

Multifunctional ARduino dcc DECoder



Le décodeur DCC multifonctionnel pour servomoteurs et accessoires pour tous avec DCCNext

(avec une centrale DCC)

Auteur : Nico Teering

Traduction : Jean-Michel Bapt

Juin 2019

Version de Mardec : 5.0a

Version du document : 5.0FR

Informations : info@arcomora.com

www.arcomora.com/mardec

Table des matières

Introduction	4
Le décodeur	4
ArCoMoRa	4
Toutes les possibilités de MARDEC en un seul coup d'œil	5
De quoi avez-vous besoin ?	6
Le matériel	6
Option 1 : Un Arduino UNO, ou un Arduino Mega2560 ou un Arduino Nano	6
Option 2 : Le décodeur DCCNext	7
Important en cas de mise à jour de la version 4 vers la version 5	8
Utilisation de relais	8
Amplificateur de puissance MOSFET	9
Logiciels	9
Exemples de connexions	10
Configuration de MARDEC	11
Les commandes	11
Commandes générales	12
Commande <I> (Initialisation)	12
Commande <P> (Port)	12
Commande <A> (Adresse)	12
Commande <T> (Test)	12
Commande <D> (Documentation)	12
Commande <R> (Reset / Remise à zéro)	13
Commande <E> (Exit / Exploitation)	13
Retour au mode configuration	13
IMPORTANT	13
Commandes destinées aux ports d'entrée	14
Commandes <2> et <3> (seconde et troisième adresse)	14
Commande <A> (Adresse - Existe également pour les accessoires et les servomoteurs)	16
Commande <N> (Numéro / Note - Existe également pour les accessoires et les servomoteurs)	16
Commande <T> (Type)	16
Commande <D> (Documentation)	16
Tests de fonctionnement	16
Commandes destinées aux servomoteurs	17
Commande <-> (signe moins).....	17
Commande <+> (signe plus)	17
Commande <9>	17
Commande <C> (Change)	17
Commande <I> (Inversion)	17
Commande <T> (Test)	17
Commande <S> (Speed / Vitesse)	17

Commande <F> (Frog point / Cœur d'aiguille)	18
Commande <A> (Adresse - Existe également pour les accessoires et les servomoteurs)	19
Commande <N> (Numéro / Note - Existe également pour les accessoires et les servomoteurs)	19
Commande <D> (Documentation)	19
Commande <Entrée>	19
Commandes destinées aux accessoires	20
Commande <M> (Mode)	22
Commande <H> (Hight / Haut)	22
Commande <L> (Low / Bas)	22
Commande <R> (Rise / Montée)	22
Commande <F> (Fall / Chute)	22
Commande <I> (Inversion)	22
Autres commandes	22
Divers	23
Démarrage	23
Installation et réglage des servomoteurs	23
Arrêt et sortie du programme Putty	23
Fichier de log	23
Documentation personnelle	23
Arduino Nano, Arduino Pro Mini et Arduino Mega2560	23
Configuration à partir du deuxième décodeur	24
Modifier le port COM	24
Configuration de Putty	24
Utilisation de modules relais	25
Exemples de configurations et de connexions	26
Connexion d'une aiguille à deux solénoïdes à l'aide de deux transistors MOSFET	26
Connexion d'une bande de LED RGB au moyen de transistors MOSFET	27
Connexion d'un servomoteur et d'un double relais à une aiguille	28

Introduction

Le programme MARDEC a été conçu pour automatiser votre réseau de façon simple et peu coûteuse. Le décodeur DCC est un décodeur DCC multifonctionnel basé sur un processeur Arduino.

Avec MARDEC, il est possible de gérer aussi bien des servomoteurs pour la motorisation des aiguilles que d'autres accessoires tels que des solénoïdes ou des LEDs.

Bien qu'utilisant un Arduino, MARDEC ne nécessite aucunement d'écrire une seule lettre de code. Mais, comme pour tout autre décodeur DCC, le MARDEC doit être configuré avant son utilisation. Cette configuration est souvent appelé de façon incorrecte « programmation ». Pour éviter cette confusion, il convient de réserver le vocable « programmation » au fait d'écrire le programme et d'utiliser le mot « configuration » lors de l'exécution du programme. Lors de cette configuration, les servomoteurs et autres accessoires seront configurés.

Le MARDEC fonctionne dans deux "états" différents.

En mode configuration, les servomoteurs et autres accessoires vont être configurés par l'intermédiaire de commandes à une seule lettre frappée au clavier. Dans ce mode, le programme MARDEC communique avec l'ordinateur via un câble USB.

Aucun signal DCC n'est nécessaire pour cette configuration.

Au moyen d'une commande spécifique (<E>), le MARDEC passe en mode exploitation.

Dans ce mode exploitation, les servomoteurs et autres accessoires sont obéissent au signal DCC. Le câble USB n'est plus nécessaire.

L'installation de la suite logicielle est très simple. Tous les logiciels nécessaires sont installés en une seule fois. Le téléchargement du programme vers le décodeur est également pratiquement entièrement automatisé.

Vous trouverez plus d'information à ce sujet dans le manuel d'installation.

Le décodeur

Pour réaliser le décodeur lui-même, vous avez trois options :

- Soit utiliser un Arduino UNO ou MEGA avec la carte fille (shield) DCC/Power comme décrit sur www.arcomora.com/mardec
- Soit utiliser un Arduino NANO avec un circuit DCC fait maison.
- Soit utiliser un le décodeur **DCCNext** tel que décrit sur www.arcomora.com/dccnext.

La version 5 du programme MARDEC est conçue pour le **DCCNext** mais peut également être utilisée sur un Arduino avec la carte fille (shield) DCC/Power.

ArCoMoRa

MARDEC fait partie du concept ArCoMoRa dont la signification est : **A**rduino **C**ontrolled **M**odel **R**ailway. ARSIGDEC, un décodeur DCC de signaux, en fait également partie, tout comme ARLOCO, une interface de rétrosignalisation Loconet.

Plus d'information sur www.arcomora.com

Aperçu rapide des possibilités de MARDEC

MARDEC dispose des possibilités et options suivantes :

- Pilote jusqu'à 12 servomoteurs sans polarisation du cœur d'aiguille ou jusqu'à 8 servomoteurs avec polarisation du cœur d'aiguille. La polarisation s'effectue par un relais externe.
- Pilote les accessoires de 10 manières différentes, y compris pour le contrôle des solénoïdes doubles et les accessoires à commande analogique PWM (Pulse Width Modulation).
- Chaque port peut être configuré comme une entrée. Ceci permet de contrôler automatiquement un servomoteur ou un accessoire connecté sur un autre port.
- Configuration de manière interactive des servomoteurs et des accessoires via l'écran et le clavier. Ceci de manière totalement indépendante de la centrale DCC utilisée. Le logiciel Arduino n'est PAS nécessaire.
- Le début et la fin de course de chaque servomoteur peuvent être réglés avec précision au degré près.
- Attribuez n'importe quelle adresse DCC (de 1 à 2000) à chaque servomoteur ou accessoire. Ces adresses ne sont pas nécessairement consécutives.
- Chaque servomoteur (au maximum 8) peut être couplé à un relais pour la polarisation des cœurs d'aiguille. Lorsque le servomoteur est en mouvement, le relais sera actionné à mi-course de la rotation du servomoteur.
- Chaque servomoteur (au maximum 5) peut être couplé à deux relais pour la polarisation des cœurs d'aiguille. Il en résulte une précision encore plus fiable du point de basculement.
- La vitesse de rotation de chaque servomoteur peut être personnalisée. Il est ainsi très facile d'utiliser les servomoteurs à d'autres fins que pour les moteurs d'aiguilles.
- Une fonction de test est disponible avec laquelle tous les servomoteurs feront deux aller-retour et chaque accessoire sera activé.
- Chaque servomoteur peut avoir son fonctionnement inversé par rapport à son paramétrage initial afin de permettre d'adapter son fonctionnement à son installation physique sur le réseau.
- Une documentation intégrée permet d'afficher tous les paramètres mémorisés pour chaque port.
- Possibilité d'attribuer un repère administratif à chaque servomoteur / accessoire / entrée.
- Possibilité de réinitialisation complète du module où tous les réglages sont effacés de la mémoire.
- La vitesse par défaut des servomoteurs est réglable (de 5 à 50 ms par degré de déplacement).
- Une fois en mode exploitation, retour en mode configuration en connectant le câble USB et en entrant simplement la commande "C".
- Possibilité de corriger le décalage d'adresse existant avec les centrales Roco (MM, z /Z21)
- Les réglages possibles pour les accessoires sont les suivants :
 1. **Etat stable** : dans ce mode, un port est en permanence haut ou bas en fonction de la commande.
 2. **Etat stable double** : mode identique à l'état stable simple, mais un deuxième port est mis à l'état inverse.
 3. **Clignotant simple** : Dans ce mode, un port va alternativement être à l'état haut (ON) et bas (OFF). La durée "ON" et la durée "OFF" peuvent être réglées séparément.
 4. **Double clignotement** : mode identique au clignotement simple, mais un deuxième port fonctionnera en mode inversé.
 5. **Etat momentané** : dans ce mode, un port sera à l'état haut pendant une courte période dont la durée est réglable lorsque la commande va de bas à haut. Ceci peut être utilisé pour commander un servomoteur ou un accessoire sur un autre port à la fin de l'impulsion.
 6. **Etat momentané double** : dans ce mode, un port changera momentanément d'état pour une durée réglable ("haut" si la commande passe de « bas » à « haut ») et un autre port changera momentanément d'état pour une durée également réglable si la commande passe de « haut » à « bas ». Ceci permet de commander des aiguilles à doubles solénoïdes.
 7. **Commande analogique PWM (Pulse Width Modulation)** : dans ce mode, et pour une durée réglable individuellement, un port passe de 0 à un maximum réglable (255 au maximum) si la commande passe de bas à haut, puis revient à 0 dans un temps réglable individuellement si la commande passe de haut à bas.
 8. **Mode scintillement** : dans ce mode, vous pouvez créer un scintillement de la LED connectée. Avec des LEDs adaptées, vous pouvez, par exemple, simuler un feu ou une lampe de soudage.
 9. **Mode aléatoire long** : ce mode vous permet d'allumer et d'éteindre en permanence une LED. Les temps de marche et d'arrêt sont déterminés de manière aléatoire entre deux limites réglables de 1 à 1000 secondes.
 10. **Mode aléatoire court** : ce mode vous permet d'allumer et d'éteindre rapidement une LED entre deux limites réglables de 5 à 2000 ms.
- Une commande d'aide affiche toutes les commandes de configuration possibles.
- Une option de journal qui enregistrera toutes les actions de configuration.

Indépendant du BUS utilisée (LocoNet, S88 ou autres). Les commandes passent uniquement par le DCC.

De quoi avez-vous besoin ?

Le matériel

Vous avez plusieurs options.

Option 1 : un Arduino UNO, ou un Arduino Mega2560 ou un Arduino Nano

Un Arduino est un micro-ordinateur avec beaucoup de connexions. C'est ce qu'on appelle les ports. Celles-ci sont numérotées sur le circuit imprimé. Avec le programme MARDEC nous utilisons jusqu'à 16 de ces ports. Ces ports peuvent être utilisés directement pour commander des servomoteurs, des modules relais (voir ci-dessous) et des LEDs (avec résistances).

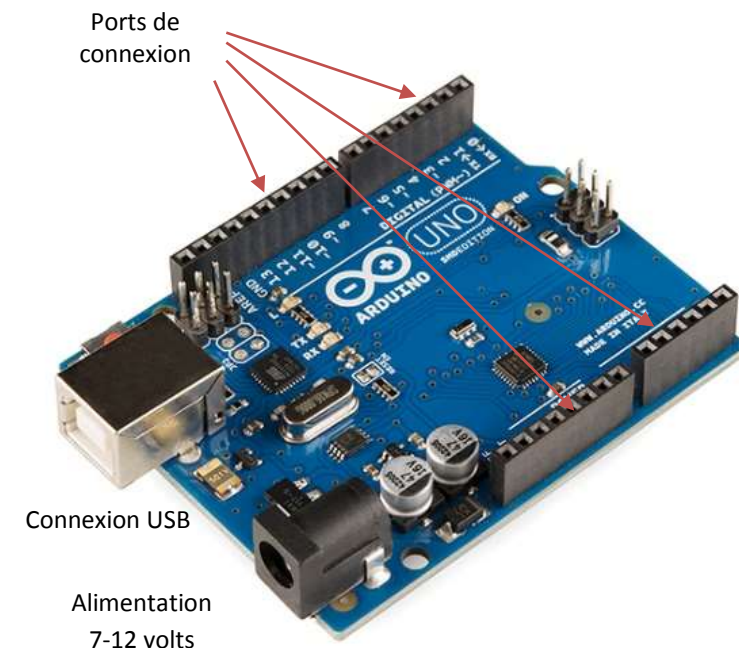
LEDs (avec résistances).

Un port d'Arduino a une tension de sortie de 0 ou de 5 volts et peut supporter jusqu'à 40 mA. Pour les accessoires qui nécessitent plus de puissance (par exemple moteurs, solénoïdes d'électro-aimant et bandes de LED), un étage d'amplification est nécessaire.

Pour les applications analogiques, il y a 4 ports à laquelle la tension peut être appliquée avec une largeur d'impulsion variable (PWM).

Sur la gauche, vous voyez la connexion USB. L'Arduino est Alimenté par ce câble USB.

La connexion USB est nécessaire pour télécharger le programme dans l'Arduino. Il est également nécessaire de pour communiquer avec l'ordinateur lors de la configuration. Si aucun câble USB n'est connecté, une alimentation extérieure doit être connectée



sur la prise Jack sur la droite. L'alimentation recommandée doit être de 7 à 12 Volt DC. L'Arduino transformera lui-même la tension d'entrée en une tension stabilisée de 5 volts. Les deux sources d'alimentation peuvent être utilisées simultanément.

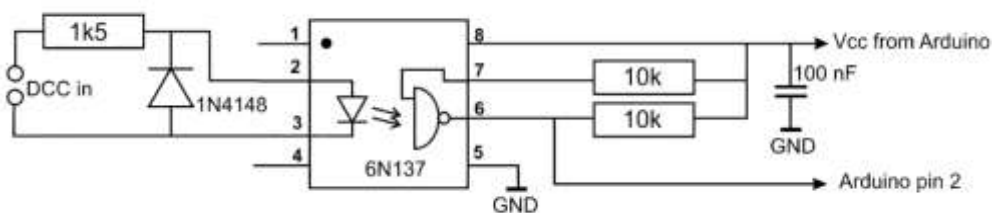
Sur l'Arduino nous voyons aussi quelques ports avec le texte « Power ». Ici nous trouvons deux ports 5 volts (sortie), deux connecteurs GND et un port Vin. A la place de la prise Jack, vous pouvez également utiliser le port « Vin ». Ne connectez pas une alimentation de 5V aux ports de sortie 5V.

DCC circuit

Deuxièmement, une petite quantité d'électronique est nécessaire pour que le signal DCC soit adapté à l'Arduino. Le composant 6N137 est un optocoupleur rapide qui transmet le signal DCC à l'Arduino. Les 5 volts nécessaires peuvent être récupérés directement sur un port 5V de l'Arduino.

Le port 6 de l'optocoupleur doit être connecté au port 2 de l'Arduino.

Ces composants supplémentaires peuvent être soudés sur une plaque de veroboard.



Une carte fille (shield) a été développée pour ce circuit, la carte DCC/Power. L'Arduino peut également être alimenté par cette carte.

Cette carte contient également une alimentation 5V (max. 1 AMP) qui peut être utilisée pour des relais, des LEDs et des servomoteurs entre autres choses.

Cependant, une alimentation externe est recommandée pour les servomoteurs et les relais.

Cette carte peut être commandée sur la page :

www.arcomora.com/order-2

Si elle n'est pas en stock, le délai de livraison peut être de quelques semaines. Vous devez par ailleurs acheter un Arduino.

Option 2: le décodeur DCCNext

Le décodeur DCCNext est complètement nouveau.

Ce décodeur intègre un processeur Arduino (ATMEGA328P) avec une alimentation et un circuit DCC et est donc une combinaison d'un Arduino UNO et de la carte fille DCC.

Une interface USB séparée (CH340) assure la connexion avec le PC.



Ce décodeur peut être équipé de bornes à vis et de broches Dupont. Un servomoteur peut être branché directement sur ces broches.

A cet effet, Une connexion est prévue pour une alimentation externe de 5V.

La présence du signal DCC est signalée par une LED.

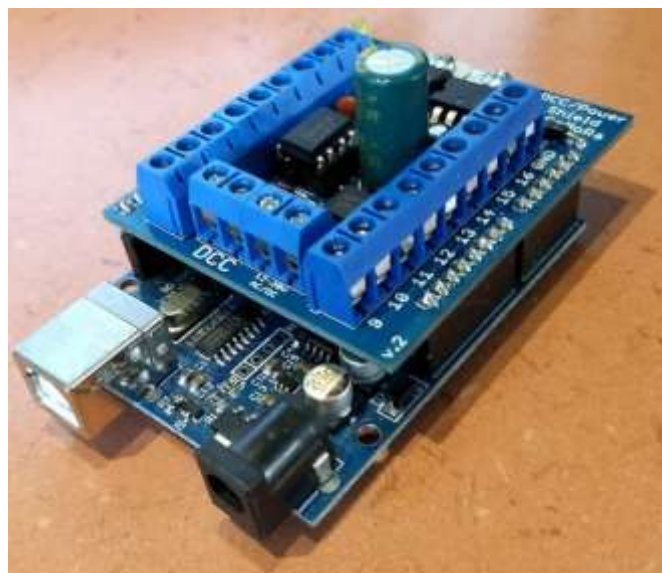
Une différence importante avec la carte fille DCC est l'utilisation des numéros de ports 1 à 16 au lieu des numéros de pins de l'Arduino (3 à 19).

Ce qui suit dans ce manuel est basé sur les numéros de port du décodeur DCCNext.

Pour plus d'informations : www.arcomora.com/dccnext

Ce décodeur DCCNext peut être commandé directement sur cette page :

www.arcomora.com/reservation



Important:

Le programme MARDEC version4 peut également être utilisé avec le **DCCNext**.

Le programme MARDEC version 5 peut également être utilisé avec la carte fille DCC/Power.

Mais, dans les deux cas, le tableau de conversion ci-dessous doit être utilisé :

Mardec 4 pins	Mardec 5 ports	Mardec 4 pins(Mega)	Mardec 5 ports
3	1	11	9
4	2	12	10
5	3	14(54)	11
6	4	15(55)	12
7	5	16(56)	13
8	6	17(57)	14
9	7	18(58)	15
10	8	19(59)	16

Le programme MARDEC version 5 utilise une nouvelle structure de données pour mémoriser la configuration du décodeur.

Le programme MARDEC version 4 ne peut donc PAS être mis à jour vers la version 5.

Si vous remplacez Le programme MARDEC version 4 par la version 5 sur un Arduino, vous devrez reconfigurer complètement le décodeur.

Il est recommandé de faire une capture d'écran de la configuration du décodeur avant d'installer la version 5 sur la version 4.

Utilisation de relais

Pour la polarisation des cœurs d'aiguilles ou pour tout autre application, n'importe quel relais peut être utilisé.

A cette fin, une broche Arduino est capable de commuter un relais au moyen d'un transistor de manière classique.

Les [modules relais](#) prêts à l'emploi sont peu chers et faciles à utiliser. Ils sont pilotables très simplement avec un Arduino.

Ils comprennent généralement deux circuits de relais indépendants dont l'activité est signalée par LED et un optocoupleur. Ils sont également disponibles en version 1, 4 ou 8 relais.

Ils peuvent être alimentés directement à partir de

l'Arduino mais il est fortement recommandé d'utiliser une alimentation externe de 5V à cet usage.

Ils sont également disponibles en version 12 volts. L'un des avantages est un courant de bobine un peu plus faible.

Voir aussi page 25 pour en savoir plus.

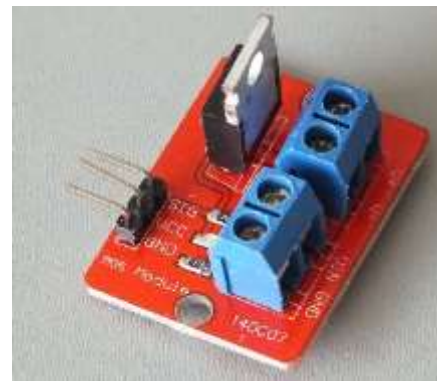


Utilisation d'amplificateur de puissance MOSFET

Les moteurs à solénoïde ainsi que les bandes de LEDs nécessitent l'emploi d'amplificateurs [MOSFET](#).

Ces amplificateurs peuvent également être utilisés en PWM (variation de largeur d'impulsion). Vous voyez ici un module prêt à l'emploi mais vous pouvez le fabriquer vous-même en utilisant un morceau de veroboard.

Les types de MOSFET appropriés sont le FQP30N06L et le RFP30N06LE. Le 'L' signifie 'niveau logique'. Cela veut dire que le MOSFET est entièrement passant avec une tension de commande de 5V et n'a pratiquement plus de résistance.



Logiciel

Evidemment, un programme est également requis :

1. Pendant l'installation de l'application, le programme MARDEC est installé sur votre PC sous forme d'un fichier binaire précompilé : MARDEC.hex
2. Ensuite, il faudra télécharger le programme MARDEC dans l'Arduino. Ceci sera fait en transférant le programme de votre PC vers l'Arduino par l'intermédiaire du câble USB. Ce téléchargement est réalisé automatiquement en utilisant le raccourci créé à cet effet sur le bureau.
3. Un émulateur de terminal indépendant (Putty.exe) est utilisé pour les communications avec MARDEC. Ce programme sera également lancé à l'aide d'un raccourci créé sur votre bureau. Cet émulateur permet d'avoir une sortie écran pour le programme en cours d'exécution dans l'Arduino et également d'envoyer des données de votre clavier vers le programme MARDEC.
4. Enfin, des drivers sont requis. Si vous avez déjà installé sur votre PC l'IDE Arduino (Environnement intégré de développement), ces drivers sont probablement déjà installés. Rappel : pour le clone chinois de l'Arduino, il est en général nécessaire d'installer le driver spécial CH340.

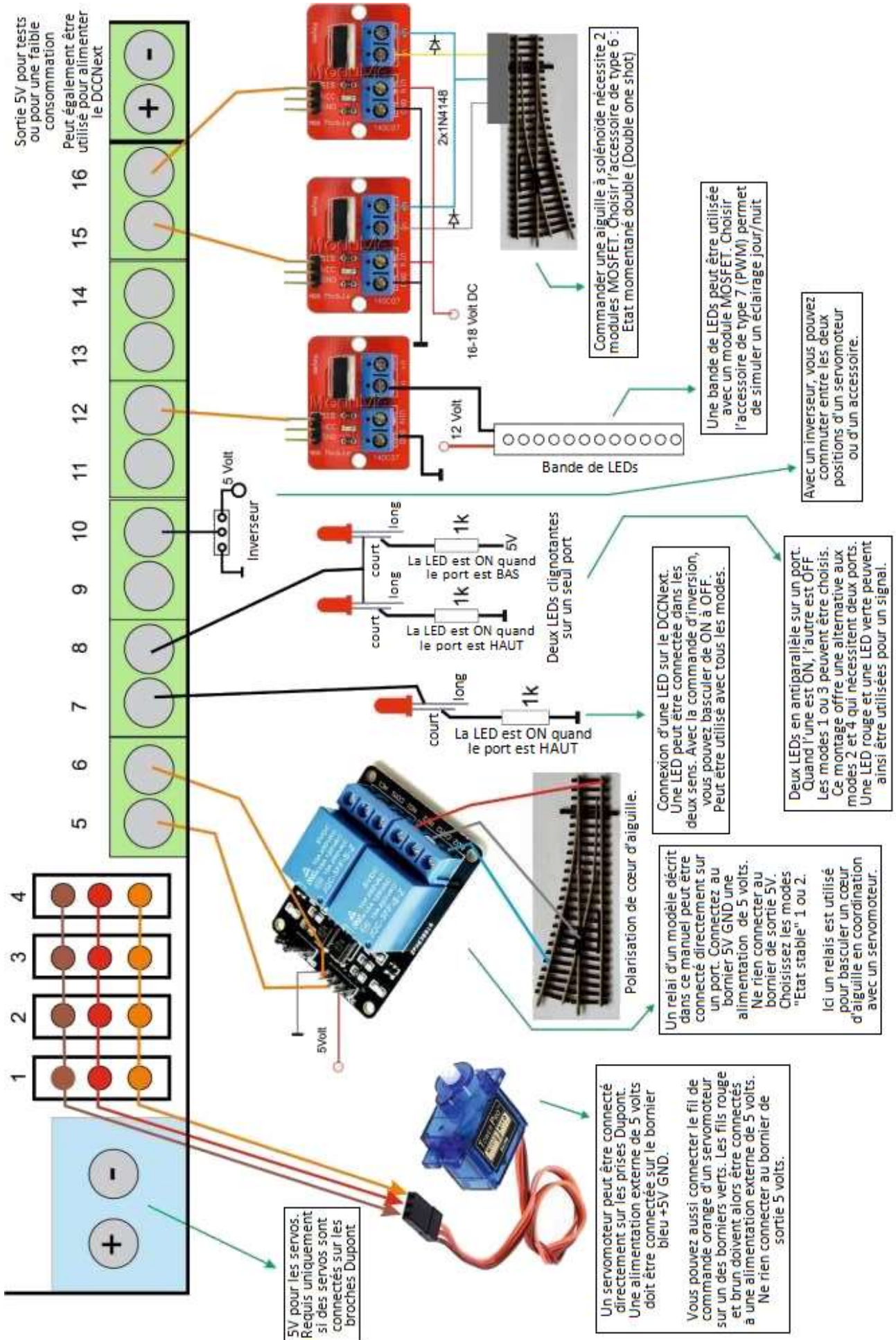
Ces quatre composants sont automatiquement installés sur votre PC pendant le processus d'installation.

Cliquez et c'est parti !

Consultez le manuel d'installation ci-dessous pour plus de détails.

[TÉLÉCHARGEZ ICI LES LOGICIELS ARCOMORA](#)

Exemples de connexions au décodeur DCCNext



Configuration de MARDEC

Si l'installation du logiciel s'est déroulée correctement et que le décodeur **DCCNext** est connecté, le processus de configuration peut commencer.

Pour ce faire, cliquez sur le raccourci "*Configure Mardec*".

Si c'est la toute première installation d'un module Arduino, les pilotes nécessaires à la communication USB seront installés, suivis par le téléchargement dans le module du programme Mardec et le lancement du programme Putty. Putty est un émulateur de terminal qui est utilisé pour la communication entre le module Arduino et votre PC. (Voir aussi le manuel d'installation).

Mardec utilise deux modes différents : le mode configuration et le mode fonctionnement. Par défaut, au premier démarrage, il sera en mode configuration. Ensuite, la communication avec le module démarre immédiatement. (Voir aussi les instructions d'installation)

Veillez noter :

- Chaque entrée numérique doit être suivie par un appui sur la touche <Entrée>.
- Seules les commandes à un seul caractère ne nécessitent pas d'être suivies par la touche <Entrée>.
- Pour la plupart des commandes, la valeur mémorisée (indiquée entre parenthèses) n'est pas modifiée par le simple appui de la touche <Entrée>.
- En mode configuration, la LED jaune sur la carte fille (shield) DCC/Power (ou le voyant intégré sur le port 13) est toujours allumé.
- En mode de fonctionnement, ce voyant est éteint.
- Appuyez sur la touche <Numlock> du pavé numérique.
- Vous pouvez utiliser indifféremment les minuscules ou les majuscules pour les commandes.
- Utilisez **UNIQUEMENT** la touche <Retour arrière> pour corriger une entrée.

La première fois que vous démarrez le programme MARDEC, ou après une réinitialisation complète (avec la commande <R>), certains paramètres généraux de configuration vous seront d'abord demandés. Ces paramètres généraux pourront ultérieurement être modifiés avec la commande <I> (voir plus loin dans ce manuel).

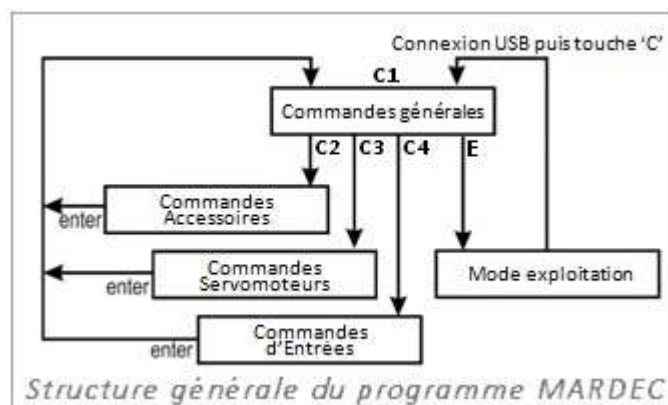
Une vue d'ensemble de la configuration mémorisée s'affiche automatiquement à chaque démarrage.

A partir de cet instant, une commande peut être saisie.

Les commandes

Il y a quatre types de commandes :

- Les commandes générales ;
- Les commandes relatives aux servomoteurs permettant de configurer ceux-ci ;
- Les commandes relatives aux autres accessoires permettant de configurer ceux-ci ;
- Les commandes relatives aux entrées permettant de configurer celles-ci.



Le processus de configuration utilise donc quatre types de commandes (C1, C2, C3 et C4).

En saisissant un ? (point d'interrogation), une vue d'ensemble des commandes disponibles en fonction du type de commandes actuellement en cours s'affichera.

Chaque commande se compose d'une seule lettre et n'a pas besoin d'être suivie de l'appui sur <Entrée>.

Les lettres utilisées pour les commandes sont basées sur la description anglaise de la fonction de chacune de celles-ci.

Un servomoteur, un accessoire ou une commande d'entrée ne pourront être configurés que si un port est préalablement sélectionné.

Certaines commandes générales peuvent également être utilisées dans les commandes de servomoteur, d'accessoire ou d'entrée.

Au démarrage du mode de configuration, vous ne pouvez saisir que des commandes générales.

Une commande importante est la commande <P> (pour Port). Elle vous permet de sélectionner un port.

Les commandes générales

Commande <I> (Initialisation)

Avec la commande <I> les paramètres initiaux sont réinitialisés.

- **Numéro (number)** de MARDEC.

Ce numéro identifiera ce décodeur.

- Utilisez-vous une centrale **Roco Multimaus**, ou **z21**, ou **Z21** ?

Si c'est le cas, MARDEC corrigera automatiquement les adresses (par défaut : Non).

REMARQUE : concernant la z21, veuillez désactiver la fonction de communication Railcom/Rail !

- Réglage par défaut de la **vitesse de rotation** des servomoteurs. Cette vitesse sera utilisée pour chaque nouveau servomoteur.

- Normalement, un servomoteur est "déconnecté"

(= **detached**) de l'Arduino après avoir terminé son déplacement. Ceci évite tout phénomène de "tremblement" et permet au servomoteur de tourner si une force externe lui est appliquée. Si cet éventuel déplacement pose problème, vous pouvez définir le servomoteur comme restant "connecté" (= **attached**). Mardec corrigera alors immédiatement tout écart de position.

- Sélection du mode à la mise sous tension (**startup mode**) : 1 (Last) = mode en fonction au moment de la dernière mise hors tension, 2 = démarrage en mode configuration, 3 (Normal) = démarrage en mode exploitation.

```
mardec on port COM28
MARDEC, the Multifunctional ARduino dcc DECoder
UNO version 5.0

Mardec starting, please wait

Specify number for this MARDEC
Enter value from 1 to 100: 1
This is MARDEC #1

Do you use a ROCO MultiMaus or z/Z21 Y/N ? (def.: N)
Address offset: No

Default rotation speed for all servo's
Enter value from 5 to 100 (25):
Default speed set to 25 ms/degree

Detach servo at end of rotation Y/N ? (def.: Y)
Servo's are detached at end of rotation

Startup mode: 1=Last (def.),2=Configuration,3=Normal
Enter value from 1 to 3 (1):
Startup mode: Last
```

Commande <P> (Port)

Avec la commande <P>, un numéro de port vous sera demandé.

Ci ce port n'est pas déjà défini, vous devrez d'abord renseigner son adresse DCC ainsi que sa fonction principale (servomoteur, accessoire ou entrée).

Si ce port est déjà défini, sa configuration actuelle sera affichée et le mode de configuration basculera dans le mode correspondant.

Vous pouvez utiliser les port numérotés de 1 à 16.

Un accessoire nécessitant l'utilisation du mode

analogique PWM (modes 7 et 8) ne peuvent être connectés qu'aux ports 1, 3, 4 et 9 (limitation liée à l'Arduino).

En fonction du choix du type d'accessoire connecté, Mardec proposera alors les commandes correspondantes.

```
Specify action (P/A/R/T/D/E/I/?): P
Specify port number
Enter value from 1 to 16: 1
Port is undefined. Specify DCC address

Set DCC address for port 1
Enter value from 1 to 2000: 236
DCC Address set to 236

Specify Accessory(1), Servo(2) or Input(3)
Enter value from 1 to 3: 1
```

Commande <A> (Adresse)

Si un port est déjà configuré, vous pouvez directement saisir son adresse DCC. Mardec trouvera alors le port correspondant.

```
Specify action (P/A/R/T/D/E/I/?): A
Specify DCC address
Enter value from 1 to 2000: 236
DCC 236, Acc.type 1 (S. Steady), , Not Inv.

Specify action for Accessory on A236/P1
enter/A/N/I/D/T/M/H/L/?/F/R: █
```

Commande <T> (Test)

Après avoir configuré tous les ports, vous pouvez tester chaque dispositif qui y est connectés. Chaque dispositif sera actionné quelques instants à tour de rôle (les servomoteurs feront quelques allers-retours). Le test des entrées se fait en connectant l'entrée correspondante à la masse (GND) avec un câble Dupont.

La commande <T> est également disponible lors du paramétrage d'un servomoteur ou d'un accessoire. Seul le dispositif en question sera alors testé.

Commande <D> (Documentation)

La commande <D> affiche l'aperçu de la configuration de chaque port.

Faites-en une capture d'écran et enregistrez-la sur votre ordinateur ou imprimez-la directement.

Ceci vous permettra d'avoir en permanence l'aperçu de l'ensemble de la configuration du décodeur.

Commande <R> (Reset / Remise à zéro)

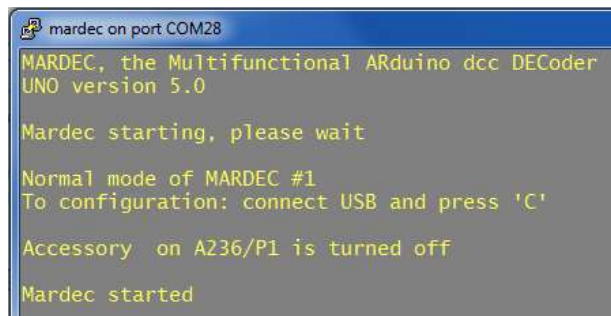
La commande R efface tous les réglages de la mémoire de l'Arduino. Une confirmation vous sera demandée deux fois. Après une réinitialisation, l'écran est effacé et la séquence de paramétrage initial du décodeur démarre immédiatement

```
Specify action (P/A/R/T/D/E/I/?): R
Clear all settings? Y/N: Y
ABSOLUTELY sure? Y/N: Y
```

Commande <E> (Exit / Exploitation)

Avec la commande <E>, le mode exploitation démarre. Les accessoires connectés au décodeur répondent maintenant au signal DCC venant de votre centrale, le câble USB n'est plus nécessaire et vous pouvez fermer le programme Putty.

La LED jaune de la carte fille DCC/Power est maintenant éteinte. Elle réagira lorsqu'un ordre DCC concernera le décodeur (rôle dévolu à la LED rouge pour le décodeur **DCCNext**).



```
mardec on port COM28
MARDEC, the Multifunctional ARduino dcc DECodeR
UNO version 5.0
Mardec starting, please wait
Normal mode of MARDEC #1
To configuration: connect USB and press 'C'
Accessory on A236/P1 is turned off
Mardec started
```

Si vous laissez le programme Putty ouvert avec le câble USB connecté, vous pourrez voir à l'écran les actions effectuées quand une instruction DCC arrive au décodeur, e qui vous permet de contrôler le bon fonctionnement.

Retour au mode configuration

Pour revenir au mode configuration, rebranchez le câble USB et lancez ensuite le programme de configuration à l'aide du raccourci "*Configure Mardec*".

Le décodeur redémarre comme le montre les 3 clignotements courts de la LED d'état, puis reste éteint. Appuyez ensuite sur la touche <C> de votre clavier. Le programme MARDEC démarre alors en mode configuration et la LED d'état est allumée de nouveau en permanence.

Important

Après chaque commande DCC reçue, le nouvel état de chaque servomoteur ou accessoire est sauvegardé immédiatement dans la mémoire du décodeur. Lorsque le décodeur est mis sous tension, les états mémorisés sont lus et il actionne éventuellement les servomoteurs et accessoires pour les repositionner tels qu'ils étaient au moment de la mise hors tension.

Il peut arriver qu'un servomoteur ait un mouvement parasite lors de la mise sous tension.

Comme le décodeur, avec un léger décalage, réinitialise les servomoteurs à leur dernière position mémorisée, une éventuelle 'discordance de positionnement' au démarrage sera rectifiée par le décodeur.

Pour palier à ces mouvements parasites, l'emploi d'une bonne alimentation électrique est conseillé ainsi que des servomoteurs numériques de haute qualité (TG9d, HXT900, ES9051).

Si nécessaire, un noyau de ferrite à proximité du servomoteur peut également être utile.

Enfin, une résistance de 4k7 entre le signal et le 5V peut également aider.

Les commandes destinées aux ports définis comme entrée (input)

Avec un port configuré comme entrée, vous pouvez simuler une commande DCC destinée à un servomoteur ou à un accessoire qui a la **même adresse** que le port d'entrée.

Exemple :

Sur le port 12, une LED clignotante est configurée avec l'adresse DCC 34.

Sur le port 5, une entrée est configurée avec l'adresse DCC 34.

Par défaut, MARDEC maintient à l'état HAUT (5V) un port d'entrée. Si cet état passe à BAS par l'action d'un dispositif quelconque (interrupteur ou détecteur d'occupation par exemple), MARDEC interprète alors ceci comme un signal DCC destiné à l'adresse 34 et la LED clignotante sur le port 12 vas s'éteindre (ou commencer à clignoter).

De la même façon, si c'est un servomoteur qui a l'adresse 34, il se mettra en mouvement.

Une entrée peut être activée de trois façons :

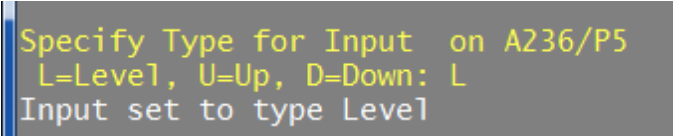
1. Quand elle passe à l'état BAS. Rien ne se passera ensuite quand l'entrée repassera à l'état HAUT. (type D) ;
2. Quand elle passe à l'état HAUT. Rien ne se passera ensuite quand l'entrée repassera à l'état BAS. (type U) ;
3. Quand elle passe à l'état BAS ou à l'état HAUT. L'accessoire ou le servomoteur "suivra " le signal d'entrée comme s'il était contrôlé par le DCC. (type L).

Avec cette option, un aiguillage peut être commandé par l'intermédiaire d'un interrupteur par exemple.

Avec les options 1 et 2, une entrée agit comme un interrupteur momentané.

Avec l'option 3, une entrée agit comme un inverseur, c'est-à-dire comme une commande DCC.

La méthode d'activation doit être précisée lors de la configuration du port.



```
Specify Type for Input on A236/P5
L=Level, U=Up, D=Down: L
Input set to type Level
```

Un port configuré comme entrée peut être piloté de différentes manières. Par exemple, par un ILS qui commutera à la masse ou par un détecteur d'occupation qui passera au niveau BAS lorsqu'un train est détecté ([Okkie](#)). Une barrière infrarouge ou un simple interrupteur peuvent également être utilisés.

Bien entendu, ce qui est affecté à l'adresse 34 peut également être commandé à partir d'un vrai signal DCC. Une différence importante cependant, est que, s'il ne se passe rien pour les options 1 ou 2 sur un port configuré comme entrée s'il revient au niveau BAS ou HAUT, un changement d'état DCC inversera lui l'état du dispositif. Avec l'option 3 (changement d'état), une entrée fonctionne de la même manière que le signal DCC.

ATTENTION :

Gardez à l'esprit que si vous mettez en route un accessoire à partir de la centrale numérique ou de l'ordinateur et que vous l'éteignez à l'aide d'un dispositif autre, la centrale numérique ou l'ordinateur "penseront " que l'accessoire est toujours en fonction.

Tenter de l'éteindre au moyen d'un ordre DCC n'aura aucun effet.

Commandes <2> et <3> (seconde et troisième adresse)

Une seconde et une troisième adresse peuvent être attribuées à un port configuré comme entrée à l'aide des commandes <2> et <3>. Ensuite, lors de l'activation de ce port, les accessoires ou servomoteurs configurés avec la seconde et la troisième adresse seront aussi activés.

Pour les secondes et troisièmes adresses vous pouvez spécifier un retard jusqu'à 25 secondes par pas de 0.1 seconde. Les accessoires ou servomoteurs configurés avec la seconde et la troisième adresse ne seront alors activés qu'une fois écoulé leur temps de retard respectif. Le décompte des deux retards commence au moment de l'activation du port concerné.

Toutes les adresses doivent bien sûr être configurées sur le même décodeur.

Si vous ne voulez que des actions 'retardées', vous pouvez attribuer une adresse fictive au port d'entrée. Il y aura donc un port inutilisé qui sera configuré avec cette adresse. Vous pouvez aussi donner l'adresse d'un autre port comme seconde et troisième adresse. Ceci vous permet de lancer toute une série d'actions.

```

Specify port number
Enter value from 1 to 16: 5
Port is undefined. Specify DCC address

Set DCC address for port 5
Enter value from 1 to 2000: 236
Address in use for servo/accessory on A236/P1
Is that OK ? (Y/N, N=default): Y

DCC Address set to 236

Specify Accessory(1), Servo(2) or Input(3)
Enter value from 1 to 3: 3
Port 5 set as Input port.

Specify Type for Input on A236/P5
L=Level, U=Up, D=Down: L
Input set to type Level

Specify action for Input on A236/P5
A/N/?/D/T/2/3/enter: 2
Specify Second control address.
Enter value from 1 to 2000: 120
Specify delay time in 0.1 seconds.
Enter value from 0 to 250: 20
Address 120 set as Second control address with a delay of 2.0 sec.

Specify action for Input on A236/P5
A/N/?/D/T/2/3/enter: 3
Specify Third control address.
Enter value from 1 to 2000: 45
Specify delay time in 0.1 seconds.
Enter value from 0 to 250: 50
Address 45 set as Third control address with a delay of 5.0 sec.

Specify action for Input on A236/P5
A/N/?/D/T/2/3/enter:

Port settings are saved!

Specify action (P/A/R/T/D/E/I/?): D

Settings of MARDEC #1

Default servo rotation speed: 25 ms/degree
Address offset: No
Servo's are detached at end of rotation
Startup mode: Last

Port 1: DCC 236, Acc.type 1 (S. Steady), , Not Inv.
Port 2: DCC 236, Servo , Angles 75/105, Not Inv., Speed 25, Frog port no
Port 3: DCC 120, Acc.type 3 (S. Flashing), , Not Inv., Time(ms) 300/400
Port 4: DCC 45, Servo , Angles 75/105, Not Inv., Speed 25, Frog port 16
Port 5: DCC 236, Input , Type: L, Second addr/delay: 120/2.0 sec, Third addr/delay: 45/5.0 sec

```

Dans cet exemple, voici ce qui se produit si le port 5 (configuré comme entrée) change de niveau :

- L'accessoire sur le port 1 change d'état car le port 1 a la même adresse que le port 5.
- Le servomoteur sur le port 2 démarre car le port 2 a la même adresse que le port 5.
- Après 2 secondes, l'accessoire sur le port 3 change d'état car le port 3 a la même adresse que la deuxième adresse du port 5.
- Après 5 secondes, le servomoteur démarre sur le port 4 car le port 4 a la même adresse que la troisième adresse du port 5.

Si l'adresse DCC 236 est commandée :

- L'accessoire sur le port 1 change d'état.
- Le servomoteur du port 2 démarre puis s'arrête.

Si l'adresse DCC 120 est commandée : l'accessoire sur le port 3 change d'état.

Si l'adresse DCC 45 est commandée : le servomoteur sur le port 4 démarre.

Commande <A> (Adresse - *existe aussi pour les accessoires et les servomoteurs*)

La commande <A> permet de définir l'adresse du port. Si cette adresse est déjà utilisée, un message vous en avertira. En confirmant votre choix, vous pourrez contrôler plusieurs accessoires ou servomoteurs avec une seule adresse.

En spécifiant l'adresse 0 (zéro) pour un port précédemment paramétré, celui-ci sera alors reseté et considéré comme libre.

```
Set DCC address for port 4
Enter value from 0 to 2000 (32): 0
Do you want to reset port 4? Y/N: Y

DCC Address set to 0

Port settings are saved!
```

Commande <N> (Numéro / Note - *existe aussi pour les accessoires et les servomoteurs*)

La commande <N> vous permet d'assigner (sur 4 caractères) une codification administrative au port.

Cette assignation n'est nullement obligatoire et n'aura aucune influence sur le paramétrage.

```
Specify action for Input on A2/P3
A/N/?/D/T/2/3/enter: N

Specify administrative note: Inp2
Administrative note set to Inp2
```

Commande <T> (Type)

Avec la commande <T>, et pour un port défini comme entrée, vous pouvez changer son type d'entrée.

```
Specify Type for Input on A2/P3
L=Level, U=Up, D=Down: d
Input set to type Down-Pulse
```

Commande <D> (Documentation)

La commande <D> affiche une vue d'ensemble de tous les paramètres.

A propos des tests

Il n'y a pas de test spécifique pour le port définis comme entrée. Néanmoins, vous pouvez les tester en sortant du mode configuration à l'aide de la commande <E>. Ensuite, en connectant un port défini comme entrée à la masse (GND) à l'aide d'un câble, vous pourrez vérifier que les servomoteurs et accessoires liés à cette entrée répondent correctement.

Les commandes destinées aux servomoteurs

Voici à quoi ressemble la boîte de dialogue concernant la configuration d'un servomoteur :

Un nouveau port configuré pour un servomoteur est réglé par défaut avec des angles de 75° pour le niveau bas et de 105° pour le niveau haut. Un port de servomoteur précédemment paramétré n'est pas affecté par ces valeurs par défaut et conserve les angles haut et bas antérieurement définis.

Dans les deux cas, le servomoteur est initialisé en position basse.

Commande <-> (signe moins)

La commande <-> diminue l'angle du niveau bas de 1°. L'angle du niveau haut doit être supérieur de 5° au minimum à l'angle du niveau bas.

Commande <+> (signe plus)

La commande <+> augmente l'angle du niveau haut de 1°. L'angle du niveau bas doit être inférieur de 5° au minimum à l'angle du niveau haut.

Commande <9>

La commande <9> positionne le servomoteur à 90°. Cette commande positionne le servomoteur à mi-course permettant ainsi de régler la tringlerie. Faites ce réglage avant d'installer le servomoteur à son emplacement définitif.

Commande <C> (Change)

La commande <C> permet d'inverser la position du servomoteur entre ses angles extrêmes. Vous pourrez ainsi régler précisément les angles bas et haut en utilisant les touches – et + puis tester directement en entrant la commande <C> plusieurs fois.

Commande <I> (Inversion)

En fonction de la façon dont le servomoteur est installé sous le réseau, ses positions basse et haute peuvent être inversées par rapport à la position réelle de l'aiguille.

La commande <I> vous permet d'inverser directement la direction de rotation du servomoteur.

T-Command (Test)

La commande <T> vous permet de tester le servomoteur : il effectue alors un triple va et vient.

Commande (Speed / Vitesse)

Avec la commande <S> vous pouvez régler précisément la vitesse du servomoteur entre 5 ms (rapide) et 50 ms (lent) par degré de rotation. La valeur par défaut est de 25 ms, valeur par défaut qui peut être paramétrée avec la commande générale <S>.

```
Specify action (P/A/R/T/D/E/I/?): P
```

```
Specify port number
```

```
Enter value from 1 to 16: 4
```

```
Port is undefined. Specify DCC address
```

```
Set DCC address for port 4
```

```
Enter value from 1 to 2000: 32
```

```
DCC Address set to 32
```

```
Specify Accessory(1), Servo(2) or Input(3)
```

```
Enter value from 1 to 3: 2
```

```
Port 4 set as Servo port.
```

```
Servo on A32/P4 set to 75 degrees
```

```
Specify action for Servo on A32/P4
```

```
A/N/9/+/-/C/F/D/T/S/I/?/enter: -
```

```
LOW angle: 74 degrees
```

```
Specify action for Servo on A32/P4
```

```
A/N/9/+/-/C/F/D/T/S/I/?/enter: -
```

```
LOW angle: 73 degrees
```

```
Specify action for Servo on A32/P4
```

```
A/N/9/+/-/C/F/D/T/S/I/?/enter: C
```

```
Servo on A32/P4 set to 105 degrees
```

```
Angle set to HIGH: 105
```

```
Specify action for Servo on A32/P4
```

```
A/N/9/+/-/C/F/D/T/S/I/?/enter: +
```

```
HIGH angle: 106 degrees
```

```
Specify action for Servo on A32/P4
```

```
A/N/9/+/-/C/F/D/T/S/I/?/enter: +
```

```
HIGH angle: 107 degrees
```

```
Specify action for Servo on A32/P4
```

```
A/N/9/+/-/C/F/D/T/S/I/?/enter: N
```

```
Specify administrative note: W12
```

```
Administrative note set to W12
```

```
Specify action for Servo on A32/P4
```

```
A/N/9/+/-/C/F/D/T/S/I/?/enter: S
```

```
Set speed for servo on A32/P4
```

```
Enter value from 5 to 100 (25): 15
```

```
Speed set to 15 ms/step
```

```
Specify action for Servo on A32/P4
```

```
A/N/9/+/-/C/F/D/T/S/I/?/enter:
```

```
Port settings are saved!
```

Commande <F> (Frog point / Cœur d'aiguille)

Avec la commande <F> vous indiquez au programme que vous désirez utiliser au moins un relais pour la polarisation du cœur d'aiguille.

Ensuite, si vous utilisez un seul relais, entrez un <1>, ou un <2> pour deux relais. (Voir ci-contre).

Le programme va alors chercher le premier port libre en partant du port 16 afin de le sélectionner pour la connexion du premier relais.

```
Specify port number
Enter value from 1 to 16: 4
DCC 32, Servo W12, Angles 73/107, Not Inv., Speed 15, Frog port no
Angle set to HIGH: 107

Specify action for Servo on A32/P4
A/N/9/+/-/C/F/D/T/S/I/?/enter: f
How many relays ? (0/1/2): 2

Port 15 is set as first frogpoint port for servo on A32/P4, No inversion
Port 13 is set as second frogpoint port for servo on A32/P4, No inversion
```

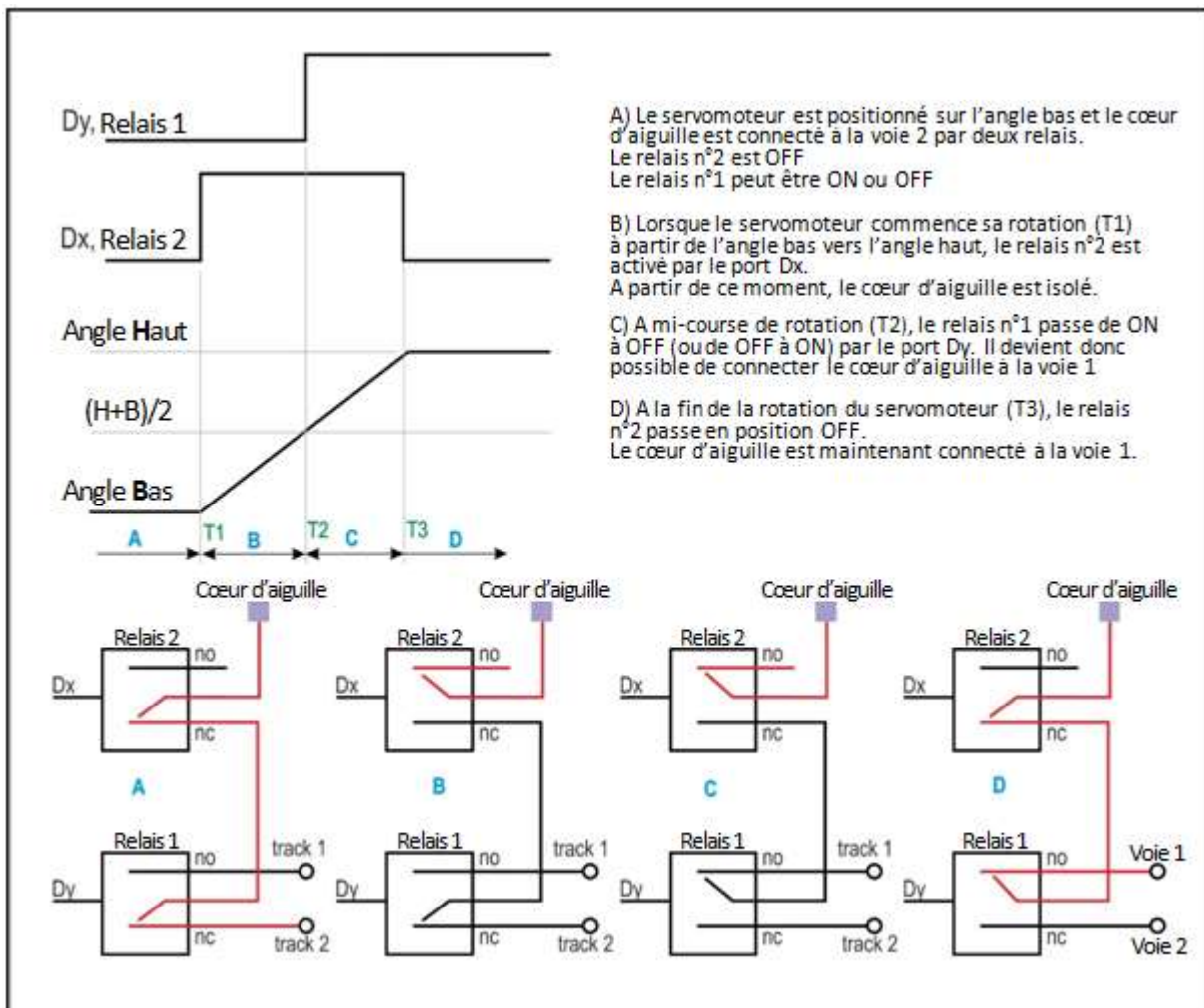
Le relais sera activé exactement à mi-course du servomoteur afin de permettre un changement de polarité sans risque. Le relais sera activé lorsque le servomoteur sera en mouvement vers son point haut et sera désactivé lors de son mouvement vers le point bas. Si l'ordre de ces actions doit être inversé, utilisez la commande <I> (Inversion).

Si la commande <F> concerne un port pour lequel un relais est déjà affecté, le programme vous demandera si vous voulez le supprimer (<R>) ou si vous voulez inverser le fonctionnement du premier relais (<1>) ou du second (<2>).

```
Specify action for Servo on A32/P4
A/N/9/+/-/C/F/D/T/S/I/?/enter: F

Remove frogpoint or Toggle Port relay inversion? R/1/2: 1
First Relay inverted
```

Pour plus de sécurité, vous pouvez utiliser deux relais. Ces deux relais travailleront de la manière suivante :



Certains relais sont activés par un passage au niveau bas (0 V) et non par un passage au niveau haut (5 V). Avec la commande <I> (Inversion), vous pourrez être sûrs que le relais ne sera activé qu'à mi-course du servomoteur.

PRÉCISION : Bien entendu, vous pouvez connecter autre chose qu'un relais sur le port dédié à la polarisation du cœur d'aiguille, par exemple faire fonctionner un clignotant ou un voyant lumineux qui indique la position de l'aiguille. Vous pouvez également utiliser le relais pour commuter autre chose qu'un cœur d'aiguille. Pour résumer, le port dédié au cœur d'aiguille est simplement un port qui devient haut ou bas simultanément au basculement de l'aiguille.

Commande <A> (Adresse - existe aussi pour les accessoires et les entrées)

La commande <A> permet de définir l'adresse du port. Si cette adresse est déjà utilisée, un message vous en avertira. En confirmant votre choix, vous pourrez contrôler plusieurs accessoires ou servomoteurs avec une seule adresse.

En spécifiant l'adresse 0 (zéro) pour un port précédemment paramétré, celui-ci sera alors reseté et considéré comme libre.

```
Set DCC address for port 4
Enter value from 0 to 2000 (32): 0
Do you want to reset port 4? Y/N: Y

DCC Address set to 0

Port settings are saved!
```

Commande <N> (Numéro / Note - existe aussi pour les accessoires et les entrées)

La commande <N> vous permet d'assigner (sur 4 caractères) une codification administrative au port. Si ce code a déjà été attribué à un autre servomoteur ou accessoires le programme affichera un avertissement.

Cette assignation n'est nullement obligatoire et n'aura aucune influence sur le paramétrage.

Commande <D> (Documentation)

La commande <D> affiche une vue d'ensemble de tous les paramètres.

Commande <Entrée>

La commande <Entrée> permet de sauvegarder les réglages du servomoteur et de revenir au menu des commandes générales. Vous pouvez ensuite sélectionner un autre port.

```
Set DCC address for port 4
Enter value from 1 to 2000: 44
DCC Address set to 44

Specify Accessory(1), Servo(2) or Input(3)
Enter value from 1 to 3: 1
Port 4 set as Accessory port.

Specify mode for this accessory
0=Help. Enter value from 1 to 10: 1
Mode set to Single Steady

Specify action for Accessory on A44/P4
enter/A/N/I/D/T/M/H/L/?/F/R:

Port settings are saved!
```

Les commandes destinées aux accessoires

Les données du signal DCC n'ont que deux valeurs possibles (0/1, on/off, bas/haut, rond/droit, rouge/vert). Le programme MARDEC peut réaliser différentes actions avec ces données.

Les actions 'doubles' présentées plus bas nécessitent l'usage d'un deuxième port, deuxième port appelé port jumelé. Ce port jumelé est automatiquement assigné lors de la configuration d'une action double.

Pour un port non précédemment configuré, il sera d'abord demandé la fonction désirée (= Mode). Pour un port déjà configuré, sa configuration actuelle sera affichée.

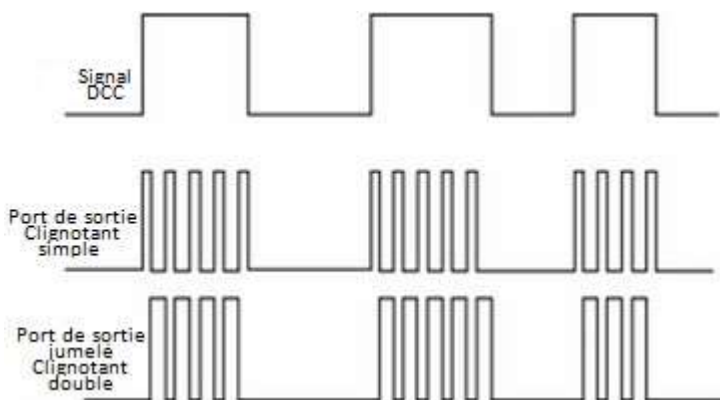
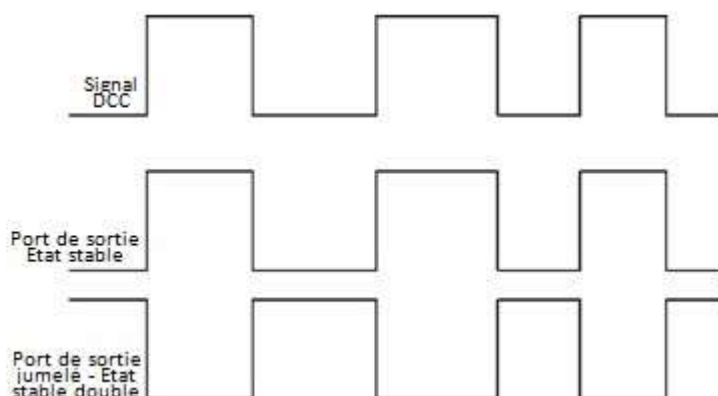
Les fonctions suivantes sont disponibles :

1. Etat stable

Le port suit le signal DCC.

2. Etat stable double

Le port suit le signal DCC et le port jumelé suit en prenant la valeur inverse.



3. Clignotement simple

Le port alterne en continu entre l'état HAUT et l'état BAS tant que le signal DCC est HAUT. Les durées ON/OFF sont réglables séparément entre 5 et 30 000 ms (= 30 secondes).

Double clignotement

Identique au clignotement simple mais avec un port jumelé qui clignote à l'inverse. Les deux ports retomberont à l'état BAS lorsque le signal DCC sera à l'état BAS. Voir également les commandes <L> (Low) et <H> (High). Une alternative qui n'utilise

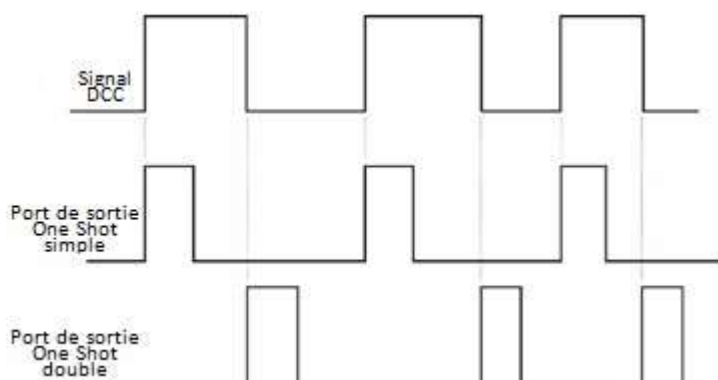
qu'un seul port est le montage de deux LEDs en antiparallèle. Voir les exemples de connexions.

4. Etat momentané "One Shot"

Avec l'état momentané "One Shot" le port sera mis à l'état HAUT pour une durée configurable de 5 à 30 000 ms lorsque le signal DCC passera à l'état HAUT. Si la même adresse est donnée à un autre port, les deux ports seront activés simultanément mais lorsque le signal DCC repassera à BAS, le deuxième port sera désactivé.

Exemple : vous avez configuré le port 5 avec l'adresse 12 comme étant de type "One Shot" et vous avez donné la même adresse 12 au port 7 que vous avez configuré comme étant un clignotant simple.

Si l'adresse 12 est activée, l'accessoire "One Shot" connecté sur le port 5 est actionné et la LED connectée sur le port 7 commence à clignoter. A la fin de l'activation de l'adresse 12, la LED du port 7 s'arrêtera de clignoter.



REMARQUE : si un port "One Shot" est lié avec un port configure comme étant de type "D" (voir les commandes d'entrée), une impulsion sera émise à chaque fois que le signal passé au niveau BAS. Ainsi une action quelconque peut être enclenchée sans DCC à l'aide d'un interrupteur momentané pour une durée maximale de 30 secondes. Exemple: un port "one shot" et un port de type "D" configures avec la même adresse 15.

Le port "One Shot" démarrera à chaque fois que le bouton sera pressé ou que l'adresse 12 sera activée.

5. Etat momentané double "Double One Shot"

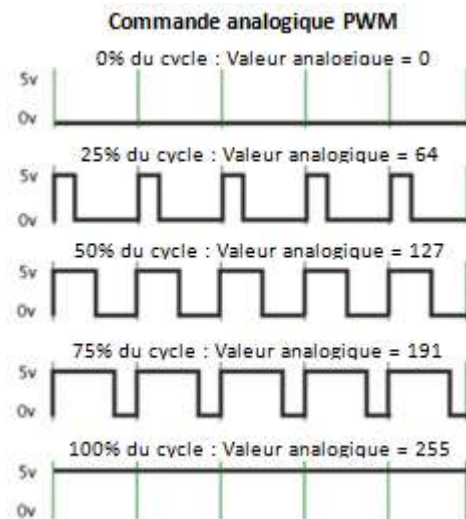
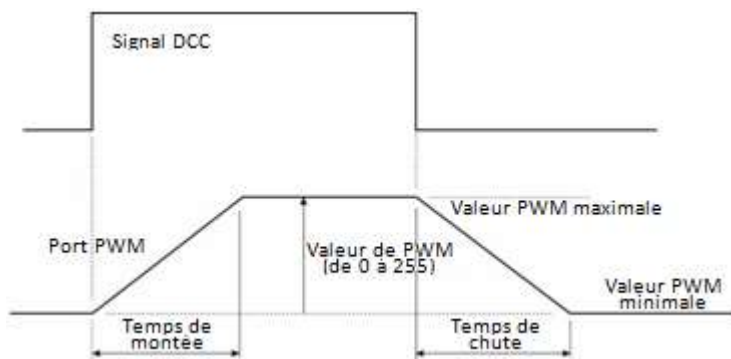
Identique à l'état momentané "One Shot" mais un deuxième port jumelé sera mis à l'état HAUT sur le front descendant du signal DCC.

Cet état est intéressant pour les aiguilles à solénoïdes.

Mais, attention ! Un solénoïde **ne peut pas** être connecté directement sur un port du décodeur car la puissance appelée est trop importante et la tension délivrée trop faible. Vous devez utiliser dans ce cas un amplificateur comme un [MOSFET](#) par exemple pour actionner les solénoïdes (voir les exemples page 26).

6. Commande analogique PWM

Les ports 1, 3, 4 et 9 (limitation inhérente à l'Arduino), peuvent être pilotés en mode PWM. Ainsi, et sur ces quatre ports, la tension (duty cycle) peut varier de 0 à 100% grâce à la largeur d'impulsion variable. Cette largeur d'impulsion se règle en "écrivant" une valeur de 0 à 255 sur ce port. Par exemple, en connectant une LED et en faisant varier la valeur PWM, la luminosité de cette LED peut varier. Cette possibilité peut être utilisée pour simuler les cycles jour/nuit par exemple.



- Vous disposez des paramètres suivants :
- Valeur maximale de la PWM (entre la valeur minimale et 255)
 - Valeur minimale de la PWM (entre 0 et la valeur maximale)
 - Le temps de montée (de 1 à 1000 secondes)
 - Le temps de chute (de 1 à 1000 secondes)

Avec ces paramètres, vous pouvez par exemple contrôler une bande de LEDs RGB. N'oubliez pas qu'une bande de LEDs nécessite un composant additionnel comme un transistor MOSFET entre elle et le décodeur. Voir également les commandes <H>, <L>, <R>, et <F> page suivante.

7. Mode scintillement

Avec ce mode, vous pouvez faire clignoter une LED avec la possibilité de régler indépendamment les luminosités minimale et maximale. En faisant clignoter trois DEL de couleurs différentes, vous pouvez simuler un incendie. Vous pouvez également utiliser cette possibilité pour simuler par exemple un poste de soudure à l'arc. L'emploi de la PWM n'est possible que sur les ports 1, 3, 4 et 9 (limitation inhérente à l'Arduino).

8. Mode aléatoire long

Une LED connectée sur un port ayant cette configuration sera allumée et éteinte de façon aléatoire tant que le signal DCC est au niveau HAUT. Chaque période d'activation et de désactivation est aléatoire mais les durées minimale et maximale peuvent être réglées entre 1 et 1000 secondes.

Ceci peut être utilisé, par exemple, pour allumer et éteindre l'éclairage intérieur des maisons ou l'éclairage public.

9. Mode aléatoire court

Identique dans son fonctionnement au mode aléatoire long mais 1000 plus rapide. Les durées minimale et maximale peuvent être réglées entre 5 et 5000 millisecondes.

Une fois la fonction désirée sélectionnée, il est possible d'assigner un port jumelé et de régler ses durées de marche/arrêt ou ses paramètres PWM, en fonction du mode sélectionné.

La sélection d'un port déjà configuré comme sortie d'accessoire affiche ses réglages actuels. Les commandes suivantes peuvent être utilisées pour en modifier les paramètres.

Commande <M> (Mode)

Avec la commande <M> vous pouvez modifier la fonction d'un port pour accessoire. Cette commande a son propre écran d'aide qui s'affiche avec <0> (zéro) :

```
Specify mode for this accessory
0=Help. Enter value from 0 to 10 (7): 0
Available Modes:
1:Single steady   2:Double steady
3:Single flashing 4:Double flashing
5:Single One shot 6:Double One shot
7:Analog PWM     8:Flickering
9:Random Long    10:Random Short
Enter value from 0 to 10 (7): █
```

H-Commande <H> (Hight / Haut)

Pour les fonctions 3,4, 5 et 6, cette commande règle la durée de fonctionnement. Elle peut varier entre 5 et 30 000 ms.

Pour la fonction 7, elle permet de régler la valeur maximale de PWM.

Pour les fonctions 9 et 10, elle règle la durée maximale.

Commande <L> (Low / Bas)

Pour les fonctions 3 et 4, cette commande règle la durée de l'arrêt. Elle peut varier entre 5 et 30 000 ms.

Pour la fonction 7, elle permet de régler la valeur minimale de PWM.

Pour les fonctions 9 et 10, elle règle la durée minimale.

Commande <R> (Rise / Montée)

Concerne uniquement la fonction 7 : elle permet de régler le temps de « montée ».

C'est la durée pendant laquelle la valeur PWM passe de la valeur la plus basse à la valeur la plus haute.

Commande <F> (Fall / Chute)

Concerne uniquement la fonction 7 : elle permet de régler le temps de « chute ».

C'est la durée pendant laquelle la valeur PWM passe de la valeur la plus haute à la valeur la plus basse.

Commande <I> (Inversion)

Avec la commande <I>, vous pouvez inverser la sortie d'un port. Cela signifie que si un port doit être normalement HAUT, il sera maintenant BAS et vice versa. Ainsi, par exemple, si une LED s'éteint au lieu de s'allumer, avec la commande <I> vous pouvez inverser son fonctionnement.

Autres commandes

Commande <T> : test de fonctionnement des accessoires (sauf pour la fonction 9)

Commande <N> : Permet de saisir un code (numéro) administratif pour l'accessoire.

Commande <A> : définir l'adresse DCC du port ou libérer le port en lui donnant l'adresse 0 (zéro).

Commande <D> : Affiche une vue détaillée des réglages.

```
Specify action (P/A/R/T/D/E/I/?): p
Specify port number
Enter value from 1 to 16: 3
Port is undefined. Specify DCC address

Set DCC address for port 3
Enter value from 1 to 2000: 20
DCC Address set to 20

Specify Accessory(1), Servo(2) or Input(3)
Enter value from 1 to 3: 1
Port 3 set as Accessory port.

Specify mode for this accessory
0=Help. Enter value from 1 to 10: 7
Mode set to Analog PWM value

Set Minimum PWM value for port on A20/P3
Enter value from 0 to 255: 5
Minimum PWM value set to 5

Set Maximum PWM value for port on A20/P3
Enter value from 5 to 255 (255): 150
Maximum PWM value set to 150

Set Rise time (sec) for PWM port
Enter value from 1 to 1000: 600
Rise time set to 600 sec.

Set Fall time (sec) for PWM port
Enter value from 1 to 1000: 500
Fall time set to 500 sec.

Specify action for Accessory on A20/P3
enter/A/N/I/D/T/M/H/L/?/F/R:

Port settings are saved!
```

Divers

Démarrage

A la mise sous tension, la LED d'état clignote brièvement trois fois signifiant le début du processus de démarrage. Le décodeur redémarrera automatiquement lorsque vous passez du mode exploitation au mode configuration. A partir de la version 3.0 du programme MARDEC, le décodeur ne redémarre plus lors du passage du mode configuration au mode exploitation.

Installation et réglage des servomoteurs

Pour mettre en place et configurer un servomoteur, procédez comme suit :

- Connectez le servomoteur au décodeur et positionnez le à 90 degrés avec la commande <9> ;
- Installez maintenant le servomoteur à son emplacement sous le réseau et réglez la timonerie de telle sorte que les lames mobiles de l'aiguille soient positionnées à mi-course de leur zone de déplacement ;
- Utilisez la commande <C> pour positionner le servomoteur en butée ;
- Utilisez les commandes <+> et <-> pour corriger la position de cette butée ;
- Utilisez à nouveau la commande <C> pour passer à l'autre butée ;
- Utilisez à nouveau les commandes <+> et <-> pour corriger la position de cette butée ;
- Concernant la polarisation du cœur d'aiguille, vérifiez la concordance de polarisation entre la pointe d'aiguille et les rails. S'il y a discordance, inversez les connexions du relais gérant la polarité de la pointe de cœur ;
- Vérifiez que le sens du mouvement de l'aiguille correspond à ce qui est indiqué par la centrale ou le programme. Si la position indiquée est inversée, alors inversez le servomoteur ;
- Terminez avec la touche <Entrée> pour enregistrer les paramètres.

Arrêt et sortie du programme Putty

Le programme Putty peut être fermé à tout moment en utilisant donc la combinaison classique <Alt-F4> ou en cliquant sur la croix habituelle en haut à droite de la fenêtre.

Assurez-vous que vous êtes dans le mode exploitation avant de quitter Putty afin que les dernières modifications éventuellement apportées à un port soient bien enregistrées.

Fichier de log

Toutes les actions effectuées lors d'une opération de configuration sont enregistrées dans un fichier dont l'emplacement et le nom sont **Mes documents/Mardec/MARDEC_<date>_<time>.log**, en sachant que <date>_<time> correspond à la date et l'heure de sortie du mode configuration.

Documentation personnelle

Une fois la configuration terminée, il est recommandé de garder une copie de tous les paramètres dans un document. A cet effet, vous pouvez effectuer des copies écran des différents paramètres que vous sauvegarderez ou imprimerez.

Arduino Nano, Arduino Pro Mini et Arduino Mega2560

Vous pouvez utiliser un Arduino Nano ou un Arduino Pro Mini de la même manière qu'un UNO. Mais, bien sûr, vous ne pourrez pas utiliser la carte fille (shield) DCC/Power.

Pour télécharger Mardec sur un Nano, vous devez utiliser l'IDE Arduino. Mais, comme le bootloader par défaut est trop volumineux, laissant ainsi trop peu de mémoire disponible pour le programme Mardec. Vous devez dans ce cas remplacer le bootloader. Téléchargez et consultez le document 'réparer le bootloader ' sur <https://www.arcomora.com/download/>

Une fois ceci fait, vous pourrez utiliser l'EDI Arduino pour télécharger le fichier .INO.

Pour télécharger Mardec sur un Mega2560, utilisez l'outil 'Upload program to Arduino' dans le menu Démarrer de Windows. Vous pouvez utiliser sans aucune modification la carte fille (shield) DCC/Power avec le Mega2560.

Avec le Mega2560, les ports 11 à 16 de la carte fille (shield) DCC/Power sont connectés aux broches 54 à 59 du Mega2560. Cependant, Mardec V.5 contrôlera les broches 54 à 59 du Mega2560 comme s'il s'agissait des ports 11 à 16. Vous pouvez donc utiliser les borniers à vis de 11 à 16 comme s'il s'agissait d'un UNO.

Configuration à partir du deuxième décodeur

Après l'installation du logiciel, lors du premier lancement du programme de configuration MARDEC, le programme destiné à l'Arduino sera automatiquement téléversé dans celui-ci, et par défaut comme si c'était un Arduino UNO.

Pour le deuxième décodeur (et pour les suivants), la procédure est différente puisqu'il n'y a plus d'installation. Donc, pour configurer les décodeurs suivants, utilisez le raccourci "Upload".

Ce raccourci doit également être utilisé si vous utilisez un Mega2560 puisque l'installation par défaut suppose un téléversement vers un UNO.

Vous le trouverez dans le dossier ArCoMoRa du menu Démarrer de Windows, dossier dans lequel figure également un manuel.

Modifier le port COM

Il peut arriver qu'un port COM de Windows soit modifié lors de la connexion d'un Arduino.

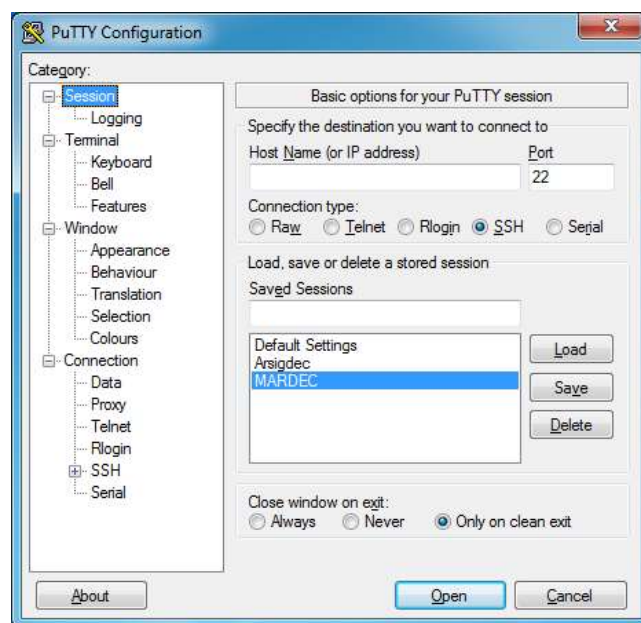
Avec l'outil 'Change COM port' vous pourrez facilement ajuster ce paramètre de port COM.

Vous le trouverez également dans le menu Démarrer de Windows.

Configuration de Putty

Vous pouvez régler l'affichage de Putty comme suit :

1. Cliquez sur le raccourci *Configure Putty*.
Vous le trouverez dans le dossier ArCoMoRa du menu Démarrer de Windows dans le dossier ArCoMoRa.
L'écran ci-contre apparaîtra :
2. Cliquez sur MARDEC puis sur le bouton Load.
3. Dans la fenêtre Session ➤ Logging, vous pouvez modifier les options de log.
Attention : si vous changez le nom du fichier, les données des sessions de configuration ne pourront plus être sauvegardées !
4. Dans la fenêtre Windows ➤ Appearance / Behavior / Colors, vous pouvez modifier l'apparence de Putty.
5. Si nécessaire, vous pourrez également modifier ici le port COM.
6. **Ne changez rien aux autres paramètres !**
7. Sélectionnez Session et cliquez sur Save pour enregistrer les nouveaux paramètres.
8. Cliquez sur Open pour démarrer Putty.



Vous pouvez également changer facilement le port COM d'un Arduino à l'aide du programme 'Change port COM'. Vous le trouverez dans le dossier ArCoMoRa du menu Démarrer de Windows.

Utilisation des modules relais

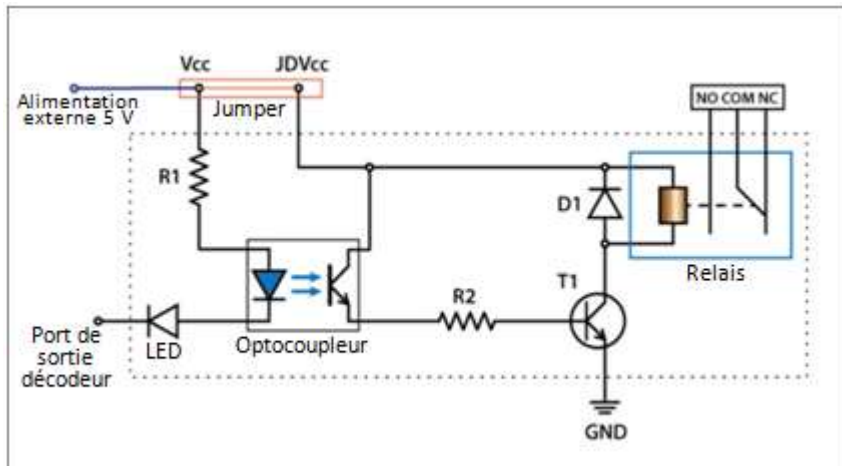
Si vous utilisez des modules relais avec optocoupleur comme mentionné page 8, il peut être important de savoir que ceux-ci sont actifs au niveau BAS.

Cela signifie que le relais est excité lorsque la tension sur le port du décodeur est de 0 volt.

Il y a un cavalier standard sur les ports Vcc et JDVcc.

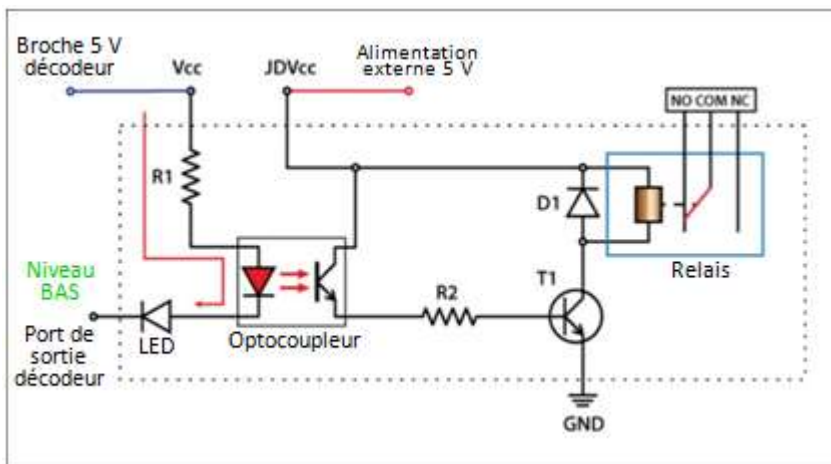
Concernant ce cavalier, vous avez deux options :

1) Laisser le cavalier en place et connecter les broches Vcc/JDVcc à une alimentation externe :



2) Retirer le cavalier et :

- connecter la broche Vcc à une sortie 5 V du décodeur
- et raccorder la broche JDVcc à une alimentation extérieure de 5 V :



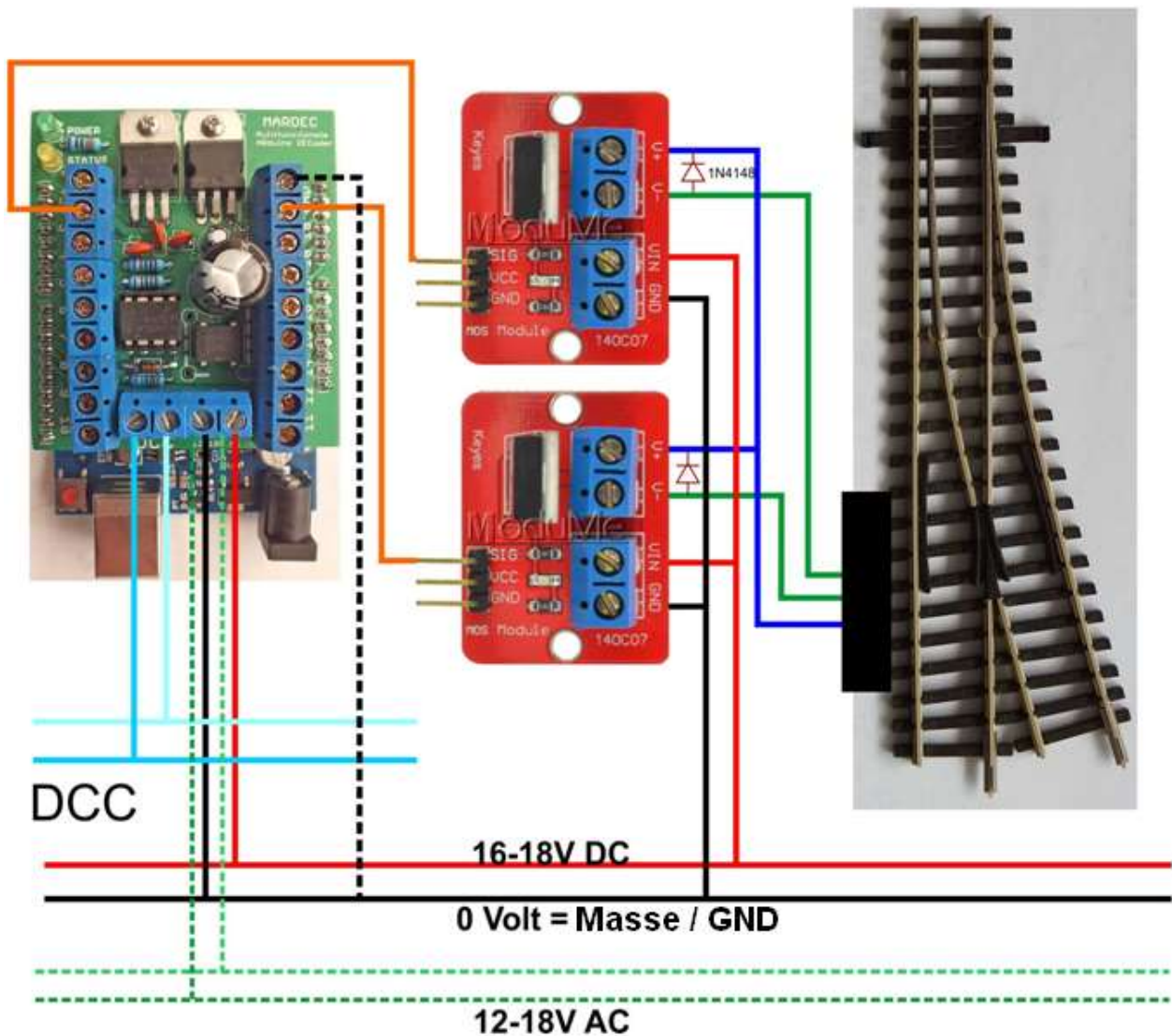
Dans cette image, vous voyez également que le relais est excité lorsque le port de sortie du décodeur est BAS.

En aucun cas, dans cette configuration, il ne faut remettre en place le cavalier car cela reviendrait à connecter directement les broches Vcc / JDVcc au décodeur.

Dans cette hypothèse, la demande de courant lorsque la bobine du relais sera excitée passera directement par le décodeur et ce n'est pas prudent.

Exemples de configurations

Connexion d'une aiguille à deux solénoïdes à l'aide de deux transistors MOSFETS



Dans cet exemple, le décodeur est alimenté par le même circuit que les aiguilles, mais le décodeur peut également être alimenté en courant alternatif.

N'oubliez pas de connecter les masses (GND) ensemble.

Nota : Les ports du décodeur qui doivent être utilisés dépendent de la configuration de celui-ci.

Les connexions montrées ici ne sont qu'un exemple.

Configuration :

```
Specify action (P/R/T/D/M/E/T): p
Specify pin number (3-12,14-19): 3
Pin is undefined. First specify DCC address

Set DCC address for pin 3
Enter value from 1 to 2048: 5
DCC Address set to 5

Specify Accessory(1), Servo(2) or Input(3)
Enter value from 1 to 3: 1
Pin 3 set as Accessory pin.

Specify mode for this accessory
D-Help. Enter value from 1 to 10: 6
Mode set to Double One shot.

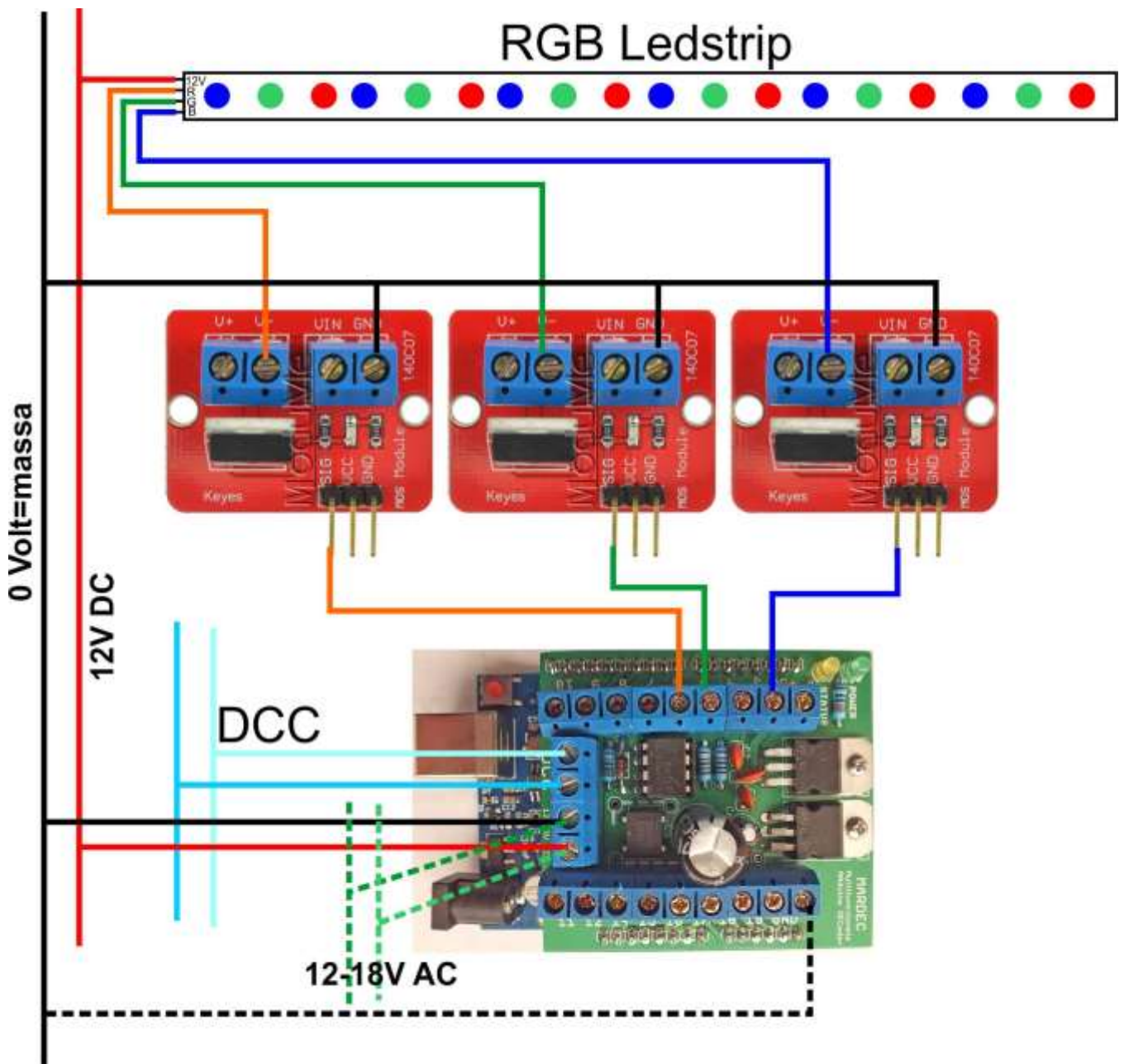
Pin 19 is assigned as buddy pin to accessory on pin 3
Specify 'On' time
Enter value from 20 to 5000: 300
'On' time set to 300 msec.

Specify action for Accessory on pin 3
enter/A/N/D/T/M/L/T/E/R:
Pin settings are saved!
```

```
Specify action (P/R/T/D/M/E/T): d
Settings of MARDEC #1
Pin 3: DCC 5, Acc.type 6 (Double One shot), #0. Buddy 19. Time(ms) 300
Pin 4: not used
Pin 5: not used
Pin 6: not used
Pin 7: not used
Pin 8: not used
Pin 9: not used
Pin 10: not used
Pin 11: not used
Pin 12: not used
Pin 14: not used
Pin 15: not used
Pin 16: not used
Pin 17: not used
Pin 18: not used
Pin 19: Buddy pin for pin 3

Default: Servo speed: 25 ms/step
```

Connexion d'une bande de LED RGB au moyen d'amplificateurs MOSFET



Dans cet exemple, le décodeur est alimenté par le même circuit que la bande de LEDs, mais le décodeur peut également être alimenté en courant alternatif.

N'oubliez pas de connecter les masses (GND) ensemble.

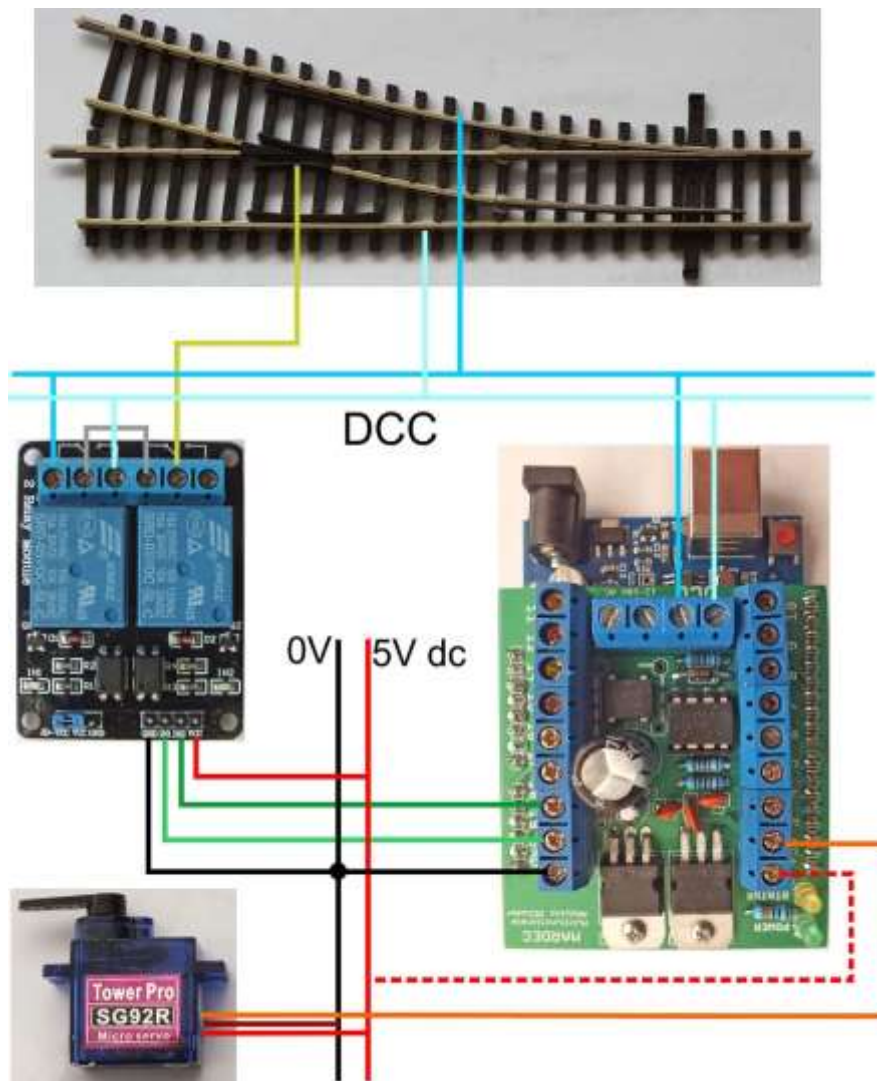
Nota : Les ports du décodeur qui doivent être utilisés dépendent de la configuration de celui-ci.

Les connexions montrées ici ne sont qu'un exemple.

Configuration :

```
Port 1: DCC 12, Acc.type 7 (PWM), , Not Inv., PWM 5/150, Time R/F(sec) 800/800
Port 2: not used
Port 3: DCC 12, Acc.type 7 (PWM), , Not Inv., PWM 5/150, Time R/F(sec) 800/800
Port 4: DCC 12, Acc.type 7 (PWM), , Not Inv., PWM 5/150, Time R/F(sec) 800/800
Port 5: not used
```


Connexion d'un servomoteur et d'un double relais à une aiguille



Les relais peuvent être alimentés par le 5 V disponible sur la carte fille DCC, mais une alimentation externe est recommandée pour le servomoteur.

Dans cet exemple, aucune alimentation électrique n'a été connectée au décodeur.

Nota : Les ports du décodeur qui doivent être utilisés dépendent de la configuration de celui-ci.

Les connexions montrées ici ne sont qu'un exemple.

```
Specify action (P/R/T/D/M/E/?): p
Specify pin number (3-12,14-19): 17
Pin is undefined. First specify DCC address.

Set DCC address for pin 17
Enter value from 1 to 2048: 34
DCC Address set to 34

Specify Accessory(1), Servo(2) or Input(3)
Enter value from 1 to 3: 2
Pin 17 set as Servo pin.
Servo on address 34 set to 75 degrees

Specify action for Servo on pin 17
A/N/9/+/-/C/F/D/T/M/I/?/enter: f
Do you want frogpoint polarisation for servo on pin 17 ?
(<enter>=No; 1,2=Yes) Enter value from 1 to 2: 2
Do you want inversion for first relay ? Y/N y

Pin 19 is set as frogpin for servo on pin 17, with inversion

Do you need inversion for second relay ? Y/N n
Pin 18 is set as second frogpin
```

```
Pin 17: DCC 34, Servo #0, Angles 75/105, Frog pin 19/18, no Inversion, Speed 25
Pin 18: Second relay (No inv.) for pin 17
Pin 19: First relay (Inv.) for pin 17
```