**POUR ALLER + LOIN (FACULTATIF)**

Puisque vous n'êtes pas aussi fainéant(e)s qu'il n'y parait, nous allons appliquer le savoir nouvellement acquis, en réalisant un exemple d'application concrète.

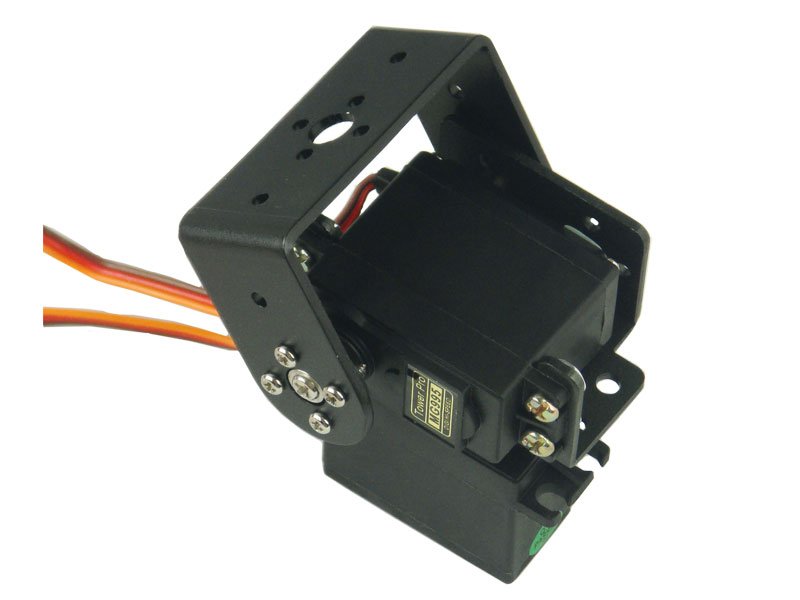
Seulement voilà, un seul moteur qui bat la mesure, c'est vite barbant. Que diriez-vous d'une machine qui servirait votre boisson favorite, sans broncher ? Que diriez-vous de faire votre premier pas en robotique sans avoir l'air d'y toucher ?

**...VOICI VENIR LE BOISSONMATIC' !**

Vous qui rêvez d'une invention géniale et susceptible de révolutionner l'univers, j'ai ce qu'il vous faut !

Le **BoissonMatic'** est une réalisation totalement absurde et c'est bien ce qui la rend indispensable. Elle nous servira ici d'illustration de principe pour un montage fort utile : le **Pan Tilt**, ou [Monture azimutale](http://fr.wikipedia.org/wiki/Monture_azimutale) dans notre bonne vieille langue de Molière.

Le principe est simple et consiste à monter un servo sur l'autre, perpendiculairement. Pendant que le premier moteur tourne à l'horizontale (**pan**oramique), le second tourne à la vertical (azimut).

****

Avec des servomoteurs basiques, les déplacements de chacun des axes seront limités à 180°. Ceci étant, vous pouvez facilement trouver un servo tournant à 360° dans n'importe quelle boutique de modèle réduit, ou encore [modifier un servo pour une rotation continue](http://www.equinoxefr.org/post/2012/11/03/modifier-des-servos-mg995-pour-une-rotation-continue/).

**CODE AGAIN**

// BoissonMatic' 2014  
// john@labfab.fr sous licence WTFPL v2.0   
  
  
#include <Servo.h>

Servo servoBas; // création d'un premier objet servo, pour le Pan

Servo servoHaut; // création du second, pour le tilt

int pos = 0; // variable pour stocker une position

int const MINPAN = 0; // valeur mini du Pan, en degré.

int const MAXPAN = MINPAN + 90; // Un quart de tour panoramique

int const MINTILT = 180; // valeur mini du Tilt (en cas de servo monté "à l'envers" ! :] )

int const MAXTILT = MINTILT - 90; // avec un max au quart de tour.

void **setup**()

{

  servoBas.attach(9); // attache le servo du bas à la broche 9

  servoHaut.attach(10); // celui du haut à la broche 10

  // Mise en place de la machine en position de départ

  servoBas.write(MINPAN);

  servoHaut.write(MINTILT);

  delay(1000); // admettant qu'il faille une seconde pour faire demi-tour

               // à un servo bien nourri.

}

void **loop**()

{

  // on fait faire un quart de tour au servo panoramique

  servoBas.write(MAXPAN);

  // La boucle "for" permet d'adoucir le déplacement du moteur

  for(pos = 180; pos>=70; pos-=1) // On dépasse le quart de tour,

                                  // pour que le versement du liquide soit plus rapide

  {

    servoHaut.write(pos);

    // le servo tourne de la valeur de l'angle indiquée par la variable 'pos'

    delay(20);

    // on laisse un peu de temp au servo pour arriver à la position demandée

  }

  delay(3000); // encore du temps pour qu'assez de liquide coule du récipient.

  for(pos = 70; pos<=180; pos+=1) // la même dans l'autre sens

  {

    servoHaut.write(pos);

    delay(25);

  }

  servoBas.write(MINPAN ); // retour en position initiale

  delay(400);

  servoHaut.detach(); // on détache les moteurs un par un.

  servoBas.detach();

  // la séquence n'aura lieu donc physiquement lieu qu'une seule fois,

  // même si la boucle principale loop() continue à tourner.

  servoHaut.attached();   
 // devrait retourner quelque chose  
 // comme "false" sur votre moniteur serie.

}

Vous l'aurez compris, la boucle for est fort utile. C'est même la base du fonctionnement d'Arduino, vous ne pouvez pas *looper* ça. :)

L'instruction for est utilisée pour répéter l'exécution d'un bloc d'instructions regroupées entre des accolades. Un compteur incrémental est habituellement utilisé pour incrémenter et finir la boucle. L'instruction for est très utile pour toutes les opérations répétitives et est souvent utilisées en association avec des tableaux de variables pour agir sur un ensemble de données ou broches.

Il y a 3 parties dans l'entête d'une boucle for :

for (initialization; condition; incrementation) {  
//instruction(s)à exécuter;

}

L'initialisation a lieu en premier et une seule fois. À chaque exécution de la boucle, la condition est testée; si elle est VRAIE, le bloc d'instructions et l'incrémentation sont exécutés. Puis la condition est testée de nouveau. Lorsque la condition devient FAUSSE, la boucle stoppe.

Ce qui se trouve entre les {} s'appelle un **bloc d'instruction**. Les variables y ont un comportement spécifique que nous aborderons une autre fois si vous le voulez bien !  
  
Vous allez devoir apprendre à la boucler, car vous utiliserez cette **structure de contrôle** pour de nombreux propos.  
Imaginez par exemple une multitude de servomoteurs que vous voudriez "détacher". En plaçant vos objets dans un tableau, vous appliquerez un bon principe de fainéantise. Il ne vous restera plus qu'à parcourir ce tableau d'un for, et d'appliquer pour chaque tour de boucle secondaire, la méthode detach sur l'objet.

Si vous vous êtes amusé(e)s dans cette modeste découverte d'une bibliothèque standard particulièrement adaptée aux servos, adoptez le bon réflexe et sachez que pour chacun de vos projets, il en existe certainement une qui fera votre affaire.

Vous n'aurez ainsi pas à résoudre des problèmes déjà réglés, ni à re-coder ce qui l'a déjà été mille fois. Vous pourrez ainsi vous concentrer sur votre problématique et pas sur le code qui gère "la tuyauterie" sous-jacente. N'hésitez pas à parcourir les bibliothèques existantes, le soir au coin du feu. Vous apprendrez beaucoup, et gagnerez au final un temps précieux.

**RESSOURCES**

* [fichier source](https://www.fun-mooc.fr/asset-v1:MinesTelecom+04017+session09+type@asset+block/boiteBoissonMatic0mm.dxf) au format .dxf pour découpe laser du socle BoissonMatic'.
* des [exemples de chassis](http://www.thingiverse.com/search/page:1?q=pan+tilt) sur thingiverse.

Pour celles et ceux qui n'auraient pas un second servomoteur à disposition, ni un lab dans les parages, vous pouvez toujours suivre le TP général qui n'utilise qu'un seul servo. Vous améliorerez votre feu bicolore en y ajoutant une barrière. Ceinture et bretelles !