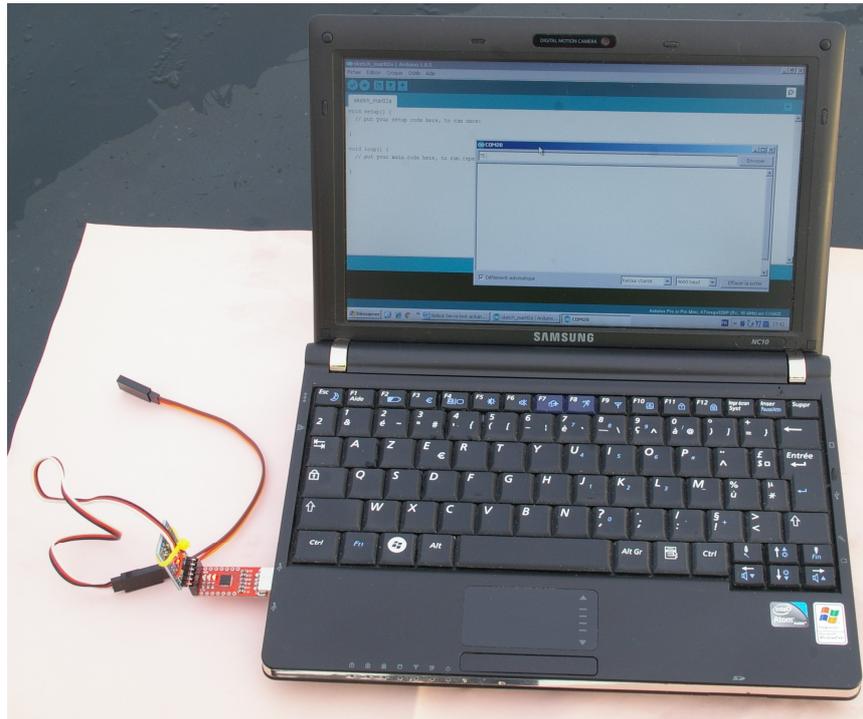


# Testeur de Servo (Servo-test) version "simple" base "Arduino Pro mini"



## 1. Intérêt du servo-test

Le servo-test prend la place du module Vcc pendant les essais (ou du récepteur pour les amateurs de RC) et permet de dégrossir rapidement le réglage de la consigne de vitesse du contrôleur (ou de vérifier le fonctionnement, voir programmer ce dernier).

Une fois le réglage désiré obtenu, il suffit de reporter la valeur par programmation dans le module Vcc.

Comme le nom l'indique, le servo-test permet également de tester le fonctionnement d'un servo RC.

Il existe de nombreux servo-test à prix très faible dans le commerce, mais ces derniers n'offrent pas la précision requise pour programmer avec précision le module Vcc.

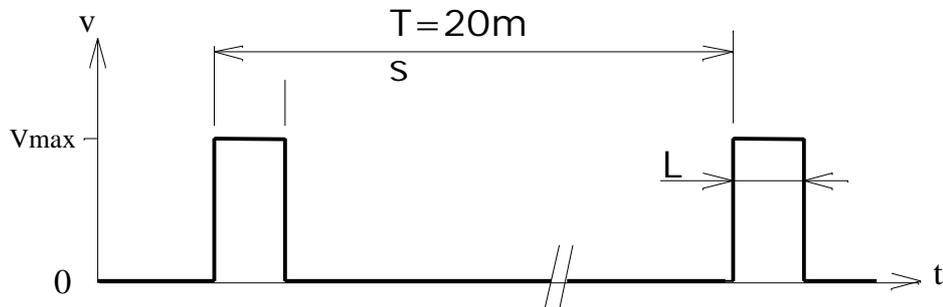
## Caractéristique de cette version :

- réalisation simple et coût très faible : il faut une carte "Arduino Pro-mini", 1 (ou 2) cordon(s) rallonge de servo et quelques soudures.
- Attention : cette simplicité à un prix, cette version n'est pas autonome ; pour fonctionner, elle nécessite un PC, l'environnement de développement (EDI) Arduino installé et la carte d'interface USB/ série (celle permettant la programmation).
- consigne passée à la carte par le PC, via l'EDI Arduino, en % (nombre compris entre 0 et 100)
- précision permise par la carte "Arduino pro-mini", c'est à dire excellente pour notre application.
- alimentation 5V empruntée au PC

## 2. Rappel du principe de la commande d'un contrôleur

En pratique, un contrôleur est branché sur le récepteur RC (ou pour nous sur le module Vcc) par l'intermédiaire d'une prise 3 broches :

- un fil noir ou marron : la masse, reliée au - alim
- un fil rouge : le +5V, fournie par le contrôleur (fonction BEC)
- un fil blanc, jaune ou orange : l'impulsion de commande reçue par le contrôleur



Un contrôleur (appellation impropre, on devrait plutôt parler de variateur) est conçu pour recevoir l'information "vitesse" du récepteur R/C sous la forme "normalisée" d'une largeur d'impulsion :  $L=1ms$  est la valeur mini et correspond à l'arrêt moteur,  $L=2ms$  est la valeur maxi et correspond à la vitesse maximale qu'il est possible d'obtenir pour la tension de l'accu de propulsion utilisé. La période de répétition des impulsions est  $T=20ms$ .

On appelle ici "Consigne de vitesse" cette largeur d'impulsion  $L$  exprimée en % : 0% correspond à une largeur de 1ms (moteur arrêté), 100% à une largeur de 2ms (vitesse de rotation maxi).

Le programme du servo-test, dès que la carte est sous tension, génère une impulsion de largeur 1ms toutes les 20ms, le moteur relié au contrôleur est ainsi à l'arrêt.

## 3. Câblage

### 3.1. Matériel nécessaire

a) Une carte Arduino Pro mini 5V.

**Attention**, cette carte existe en 4 versions : 3,3V et 5V pour les tensions d'alimentation ; 8 et 16Mhz pour les fréquences d'horloges. **Choisir impérativement une carte 5V** (en général, la fréquence est 16MHz avec cette tension).

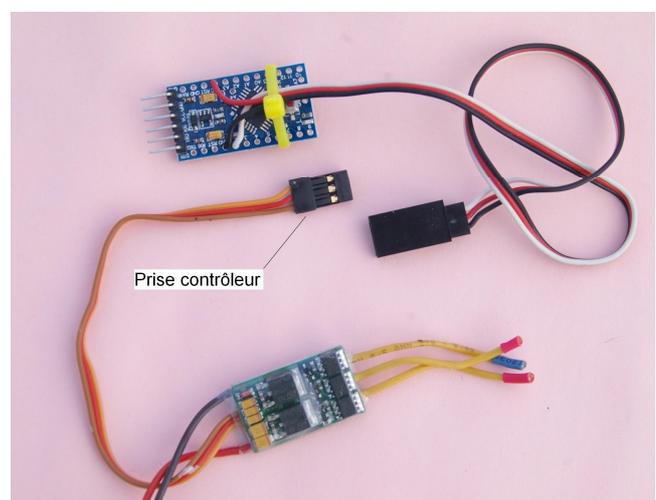
b) Des picots à souder en barrette (x6) coudés à 90° ou droits (souvent livrés avec la carte "Arduino").

c) un cordon 30 à 50cm muni d'une prise type servo RC femelle marque Futaba (à défaut JR)

Rq :

- il est souvent plus économique d'acheter un cordon prolongateur de servo équipé d'une prise mâle à une extrémité et femelle à l'autre, car moins cher, il suffit de couper et d'éliminer la prise inutile !

- Il existe souvent une ambiguïté dans l'appellation mâle ou femelle de la prise. Celle à conserver ici doit pouvoir recevoir la prise du contrôleur (ou d'un servo), cf. photo ci-contre.



d) éventuellement un cordon prolongateur servo Male/femelle Futaba ou JR (cf.3.2. option2)

e) Une carte d'interface pour la programmation de l' "Arduino Pro mini", avec éventuellement le cordon USB adéquat et l'ensemble connecteur et/ou cordon de liaison avec la carte "Arduino Pro-mini".

Si vous n'avez pas déjà une telle carte, et si vous n'êtes pas familier de la programmation "Arduino", voir paragraphes 4 et 5.

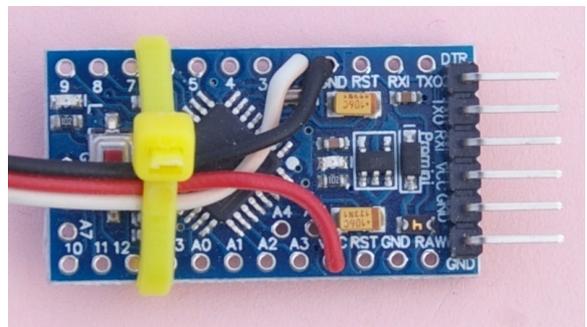
### 3.2. Câblage de la carte "Arduino Pro mini"

**Attention** : utiliser impérativement un fer à souder à panne fine et de la soudure type "électronique"

a) Souder une barrette (coudée ou droite au choix) de 6 picots mâle sur le petit côté de la carte "Arduino Pro mini" (broches repérées BLK, GND, VCC, RXI, TXI, GRN). Placer les picots côté composants et souder côté opposé.

b) Souder le cordon sur la carte Arduino, de la façon suivante :

- souder le fil noir ou marron du cordon sur la broche "GND" de la carte Arduino
- souder le fil blanc, jaune ou orange du cordon sur la broche "2" de la carte Arduino



- **Attention**, en ce qui concerne le fil Rouge (+5V), 2 options sont possibles :

1° option : Le servo-test sera utilisé exclusivement pour faire fonctionner un contrôleur avec BEC

**Ne pas souder le fil rouge du cordon** (le supprimer, ou isoler son extrémité par un petit morceau de gaine thermorétractable)

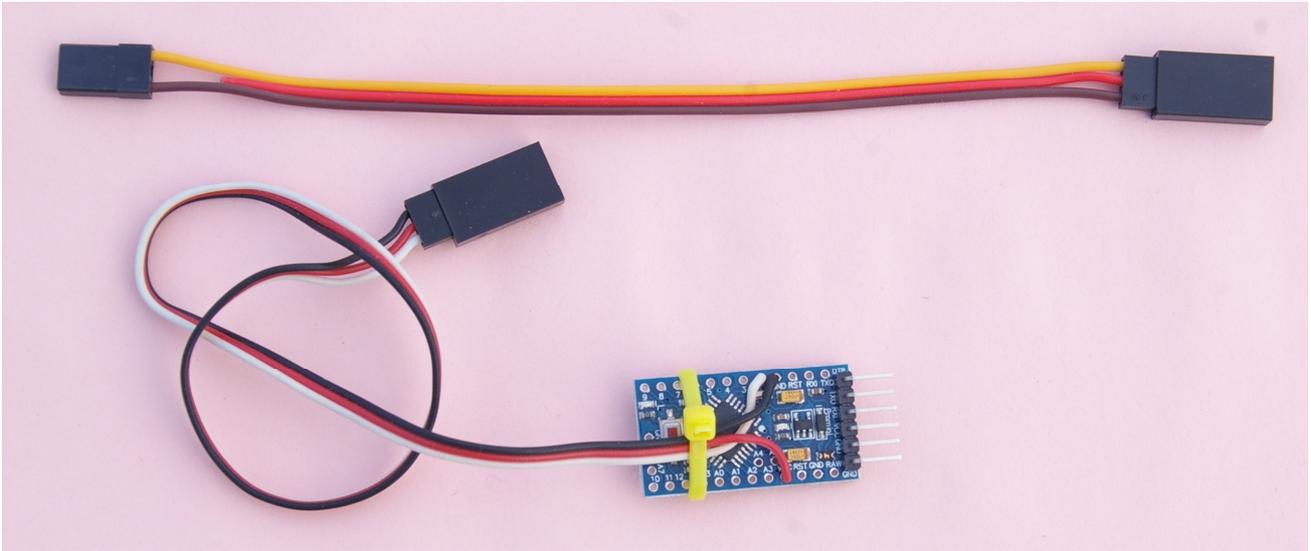
En effet, le contrôleur fournit du +5V s'il possède la fonction BEC (indispensable pour alimenter le module Vcc). Souder le fil rouge sur le +5V de la carte reviendrait à créer un conflit entre le 5V fourni par la carte et celui fourni par le contrôleur.

2° option : le servo-test sera utilisé soit pour tester un servo, soit pour faire fonctionner un contrôleur

Dans ce cas souder le fil rouge sur la broche Vcc de la carte Arduino.

- Pour tester un servo, brancher directement le servo sur le cordon soudé sur la carte Arduino (Rq : le courant maxi susceptible d'être fourni par un port USB est d'environ 0,5A. Il n'y aura pas de problème si le servo est à vide, par contre, si le servo entraîne une charge importante, l'intensité risque d'être insuffisante).

- Pour faire fonctionner un contrôleur muni de la fonction BEC le mieux est alors d'intercaler entre le contrôleur et le cordon soudé sur la carte "Pro-mini" un cordon rallonge de servo Mâle/femelle dont on aura **coupé le fil rouge** (il est facile d'extraire de la prise mâle la broche reliée au fil rouge et de la couper, en soulevant le picot plastique qui retient la broche), voir photo ci-dessous. **Il ne doit pas y avoir de liaison entre le 5V de la carte "Pro-mini" et le contrôleur.**



c) Il est conseillé d'isoler la carte "Pro-mini" par de la gaine thermo rétractable. A défaut, placer un collier pour immobiliser le cordon (cf. photo ci-dessus).

## 4. Carte d'interface "USB-série"

(paragraphe pouvant être sauté par ceux qui sont familiers de la programmation de la "Pro-mini")

Contrairement aux autres cartes Arduino, la "Pro mini" n'est pas équipée d'un connecteur USB autorisant la programmation directe à partir d'un PC. C'est le prix à payer pour avoir une carte très légère ! Bien que l'on puisse arriver à programmer la "Pro mini" de différentes façons, la plus simple est de se procurer une carte spécifique, connue sous le nom de carte d'interface USB-Série, ou "USB to UART breakout", ou Serial Adapter Module USB to TTL", ou carte "FTDi-USB" (cout très modique).

Attention, si cette carte est fournie sans connecteur ou sans cordon de liaison avec la carte "Arduino Pro-mini" ne pas oublier de se procurer l'un et/ou l'autre. Ne pas oublier également un cordon USB-USB pour pouvoir relier cette carte au PC si la carte n'est pas munie d'une prise pouvant être enfichée directement dans le port USB du PC.

### 4.1. Câblage de la carte d'interface

Si la carte d'interface "FTDi" est vendue déjà équipée d'un connecteur femelle 6 broches (la meilleure solution car la plus fiable), ou d'un connecteur mâle + cordon femelle-femelle, il n'y a rien de plus à faire ici !

Dans le cas contraire le plus simple est de souder une barrette de 6 connecteurs femelle. Placer la barrette côté composants et souder côté cuivre.

### 4.2. Liaison carte d'interface "FTDi" et carte "Arduino Pro-mini"

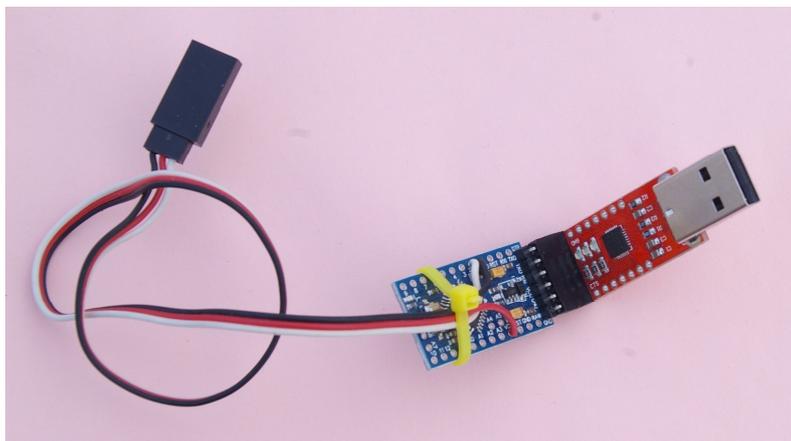
**Attention** : on rencontre plusieurs types de cartes "Arduino Pro-mini" : celles d'origine, fabriquées par "Sparkfun" et des clones d'origines diverses. **Ces cartes n'ont pas toutes le même brochage pour le connecteur d'interface** : bizarrement, les broches sont inversées sur certaines cartes et nécessitent de bien vérifier le sens lors de la liaison avec la carte d'interface.

Relier les deux cartes en veillant à bien respecter le brochage suivant (Les broches sont en général clairement repérées sur les circuits imprimés des 2 cartes) :

Broche carte d'interface FTDi	Broche carte Arduino Pro-mini	Remarque
GND	BLK (Ou GND)	
CTS	GND	(1)
VCC	VCC	(2)
TX (ou TXD)	RXI	
RX (ou RXD)	TX0	
DTR	GRN (ou DTR)	

(1) Si liaison par cordon souple 5 fils, cette liaison n'est pas nécessaire

(2) bien veiller à la position de cette liaison, si elle est bonne, le reste doit l'être !



## **5. Installation logicielle et test**

(paragraphe pouvant être sauté par ceux qui sont familiers de la programmation de la "Pro-mini")

- Installer l'environnement de développement "Arduino" (IDE) disponible gratuitement en téléchargement sur le site du même nom.
  - ouvrir l'environnement "Arduino"
  - Pour tester la carte, ouvrir le fichier programme suivant (Croquis en français ou Sketch en terminologie "Arduino", on se demande bien pourquoi !) : Menu "Fichier", "Exemples", "0.1.Basics", "Blink"
  - Sélectionner le bon type de carte : Menu "Outils", "Type de carte", sélectionner "Arduino Pro or Pro Mini"
  - Vérifier que dans le menu "Outils" le processeur correspond bien à celui de la carte utilisée (en général aujourd'hui : ATmega 328 (5V, 16MHz)
  - Vérifier également dans le menu "Outils" que le programmeur utilisé est bien "AVRISP mkII"
  - relier une carte "Arduino Pro-mini" et la carte d'interface USB comme indiqué en 4.
  - brancher le connecteur USB de la carte d'interface dans un port USB du PC, soit directement, soit à l'aide du cordon USB adéquat (il est conseillé de toujours utiliser le même). En principe, les pilotes s'installent tous seuls (cela peut prendre pas mal de temps).
  - Sélectionner le port COM utilisé par la liaison USB : Menu "Outils", "Port" puis cliquer sur le port utilisé. Ce dernier n'apparaît pas toujours dans le menu "Outils". Si on a du mal à identifier le port COM, ouvrir le panneau de configuration de Windows, "Périphériques et imprimantes" : le périphérique de la carte d'interface apparaît avec le port COM attribué.
  - Téléverser le programme "Blink" (icône "flèche" en haut à gauche, ou menu "Croquis", "Téléverser")
  - Si tout se passe bien, au bout de quelques instants, la barre de progression en bas à droite arrive à son extrémité et la led de la carte Arduino Pro-mini clignote (1s allumée, 1s éteinte).
- Le fonctionnement de l'ensemble est correct : l'IDE et la carte sont prêts à recevoir le programme du servo-test.

Rq : Si du texte orange apparaît dans l'IDE "Arduino", c'est que quelque chose ne va pas : bien vérifier le câblage. Consulter les nombreux forums relatifs à Arduino si les pilotes n'arrivent pas à s'installer correctement (installation problématique avec certaines versions de Windows : Xp, 8).

## **6. Programmation de l' "Arduino Pro-mini"**

(cette programmation n'est à effectuer qu'une seule fois avant mise en service de la carte servo-test)

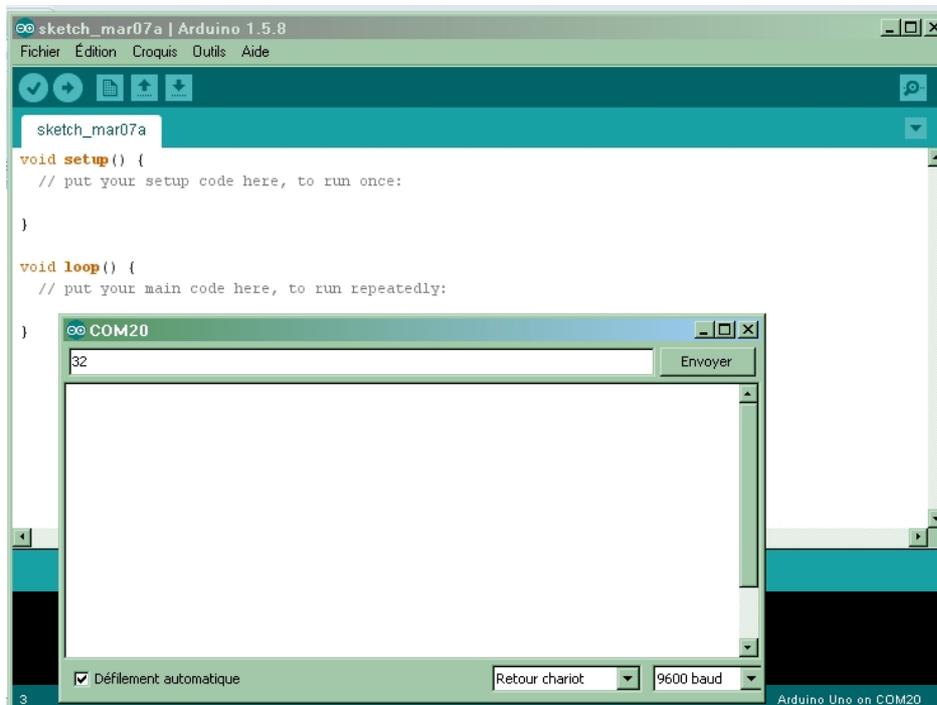
- Ouvrir l'environnement de développement Arduino et ouvrir le programme "Vcc\_servo-test\_s\_vx.ino", où x est la version de programme (menu "Fichier", "Ouvrir"). Le fichier peut se trouver n'importe où sur l'ordinateur mais attention, seule restriction en "Arduino" : ce fichier doit être impérativement placé dans un répertoire du même nom : " Vcc\_servo-test\_s\_vx"
- Vérifier que la configuration de l'IDE est correcte : carte et processeur utilisés, programmeur, port COM (cf. paragraphe 5 pour plus de détails et les menus à utiliser) et brancher la carte d'interface USB-Série dans le port USB du PC (directement ou par l'intermédiaire d'un cordon USB), si ce n'est déjà fait (même port que celui utilisé lors du test de la carte, cf. paragraphe 4.).
- Téléverser le programme (icône 'flèche' en haut à gauche, ou menu "Croquis", "Téléverser")
- Si tout se passe bien, la barre d'avancement en bas à droite arrive jusqu'à son extrémité sans qu'aucun message d'erreur n'apparaisse (en orange)
- Laisser brancher l'ensemble et laisser tourner le programme Arduino pour pouvoir vérifier le fonctionnement du servo-test (cf. paragraphe suivant).

## 7. Utilisation du servo-test

**SECURITE** : lors de l'utilisation du servo-test, penser à immobiliser l'avion et attention à l'hélice !

Pour la mise en œuvre, procéder de la façon suivante :

- lancer si ce n'est déjà fait, l'environnement de développement Arduino (EDI) sur le PC ; il n'est pas nécessaire de charger le programme, l'écran vierge (nouveau programme) convient parfaitement
- relier si ce n'est déjà fait la carte d'interface à la carte Arduino en respectant le brochage (cf. paragraphe 4)
- relier l'ensemble carte d'interface/carte Arduino Pro-mini au port habituel du PC
- vérifier la configuration de l'EDI en ce qui concerne le port COM utilisé (cf. paragraphe 5).
- brancher la prise 3 broches du contrôleur dans la prise femelle du cordon (attention au sens si pas de détrompeur : le fil noir ou marron du contrôleur doit être relié au fil noir ou marron du cordon de la carte arduino). **Rappel** : si le fil rouge du cordon (+5V) est soudé sur la carte Arduino, il est indispensable d'intercaler une rallonge sans fil rouge pour un contrôleur muni de la fonction BEC, cf. 3.2.
- brancher immédiatement l'accu de propulsion.  
Par défaut, le servo-test génère un créneau de largeur 1ms (0%), on doit entendre le contrôleur s'initialiser et le moteur doit être arrêté.
- ouvrir dans l'EDI la fenêtre du moniteur série en cliquant sur l'icône "loupe" situé en haut et à droite de l'EDI (le texte "Moniteur série" s'affiche quand on passe la souris dessus)
- vérifier la configuration du moniteur série en bas et à droite : vitesse à "9600 baud" et caractère de terminaison sur "Retour chariot"
- entrer dans la fenêtre la consigne de vitesse désirée en % (exemple ici : 32) et cliquer sur "Envoyer" (ou appuyer sur la touche "Entrée" du PC) : le moteur doit démarrer.  
Rq : à chaque envoi d'une consigne, la led de la carte Arduino émet 3 brefs clignotements.



- Pour mettre hors-tension :
  - arrêter le moteur en envoyant la consigne "0"
  - débrancher l'accu de propulsion puis immédiatement la prise 3 broches du contrôleur du cordon de la carte arduino. Débrancher la prise USB de la carte d'interface

Bons réglages.

