

# Notice de montage et d'utilisation « **Commande de servomoteurs ARDUINO** »

## Version 2

**Le projet décrit la réalisation d'une commande de 6 servomoteurs pour piloter des commandes d'aiguillages en mode digital ou analogique.**

**La carte permet de commander 1 ou 2 ou 3 aiguillages triples équipés de 2 servomoteurs.**

**Compatible avec une carte Arduino UNO, NANO**

La base de ce projet est une carte ARDUINO UNO ou NANO que l'on trouve dans le commerce à partir de 20€ environ suivant les modèles.

Cette notice présente la carte et le programme développés par la section Modélisme de l'UAICF Nevers-Vauzelles.

Pour la commande en digital pour le modélisme ferroviaire, le décodeur d'accessoires a été testé avec une centrale Roco 10764 et sa Multimauss, une centrale Roco Z21 et une centrale NanoX. Le pilotage des accessoires a également été testé avec Rocrail et une centrale Roco et NanoX.

Le projet 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 assez simple.

**Avec la version 2, l'interface ou platine est entièrement compatible avec la version 1.**

**La version 2 permet de réaliser les réglages sans avoir besoin de recharger le programme : le paramétrage sera fera par le port-série ou en Wifi avec un ESP.**

---

<b>1. Principe du servomoteur</b> _____	<b>2</b>	<b>5. Platine / Carte d'extension</b> _____	<b>7</b>
<b>2. Présentation du décodeur de servomoteurs</b> _____	<b>4</b>	a. Schéma pour la carte Arduino NANO .....	8
<b>3. Principaux changements entre V1 et V2</b> _	<b>4</b>	b. Platine pour la carte Arduino NANO .....	9
<b>4. Configuration de la carte ARDUINO</b> _____	<b>5</b>	<b>6. Paramétrage et commandes</b> _____	<b>11</b>
a. Installation du logiciel ARDUINO .....	5	a. Paramétrage général via le port Série .....	12
b. Configuration .....	6	a. Paramétrage individuels des servomoteurs .....	12
c. Librairies à installer.....	6	b. Paramétrage via une application dédiée.....	13
d. Téléverser le Programme Source .....	6	<b>7. Dépannage</b> _____	<b>14</b>
e. Ou Uploader un fichier « HEX » .....	7	a. Carte ARDUINO reconnue ? .....	14
		b. Débogage à l'aide du Moniteur Série .....	14
		c. Erreur lors de la compilation.....	14

---

# 1. Principe du servomoteur

## a. Fonctionnement

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 :

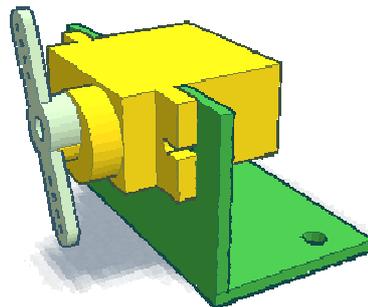
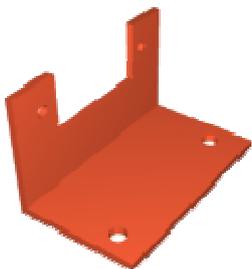
- le **noir** ou **marron** raccordé à la masse,
- le **rouge** raccordé à la tension d'alimentation continue (+5 Volts en général),
- le **jaune, orange ou blanc** raccordé au signal de commande.

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

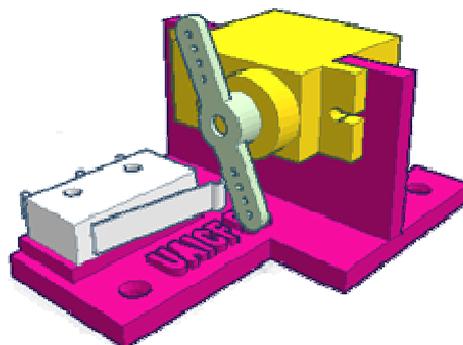
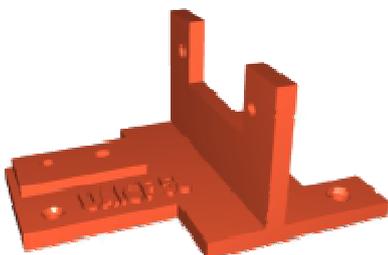
Les servomoteurs peuvent servir à piloter des aiguillages mais aussi tout type d'animation comme des signaux mécaniques.

## b. Supports de servomoteur

Il existe plusieurs systèmes pour fixer les servomoteurs. En voici 3 :



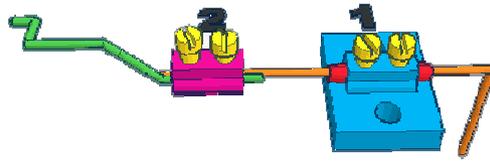
*Support en équerre pouvant être réalisé dans une équerre en aluminium*



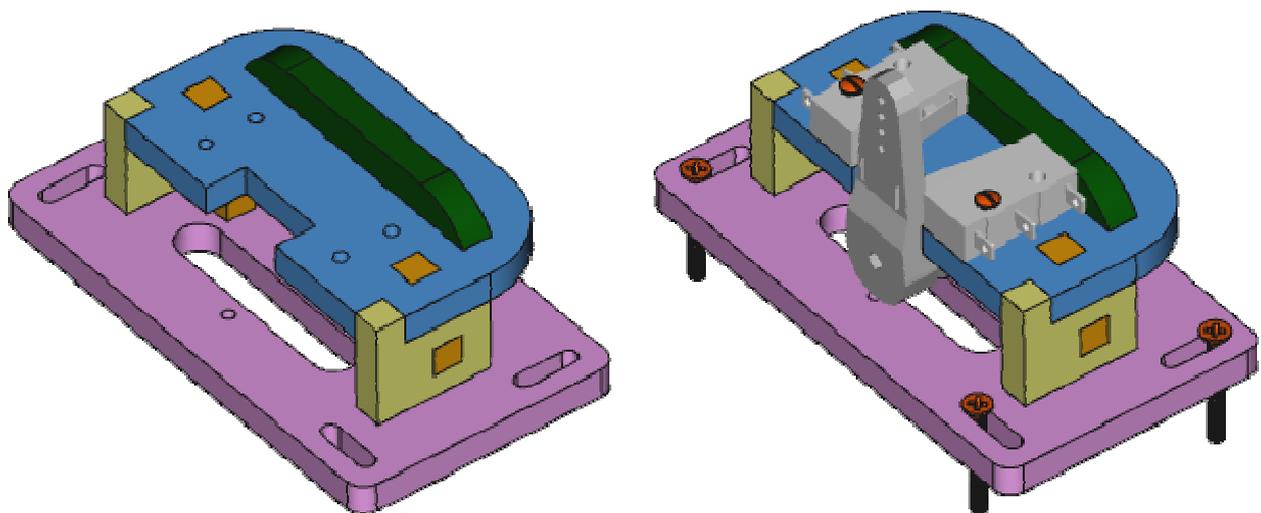
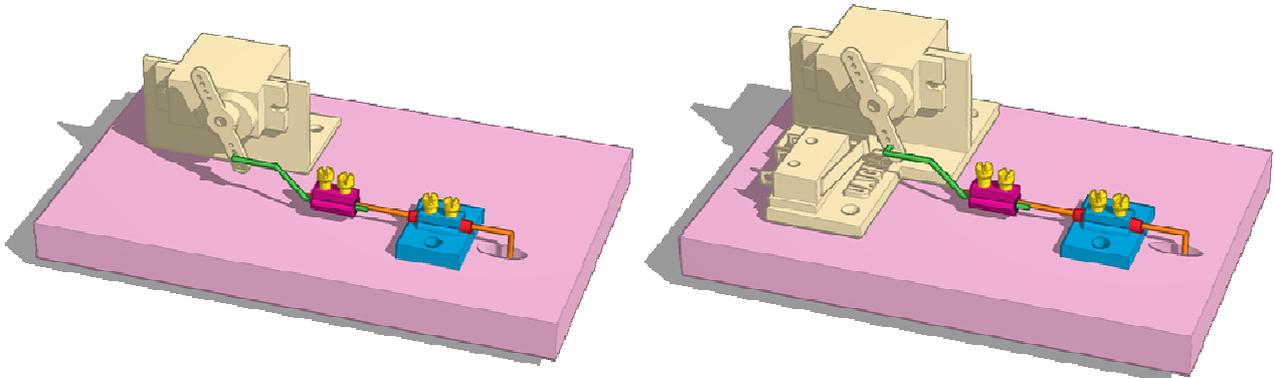
*Support réalisé en impression 3D avec un fin de course*

Pour imprimer le support, le fichier « **STL** » est disponible sur le site de <http://modelisme58.free.fr>

Pour positionner et régler le servomoteur, il est suggéré d'utiliser 2 dominos à vis.



Axe Vert : 0,8 à 1 mm | Axe Orange : 1 mm



*Support découpé dans du bois contreplaqué de 5 mm*

Le support en bois peut recevoir 4 fins de course. On peut superposer 2 fins de course les uns sur les autres.

2 fins de course sont utilisés pour gérer l'alimentation du cœur d'aiguille si nécessaire.

Pour la découpe laser, les fichiers au format « **DXF** » sont disponibles sur le site de <http://modelisme58.free.fr>

## 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.

Le projet est compatible avec les cartes Arduino UNO et NANO.

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.

Dans le cadre de ce projet, il est proposé une platine conçue pour une carte Arduino Nano.

**Adresse DCC, angles, vitesse sont paramétrables directement via la liaison série, une application gratuite ou une interface Web SANS avoir besoin de reprogrammer la carte Arduino.**

## 3. Principaux changements entre V1 et V2

	Version 1	Version 2
<b>Commande</b>	6 boutons à levier	6 boutons à levier
<b>Paramétrage adresse DCC, nombre de fichiers, temporisation en mode aléatoire, ...</b>	Valeurs enregistrées dans le programme : reprogrammer la carte après modifications	Valeurs modifiables via des commandes envoyées sur les ports Série : les valeurs sont enregistrées dans l'Eeprom
<b>Envoi des commandes, configuration via le port Série 0</b>	NON	OUI
<b>Choix du type de centrale DCC pour gérer le décalage des adresses des accessoires</b>	NON	OUI
<b>Sélection du type de Centrale (décalage de 4 adresses entre Lenz et Roco)</b>	NON	OUI

La principale modification de cette version « 2 » du décodeur réside dans le paramétrage et la possibilité de communiquer avec un ESP8266 pour entrer dans le monde des objets connectés : les modifications du programme portent essentiellement sur une modification du code source.

La platine de la version « 1 » est totalement compatible.

Les paramètres de configuration sont sauvegardés dans la mémoire Eeprom de l'Arduino : les données ne sont pas perdues lors de la mise hors tension de la carte.

## 4. Configuration de la carte ARDUINO

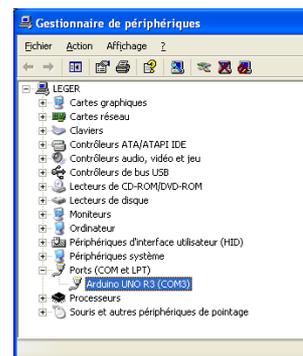
### c. Installation du logiciel ARDUINO

Il faut télécharger la dernière version du logiciel ARDUINO disponible sur un 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.

- 1 - Télécharger le logiciel ARDUINO
- 2 - Dézipper 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.

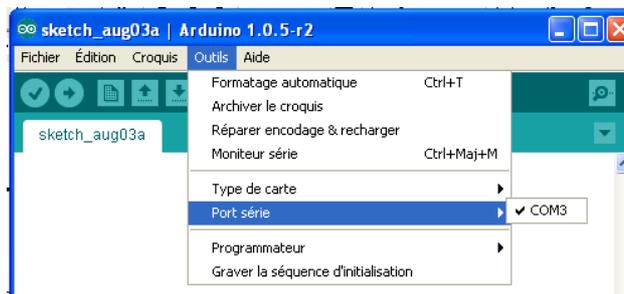


## d. Configuration

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 (ARDUINO UNO ou NANO) et cocher le numéro du port COM (USB-Série) où est connectée la carte.

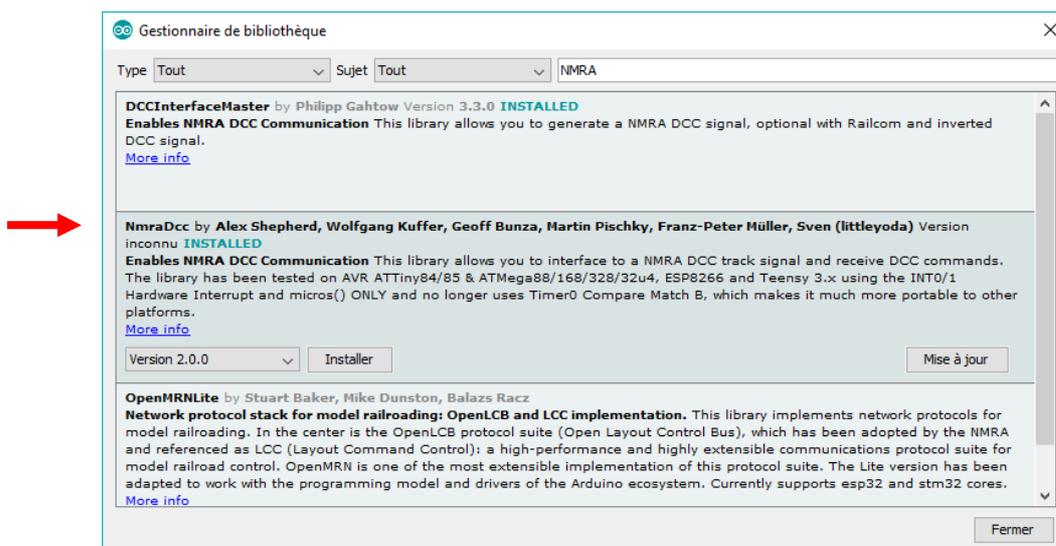
Ces paramètres sont modifiés depuis le menu « Outils ».



## e. Bibliothèques à installer

Pour utiliser la carte ARDUINO avec le DDC, 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 <https://github.com/mrrwa/NmraDcc> ou directement à l'aide du gestionnaire de bibliothèque dans le logiciel Arduino. Saisir « NMRA » dans le champ de recherche :



Si l'installation est manuelle, il faut dézipper les fichiers et les copier dans le répertoire « C:\Utilisateur\MON-NOM\Documents\Arduino »

La bibliothèque « EEPROM.h » permet d'enregistrer des données dans la mémoire de la carte Arduino et conservée une fois l'énergie coupée. Cette fonction est utilisée pour conserver les réglages des servomoteurs.

## f. Téléverser le 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.

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

Lorsque le programmer est prêt, il faut le téléverser à l'aide d'un cordon USB raccordant l'ordinateur et la carte ARDUINO.

Il faut bien veiller à sélectionner la bonne carte « **Arduino** » dans le menu « **Outils** » ainsi que le port « **COM** ».



Le bouton « **Moniteur Série** » permet d'afficher la configuration de la carte au démarrage et de visualiser les échanges d'information. Ceci est très utile pour identifier des dysfonctionnements et de cibler l'origine d'une panne.

### **g. Ou Uploader un fichier « HEX »**

Il est possible de téléverser le programme dans la carte ARDUINO directement à partir d'un fichier « hex » généré lors de la compilation.

Il faut au préalable installer le logiciel Xloader 1.00 qui est un petit programme s'exécutant en version portable : <https://commodore.software/downloads/download/246-wifly-commodore-wi-fi-modem/1966-xloader-v1-00> ou <https://www.hobbytronics.co.uk/arduino-xloader>

Connecter à l'ordinateur la carte ARDUINO NANO à programmer.

Lancer l'application « **XLoader** ».

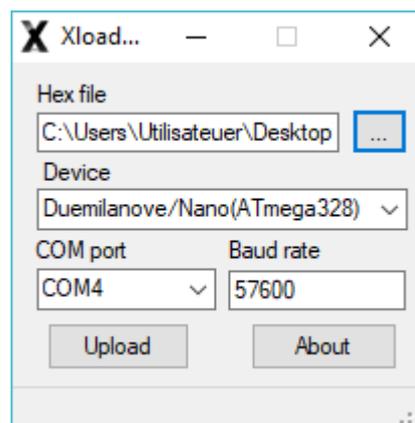
Sélectionner le port COM de l'Arduino dans le menu déroulant en bas à gauche.

Sélectionner la carte « **Duemilanove / Nano (ATmega 328)** » dans la liste déroulante « **Device** ».

Vérifier que l'application XLoader a défini le débit en bauds à 57600.

Indiquer le nom et l'adresse du fichier « **hex** » à téléverser.

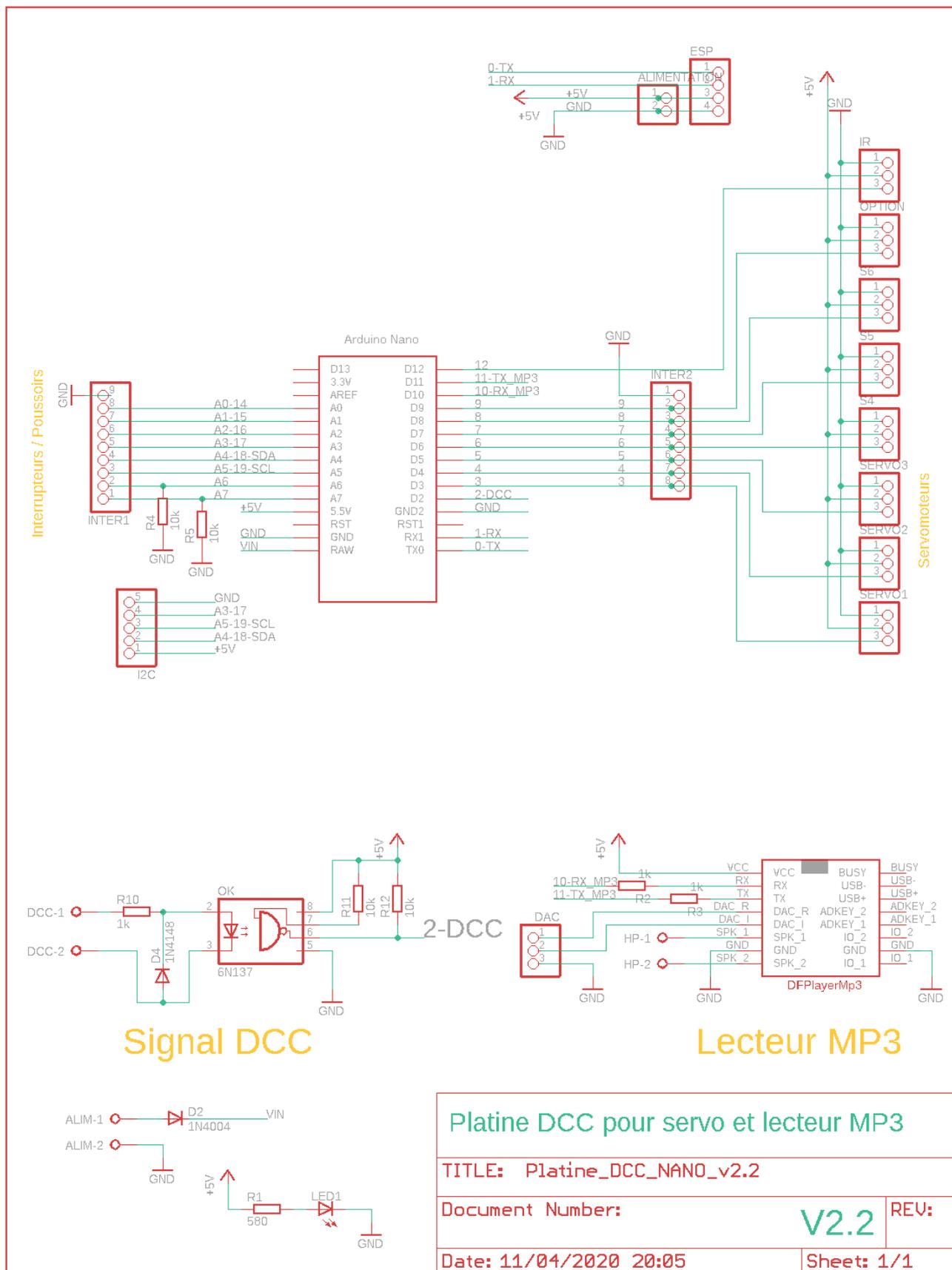
Cliquer sur le bouton « **Upload** » pour charger le fichier. Les LED RX/TX de la carte Arduino doivent clignoter. Après un certain temps, le programme est téléversé.



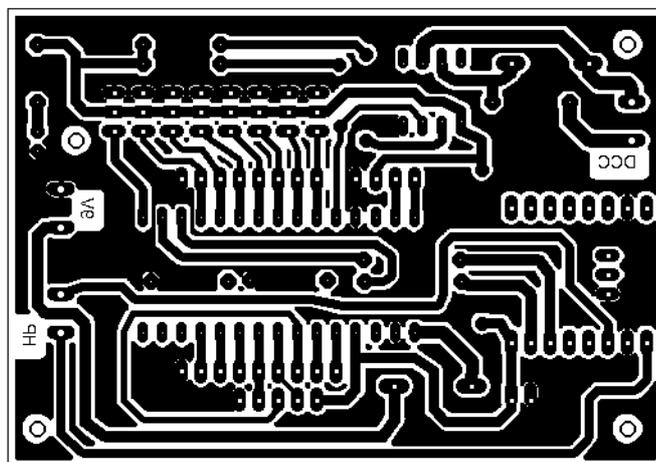
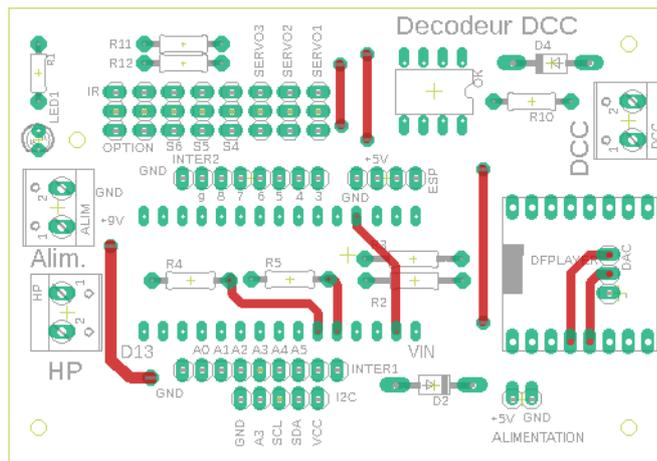
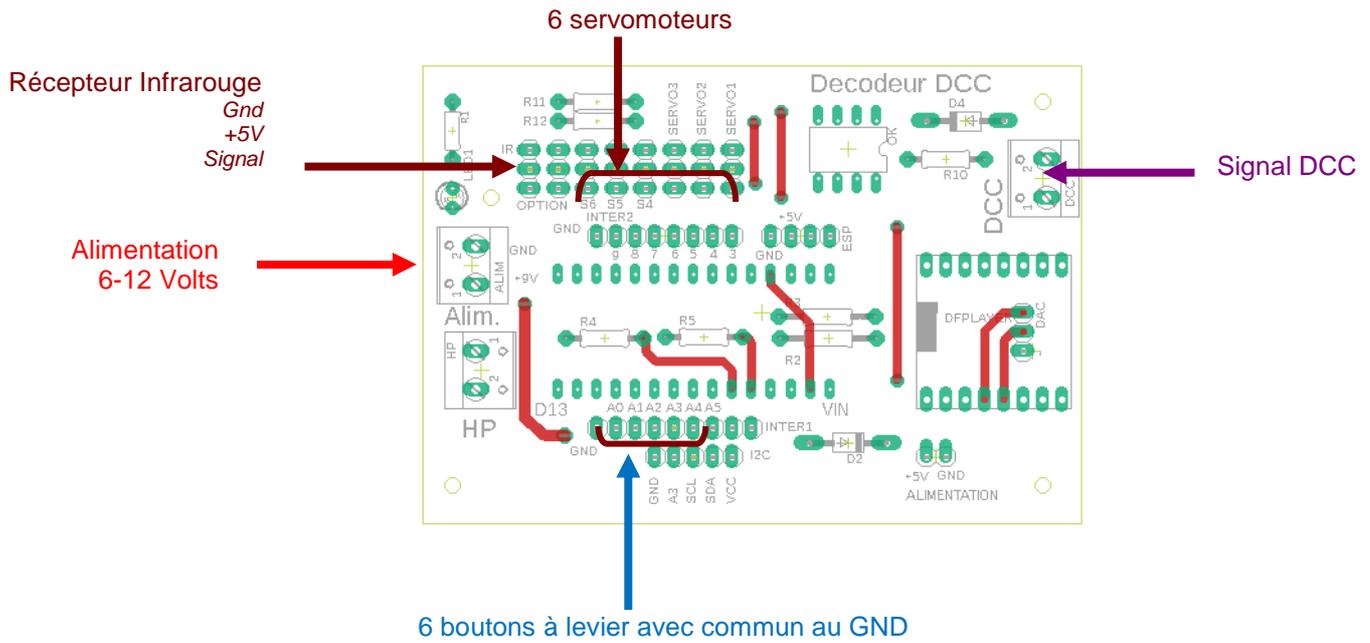
## **5. Platine / Carte d'extension**

La section Modélisme de l'UAICF Nevers-Vauzelles a développé une platine ou carte d'extension spécifique où vient s'emboîter la carte ARDUINO NANO.

## a. Schéma pour la carte Arduino NANO



## b. Platine pour la carte Arduino NANO



Remarque :

Il est également possible de réaliser cette platine avec une carte d'essai (plaque avec bandes).

La carte sera sûrement d'une taille un peu plus importante mais plus facile à réaliser (pas de produits chimiques à manipuler,...).



### c. Composants

Réf.	Composant	Fonction / Référence
U\$1	Carte DFPlayer	Ne pas installer
U\$2	Carte Arduino NANO	
DCC	Bornier à vis 2 broches	Entrée signal DCC (si utilisation fonction décodeur DCC)
Alim	Bornier à vis 2 broches	Alimentation platine Arduino
Alimentation	Connecteur 2 broches	Alimentation +5 Volts et GND (option)
HP	Bornier à vis 2 broches	Ne pas installer
DAC	Connecteur 2 broches	Ne pas installer
ESP	Connecteur 4 broches	Option pour liaison port série
I2C	Connecteur 5 broches	Ne pas installer – Bus I2C
SERVO1	Connecteur 3 broches	Servomoteur
SERVO2	Connecteur 3 broches	Servomoteur
SERVO3	Connecteur 3 broches	Servomoteur
S4	Connecteur 3 broches	Servomoteur
S5	Connecteur 3 broches	Servomoteur
S6	Connecteur 3 broches	Servomoteur
INTER1	Connecteur 9 broches	Connecter 6 fis aux interrupteurs à levier + commun au GND
INTER2	Connecteur 8 broches	Ne pas installer
IR	LED	Récepteur Infrarouge (option)
LED1	LED	LED de contrôle de la mise sous tension (option)
OK	Connecteur 2 broches	Optocoupleur (si utilisation fonction décodeur DCC)
OPTION	Connecteur 3 broches	Ne pas installer

Réf.	Composant	Fonction / Référence
R1	Résistance ¼ W	580 Ohms à 1 k
R2	Résistance ¼ W	Ne pas installer
R3	Résistance ¼ W	Ne pas installer
R4	Résistance ¼ W	Ne pas installer
R5	Résistance ¼ W	Ne pas installer
R10	Résistance ¼ W	1 k (si utilisation fonction décodeur DCC)
R11	Résistance ¼ W	10 k (si utilisation fonction décodeur DCC)
R12	Résistance ¼ W	10 k (si utilisation fonction décodeur DCC)
D2	Diode 1N4004	
D4	Diode 1N4148	Si utilisation fonction décodeur DCC

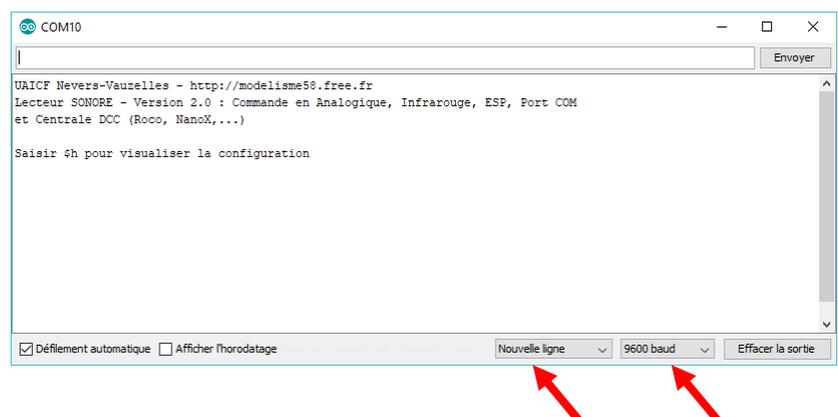
## 6. Paramétrage et commandes

Pour envoyer des commandes via le port Série 0 (via le port USB), il faut ouvrir le Moniteur Série.



Sélectionner la vitesse de transmission des données :  
« **9600 bauds** ».

Dans le menu déroulant, sélectionner « **Nouvelle ligne** » ou « **Retour chariot** » ou « **Les deux, NL et CR** ».



### a. Paramétrage général via le port Série

Pour envoyer les paramètres de configuration, saisir le caractère « \$ » suivi d'une lettre (respecter la police), de la valeur éventuellement puis appuyer sur la touche « Entrée ».

Les paramètres envoyés sont sauvegardés dans la mémoire Eeprom de l'Arduino : les données ne sont pas perdues lors de la mise hors tension de la carte.

Fonctions	Commande à envoyer via le port Série
Envoi de la configuration à l'application par le port COM	<b>\$C</b>
Redémarrage de la carte : RESET	<b>\$R</b>
Type de centrale DCC <i>X = 0 -&gt; Centrale Lenz</i> <i>X = 1 -&gt; Centrale Roco</i>	<b>\$Zx</b> – exemple : \$Z1
Vitesse des servomoteurs <i>X entre 0 et 255</i>	<b>\$Vx</b> – exemple : \$V50
Nombre de servomoteurs <i>X entre 1 et 6</i>	<b>\$Nx</b> – exemple : \$N2

### a. Paramétrage individuels des servomoteurs

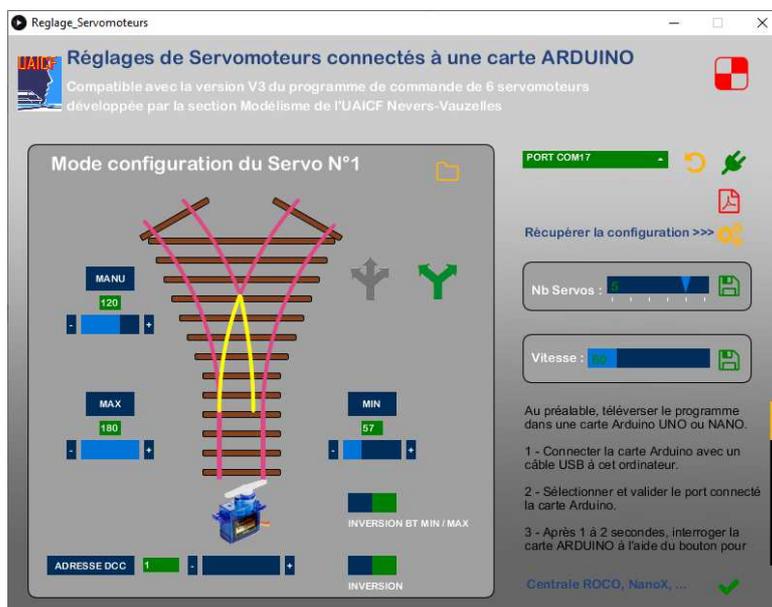
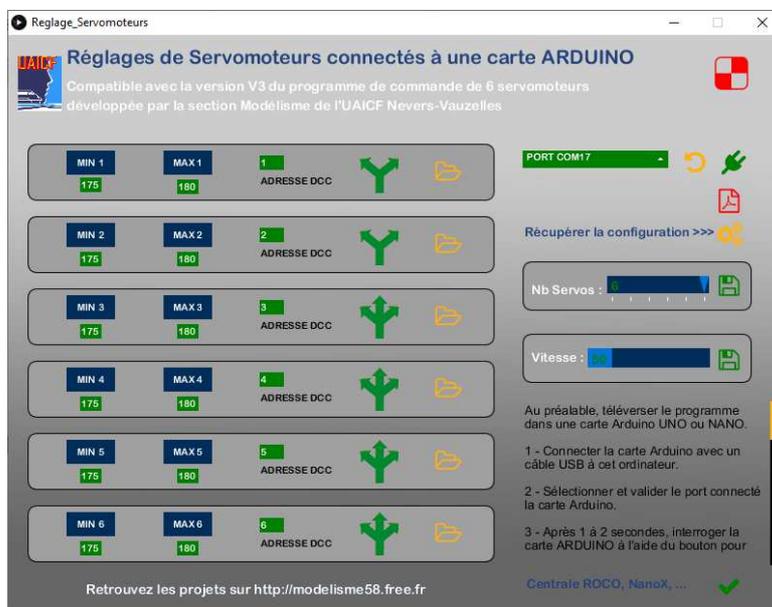
Pour envoyer les paramètres de configuration, saisir le caractère « \$ » suivi d'une lettre (respecter la police), la valeur à configurer et le numéro du servomoteur.

Les valeurs de position des servomoteurs varient entre 0 et 180, ce qui correspond à l'angle de la position du servomoteur.

Fonctions	Commande à envoyer via le port Série
Positionner manuellement un servomoteur <i>X entre 0 et 180</i> <i>Y Numéro servomoteur entre 1 et 6</i>	<b>\$Sxy</b> exemple : \$A601
Positionner un servomoteur sur la position Mini <i>X entre 0 et 180</i> <i>Y Numéro servomoteur entre 1 et 6</i>	<b>\$m xy</b> exemple : \$m301
Positionner un servomoteur sur la position Maxi <i>X entre 0 et 180</i> <i>Y Numéro servomoteur entre 1 et 6</i>	<b>\$M xy</b> exemple : \$M901
Positionner un servomoteur sur position Mini ou Maxi <i>X : 0 ou 1</i> <i>Y Numéro servomoteur entre 1 et 6</i>	<b>\$A xy</b> exemple : \$A11 exemple : \$A01
Type d'aiguillage <i>X = 0 -&gt; aiguillage normal</i> <i>X = 1 -&gt; aiguillage triple</i> <i>Y Numéro servomoteur entre 1 et 6</i>	<b>\$Txy</b> exemple : \$T01 exemple : \$T02
Adresse DCC du servomoteur <i>X entre 0 et 999</i> <i>Y Numéro servomoteur entre 1 et 6</i>	<b>\$Dxy</b> exemple : \$D151

## **b. Paramétrage via une application dédiée**

Pour faciliter la modification des paramètres, la section Modélisme de l'UAICF Nevers-Vauzelles a développé une application gratuite avec le logiciel « Processing ». La carte doit être reliée via le port USB.



L'application et le code source sont téléchargeables sur le site <http://modelisme58.free.fr>

Un tutoriel dédié à l'utilisation de l'interface a été développé.

## 7. Dépannage

Lors de l'installation et de la configuration de ce kit, on peut rencontrer plusieurs problèmes. Vous trouverez ci-après quelques points à vérifier.

### **b. Carte ARDUINO reconnue ?**

Dans le programme « Arduino », il faut veiller à ce que la carte ARDUINO soit reconnue et sélectionner le port « COM » pour programmer ou lire les informations provenant de la carte.

### **c. Débogage à l'aide du Moniteur Série**

L'utilisation du moniteur série apporte de précieuses informations sur le fonctionnement de la carte.

Il faut bien sélectionner la vitesse de communication : « 9600 Baud » comme indiqué dans le code source du programme.

### **d. Erreur lors de la compilation**

Si une erreur se produit lors de la compilation, il peut y avoir plusieurs origines :

- La carte ARDUINO n'est pas sélectionnée : type de carte, port COM ;
- Une erreur dans le code de programmation : il manque une parenthèse, une virgule, un point-virgule,...
- Une ou plusieurs bibliothèques ne sont pas installées ;

Pour vérifier que les bibliothèques sont bien installées, il est conseillé d'ouvrir un programme de démonstration à partir du menu « Fichier ---> Exemples ». Chaque bibliothèque dispose d'exemples et démonstrations. Compiler les programmes de démonstration pour vérifier s'il n'y a pas d'erreurs.

Si une erreur apparaît lors du téléversement d'une carte NANO, il se peut qu'il faille changer le Processeur et utiliser le « Old Boatloader ».

