

Notice de montage et d'utilisation

Réaliser son « **Décodeur Digital et Analogique pour servomoteurs avec une carte Arduino** »

Le projet décrit la réalisation d'une carte de commande de 6 servomoteurs compatibles analogique et digital (DCC).

La base de ce projet est une carte ARDUINO que l'on trouve dans le commerce pour environ 20€ à laquelle sera rajoutée une interface « carte maison » pour piloter les servomoteurs.

Cette notice présente la carte développée par la section Modélisme de l'UAICF Nevers-Vauzelles (schéma & programme).

Le décodeur a été testé avec une centrale Roco 10764 et sa Multimauss, une centrale Roco Z21, une centrale NanoX et une centrale Arduino avec DCC++.

Le décodeur est issu d'une compilation et d'une adaptation de plusieurs projets trouvés sur le net. La notice peut paraître assez longue mais l'utilisation est très simple.

1. Principe du servomoteur _____	1	5. Carte à réaliser _____	5
2. Présentation du décodeur de servomoteurs _____	2	a. Sources d'alimentation.....	5
3. Présentation des cartes ARDUINO _____	2	b. Schéma.....	6
4. Configuration de la carte ARDUINO _____	3	c. Description de la carte.....	7
a. Installation du logiciel Arduino	3	d. Utilisation	7
b. Configuration	3	e. Paramétrage du programme	7
c. Librairie à installer	4	6. Réalisation _____	8
d. Modification à apporter dans le code source.....	4	a. Schéma de la carte d'extension pour carte Arduino8	
e. Programme Source	4	b. Programme source.....	9
		c. Configuration.....	9
		7. Rapporteur _____	9

1. Principe du servomoteur

Contrairement à un moteur à courant continu, qui peut être piloté par des variations de tension ou par allumage / extinction, le servomoteur réagit en fonction d'une impulsion de durée variable.

Le signal de commande est un signal logique périodique. La largeur des créneaux va commander l'angle de l'arbre du servomoteur.

Un servomoteur se pilote par l'intermédiaire d'un câble à trois fils de couleurs différentes :

- le **noir** ou **marron** raccordé à la masse,
- le **rouge** raccordé à la tension d'alimentation continue (+5 Volts en général),
- le **jaune, orange ou blanc** raccordé au signal de commande.

Ce câble permet donc à la fois d'alimenter et transmettre les consignes de position par le fil de signal.

Les servomoteurs peuvent servir à piloter des aiguillages mais aussi tout type d'animation comme des signaux mécaniques.

2. Présentation du décodeur de servomoteurs

L'ensemble du système décrit ci-après permet de piloter des servomoteurs en digital et / ou analogique :

- piloter jusqu'à 6 servomoteurs,
- pilotage à l'aide bouton à levier,
- pilotage en DCC,
- paramétrage des angles de rotation de chaque servomoteur,
- paramétrage de la vitesse de rotation de l'ensemble des servomoteurs.

3. Présentation des cartes ARDUINO

Les cartes ARDUINO sont des cartes électroniques basées autour d'un microcontrôleur Atmega du fabricant Atmel, dont le prix est relativement bas pour l'étendue possible des applications.

Le logiciel de programmation est gratuit et compatible Windows, Mac et Linux.



Ce microcontrôleur peut être programmé pour analyser et produire des signaux électriques, de manière à effectuer des tâches très diverses pour le pilotage de divers équipements.

Les modules ARDUINO trouvent pleinement leur place dans le monde du modélisme ferroviaire pour piloter des locomotives, des accessoires, des feux de signalisation.

Les cartes comprennent un certains nombre d'entrées / sorties analogiques et numériques paramétrables dans un langage proche du « langage C ». Ces entrées / sorties sont accessibles via des connecteurs femelles (au pas de 2,54 mm). On peut ajouter des platines optionnelles (ou carte d'extension) qui permettent de leur adjoindre de multiples possibilités et interfaces supplémentaires.

On peut notamment développer des interfaces pour piloter du « matériel » en digital.

De nombreux sites proposent à la vente les cartes ARDUINO. Pour le projet de pilotage des servomoteurs, on utilisera une carte **ARDUINO UNO** qui coûte environ 20 €.

L'intérêt est de pouvoir personnaliser les programmes et de pouvoir réaliser des kits « maison » à un prix très raisonnable.

Pour programmer la carte, il faut disposer d'un ordinateur (PC, Mac, Linux) et d'un cordon USB.

La carte fonctionne avec une alimentation externe lorsqu'elle n'est pas auto alimentée par le port USB.

4. Configuration de la carte ARDUINO

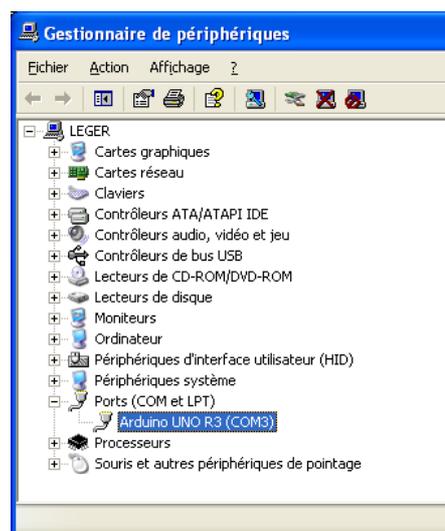
a. Installation du logiciel Arduino

Il faut télécharger la dernière version du logiciel Arduino disponible sur le site officiel <http://www.arduino.cc/en/Main/Software>

Le programme fonctionne sous Windows, Macintosh et Linux. Il faut suivre les conseils d'installation. Il n'y a pas de difficultés particulières.

L'Arduino est un Logiciel Libre et Matériel Libre sous license Creative Commons "paternité, non commercial et licence contaminante". Toute liberté est permise à qui voudrait faire évoluer le matériel ou la plateforme de programmation dans le respect de la licence. Le site officiel du projet Arduino est <http://www.arduino.cc>

- 1 - Télécharger le logiciel Arduino
<http://www.arduino.cc/en/Main/>
- 2 - Dé zipper le fichier téléchargé
- 3 - Brancher l'Arduino à l'ordinateur avec le port USB
- 4 - Lancer le programme « arduino.exe »
- 5 - Aller dans le gestionnaire de périphérique pour vérifier que la carte a bien été reconnue (Port COM)
- 6 – Si la carte n'a pas été reconnue, mettre à jour le pilote : les drivers sont présents dans le répertoire « Drivers » du dossier Arduino. Eventuellement utiliser le driver « Arduino UNO.inf » contenu dans le fichier zip « Old_Arduino_Drivers.zip » si celui-ci n'est pas reconnu.

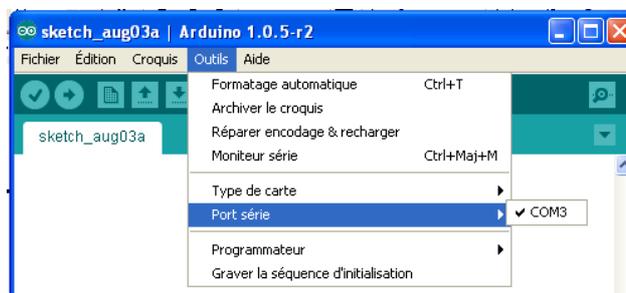


b. Configuration

Pour la configuration et la programmation des cartes ARDUINO, il existe de nombreux tutoriaux disponibles sur Internet comme : <http://fr.openclassrooms.com/sciences/cours/arduino-pour-bien-commencer-en-electronique-et-en-programmation>

Lors du lancement du programme Arduino, il faut désigner le type de carte utilisée (UNO) et cocher le numéro du port COM (USB-Série) où est connectée la carte.

Ces paramètres sont modifiés depuis le menu « Outils ».



c. Librairie à installer

Pour utiliser la carte ARDUINO avec le DCC, il faut télécharger et installer une bibliothèque gratuite développée par un projet américain qui s'appelle "Model Railroading with Arduino".

Il faut télécharger le fichier ZIP depuis le site <http://mrrwa.org> ou à l'adresse : <http://sourceforge.net/projects/mrrwa/files/MRRwA-2011-12-31.zip/download>

Il faut dézipper le répertoire « NmraDcc » et le copier dans le répertoire « librairies » où est installé le programme Arduino.

```
// Bibliothèques nécessaires pour le DCC
#include <NmraDcc.h>
NmraDcc Dcc ;
DCC_MSG Packet ;
```

```
// Définition des variables, constantes
```

```
'NmraDcc' does not name a type
```

```
'FLAGS_ENABLE_INT0_PULL_UP' was not declared in this scope
```

Remarque :

Si le message ci-contre s'affiche, c'est que la bibliothèque « NmraDcc » n'est pas installée correctement.

Si le message ci-contre apparaît, ce n'est pas la bonne bibliothèque NmraDCC qui a été téléchargée. Le fichier « NmraDcc.h » situé dans la librairie NmraDcc doit notamment contenir le code suivant :

```
[...]
// Flag values to be logically ORed together and passed into the init() method
#define FLAGS_MY_ADDRESS_ONLY          0x01 // Only process DCC Packets with My Address
#define FLAGS_ENABLE_INT0_PULL_UP     0x02 // Enable the internal Pull-Up re
#define FLAGS_OUTPUT_ADDRESS_MODE     0x40 // CV 29/541 bit 6
#define FLAGS_DCC_ACCESSORY_DECODER   0x80 // CV 29/541 bit 7
[...]
```

Il faut également utiliser la bibliothèque « Servo ».

d. Modification à apporter dans le code source

Depuis la réalisation du programme de commande des servomoteurs, la bibliothèque NmraDcc a été modifiée.

Avec la nouvelle bibliothèque, il faut remplacer la ligne suivante dans le SETUP

```
Dcc.init( MAN_ID_DIY, 10, FLAGS_OUTPUT_ADDRESS_MODE |
FLAGS_DCC_ACCESSORY_DECODER | FLAGS_ENABLE_INT0_PULL_UP, 0 );
```

par la ligne :

```
Dcc.init( MAN_ID_DIY, 10, CV29_ACCESSORY_DECODER |
CV29_OUTPUT_ADDRESS_MODE, 0 );
```

e. Programme Source

Le programme proposé par la section Modélisme de l'UAICF Modélisme Nevers-Vauzelles doit être chargé dans la carte ARDUINO à l'aide d'un cordon USB / Série.

Le code source de la dernière version peut être téléchargé sur le site <http://modelisme58.free.fr>

5. Carte à réaliser

La section Modélisme a développé une carte d'extension spécifique pour le DCC qui permet plusieurs usages et notamment le pilotage d'accessoires.

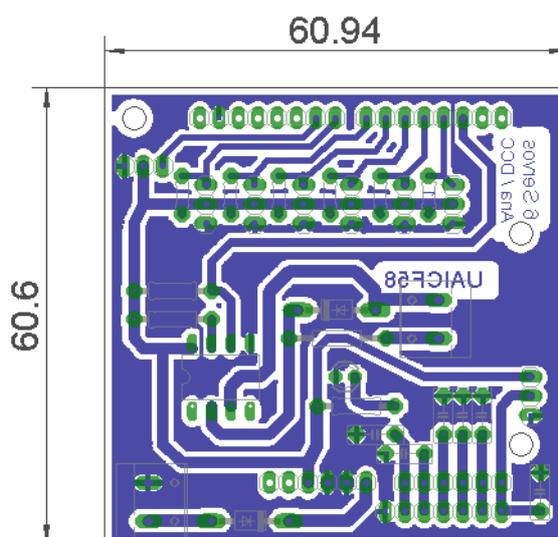
La carte est issue d'une base du projet « DCC Monitor ».

a. Sources d'alimentation

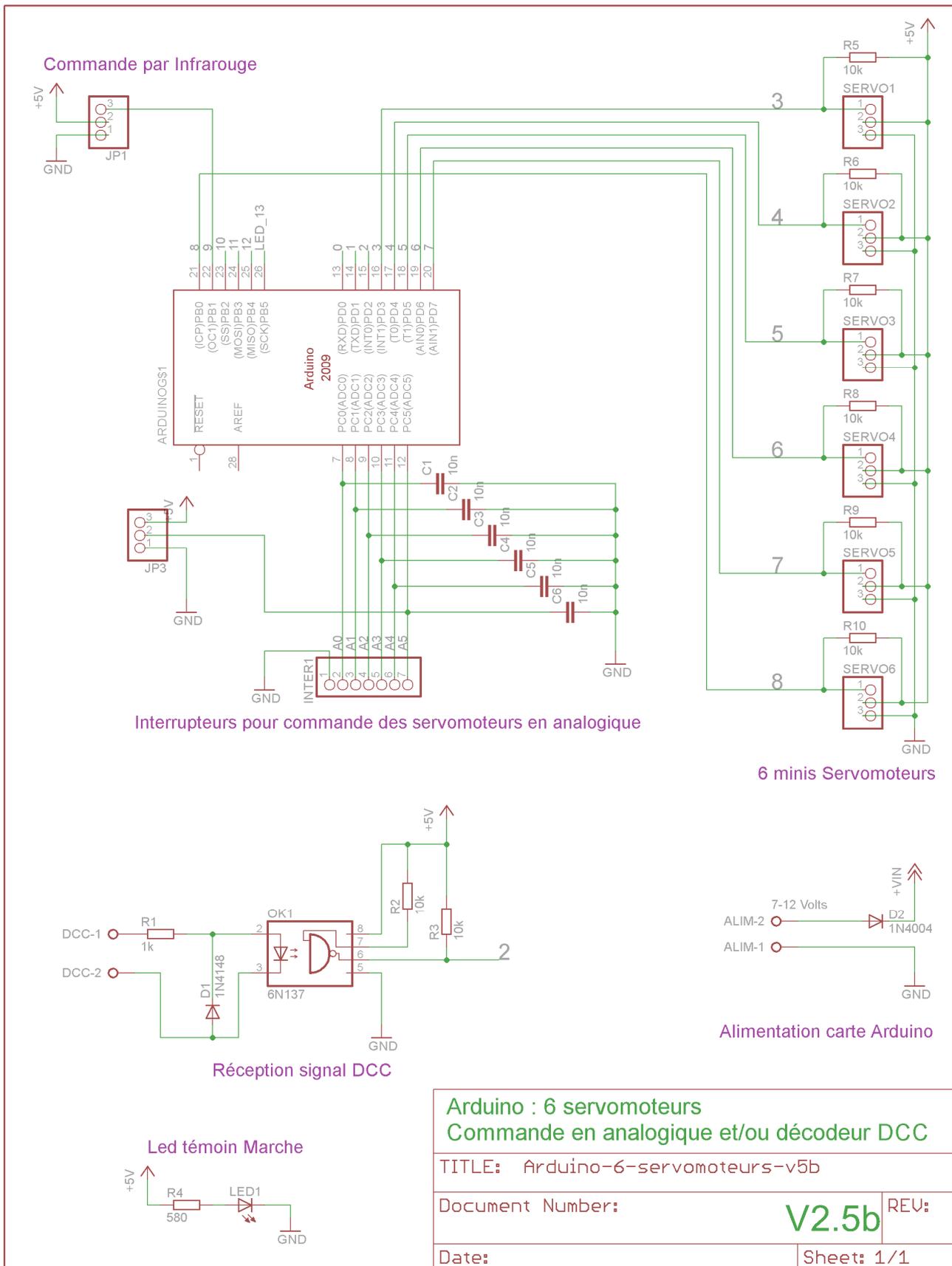
Rappel : la carte ARDUINO doit disposer de sa propre alimentation en 9 –12 Volts continu ou être alimentée via le port USB (limité à 500 mA).

La carte d'extension doit se connecter à la carte ARDUINO UNO en respectant les polarités et les ports de connexion. La carte comprend donc 2 sources d'alimentation :

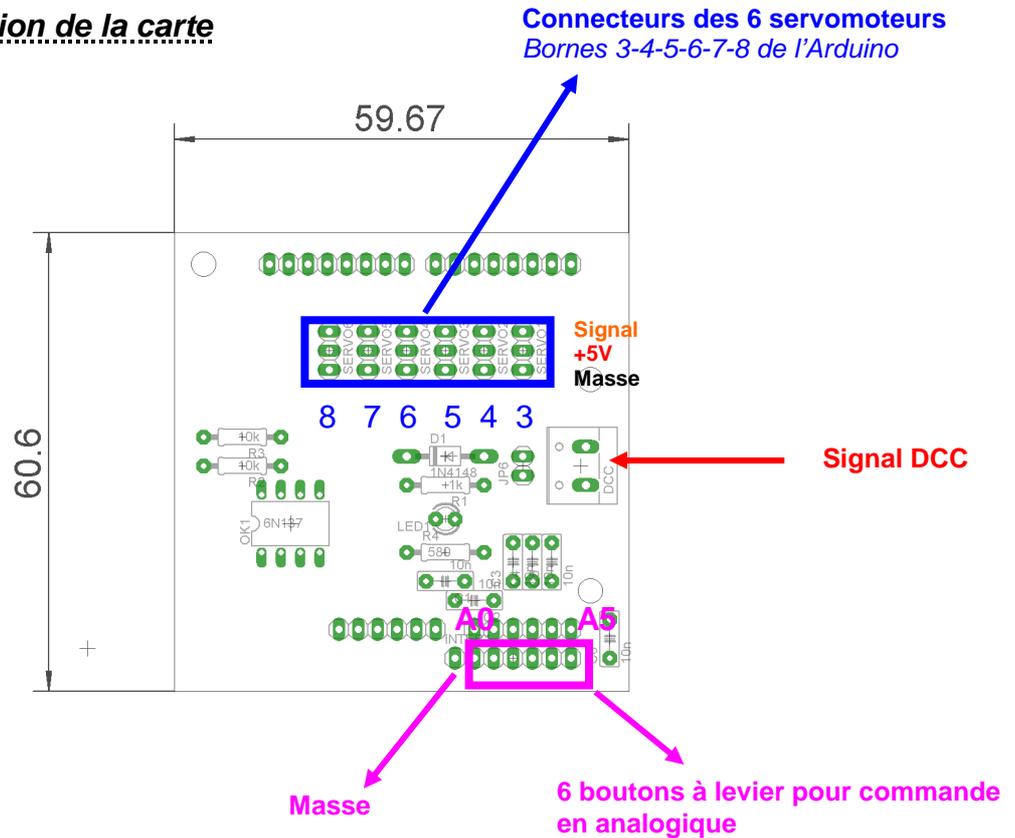
- Alimentation DCC provenant de la centrale,
- Alimentation continue de 7 à 12 Volts pour la carte.



b. Schéma



c. Description de la carte



d. Utilisation

Les servomoteurs peuvent être pilotés à la fois en DCC et à l'aide de boutons à levier. En analogique, la commande se fait par une mise à la masse des entrées A0, A1, A2, A3, A4, et A5 de la carte Arduino.

Les servomoteurs disposent donc d'une double commande.

e. Paramétrage du programme

▪ Adresses DCC

La configuration des adresses DCC se réalise dans la fonction « Void ConfigureDecodeursServos() ».

```
void ConfigureDecodeurServos()
{
  accessory[0].address = 1; // Adresse DCC de l'aiguillage
  ArduinoBorne[0] = 3; // Borne de l'Arduino utilisée pour piloter le servomoteur
  servos[0].minangle=30; // Position extrême du servomoteur (mini - valeur mini = 0 )
  servos[0].maxangle=135; // Position extrême du servomoteur (maxi - valeur maxi = 180)
  servos[0].angle=servos[0].minangle; // Position initiale du servomoteur
  servos[0].inverted=true; // true = inversion des positions mini et maxi
  servos[0].functionnumber=0; // Numéro d'ordre de configuration des accessoires pilotés
  servos[0].sens=2; // Position de l'aiguillage
  ...
}
```

- Angle de rotation de l'axe

L'angle de rotation est configurable pour chaque servomoteur. Un rapporteur (cf. ci-après) peut être utilisé pour visualiser l'angle de rotation.

Dans la fonction « Void ConfigureDecodeursServos() », il est configuré pour chaque servomoteur la position mini et maxi du servomoteur.

- Temps impulsion

La vitesse de déplacement est définie avec la variable « servotimer ». Elle est identique pour tous les servomoteurs.

```
const byte servotimer = 30; // Vitesse de déplacement du servomoteur
// Plus la valeur est faible, plus la vitesse est rapide
```

- Aiguillage avec cœur non isolé

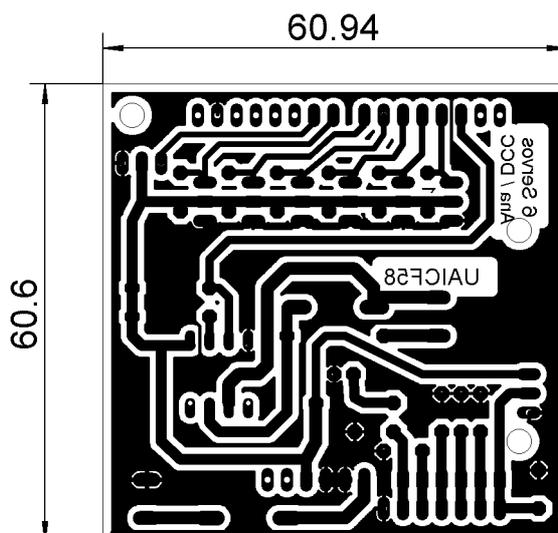
Pour les aiguillages avec cœur non isolé, la carte décrite ne gère pas le changement de la polarité. Pour les cœurs non isolés, il peut être rajouté un fin de course sur l'axe du servomoteur.

6. Réalisation

a. Schéma de la carte d'extension pour carte Arduino

La carte a été réalisée avec le logiciel Eagle, logiciel de conception assistée par ordinateur de circuits imprimés : <http://www.cadsoftusa.com>

La carte peut également être réalisée avec une platine d'essais à partir du schéma donné ci-dessus.



b. Programme source

Le programme source de la carte de commande des accessoires est disponible sur le site <http://modelisme58.free.fr> ou sur demande pour courriel à l'adresse modelisme58@free.fr

c. Configuration

La configuration de la carte peut être affichée au démarrage en lançant depuis le programme Arduino, la barre de menu « Outils » puis « Moniteur série ». L'alimentation de la carte doit être réalisée par le cordon USB.

7. Rapporteur

Le rapporteur proposé va permettre de calculer l'angle de rotation du servomoteur. Ce rapporteur peut-être très utile pour réaliser les réglages.

1. Imprimer la fiche sur une feuille de papier rigide ou coller la feuille sur une plaque de plastique.
2. Découper le carré et réaliser l'encoche pour pouvoir enfiler le rapporteur dans le servomoteur.
3. Faire varier les paramètres du servomoteur et noter pour chaque réglage l'angle du déplacement de l'axe.

Astuce : Mettre une plaque de plastique sous le rapporteur pour maintenir le rapporteur en équilibre sur le servomoteur.

