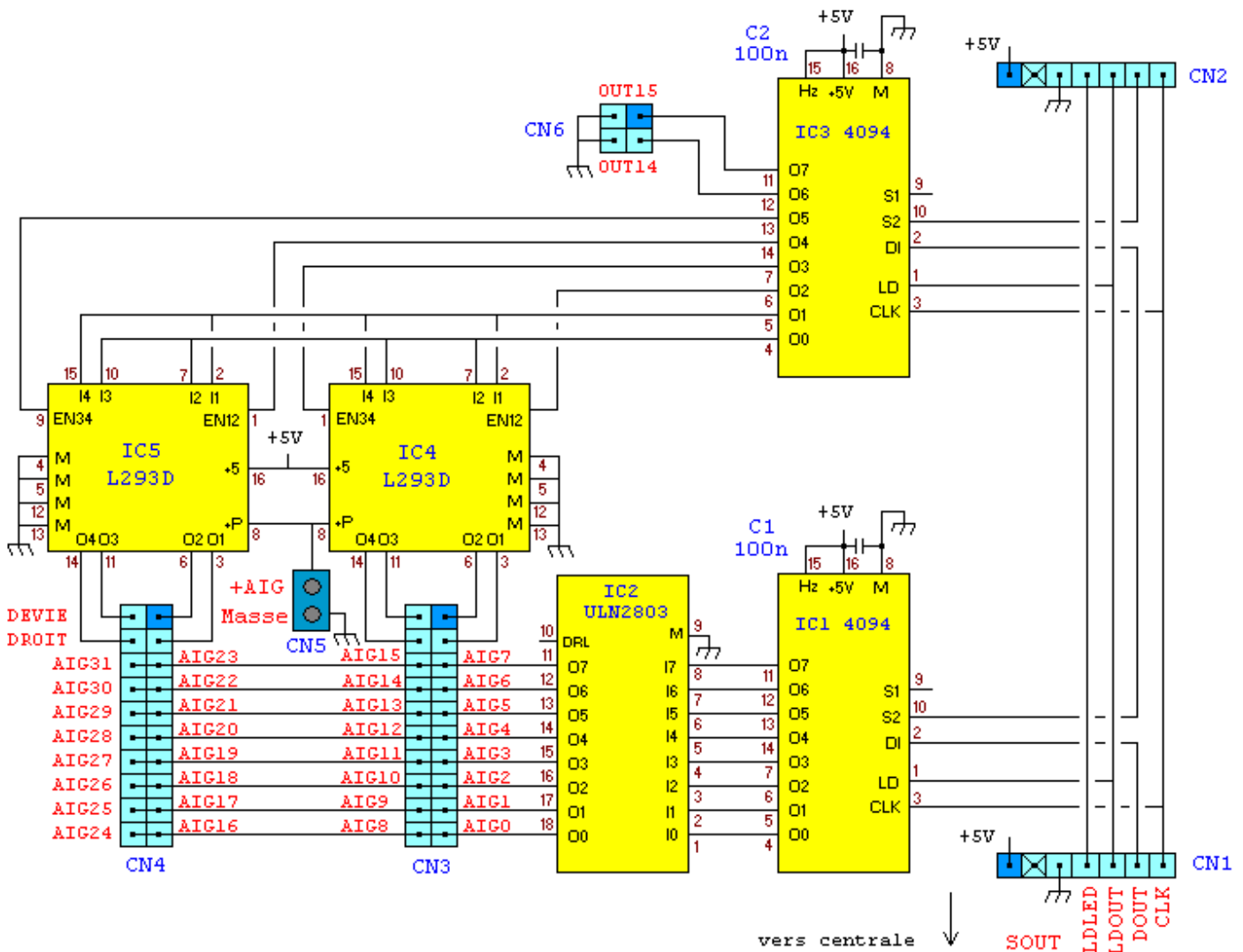
Il est à noter que ce système ne fonctionne pas correctement car le courant passe également par des interrupteurs qui ne sont pas sélectionnés. Pour y remédier, il suffit d'ajouter une diode sur chaque bobine.

Maintenant que nous avons une structure de commande matricielle, nous allons réduire le nombre d'interrupteur en répartissant nos aiguillages en 4 groupes de 8. Avec cette astuce nous n'avons besoin plus que de 16 interrupteurs (8 avec le + et 8 avec la masse).

Pour les interrupteurs à la masse, nous utiliserons un ULN2803 qui en possède 8. Il autorisera la commande d'aiguillages de 500mA. Pour de plus grosse intensité, il suffit de superposer un 2eme ULN au premier. Il n'y a pas de risque de surchauffe vu le faible temps de commande.

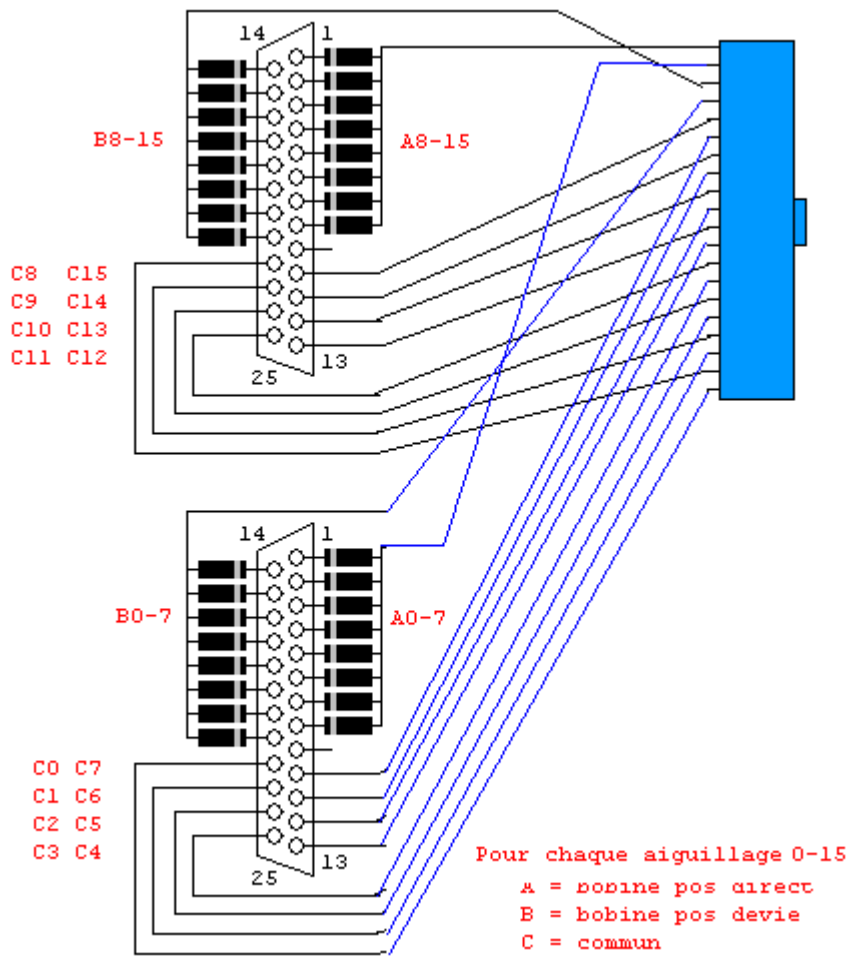
Pour les interrupteurs au plus, nous utiliserons des L293D. Chaque circuit dispose de 4 demi pont, qui peuvent chacun être utilisé pour commuter la sortie au + ou à la masse. Avec les diodes précédentes, nous utiliseront uniquement le +. Pour nos 4 groupes de 8 aiguillages, il nous faudra 8 interrupteurs et donc 2 L293D. Ce circuit est capable de commander des charges de 700 mA ce qui est suffisant pour la plupart des aiguillages.

Avec ces informations, l'étude du schéma est maintenant aisée. On retrouve 2 4094 pour extraire les données série du bus SOUT. IC1 commande les 8 interrupteurs à la masse d'IC2, tandis qu' IC3 commande les 8 interrupteurs au + d'IC4 et IC5. Au lieu de commander directement les interrupteurs au + avec 8 sortie, j'ai préféré contrôler la position des aiguillages avec les 2 première sortie puis ensuite activer le bon groupe d'aiguillage avec les entrées « enable » des L293D. Ceci permet de sauver 2 sorties qui peuvent être utilisées pour commander ce que vous voulez. Mais elles offrent surtout la possibilité d'une future extension à 48 aiguillages en ajoutant un L293D supplémentaire.

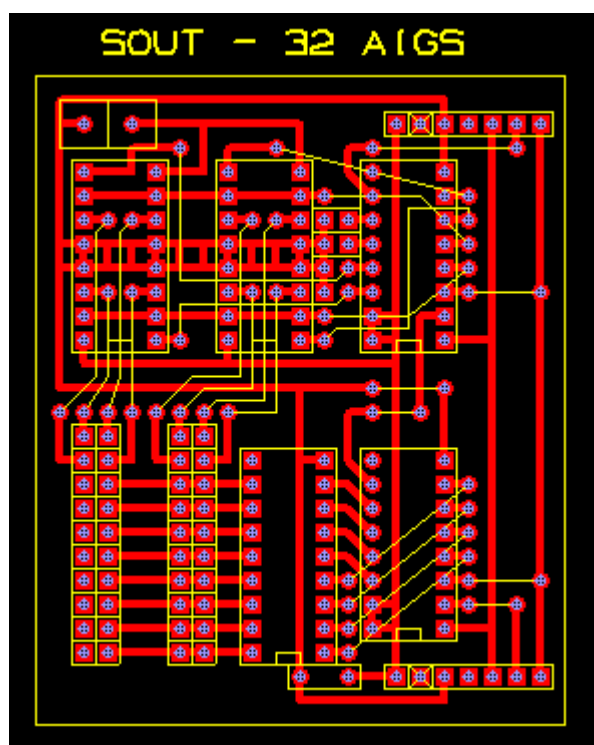


Le connecteur CN5 est destiné à recevoir l'alimentation 15V des aiguillages

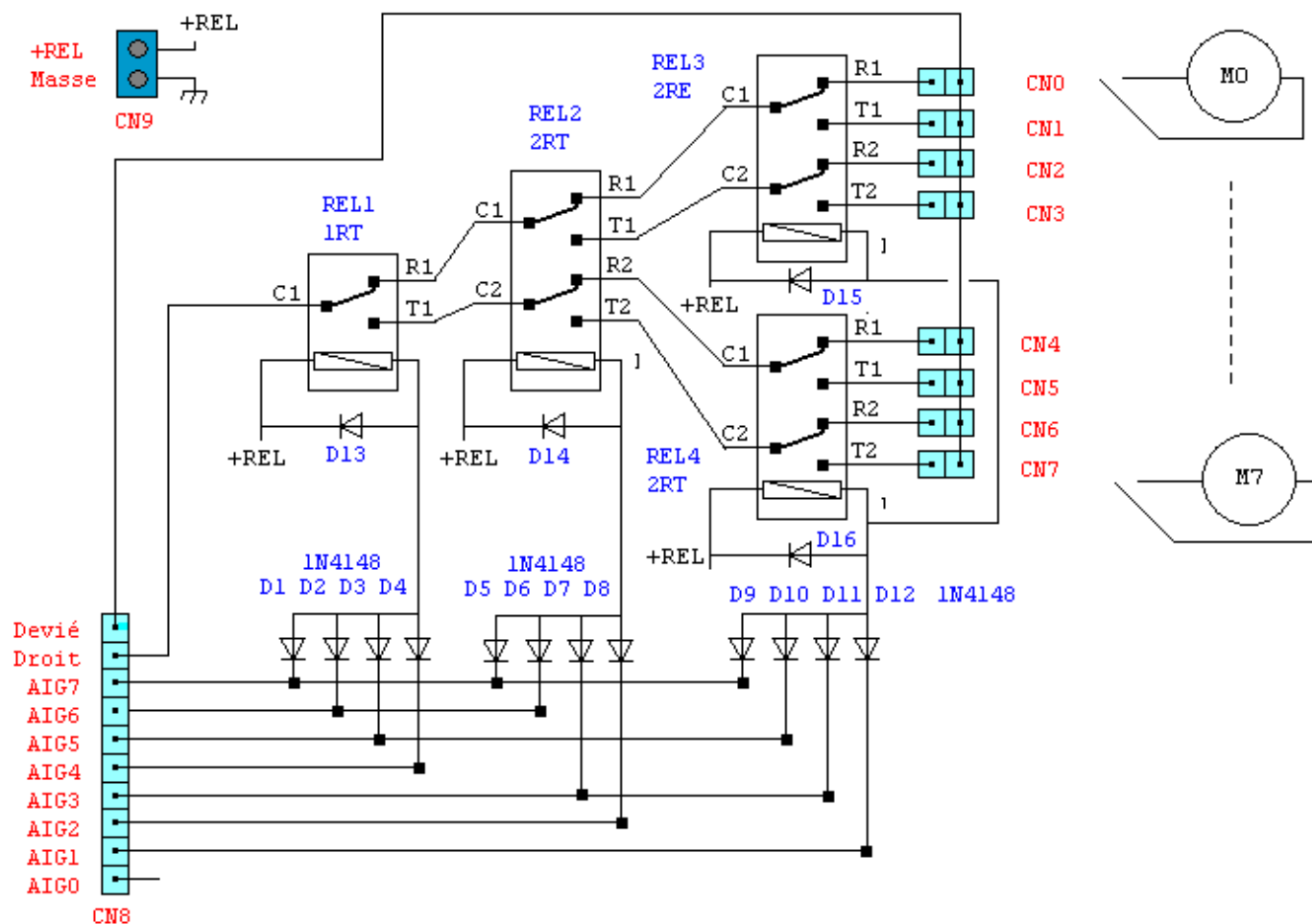
Les connecteurs CN3 et CN4 permettent chacun de commander 16 aiguillages. Pour la connexion aux aiguillages, je conseille de fabriquer l'adaptateur suivant. Sertir un câble nappe sur un connecteur HE10 2\*10 femelle, détacher les fils de la nappe et y connecter 2 connecteurs DB25 sur lesquels on soude les 32 diodes nécessaires aux 16 aiguillages. La figure suivante explicite le concept.



Coté réalisation vous pouvez utiliser de la plaque pastillée percée ou le typon suivant :



Il est également possible de contrôler des aiguillages à mouvements lents (donc muni de moteurs à la place de solénoïdes) en ajoutant 4 relais par groupe de 8 aiguillages comme le présente le schéma suivant.



Pour commander les aiguillages le L293D positionne les signaux « Dévié » et « Droit » du groupe de 8 aiguillages sélectionné comme suit :

- Pour une commande déviée : il met +AIG sur « Dévié » et la masse sur « Droit »
- Pour une commande droite : il met la masse sur « Dévié » et +AIG sur « Droit »
- Au repos, il laisse ces 2 signaux sont en l'air.

Donc, si l'on branche un moteur entre ces 2 signaux, alors il tournera dans un sens lors de la commande déviée, dans l'autre sens lors de la commande droite, tandis qu'au repos il ne tournera pas.

Il faudra bien entendu augmenter le temps de manœuvre car contrairement à un solénoïde qui déplace l'aiguille en quelques centaines de millisecondes, un moteur à besoin de plusieurs secondes. Il est à noter que les mécanismes des moteurs lents d'aiguillages disposent d'interrupteurs qui coupent la commande du moteur lorsqu'il est en position souhaité. Il n'y a donc aucun risque à commander trop longtemps un tel dispositif.

Pour contrôler jusqu'à 8 moteur par groupe de 8 aiguillage, il suffit de distribuer les signaux « Droit » et « Dévié » aux 8 moteurs. Ceci est fait par les 4 relais REL1-4 qui en fonction des signaux de commandes AOG0-7, relie un des moteurs aux 2 signaux précédents.

Afin de ne pas modifier les signaux AIG0-7 suivant qu'il s'agit de commander des solénoïdes ou des moteurs, quelques diodes alimentent les bons relais en fonction du signal actif. Par exemple, lorsque l'aiguillage 5 à besoin d'être commandé, alors AIG5 passe à la masse au lieu d'être en l'air. Les bobines des relais REL1, 3 et 4 sont donc alimentées entre +REL et la masse ce qui relie les broches C et T de ces relais. Pour le relais REL2 non sélectionné, les broches C et R sont reliées. Au final le moteur de l'aiguillage 5 se retrouve donc alimenté par les signaux « Droits » et « Dévié ». Les 700mA max du L293D sont largement suffisants pour les moteurs.

Vous pouvez bien entendu utiliser un relais 4RT à la place des 2 relais REL3 et REL4. N'oubliez pas d'alimenter les relais par CN9 et de relier la masse de cette alim à la carte de commande des aiguillages.

### 5.1. Branchements

Pour les branchements, utilisez :

- Le chapitre sur la présentation de Free-DCC présente les connexions à haut niveau.
- La partie électronique

Vous pouvez par exemple suivre le cheminement suivant :

- Programmer le UC et le mettre sur la centrale
- Alimenter la centrale avec du 5V régulé et vérifier que la led d'état clignote
- Eteindre l'alimentation 5V
- Connecter les modules d'entrées en chaîne sur le bus S88
- Connecter les modules aiguillages et leds en parallèle sur le bus Sout de la centrale puis les modules de sortie en chaîne sur le bus Sout
- Connecter le booster
- Connecter le bouton d'arrêt d'urgence
- Connecter le capteur infra-rouge
- Connecter la console
- Connecter les souris
- Alimenter la centrale et vérifier le fonctionnement de l'ensemble.
- Alimenter le booster
- Alimenter les aiguillages et sorties
- connecter les voies aux détecteurs ou booster
- connecter les détecteurs aux modules d'entrées
- connecter les aiguillages au module d'aiguillage
- connecter les feux et autres leds au module des leds
- connecter les sorties aux accessoires
- vérifier le fonctionnement de l'ensemble

La console, les souris et le bouton d'arrêt d'urgence peuvent être connectés et déconnectés avec la centrale sous tension. Tandis que la connexion des modules d'entrées, de sorties, des leds, des aiguillages, du capteur IR doit impérativement se faire hors tension.



### Les écrans

La console permet de configurer et piloter la centrale et le réseau. Elle se connecte sur le bus I2C et peut être déconnectée et reconnecter à tout moment sans même couper l'alimentation. Elle se compose d'un écran à cristaux liquides LCD de 2 lignes de 16 caractères ainsi que d'un clavier à 12 touches comportant les chiffres 0 à 9 plus l'étoile et le dièse. En termes informatiques, la console constitue l'interface utilisateur de la centrale. Cette interface se compose de plusieurs menus que l'on appellera pour plus de commodité des écrans. On passe généralement d'un écran à un autre avec des appuis courts ou longs sur les touches \* ou #. La figure suivante présente ces différents écrans :

```
    ^**
sel bienvenue          <** #,##> bienvenue
v*    ^**
sel traction           <** #,##> conduite    <** ##> fonctions <** ##> adresses/souris ##>
v*    ^**
sel aiguillages/iti    <** #,##> aiguillages <** ##> itineraires  ##>
v*    ^**
sel deplacements       <** #,##> deplacements
v*    ^**
sel horloge            <** #,##> horloge
v*    ^**
sel programmation-CV  <** #,##> programmation-CV
v*    ^**
sel configuration     <** #,##> configuration
v*    ^**
sel edition            <** #,##> edition
v*    ^**
sel test-debug        <** #,##> test-debug
v*

* = appuie court sur *      # = appuie court sur #
** = appuie long sur *     ## = appuie long sur #
```

La mise en place d'appuie courts et long est nécessaire vu le faible nombre de touche du clavier. Un appuie court est un appuie normale, comme par exemple lorsque vous utiliser un digicode, un appuie long quand a lui est plus long. Pour tester ces différents appuies, entraînez vous avec les écrans de sélection (à gauche), un appuie court, fait descendre, tandis qu'un appuie long fait monter.

L'écran de démarrage est configurable par les CC. Par défaut il s'agit de l'écran bienvenu. A gauche, les écrans de sélections permettent de choisir un des écrans de travail de droite.

Afin de faciliter l'utilisation, Il existe les raccourcis suivants :

```
Conduite <88 88> aiguillages
Conduite <88      itinéraires
Tous <** , ## 00> arrêt d'urgence
```

### Ecran de bienvenu

Cet écran indique la version et la date du logiciel de la centrale. Il est donc très utile de le consulter avant de me remonter un problème. C'est également l'écran par défaut (à moins que vous ne le changiez).

```
Free-DCC v0.6
Ulysse mai2007
```

### Ecrans de sélections

Ces écrans servent à sélectionner un écran d'utilisation. Ils comportent tous le texte Menu principal sur la première ligne. La seconde ligne indique l'écran sélectionné. Il est donc possible de le changer avec la touche \* (ou \*\*) et de le sélectionner avec la touche # (ou ##). Voici un exemple avec le menu traction.

```
Menu principal .  
Traction .
```

## Ecrans de traction

Les écrans de tractions au nombre de 3 permettent de conduire les locomotives, manipuler les fonctions, choisir les adresses et sélectionner les souris. Free-DCC dispose de 16 canaux numérotés de 0 à 15 pour piloter les locomotives. Seulement 3 apparaissent en simultanée à l'écran. Pour afficher les autres canaux, il suffit de se déplacer avec des appuis longs sur les touches 7 et 9. A la mise sous tension les canaux 0/1/2 sont affichés, un appui long sur la touche 9, affiche les canaux 1/2/3 puis 2/3/4 ... Inversement un appui long sur 7 produit l'effet inverse 0/1/2 puis 15/0/1 ...

### Ecran de conduite

```
*09 *58 -- 0640  
+10 -05 -- 2C00
```

Cet écran permet de conduire 3 locomotives. Il peut donc être utilisé comme une souris.

Le bloc de droite indique la date et l'heure ainsi que le numéro du canal le plus à gauche. Soit jour 2 (mardi) 6h40, canal 0. L'exemple affiche donc des informations sur les canaux 0/1/2. Le nombre du haut correspond à l'adresse de la locomotive, tandis que celui du bas indique la vitesse (0 à 31), et le sens (+/-). L'étoile indique si la fonction 0 est active. (La fonction 0, aussi connu sous le nom de F0 ou FL est généralement utilisée pour l'activation des phares) Un canal inutilisé (adresse 0) est représenté par des –

Les touches 1,4,7,\* sont utilisés pour le canal de gauche, 2,5,8,0 pour le canal central et 3,6,9,# pour celui de droite. 1,2,3 servent à accélérer (Tant que vous restez appuie, la vitesse augmente). Inversement 4,5,6 servent à ralentir. Chaque appuie sur les touches 7,8,9 inverse les sens tandis que \*,0,# inverse la fonction 0.

```
1 2 3 = +  
4 5 6 = -  
7 8 9 = sens  
* 0 # = F0
```

### Ecran des fonctions

```
*09 *58 --  
1--3 -2-- ----
```

Cet écran permet de piloter les fonctions F1, F2, F3 et F4 (si le décodeur en dispose). La première ligne indique les adresses et l'état de la fonction 0 tandis que la seconde ligne indique l'état des fonctions 1 à 4. Un – indique que la fonction est désactivée, un chiffre correspondant au numéro de la fonction indique qu'elle est activée.

Les touches 1,4,7,\* sont utilisés pour le canal de gauche, 2,5,8,0 pour le canal central et 3,6,9,# pour celui de droite. 1,2,3 servent à F1, 4,5,6 à F2 et ainsi de suite

```
1 2 3 = F1  
4 5 6 = F2  
7 8 9 = F3  
* 0 # = F4
```

Si une souris ne permet pas de piloter les fonctions, alors la console peut palier à ce manque.

### Ecran des adresses et souris

```
*09 *58 -- X>00
CH00 X00
```

Cet écran permet de sélectionner une adresse de décodeur ainsi qu'une souris pour chaque canal.

Afin de se repérer, comme précédemment les 3 premiers chiffres éventuellement précédé par \* sont les adresses et F0. La ligne du bas indique le numéro du canal à configurer ainsi que le type et numéro de souris associé.

L'élément de droite permet de choisir le type de souris et rentrer des nombres.

A l'entrée dans le menu, le curseur est positionné sur le canal. Il est alors possible de garder ce canal ou alors de le changer avec des appuis longs sur 7 et 9 ou en entrant un numéro qui s'affiche en haut à droite pour vérification. Un appui sur # change le canal. Un appui sur \* passe à l'étape suivante

Le curseur se positionne ensuite sur l'adresse. Il est alors possible de la garder ou alors de la changer en entrant un numéro qui s'affiche en haut à droite pour vérification. Un appui sur # change l'adresse tandis que \* permet de passer à l'étape suivante.

Le curseur se positionne maintenant sur le type de souris représenté par une lettre. Chaque appui sur # change ce type. Un appui sur \* passe à l'étape suivante.

Enfin le curseur se positionne sur le numéro de souris. Saisir un numéro de souris. Il s'affiche en haut à droite pour vérification. Un appui sur # change la souris (type + numéro). Un appui sur \* permet de revenir au choix du canal.

Il est à noter que l'adresse 0 désactive le canal. Si l'utilisateur veut affecter une adresse déjà utilisée par un autre canal, il doit en premier désactiver cette adresse. A la mise sous tension, les adresses ainsi que la configurations des souris sont récupérées des CC. Les modifications faites par ce menu ne sont pas conservées à la mise hors tension, pour les réglages permanents, utilisez les CC.

Les types de souris sont les suivants :

```
X : Pas de souris associée (mais la console peut très bien servir de souris)
Y : non utilisé (comme X)
U<0-31> : UM (Pour les unités multiples, ce canal suit les consignes de vitesse du canal
           dont l'adresse est indiquée (adresse 31 max))
V<0-31> : UM inverse (comme U mais le sens est inversé)
P<0-4> : souris potentiométriques
I<0-2> : souris infrarouge
N<1-15> : souris I2C numériques (1-7 pour PCF8574 (0 pour console), 8-15 pour PCF8574A)
A<0-31> : souris I2C analogiques (PCF8591 (0-7) + 8*numéro du canal (0-3))
```

## Ecrans d'aiguillages et d'itinéraires

Ces écrans au nombre de 2 permettent de manoeuvrer les aiguillages et activer des itinéraires prédéfinis

### Ecran des aiguillages

```
Aig-->01 /
- / - - - . / / - - - - - . . .
```

Cet écran permet de changer les aiguillages de position.

Pour ce faire, il suffit d'entre le numéro de l'aiguillage puis de le mettre :

- en position directe avec le bouton \*
- ou en position déviée avec le bouton #.

La position de l'aiguillage est indiquée juste à coté du numéro par

- un « -« (direct)
- ou « / » (dévié)
- ou « . » (en manœuvre, lorsque la consigne est différente de la position)
- EE si la saisie dépasse 31 car Free-DCC gère 32 aiguillages (0-31)

- V complète ces indications si l'aiguillage est utilisée pour un déplacement automatique (dans ce cas il ne peut pas être déplacé)

La seconde ligne affiche la position du groupe de 16 aiguillages où se trouve la sélection. Le plus à gauche est soit l'aiguillage 0 soit le 16

La position des aiguillages à la mise sous tension est paramétrable dans les CC

## Ecran des itinéraires

```
It1-->22 OK  
----//----/--/-
```

Cet écran permet d'établir les itinéraires.

Pour ce faire, il suffit d'entre le numéro de l'itinéraire puis de l'activer avec \* ou #.

A ce moment la, les commandes des aiguillages définis dans l'itinéraire sont mises à jour. Ces itinéraires se comportent donc simplement comme les itinéraires à diode que l'on câble généralement. Il ne sont donc pas verrouillables, n'attendent pas les trains ... Ceci pour faire simple et compréhensible.

Lorsque l'itinéraire est établi un OK s'affiche à côté du numéro.

Les itinéraires (aiguillages à utiliser ainsi que leur position. ) sont stockés en mémoire flash du UC et peuvent être facilement mis à jour avec le logiciel adéquat par la rs232 ou alors en utilisant le menu édition de la centrale.

Il est à noter qu'un itinéraire ne peut pas être activé si un des aiguillages est utilisé dans un déplacement automatique.

La seconde ligne affiche la position des 16 premières aiguilles. Oae plus à gauche est donc l'aiguillage 0

## Ecran des déplacements

```
15B 02E --A 0616  
--A --A --A 2>15
```

Cet écran permet de lancer, surveiller et interrompre les déplacements automatiques. Free-DCC peut gérer 6 déplacements automatiques en même temps qui utilisent les 6 derniers canaux (10,11,12,13,14,15). L'écran comporte donc les numéros des déplacements en cours (0-49) ainsi que leurs états qui sont représentés par une lettre. La droite de l'écran affiche l'heure, le jour (ici 2) ainsi qu'une zone de saisie.

Pour lancer un déplacement, il suffit de saisir son numéro puis d'appuyer sur la touche #. A ce moment Free-DCC recherche le premier canal libre parmi les 6 derniers et lance le déplacement. Le déplacement passe alors de l'état A (inactif) à l'état B (réservation des aiguillages ...) et ainsi de suite. A la fin du déplacement, le canal est libéré. Un chapitre traite en détail les déplacements. Un canal est considéré comme libre s'il contient l'adresse 0. Il est donc possible d'utiliser également les 6 derniers canaux en manuel, car le mode automatique ne va pas utiliser un canal qui contient déjà une adresse. Avant de lancer un déplacement, si la locomotive est déjà utilisée en manuel, il faut la retirer du mode manuel, par exemple en mettant l'adresse de son canal à 0.

Lancer les déplacements en manuel ne présente que peu d'intérêts sauf pour le test ou pour envoyer ou recevoir des trains (dans une coulisse). C'est généralement le rôle du séquenceur de lancer les différents déplacements. Avec le séquenceur, il est possible de définir un temps maximal d'exécution de chaque déplacement afin de bloquer l'horloge si un train n'a pas fini son déplacement dans les temps suite par exemple à un problème de captage de courant ou à un déraillement. Ceci est vital en automatique afin d'éviter les collisions et dans une moindre mesure, autoriser les correspondances. Lorsque l'horloge est bloquée le > se transforme en \*. Si c'est un canal qui a bloqué l'horloge alors un \* est également ajouté à côté de la lettre d'état du canal.

Si besoin, pour stopper prématurément un déplacement, il suffit de saisir le numéro du canal de déplacement (sans les dizaines) (0 pour le canal en haut à gauche qui est en fait le canal 10 ...) puis de faire un appui long sur la touche #.

10	11	12	numéros des canaux free DCC
0	1	2	numéros à saisir pour stopper
15B	02E	--A	0616
--A	--A	--A	2>15
3	4	5	numéros à saisir pour stopper
13	14	15	numéros des canaux free DCC

## Ecran de l'horloge

Afin d'exploiter le réseau avec réalisme et donner une référence au séquenceur, Free-DCC intègre une horloge de modélisme. Une telle horloge présente la particularité de pouvoir faire passer le temps plus vite qu'en réalité. En effet, qui veut attendre une heure entre chaque train. Pour ce faire il est possible de définir la durée d'une minute de modélisme en seconde réelles. Par exemple avec 1 seconde réelle pour une minute de modélisme, une heure de modélisme dure 60s soit 1 minute. Afin de rajouter de la crédibilité, l'horloge gère aussi le jour de la semaine car comme en réalité le trafic du dimanche n'est sans doute pas le même que celui du lundi ... Afin d'éviter de trop longs temps morts, il est possible de définir l'heure de fin de journée et celle du début. En effet si votre gare ne voit pas passer de trains entre 22h et 6h à quoi bon attendre. Bien entendu l'heure et le jour de démarrage sont également paramétrables. Tous ces paramétrages se font comme d'habitude grâce aux CC.

```
02-06:42 1m=10s
>10 .
```

Cet écran affiche l'horloge de modélisme. On y trouve :

- Le numéro du jour de la semaine (0-6, par exemple 0 pour Lundi ...)
- L'heure courante (0-23)
- La minute courante (0-59)
- La durée de la minute de modélisme en secondes réelles (1-255)

Une zone de saisie complète l'affichage.

Le symbole « : » se transforme en « B » lorsque l'horloge est bloquée par le séquenceur ou en « b » lorsqu'elle est bloquée par l'utilisateur.

Pour changer une valeur, il suffit de placer le curseur dessus avec la touche \*, de saisir la valeur qui s'affiche dans la zone de saisie puis de valider avec la touche #.

Lorsque le curseur est sur le « : », un appui sur la touche # permet de bloquer l'horloge. On parle alors de blocage utilisateur et la lettre « b » remplace le « : ». Ne pas confondre avec un blocage du séquenceur qui affiche la lettre « B » en cas d'un déplacement qui ne se termine pas dans les temps ou d'un déplacement qui ne peut pas être lancé suite à l'absence d'un canal automatique libre. Un blocage séquenceur ne peut être débloqué que par la résolution du problème (fin du déplacement fautif ou canal automatique disponible). L'affichage du « B » est prioritaire sur le « b ».

## Ecran de programmation des CV

Cet écran permet de programmer les variables de configuration des décodeurs des locomotives.

En fait il programme tous les décodeurs des loco présentes sur les voies. Il faut donc impérativement qu'il n'y en ai qu'une seule au moment de la programmation car sinon elles seront toutes programmées avec les mêmes valeurs ! Pour éviter ce genre de mésaventure, il est possible de définir dans les CC les entrées qui doivent être à 0 avant d'accéder au menu de programmation. 3 solutions s'offrent alors à vous :

- Ne rien faire et courir le risque
- Ne pas utiliser Free-DCC pour la programmation
- Installer des détecteurs de courant sur toutes les voies et interdire l'accès au menu de programmation si une des voies (à l'exception de la voie de programmation) est occupée.

- Utiliser la première partie d'un interrupteur double pour couper l'alimentation des voies (sauf celle de programmation). L'autre partie de l'interrupteur étant connecté à une entrée de Free-DCC qui interdit l'accès au menu de programmation lorsque l'interrupteur alimente toutes les voies.

Pour savoir comment fonctionne la programmation en DCC, reportez vous au chapitre DCC. Vu les différents modes de programmations qui existent dans la norme et les implémentations plus ou moins bien faites, il est possible que Free-DCC n'arrive pas à programmer certains décodeurs. Dans ce cas, merci de me signaler les décodeurs en question. Vous pouvez bien entendu essayer avec un autre programmeur.

```
cv=001 dat=022
p001/r0 ---1-11-
```

A droite, l'écran affiche le numéro de la variable (1-255) de configuration « CV ». En fait la norme définit jusqu'à 1024 variables mais elles ne servent pas toutes et les 256 premières suffisent largement. Pour info la page et le registre de programmation sont également affichés mais ne servent pas à grand-chose car la plupart des gens et fiches techniques des décodeurs parlent directement de cv.

La partie gauche affiche la valeur en décimal en haut ainsi qu'en binaire en bas pour vérifier que vous ne vous trompez dans la valeur des bits.

Les touches numériques sont utilisées pour saisir la valeur

La touche \* est utilisée pour passer du cv à la donnée

Un appui long sur # programme le CV.

Free-DCC n'a aucun moyen de lire les CV du décodeur, mais la locomotive devrait confirmer la programmation en allumant brièvement les feux ou le moteur pendant un court instant.

Pour savoir à quoi correspond un CV, lisez la partie DCC de ce document et surtout la notice de votre décodeur.

## Ecran de configuration (des CC)

Cet écran permet de lire et mettre à jour les variables de configuration de la centrale, les fameuses CC. Ces variables n'ont rien à voir avec la norme DCC et les cv, elle servent juste à rendre la centrale très flexible. Ces variables sont stockées en sûreté dans l'EEPROM du microcontrôleur.

```
Conf=002 dat=012
-----11--
```

Conf indique le numéro de la variable (0-255) et dat sa valeur (0-255). Pour vérifier les bits, la valeur est également affichée en binaire.

La touche \* permet de passer de conf à dat tandis

Un appui sur # lit la variable tandis qu'un appui long l'écrit.

Un chapitre a part décrit toutes les variables de configuration CC.

## Ecran d'édition

```
I000-00=015 .
>000 -----1111
```

Cet écran permet d'écrire directement dans la mémoire flash du microcontrôleur afin de modifier les tables suivantes qui contiennent les:

- 32 itinéraires (0-31)
- 256 équations
- 256 ordres du séquenceur
- 50 déplacements.

Vu le risque d'erreur important, je ne décris pas cet écran actuellement et je vous conseille d'utiliser à la place le logiciel adéquat pour charger ces tables via la rs232 et à partir d'un fichier au format bien plus agréable et compréhensible.

## Ecran de test et debug

Cet écran permet de lire et modifier la mémoire vive (RAM) et les registres du microcontrôleur. Il pourra par exemple être utilisé pour lire des emplacements particuliers comme le dernier code IR reçu (adresse I+41), les can0 (adresse 47) ou lors de mise au point, ou en cas de problèmes.

```
adr=012  dat=080  
--      -1-1----
```

adr indique l'adresse mémoire et dat sa valeur en décimal et en binaire.

La touche \* permet de passer de adr à dat.

Un appui sur # lit la valeur et un appui long sur # écrit la valeur.

Un appui long sur la touche 7 permet d'utiliser les 256 adresses supplémentaires du UC. Le premier tiret du bas se change alors en I

Un appui long sur la touche 8 active la lecture en continue. Le second tiret se transforme alors en C.

### 5.3. Configuration de la centrale avec les CC

Les variables CC permettent de configurer la centrale. Elles sont stockées en EEPROM afin de garder leur contenu lorsque la centrale n'est plus alimentée et permettre une modification aisée. Pour les modifier, il est possible de le faire lors de la programmation du UC avec ICPROG, lors de l'utilisation avec la console ou encore via RS232 grâce au logiciel fournit. Il est impératif de ne pas mettre n'importe quoi dans certaines variables, par exemple une impulsion de 10s sur un aiguillage à solénoïde risque de le détruire. Les tableaux suivants indiquent décrivent chaque CC et donnent la valeur par défaut présente dans le fichier hex livré.

Certaines variables sont codées en binaire. Cela a déjà été expliqué au chapitre DCC, mais voici un rappel. Pour un octet qui contient donc 8 bits (0/1). Celui le plus à gauche a un poids de 1, le suivant 2 et ainsi de suite. Pour connaître la valeur d'un octet, il suffit de remplacer par 0 les bits à 0 à partir de l'expression suivante et de faire la somme :  $128+64+32+16+8+4+2+1$

ex :  $0.0.1.0.1.1.0.0 = 0+0+32+0+8+4+0+0 = 44$

CC 0-15 - Adresse des locomotives pour chaque canal à la mise sous tension.

L'adresse 0 signifie que le canal est non utilisé.

CC	Défaut	Description
0	9	Adresse de la locomotive du canal 0
1	10	Adresse de la locomotive du canal 1
2	11	Adresse de la locomotive du canal 2
3	12	Adresse de la locomotive du canal 3
4	13	Adresse de la locomotive du canal 4
5	8	Adresse de la locomotive du canal 5
6	0	Adresse de la locomotive du canal 6
7	0	Adresse de la locomotive du canal 7
8	0	Adresse de la locomotive du canal 8
9	0	Adresse de la locomotive du canal 9
10	0	Adresse de la locomotive du canal 10
11	0	Adresse de la locomotive du canal 11
12	0	Adresse de la locomotive du canal 12
13	0	Adresse de la locomotive du canal 13
14	0	Adresse de la locomotive du canal 14
15	0	Adresse de la locomotive du canal 15

CC 16-31 - Choix des souris pour chaque canal à la mise sous tension.

La valeur à mettre est la suivante type.type.type.num.num.num.num.num

Le tableau suivant définit le type. Dans l'interface, le type est désigné par une lettre.

Valeur	Valeur	Lettre	Description
0	0.0.0.n.n.n.n.n	X	Pas de souris
1	0.0.1.n.n.n.n.n	Y	Pas de souris
2	0.1.0.n.n.n.n.n	U	UM, suit la consigne du canal d'adresse de loco nnnnn 0 -31
3	0.1.1.n.n.n.n.n	V	UM inverse, idem à U mais en inversant le sens
4	1.0.0.n.n.n.n.n	P	Potentiomètre (nnnnn = 0 / 1 / 2 / 3 / 4)
5	1.0.1.n.n.n.n.n	I	Infra-rouge (nnnnn = 0 / 1 / 2)
6	1.1.0.p.n.n.n.n	N	I2C Numerique (nnnn = 0-7, p=0=PCF8574 p=1=PCF8574A)
7	1.1.1.c.c.n.n.n	A	I2C Analogique (adresse I2C nnn = 0-7, numero du canal 0-3)

Valeur =  $32 * \text{type} + \text{numéro}$



CC	Défaut	Description
16	0	Souris du canal 0
17	0	Souris du canal 1
18	0	Souris du canal 2
19	0	Souris du canal 3
20	0	Souris du canal 4
21	0	Souris du canal 5
22	0	Souris du canal 6
23	0	Souris du canal 7
24	0	Souris du canal 8
25	0	Souris du canal 9
26	0	Souris du canal 10
27	0	Souris du canal 11
28	0	Souris du canal 12
29	0	Souris du canal 13
30	0	Souris du canal 14
31	0	Souris du canal 15

#### CC 32-39 – Phase des leds

Phase des leds en mode clignotant.

0 = même phase que la led d'état du UC / 1 = phase inverse.

CC	Défaut	Description (phases pour les leds suivantes)
32	0	07.06.05.04.03.02.01.00
33	0	15.14.13.12.11.10.09.08
34	0	23.22.21.20.19.18.17.16
35	0	31.30.29.28.27.26.25.24
36	0	39.38.37.36.35.34.33.32
37	0	47.46.45.44.43.42.41.40
38	0	55.54.53.52.51.50.49.48
39	0	63.62.61.60.59.58.57.59

#### CC 40-47 – Initialisation du clignotement des leds

Mode clignotant des leds à la mise sous tension. Lorsqu'une led est activée elle clignotera ou non suivant ce mode. Le mode clignotant est également accessible par les équations contrairement à la phase qui est fixée par les CC32-39.

0 = mode continu / 1 = mode clignotant.

CC	Défaut	Description (phases pour les leds suivantes)
40	0	07.06.05.04.03.02.01.00
41	0	15.14.13.12.11.10.09.08
42	0	23.22.21.20.19.18.17.16
43	0	31.30.29.28.27.26.25.24
44	0	39.38.37.36.35.34.33.32
45	0	47.46.45.44.43.42.41.40
46	0	55.54.53.52.51.50.49.48
47	0	63.62.61.60.59.58.57.59

#### CC 48-66 – Touches de la télécommande IR

Une télécommande infrarouge (IR) peut être utilisée pour piloter jusqu'à 3 canaux DCC.

Les CC suivantes permettent de définir le numéro des touches pour chaque action.

Comme indiqué dans le chapitre des écrans, l'écran de test avec l'adresse I+41 indique le numéro de la dernière touche appuyée.

CC	Défaut	Description (fonction de la touche)
48	32	Canal IR 1 - Incrémentation de la vitesse
49	1	Canal IR 1 - Incrémentation (2 touches sont possibles)
50	205	Canal IR 2 - Incrémentation
51	2	Canal IR 2 - Incrémentation
52	222	Canal IR 3 - Incrémentation
53	3	Canal IR 3 - Incrémentation
54	33	Canal IR 1 - Décrémentation de la vitesse
55	4	Canal IR 1 - Décrémentation (2 touches sont possibles)
56	0	Canal IR 2 - Décrémentation
57	0	Canal IR 2 - Décrémentation
58	0	Canal IR 3 - Décrémentation
59	0	Canal IR 3 - Décrémentation
60	0	Canal IR 1 - Inversion du sens
61	0	Canal IR 2 - Inversion du sens
62	0	Canal IR 3 - Inversion du sens
63	0	Canal IR 1 - Fonction F0
64	0	Canal IR 2 - Fonction F0
65	0	Canal IR 3 - Fonction F0
66	0	Arrêt d'urgence

#### CC 67-98 - Durée des impulsions des aiguillages

Durée des impulsions de commande des aiguillages par pas de 100ms.

1=100ms / 2=200ms / 10=1s / 255=25.5ms

Pour les aiguillages à solénoïdes, commencer par la valeur la plus faible (1) puis augmentez la si cela ne suffit pas. Les grandes valeurs sont plutôt réservées aux aiguillages à moteur (à mouvements lents).

CC	Défaut	Description
67	1	Aiguillage 0 - Durée de l'impulsion (par pas de 100ms)
68	1	Aiguillage 1
69	1	Aiguillage 2
70	1	Aiguillage 3
71	1	Aiguillage 4
72	1	Aiguillage 5
73	1	Aiguillage 6
74	1	Aiguillage 7
75	1	Aiguillage 8
76	1	Aiguillage 9
77	1	Aiguillage 10
78	1	Aiguillage 11
79	1	Aiguillage 12
80	1	Aiguillage 13
81	1	Aiguillage 14
82	1	Aiguillage 15
83	1	Aiguillage 16
84	1	Aiguillage 17

85	1	Aiguillage 18
86	1	Aiguillage 19
87	1	Aiguillage 20
88	1	Aiguillage 21
89	1	Aiguillage 22
90	1	Aiguillage 23
91	1	Aiguillage 24
92	1	Aiguillage 25
93	1	Aiguillage 26
94	1	Aiguillage 27
95	1	Aiguillage 28
96	1	Aiguillage 29
97	1	Aiguillage 30
98	1	Aiguillage 31

CC 99-102 - Positionnement des aiguillages à la mise sous tension.

0 = position direct / 1 = position dévié

CC	Défaut	Description (position des aiguillages suivants)
99	0	07.06.05.04.03.02.01.00
100	0	15.14.13.12.11.10.09.08
101	0	23.22.21.20.19.18.17.16
102	0	31.30.29.28.27.26.25.24

CC 103-107 - Réglage de l'horloge de modélisme

CC	Défaut	Description (position des aiguillages suivants)
103	2*32+5	j.j.j.h.h.h.h.h jour (0-7) + heure (0-23) à la mise sous tension
104	55	0.b.m.m.m.m.m.m blocage + minute (0-59) à la mise sous tension
105	22	0.0.0.h.h.h.h.h heure de fin du jour (0-23)
106	5	0.0.0.h.h.h.h.h heure de début du jour (0-23)
107	10	s.s.s.s.s.s.s.s durée d'une minute de modélisme en secondes réelles

CC 108 - Mode PWM

Afin d'utiliser une locomotive analogique (sans décodeur), Free-DCC peut génère un signal PWM à la place du signal DCC. Attention, dans ce mode vous devez soit ajouter une diode en sortie d'un booster classique et un inverseur pour gérer le sens ou alors utiliser un booster dont vous pouvez contrôler les 2 demis ponts. Comme une locomotive analogique fonctionne en 12V et un booster dans les 18V, il faut impérativement baisser la valeur moyenne de la tension. La vitesse de consigne max étant 31, il suffit de reboucler le compteur de comparaison plus loin comme 48. Pour que les détecteurs DCC fonctionnent à coup sûr en PWM et uniquement avec les boosters à doubles ponts, il est nécessaire de créer une légère impulsion en sens inverse. Cette dernière peut être désactivée car elle peut être désagréable à l'oreille.

CC	Défaut	Description
108	48	pwm_on.pwm_det.cpt.cpt.cpt.cpt.cpt.cpt

pwm\_on : utilisation du mode pwm à la place du DCC

pwm\_det : génération d'une impulsion inverse pour la détection

cpt : valeur de rebouclage du compteur (31-63)

## CC 109 – Valeur de rebouclage du compteur des équations

Des comparaisons avec ce compteur permettent par exemple de simuler des feux routiers. Voir le chapitre sur les équations pour plus de détail

CC	Défaut	Description
109	24	Valeur de rebouclage du compteur (en secondes)

## CC 110 – Numéro du 1<sup>er</sup> canal de la console

CC	Défaut	Description
110	0	du 1 <sup>er</sup> canal de la console

## CC 111 – Nombre d'aiguillages

CC	Défaut	Description
111	15	Nombre d'aiguillages

## CC 112 – Durée des impulsions de tous les aiguillages

CC	Défaut	Description
112	0	Durée des impulsions de tous les aiguillages en pas de 100ms. Si 0, utilisation des CC 77-98

## CC 113 – Durée de préparation avant de déplacer une aiguille

Cette durée permet par exemple de charger le condensateur dans le cas d'un pilotage capacitif des aiguillages.

CC	Défaut	Description
113	10	Durée en pas de 100ms.

## CC 114-115 - Activation des différents modules

CC	Défaut	Description
114	11111110 = 254	TDF.OS.IR_IT.TDV.SEQ.DEPL.DAY.0
115	11111111 = 255	S88.OUT.AIG.LED.MOUSEIR.MOUSEPOT.MOUSEI2C.MOUSEALL

TDF: Activation de la tâche de fond

OS : Activation de l'OS (qui gère les écrans)

IR\_IT : Activation de la procédure qui reçoit et décode le signal IR pour extraire le numéro de la touche

TDV : Activation des équations

SEQ : Activation du séquenceur

DEPL : Activation des déplacements

DAY : Utilisation de l'information jour pour le séquenceur

S88: Activation du bus d'entrée S88

OUT\*: Activation du bus des sorties (Les aiguillage en ont besoin)

AIG: Activation de la mise à jour des aiguillages

LED: Activation des LEDs

MOUSEIR: Activation de la gestion de la télécommande IR

MOUSEPOT: Activation de la gestion des souris potentiométriques

MOUSEI2C : Activation de la gestion des souris I2C

MOUSEALL : Activation des souris – Si 0 alors toutes les souris sont désactivées

\*Attention, il est dangereux de désactiver OUT si vous utilisez des aiguillages car sans OUT les drivers d'aiguillages ne sont pas inutilisées à 0 et risquent donc de détruire la solénoïde d'un aiguillage.

Sauf cas particulier, vous n'avez pas besoin de désactiver les modules.

#### CC 116 - Menu de départ

CC	Défaut	Description
116	10	Numéro du menu actif au démarrage

Les numéros des différents menus sont :

0	Sélection écran bienvenu
1	Sélection écran traction
2	Sélection écran aiguillages/iti
3	Sélection écran déplacements
4	Sélection écran horloge
5	Sélection écran programmation-CV
6	Sélection écran configuration-CC
7	Sélection écran test
8	Sélection écran édition
9	X
10	écran bienvenu
11	écran traction (régulateurs)
12	écran aiguillages
13	écran déplacements
14	écran horloge
15	écran programmation-CV
16	écran configuration-CC
17	Sélection écran test
18	Sélection écran édition
19	X
20	écran traction (fonctions)
21	écran traction (configuration des souris)
22	écran itinéraires
23-255	X

#### CC 117 – Configuration de la RS232

CC	Défaut	Description
117	01000001 = 65	MODE.RSTIN.0.0.2400.4800.9600.19200

#### MODE

- 0 = mode spécial (Ulysse)
- 1 = mode compatible 6050

#### RSTIN

- 0 = Remise à 0 des entrées après lecture par RS232
- 1 = Non Remise à 0 des entrées après lecture par RS232

## VITESSE

- 2400
- 4800
- 9600
- 19200

### CC 118-125 - Entrées interdisant l'entrée dans l'écran de programmation des CV

La partie dédiée au menu programmation des CV dans le chapitre de la console explique ce mécanisme.

0 = L'entrée n'interdit pas l'entrée dans le menu programmation-CV

1 = L'entrée interdit l'entrée dans le menu programmation-CV si elle est à 1

CC	Défaut	Description (numéro des entrées)
118	0	07.06.05.04.03.02.01.00
119	0	15.14.13.12.11.10.09.08
120	0	23.22.21.20.19.18.17.16
121	0	31.30.29.28.27.26.25.24
122	0	39.38.37.36.35.34.33.32
123	0	47.46.45.44.43.42.41.40
124	0	55.54.53.52.51.50.49.48
125	0	63.62.61.60.59.58.57.59

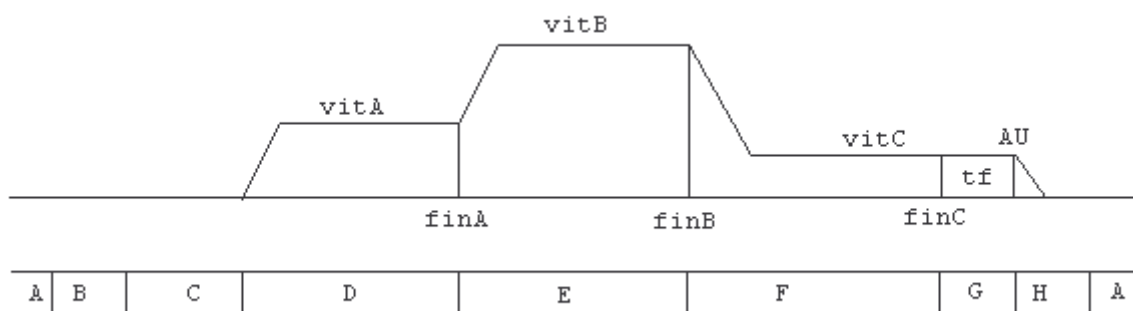
## 5.4. Les déplacements et le séquenceur

Les déplacements permettent aux trains de suivre automatiquement des itinéraires prédéfinis en toute sécurité d'un point à un autre du réseau tandis que le séquenceur sert à lancer ces déplacements à des heures fixées. Cela permet de contempler le réseau sans intervenir et d'essayer d'établir des circulations se rapprochant de la réalité.

### 5.4.A. Les déplacements

Free-DCC permet de définir 50 déplacements et d'en exécuter jusqu'à 6 simultanément en utilisant les 6 derniers canaux s'ils ne sont pas déjà utilisés en manuel. Chaque déplacement permet d'amener un train d'un point du réseau à un autre en contrôlant la vitesse et les aiguillages.

Le diagramme suivant présente le profil de vitesse ainsi que les différents états ou étapes du déplacement.



L'état A indique que le canal est libre. Le séquenceur ou l'utilisateur peut donc lancer un déplacement. A ce moment on passe à l'étape B qui vérifie si le déplacement peut être lancée. En effet avant de vérifier un déplacement, il est impératif de vérifier que la locomotive n'est pas utilisée en manuel, les aiguillages ne sont pas déjà utilisés par un autre déplacement, que les zones traversées ne sont pas déjà utilisées par un autre déplacement et enfin que les voies uniques sont configurées dans les bons sens. S'il n'y a aucun problème alors les aiguillages et zones sont réservés afin qu'un autre déplacement ne puisse pas les utiliser. De plus les commandes d'aiguillages sont mises à jour. Nous sommes maintenant à l'étape C qui attend la fin du déplacement de tous les aiguillages du parcours. Lorsque cela est fait, au début de l'étape D, les fonctions F0-F4 sont mises à jour si besoin et la consigne de sens et vitesse vitA est envoyée à la locomotive. La vitesse vit A est souvent utilisée pour sortir d'une gare. La fin de l'étape D finA peut être une temporisation de durée de l'étape ou une détection sur une entrée. Nous voilà maintenant à l'étape E qui définit une nouvelle consigne de sens et vitesse. Cette vitesse est souvent une vitesse de croisière entre 2 gares. Comme précédemment, la fin de cette étape finB survient après un délais ou l'activation d'une entrée. Nous sommes maintenant à l'étape E qui définit une nouvelle consigne de sens et vitesse. Cette vitesse peut par exemple être utilisée pour entrer en gare pas trop vite afin d'être réaliste mais surtout pour mobiliser le convoi avec précision. La fin de cette étape est également déclenchée par une temporisation ou une entrée de détection. Nous voilà à l'étape G qui rajoute une temporisation supplémentaire permettant dans le cas d'une détection de s'arrêter un peu plus loin. Lorsque cette temporisation expire la consigne d'arrêt est donnée à la locomotive. Pour un arrêt précis il est également possible d'utiliser l'arrêt d'urgence ou immédiat. L'étape H attend quelques secondes pour être sûre que la décélération est terminée et la locomotive immobilisée avant de mettre à jour les fonctions F0-F4 si cela est souhaité. Ceci fait, les ressources utilisées par le déplacement son libérées et peuvent à nouveau être utilisées pour d'autres déplacements. Le canal d'exécution automatique retourne dans l'état libre A.

Le tableau suivant résume les différents états :

Etat	Description
A	Canal d'exécution automatique disponible
B	Attente ressources disponibles (loco, aiguillages, zones)
C	Attente positionnement des aiguillages
D	Consignes de départ et attente finA
E	Consignes de croisière et attente finB

F	Consignes d'arrivée et attente finC
G	Temporisation finale
H	Attente immobilisation et libération des ressources

Ce profil de vitesse autorise des déplacements réalistes. Néanmoins si vous ne voulez pas utiliser une étape, il suffit de sélectionner le mode temporisation comme fin et de la mettre à 0.

La sécurité entre les différents déplacements automatique est assurée grâce à plusieurs dispositifs.

- Le premier est la réservation des aiguillages, de ce fait si un déplacement utilise au moins un aiguillage commun avec un autre déplacement déjà activé, alors il ne pourra pas être lancé et attendra dans l'état B.
- Il est possible de définir des zones sur le réseau qui ne peuvent pas être empruntées par plus d'un déplacement à la fois. Pour cela 8 zones sont disponibles et vous pouvez demander aux déplacements de vérifier si les zones en questions sont libres avant de les réserver et de les utiliser pour le déplacement.
- Enfin il est possible de définir 4 voies uniques et de les emprunter uniquement si on est dans le bon sens. Si les 2 derniers mécanismes vous semblent complexes, notez que la réservation des aiguillages du parcours suffit dans la majorité des cas à assurer la sécurité entre les trains se déplaçant en automatique. Au niveau sécurité il est préférable d'utiliser une entrée en finC pour détecter la fin d'un déplacement plutôt qu'une temporisation pour être sûr que le train a bien atteint sa destination supposée avant de libérer le parcours.

Si la sécurité est assurée entre les trains automatiques, en revanche, elle ne l'a pas du tout avec les trains fonctionnant en manuel et il est primordial aux conducteurs des trains manuels de faire attention. En cas de problèmes le bouton d'arrêt d'urgence est là pour éviter des catastrophes. En effet il est impossible pour Free-DCC de deviner la position des trains manuels et encore plus leurs intentions. Les plus astucieux pourront utiliser le mécanisme des zones pour réserver des zones en manuel en interdisant d'avoir des déplacements en automatique dans cette zone ou encore jouer avec la signalisation. Mais c'est assez difficile à faire. Notons tout de même que le lancement d'un déplacement ne peut se faire que si la locomotive spécifiée n'est pas déjà utilisée en manuel. Si cela est le cas, l'utilisateur devra la rendre en désactivant son adresse (mettre l'adresse à 0 avec l'écran traction).

Il a été décidé de lier un déplacement à une locomotive car le comportement dynamique est lié à un convoi particulier. En effet la vitesse, l'accélération et la décélération ne sont pas identiques pour chaque train ce qui peut poser des problèmes en automatique.

Pour programmer un déplacement, vous pouvez soit utiliser le logiciel de configuration RS232, soit utiliser l'éditeur intégré. Il est recommandé d'utiliser le logiciel de configuration car il est plus simple.

Pour ce faire, il suffit de modifier une des 50 lignes de déplacement du fichier de configuration par exemple :

```
depl_0: l10 fd10--- ff00--- a/--/..... z*..*.... u+..-
      v+12t10+24i30+10i5tf2au
```

Cela signifie :

Code	Commentaires
d0	Déplacement 0
l10	Locomotive d'adresse 10
fd10---	Fonctions au départ F0=1 F1=0 F2,3,4=inchangées
ff00---	Fonctions à la fin F0=0 F1=0 F2,3,4=inchangées
a/--/.....	Aiguillages 0 et 3 déviés / 1 et 2 directes / 4-31 inchangés
Z*..*....	Utilisation des zones 0 et 3
u+..-	Voie unique 0 sens 1 / 3 sens 0 / autres indifférent
v+12t10+24i30+10i5tf2au	vitA = 12 (avant) finA = tempo 1s (10*100ms) vitB = 24 (avant) finB = entrée 30 vitC = 10 (avant) finC = entrée 5 tempo finale = 200ms fin en arrêt d'urgence



Si vous utilisez l'éditeur intégré, il faudra utiliser les paramètres suivants

Ligne	Codage	Commentaires
0 - loco	0LLLLLLL	LLLLLLL=adresse de la locomotive (0-99)
1 - fct	D4321.F4321	D4321=fonctions au début / F4321=fonctions à la fin
2 - fct0	AU.D0.F0.M0/M4321	AU=Arrêt d'urgence / DFM0= début,fin,masque F0 / M4321 masques
3 - vitA	00.S.VVVVV	S=sens (1 pour avant) / VVVVV=vitesse (0-31)
4 - finA	T.NNNNNNN	T=1=tempo NNNNNNN=durée*100ms (0-127) T=0=entrée NNNNNNN=numéro entrée (0-63)
5 - vitB	00.S.VVVVV	S=sens (1 pour avant) / VVVVV=vitesse (0-31)
6 - finB	T.NNNNNNN	T=1=tempo NNNNNNN=durée*100ms (0-127) T=0=entrée NNNNNNN=numéro entrée (0-63)
7 - vitC	00.S.VVVVV	S=sens (1 pour avant) / VVVVV=vitesse (0-31)
8 - finC	T.NNNNNNN	T=1=tempo NNNNNNN=durée*100ms (0-127) T=0=entrée NNNNNNN=numéro entrée (0-63)
9 - tf	1.NNNNNNN	Tempo finale NNNNNNN=durée*100ms (0-127)
10 - msk aig 0-7	7.6.5.4.3.2.1.0	Masque aiguillages (0=non utilisé / 1=utilisé)
11 - pos aig 0-7	7.6.5.4.3.2.1.0	Position aiguillage (0=droit / 1=dévié)
12 - msk aig 8-15	15.14.13.12.11.10.9.8	Masque aiguillages (0=non utilisé / 1=utilisé)
13 - pos aig 8-15	15.14.13.12.11.10.9.8	Position aiguillage (0=droit / 1=dévié)
14 - msk aig 16-23	23.22.21.20.19.18.17.16	Masque aiguillages (0=non utilisé / 1=utilisé)
15 - pos aig 16-23	23.22.21.20.19.18.17.16	Position aiguillage (0=droit / 1=dévié)
16 - msk aig 24-31	31.30.29.28.27.26.25.24	Masque aiguillages (0=non utilisé / 1=utilisé)
17 - pos aig 24-31	31.30.29.28.27.26.25.24	Position aiguillage (0=droit / 1=dévié)
18 - zones	Z7.6.5.4.3.2.1.0	Z7=1=utilisation de la zone 7 / idem pour 6543210
19 - vu	S3210.M3210	S=sens de la voie unique / M=masque

Ces paramètres sont stockés dans la mémoire flash du microcontrôleur qui est limitée à 1000 écritures pour le PIC16F876 et 100000 pour le PIC16F876A. Il est donc possible que vous ne soyez plus en mesure de modifier ces paramètres après un grand nombre de modification sur le PIC16F876. Afin de réduire les écritures le logiciel n'écrit que ce qui est nécessaire en ayant pris soin de lire avant d'écrire pour voir si l'écriture est vraiment nécessaire.

#### 5.4.B. Le séquenceur

Le séquenceur quand à lui est principalement utilisé pour lancer les déplacements à des heures prédéfinies afin de créer une exploitation réaliste. Il peut également être utilisé pour modifier des sorties ou des LEDs ce qui permet par exemple de modifier l'éclairage de la maquette et des bâtiments en fonction de l'heure. Il n'est pas nécessaire de classer les enregistrements dans un ordre quelconque car à chaque minute de modélisme le microcontrôleur parcourt tous les enregistrements afin de définir les actions à effectuer. Le séquenceur permet de gérer jusqu'à 256 déclenchements qui présagent de nombreuses possibilités d'exploitation.

Les actions sont bien entendu déclenchées à des heures particulières. Pour plus de réalisme, il est aussi possible de définir quels jours de la semaine l'action sera lancée. En effet les circulations ne sont pas les mêmes chaque jour de la semaine. Cela vous autorise par exemple à réduire le trafic le week-end, à rajouter un train le mercredi ... Si vous souhaitez rester simple, vous pouvez désactiver la vérification du jour par les CC.

Pour une gestion scrupuleuse des circulations et correspondance, le séquenceur dispose d'une arme ultime qui lui permet d'arrêter le temps en cas de problème. dommage que cela n'existe pas dans la réalité ! Le séquenceur peut donc arrêter l'horloge si :

- un déplacement ne peut pas être lancé faute d'un canal d'exécution automatique libre. L'horloge sera réactivée lorsqu'un canal se libérera et que tous les déplacements pour l'heure en question seront lancés.
- un déplacement ne se finit pas dans les temps. Il est en effet possible de définir une durée d'exécution du déplacement, pour permettre d'assurer les correspondances, mais surtout pour éviter qu'un train immobilisé suite à un problème de captage de courant ou à un déraillement ne dérègle irrémédiablement les circulations.

Pour programmer un déplacement, vous pouvez soit utiliser le logiciel de configuration RS232, soit utiliser l'éditeur intégré. Il est recommandé d'utiliser le logiciel de configuration car il est plus simple.

Pour ce faire, il suffit de modifier une des 256 lignes de séquenceur du fichier de configuration par exemple :

```
seq_0: *****-- 10h31 d12
seq_1: -----** 10h35 b10 d15
seq_2: -----*- 10h40 b10h54 d16
seq_3: x
seq_4: ***** 22h30 out14=1
seq_5: ***** 06h30 out14=0
seq_6: *----- 08h20 led2=1
seq_7: fin
```

Cela signifie :

Code	Description
seq_0: *****-- 10h31 d12	Enregistrement 0 : La semaine (du Lundi au Vendredi), lancer le déplacement 12 à 10h31
seq_1: -----** 10h35 b10 d15	Le WE, lancée le déplacement 15 à 10h35 et bloquer l'horloge sur 10h35+10=10h45 si le déplacement n'est pas terminé
seq_2: -----*- 10h40 b10h54 d16	Le samedi, lancer le déplacement 16 à 10h40 et bloquer l'horloge à 10h54 si le déplacement n'est pas terminé
seq_3: x	L'enregistrement 3 est non utilisé (vous pouvez aussi ne pas mettre le x)
seq_4: ***** 22h30 out14=1	Tous les jour à 22h30n activer la sortie 14 (surement pour allumer l'éclairage publiques)
seq_5: ***** 06h30 out14=0	Tous les jour à 6h30, désactiver la sortie 14 (surement pour éteindre l'éclairage publiques)
seq_6: *----- 08h20 led2=1	Le lundi à 8h20, allumer la led 2
seq_7: fin	L'enregistrement 7 indique la fin de la table (il n'est pas indispensable)

Si vous utilisez l'éditeur intégré, il faudra utiliser les paramètres suivants

Pour un déplacement :

Ligne	Codage	Commentaires
0 - heure	Lu.Ma.Me.HHHHH	Lu=Lundi Ma=Mardi Le=Mercredi HHHHH=Heure (0-23)
1 - minute	Je.Ve.MMMMM	Je=Jeudi Ve=Vendredi MMMMM=Minute (0-59)
2 - déplacement	Sa.Di.DDDDD	Sa=Samedi Di=Dimanche DDDDD=Déplacement (0-49)
3 - blocage	0.TTTTTT	TTTTTT=Délais avant blocage en minutes (0-127)

Pour une sortie ou une LED :

Ligne	Codage	Commentaires
0 - heure	Lu.Ma.Me.HHHHH	Lu=Lundi Ma=Mardi Le=Mercredi HHHHH=Heure (0-23)
1 - minute	Je.Ve.MMMMM	Je=Jeudi Ve=Vendredi MMMMM=Minute (0-59)
2 - déplacement	Sa.Di.N876543	Sa=Samedi Di=Dimanche N=numéro de la variable
3 - blocage	1.000.val.N210	Val=valeur N210=suite du numéro de la variable

Le numéro de la variable à activer ou désactiver N876543210 (0-511) est décrit dans le chapitre sur les équations logiques.

Par exemple pour la sortie 14, il s'agit du numéro 334.

$N = 320 + \text{numéro de la sortie}$

$N = 128 + \text{numéro de la LED}$

HHHHH est aussi utilisée pour les indications suivantes :

- 30 rend un déclenchement inactif
- 31 indique la fin de la table des déclenchements, mais il n'est pas nécessaire de la mettre.

## 5.5. Les équations logiques

Les équations logiques permettent de positionner les sorties, aiguillages et leds en fonction des entrées comme les détecteurs, boutons poussoirs ... Elles sont donc particulièrement bien adaptées pour la signalisation, la réalisation de TCO, ou la commande du système rendent le système extrêmement flexible car il n'y a plus besoin d'utiliser des relais ou des circuits logiques.

Les équations logiques manipulent des variables binaires qui valent donc 0 ou 1. Free-DCC expose les variables suivantes :

Nom de la variable	acces	numero	Description
aig_cmd_<0-31>	rw*	0-31	cmd des aiguillages (0=direct, 1=dévié). Si l'aiguillage est réservé, il n'est pas possible de changer cette variable
aig_pos0_<0-31>	r-	32-63	aiguillage en position directe si 1
aig_pos1_<0-31>	r-	64-95	aiguillage en position dévié si 1
aig_res_<0-31>	r-	96-127	réservé (par les déplacements automatiques)
iti_<0-31>	-w	448-479	l'accès a cette variable positionne l'itinéraire (si les aiguillages ne sont pas réservés)
led_<0-63>	rw	128-191	
led_cli_<0-63>	rw*	192-255	si 1, la led clignotera lorsqu'elle sera activée avec led_on
in_<0-63>	r-	384-447	Entrées
in_edge_<0-63>	r-	256-319	passé à 1 si ancienne lecture de l'entrée est à 0 et la nouvelle à 1
out_<14-31>	rw	320-351	sorties (0-13 ne sont pas disponibles car utilisées pour la cmd des aiguillages)
bi_<0-15>	rw	352-367	variables libres pour l'utilisateur
res_<0-15>	rw	368-383	variables de réservation de zones et voies uniques (explications dans le chapitre des déplacements)

r=lecture possible

w=écriture possible

\*=ces variables sont initialisées à la mise sous tension à partir des variables de configurations CC de la centrale

La position des aiguillages est indiquée par 2 variables, car lorsque l'aiguillage est en déplacement ou en attente de déplacement il ne doit pas être considéré comme étant directe ou dévié mais non disponible. Il est donc préférable de tester uniquement la valeur 1 de ces variables. Un groupe de déplacement supplémentaire led\_phase\_<0-63> permet d'inverser le clignotement des leds. Ce groupe n'est pas accessible par les variables car il n'est pas nécessaire de changer la phase en cours d'exécution. Les variables CC permettent néanmoins d'initialiser ces variables à la mise sous tension..

Les équations utilisent des opérateurs logiques pour manipuler les variables. Ces opérations se font la plupart du temps entre une variable de travail que nous appellerons accumulateur acc et les autres variables. Le tableau suivant décrit ces opérateurs :

format	Description	numéro
l <var>	chargement de acc avec var	0
ln <var>	chargement de acc avec le complément de var (0 si 1 et inversement)	4
a <var>	ET logique entre acc et var avec mise du résultat dans acc	8
an <var>	ET logique entre acc et le complément de var avec mise du résultat dans acc	12
o <var>	OU logique entre acc et var avec mise du résultat dans acc	16
on <var>	OU logique entre acc et le complément de var avec mise du résultat dans acc	20
x <var>	OU EXCUSIF entre acc et var avec mise du résultat dans acc	24
xn <var>	OU EXCUSIF logique entre acc et le complément de var avec mise du résultat dans acc	28
= <var>	affectation de la valeur de acc à var	32
=s <var>	mise à 1 de var si acc vaut 1	36

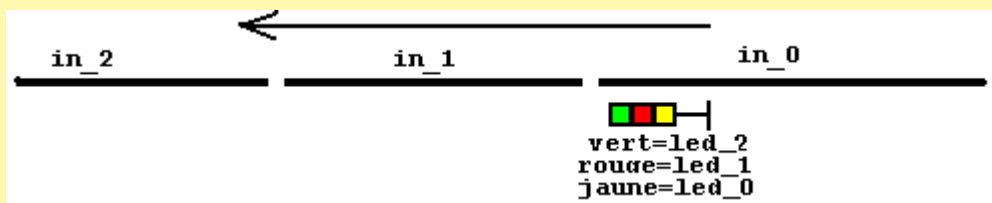
=c <var>	mise à 0 de var si acc vaut 1	40
=x <var>	inversion de var si acc vaut 1	44
Inv	inversion de acc	48
Cpt <valeur>	met à 1 acc si le compteur routier vaut la valeur, sinon met acc à 0	52
Nop	ne fait rien	56
end	Fin des équations (pour gagner en rapidité)	60

ET logique = resultat à 1 si variable et acc sont toutes 2 à 1, sinon resultat = 0  
OU logique = resultat à 1 si variable ou acc ou les 2 sont à 1, sinon resultat = 0  
OU EXCLUSIF = resultat à 1 si seulement une des 2 variables est à 1

Free-DCC autorise 256 opérations. Afin d'être très réactif, la totalité des opérations est exécutée toutes les 100ms. Les équations sont stockées dans la mémoire flash du microcontrôleur. Pour les transférer dans cette mémoire, le plus simple est d'écrire les équations dans un fichier texte puis de les transférer via l'interface RS232 par le logiciel fourni à cet effet. Il est également possible de modifier cette mémoire avec le menu édition de la centrale, mais c'est vraiment plus difficile (le 1ere octet et l'ordre augmenté de 1 si la variable est supérieure ou égale à 256, le second octet est le numéro de la variable diminué de 256 si elle est supérieure ou égale à 256).

Ce début de chapitre vous laissera sûrement dubitatif à sa première lecture. Fin d'y remédier les exemples suivants illustrent la facilité d'utilisation et la puissance de ces équations

Exemple 1 : Feu de block automatique lumineux. Il s'agit dans cet exemple de piloter les 3 leds (verte, rouge et jaune) d'un feu de block. Il doit donc être rouge quand le canton précédent est occupé, jaune lorsqu'il est libre mais que le suivant est occupé et vert lorsque les 2 sont libres. Les trains sont détectés sur les différents cantons grâce à des détecteurs de courants reliés aux entrées in\_0 à in\_2



! in\_1 = led\_1  
!n in\_1 a in\_2 = led\_0  
!n in\_1 an in\_2 = led\_2

Le feu est géré simplement à l'aide de ces 3 équations (8 opérations).

La première équation allume la led rouge si le canton de in\_1 est occupé ...

Le système est suffisamment dimensionné car la capacité de 256 opérations permettraient donc de gérer 32 feux. (256/8). (Ce qui est faux car les 64 leds ne peuvent former que 21 feux à 3 leds (64/3=21)).

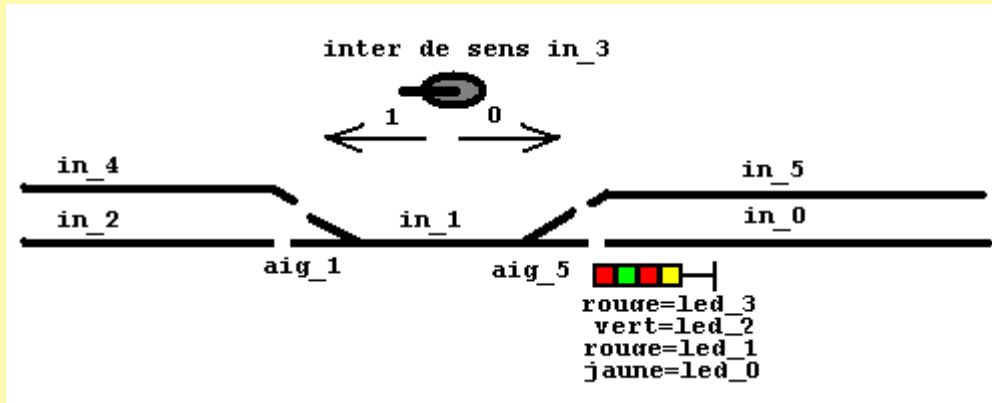
Il est bien entendu possible de complexifier ce cas d'école par exemple dans le cas d'une voie unique en ajoutant un interrupteur sur le TCO pour fixer le sens de parcourt. Relions cet interrupteur à in\_4 et disons qu'il vaut 1 dans le sens bon sens et 0 en sens inverse. Les équations seraient alors :

! in\_1 on in\_4 = led\_1  
! in\_2 an led\_1 = led\_0  
!n led\_1 an led\_0 = led\_2

Il serait également possible d'ajouter une zone d'arrêt au droit du feu afin d'empêcher un train de franchir le signal au rouge. Pour ce faire, il suffit de couper l'alimentation de la zone d'arrêt par un relais commandé par une sortie. Par exemple si ce relais est connecté sur out\_14 et que son activation coupe la section d'arrêt, il devra être actif en même temps que la led rouge. La fin de la première équation devient alors ... = led\_1 = out\_14

Exemple 2 : Feu en fonction de la position des aiguillages en pointe et talon sur voie unique. Le feu présente le carre (2 leds rouges) si le sens est à 0, si un train est sur in\_1 ou si l'aiguillage 5 est dévié. Sinon il présente le

jaune si un train se trouve sur in\_2 si aig\_1 est directe ou sur in\_4 s'il est dévié. Dans le cas contraire, le feu est vert.

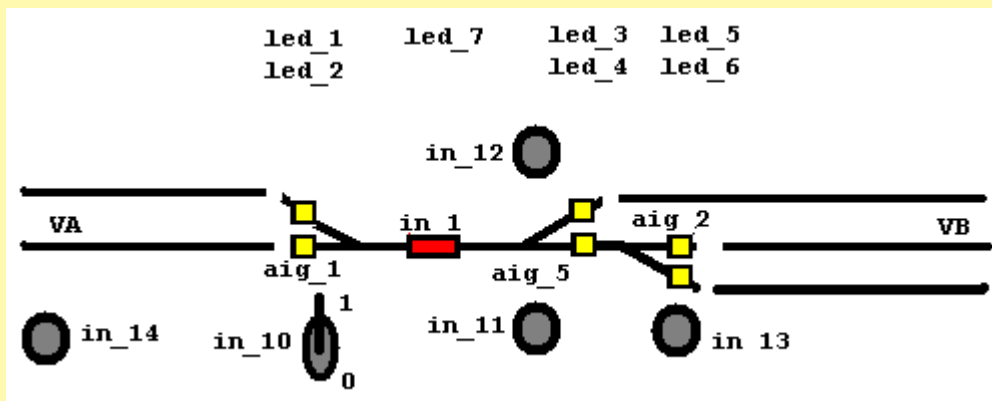


```

l in_1 on aig_pos0_5 on in_3 = led_3 = led_1
l in_2 a aig_pos0_1 = bi_0
l in_4 a aig_pos1_1 = bi_1
l bi_1 o bi_0 an led_3 = led_0
l in led_1 an led_0 = led_2
  
```

Vous remarquerez l'utilisation de variables utilisateur bi\_0 et bi\_1 afin de stocker des résultats intermédiaires. La cas peut être encore complexifié en mettant le feu au rouge ou jaune si aig\_1 était en cours de manœuvre. Ln aig\_pos0\_aig1 an aig\_pos1\_aig1 = bi\_2 ...

Exemple 3 : cet exemple présente un extrait de TCO et montre diverses techniques de commande d'aiguillages. La led\_7 s'allume si un train est détecté sur le canton in\_1  
L'aiguillage 1 se commande avec un interrupteur, le 5 avec 2 boutons poussoirs et le 2 avec un seul. La position des aiguilles est répétée grâce à des diodes.  
Le bouton poussoir in\_14 permet d'établir un itinéraire entre les voies VA et VB

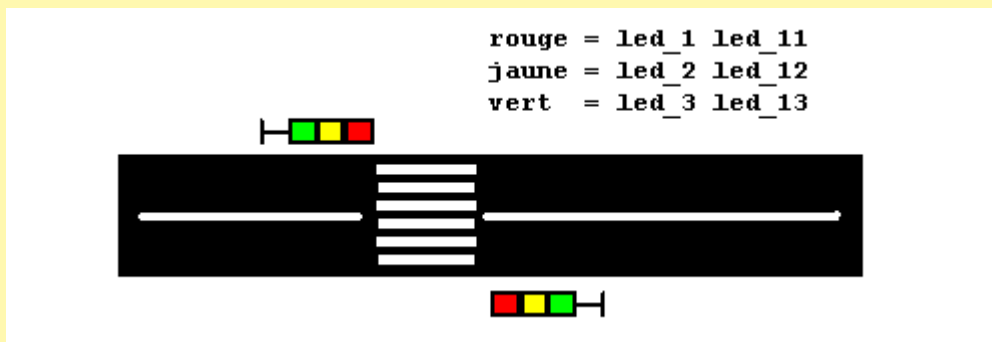


```

l in_1 = led_7
l in_10 = aig_cmd_1
l in_11 =c aig_cmd_5
l in_12 =s aig_cmd_5
l in_edge_13 =x aig_cmd_2
l in_14 =c aig_cmd_1 =c aig_cmd_5 =c aig_cmd_2
l aig_pos0_1 = led_2
l aig_pos1_1 = led_1
l aig_pos0_5 = led_4
l aig_pos1_5 = led_3
l aig_pos0_2 = led_5
l aig_pos1_2 = led_6
  
```

L'itinéraire ne fonctionnera pas car l'aiguillage 1 sera toujours positionné suivant la position de l'interrupteur. Pour cette raison, il est préférable d'utiliser des boutons poussoirs. Cet itinéraire pourrait aussi utiliser un des 32 itinéraires à définir. Par exemple avec l'itinéraire 5, l'équation devient :  $! in\_edge\_14 =s iti\_5$ . Les déplacements automatiques sont prioritaires sur la commande des aiguillages et il n'est pas possible de les modifier lors d'un déplacement automatique car les aiguillages des déplacements sont réservés. Pour plus de sécurité vous pouvez modifier les équations autoriser la manœuvre des aiguillages uniquement lorsque aucun train ne se trouve sur in\_1 ... Vous voyez que réaliser un TCO est un jeu d'enfant !

Exemple 4 : Le passage piéton. De plus en plus de modélistes ajoute un carrefour routier avec des feux tricolores pour animer leur maquette. Free-DCC autorise ceci grâce à l'ajout d'un compteur qui s'incrémente chaque seconde. Il suffit d'indiquer dans les variables CC la valeur de rebouclage et d'utiliser les comparaisons avec les équations. Pour faire simple nous étudierons le cas d'un passage piéton devant notre gare afin que les voyageurs traversent en toute sécurité. Le feu sera rouge pendant 10s, vert pendant 20s et jaune pendant 1s. La valeur de rebouclage est donc de 30. Le compteur comptera de 0 à 30



```

cpt 0 =s led_1 =c led_2 =c led_3
cpt 10 =c led_1 =s led_3
cpt 30 =c led_3 =s led_2
! led_1 = led_11
! led_2 = led_12
! led_3 = led_13

```

Je vous laisse le soin de rajouter le bonhomme lumineux. Amusez vous aussi à trouver les équations d'un véritable carrefour !

Vous en savez maintenant assez sur les équations pour animer votre réseau.

## 5.6. Le logiciel de configuration par RS232

Ce petit logiciel permet de configurer ou lire la configuration de la centrale par le biais de la liaison série RS232. Il permet d'accéder aux zones suivantes :

- L'EEPROM de configuration où sont stockées les variables CC de configuration de la centrale.
- La table des itinéraires en mémoire flash (32 itinéraires)
- La table des équations logiques (256 éléments d'équation)
- La table des déplacements (50 déplacements)
- La table du séquenceur (256 déclenchements)

Les CC sont stockées dans la petite EEPROM tandis que les tables sont stockées dans la grosse mémoire flash. Si l'écriture dans l'EEPROM est quasiment illimitée pour nous, l'endurance typique de la flash varie de 1000x pour les 16F876 à 100000x pour les 16F876A. Les 16F876 peuvent donc montrer des signes de fatigue après 1000 programmations, dans ce cas il conviendra donc de procéder au remplacement. Bien entendu le logiciel vérifie les écritures pour détecter tout problème. De même il lit avant d'écrire pour éviter d'écrire la même chose ce qui est inutile et diminue la durée de vie de la flash.

Le logiciel `fdcc_conf.exe` est un simple exécutable qui une fois extrait de l'archive peut être mis où bon vous semble et lancé en cliquant simplement dessus. Vous avez ensuite le choix entre une lecture qui sauvera toutes la configuration de la centrale dans le fichier « `fdcc_conf_lect.txt` » dans le répertoire où vous avez lancé le logiciel. Ou alors vous pouvez choisir une écriture qui mettra à jour la centrale avec le contenu du fichier « `fdcc_ecr_lect.txt` ». Si vous voulez sauvegarder une configuration particulière ou utiliser plusieurs fichiers, alors faites des copier/coller du fichier « `fdcc_conf_ecr.txt` ». La suite de ce document détaille le format de ce fichier.

Des commentaires précédés d'un # peuvent être ajoutés dans le fichier d'écriture. Bien entendu, ils n'apparaîtront pas dans le fichier de lecture.

Les différents éléments de configuration doivent être dans des sections.

- [iti] pour mes itinéraires
- [eq] pour les équations
- [depl] pour les déplacements
- [seq] pour les entrées du séquenceur

Pour désactiver une section il suffit simplement de lui donner un nom différent.

Les CC peuvent également être modifiées via l'interface de la centrale en utilisant le menu de configuration. Les tables quand à elles peuvent être modifiées en utilisant le menu d'édition, mais il faut connaître avec précision le codage en mémoire de ces tables (Il est décrit dans les chapitres précédents). Sauf cas de force majeure (pas de PC disponible) ou pour de toutes petites modifications, je conseille d'utiliser le logiciel à la place du menu édition de la centrale. En effet il est bien plus simple à utiliser et évitera de nombreuses erreurs.

### 5.6.A. Les CC

Il suffit de spécifier le numéro de la variable CC ainsi que sa valeur.

Ex :

```
[cc]
cc_0: 9
cc_1: 10
```

Les CC non spécifiés ne seront pas modifiés.

### 5.6.B. Itinéraires

Numéro de l'itinéraire suivi de la position de l'aiguillage 0 puis du 1 ... et ceci jusqu'au 31.

Les positions peuvent être

- « - » pour position directe
- « / » pour position déviée
- « . » pour indiquer que l'aiguillage ne fait pas partie de l'itinéraire

Pour améliorer la lisibilité des blancs peuvent être insérés. Les itinéraires doivent être dans la section [iti].

Ex :

```
[iti]

# tous les aiguillages en position directe
iti_0: -----
# aig0 et 3 en position directe / aig1 et 2 en position déviée
iti_1: -//-.....
```

Les itinéraires non spécifiés ne seront pas modifiés.

### 5.6.C. Equations

Il suffit de les écrire comme indiqué au chapitre 5.5. Ils doivent être dans une section [eq]

Ex :

```
[eq]

l in_1 = led_1
ln in_1 a on_2 = led_0
ln in_1 an in_2 = led_2
end
```

Il est conseillé de terminer la liste des équations par end pour gagner en rapidité.

Les équations logiques restantes ne seront pas affectées mais ne devrait pas être exécutées si vous utilisez end.

En lecture chaque élément d'équation est précédé de « eq\_<numero>: » (En écriture ce préfix ne gêne pas)

### 5.6.D. Les déplacements

Il suffit de les écrire comme indiqué au chapitre 5.4.A. Ils doivent être dans une section [depl].

Ex :

```
[depl]

depl_0: l10 fd10--- ff00--- a/--/..... z*...*.... u+...-
      v+12t10+24i30+10i5tf2au
```

Le déplacement doit tenir sur une seule ligne. Attention a ne pas mettre d'espace dans les différents groupes.

Les déplacements non spécifiés ne seront pas modifiés.

### 5.6.E. Les déclenchements du séquenceur

Il suffit de les écrire comme indiqué au chapitre 5.4.B. Ils doivent être dans une section [seq].

Ex :

```
[seq]

seq_0: ***** 10h31 d12
seq_1: -----** 10h35 b10 d15
seq_2: -----*- 10h40 b10h54 d16
seq_3: x
seq_4: ***** 22h30 out14=1
seq: ***** 06h30 out14=0
seq: *----- 08h20 led2=1
seq : fin
```

Le numéro de déplacement est purement anecdotique et vous pouvez le remplacer par « seq: »

Il vaut mieux mettre fin à la fin des déplacements



## 5.7. Pilotage de la centrale par RS232

La centrale offre la possibilité d'être reliée à un ordinateur par l'intermédiaire de l'interface RS232. Cela permet à un logiciel sur le PC de faire office d'un TCO, de piloter tout ou en partie le réseau ou encore de configurer la centrale.

Pour ce faire la centrale dispose de 2 protocoles de communication.

- Le protocole standard Marklin 6050 permet à de nombreux logiciels existants de piloter la centrale. Ou pourquoi pas de créer vous-même un logiciel capable de la piloter.
- Le protocole propriétaire que j'ai codé pour mes propres besoins. Il peut être également utilisé pour piloter la centrale, mais également pour la configurer et la déboguer.

Le choix du protocole se fait par CC. Il est à noter que si vous démarrez avec le bouton d'arrêt d'urgence appuyé, alors le mode passe en propriétaire et la vitesse à 19200bps. Ceci pour vous permettre de configurer la centrale même lorsque le mode standard est configuré. Les CC permettent également de configurer la vitesse de la liaison série qui peut être de 2400bps / 4800bps / 9600bps ou 19200 bps. Si vous n'avez plus de liaison série sur votre ordinateur, vous pouvez toujours utiliser un câble convertisseur USB/RS232 que l'on trouve à une vingtaine d'euros chez tout bon revendeur informatique.

### 5.7.A. Le protocole standard Marklin 6050

Ce protocole utilisé depuis longtemps pour piloter la centrale Marklin 6050 qui reposée alors sur le protocole Motorola et non DCC est devenu au fil du temps un standard de fait et de nombreux logiciels l'utilisent. Il est très simple à utiliser et permet de:

- Contrôler 80 locomotives en sens / vitesse / fonctions F0-F4
- Contrôler les aiguillages et sorties
- Lire l'état des entrées des modules S88
- Gérer l'arrêt d'urgence

Le tableau suivant résume ce protocole :

Titre	Codage 1er octet	2eme octet	Commentaires
Vitesse	000.F0.VVVV	Num CH	VVVV=0-14=vitesse VVVV=15=inv de sens
Fonctions	010.0.F4.F3.F2.F1	Num CH	CH=1-16
Aig/Out/Led	001.000.R.V	Num Aig/Out	R=rouge/dévié/off V=Vert/droit/on
AU	011.0000.AU		AU=1=Arrêt d'urgence
Lecture N S88	100.NNNNN	-> entrées	NNNN=Nb de S88 à lire (0=mode sans eff)
Lecture 1 S88	110.NNNNN	-> entrées	NNNN=Num du S88 à lire (0=mode avec eff)

Arrêt d'urgence :

Pour l'arrêt d'urgence il suffit d'envoyer #01100001 soit 97 en décimal.

Pour revenir en mode normal il suffit d'envoyer #01100000 soit 96 en décimal.

De nombreux logiciels commence par envoyer 2 arrêt d'urgence pour se synchroniser, font leurs initialisations puis envoient 1 mode normal.

## Vitesse/Sens/F0 :

Le premier octet indique la vitesse, l'état de la fonction 0 (souvent l'éclairage) et le sens.

La vitesse est la suivante :

0 = arrêt

1 = cran de vitesse 1/14

14 = cran de vitesse 14/14

Bien entendu, la centrale adapte ces crans au système DCC (4-31)

Pour le sens, il n'y a pas de bit consacré mais seulement un ordre d'inversion lorsque l'on envoie la commande 15. A l'initialisation de la centrale tous les sens sont en avant, ensuite il convient au logiciel sur PC de mémoriser le sens courant et de le modifier si besoin.

Pour activer la fonction 0, il suffit d'ajouter 16 à la commande

Par exemple la commande 26 = #00011010 indique F0=on / vitesse = cran 10/14

Le second octet indique le numéro de la locomotive et le protocole définit 80 locomotives (1-80). Ce nombre est réduit à 16 (1-16) avec Free-DCC et spécifie le canal Free-DCC utilisé et non l'adresse de la locomotive. Cela permet de changer l'adresse par la centrale. Le canal 0 de Free-DCC correspond à l'adresse 16. Le 1 à l'adresse 1 ... Le 15 à l'adresse 15. Par exemple si le canal 1 de Free-DCC est configuré avec la locomotive d'adresse 23, alors #00011010+#00000001 signifie F0=on / vitesse = cran 10/14 pour locomotive du canal 1, c'est à dire la locomotive d'adresse 23 !

## Fonctions F1-F4

Cette commande permet de jouer avec les fonctions spéciales 1 à 4.

Par exemple #01001011+#00000001 active les fonctions 4, 2, 1 et désactive la fonction 3 de la locomotive qui se trouve sur le canal 1.

## Lecture des modules de feedback S88

Le standard permet de lire 31 modules de 16 entrées, Free-DCC limite ce nombre à 4, ce qui fait tout de même  $4*16=64$  entrées. Le mode de lecture peut être choisi avec ou sans effacement. (#1000000=sans effacement, #11000000=avec effacement). Cela signifie que l'état de l'entrée est remise à 0 après lecture et non conservée. Sauf cas particulier, il convient de choisir le mode avec effacement pour lire la valeur courante. Le mode par défaut se choisit par les CC.

Ensuite à vous de choisir si vous voulez lire :

- un module en particulier avec #110NNNNN avec NNNNN = numéro du module (1-4)

- un nombre spécifié de module avec #100NNNNN avec NNNNN = nombre de module (1-4). 1 pour le premier module, 2 pour les 2 premiers modules ...

En général les logiciels lisent tous les modules à la fois puis interagissent sur les locomotives, aiguillages et sorties en fonction de ces entrées.

## Commande des aiguillages, sorties et LEDs

Le premier octet définit l'état d'une ou de 2 sorties complémentaires #001000RV puis le second indique le numéro de la sortie.

Avec de vrai décodeur d'accessoires DCC, les sorties sont souvent utilisées 2 à 2 comme suit :

- Pour commander un aiguillage, une bobine est placée sur la première sortie et la seconde sur la 2eme sortie. -

- Pour les feux, la led verte est placée sur la première tandis que la rouge est placée sur la seconde

R.V=1.0 est donc utilisé pour mettre l'aiguillage en position direct ou le feu au vert

R.V=0.1 est utilisé pour mettre l'aiguillage en position dévié ou le feu au rouge

R.V=0.0 est utilisé pour arrêter l'impulsion sur l'aiguillage ou éteindre les 2 LEDs. 0.0 est de moins en moins utilisé car les décodeurs d'accessoires permettent maintenant de définir la longueur des impulsions pour les aiguillages et il n'y a aucun intérêt à éteindre toutes les led d'un feu

Cette façon de coupler 2 sorties n'a aucun intérêt avec Free-DCC hormis pour les feux, si vous utilisez uniquement des feux à 2 leds. Les accessoires sont le point faible de la norme DCC car il faut beaucoup de temps pour en changer un grand nombre, ce PB se retrouve sur ce protocole car il faut envoyer 2 octets pour changer une sortie alors qu'avec ces 16 bits, on aurait pu en changer 16 comme dans le cas des entrées.

Le tableau suivant indique les accessoires commander avec Free-DCC en fonction de l'adresse

Type	Adresses	Nb	Fonctionnement (octet 1 & octet 2)
Aiguillages	0-31	32	00100001 & 0 -> aig0 direct pas besoin d'envoyer 00100000 pour 00100010 & 0 -> aig0 dévié stopper l'impulsion car automatique
Sorties	78-95	18	00100001 & 78 -> out14=0 00100010 & 78 -> out14=1
LEDs	128-191	64	00100001 & 128 -> led0=0 00100010 & 128 -> led0=1
LEDs couplées	192-255	2*32	00100001 & 192 -> led0=0 led1=1 le fonctionnement des 2 leds est lié 00100010 & 192 -> led0=1 led1=0 la 1ere led est celle de l'adresse, 00100000 & 192 -> led0=0 led1=0 la seconde la suivante. 00100001 & 194 -> led2=0 led3=1 00100010 & 194 -> led2=1 led3=0 00100000 & 194 -> led2=0 led3=0

Voilà, vous savez tout sur ce protocole très simple à utiliser. Il constitue un point d'entrée très accessible pour les personnes qui veulent coder leur propre logiciel de contrôle. On trouve d'ailleurs sur Internet quelques exemples.

### 5.7.B. Le protocole propriétaire

Ce protocole permet de contrôler la centrale, la configurer et la déboguer. Il est vraiment spécifique et ne vous servira pas à grand-chose donc je ne le décris pas. Pour le contrôle il est plus efficace que le précédent car son mode transfert de masse permet de piloter toutes les fonctions intéressantes et lire les états en 72 échanges. Si vous voulez l'utilisez envoyer quelques 0 pour la synchro puis 1 pour voir si la centrale répond 0x83 puis la commande 9 pour chaque transfert de masse dont les échanges sont décrits si dessous.

```

;=====
; Moniteur RS232
;
; 0: sync
; 1: test      ->0x83
; 2: setadr   + adr
; 3: write    + dat
; 4: read     ->dat
; 5: inc+write + dat
; 6: inc+read ->dat
; 7: inc+read+clear ->dat
; 8: mode     + dat (bit0=IRP)
; 9: transfert de masse 72x + dat -> dat (voir spécificités dans rs_cmd_14)
;
; attention: au niveau du PC:
; - A 19200 bps, rajouter lms entre l'envoi de 2 paquets,
;   sinon UART micro plante et micro écrit n'importe ou dont
;   en rs_config ce qui peut provoquer une écriture flash ds le programme !
;   c'est pourquoi les écritures sont maintenant protégées !
;   pas de PB avec les vitesses plus lentes
; - entre une émission et réception sur mon PC, j'ai note un besoin
;   de 4ms a 19200bps et 8ms a 9600bps
;
;=====
; Transfert de masse (72 échanges)
;
; PC->UC: ADR(16)+CMD(16)+FCT(16)+LED(8)+CLI(8)+AIG(4)+OUT(4)
; UC->PC: IN_MEM(8)+AIG_POS(8)+MOUSE_W(16)+can%8_pour_le_reste
;
;=====

```

## 6. Exemple

Cette partie présente Free-DCC dans un cas concret. Je prends bien évidemment comme exemple mon réseau.

J'ai choisis de réaliser un réseau modulaire afin de

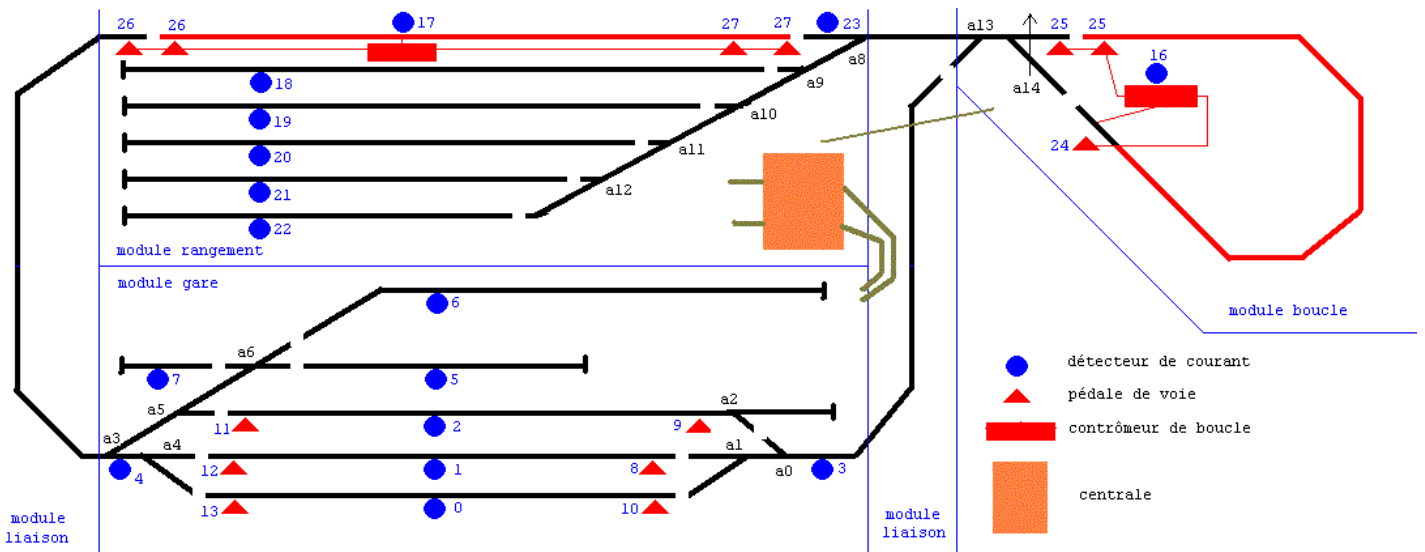
- construire les modules facilement
- le ranger facilement (un module inutilisé peut même décorer un meuble ou une étagère)
- le déplacer facilement (pour cela, il est nécessaire de ne pas dépasser 1.80m x 50cm par module)
- le faire évoluer sans modifier l'existant.
- configurer le réseau différemment (par exemple on peut associer le module gare à celui coulisse de rangement pour exploiter la gare comme une gare terminale.)

Pour l'instant le seul module destiné à être vu est le module gare. Mais pour faire circuler des trains, il est nécessaire d'utiliser d'autres modules « techniques ». Le réseau a été conçu de telle manière qu'un train sortant peut re-renter par n'importe quel coté en étant retourné ou pas. Il peut également disparaître dans la coulisse de rangement. Ceci autorise d'énormes possibilités de jeux.



Photo du module gare rangé pour décoration

Le plan suivant présente mes différents modules assemblés. D'autres configurations sont possibles, comme une gare terminale en utilisant les modules gare et rangements dans un minimum de profondeur.



Afin de ne pas faire dépendre mon réseau d'une technologie ou d'une autre, je câble les cantons, pédales de voies et aiguillages sur des connecteurs informatiques à 25 broches (DB25). Il est ainsi possible de l'utiliser en DCC en le connectant sur la centrale ou en analogique en branchant ces connecteurs sur une autre « boîte ».

J'ai défini le brochage suivant pour ces connecteurs :

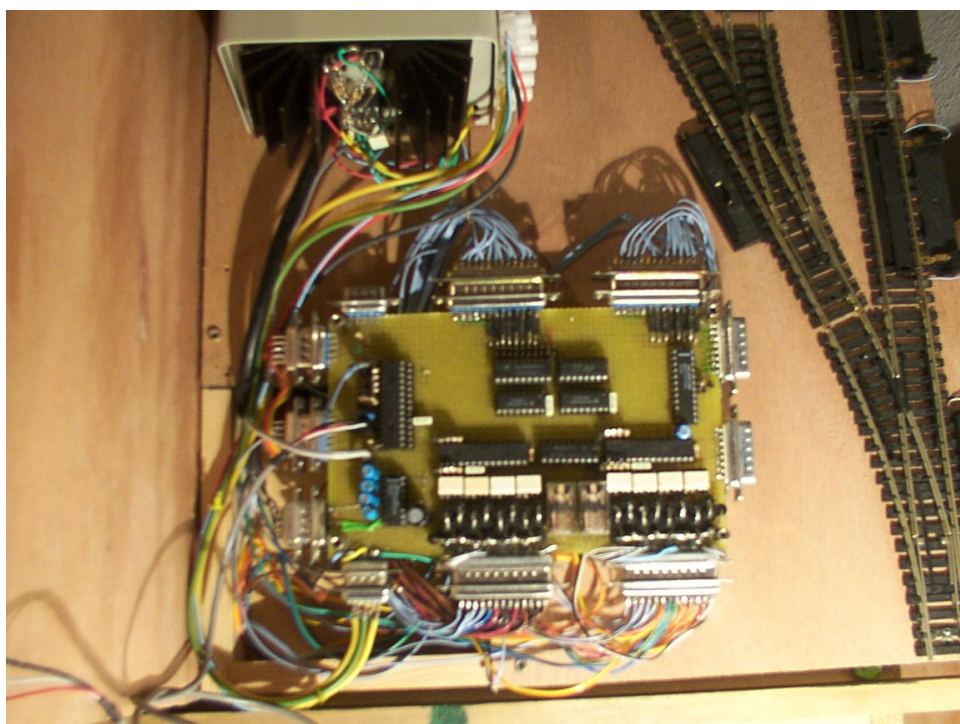
#### DB25 des voies

Pattes	Description
1 et 14	Canton 0
2 et 15	Canton 1
3 et 16	Canton 2
4 et 17	Canton 3
5 et 18	Canton 4
6 et 19	Canton 5
7 et 20	Canton 6
8 et 21	Canton 8
9	Commun des capteurs (ILS ou pédales)
10	Capteur 0
11	Capteur 1
12	Capteur 2
13	Capteur 3
22	Capteur 4
23	Capteur 5
24	Capteur 6
25	Capteur 7

#### DB25 des aiguillages

Pattes	Description
1-8	Bobines directes
14-21	Bobines déviées
9	Non utilisé
10	Commun des bobines de l'aiguillage 0
11	Commun des bobines de l'aiguillage 1
12	Commun des bobines de l'aiguillage 2
13	Commun des bobines de l'aiguillage 3
22	Commun des bobines de l'aiguillage 4
23	Commun des bobines de l'aiguillage 5
24	Commun des bobines de l'aiguillage 6
25	Commun des bobines de l'aiguillage 7

J'ai choisi de regrouper toute l'électronique sur une seule carte afin de réduire les connections entre modules électroniques. Ma photo suivante montre cette carte « prototype » ainsi que les fameux connecteurs DB25



Comme le montre le plan, 2 ensembles de fils munis de connecteurs DB25 sortent des moules gare et coulisse de rangement. A chaque fois un ensemble est dédié aux voies et pédales de voie tandis que l'autre sert aux aiguillages. Le module de retournement se branche sur le module de coulisse. Chaque canton de voie est relié à un détecteur de courant inclus sur la carte. Les numéros sur le plan indiquent les numéros des entrées S88.

Ce plan implique l'utilisation de 2 boucles de retournement.

- Celle du module boucle est destinée soit à retourner les trains, soit à servir de tiroir pour les rapatrier en zone de rangement. Il n'y a donc pas besoin de parcourir la boucle dans les 2 sens, c'est pourquoi l'aiguillage aig14 n'est pas commandé et maintenu en position directe. Pour retourner les trains seuls les pédales d'entrée et de sorties servent. Par contre, lorsqu'il n'est pas nécessaire de les retourner la seconde pédale 25 permet de réalimenter la boucle dans le bon sens si on se sert de la pédale 24 comme signal d'arrêt avant rebroussement.
- Celle du module de rangement doit quand à elle être parcourue dans les 2 sens, ce qui implique l'utilisation de 4 détecteurs.

Les contrôleurs de boucles sont aussi inclus sur ma carte principale.

Je n'ai pas encore utilisé la signalisation et les sorties, mais à terme quelques feux de signalisation assureront leur rôle de protection. Grâce aux différents capteurs d'occupations, il sera facilement possible d'avoir une signalisation cohérente. J'utiliserai également des sorties pour éclairer la gare et les lampadaires des quais durant le service nocturne.

Les capteurs de courant servent à la signalisation à la répétition des occupations sur le TCO et aux déplacements automatiques. Dans le cas de ces déplacements, les pédales de voies, sont d'une aide précieuse pour assurer des arrêts précis. Il est préférable de les placer 15cm avant la zone d'arrêt souhaité et de ne pas arriver trop vite dessus. Pour ma part, j'utilise des pédales Jouef, bien que volumineuse, elles ne sont pas 100% fiables pour détecter le matériel récent qui dispose de roue à boudin fin et j'envisage de les remplacer par des détecteurs de courant. Pour les positionnements précis vous pouvez également utiliser des ILS.

L'électronique est une des nombreuses disciplines qui compose le modélisme ferroviaire, et rien ne sert d'avoir une superbe électronique si le reste n'est pas à la hauteur. En effet manoeuvrer sur un réseau avec un plan de voie mal conçu (manoeuvres trop compliquées, pas assez de possibilités, voies trop courtes ...) ou sur un réseau ou aucune ambiance réaliste se dégage vous lassera vite. Il est donc nécessaire, de bien étudier le plan des voies, trouver des justifications d'exploitation plausibles... Et également réaliser avec soin !

Pour ma part, j'ai choisi de réaliser un module avec une petite gare de campagne desservant un village de taille moyenne. Cette gare est implantée sur une ligne à voie unique afin de décupler l'intérêt du jeu. En effet il est bien plus intéressant d'exploiter une gare de voie unique car elle nécessite le croisement et le dépassement des trains. De plus les installations minimales d'une telle gare sont plus facile à reproduire et cela fait plus vrai. Cette gare est conçue pour être modulable à souhait afin de s'adapter à de nombreuses possibilités de jeu. Dans cette optique, l'utilisation d'un quai centrale outre le charme qu'il procure permet de rajouter un second quai pour faire passer la gare de la catégorie 2 voies à quai, à celle de 3 voies à quai afin d'en augmenter son importance. Si la gare possède un côté voyageur, elle dispose également d'une face marchandise avec la cour à marchandise encadrée par la voie de la halle et la voie de débord. Ces 2 voies se rejoignent sur un tiroir qui peut être recouvert d'une remise si la gare est la tête d'une seconde voie unique ou si au contraire il s'agit d'une gare terminale. L'aiguillage a2 permet de garer un wagon, d'accéder à un embranchement particulier ou encore d'accéder à une seconde voie unique de faible importance (étant donnée qu'elle n'est raccordée qu'à une voie de la gare). Pour la réalisation j'ai opté pour une base en contreplaqué de 6mm reposant sur un cadre en tasseau de 27mm x 13mm afin de faire léger. Un contreplaqué de 5mm peut faire office de fond de décor. Coté voie, j'ai utilisé des voies diverses Jouef / Roco / Peco ... Par contre pour les appareils de voie, j'ai opté pour des Jouef car ils sont compactes, fiables, bon marché et le moteur visible est assez petit par rapport à ses concurrents. Mais bien entendu pour des réalisations plus réalistes des moteurs sous tables sont plus indiqués. Mais j'ai préféré la véracité de l'exploitation par rapport à une reproduction fidèle du décor. Coté exploitation 2 vénérables EAD Jouef (un double et un triple) assurent la desserte régulière de la ligne. Sporadiquement un express ou ce qu'il en reste dessert également la gare. Il est composé d'une 67000 ainsi que de 3 corails VTU de chez Piko. Bien entendu les mouvements suivent avec rigueur des horaires spécialement créés à cet effet. Coté marchandise une 66000 Piko avec 5 GAS ou une 63000 Roco avec 3 RILS assurent le spectacle.

## 7. Conclusion

J'espère que cette documentation aura été agréable et vous aura apporté des connaissances que vous réalisiez ou pas cette centrale.

Si vous voulez continuer la lecture, voici quelques liens intéressants:

- Norme DCC de la NMRA : [http://www.nmra.org/standards/DCC/standards\\_rps/DCCStds.html](http://www.nmra.org/standards/DCC/standards_rps/DCCStds.html)
- Centrale miniDDC : <http://www.minidcc.com/>
- Centrale + décodeurs + détecteurs de Paco : [http://www.tinet.org/~fmco/home\\_en.htm](http://www.tinet.org/~fmco/home_en.htm)
- Centrale + décodeur du professeur Mark : <http://www.technology.niagarac.on.ca/staff/mcsele/dcc.htm>
- Décodeurs à réaliser soit même : <http://www.merg.org.uk/resources/dcc.htm>
- Logiciel TCI pour dessiner des typons : <http://b.urband.free.fr/pagetci/tci.htm>
- Site pour réaliser des circuits imprimés : <http://etronics.free.fr/boutique/boutique.htm>
- Cours de BigOnOff (programmation des PIC en assembleur) <http://www.abcelectronique.com/bigonoff/>
- Site de Microchip (PIC) pour doc et échantillons : <http://www.microchip.com>
- Site de Maxim (MAX232/7219/7221) : <http://www.maxim-ic.com>
- Informations sur le bus I2C : [http://1100f.free.fr/tout\\_sur\\_le\\_bus\\_i2c.htm](http://1100f.free.fr/tout_sur_le_bus_i2c.htm)
- Signalisation SNCF : <http://transurb.net/bal/signaux1s.htm>
- Signalisation SNCF : <http://carreweb.fr/signalisation.html>
- Diagramme du Materiel SNCF : <http://www.train-rail.com> (malheureusement fermé temporairement)

L'archive free\_dcc.zip contient les fichiers suivants :

readme.txt : contenu de l'archive

free-dcc.doc : cette documentation

free\_dcc.tci : Les typons

enc\_it6.hex : programme pour le microcontrôleur de la centrale

enc\_it6.asm : sources de la centrale

s88\_16.hex : programme pour les microcontrôleurs des modules S88

s88\_16.asm : sources pour les microcontrôleurs des modules S88

fdcc\_conf.exe : Logiciel de configuration de la centrale par RS232

fdcc\_conf\_main.cpp : sources de ce logiciel

N'hésiter pas à me contacter, m'envoyer des photos, des remarques, des suggestions, des critiques constructives. Pour les questions, merci de lire la documentation en entier avant ;-) Pour lutter contre le Spam, merci de faire figurer dans le titre du mail le mot DCC afin que n'efface pas votre mail par inadvertance.

Mon adresse : <http://www.laposte.net/udelmas>

# Bonne réalisation

(si vous osez franchir le pas !)