

## Présentation de mBlock

Le logiciel mBlock intègre l'environnement de Scratch tout en ajoutant la possibilité de piloter du matériel tel que les produits de la gamme Arduino™.

Tout ce qui est possible avec Scratch est réalisable avec mBlock.

mBlock intègre la gestion de connexion à une carte de la gamme Arduino™.

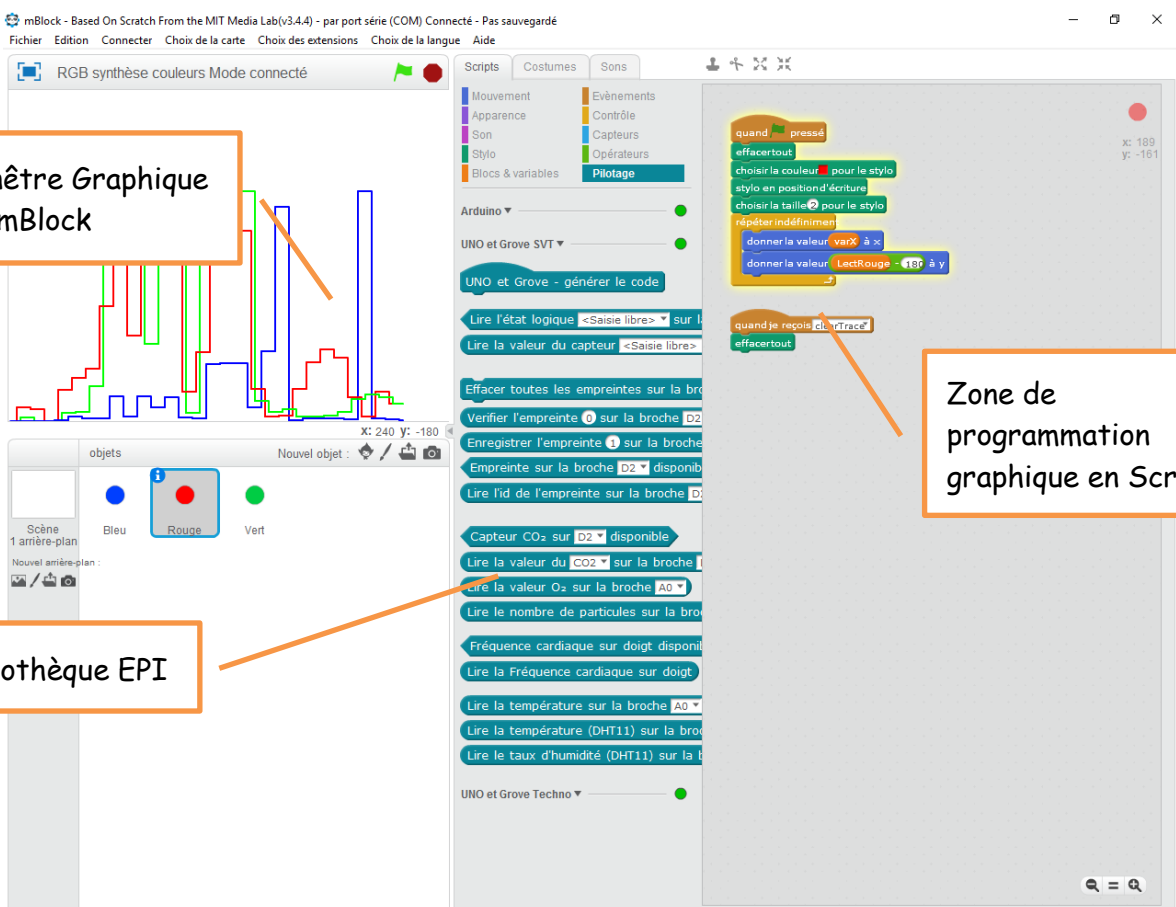
Jeulin a développé une bibliothèque pour les **EPI** afin de rendre compatible mBlock avec les capteurs et actionneurs de la gamme Grove pour faire de mBlock "la solution de programmation" pour la gamme Arduino™.

## Deux modes de fonctionnement sont alors possibles.

- En mode « Connecté » : il faudra au préalable télécharger un firmware dans la carte Arduino afin de permettre le dialogue entre la carte et mBlock sur votre ordinateur.

Ce mode permet :

- ⇒ D'accéder en direct à l'ensemble des entrées / sorties de la carte Arduino™ et/ou Grove
- ⇒ De créer et déboguer votre programme
- ⇒ D'afficher les valeurs mesurées dans la fenêtre graphique de mBlock



The screenshot shows the mBlock software interface in 'Connected Mode'. The interface is divided into several sections:

- Top Bar:** Displays 'mBlock - Based On Scratch From the MIT Media Lab (v3.4.4) - par port série (COM) Connecté - Pas sauvegardé'. It includes menus for 'Fichier', 'Edition', 'Connecter', 'Choix de la carte', 'Choix des extensions', 'Choix de la langue', and 'Aide'.
- Left Panel:** Contains a 'RGB synthèse couleurs Mode connecté' window showing a color spectrum graph. Below it is the 'objets' panel with 'Scène' and 'Nouvel objet' buttons, and a 'Nouvel arrière-plan' section.
- Center Panel:** The 'Scripts' panel, which lists various categories like 'Mouvement', 'Apparence', 'Son', 'Style', 'Blocs & variables', 'Evénements', 'Contrôle', 'Capteurs', 'Opérateurs', and 'Pilotage'. It also shows a list of available sensors and actuators for the connected Arduino board, such as 'Capteur CO2', 'Fréquence cardiaque', 'Température', and 'Humidité'.
- Right Panel:** The 'Zone de programmation graphique en Scratch', showing a Scratch-style script area with blocks for 'quand pressé', 'effacer tout', 'choisir la couleur', 'stylo en position d'écriture', 'choisir la taille', 'répéter indéfiniment', 'donner la valeur', and 'LectRouge'.

Annotations in the image point to specific features:

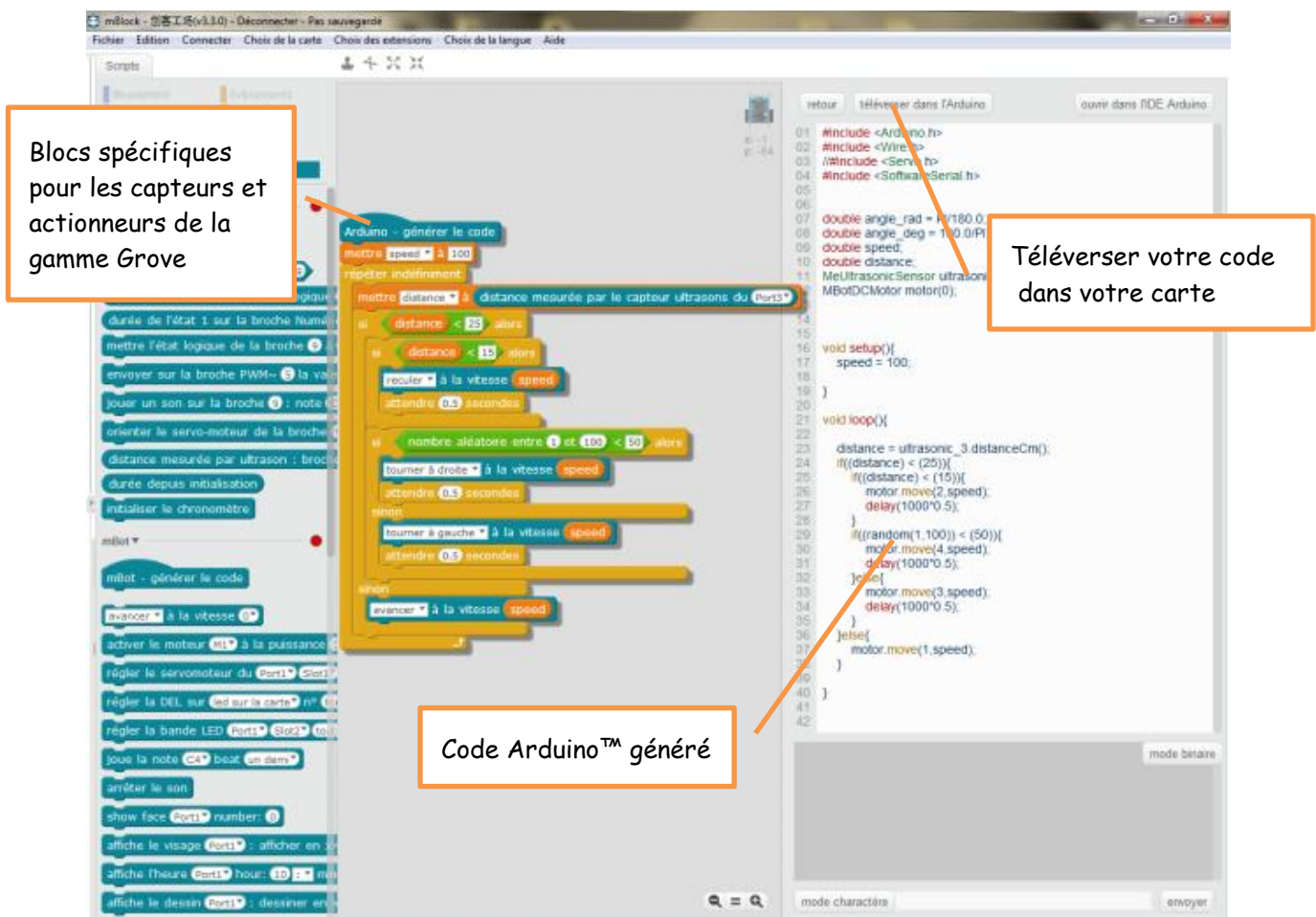
- Fenêtre Graphique de mBlock:** Points to the RGB synthesis window on the left.
- Bibliothèque EPI:** Points to the list of sensors and actuators in the center panel.
- Zone de programmation graphique en Scratch:** Points to the script area on the right.

Environnement de programmation graphique en mode connecté

- En mode « Déconnecté » « Automate » : votre programme sera téléchargé dans la carte Arduino™, l'ordinateur ne sera plus nécessaire.

Ce mode permet :

- ⇒ De générer du code pour votre carte Arduino™
- ⇒ Une autonomie totale de votre TP
- ⇒ La possibilité de créer et personnaliser un capteur et/ou appareil de mesure autonome
- ⇒ De comprendre l'étalonnage d'un capteur, de convertir le signal pour obtenir la bonne unité de mesure



**Blocs spécifiques pour les capteurs et actionneurs de la gamme Grove**

**Téléverser votre code dans votre carte**

**Code Arduino™ généré**

```


01 #include <Arduino.h>
02 #include <Wire.h>
03 #include <Servo.h>
04 #include <SoftwareSerial.h>
05
06
07 double angle_rad = PI/180.0;
08 double angle_deg = 180.0/PI;
09 double speed;
10 double distance;
11 MeUltrasonicSensor ultrasonic;
12 MBotDCMotor motor(0);
13
14
15
16 void setup(){
17   speed = 100;
18 }
19
20
21 void loop(){
22
23   distance = ultrasonic_3.distanceCm();
24   if((distance) < (25)){
25     if((distance) < (15)){
26       motor.move(2, speed);
27       delay(1000*0.5);
28     }
29     if((random(1,100)) < (50)){
30       motor.move(4, speed);
31       delay(1000*0.5);
32     }
33     else{
34       motor.move(3, speed);
35       delay(1000*0.5);
36     }
37   }
38   else{
39     motor.move(1, speed);
40   }
41 }
42
  
```

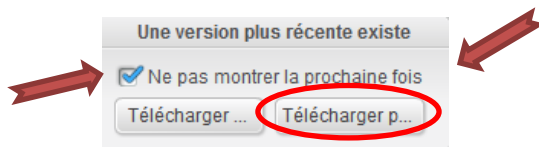
Environnement de programmation graphique en mode téléversé

## I) Installation

Pour le bon fonctionnement du logiciel, nous vous conseillons de suivre les instructions ci-dessous, qui sont à réaliser une seule fois par poste de travail.

Télécharger le logiciel mBlock sur <http://www.mblock.cc/> et l'installer sur un poste informatique.

 A la première exécution du logiciel mBlock, il est recommandé lorsque cette fenêtre apparaît de cocher « Ne pas montrer la prochaine fois » et de cliquer «Télécharger plus tard ».



Sur le site [www.jeulin.fr](http://www.jeulin.fr), dans les références des valises EPI, télécharger et décompresser le contenu du fichier nommé « Extension\_mBlock\_EPI\_Jeulin.zip»

Ce zip est composé de 2 fichiers.

- Un fichier **EPI\_Jeulin.zip** qui contient notre extension pour mBlock.
- Un fichier **uno.hex** qui est le firmware compatible avec cette extension.


### *Pour uno.hex*

En mode administrateur, placer ce fichier dans le répertoire d'installation de mBlock, par défaut dans :

**C:\Program Files (x86)\mBlock\tools\hex**

Si le fichier uno.hex existe déjà, valider le remplacement.

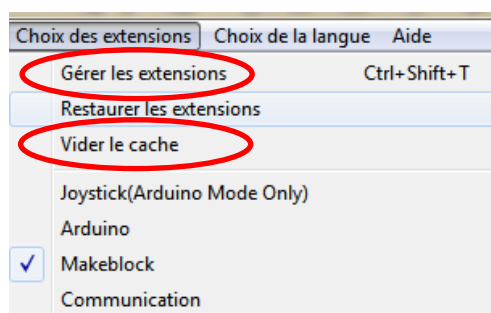
### *Pour « EPI\_Jeulin\_1\_0.zip »*

 ATTENTION : NE PAS DECOMPRESSER LE FICHIER

Lancer le logiciel mBlock.

Si vous avez déjà installé des extensions, il est préférable de réinitialiser celles-ci.

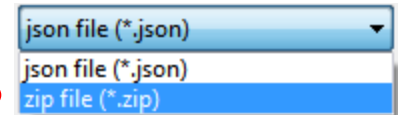
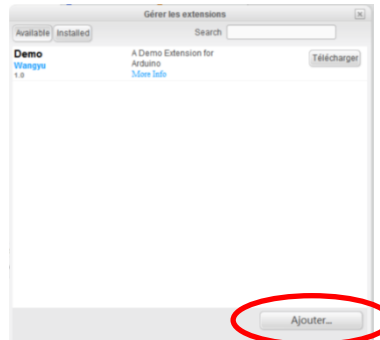
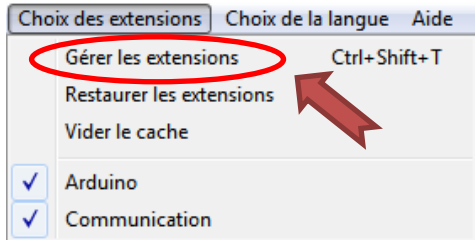
⇒ Dans le menu de mBlock cliquer sur « Choix des extensions » puis « Vider le cache ».



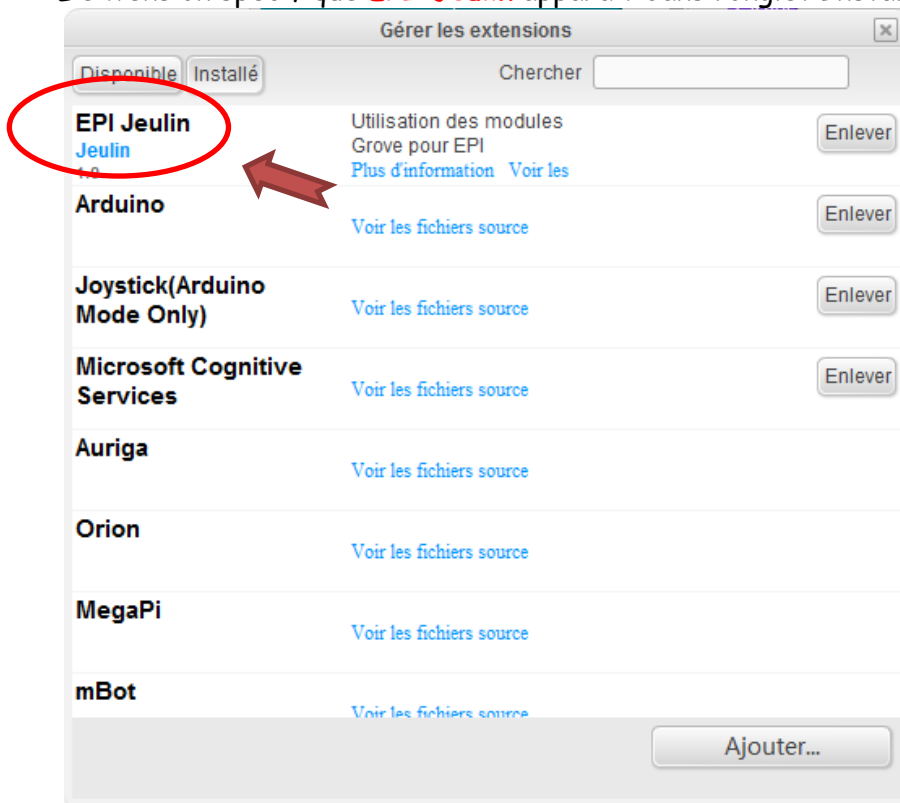
Redémarrer mBlock, il sera à présent réinitialisé. Changer la langue dans « Choix de langue ».

Dans le menu de mBlock cliquer sur « Choix des extensions » puis « Gérer les extensions » puis « Ajouter ».

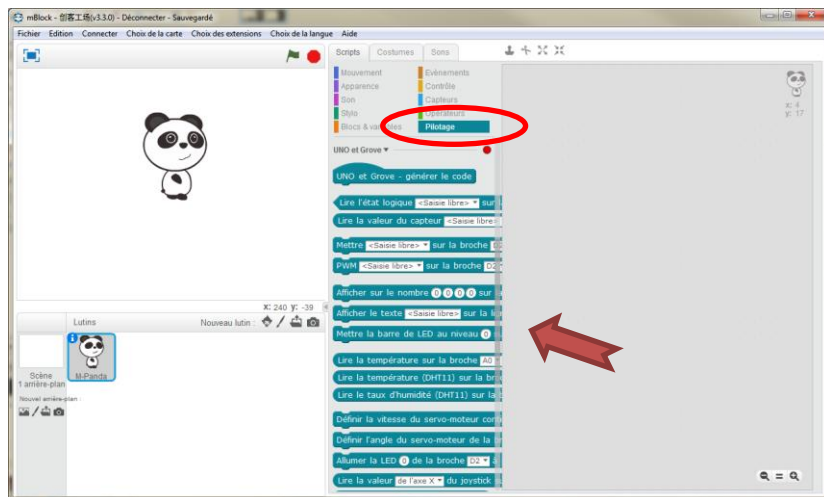
En bas à droite de la boîte de dialogue, modifier le type d'extension «.json » par «.zip» puis ouvrir le fichier «EPI\_Jeulin\_1\_0.zip».



L'extension spécifique **EPI Jeulin** apparaît dans l'onglet Installé



Dans l'onglet scripts « Pilotage », une nouvelle extension avec une multitude de blocs compatibles aux applications EPI est alors accessible.



Remarque :

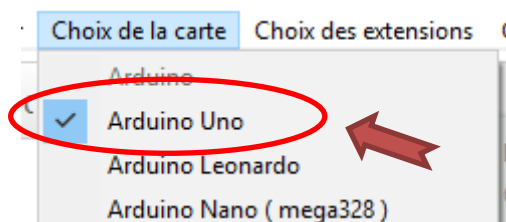
Lors de chaque mise à jour du logiciel mBlock, cette procédure d'installation de l'extension EPI\_Jeulin.zip est à refaire.

## II) Utilisation de mBlock avec une carte Arduino™ Uno

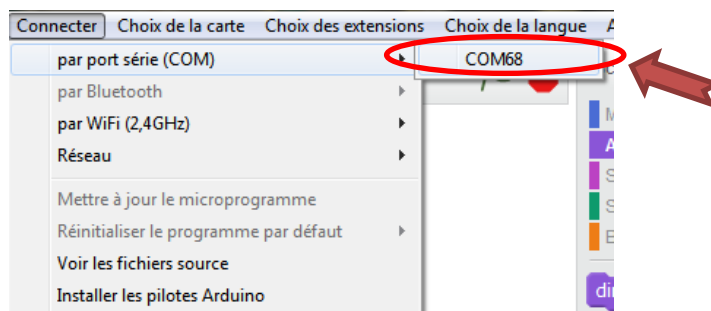
### 1. Mode connecté

Voici les étapes pour utiliser une carte Arduino™ Uno avec mBlock en mode connecté.

a. Dans « Choix de la carte », sélectionner Arduino Uno



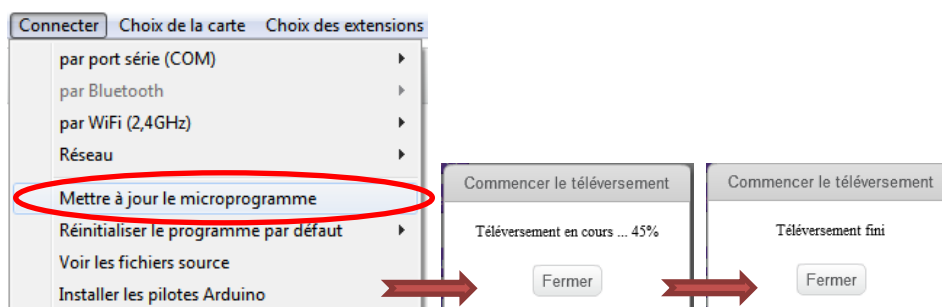
b. Connecter la carte Arduino en sélectionnant le port COM correspondant à la carte Arduino™.



c. Vérifier que la connexion est bien établie (le mot « Connecté » doit être affiché).



d. Mettre à jour le microprogramme.



Vous pouvez à présent utiliser votre carte Arduino™ en mode connecté.

Pour le contrôle et l'exécution de votre programme, il faut utiliser le(s) bloc(s) situé(s) dans le script « Evènements ».





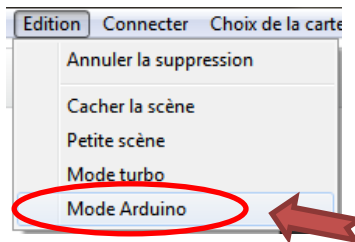
## 2. Mode déconnecté

Voici les étapes pour utiliser une carte Arduino™ Uno avec mBlock en mode déconnecté. Les étapes de connexion sont identiques aux points 1. et 2. du mode connecté.

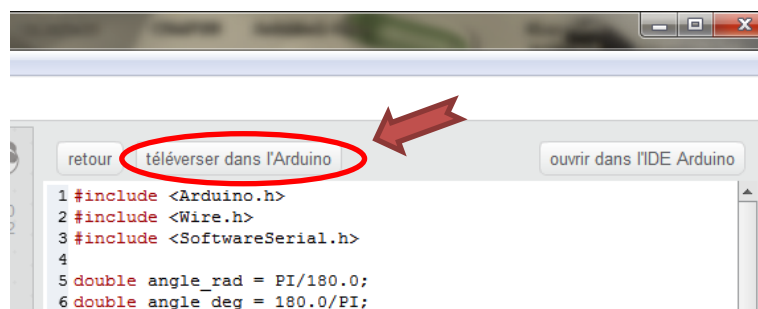
1. Après avoir créé votre programme, le bloc de départ doit être :

UNO et Grove - générer le code

2. Dans « Edition », activer le mode Arduino; un volet apparaît sur la droite.



3. Une fois le programme écrit, cliquer sur « Téléverser dans l'Arduino ».



Une fenêtre indique le statut du téléversement.



Votre programme est à présent dans la carte Arduino™, vous pouvez déconnecter le câble USB et intégrer votre carte Arduino™ dans votre projet.



Pour utiliser le « Mode connecté » après avoir téléversé un programme, ne pas oublier de mettre à jour le microprogramme de votre carte Arduino™ et réutiliser les contrôles « Evènements »



### III) Utilisation de l'extension

Notre extension est exclusivement compatible avec les carte Arduino Uno et les modules Grove. Voici comment utiliser ces nouveaux blocs Arduino & Grove :

mBlock - Based On Scratch From the MIT Media Lab(v3.4.4) - par port série (COM) Connecté - Pas sauvegardé

Fichier Edition Connecter Choix de la carte Choix des extensions Choix de la langue Aide

Untitled

Scripts Costumes Sons

Mouvement Apparence Son Stylo Blocs & variables Evénements Contrôle Capteurs Opérateurs Pilotage

EPI Jeulin

UNO et Grove - générer le code

Lire l'état logique <Saisie libre> sur la broche D2

Lire la valeur du capteur <Saisie libre> sur la broche A0

Mettre <Saisie libre> sur la broche D2 à haut

Effacer le texte

Eclairer l'écran avec la couleur rouge: 255 vert: 255 bleu: 255

Afficher le texte <Saisie libre> sur la ligne 0

Afficher sur le nombre 0 sur la broche D2 et afficher ":"

Mettre la barre de LED au niveau 0 sur la broche D2

Lire la valeur Rouge sur le capteur couleur

Allumer la LED 0 de la broche D2 à la couleur rouge: 255 vert: 255 bleu: 255

Lire la température sur la broche A0 en °C

Lire la température (DHT11) sur la broche D2 en °C

Lire le taux d'humidité (DHT11) sur la broche D2 en %

RFID D2 disponible

Lire la valeur du capteur RFID sur la broche D2

Effacer toutes les empreintes sur la broche D2

Vérifier l'empreinte 0 sur la broche D2

Enregistrer l'empreinte 1 sur la broche D2

Scène 1 arrière-plan

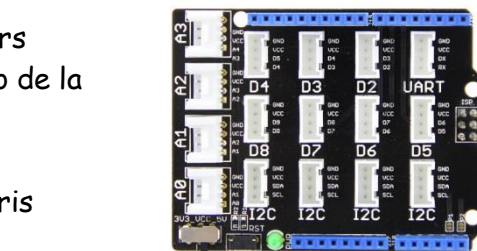
Nouvel arrière-plan :

M-Panda



Presque tous nos blocs Arduino & Grove présents dans les valises EPI, fonctionnent en mode connecté (via le microprogramme) et en mode déconnecté (en téléversement).

Il existe plusieurs types de protocole des capteurs/actionneurs  
En fonction de celui-ci, il sera nécessaire d'affecter le numéro de la broche de connexion.



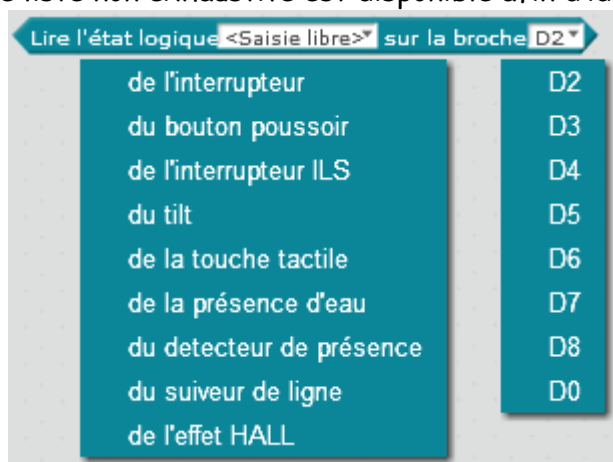
- Branchement sur des Entrée/Sortie numérique Dx (x compris entre 2 à 8)
  - ⇒ Les informations sont numériques, tout ou rien, ou série.
  - ⇒ Le numéro de broche correspond au numéro du connecteur sur le shield Grove Arduino sur lequel il faut brancher le module.
- Branchement sur le connecteur I2C (peu importe le connecteur, le module sera automatiquement reconnu)
  - ⇒ Les informations sont de type série I2C.
- Branchement sur le connecteur UART
  - ⇒ Les informations sont de type série
- Branchement sur des entrées analogiques Ax (x compris entre 0 à 3)
  - ⇒ Les informations sont de type analogique 0 à 5V, converties de 0 à 1023.

### Bloc « Lire l'état logique... »

Ce bloc retourne la valeur numérique de la broche Dx correspondant au connecteur Dx du shield Arduino Grove.

Cette valeur est numérique: 0 pour un état bas ou 1 pour un état haut.

Dans le menu déroulant, une liste non exhaustive est disponible afin d'identifier votre capteur :



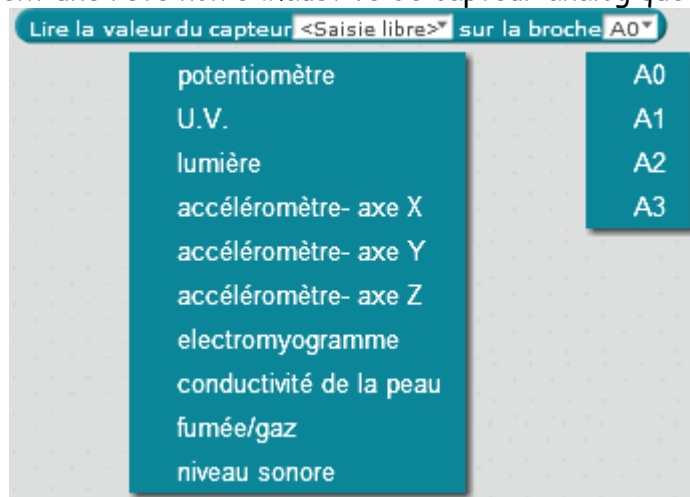
Pour personnaliser le nom de votre capteur, cliquer dans la zone de texte <Saisie libre> et taper le nom de votre capteur.

### Bloc « Lire la valeur du capteur... »

Ce bloc retourne la valeur de l'entrée analogique de la broche Ax correspondant au connecteur Ax du shield Arduino Grove.

Cette valeur est analogique, elle oscille de 0 à 1024.

Le menu déroulant contient une liste non exhaustive de capteur analogique:



Pour personnaliser le nom de votre capteur, cliquer dans la zone de texte <Saisie libre> et taper le nom de votre capteur.

### Bloc « Mettre sur la broche... »

Ce bloc pilote les actionneurs Grove branchés sur une sortie numérique de la broche Dx du shield Arduino Grove.

Cette valeur est numérique : 0 pour un état bas ou 1 pour un état haut.

Dans le menu déroulant, la liste est non exhaustive.



Pour personnaliser le nom de votre actionneur, cliquer dans la zone de texte <Saisie libre> et taper le nom de votre actionneur.

### Bloc « Afficher le texte... »

Afficher le texte <Saisie libre> sur la ligne 0

Ce bloc pilote le module Grove afficheur LCD 16x2 branché sur un des bus I2C du shield Arduino Grove.

Cet afficheur peut afficher 20 caractères sur 2 lignes : ligne 0 (1<sup>ère</sup> ligne) et ligne 1 (2<sup>ème</sup> ligne).

Cliquer dans la zone de texte <Saisie libre> et taper votre texte pour qu'il soit lisible sur l'afficheur.



Effacer le texte

Efface le texte de l'afficheur.

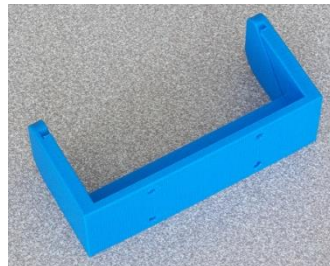
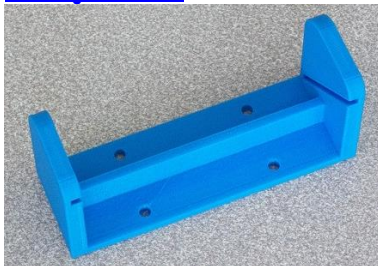
Eclairer l'écran avec la couleur rouge: 255 vert: 255 bleu: 255

Il est possible d'éclairer l'écran de l'afficheur suivant les 3 couleurs fondamentales rouge, vert, bleu

Chaque couleur a une valeur de 0 à 255.

Pour les possesseurs d'imprimante 3D, il est possible de créer un support d'afficheur qui se fixe avec 4 vis ou avec du scotch double face. Les fichiers au format STL sont disponibles sur

[www.jeulin.fr](http://www.jeulin.fr)



### Bloc « Afficher le nombre... »

Afficher sur le nombre Nombre sur la broche D2 et afficher ":"

D2  
D3  
D4  
D5  
D6  
D7  
D8  
D0

cacher  
afficher

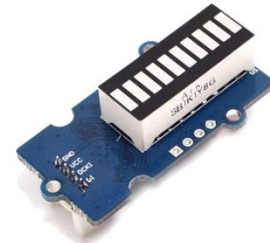


Ce bloc pilote le module Grove afficheur 4 digits branché sur un des connecteurs Dx du shield Arduino Grove.

Cet afficheur peut afficher un nombre de 0 à 9999.

Il est possible d'afficher ou de cacher le caractère « : » de l'afficheur.

### Bloc « Mettre la barre de LED au niveau... »



Ce bloc pilote le module Grove bargraphe LED branché sur un des connecteurs Dx du shield Arduino Grove.

Ce bargraphe possède 10 leds et permet d'afficher de 0 à 100 niveaux (chaque led peut avoir une luminosité variable en fonction de la valeur du niveau.

### Bloc « Lire la valeur ... sur le capteur de couleur »

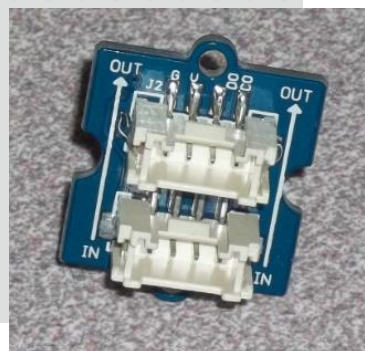
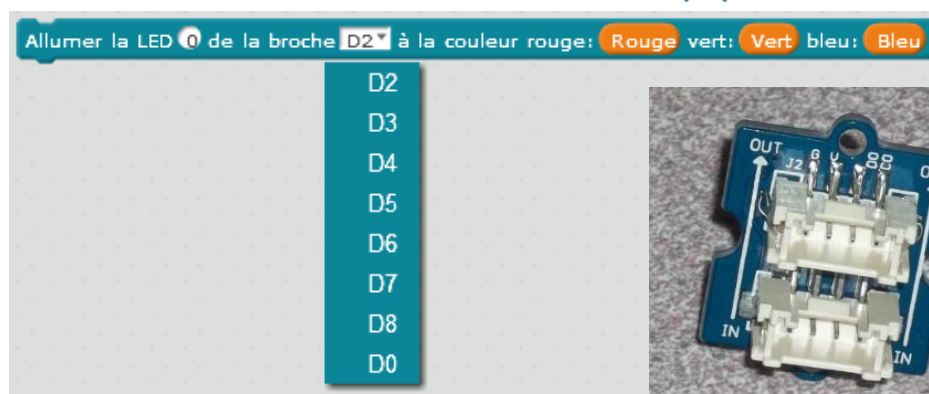


Ce bloc pilote le capteur de couleur Grove branché sur un des bus I2C du Shield Arduino Grove.

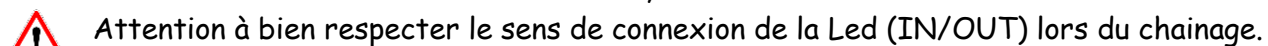
La valeur de la couleur retournée est comprise entre 0 et 255

Le choix de la couleur lue se fait parmi les 3 couleurs fondamentales Rouge, Bleu, Vert.

### Bloc « Allumer la LED ... à la couleur R,V,B »



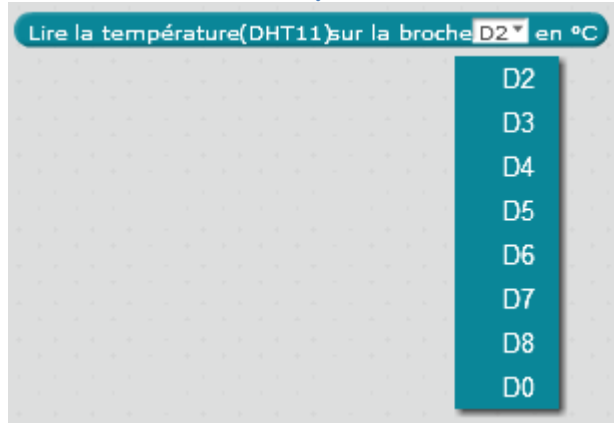
Ce bloc pilote la Led multicolore Grove branché sur un des connecteurs Dx du shield Arduino Grove. Ce bloc peut piloter jusqu'à 5 Led RGB montées en série. Le premier paramètre correspond à la Led à allumer : la 1<sup>ère</sup> étant « Allumer la Led 0 », la 2<sup>ème</sup> étant « Allumer la Led 1 ».

 Attention à bien respecter le sens de connexion de la Led (IN/OUT) lors du chainage.

Le 1<sup>er</sup> branchement doit être sur la broche IN, puis Out vers IN de la Led suivante ....

La couleur de la Led se pilote via les paramètres « Rouge », « Vert » et « Bleu » de 0 à 255 pour chaque composante.

### Bloc « Lire la température (DHT11) En °C ...»



Ce bloc pilote le capteur Température/hygrométrie Grove branché sur un des connecteurs Dx du shield Arduino Grove.

Ce bloc retourne une température de 0 à 50 °C.

### Bloc « Lire le taux d'humidité (DHT11)... En % ...»



Ce bloc pilote le capteur Température/hygrométrie Grove branché sur un des connecteurs Dx correspondant au connecteur Dx du shield Arduino Grove.

Ce bloc retourne le taux d'humidité en pourcentage 0 à 100 % à +/- 1%.

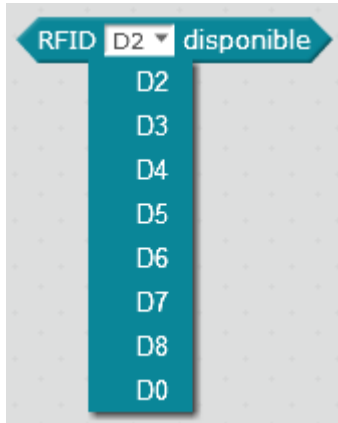
### Bloc « RFID...»

Le capteur RFID Grove est branché sur un des connecteurs Dx du Shield Arduino Grove.

Fonctionnement du capteur :



1) Un bloc fonction interroge la présence d'un badge ou d'une carte (passage de celui-ci)



RFID disponible

Le résultat logique (0/1) indique le passage d'un badge ou d'une carte RFID sur le capteur



2) Lecture de la valeur du capteur RFID



Lire la valeur du capteur RFID

Retourne la valeur codé du badge ou de la carte RFID (nombre à 7 chiffres) après détection du passage de celui-ci.

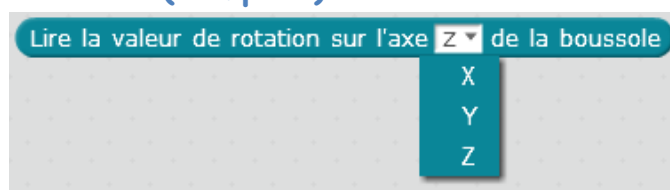


Ex: Code à 7 chiffres visible sur les badges.



Ex : Code à 7 chiffres visible sur les cartes

Bloc « Lire la valeur de rotation .....  
boussole (Compass) »



de la



Ce bloc pilote le capteur Compass (boussole) 3 axes Grove branché sur un des **bus I2C** du shield Arduino Grove.

Le bloc retourne la valeur de rotation de la boussole sur un des 3 axes X,Y,Z

La fonction permet de sélectionner l'axe pour la mesure.

La valeur lue sur les 3 axes varie de 0 à 360 degrés.

### Bloc « Transformer la valeur... »

Transformer la valeur **Valeur** de [ **IN\_min** à **IN\_max** ] vers [ **OUT\_min** à **OUT\_max** ]

Ce bloc converti une plage de mesure à une autre pour changer d'échelle par exemple.

Une fonction très puissante et très utile en pratique qui permet de réaliser une adaptation d'échelle de valeur (entre la valeur d'une conversion analogique numérique brute et la valeur réelle de la tension par exemple) en une seule ligne, là où plusieurs lignes de calcul seraient normalement nécessaires.

$$\text{Résultat} = \frac{(\text{Valeur} - \text{IN\_min}) \times (\text{OUT\_max} - \text{OUT\_min})}{(\text{IN\_max} - \text{IN\_min})} + \text{OUT\_min}$$

### Bloc Empreinte....

Le module Empreinte Grove est branché sur un des connecteurs Dx du shield Arduino Grove.

Une liaison série est réalisée sur le connecteur numérique pour accéder à plusieurs fonctions et valeurs.

Pour des raisons techniques liées au capteur lui-même, les blocs ci-dessous, ne fonctionnent qu'en « mode téléversé »



5 Blocs sont nécessaires pour l'utilisation du capteur d'empreinte.

Effacer toutes les empreintes sur la broche **D2**

Ce bloc est utile pour vider totalement la mémoire des empreintes mémorisées (167)

Cette étape prend quelques secondes.

Empreinte sur la broche **D2** disponible

Ce bloc attend la présence d'une empreinte sur le capteur.

Dès la détection d'une empreinte :

Si le résultat retourné est égal à 0, l'empreinte n'est pas actuellement en mémoire.

Si la résultat retourné est égal à 1, l'empreinte a déjà été enregistrée.

Vérifier l'empreinte **id** sur la broche **D2**

Ce bloc vérifie si l'image de l'empreinte du doigt est valide et la mémorise provisoirement dans l'id (0 à 166).

Si le résultat retourné est égal à 1, l'empreinte est valide et est mémorisée (dans l'id).

Si le résultat retourné est égal à 0, l'empreinte n'est pas valide.

Enregistrer l'empreinte  sur la broche

Ce bloc compare l'empreinte à celle mémorisée provisoirement dans la fonction précédente.  
Si le résultat retourné est égal à 1, L'empreinte avec l'id (0 à 166) est mémorisée.  
Si le résultat retourné est égal à 0, l'empreinte ne correspond pas à celle préenregistrée précédemment et n'est pas mémorisée.

Lire l'id de l'empreinte sur la broche

La valeur retournée correspond au numéro de l'empreinte mémorisée.  
Ce bloc fonction doit être placé juste après un test qui interroge la présence d'une empreinte en mémoire

Pour les possesseurs d'imprimante 3D, des fichiers au format STL sont disponibles sur [www.jeulin.fr](http://www.jeulin.fr)

Assembler le capteur comme ci-dessous avec du scotch double face.



### Bloc Rythme cardiaque doigt

Ce bloc pilote le module rythme cardiaque doigt Grove branché sur un des bus **I2C** du shield Arduino Grove.

Fréquence cardiaque sur doigt disponible

Ce bloc Interroge la présence de pouls cardiaque  
Si le résultat retourné est égal à 0, aucune fréquence cardiaque n'est détectée.  
Si la résultat retourné est égal à 1, une fréquence cardiaque est détectée.

Lire la Fréquence cardiaque sur doigt

Ce bloc retourne la valeur du pouls en pulsation par minute.  
Cette fonction doit être placée juste après la précédente qui détecte la présence d'un pouls.



### Capteur Rythme cardiaque oreille

Ce capteur Rythme cardiaque oreille Grove est branché sur un des connecteurs numériques Dx du shield Arduino Grove.

La fonction standard « Lire l'état logique » permet de lire les impulsions du poulx



Lire l'état logique Capteur Oreille sur la broche D2

Pour la clarté du programme, saisir « Capteur Oreille ».

### Capteur EMG

Ce module capteur EMG Grove permet de mesurer les courants électriques qui accompagnent l'activité musculaire.

Le branchement se fait sur un connecteur analogique Ax.

La fonction bloc standard permet de lire les impulsions électriques.

La valeur analogique lue est comprise entre 0 et 1024.



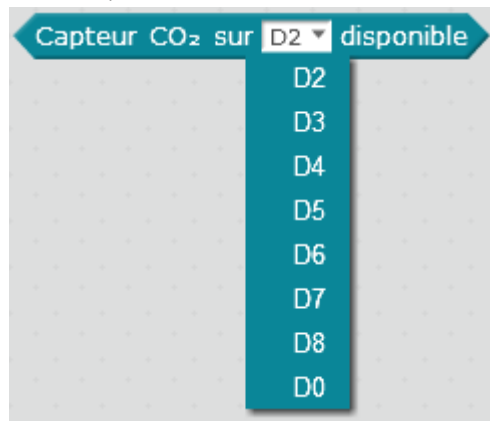
Lire la valeur du capteur Capteur EMG sur la broche A0

Pour la clarté du programme, saisir « Capteur EMG ».

### Capteur CO<sub>2</sub>

Le module CO<sub>2</sub> Grove branché sur un des connecteurs numériques Dx du shield Arduino Grove.

Une liaison série est réalisée sur le connecteur numérique pour accéder à plusieurs fonctions et valeurs.



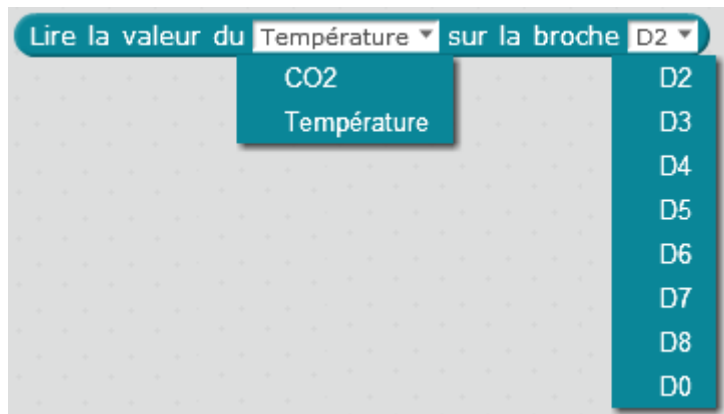
Si le résultat retourné est égal à 0, aucun capteur ou mesure de CO<sub>2</sub> n'est disponible.

Si la résultat retourné est égal à 1, une mesure de CO<sub>2</sub> est disponible.

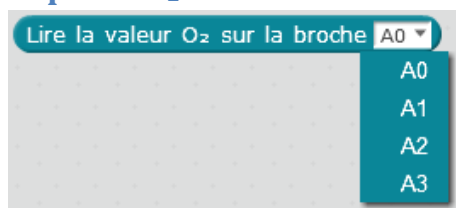
Ce bloc est indispensable avant de faire une acquisition.

Lecture de la valeur de CO<sub>2</sub> La plage de mesure est de 0 à 2000 ppm

Lecture de la température



### Capteur O<sub>2</sub>

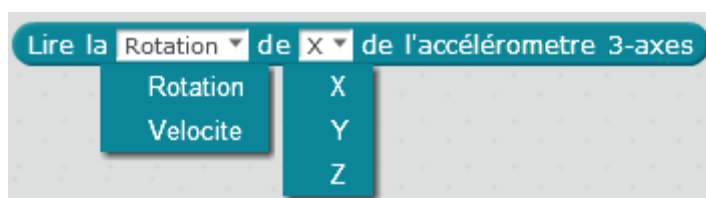


Le branchement se fait sur un des connecteurs analogue Ax.

La valeur lue est directement en % d'oxygène dans l'air, de 0 à 100%.

Remarque : il est indispensable de laisser sous tension le capteur pendant 30 minutes avant de faire des mesures.

### Bloc « Accéléromètre... »



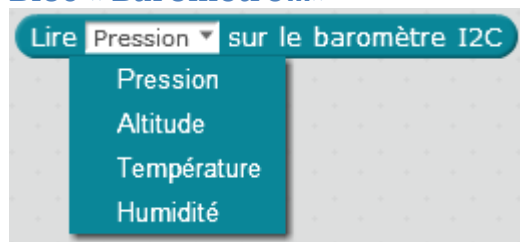
Ce bloc pilote le capteur Accéléromètre Grove branché sur un des **bus I2C** du shield Arduino Grove.

Lecture possible sur l'un des 3 axes au choix

Angle de rotation => en degrés

Vélocité => en m/s<sup>2</sup>

### Bloc « Baromètre... »



Ce bloc pilote le capteur Baromètre Grove branché sur un des bus **I2C** du shield Arduino Grove.

Choix de lecture :

Il est possible de lire l'une des 4 mesures.

Choix de mesure :

La pression atmosphérique => en hPa      Plage 300 - 1100 hPa

L'altitude => en mètre (calcul à partir de la pression et température)

La température => en °C

Le taux d'humidité => en %HR

### Bloc « Configurer pression au niveau de la mer »

Configurer pression au niveau de la mer à 1013.25



Cet étalonnage du capteur Baromètre est indispensable pour obtenir des valeurs de mesure correctes.

### Bloc « Lire la distance par ultrasons... »

Ce bloc retourne la valeur du module ultrason Grove branché sur un des connecteurs Dx du shield Arduino Grove.

Lire la distance par ultrasons sur la broche D2

D2  
D3  
D4  
D5  
D6  
D7  
D8  
D0



La valeur retournée est en centimètre.

La plage de mesure est de 3cm à 400 cm.

## Dépannage

Dysfonctionnement	Cause probable	Intervention
La carte de programmation ne répond plus en mode connectée	Perte de connexion avec la carte de programmation	Vérifier la sélection du COM dans le menu « Connecter » puis « par port série (COM) »
	Mauvais choix de carte	Sélectionner la carte de programmation Arduino uno dans le menu « Choix de la carte »
	Pas de microprogramme de communication dans la carte	Téléverser le firmware dans la carte de programmation dans le menu « Connecter » puis «Téléverser le microprogramme de communication»
	Problème de connexion à l'ordinateur	Vérifier la connexion du câble USB entre la carte de programmation et l'ordinateur
Les modules Grove fonctionnent de manière aléatoire	Alimentation réglée sur 3V	Commuter l'interrupteur du shield Grove sur la tension 5V
Erreur de compilation lors du téléversement	L'extension utilisée n'est pas la bonne	Cliquer sur « Choix des extensions », puis « Vider le cache » et redémarrer mBlock. Recommencer le processus d'ajout des extensions (voir page 3)
L'extension EPI_Jeulin.zip n'apparaît pas après son intégration	Problème d'ajout de l'extension	Recommencer le processus d'ajout des extensions (voir page 3)
Le PC ne reconnaît pas la carte Arduino	Pilotes sur le PC non installés ou problème de connectique USB	Installer les pilotes nécessaires au fonctionnement à l'aide du menu « Connecter » puis « Installer les pilotes Arduino », vérifier le câble USB