



mBlock / Arduino  
Ph. Truillet

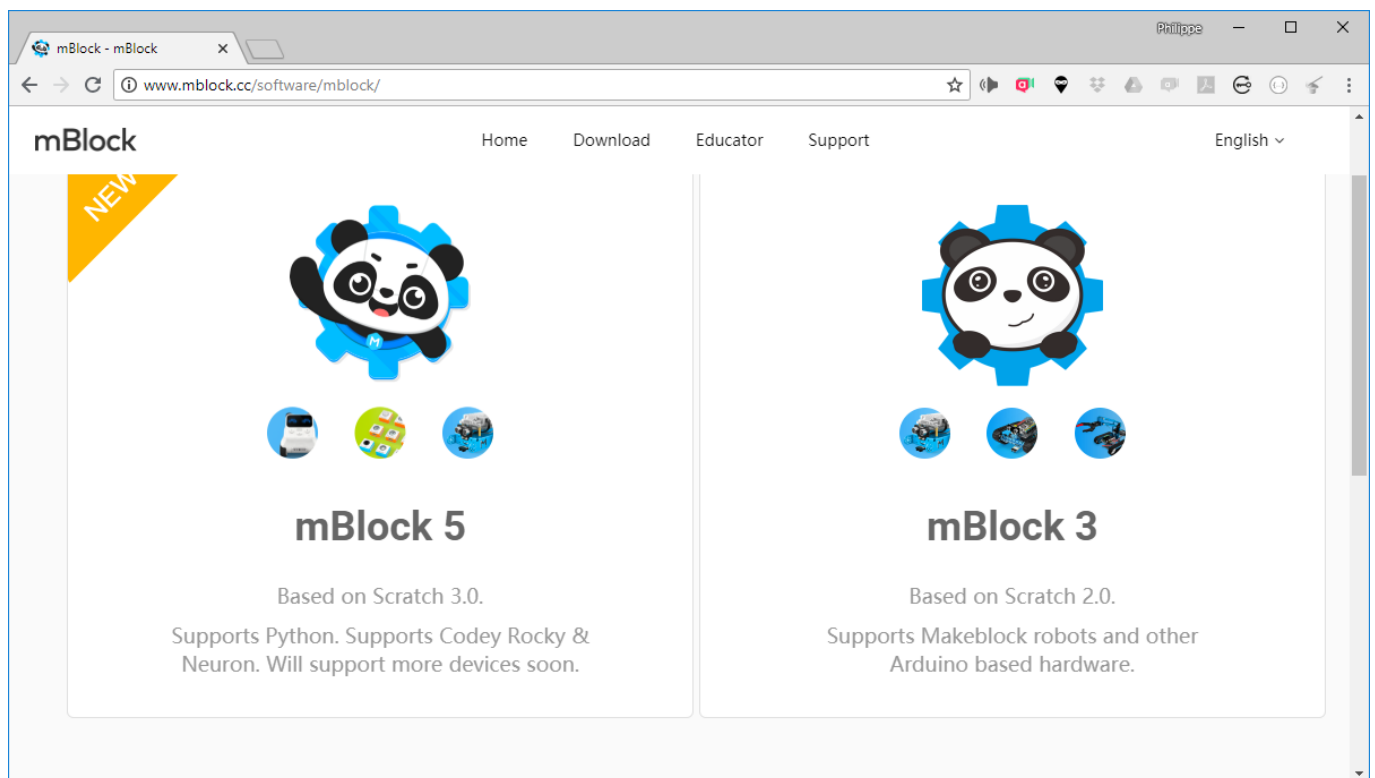


Février 2019 – v. 1.2



## 0. déroulement de la séance

- Avant de commencer, il faut installer sur le poste de travail l'environnement de travail (IDE) mBlock (A télécharger ici : <http://www.mblock.cc/software/mblock>).



**Figure 1 - Téléchargement de l'Environnement mBlock (version v. 3 préférée)**

Une fois téléchargé, cliquez sur l'exécutable (Ex : **mBlock\_win\_V3.14.11.exe** et laissez-vous guider). Une icône devrait apparaître sur le bureau

**Nous pouvons commencer à travailler 😊.**

## 1. introduction

### 1.1 Histoire d'arduino

(cf. The Making of Arduino - <http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino/0>)

C'est à Ivrea en Italie, terres du roi Arduin (Arduino en italien) vers l'an mil que commence l'histoire de cette plateforme électronique. Créé en 2005 à partir des projets Wiring et plus tôt Processing.org -

<https://www.processing.org> comme outil pour les étudiants de l'Interaction Design Institute d'Ivrea, Arduino est devenu en moins de 8 ans le projet de plus influent de l'électronique moderne.

Sous licence **Creative Commons** (les plans sont libres), arduino peut permettre d'effectuer des tâches extrêmement diversifiées comme des tâches domotique ou robotique.

Il existe de nombreux matériels compatibles Arduino comme Freeduino, (<http://www.freeduino.org>) ou Educaduino ([educaduino.fr](http://educaduino.fr)), NodeMCU, etc.

## 1.2 Environnement mBlock

L'Environnement de Développement mBlock (IDE) permet de préparer ses programmes et les « exporter » vers la carte Arduino (cf. Figure 2).

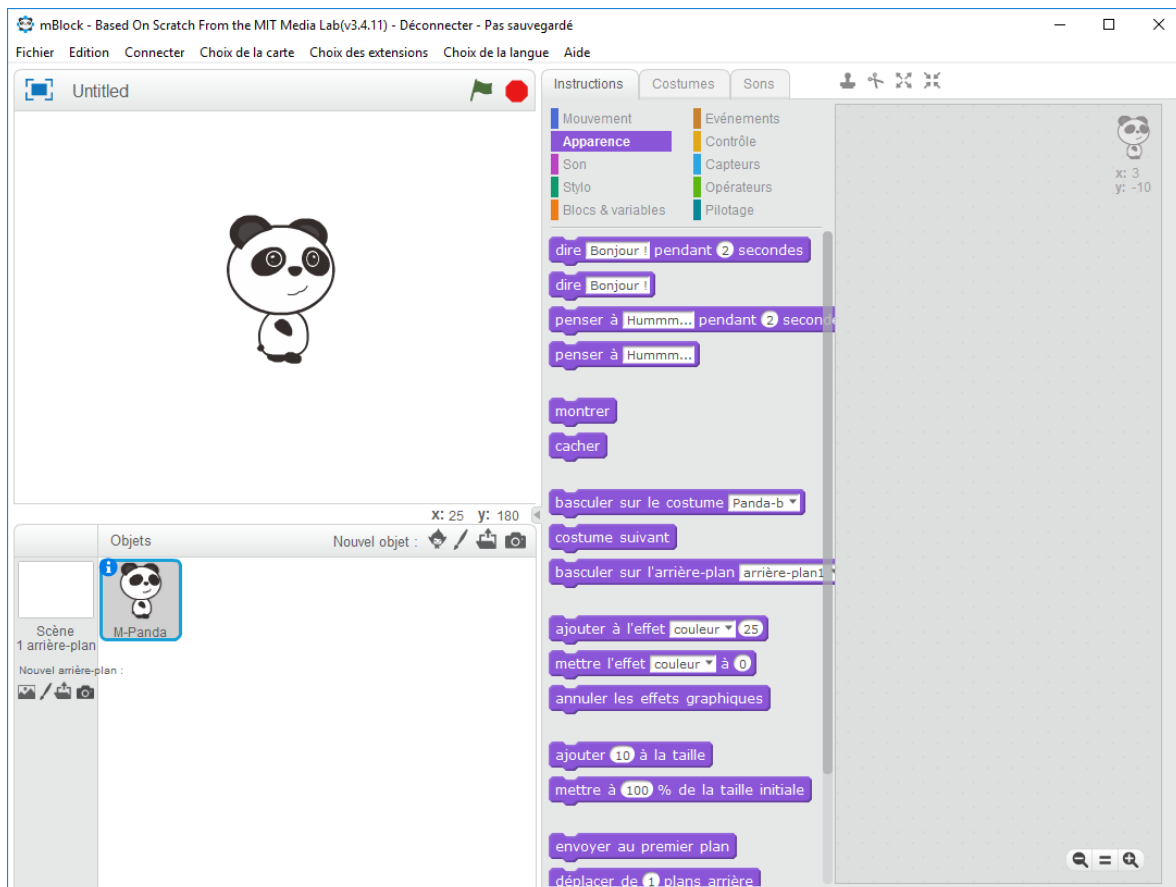
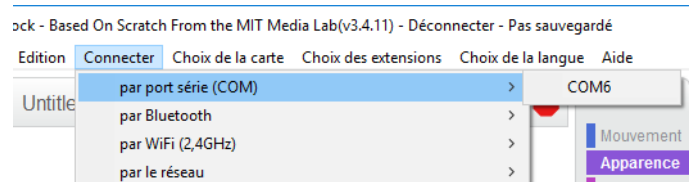
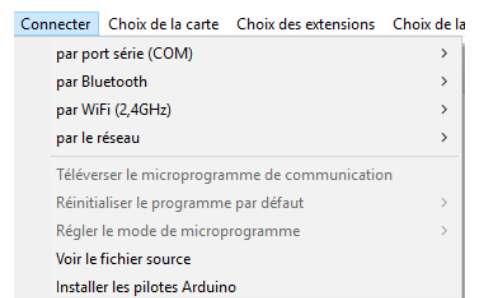


Figure 2 - Vue de l'environnement de programmation mBlock

### A ne faire qu'une fois →

- Dans le premier temps (si vous le souhaitez), configurez mBlock en Français en utilisant le menu « **Choix de la langue | Français** »
- Installez ensuite les pilotes arduino e utilisant le menu « **Connecter | Installer les pilotes Arduino** »
- Branchez votre carte arduino (par exemple, Arduino Uno)
- Dans le menu « **Choix de la carte** », cochez « **Arduino Uno** »
- Et cochez enfin dans le menu '**Connecter | par port série (COM)** », le port défini par la carte



Si tout se passe bien, le point rouge de l'onglet *instructions* passe au vert (cf. Figure 3)



Figure 3 - carte arduino connectée

## 2. mon premier programme !

Passez maintenant dans le monde arduino en utilisant le menu « **Edition | Arduino** »

Les blocs mBlock spécifiques arduino sont dans l'onglet *instructions* sous le sous-menu *pilotage* mais les autres onglets sont disponibles.

Créer le programme suivant (cf. Figure 4) en déposant les blocs dans la zone de travail.



Figure 4 – programme « blink »

Le programme consiste à allumer et éteindre successivement (à l'infini !) une led branchée sur le port 13 de la carte arduino (ce port a la particularité de posséder en interne une résistance de 220 ohm).

Branchez une led sur le port 13 (patte la plus longue « + ») et GND (masse) pour la patte la plus courte.

Transférez maintenant le programme vers la carte en appuyant sur le bouton

Téléverser dans l'Arduino

Si tout se passe bien, vous devriez avoir cette page :

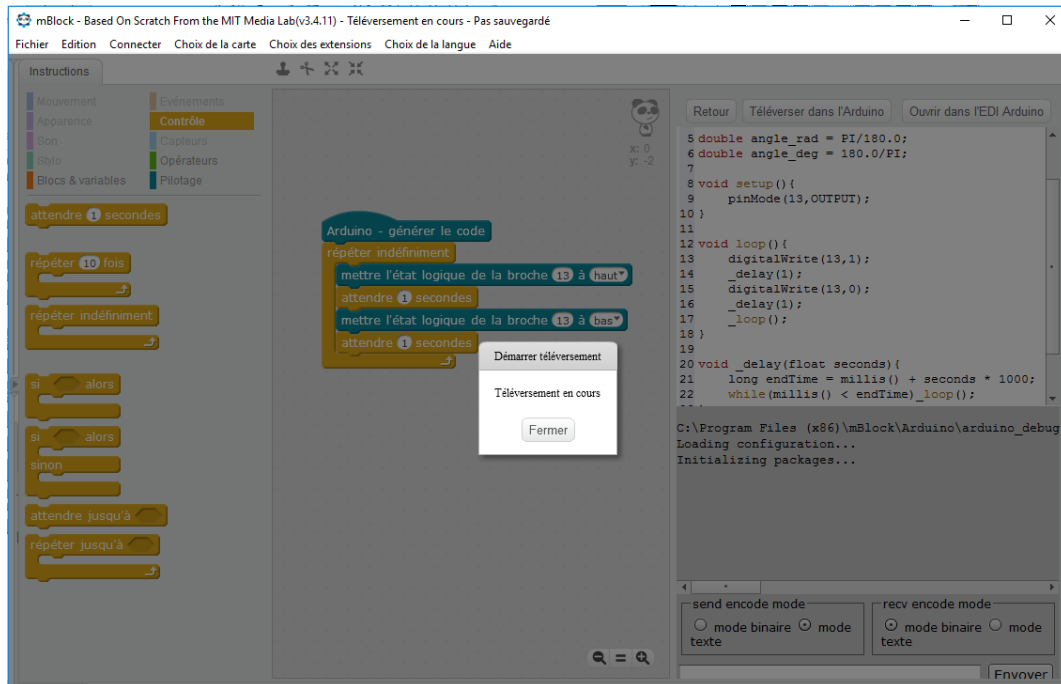


Figure 5 – téléversement en cours

La led devrait s'allumer puis s'éteindre toutes les secondes 😊

Ce rapide aperçu laisse entrevoir de très nombreuses possibilités simples à mettre en œuvre : contrôle de capteurs, d'effecteurs (led, buzzer, moteurs, ...), contrôle à distance sans fil, etc.

Pour aller plus loin, le mieux reste de lire et de partager ses expériences ! Il existe de très nombreux sites accessibles via votre moteur de recherche préféré

### 3. on continue ...

#### 3.1 Un moteur pour bouger

Nous allons maintenant brancher un servo-moteur sur la carte.

Le fil rouge va aller sur la broche « 5V », le marron sur le GND (masse) et le jaune/orangé sur la broche D8 par exemple. Nous allons envoyer des « trains d'impulsions électriques » à durées et intervalles (périodes) précis. Créez le programme suivant (cf. Figure 6 et faites comme pour le premier exercice).

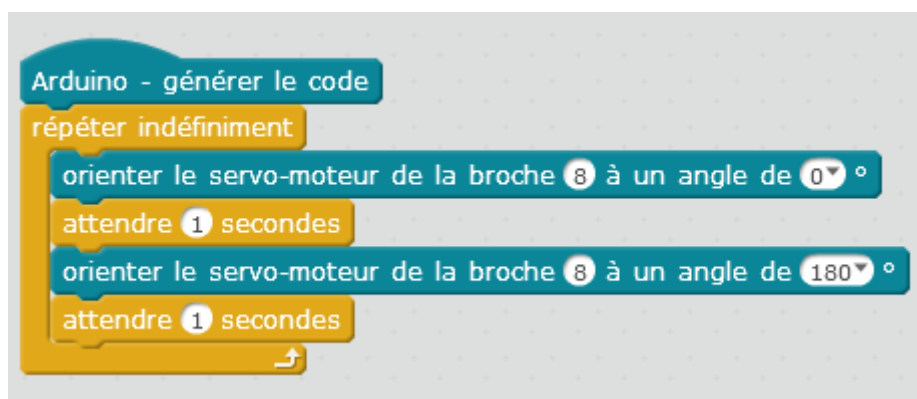


Figure 6 – programme « servo »

### 3.2 une led comme capteur de lumière

Nous allons utiliser une led infrarouge pour allumer un éteindre une led branchée sur le pin 13.

Branchez la led infrarouge sur l'entrée analogique A0 et GND.

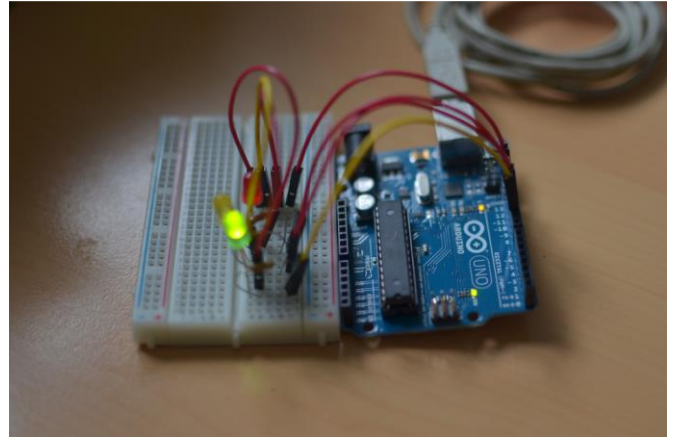
Ecrire un programme qui lit une valeur sur l'entrée A0, allume la led « 13 » si une valeur de seuil (exemple 100) est dépassée et fait tourner un servo-moteur afin d'indiquer la valeur captée.

### 3.3 Encore plus loin ... les feux tricolores

Un feu tricolore peut être dans l'état rouge, orange, vert. Dans le cas classique, le feu est successivement vert, puis orange, puis rouge, puis de nouveau vert, etc.

Dans ce cas, il reste 6 secondes au vert, 1 seconde à l'orange, et 7 secondes au rouge (histoire de ne pas attendre trop longtemps 😊)

Déterminez le montage électronique (**Attention** : pensez à utiliser une résistance de 220 ohm si vous utilisez un autre pin que le pin 13), codez et testez !



**Nota** : vous pouvez utiliser le logiciel Fritzing (<http://fritzing.org/download>) pour dessiner et sauvegarder votre montage !