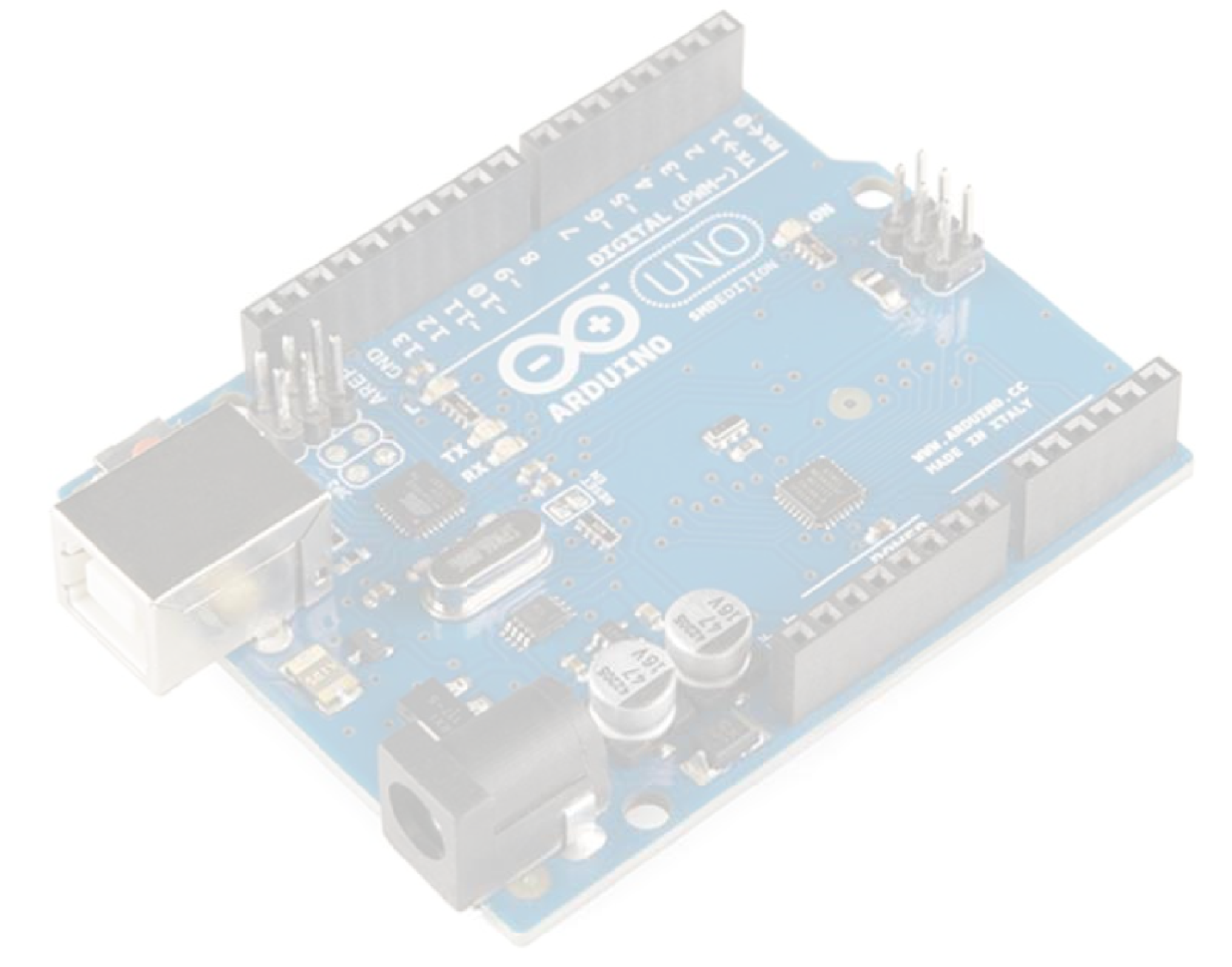
**M**ultifonction**AR**duino dcc**DEC**oderd Le Multi fonctionnel

DCC decoder pour

servos et accessoires

avec Arduino

pour tout le monde

(met een DCC centrale)

Auteur: Nico Teering

September 2017

Mardec versie: 3.3.1

Document versie: 3.3.1A

Info: MARDEC@Ziggo.nl of [Arcomora.wordpress.com/Contact](http://arcomora.wordpress.com/Contact)

[Arcomora.wordpress.com/Mardec/](http://arcomora.wordpress.com/Mardec)

Introduction...............................................................................................................................................................4

Pourquoi un Arduino? ......................................................................................................................................... 4

ArCoMoRa................................................................................................................................................................ 4

Toutes les possibilités d’un MARDEC en un coup d’oeil.......................................................................................... 5

De quoi avez-vous besoin? ...................................................................................................................................... 6

Le matériel ........................................................................................................................................................... 6

DCC circuit............................................................................................................................................................ 7

Relais ................................................................................................................................................................... 7

**Commutateur MOSFET**  ...................................................................................................................................... 8

Logiciel… .............................................................................................................................................................. 8

Diagramme de connexion MARDEC......................................................................................................................... 9

Configuration du MARDEC...................................................................................................................................... 10

Les commandes...................................................................................................................................................... 10

Commandes générales........................................................................................................................................... 11

P-Commande (Pin) ............................................................................................................................................. 11

T-Commande (Test) .......................................................................................................................................... 11

M-Commande (Vitesse de déplacement) .......................................................................................................... 11

D-Commande (document) ................................................................................................................................. 11

R-Commande (Réinitialiser)................................................................................................................................ 11

E-Commande (Sortie) ........................................................................................................................................ 12

Retour à la configuration.................................................................................................................................... 12

IMPORTANT........................................................................................................................................................ 12

Commandes d'entrée............................................................................................................................................. 13

2- et 3- Commande (deuxième et troisième adresse) ....................................................................................... 13

A-Command (Adresse, également pour les accessoires et les servos)............................................................... 14

N-Command (numéro, également pour les accessoires et les servos) ……………………………………………………………14

D-commande (Document) ................................................................................................................................. 14

Tests................................................................................................................................................................... 14

Commande servo................................................................................................................................................... 15

- (min) Commande ............................................................................................................................................. 15

+ (plus) Commande ............................................................................................................................................ 15

9-Commande ..................................................................................................................................................... 15

C-Commande (Change) ...................................................................................................................................... 15

I-Commande (Invertion) ..................................................................................................................................... 15

T-Commande (Test) ............................................................................................................................................ 15

M-Commande (vitesse de déplacement) ........................................................................................................... 15

F-Commande (Point de grenouille = puntstuk)................................................................................................... 16

A-Commande (Adresse, accessoires en entrée)................................................................................................. 17

N-Commande (Numéro, accessoires pour les entrées)...................................................................................... 17

D-commande (Document) ................................................................................................................................. 17

<enter>-Commande ........................................................................................................................................... 17 Commande d'Accessoire.................................................................................................................................... 18

M-Commande (Mode) ....................................................................................................................................... 20

H-Commande (Haut) .......................................................................................................................................... 20

L-Commande (Bas) ............................................................................................................................................. 20

R-commande (Rise) ............................................................................................................................................ 20

F-Commande (Chute) ......................................................................................................................................... 20

Autres commandes............................................................................................................................................. 20

Autres questions .................................................................................................................................................... 21

Demarrer............................................................................................................................................................. 21

Sortie .................................................................................................................................................................. 21

Enregistrement .................................................................................................................................................. 21

Documentation .................................................................................................................................................. 21

Configuration des décodeurs suivants ............................................................................................................... 21

Écran de configuration des paramètres (Putty................................................................................................... 22

Arduino Pro Mini ................................................................................................................................................ 23

Relais modules ................................................................................................................................................... 24

Connexion de l'amplificateur MOSFET pour changements de bobine............................................................... 25

Connexion d'une bande de LED RGB avec des amplificateurs MOSFET............................................................. 26

Pièce de connexion avec double relais et servo................................................................................................. 27

Historique des versions ......................................................................................................................................... 28

[**DOWNLOAD HIER DE ARCOMORA SOFTWARE**](http://www.globalvisuals.nl/downloads/Arcomora.zip)

Bibliothèque DCC et codage d'échantillons fournis par Mynabay.  
Le code source, ou toute partie de celui-ci, et le fichier Mardec.hex ne peuvent être utilisés dans aucun produit commercial.  
Il ne peut pas non plus être modifié dans un projet privé sans en informer l'auteur.  
S'il vous plaît envoyez un courriel à MARDEC@ziggo.nl si vous avez des commentaires, des questions ou des suggestions d'amélioration sur ce programme.

# Introduction

L'utilisation d'un micro-ordinateur Arduino dans la construction de modélisme ferroviaire est bien connue de beaucoup.  
Cependant, en fait l'appliquer vous-même est un pont trop loin pour la plupart des gens. Programmation ou faire des ajustements simples à un programme est considéré comme un gros problème.  
En conséquence, l'Arduino reste le domaine d'un groupe très limité avec plus de connaissances techniques. C'est dommage, car avec l'Arduino, vous pouvez automatiser à bon prix votre modèle de chemin de fer. Pour rendre l'Arduino accessible à tous, j'ai donc réalisé un programme pour un décodeur DCC multifonctionnel basé sur un Arduino (MARDEC). Avec le MARDEC, les deux servos d'aiguillage et autres accessoires peuvent être utilisés.  
Avec MARDEC, il n'est plus nécessaire d'écrire une lettre de code Arduino.

Comme avec tout autre décodeur DCC, le MARDEC doit également être configuré.  
Cette configuration est généralement appelée programmation incorrecte. Pour éviter toute confusion, nous utilisons le terme programmation pour écrire le code Arduino et configurer le terme pour exécuter le programme. Les servos et accessoires sont configurés lors de cette configuration.

Le MARDEC fonctionne dans deux «états» différents.  
En mode de configuration, les servos / accessoires peuvent être réglés au moyen de commandes à une lettre du clavier. Le MARDEC communique avec l'ordinateur via un câble USB.  
Aucun signal DCC n'est requis pour cela.  
Au moyen d'une commande spécifique ('E'), nous passons à l'état de fonctionnement.  
En état de fonctionnement, les servos et accessoires sont commandés avec le signal DCC. Aucun câble USB n'est nécessaire

Pourquoi un Arduino?

La raison principale est le prix. Prenez une note:  
 • Un (clone) Arduino coûte au maximum 3,50 € sans câble USB.  
 • Un servo (SG92R) coûte environ 2,30 €.  
 • Deux relais pour la polarisation ponctuelle, y compris le contrôle par optocoupleur, coûtent environ 1,50 €.  
 • Câble de connexion Dupont pour 1,00 €.  
 • Les composants séparés supplémentaires pour le traitement du signal DCC coûtent 2,00 €.  
Pour 9 points avec polarisation ponctuelle, cela coûte: 3,00 € + 9x 2,30 € + 5x 1,50 € + 1,00 € + 2,00 € = 34,20 €.  
C'est environ 3,80 € par changement, y compris les servos et les relais. Tous les logiciels sont gratuits.  
Aucun produit commercial ne peut répondre à cela.  
Pour 12 points sans polarisation ponctuelle, il en coûte 2,80 € par changement, y compris le servo.

Une autre raison est la facilité avec laquelle tout peut être configuré. Ceci est complètement interactif et aucun centre DCC n'est requis

L'installation de tous les logiciels est très conviviale. Tous les logiciels requis sont installés en même temps. En outre, le téléchargement du programme de décodeur sur l'Arduino est presque entièrement automatique. Voir le manuel d'installation pour cette partie.

# ArCoMoRa

Le Mardec fait partie du concept ArCoMoRa. Cela signifie Arduino Controlled Model Railway. L'Arsigdec, un décodeur de signal DCC, en fait partie ainsi que l'ArLoco, un Feedback LocoNet configurable avec Arduino. L'ASAR sera également inclus. Voir aussi:[Arcomora.wordpress.com](http://arcomora.wordpress.com/)

Toutes les possibilités du MARDEC en un coup d'œil

Le décodeur MARDEC propose les options suivantes:

• Commande jusqu'à 12 servomoteurs pour entraînement alternatif sans polarisation ponctuelle ou jusqu'à 8 servomoteurs avec polarisation ponctuelle. La polarisation ponctuelle est réalisée au moyen d'un relais externe.  
• Contrôle des accessoires de 10 manières différentes, y compris le contrôle des bobines alternées et le contrôle PWM variable.  
• Une broche Arduino peut être configurée en entrée. Avec ceci, un servo / accessoire peut être allumé ou éteint sur une autre broche.  
• Interactivement, via l'écran et le clavier, pour configurer les servos et accessoires. Ceci est complètement indépendant du centre DCC utilisé. Le logiciel Arduino n'est PAS nécessaire ici.  
• Les angles de début et de fin de chaque servo peuvent être réglés séparément selon le degré.  
• Attribuez une adresse DCC aléatoire (1-2048) à chaque servo ou accessoire.  
Donc, pas nécessairement des adresses consécutives.  
• Chaque servo (jusqu'à 8) peut être connecté avec un relais pour la polarisation ponctuelle.  
Lorsque le servo est tourné, ce relais sera converti à mi-chemin de la rotation.  
• Chaque servo (maximum 5) peut être connecté avec deux relais pour la polarisation ponctuelle. Cela se traduit par un changement encore plus fiable de la pièce de pointe.  
• Une vitesse de rotation séparée peut être définie pour chaque servo. Cela rend plus facile l'utilisation des servos à d'autres fins que la simple participation  
• Une option de test. Dans ce cas, tous les servos vont et viennent deux fois et les accessoires sont activés.  
• Capture 'inversion' pour les servos. Ceci peut être utilisé pour définir si un commutateur doit être redressé ou dévié lorsque, par exemple, une rotation est effectuée sur le plus petit angle. Ceci est nécessaire car le servo peut être monté de plusieurs façons.  
• Une option de documentation qui montre tous les paramètres.  
• Attribution d'un numéro administratif à chaque servo / accessoire.  
• Une option de réinitialisation dans laquelle tous les paramètres sont supprimés de la mémoire du MARDEC.  
• Vitesse par défaut réglable du bras de servo (5-50 ms par degré).  
• Revenez en mode de configuration en connectant le câble USB et en entrant la commande 'C'. • Possibilité de corriger le décalage d'adresse dans les centrales Roco (MM, z / Z21) • Les accessoires ont les options suivantes:

1 Simple stable. Ici, une broche est toujours haute ou basse selon le contrôle

2 Double stable. Identique à une seule constante, mais une seconde broche a la valeur «inverse».  
 3 Clignotement unique. Ici une épingle alterne alternativement haut (on) et bas (off). L'heure d'activation et l'heure d'arrêt peuvent être ajustées individuellement.  
 4 Double clignotant. Identique au double simple, mais une seconde broche a la valeur 'inverse'.  
 5 Un tir unique. Dans ce cas, une broche a un temps réglable aussi court que le contrôle passe de bas en haut. Il est également possible de connecter un servo / accessoire à une autre broche à la fin de l'impulsion.  
 6 Double coup. Dans ce cas, une broche une courte durée réglable devient 'haute' quand la commande passe de bas en haut et une autre broche quand elle passe de haut en bas. Avec cela, des bobines alternées peuvent également être contrôlées.  
 7 Contrôle analogique (PWM = Pulse Width Modulation). Dans ce cas, une broche se déplace, dans un temps réglable séparément, de 0 à un maximum réglable (maximum 255) lorsque le contrôle passe de bas en haut et de nouveau à 0 dans un temps réglable séparément si le contrôle passe de haut en bas.  
 8 Mode scintillement. Avec cela, vous pouvez faire un scintillement de LED connecté. Avec des LED appropriées, vous pouvez simuler, par exemple, un feu ou une lumière de soudage.  
 9. Aléatoire activé / désactivé. Avec ceci, vous pouvez allumer et éteindre en permanence une LED connectée. Les temps d'activation et de désactivation sont déterminés arbitrairement (au hasard) entre deux limites ajustables (2-500 sec.)

* 1. Lumières disco. Avec cela, vous pouvez allumer et éteindre une led rapidement entre deux limites réglables. (20-500 millisec.)  
  • Une option d'aide affiche toutes les commandes pour la configuration.  
  • Une option de journal qui préserve toutes les sessions de configuration.  
  • Indépendant de la structure de bus utilisée (LocoNet, S88 et autres). Cependant, le contrôle est spécifique à DCC.

DE QUOI AVEZ-VOUS BESOIN ?

## Le matériel

En premier lieu bien sur [un Arduino UNO,](https://www.arduino.cc/en/Main/ArduinoBoardUno)  [Arduino Pro Mini,](https://www.arduino.cc/en/Main/ArduinoBoardProMini) [Arduino Mega2560](https://www.arduino.cc/en/Main/ArduinoBoardMega2560) ou [Arduino Nano](https://www.arduino.cc/en/Main/ArduinoBoardNano)

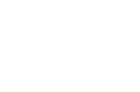
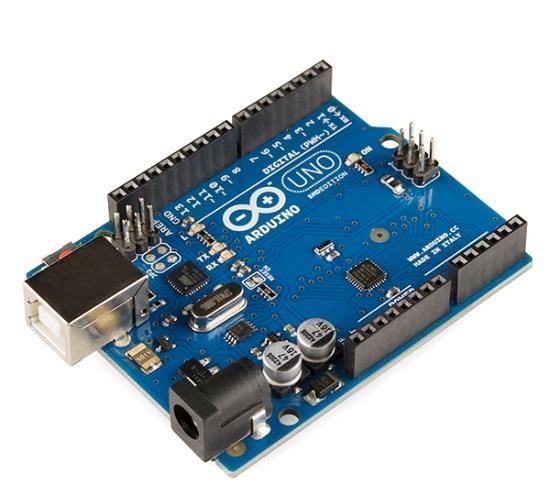
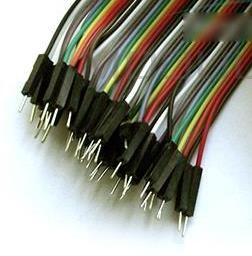
Dans ce manuel, nous allons plus loin de l'UNO Arduino (figure 1)

Un Arduino est un micro-ordinateur avec beaucoup de connexions. Ce sont des broches.

Ceux-ci sont numérotés sur le PCB.

Nous utilisons le maximum du décodeur MARDEC

Connexion de ces 16 broches. Sur ces broches, vous pouvez directement brancher des servos, des modules relais (voir ci-dessous) et des LED (avec résistance de commutation de 330 ohms) sont connectés.



Une broche Arduino a une tension de sortie de 0 ou 5 volts et peut supporter un maximum de 40 mA. Pour les accessoires nécessitant plus de puissance (tels que les moteurs, les bobines alternées et les bandes LED), une étape d'amplification supplémentaire est nécessaire.

Pou Pour les applications analogiques, il y a 4 broches s pour lesquelles une tension de bloc peut être réglée a avec une largeur d'impulsion variable.

Connexions vie des câbles Dupont.

USB raccordement

Les broches du câble sont insérés dans les connecteurs noirs.

9-12 volt entrée Les câbles sont disponibles en différentes longueurs et

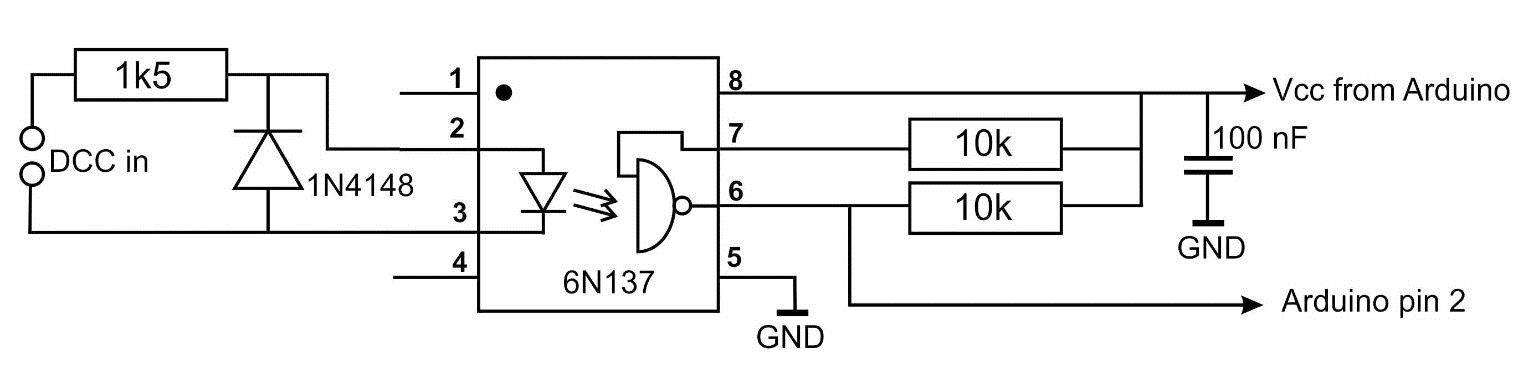
combinaisons de diamètres (M-M, M-V, V-V)

. Sur la gauche, vous voyez la connexion USB. L'Arduino est alimenté par ce câble USB.  
La connexion USB est requise pour copier le programme sur l'Arduino. Il est également nécessaire de communiquer avec l'ordinateur lors de la configuration.  
Si aucun câble USB n'est connecté, l'alimentation doit se faire via l'autre prise noire (prise d'alimentation). Une source de tension de 7-12 Volts DC peut être connectée à cela. L'Arduino lui-même fait une tension stabilisée de 5 volts. Les deux connexions peuvent être utilisées en même temps. Pour les servos, nous n'utilisons que

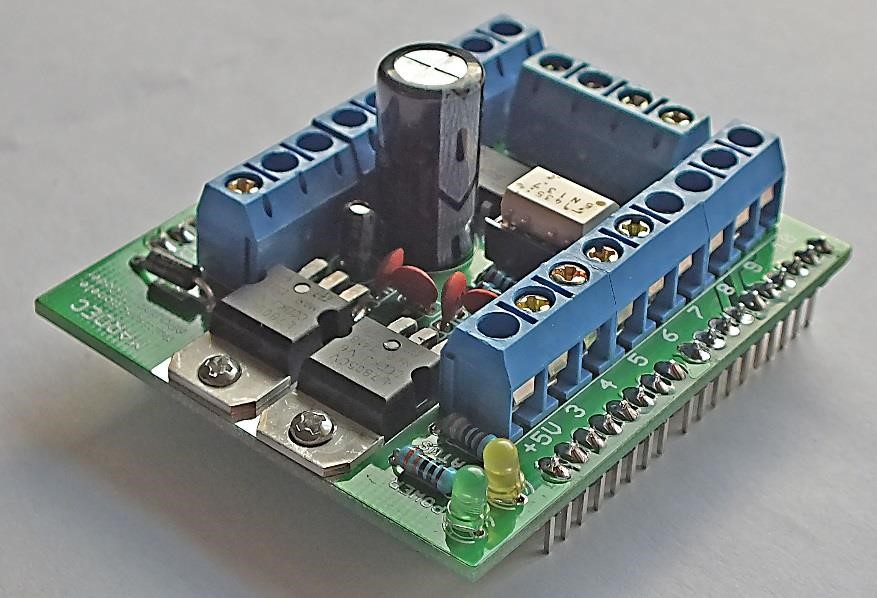
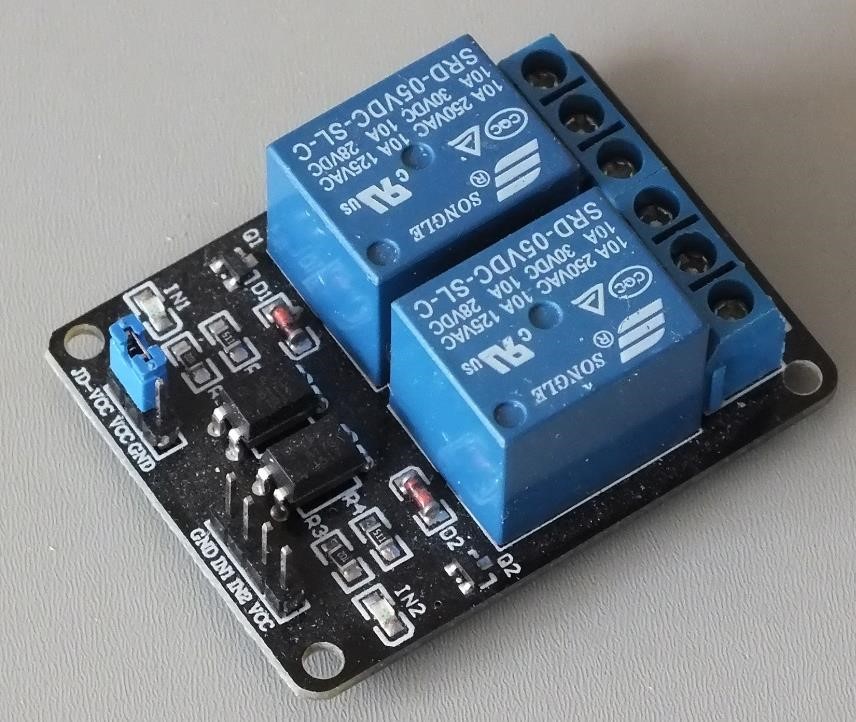
ur l'Arduino, nous voyons également un certain nombre de broches avec le texte Power. Ici nous trouvons deux broches de 5 volts (sortie), deux connexions GND et une broche Vin. Au lieu de la prise d'alimentation, le Vin-pin peut également être utilisé pour alimenter l'Arduino. Ne connectez pas 5V aux broches 5V.

ASTUCE:  
Commandez un Arduino comprenant un câble USB, généralement court.  
Lorsque vous placez sous la piste, laissez ce câble dans l'Arduino et placez l'autre extrémité sur le bord de la piste. Pour changer la configuration, connectez ce câble au PC en utilisant un câble un peu plus long. Vous n'avez pas à ramper sous la piste avant d'insérer le câble.

## DCC circuit

Deuxièmement, une petite quantité d'électronique est nécessaire pour que le signal DCC soit adapté à l'Arduino. L'IC 6N137 est un optocoupleur rapide avec lequel le signal DCC est transmis à l'Arduino.Les 5 volts requis peuvent être connectés directement à une broche 5V de l'Arduino.  
La broche 6 de l'IC est connectée à la broche 2 de l'Arduino.  
Vous devez souder vous-même ces composants supplémentaires sur une plaque de montage.

Une impression de conception (bouclier) a également été développée pour ce circuit. L'alimentation de l'Arduino peut également être placé sur cela. L'impression contient également une alimentation de 5V (maximum 1 ampère) qui peut être utilisée pour les relais, les LED et les servos, entre autres.



Cependant, une alimentation externe est recommandée pour les servos.

Cette impression peut être commandée avec le bon de commande sur:*Arcomora.wordpress.com/order.*

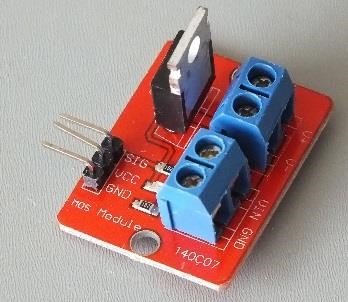
S'il n'est pas en stock, le délai de livraison peut être un nombre de semaines

Relais

Tous les relais peuvent en principe être utilisés pour la polarisation ponctuelle ou toute autre application.  
A cette fin, une broche Arduino peut commuter un relais de la manière habituelle au moyen d'un transistor.

Moins cher et plus facile sont les modules de relais prêts à l'emploi. Ceux-ci sont très appropriés pour être contrôlés directement à partir d'un Arduino. Ils contiennent deux circuits relais indépendants avec une LED d'indication et un optocoupleur et sont également disponibles avec 4, 6 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 ou les 5V du blindage Mardec.

Une version plus simple avec un relais et sans optocoupleur est celle-ci.  
Ils sont également disponibles dans une version 12 volts. Un avantage de ceci est un courant de bobine légèrement inférieur. Voir aussi page 24 pour plus d'informations.

**Commutateur MOSFET**

La commutation est également possible avec un amplificateur MOSFET.  
Ils conviennent entre autres pour des bobines alternatives et des bandes LED. Ils peuvent également être utilisés pour la gradation au moyen de la modulation de largeur d'impulsion (PWM). Vous pouvez utiliser un module prêt à l'emploi ou les souder vous-même sur un PCB de montage.  
Les types de MOSFET appropriés comprennent le FQP30N06L et le RFP30N06LE. Le «L» signifie «niveau logique». Cela signifie que ces MOSFETS sont complètement ouverts à une tension de contrôle de 5V et n'ont pratiquement aucune résistance.

## Logiciel

Bien sur un logiciel est également requis

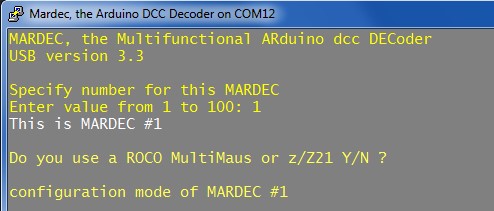
1. Pendant l'installation du logiciel, le programme MARDEC est placé comme un fichier binaire pré-compilé sur votre PC: MARDEC.hex.
2. Pour obtenir ce retour sur l'Arduino, vous devez télécharger un logiciel. Cela signifie que MARDEC est transféré de votre PC à l'Arduino via le câble USB. Ce téléchargement est effectué par un script qui commence par un raccourci sur le bureau.
3. Un émulateur de terminal séparé est utilisé pour la communication avec MARDEC. (Putty.exe) Ceci est également démarré avec un raccourci. Avec cela, la sortie du programme en cours d'exécution sur l'Arduino peut être affichée et l'entrée de votre clavier peut également être envoyée à MARDEC.

4) De plus, les pilotes ne sont requis. Si vous avez déjà installé l'IDE Arduino (Integrated Development

Environment), vous en avez probablement déjà un .Pour le clone chinois Arduino, vous avez besoin de pilotes spéciaux, ce que l'on appelle le driver CH340.

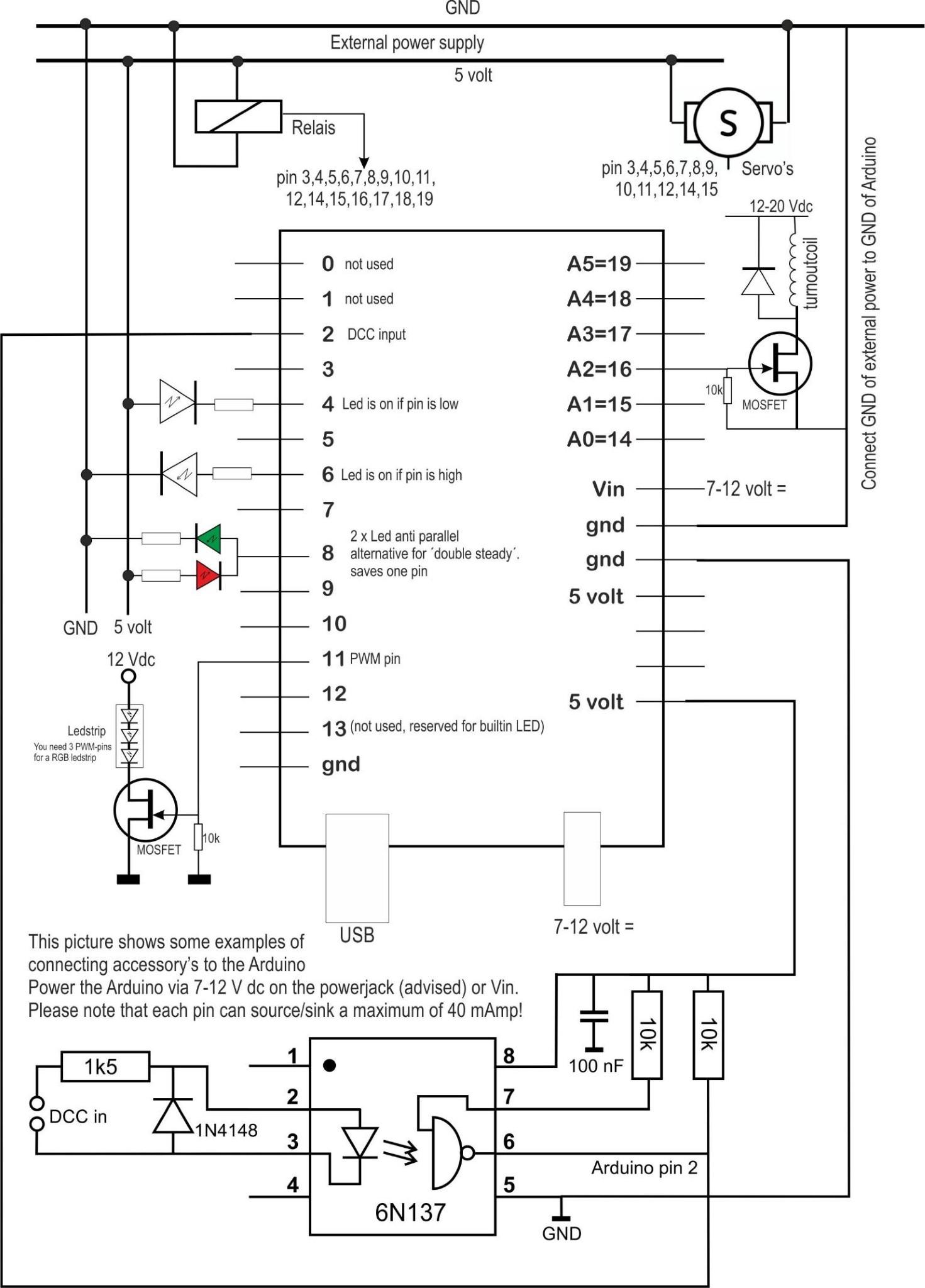
Tous ces quatre composants sont placés sur votre PC en une seule installation. Voir le manuel d'installation pour cela.

[DOWNLOAD HIER DE MARDEC SOFTWARE](http://www.globalvisuals.nl/downloads/Arcomora.zip)



*Openingsscherm van MARDEC*

# Diagramme de connexion MARDEC



*Figuur*

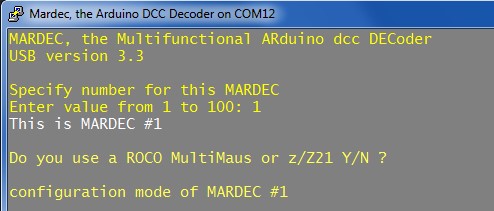
*5*

# Configuration du MARDEC

Si tous les logiciels sont correctement installés et que le MARDEC est connecté, la configuration peut commencer.  
Cliquez sur le raccourci 'Configurer MARDEC'.  
Ce n'est que si vous faites cela pour la toute première fois avec le premier décodeur que les pilotes USB seront chargés en premier, suivi du téléchargement du programme de décodeur sur l'Arduino. Après cela, la communication avec le programme de décodage sera toujours démarrée immédiatement. (Voir aussi le manuel d'installation)

ATTENTION :

Fermez chaque entrée numérique avec <enter>.  
• Toutes les commandes à un caractère NE DOIVENT PAS être terminées avec <enter>.  
• Pour la plupart des soumissions, la valeur existante n'est pas modifiée pour <enter> seulement. La valeur existante est ensuite placée entre parenthèses.  
• Dans l'état de configuration, une LED sera continuellement sur l'Arduino.  
• En état de fonctionnement, ce voyant est éteint.  
• Assurez-vous que NumLock est activé pour le pavé numérique.  
• Vous pouvez entrer les majuscules et les minuscules.  
• Utilisez uniquement la touche Retour arrière pour corriger

Une commande peu maintenant etre une entrée.

La toute première fois que vous démarrez le MARDEC, ou après une réinitialisation complète, vous devez d'abord entrer un numéro administratif pour ce MARDEC. Ce numéro identifie le décodeur..

On vous demandera aussi si vous voulez un Roco  
Multimaus, z21 ou Z21 utilisé. Dans ce cas  
le MARDEC implémentera automatiquement une correction d'adresse.  
ATTENTION: Désactivez la fonction Communication Railcom / Rail dans le z21!

Une vue d'ensemble de la configuration sera automatiquement affichée à chaque démarrage de la configuration.

# Les commandes

Il y a quatre types de commandes:

- Commandes générales,  
- Commandes asservies.  
Cela configure un servo.  
- Commandes liées aux accessoires  
Cela configure un accessoire  
- Commandes liées d'entrée  
Cela configure une entrée

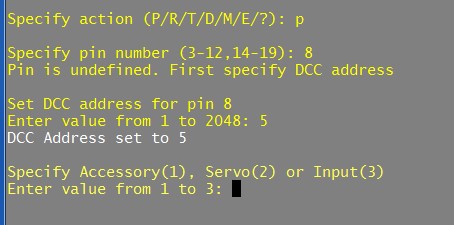
Le processus de configuration a donc quatre états différents.  
En entrant un? (point d'interrogation) un aperçu de la structure générale des commandes disponibles de MARDEC est donné.  
Vous ne voyez que les commandes associées à la situation actuelle.

Chaque commande consiste en une seule lettre et ne doit pas nécessairement être terminée avec <enter>.  
Les lettres sont basées sur la description en anglais de la fonction de chaque commande.  
Un servo, un accessoire ou une commande d'entrée ne peut être donné que si une broche est sélectionnée.  
Certaines commandes générales peuvent également être données dans l'état d'asservissement, d'accessoire ou d'entrée.  
Après avoir démarré dans l'état de configuration, vous ne pouvez entrer que des commandes générales.  
Une commande importante est la commande P. Ceci sélectionne une broche.

# Commandes générales

## P-Commande (Pin)

Un numéro de broche peut être sélectionné en entrant P.  
Si la broche n'a pas encore été utilisée, l'adresse DCC doit d'abord être entrée, puis on lui demande si elle doit être une entrée, un servo ou une broche accessoire.  
Vous pouvez ensuite entrer des commandes servo, des commandes accessoires ou des commandes d'entrée.

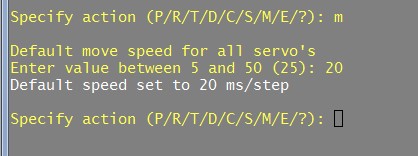
 Si la broche a déjà été configurée auparavant, le réglage de la broche sera affiché et l'état de la commande correspondante sera transmis directement.

Le servo / accessoire / entrée connecté à la broche correspondante peut alors être réglé. Vous ne pouvez utiliser que les broches 3 à 12 et 14 à 19 pour les accessoires et les entrées. Pour les servos seulement les pins 3 à 12 et 14 et 15. Ici les pins sur le  
Les impressions Arduino numérotées de A0 à A5 sont les numéros de broche 14 à 19.  
Cependant, un accessoire de type PWM (mode 7 et 8, voir commande M pour les accessoires) n'est disponible que sur les broches 3, 5, 6 et 11.

## T-Commande (Test)

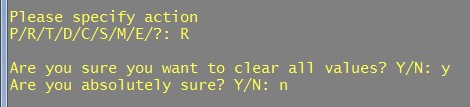
i vous avez réglé tous les servos / accessoires / entrées, vous pouvez tout tester avec la commande T. Tous les servos font ensuite plusieurs allers-retours et les accessoires s'allument et s'éteignent.  
La commande T peut également être entrée en tant que commande servo ou accessoire. Dans ce cas, seuls les servos ou accessoires respectifs sont testés. Le test des entrées se fait en connectant l'entrée correspondante à GND avec un câble Dupont.

## M-Commande (déplacer la vitesse)

 Avec la commande M, vous pouvez définir la vitesse de rotation standard de tous les servos. Ceci est réglable entre 5 et 50 millisecondes par degré. Il y a 5 ms. très rapide et 50 ms. très lentement. Le paramètre par défaut est 25 ms.  
La commande M peut également être saisie en tant que commande d'asservissement. La vitesse du servo peut être réglée séparément.

## D-Commande (document)

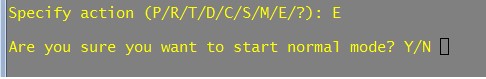
a commande D donne un aperçu de la façon dont toutes les broches sont définies.  
Faites une capture d'écran de ceci et enregistrez-le sur votre ordinateur et / ou imprimez-le sur votre imprimante.

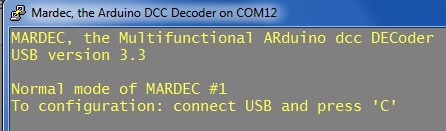
Vous savez toujours comment le décodeur servo est réglé. Vous pouvez également utiliser le fichier journal.

## R-Commande (Reset)

Avec la commande R tous les paramètres sont supprimés de la mémoire de l'Arduino.  
Vous serez invité à confirmer deux fois.  
Après une réinitialisation, l'écran est "vide".

## E-Commande (Exit)

L'état de fonctionnement sera démarré avec la commande E. Le câble USB n'est plus nécessaire.

 Pour cela, vous serez d'abord demandé confirmation. Après avoir entré 'Y', l'écran est 'vide' et l'état de fonctionnement commence.  
Vous pouvez maintenant changer les servos et les accessoires de commande avec votre unité centrale DCC.  
Notez que le voyant d'état sur l'Arduino est maintenant éteint. Ici vous pouvez voir que l'Arduino est en état de fonctionnement. A chaque adresse DCC configurée qui est reçue, la LED donnera une impulsion courte.

## Vous pouvez maintenant retirer le câble USB et fermer le panneau de contrôle. En état de fonctionnement, le décodeur signale toute activation d'un servo / accessoire. Ceci est utile pour tester la condition de fonctionnement. Normalement, vous n'avez plus le câble USB et vous ne voyez pas ces messages.

Retour à la configuration

Pour revenir au mode de configuration, reconnectez le câble USB. Ensuite, démarrez le programme de configuration en utilisant le raccourci sur le bureau. Cela va redémarrer l'Arduino, que vous pouvez voir en clignotant la LED d'état 3 fois brièvement, puis rester éteint. Maintenant, appuyez sur la touche «C» dans le programme de configuration. Par conséquent, MARDEC passe à l'état de configuration et la LED d'état s'allume en continu.

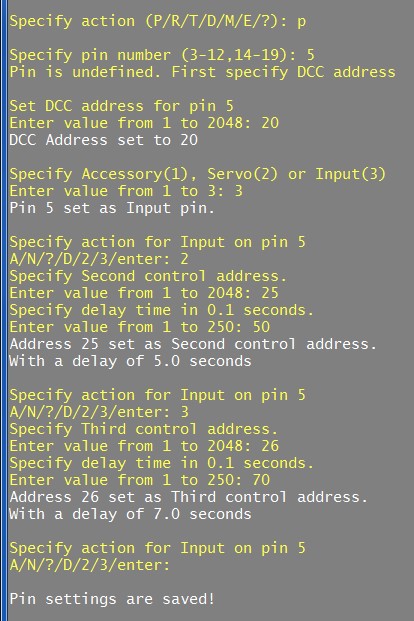
## IMPORTANT

Après chaque commande DCC reçue, la position modifiée d'un servo ou d'un accessoire est stockée directement dans la mémoire du MARDEC. Lorsque l'Arduino est allumé, ces états stockés sont à nouveau lus et les servos et accessoires sont réinitialisés tels qu'ils étaient lorsque l'Arduino a été arrêté.  
Pour un servo, cela signifie qu'il ne bouge plus entre l'arrêt et le redémarrage de l'Arduino!  
Il peut arriver qu'un servo bouge encore quelque peu à la suite de signaux d'interférence. Une bonne alimentation est donc indispensable ainsi que des servos numériques de bonne qualité (TG9d, HXT900, ES9051).  
En option, un noyau de ferrite près du servo peut également aider.  
Une résistance de 4k7 entre le signal et 5V peut également aider.

# Commandes d’entrée

Avec une broche configurée en entrée, un signal DCC peut être simulé pour le servo ou l'accessoire ayant la même adresse que la broche d'entrée.  
Exemple:  
Une seule DEL clignotante pour l'adresse DCC 34 est configurée sur la broche 12.  
Une entrée avec l'adresse DCC 34 est également configurée sur la broche 5.

Par défaut, Mardec maintient une broche d'entrée "haute" (5 V). Si cela est maintenant "bas", Mardec le verra comme un signal DCC pour l'adresse 34 et la LED clignotante sur la broche 12 s'éteindra ou commencera à clignoter.  
S'il y a aussi un servo à l'adresse 34, le servo commencera également à tourner.  
Rien ne se passe lorsque la broche d'entrée est à nouveau haute.  
Une broche d'entrée peut être réduite de diverses manières. Par exemple, en commutant un interrupteur reed à la terre ou un signal de détection de courant qui devient faible quand un train «passe». Une barrière lumineuse infrarouge peut également être utilisée.

Bien entendu, les attributs à l'adresse 34 peuvent également être commandés à partir d'un vrai signal DCC. Une différence importante, cependant, est que, avec une broche d'entrée, il ne se passe rien quand il redevient haut alors qu'il arrive avec un signal DCC. Gardez à l'esprit que si vous allumez un accessoire à partir de la centrale / de l'ordinateur et que vous l'éteignez à nouveau avec une broche d'entrée, l'ordinateur central / ordinateur «pense» que l'accessoire est allumé.  
La désactivation à l'aide de DCC n'a plus d'effet.  
Une entrée agit ainsi comme un interrupteur momentané et une commande DCC comme interrupteur marche / arrêt.

## 2- et 3- Commande (2e et 3e adresse)

Une deuxième et une troisième adresse DCC peuvent être liées à une broche d'entrée. Avec la commande '2' ou '3', cette adresse peut être entrée.  
Lorsque la broche d'entrée est activée, le servo / accessoire sera maintenant converti en son propre adresse (1) et la deuxième adresse (2) et éventuellement la troisième adresse (2).  
Pour cette deuxième et troisième adresse, un délai de 0 à 25 sec. être spécifié.

Le servo / accessoire sur cette 2ème et 3ème adresse ne sera commuté qu'après que le délai soit écoulé. Les deux temps sont comptés à partir du moment de l'activation de la broche d'entrée.

Dans la figure de droite, lorsque la broche d'asservissement 5 est activée, le servo / accessoire à l'adresse 20 (propre adresse) sera d'abord activé. Aucun retard n'est possible ici.  
Après 5 sec. l'adresse 25 (2ème adresse) est activée et après 7 sec. adresse 26 (3ème adresse).  
Toutes les adresses doivent bien sûr être configurées sur le même Arduino.  
Si vous voulez seulement des actions 'retardées', vous pouvez entrer une adresse fictive comme adresse 'propre' de la broche d'entrée. Vous pouvez également entrer l'adresse d'une autre entrée en tant que 2ème ou 3ème adresse! Cela vous permet de démarrer toute une série d'actions.

Aussi avec l'adresse DCC

Une entrée peut aussi etre activée avec une adresse DCC.

Cela se produit avec une transistion de 10 entre le signal DCC et la "propre adresse" de l'entrée..

Cependant, un servo / accessoire qui est à la même adresse que l'entrée répondra à la fois à la transition 10 et à la transition01 transition

Dans l'image ci-dessus, cela donne les résultats suivants:

Pour l'adresse DCC 20 de 10: Servo / accessoire à l'adresse 20 activé. Après 5 sec. adresse 25 et après 7 secondes adresse 26 Pour l'adresse DCC 20 de 01: Seul l'accessoire Servo / accessoire à l'adresse 20 est activé.

Entrée sur la broche 5 de 10: Servo / accessoire à l'adresse 20 activé. Après 5 sec. adresse 25 et après 7 secondes adresse 26. Entrée sur la broche 5 de 01: Rien ne se passe!

A-Command (Adresse, également pour les accessoires et les servos)

Avec la commande A, l'adresse DCC de l'entrée est définie. Vous recevrez un avertissement si l'adresse a déjà été attribuée à un autre servo, accessoire ou entrée. En acceptant que vous pouvez contrôler plusieurs servos / accessoires simultanément avec une adresse.  
En entrant l'adresse 0 (zéro), une broche définie précédemment est à nouveau libérée et la broche est considérée comme "non réglée".

N-Command (numéro, également pour les accessoires et les servos)

Avec la commande N, vous pouvez attribuer un numéro administratif à la broche d'entrée.  
Il n'a aucune signification technique et aucune contribution n'est requise.  
Vous recevrez un avertissement si le numéro a déjà été attribué à une autre entrée / servo / accessoire.

D-commande (Document)

La commande D affiche un aperçu de tous les paramètres.

## Test

Il n'y a pas de fonction de test séparée pour les broches d'entrée. Cependant, vous pouvez les tester en passant en mode de fonctionnement avec la commande Quitter. En connectant ensuite la broche d'entrée à une broche GND avec un câble Dupont, vous pouvez vérifier si le servo / accessoire lié répond correctement.

# Commandes servos

Une boîte de dialogue pour configurer un 'nouveau' servo ressemble à ceci:

Sur une broche inutilisée, le servo sera réglé sur des angles de 75 ("Low") et 105 ("High"). Un servo réglé précédemment obtient naturellement les angles déjà définis.  
Dans les deux cas, le servo est en position "Low".

## - (min) Commande

La commande '-' abaisse l'angle d'un degré.  
L'angle «haut» doit être supérieur d'au moins 5 degrés à l'angle «bas».

## + (plus) Commande

La commande '+' augmente l'angle d'un degré.  
L'angle «bas» doit être inférieur d'au moins 5 degrés à l'angle «haut».

## 9-Commande

Avec la commande 9, le servo est réglé à 90 degrés. Ceci est pratique pour mettre le bras rotatif du servo au bon endroit sur le servo.

## C-Commande (Change)

La commande C change entre les deux coins.  
Maintenant, l'autre angle peut être ajusté en utilisant les commandes plus et moins.  
Si les deux modes sont ajustés, vous pouvez le faire plusieurs fois  
Commande C pour entrer dans le test d'asservissement.  
Ceci est également possible avec la commande T.

## I-Commande (Inverse)

Cela dépend de la façon dont le servo est monté sous la voie ou le branchement, par exemple en tournant dans le plus petit angle de la ligne droite ou dans la position de déviation.  
Cependant, vous voulez que le commutateur corresponde à la position sur la station de commande DCC et / ou le programme d'ordinateur.  
S'il s'avère que ce n'est pas la même chose, vous pouvez utiliser la commande I pour faire fonctionner le relais dans l'autre sens. Vous pouvez seulement entrer 'Y' ou 'N' où 'N' est la valeur par défaut.

## T-Commande (Test)

Cela teste le mouvement du servo. Le servo va et vient 3 fois.  
La commande T peut également être entrée en tant que commande servo ou accessoire. Dans ce cas, seuls les servos ou accessoires respectifs sont testés.

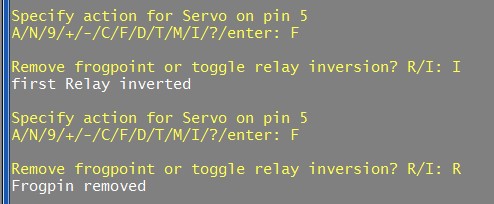
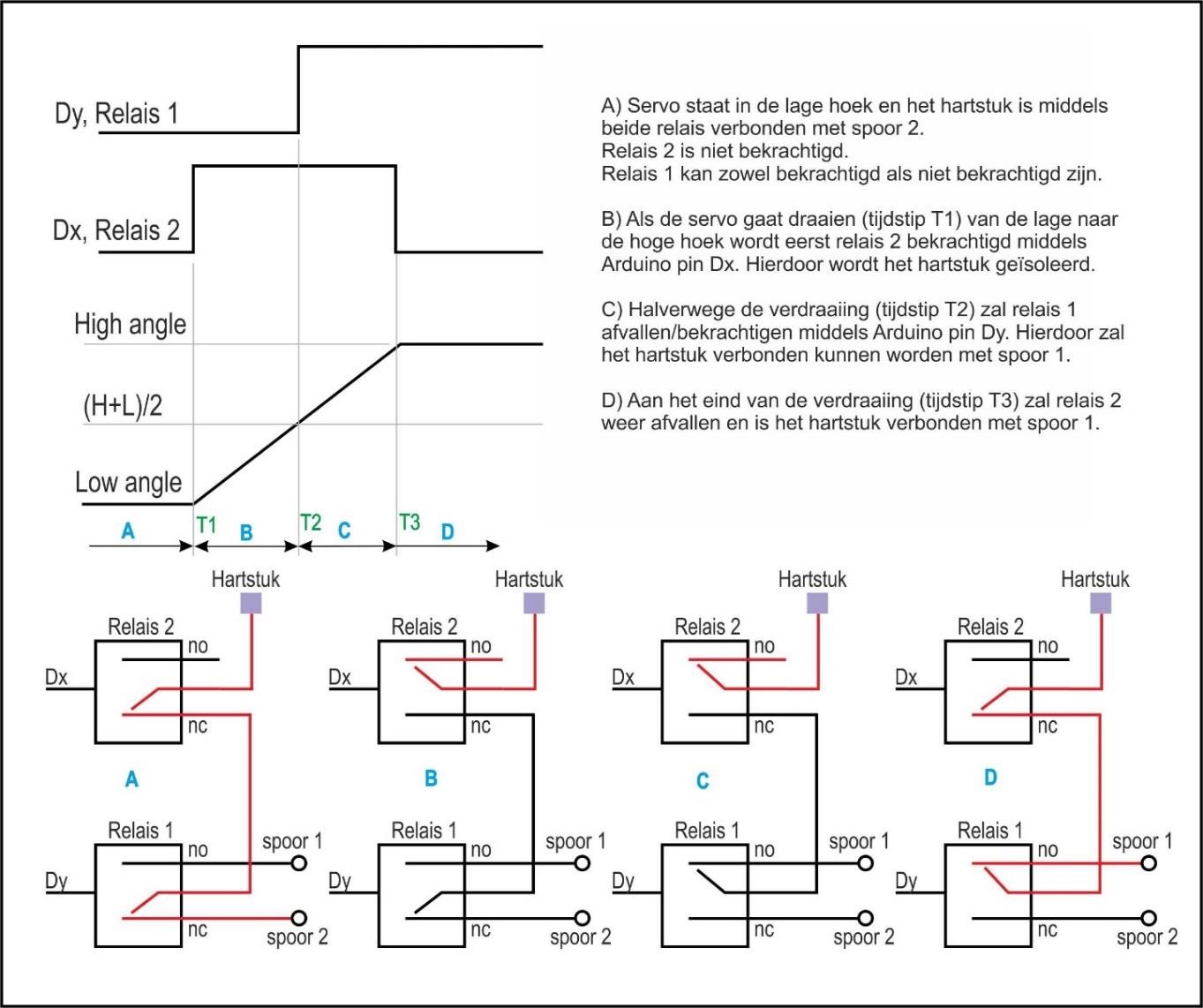
.

## M-Commande (déplace la vitesse)

## Avec la commande M, vous pouvez définir la vitesse de rotation individuelle du servo sélectionné. Ceci est réglable entre 5 et 50 millisecondes par degré. Il y a 5 ms. très rapide et 50 ms. très lentement. Le paramètre par défaut est 25 ms. Cela peut également être défini avec la commande M en tant que commande générale.

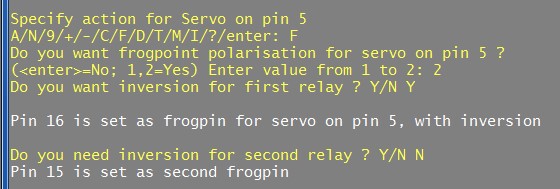
## F-Commando (Frog point = puntstuk)

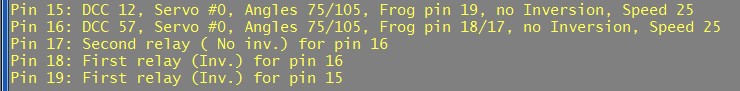
 Avec la commande F, vous pouvez déterminer automatiquement un numéro de broche sur lequel vous pouvez connecter un relais pour la polarisation ponctuelle (point de grenouille). Si vous voulez un relais, vous entrez un '1'. Le MARDEC va maintenant choisir une broche gratuite et commencer à chercher de la broche 19 à la broche 3

 Avec la commande F, Mardec attribue automatiquement une polarisation de point de grenouille. Pour un seul relais, sélectionnez '1'.  
Si un point de grenouille est déjà attribué, Mardec vous demandera d'enlever le point de grenouille.  
Le relais commutera lorsque le servo sera à mi-course. Cela assure un changement de polarité en douceur.  
En tournant vers l'angle haut, le relais sera activé et désactivé lors de la rotation à l'angle bas. Au cas où cela ne serait pas correct, vous pouvez régler ce relais sur 'Inversé'; il va basculer puis inversement.  
Pour encore plus de sécurité, vous pouvez utiliser 2 relais. Les deux relais fonctionnent ensemble comme suit:

Certains relais sont 'actifs-bas' (activés avec 0 Volt au lieu de 5 V) Mardec vous demande 'Inversion'.  
Cela garantit que le relais n'est activé que lorsque le servo tourne.

our deux relais, vous entrez un '2'. Parce que certains relais sont 'actifs bas' (étant alimentés par 0 volt au lieu de 5 volts), il vous sera demandé si vous voulez 'inversion' pour ce relais. Entrez "Y" pour ce type de relais. Le module relais tel que décrit à la page 7 est un tel relais «actif bas». Voir aussi page 27.

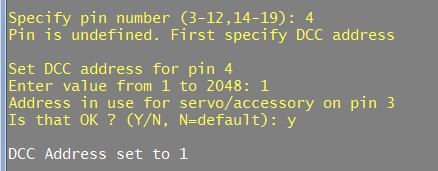
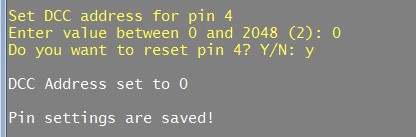
STUCE: Bien sûr, vous pouvez également connecter autre chose qu'un relais sur la broche de grenouille; par exemple actionnez un point de vue ou une LED d'indication qui indique la position du point. Vous pouvez également utiliser le relais pour changer légèrement différemment de la pièce ponctuelle. Dans l'ensemble, la broche de grenouille est simplement une broche qui devient haute ou basse en même temps que la conversion de l'interrupteur.



Pour changer l'inversion du relais 2, le point de grenouille doit d'abord être retiré!

A-Command (Adresse, également pour les accessoires et les entrées)

Avec la commande A, l'adresse DCC du servo est définie. Vous recevrez un avertissement si l'adresse a déjà été attribuée à un autre servo, accessoire ou entrée. En acceptant que vous pouvez contrôler plusieurs servos / accessoires simultanément avec une adresse.



En entrant l'adresse 0 (zéro), une broche définie précédemment est à nouveau libérée et la broche est considérée comme "non réglée".

N-Command (numéro, également pour les accessoires et les entrées.)

Dans la conception de votre piste, vous avez probablement donné un numéro aux factures et accessoires. Vous pouvez également affecter ce numéro administratif à la broche avec la commande N.  
Il n'a aucune signification technique et aucune contribution n'est requise.  
Vous recevrez un avertissement si le numéro a déjà été attribué à un autre servo / accessoire.

D-commande (Document)

La commande D affiche un aperçu de tous les paramètres.

## <enter>-Commande

En entrant <enter> les paramètres du servo sont stockés et les commandes générales sont renvoyées. Une broche différente peut ensuite être sélectionnée.

Commandes d'accessoires

Les données du signal DCC pour un accessoire n'ont que deux valeurs (0/1, marche / arrêt, haute / basse, droite / déviée). Le MARDEC peut réaliser différentes fonctions avec ces données accessoires.  
Les fonctions 'doubles' listées ci-dessous nécessitent l'utilisation d'une seconde broche. C'est la broche "mon pote". Ceci est automatiquement affecté aux doubles fonctions.

Pour une broche non définie précédemment

d'abord la fonction désirée (

=

’mode’)

est demandée en premier

:

1.

Simple stade

.

La broche Arduino suit le signal DCC

2.

Double Stade

.

La broche Arduino suit le signal DCC

et une seconde broche arduino reçoit la valeur

inverse

3. Clignotement unique. La broche Arduino est alternativement haute et basse, tant que les données DCC sont élevées. Les temps d'activation et de désactivation peuvent être réglés séparément entre 50 et 50000 ms

.

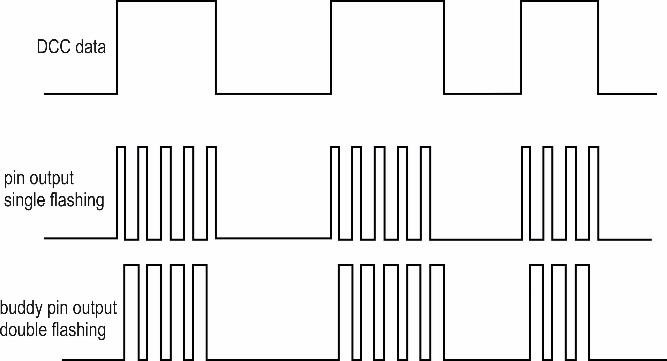
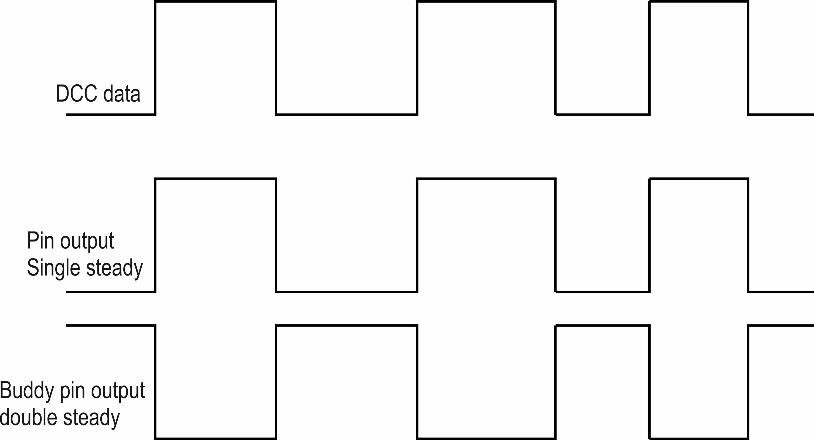
.

-

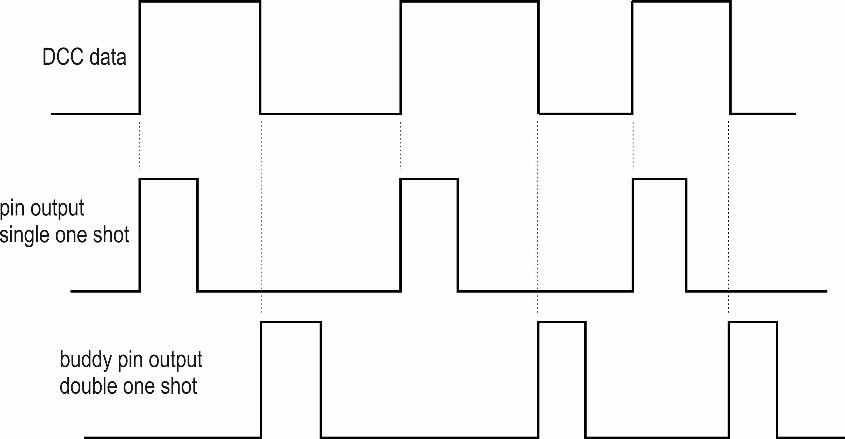
. 4. Double clignotant. La même chose que le clignotement simple, mais une broche buddy obtient la valeur inverse.

.

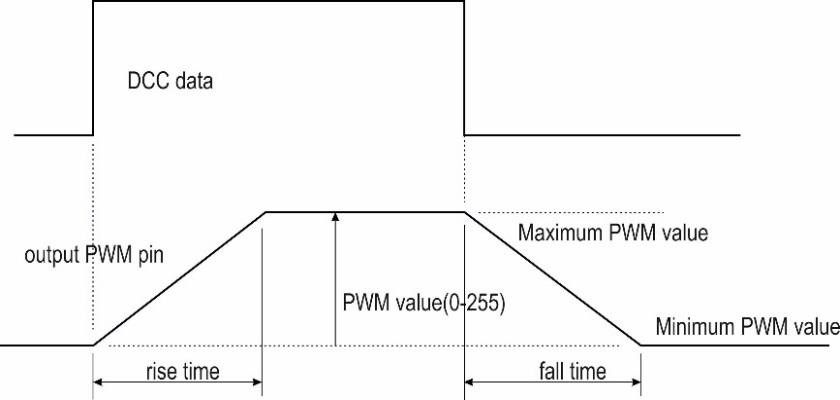
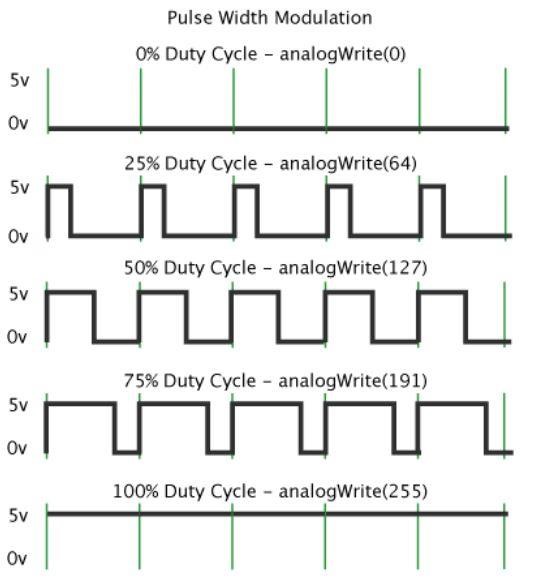
.



Assurez-vous de connecter correctement les voyants. Pour un signal AHOB (LED éteinte comme clignotant), vous connectez l'anode avec l'Arduino et pour un signal WUBO (LED allumée comme clignotant), vous connectez la cathode à l'Arduino. Voir également les exemples de connexions des broches 4 et 6 dans le schéma de connexion à la page 9.

1. Single One Shot. La broche Arduino devient élevée pendant une certaine période (50-5000 ms) lorsque les données DCC deviennent élevées.  
   Si l'adresse d'un seul coup est la même que celle d'un autre servo ou accessoire, à la fin du pouls, il sera  
   'Converted' comme si un signal DCC avait été envoyé. Exemple: A l'adresse 12, un seul coup et un double clignotement sont configurés. Si MARDEC voit maintenant l'adresse 12, le coup unique et le double clignotement commenceront. À la fin de l'impulsion, le double clignotement sera à nouveau désactivé.
2.  . Cela signifie qu'un accessoire peut être activé pendant une courte période (5 secondes maximum) sans l'intervention d'un signal DCC.

7. Double coup.  
La même chose que le tir unique.  
Une broche de copain, cependant, va haut pendant la même période lorsque les données DCC devient faible. Avec cette fonction, vous pouvez également contrôler les bobines alternées.  
Les bobines interchangeables nécessitent au moins 12 volts et consomment un courant relativement important. C'est pourquoi vous ne pouvez pas les connecter directement à une broche Arduino. Cela nécessite un étage amplificateur supplémentaire.  
Voir l'exemple dans le diagramme de la page 9 (broche 16). Pour un changement, vous avez donc besoin de deux étages d'amplification. Un pour chaque bobine. Les modules MOSFET mentionnés à la page 7 conviennent pour cela.

1. Analogique, contrôle PWM..

Sur un certain nombre de broches (3.5,6 et 11), un Arduino peut définir une tension de bloc dont la largeur d'impulsion (cycle de service) peut varier de 0 à 100%. Ceci est réalisé en 'écrivant' une valeur de 0 à 255 à cette broche. Par exemple, en connectant une LED et en variant la valeur PWM, cette LED peut être grisée. Cela peut être utilisé pour, par exemple, un éclairage de travail variable.

Principe de la modulation de largeur d'impulsion

Ce qui suit peut être réglé pour cette fonction:  
- la valeur PWM maximale (valeur min - 255)  
- la valeur PWM minimale (0 - valeur maximale)

Ce qui suit peut être réglé pour cette fonction:  
- la valeur PWM maximale (valeur min - 255)  
- la valeur PWM minimale (0 - valeur maximale)  
- le temps de montée  
- le temps de descente Les temps peuvent être réglés indépendamment de 0 à 500 secondes par pas de 2 secondes. Avec trois de ces accessoires, vous pouvez, par exemple, contrôler une bande de LED RGB

Une bande de LED nécessite également un étage amplificateur sous la forme d'un transistor MOSFET. Voir le schéma de câblage à la page 9 (broche 11).  
   
Vous pouvez trouver encore plus de fonctionnalités avec PWM dans ce fil de Ruud Boer.

1. Mode lumière scintillante. Cela vous permet de scintiller une LED connectée avec une intensité lumineuse minimum et maximum réglable. En scintillant trois LED de couleurs différentes, vous pouvez simuler un incendie. Vous pouvez également l'utiliser par exemple. simulation de lumière de soudage. Cela utilise également PWM et ne peut donc être utilisé que sur les broches 3.5.6 et 11.

9 . Aléatoire activé / désactivé. Avec cela, une LED connectée est allumée et éteinte en continu tant que les données DCC sont élevées. Une heure est déterminée au hasard pour chaque période d'activation et de désactivation. Le temps minimum et maximum peut être réglé entre 2 et 500 secondes par pas de 2 secondes.  
Cela peut être utilisé, par exemple, pour allumer et éteindre l'éclairage intérieur des maisons ou l'éclairage public.

1. Lumière disco. Avec cela, une LED connectée est allumée et éteinte en continu tant que les données DCC sont élevées. Une heure est déterminée au hasard pour chaque période d'activation et de désactivation. Les temps minimum et maximum peuvent être réglés entre 2 et 500 millisecondes.  
   Cette fonction est en fait la même que la fonction 9; ça ne fait que 1000 fois plus vite

10.

10. Après le choix de la fonction désirée, une broche de contact sera attribuée et on demandera des heures d'activation et de désactivation ou des réglages PWM. Cela dépend de la fonction choisie.

Les paramètres sont affichés pour une broche qui a déjà été configurée en tant qu'accessoire. Ceux-ci peuvent être ajustés avec les commandes suivantes.

M

-

C

ommande

(

M

ode)

-

Avec la commande M, une fonction (differente)

te) peut être utilisée avec la commande M

peut etre selectionnée pour la broche

-

sélectionnée . La commande M à sa propre fonction d’aide :

-

:

H

-

Commande

Haute

)

(

Aves les fonctions 3.4et6 l’heure d’activation peut

être définie. Cela peu varier de 100 à 500ms par

pas de 20ms

Avec la fonction 7 P(PWM),la valeur PWM la plus

élevée peut etre définie (min255)

Avec la fonction 9 ; le temps maximum peut etre

Réglé (min 500 sec par pas de 2sec)

Avec la fonction 10, la période maximale peut être utilisée

-

L

-

Commande

(

Bas

)

Avec les fonctions 3 et 4, le temps d'arrêt peut être défini.  
Cela peut varier de 100 à 5000 ms. par pas de 20 ms.  
Avec la fonction 7 (PWM), la valeur PWM la plus basse peut être réglée. (0-255)  
Avec la fonction 9, le temps minimum peut être réglé (2 secondes maximum par pas de 2 secondes)  
Avec la fonction 10, la période minimale peut être utilisée être réglé (2-max msec)

être réglé (2-max msec)

-

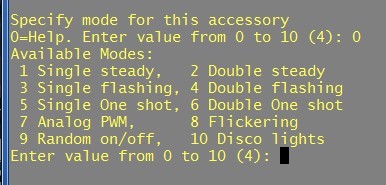
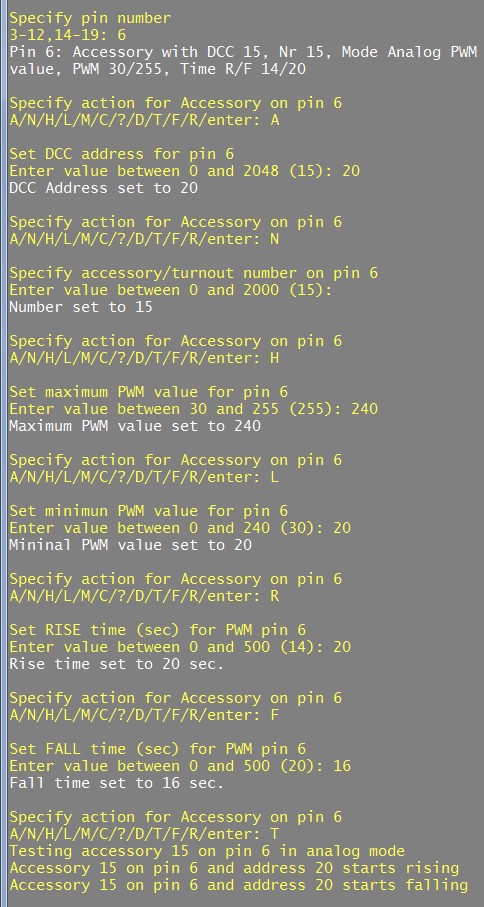
R

-

Commande Rise (Augmenter)

Uniquement pour la fonction 7. Cela permet de régler le temps de montée.  
C'est le moment où la valeur PWM de la valeur la plus basse est la plus élevée.

.



Configuration du contrôle PWM analogique

## F-Commande Fall (Chutte)

Uniquement pour la fonction 7. Cela permet de régler l'heure de chute C'est le moment où la valeur PWM de la valeur maximale revient à la valeur la plus basse.

## Autres Commandes

Les commandes générales suivantes sont également disponibles en tant que commande servo.

T-commande: test de la fonction accessoire individuelle. (en utilisant la fonction 9)

N-commande: définir le numéro administratif.  
 A-commande: régler l'adresse DCC ou la broche 'release' (adresse = 0).

D-commande: vue d'ensemble de tous les paramètres.

# Autres questions

## Démarrer

Lors du démarrage de l'Arduino, la LED d'état clignote brièvement trois fois. Vous pouvez voir que l'Arduino est démarré. L'Arduino est également redémarré lorsque vous démarrez Putty dans l'état de configuration. À partir de MARDEC 3.0, il n'y a pas de redémarrage lors du changement entre la configuration et l'état de fonctionnement.

## Quitter

## Le panneau de contrôle (Putty) peut être fermé à tout moment. Aucune commande séparée n'est disponible pour cela. Alors utilisez Alt-F4 ou cliquez sur la croix familière dans le coin supérieur droit de la fenêtre. Assurez-vous que vous êtes dans l'état de commande générale, de sorte que les dernières modifications pour une épingle sont enregistrées.

## Journalisation

La boîte de dialogue de configuration entière est enregistrée dans le fichier:  
Mes documents / Mardec / MARDEC\_ <date> \_ <heure> .log.  
De plus, <date> \_ <heure> est l'heure de fermeture.

## Document

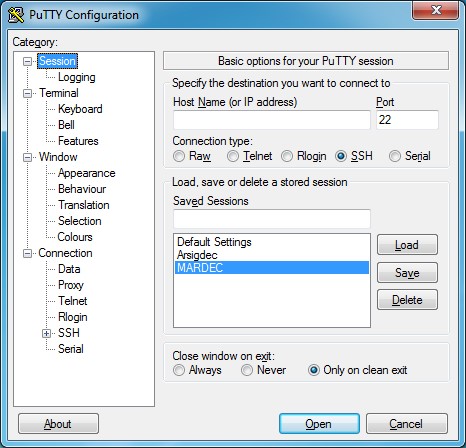
ne fois la configuration terminée, il est judicieux d'enregistrer les paramètres dans un document. Pour ce faire, il existe le raccourci Documentez votre MARDEC dans le dossier MARDEC du menu Démarrer de Windows. Cela ouvre un document MS Word que vous pouvez remplir. (Donc, nécessite MS Word sur votre PC).

## Configurer les décodeurs suivants

Une fois le logiciel installé, le programme de décodage est automatiquement chargé sur l'Arduino lors du premier démarrage de «Configuration MARDEC».

Il existe une autre méthode pour les décodeurs suivants.  
Pour ce faire, démarrez le raccourci 'Ajouter un nouvel Arduino'.  
Vous pouvez les trouver dans le menu Démarrer de Windows dans le dossier Arcomora

Écran de configuration de la configuration (Putty)

 Vous pouvez ajuster l'affichage du panneau de commande comme suit.

1. Cliquez sur le raccourci Configure Putty. Vous pouvez les trouver dans le menu Démarrer de Windows dans le dossier  
   Arcomora  
   L'écran adjacent apparaît

2. Cliquez sur MARDEC puis sur ‘Open.’

1. A la session *l*a journalisation vous permet d'ajuster les options du journal. Note: Si vous changez le nom de fichier, les sessions de configuration ne peuvent pas être sauvegardées!
2. A la fenetre Apparence / Comportement / Couleurs, vous pouvez également ajuster l'apparence.
3. Si necessaire vous pouvez également changer le port COM ici.
4. Ne faites rien à toutes les autres parties
5. Sélectionnez Session et cliquez sur Enregistrer pour enregistrer à nouveau les paramètres..
6. Cliquez sur Ouvrir pour ouvrir à nouveau le panneau de configuration

***Vous pouvez également facilement modifier le port COM d'un Arduino avec le programme de port Change COM.  
Vous pouvez le trouver dans le menu Démarrer de Windows, dans le dossier Arcomora.***

## Arduino Pro Mini

Une alternative à l'utilisation de l'Arduino Uno est **l'Arduino Pro Mini.**

Il est 100% compatible avec l'Uno et a donc les mêmes spécifications pour la mémoire interne et les connexions à broches.

Bien sûr, il y a des différences:

* Le Pro Mini est beaucoup plus petit
* Il n’y a pas de port USB. C’est pouruoi un convertisseur [speciale USB](http://nl.aliexpress.com/item/CH340-module-USB-to-TTL-CH340G-upgrade-download-a-small-wire-brush-plate-STC-microcontroller-board/32354359382.html?isOrig=true&isOrigTitle=true)-TTL est nécéssaire pour charger les programmes sur un Pro Mini. Bien sur vous n’en avez besoin que d’un seul.
* Vous devez d’abord souder vous-même les broches de connexion (fournies) au Pro Mini

Ce sont des connecteurs males mais vous pouvez connecter des connecteurs femelles.

* Parce que les scripts de téléchargement ne conviennent que pour l'UNO, vous devez télécharger vous-même MARDEC en utilisant Xloader
* Ajuster le chargeur de démarrage

Par défaut, le Pro-mini a un bootloader différent et plus grand que l'UNO. Par conséquent, la taille maximale du programme est seulement de 30270 octets.  
C'est trop peu pour le MARDEC; il a besoin de 32176 octets.  
Cela peut être résolu en définissant le même chargeur de démarrage sur le Pro Mini que sur l'UNO.  
C'est ce qu'on appelle le bootloader Optiboot.  
Vous pouvez également utiliser cette méthode pour fournir un UNO réticent avec un nouveau bootloader!

De quoi avez-vous besoin:

* + Un Arduino UNO qui fonctionne très bien
  + Un Arduino Pro-Mini Qui fonctionne bien et don’t vous voulez ajustez la charge de démarrage.

\* 6 cablesDupont M-V

Faites comme suit

* + Connectez l’UNO au Pc aves le cable USB

\* Connectez l’UNO avec le PRO-Mini comme suit:

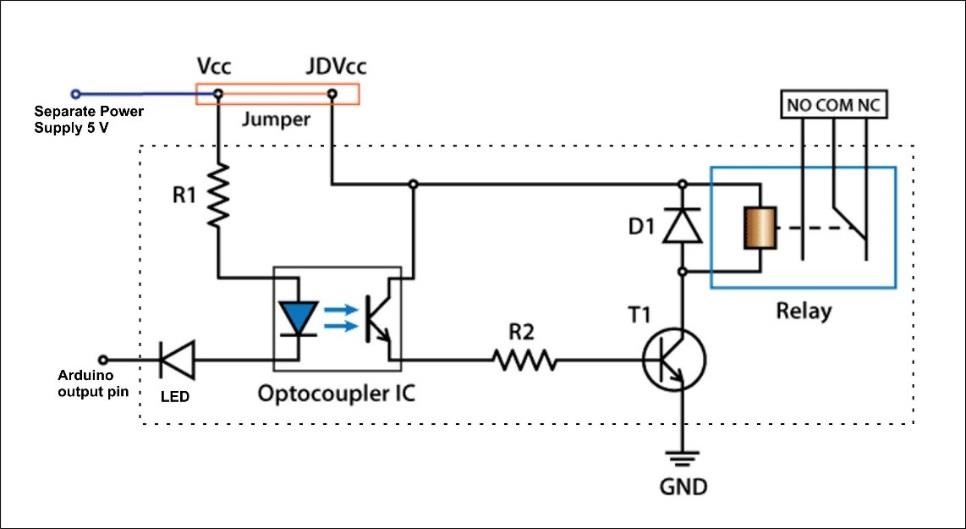
* + Broche 10 de l'UNO à la broche RST (réinitialisation) du Pro-Mini
  + Broche 11, 12 et 13 de l'UNO avec les broches 11,12 et 13 du Pro-Mini - 5V et GND de UNO avec les Vcc et GND du Pro Mini.  
           (pour un UNO connecter les deux broches 5V) de Pro Mini.
  + Définir dans ‘l’outils’
  + Le bon port COM
  + Tableau : Arduino/Geenuino Uno
  + Programmeur: Arduino as ISP
  + Ouvrez l’esquisse ‘Arduino ISP’ à partir de fichiers-> exemples \* Téléchargez-le sur Arduino UNO.
  + Démarrer l'option 'Bootloader' depuis 'Tools'

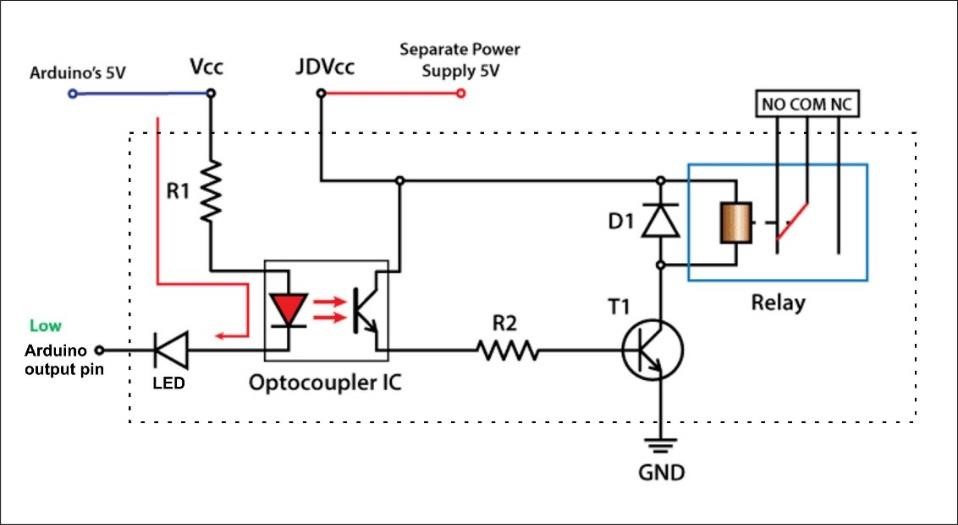
e pro-mini est maintenant équipé du bootloader optiboot. Cela ne prend que peu de temps.  
Vous pouvez maintenant connecter à nouveau le Pro-Mini, via l'interface USB spéciale.  
Maintenant, ouvrez MARDEC et essayez de le télécharger. Vous verrez que vous avez maintenant 32 256 octets disponibles.

NOTE: Le type de carte DOIT maintenant rester Arduino / Genuino UNO!

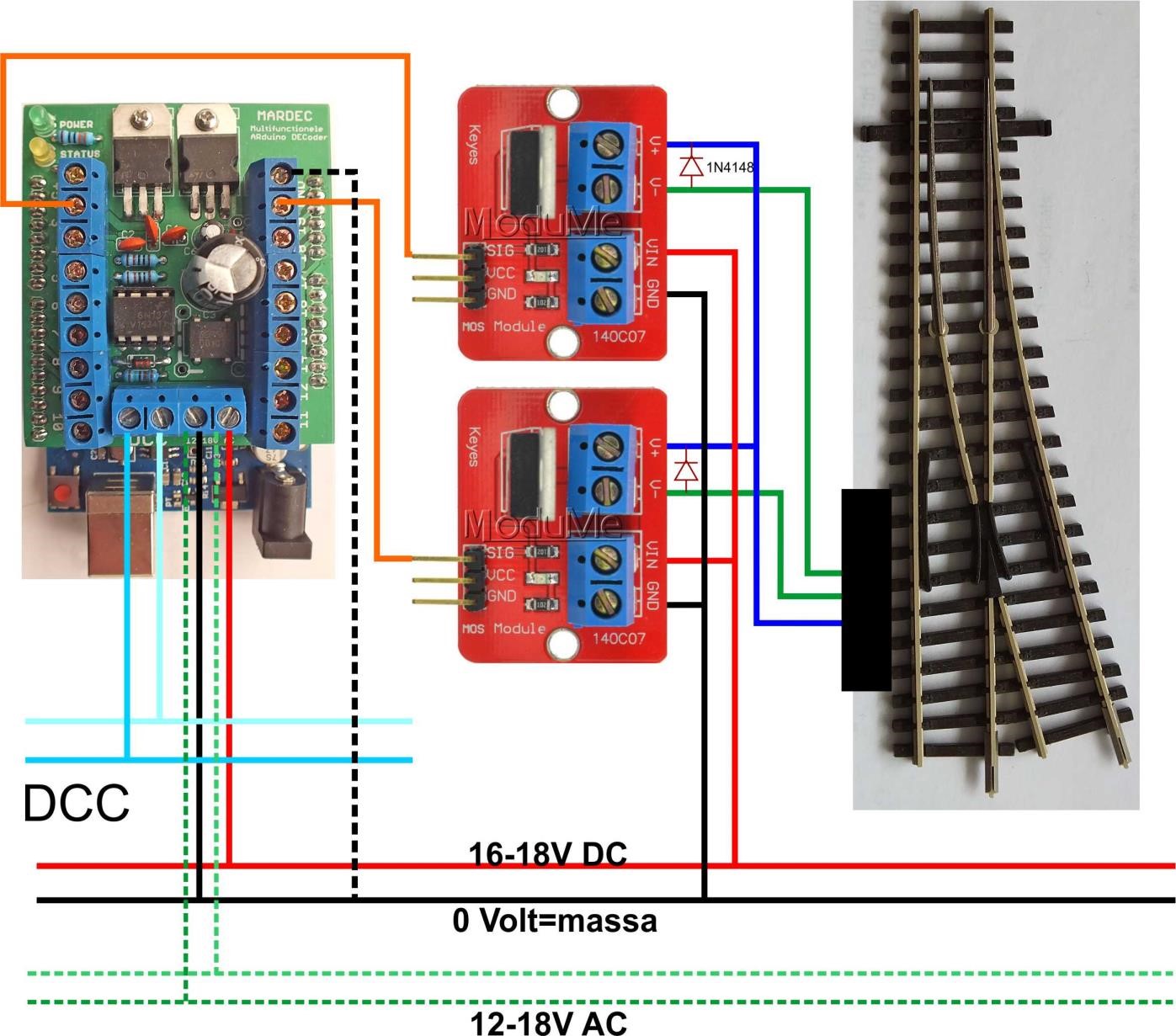
Si vous voulez en savoir plus sur le bootloader, veuillez vérifier ici: <https://www.baldengineer.com/arduino>[-bootloader.html](https://www.baldengineer.com/arduino-bootloader.html)

## Relais modules

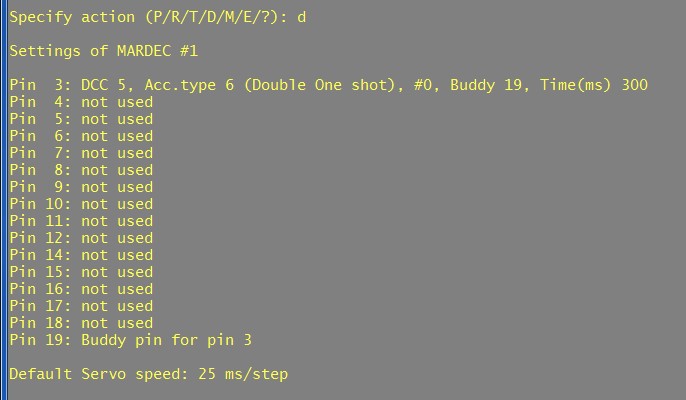
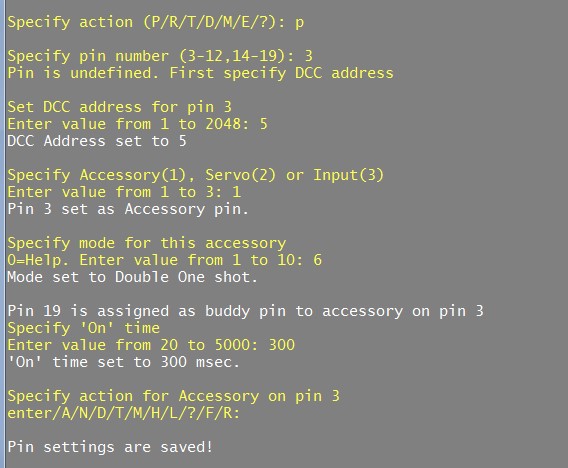
Si vous utilisez les modules de relais avec optocoupleur comme mentionné à la page 7, il peut être important de savoir qu'ils sont très bas.  
Cela signifie que le relais est sous tension lorsque la tension sur la broche Arduino est de 0 volt.  
   
Il y a un cavalier standard sur les broches Vcc et JDVcc. Vous avez maintenant deux options:  
   
1) Laissez le cavalier et connectez Vcc / JDVcc à l'alimentation externe

1. Retirez le cavalier et:  
   - Connectez Vcc à une sortie Arduino 5V
2. - Connectez JDVcc à l'alimentation externe

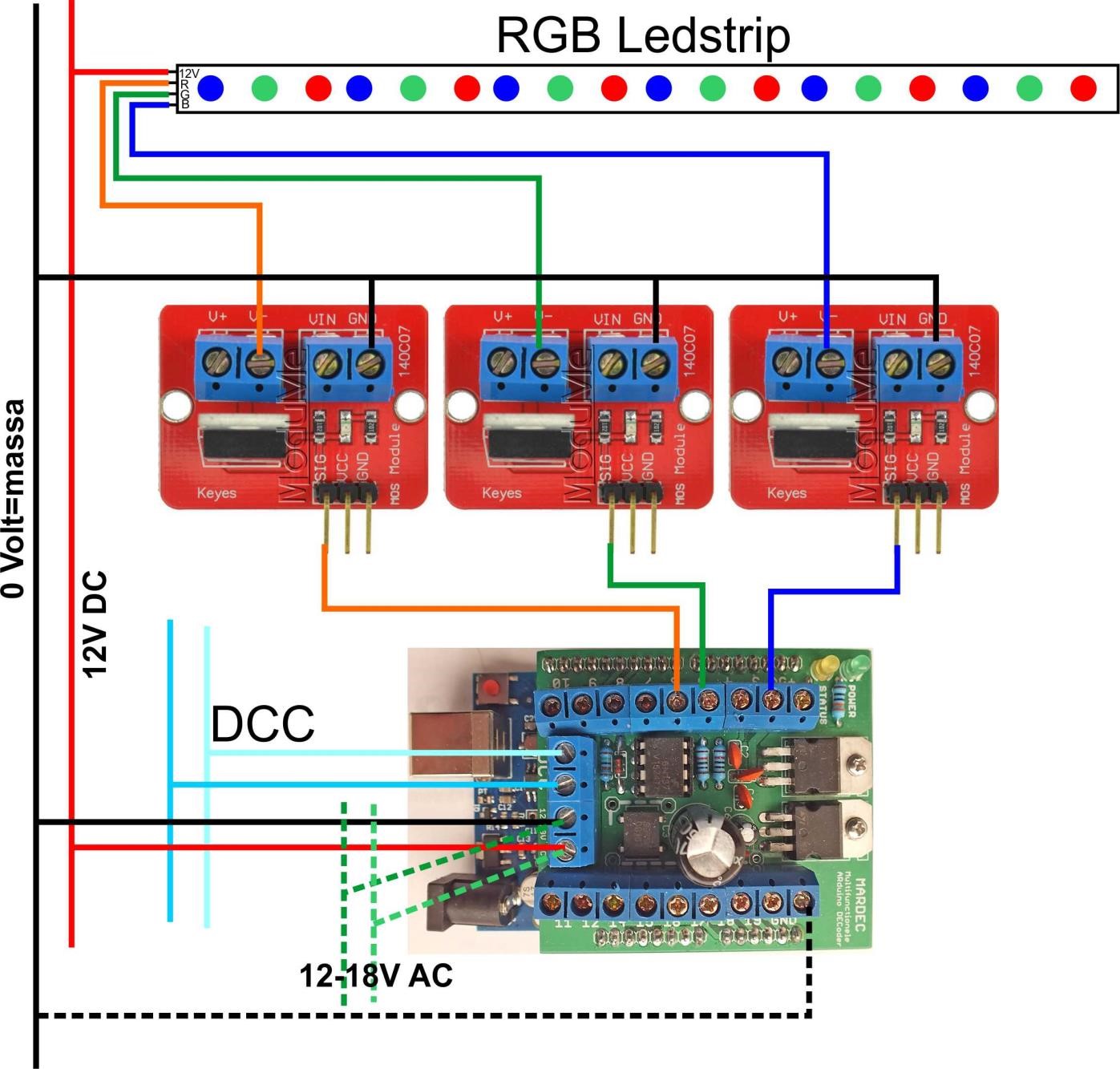
Sur cette image, vous voyez également que le relais est activé lorsque la broche Arduino est BASSE.  
   
Dans tous les cas, vous ne devez PAS laisser le cavalier et connecter Vcc / JDVcc à l'Arduino. Dans ce cas, vous tirez également le courant de la bobine de l'Arduino et ce n'est pas sage.

Connexion de l'amplificateur MOSFET pour les changements de bobine

Ici, l'Arduino est alimenté par la même tension que les commutateurs. L'Arduino peut également être alimenté en courant alternatif.  
Connectez la connexion GND à la boucle de masse.  
Remarque: Les broches Arduino utilisées dépendent de la configuration réelle. Les connexions montrées ici sont juste un exemple.  
   
Configuration:

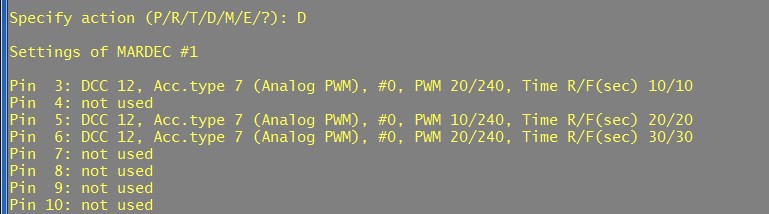


Connexion de la bande LED RGB avec des amplificateurs MOSFET

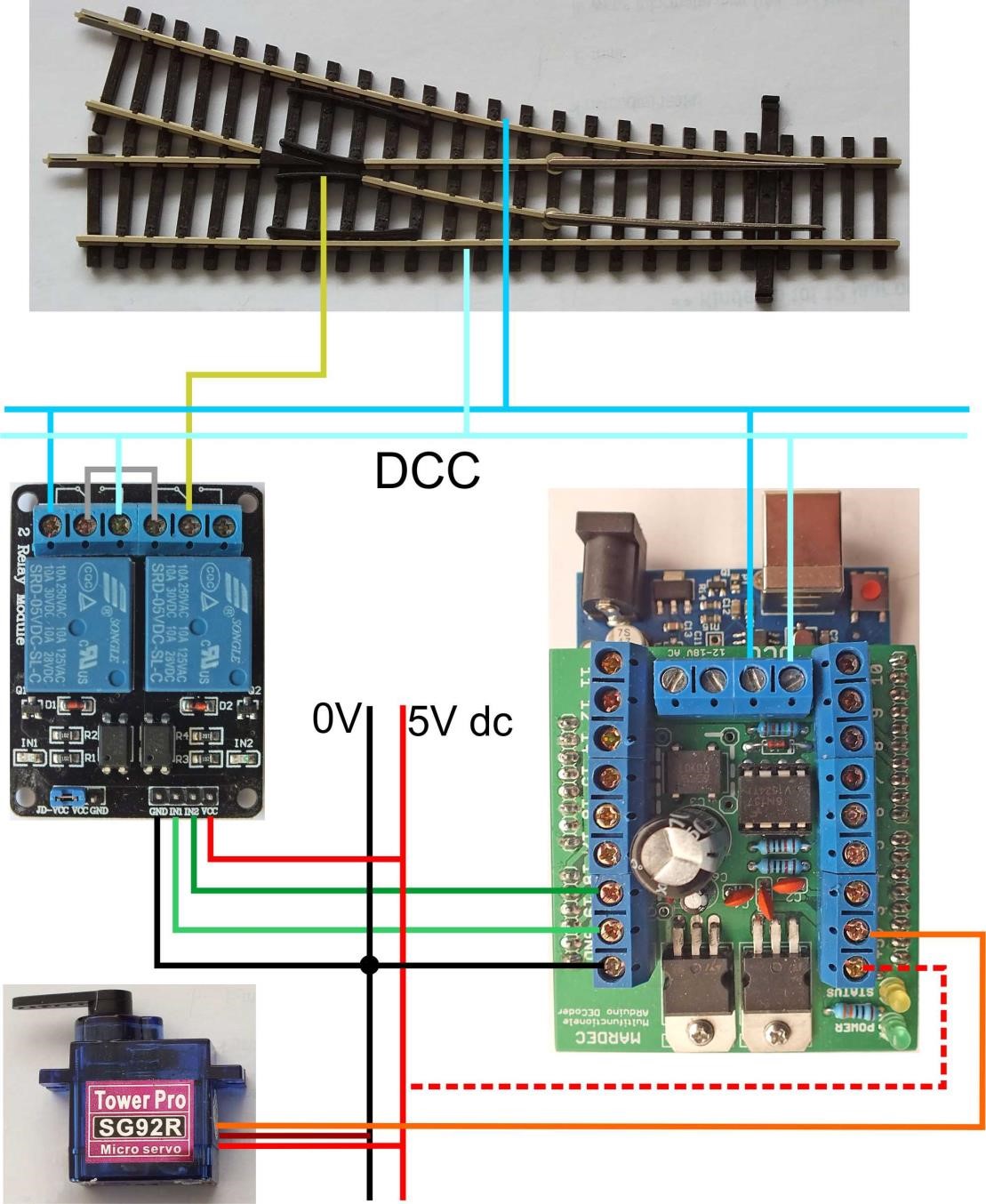


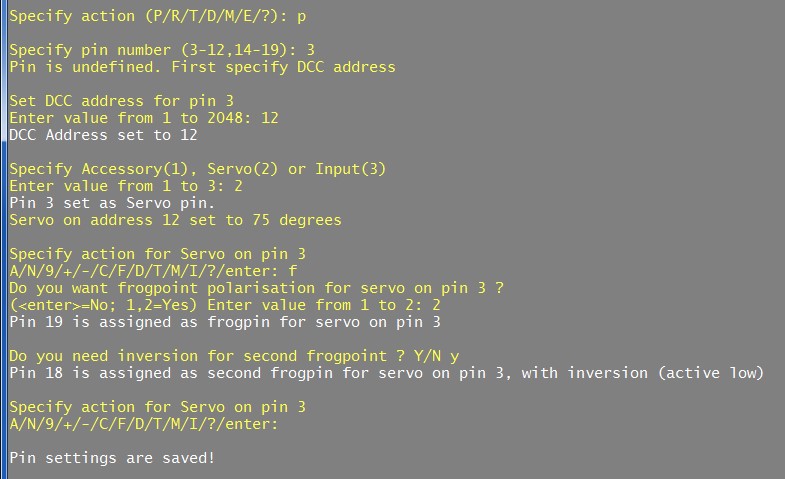
Arduino alimenté à partir de la même tension que la bande de LED. L'Arduino peut également être alimenté en courant alternatif.  
Connectez la connexion GND à la boucle de masse.Remarque: Les broches Arduino utilisées dépendent de la configuration réelle. Les connexions montrées ici sont juste un exemple.

Configuration:



Point de connexion avec double relais et servo



Les relais peuvent être fournis à partir du 5V sur le blindage DCC. Une alimentation externe est recommandée pour le servo. Aucune alimentation n'a été établie pour l'Arduino.  
Remarque: Les broches Arduino utilisées dépendent de la configuration réelle.  
Les connexions montrées ici sont juste un exemple.

# Historique des Versions

Notes de publication 2.1, mars 2016 Bugs:  
  Bug corrigé dans l'option de test générique; MARDEC se bloque lors du test du servo Bug corrigé dans l'horodatage du fichier journal; fichier journal non renommé avant 10 heures  
  Bug corrigé dans le texte d'aide; Texte d'aide incorrect pour les accessoires Bug corrigé dans la question O / N (Y majuscule a été interprété comme 'Non') Nouvelles fonctionnalités:  
  Flickermode ajouté pour les accessoires Ajouté aléatoire on / off pour les accessoires.  
Discolights ajoutés  
  Ajout de l'option permettant d'utiliser le bouton de réinitialisation pour le mode de configuration en absence ou en signal DCC. Ajout d'une option pour corriger le décalage d'adresse ou ROCO MM, z21 ou Z21.  
 Changements  
Frogpin est maintenant automatiquement affecté  
La broche de copain (précédemment appelée broche «compagnon») est maintenant automatiquement affectée  
Maximum '1'-period augmenté de 64 à 80 uS pour la conformité avec Digikeijs DR5000  
 -------------------------------------------------- ----------  
 Notes de publication 2.2, mai 2016 Bugs:  
Bug corrigé lors de la restauration ou de la position du servo  
Bug corrigé lors du changement de mode ou d'accessoire si le binôme est déjà présent.  
 Nouvelles fonctionnalités:  
  Commande de document automatique lors du démarrage du mode de configuration  
 Suppression de l'option reset et remplacé par la présence du câble USB (Merci Sjoerd)  
 -------------------------------------------------- ----------  
 Notes de version 2.3, mai 2016 Modifications:  
Indicateur led continu en mode de configuration  
La DEL clignote lorsque l'adresse configurée est détectée  
 -------------------------------------------------- ----------  
 Notes de version 2.3.1, mai 2016 Modifications:  
Le temps «On» minimum pour le one-shot et le clignotement est abaissé de 100 à 50 ms.  
 Bugs:  
Le message "commande non applicable" ne s'affiche plus lorsque vous appuyez sur "entrer"Les broches 9 et 10 ne peuvent pas être une broche PWM  
 -------------------------------------------------- ----------  
 Notes de publication 3.0 Septembre 2016 Modifications:  
  L'adresse DCC pour le mode de configuration a été supprimée. A déjà été remplacé par l'adresse DCC 'C'-clé pour enregistrer-option supprimée.  
Tous les changements en mode normal sont stockés dans l'EEPROM  
  Tous les servos et accessoires sont stockés sous tension. L'initialisation des servos ne déplace plus les servos Arduino démarre avec 3 clignotements ou LED d'état.  
 Passer du mode normal au mode configuration est maintenant sans redémarrage -------------------------------------- ----------------------  
Notes de version 3.1 Février 2017  
    
 Changements  
Ajout d'une option pour une polarisation améliorée du point de grenouille avec deux relais  
Ajout d'une option pour détecter une entrée pour contrôler les sorties configurées  
Ajout d'une option d'aide sur M-command pour les accessoires  
Le type d'accessoire mono-coup permet de basculer la sortie d'un accessoire ou d'un servo avec la même adresse DCC une fois terminé.  
 Bugs:  
La fonction marche / arrêt aléatoire n'a pas fonctionné correctement pendant de longues périodes (grâce à Bart Sanders). Le seul tir a également tiré sur la transition 1-> 0 du signal DCC.  
Lors d'une activité de haute servo, Mardec pourrait se terminer en boucle (merci Henk de Gans)  
-------------------------------------------------- ----------  
Notes de publication 3.1.1 mars 2017  
   
Bugs: (merci Henk Schippers)  
 Un numéro de broche <enter-only> -1 Il était possible de modifier un numéro de broche qui était déjà une «deuxième» broche de grenouille.  
 -------------------------------------------------- ----------  
Notes de version 3.2 Juin 2017  
    
 Changements  
 Ajout de la troisième adresse à contrôler par entrée  
 Temps de retard ajoutés et entrées de deuxième et troisième adresse  
 Ajout du support DCC pour les entrées; les entrées peuvent être activées par un signal DCC  
    
 Bugs (Merci Bart Sanders):  
Les types d'accessoires 3 et 4 ne fonctionnaient pas toujours correctement.  
Le type d'accessoire 7 n'a pas été enregistré après le changement.  
Les broches 16-19 n'ont pas pu être réglées en entrée  
    
-------------------------------------------------- ----------  
Notes de publication 3.3 Septembre 2017  
   
Changements  
Toutes les broches peuvent être une broche servo maintenant.  
Ajout d'une option pour l'inversion du relais de grenouille  
Outil ajouté pour la gestion des sessions Putty  
   
-------------------------------------------------- ----------Release Notes 3.3.1 september 2017

Changes: (Thanks Ruud Boer)

Lors de la définition d'un nouveau servo est le premier ensemble de 90 degrés avant de définir à 75 degrés. Cela empêche que le servo commence à balayer à 0 degré.

[DOWNLOAD HIER DE ARCOMORA SOFTWARE](http://www.globalvisuals.nl/downloads/Arcomora.zip)

**DCC et accessoires**

Quel est exactement le contrôle des accessoires avec un signal DCC?  
Un accessoire est tout sur et le long de votre voie qui peut être actionné électriquement. Ce sont des servos, des LED, des lumières, des bobines de commutation, des signaux, des relais et des moteurs.  
Il est important de se rappeler qu'il existe deux types de signaux DCC. Un pour les locomotives et un pour les accessoires. Au centre, les accessoires sont généralement appelés articles magnétiques.  
Un signal DCC contient toujours une adresse et des données. L'adresse indique l'appareil auquel les données sont destinées et les données elles-mêmes indiquent ce que l'appareil doit faire.  
Les données DCC pour les locomotives contiennent beaucoup d'informations telles que la vitesse, la direction et l'éclairage.  
Les données DCC pour les accessoires ne contiennent cependant qu'un seul bit de données et n'ont donc que deux valeurs.  
Dans le monde numérique, nous appelons cela les valeurs 0 (zéro) et 1 (un).  
Cela signifie que vous ne pouvez allumer ou éteindre qu'un accessoire. Donc, cela fonctionne comme un interrupteur marche / arrêt normal. Mais vous pouvez faire beaucoup avec ça de toute façon.  
Par exemple:  
- Un servo peut être tourné dans deux positions - Une lumière ou une LED peut être allumée ou éteinte. - Une lumière ou une LED clignote ou est simplement éteinte - Un moteur peut tourner ou rester immobile.  
- Un signal peut montrer deux images de signal différentes  
- Dans le cas des interrupteurs électromagnétiques, l'une ou l'autre des bobines est excitée.

Pour réaliser tous ces effets il y a des décodeurs accessoires. Un décodeur a une entrée pour le signal DCC et un certain nombre de sorties. Les appareils à contrôler sont connectés à ces sorties. En configurant un décodeur, vous pouvez décider vous-même de ce qu'il faut faire exactement. En outre, cela signifie que vous spécifiez à quelles adresses DCC le décodeur doit "écouter" et comment le décodeur doit répondre au bit de données correspondant. Selon l'application, il existe différents types de décodeurs. Il existe des décodeurs séparés pour les servos, pour les signaux, pour les bobines AC et les décodeurs généraux.  
Parce que les bobines alternatives sont l'une des plus anciennes applications pour les accessoires, les deux états sur un panneau sont généralement marqués d'un symbole de changement. Dans une position, l'aiguillage est droit et dans l'autre position, le virage a été dévié. Certains panneaux de contrôle ont deux boutons séparés pour les deux modes. Ceux-ci sont souvent colorés en rouge et vert. Rouge signifie «dévié» (= Rund = rouge) et vert pour «droit» (= revenu = vert). Ce symbolisme est également utilisé pour tous les autres accessoires.  
Dans le cas de la plupart des plantes, le rouge correspond à un 0 numérique (zéro) et vert à un 1 (un).

Mardec

Le Mardec est un décodeur DCC général approprié pour les servos, les bobines alternatives et une multitude d'autres effets. Il utilise un matériel standard, à savoir un micro-ordinateur Arduino et Mardec lui-même est un programme standard pour un Arduino. Avec Mardec, vous n'avez pas à vous programmer, mais vous pouvez configurer de nombreuses applications de manière simple.

ArSigDec

L'Arsigdec est un décodeur DCC spécialement conçu pour les signaux lumineux. Deux images de signaux complètement différentes peuvent être affichées avec les deux «positions» des données accessoires. Vous pouvez configurer toutes ces images de signal vous-même.  
Il utilise exactement le même matériel que le Mardec. Seul le programme est différent.