

# Les bases pour comprendre les modules :

Dans les abréviations dans un circuit électronique, on peut retrouver :

VCC (ou VDD ) : **FIL ROUGE** : Alimentation ou niveau haut Logique, si l'on branche une batterie de 5V , alors VCC = 5v

Gnd (VSS) : **FIL NOIR** masse (GROUND en anglais) ou niveau Logique bas pour les circuits.



**4 entrées analogiques pour les modules capteurs analogiques**

**1 sélecteur de tension 5v pour les modules connectés sur l'I2C ou 3,3v pour les autres modules Grove**

**1 port UART utilisé pour communiquer en liaison ASYNCHRONE (exemple : le module Bluetooth ...)**

**7 entrées / sorties numériques pour les modules capteurs et actionneurs logiques et numériques**

**4 ports I2C pour communiquer en liaison SYNCHRONE (exemple : afficheur / carte moteur ...)**

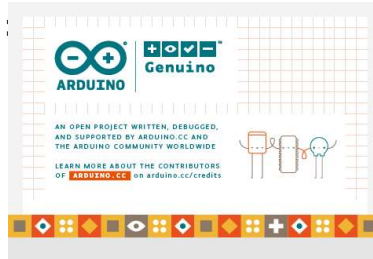
**Sommaire** (pour ajouter des liens (il faut ajouter un signet à un mot) puis créer un lien hypertexte dans les mots du sommaire)

| <div> <b>RAPPEL :</b><br/> <b>UTILISATION DES LOGICIELS</b> </div> <div> <a href="#">Ardublock education ou augmenté</a><br/> <a href="#">MBlock</a><br/> <a href="#">Applinventor</a> </div> |   |
|---|---|
| Les capteurs  | Les actionneurs   |
| <a href="#">Le potentiomètre</a>  | <a href="#">Le support de DEL</a>   |
| <a href="#">Le microphone</a>   | <a href="#">Le buzzer</a>   |
| <a href="#">Le bouton poussoir/ tactile</a>   | <a href="#">Carte I2C contrôleur moteurs à courant continue</a><br><a href="#">Autre cartes contrôleur moteur L293D</a> |
| <a href="#">Le thermomètre</a>  | <a href="#">L'écran LCD</a>   |
| <a href="#">Le capteur Ultrason</a>   | <a href="#">Le relais Grove ( Autres Relais )</a>   |
| <a href="#">Le capteur de lumière</a>   |   |
| <a href="#">Le joystick</a>   |   |
| <a href="#">Infra rouge d'obstacle standard</a>   |   |

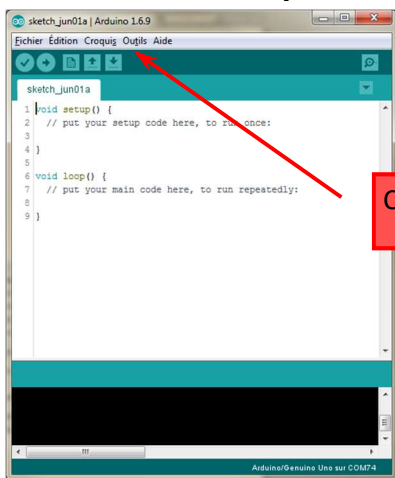
# Lancez ArduBlock Education ou lancez Arduino Education

**Avant toutes choses vérifiez que vous n'êtes pas connecté avec MBlock en même temps que Ardublock !**

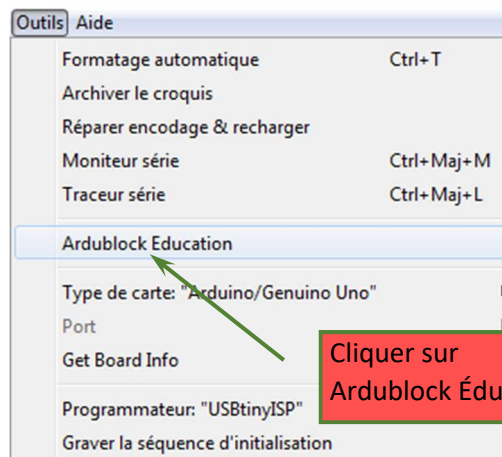
L'application Arduino IDE se lance :



Une fois lancé, cliquer sur « Outil », puis cliquer sur ArduBlock Éducation.



Cliquer sur Outils



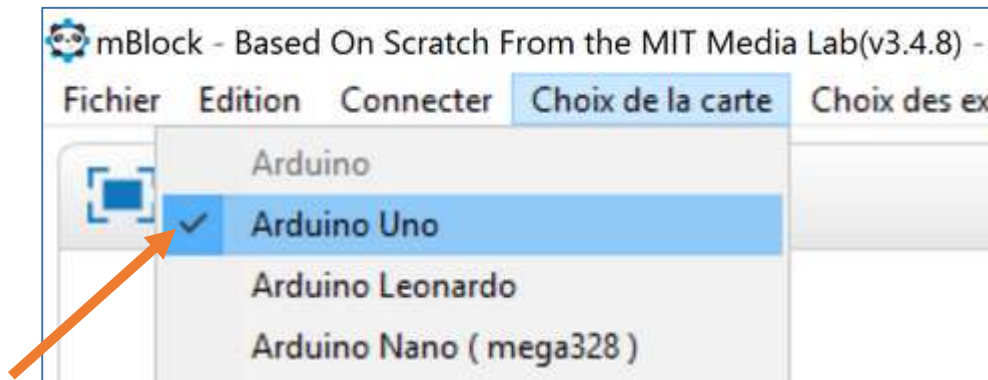
Cliquer sur  
ArduBlock Éducation

## Lancez MBlock

Il est nécessaire de vérifier ces étapes avant de commencer à créer votre programme en blocks :

**Avant toutes choses vérifiez que vous n'êtes pas connecté avec Ardublock en même temps que MBlock !**

1) Est-ce que j'ai sélectionné la bonne carte Arduino Uno ?

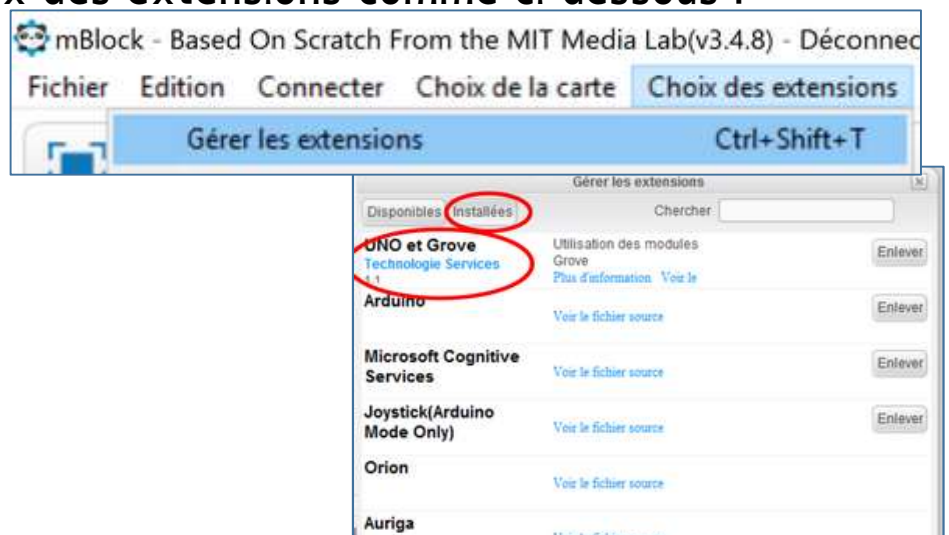


Vérifiez en cochant comme cela :

2) Est-ce que j'ai les bonnes extensions pour utiliser Uno et Grove ?

Dans le menu Choix des extensions comme ci-dessous :

Puis vérifiez cela :



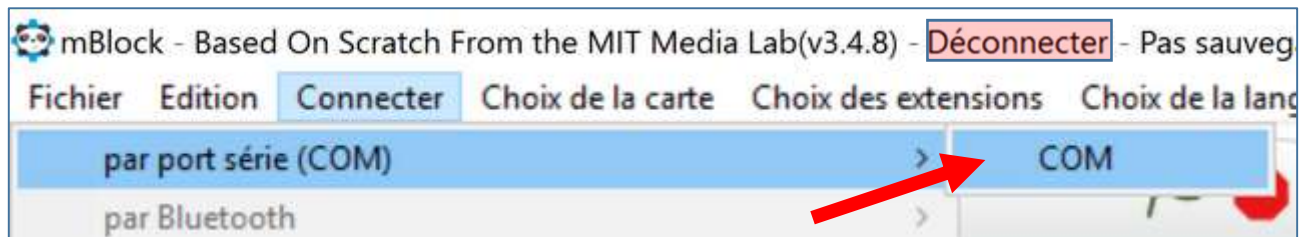
3) Est-ce que j'ai bien connecté la carte sur le port série (com=USB) ?

Dans le menu suivant

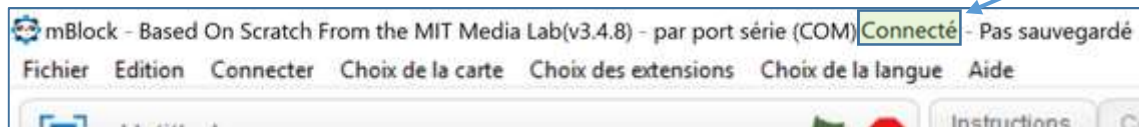


Cela veut dire que la carte Arduino n'est pas câblée !

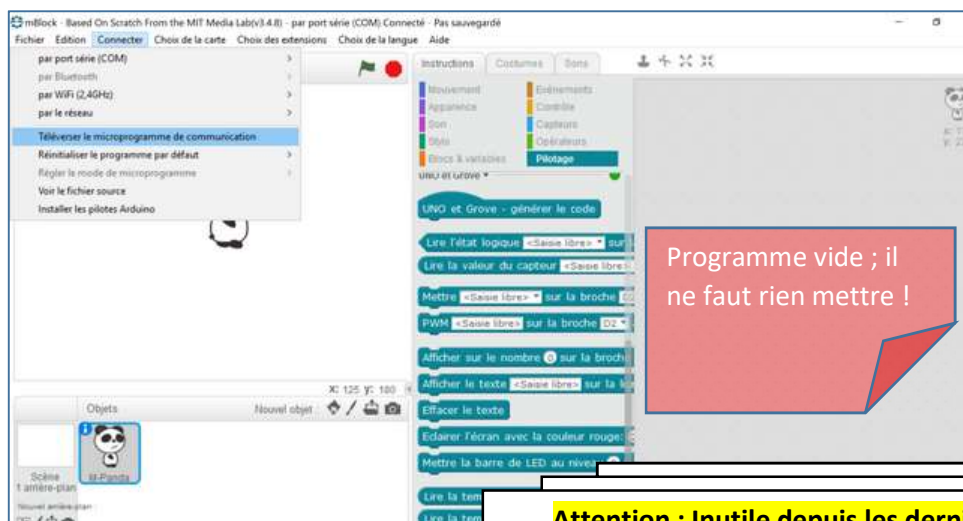
Si elle est câblée il faudra cliquer sur le « COM » comme ci-dessous :



On voit alors dans la barre de titre le mot « connecté »



4) Avant de commencer ; n'oubliez pas qu'il vaut mieux avoir une carte prête à fonctionner avec MBlock et lui téléverser un programme vide !



Attention : Inutile depuis les dernières versions

3.4.10 et +



# Le potentiomètre



Le potentiomètre convertit la tension entre 0v et VCC.

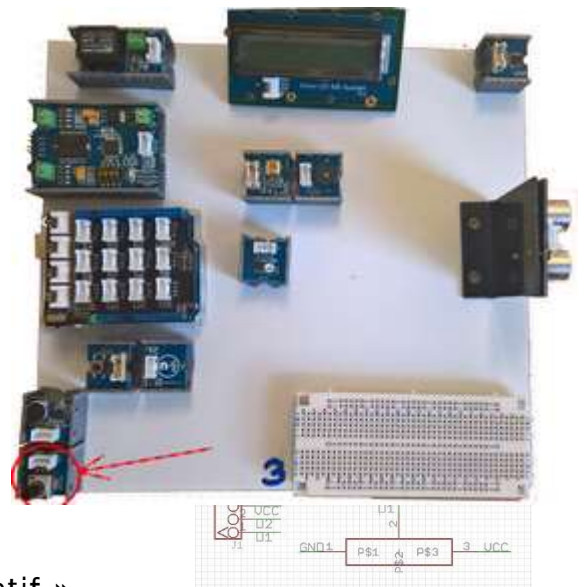
Il se branche sur une **entrée ANALOGIQUE A0 à A4.**

Il transmet un signal sur la broche 1 (fil jaune ). La broche 2 n'est

pas utilisé. Les valeurs vont **de 0 à 1023.**

Cette valeur est l'image de l'angle compris entre 0 et 300°.

Avec Arduino, on l'appelle aussi « capteur d'angle rotatif » (rotary angle sensor)



## Utilisation avec



Untitled

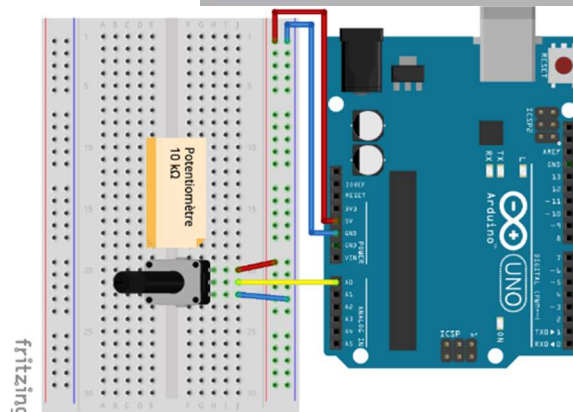
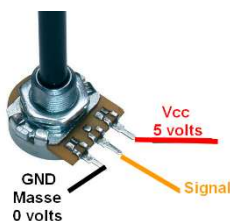


## Utilisation avec



Dans ce programme j'allume la DEL (D2) lorsque j'ai suffisamment tourné le potentiomètre.

On peut aussi utiliser un potentiomètre 10 Kohms en le branchant sur une carte Arduino Uno par exemple :



# Le microphone

Le module microphone est un microphone simple, qui est basé sur l'amplificateur LM358 avec un microphone à électret.

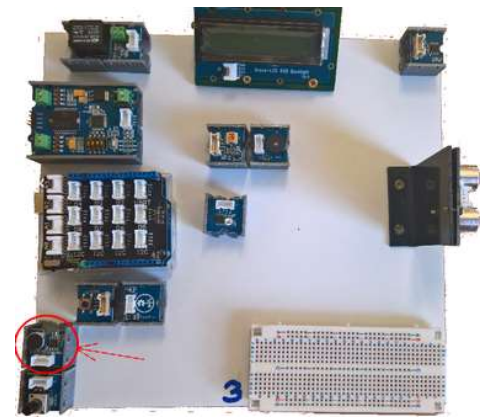
Il se branche sur une **entrée ANALOGIQUE A0 à A4**.

et peut être facilement échantillonné.

Il permet de **détecter le niveau sonore**.

Ce module **ne doit pas être utilisé pour recueillir le signal sonore**.

On peut par exemple l'utiliser pour contrôler une lampe à partir d'un volume sonore, mais pas comme un appareil d'enregistrement.



## Utilisation avec



Au lancement du programme, lorsque j'émetts un son, le point vert monte en fonction de l'intensité du son.

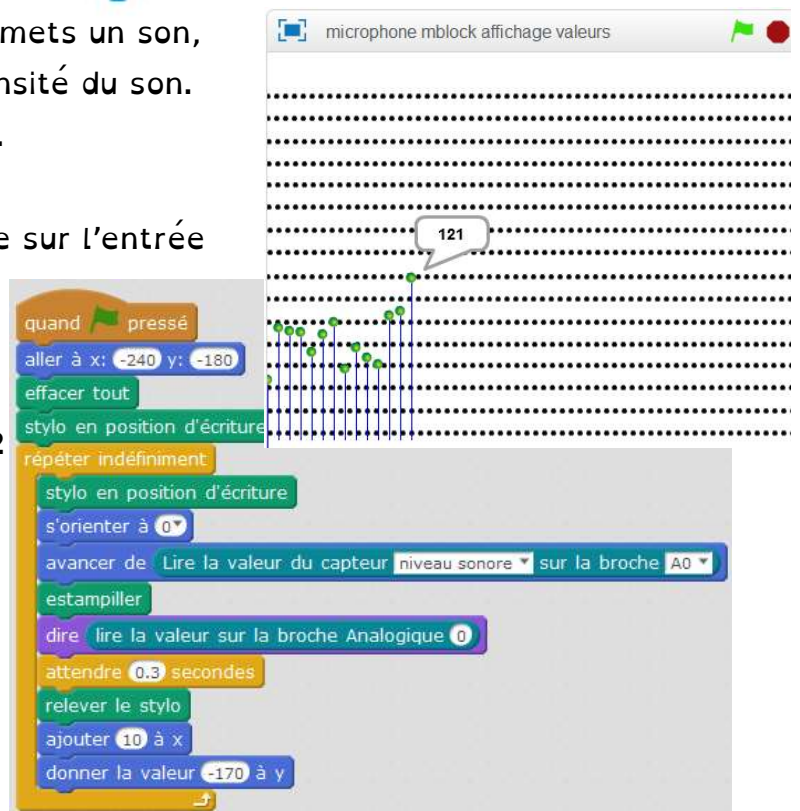
Il capte les sons toutes les 0.3 secondes.

Branchez le module microphone de Grove sur l'entrée analogique A0.

Avec mBlock, ouvrez le fichier :

microphone mBlock affichage valeurs.sb2



On peut alors décider de commander un actionneur qui réagirait à un niveau sonore.



**Nb : Ici, le programme échantillonne toutes les 0.3 seconde. Si le son est trop court, il ne sera pas affiché à la bonne valeur / hauteur.**

## Le Joystick

Emprunté aux consoles de jeux vidéo (PS2) ces modules ont deux axes et un bouton poussoir. IL possède deux potentiomètres de 10K $\Omega$  sur chacun des axes X et Y.

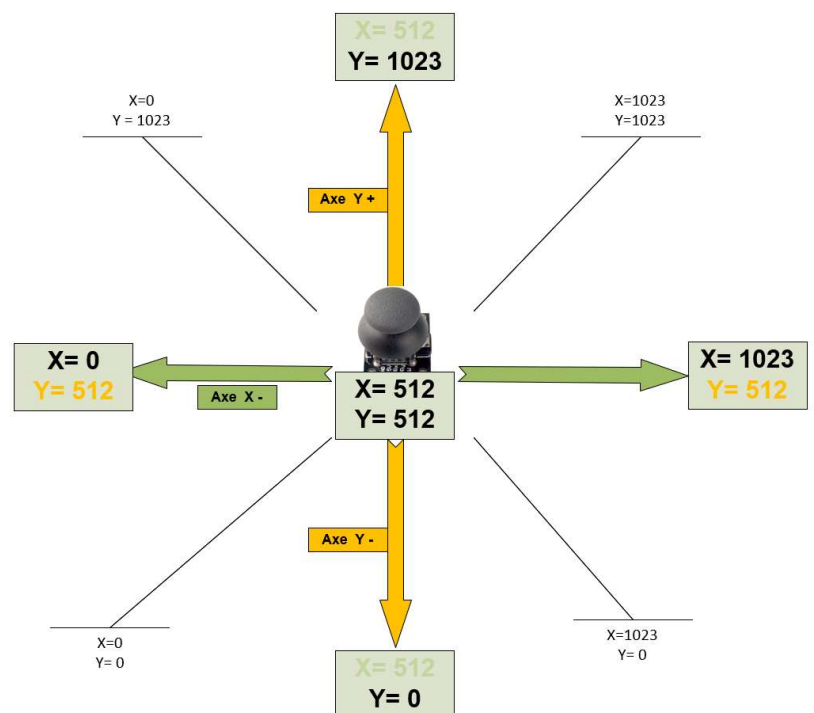
|   |  |
|---|--|
|    |   |
| <p>Le module Grove permet de coder les déplacements sur X et Y.</p> <p>Lorsque l'on appui sur le bouton poussoir, il retourne la valeur 1023.</p> <p>Toutes les autres valeurs sont proportionnelles à l'inclinaison du joystick.</p> | <p>Ce module permet de lire les valeurs de 0 à 1023 sur l'axe X branché sur et 0 à 1023 l'axe Y branchée chacun sur une entrée analogique (exemple A0 pour X et A1 pour Y).</p> <p>Le bouton poussoir doit être connecté sur une entrée numérique, la valeur retourne 1 pour un appui.</p> |

On peut voir sur le graphique ci-contre les plages de valeur. Sur chaque axe, pour ce module, les valeurs vont de 0 à 1023.


La valeur théorique, lorsque le bouton est relâché est de 512 ( $1024/2$ ).

Mais cela dépend de la qualité des composant du joystick.

Il est nécessaire de prendre en compte une plage de valeur (entre 500 et 520 par exemple).



## Utilisation avec mBlock

 mBlock - Based On Scratch From the MIT  
Fichier Edition Connecter Choix de la c





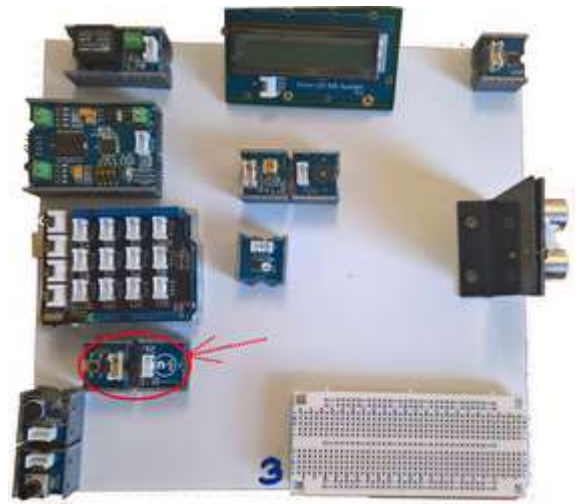
# Le bouton poussoir / La touche tactile

Ces modules logiques sont à définir comme des **Entrées numériques (D2 à D8)**.

Grove - Touch Sensor vous permet de remplacer la pression avec le toucher.

Il peut détecter le changement de capacité quand un doigt est proche. Cela signifie que peu importe votre doigt touche directement le pad ou juste reste près du pad

La valeur envoyée lors de l'appui, ou la détection est un état HAUT (high).



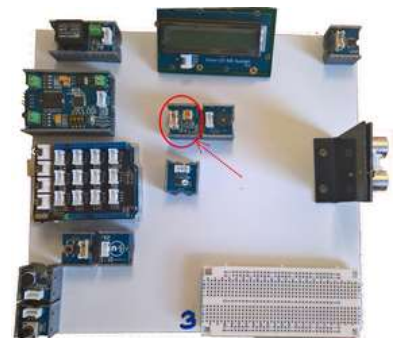
## Avec le support de LED



Le module support de LED est contrôlé à partir des **sorties numériques (D2 à D8)**.

La luminosité de la LED peut être réglée par potentiomètre sur le module.

La DEL est active à l'état Haut (VCC).

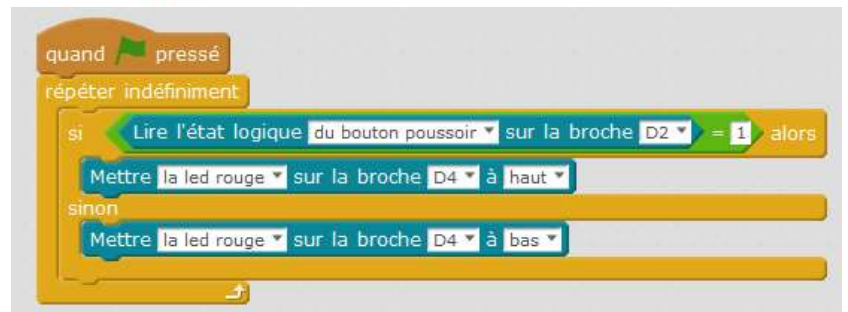


### Utilisation avec



En mode connecté, nous allons programmer la carte pour que le bouton poussoir allume une Del tant qu'on a le doigt sur le bouton poussoir.

On peut aussi dire qu'on allume si j'appuie, mais si je relâche la Del s'éteint.



### Utilisation avec

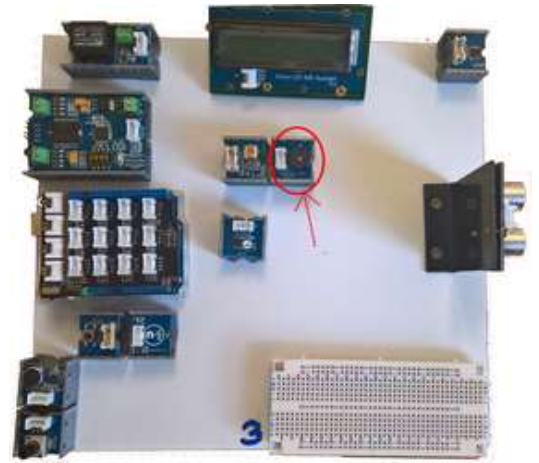




# Le buzzer



Le module Grove - Buzzer comporte un piézomètre comme composant principal. Le piézo peut être connecté à des **sorties numériques**, et émet une tonalité **lorsque la sortie est HIGH**.



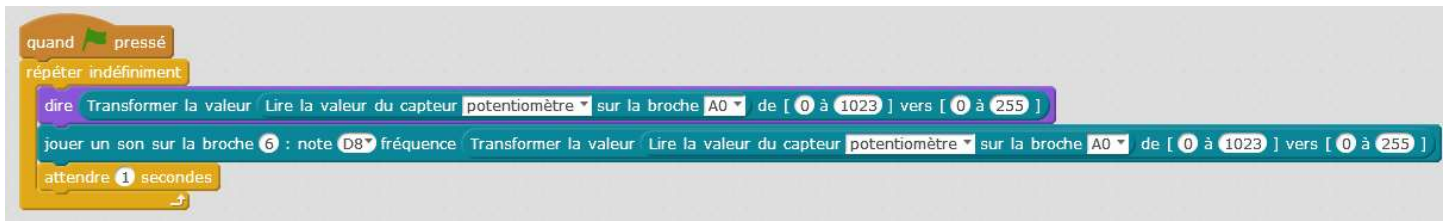
Alternativement, il peut être connecté à une **sortie analogique** de modulation de largeur d'impulsion pour **générer diverses tonalités et effets**. Dans l'exemple qui suit on gère le volume du buzzer avec le potentiomètre :

## Utilisation avec **mBlock**

Le potentiomètre est branché sur l'entrée analogique A0 et le buzzer sur la sortie PWM D6



Mais en programmant ainsi, on a une note différente (ici j'ai choisi un son acceptable pour nos oreilles si fragiles...)

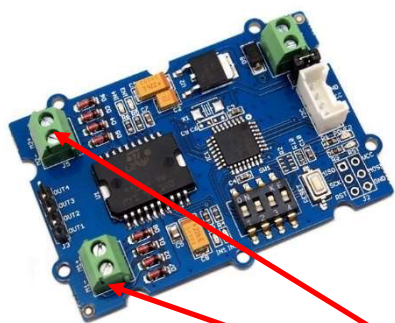


## **POUR JOUER UNE MELODIE :**

Vous pouvez entrer les notes comme «do», «re», «mi», etc ... ou vous pouvez utiliser la notation musicale anglaise. Vous pouvez également faire une note plus haut ou plus bas avec le chiffre sur le côté de celui-ci. Par exemple, le milieu C sur le piano serait "C4".

*Do Re Mi Fa Sol La Si Do*  
**C D E F G A B C**

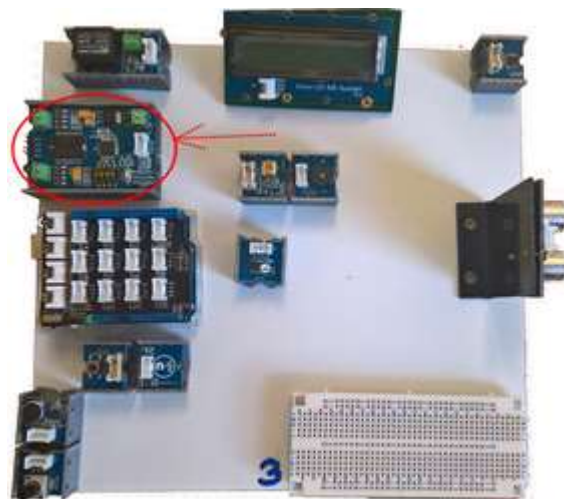
## La carte I2C pour les moteurs à courant continue



Le module Grove I2C peut contrôler directement le moteur pas à pas ou le moteur à courant continu DC dans les 2 sens.

Sa puce à double pont en H (L298P) est capable de

gérer jusqu'à 2A par canal M1 et M2 correspondant au moteur 1 et moteur 2.



La carte Arduino le contrôle par **n'importe quelle prise I2C**.

Les deux moteurs peuvent être entraînés simultanément tout en étant réglés sur une vitesse (PWM intégré ! ) et une direction différente.

Il peut alimenter deux moteurs à courant continu ou un moteur pas à pas à quatre fils.

Il nécessite une alimentation de 6V à ~~15V~~ 9V (attention j'en ai grillé un avec 12v-2A) pour alimenter le moteur et dispose d'un régulateur de tension intégré de 5V qui peut alimenter le bus I2C et l'Arduino (sélectionnable par cavalier).

La carte Arduino est protégée par des diodes (de roues libres ?)

Contrairement au moteur V1.2 Grove - I2C, le V1.3 permet aux utilisateurs de contrôler le pas à pas plus facilement. Vous n'avez plus besoin de contrôler les steppers tout le temps.

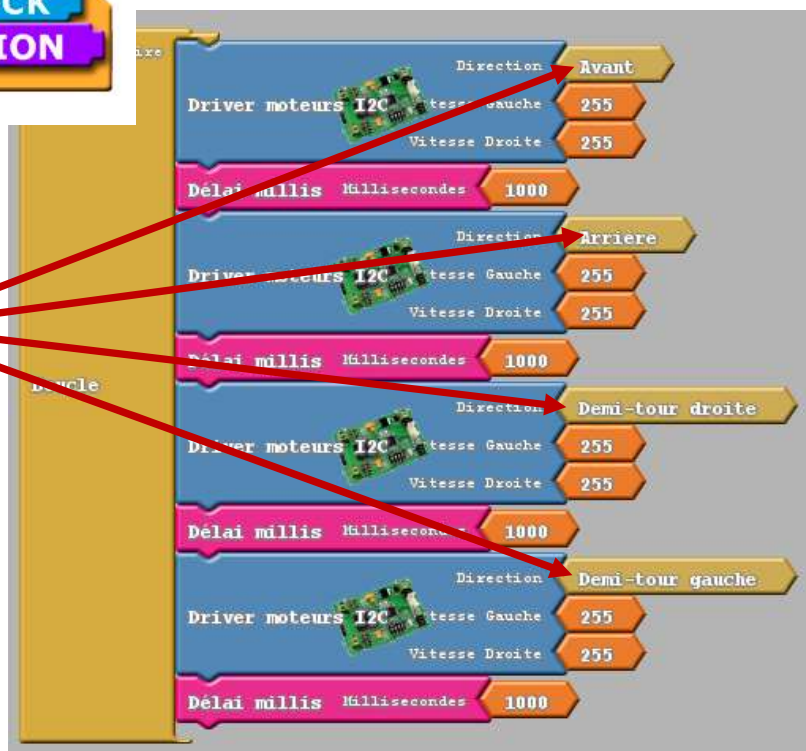
### Utilisation avec



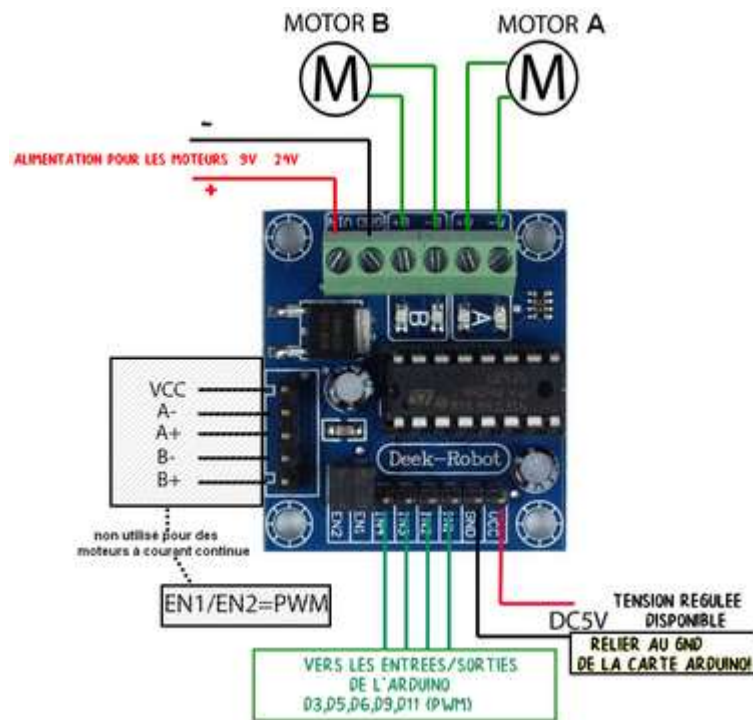
Les 2 moteurs avanceront dans le même sens, puis chaque seconde, ils changeront d'orientation

NB : LES CONSEILS DE DUINOEDU

- LORS DE NOS TESTS,
  - NOUS AVONS REMARQUE QU'UN TEMPS DE PLUSIEURS SECONDES POUVAIT ETRE NECESSAIRE APRES UN RESET DE LA CARTE ARDUINO
  - UN RESET COMPLET EST OBTENU PAR : RESET CARTE ARDUINO > RESET MODULE > RESET MODULE
- N'OUBLIEZ PAS
  - QU'IL Y A UN RESET SUR LE MODULE LUI-MEME (IL EST AUTONOME)
  - DE TELECHARGER OU METTRE A JOUR ARDUINO AUGMENTE - AIDE INSTALLATION ARDUINO AUGMENTE



## Carte contrôleur pour moteur base L293D



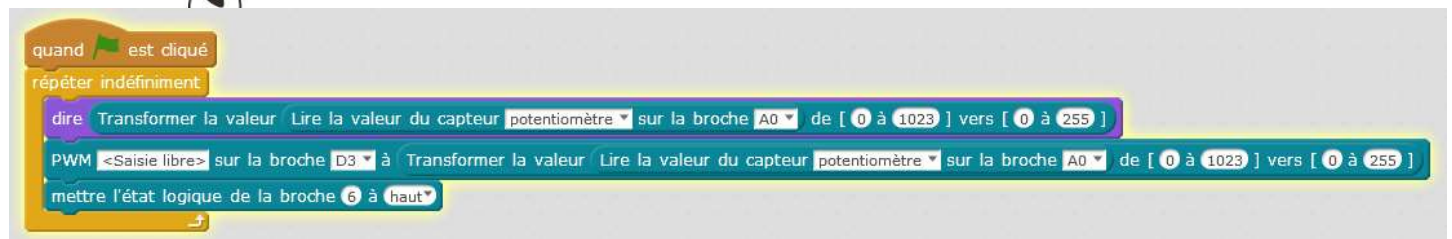
Cette carte permet de contrôler le sens de rotation ainsi que la vitesse de rotation du moteur.

Dans l'exemple ci-dessous, j'alimente avec 9v sur le bornier et je n'ai branché qu'un moteur sur Motor A.

La carte Arduino reçoit sur la broche D3(PWM) la valeur du potentiomètre. D6 permet de déterminer le sens de rotation (0 dans un sens et 1 pour tourner dans l'autre sens).

 Untitled

## Utilisation avec



Astuce : On peut pour simplifier le câblage utiliser les 5v de cette carte contrôleur pour alimenter l'Arduino.



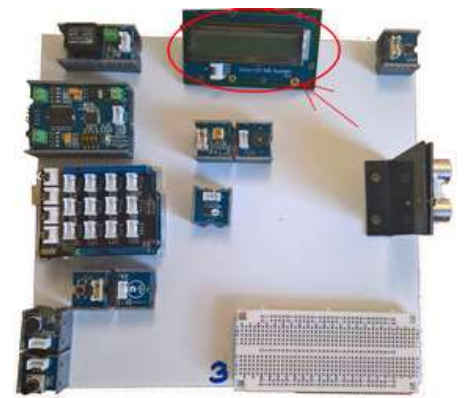


## L'écran LCD

Alimenté en 5v sur le port I2C, l'écran permet de pouvoir afficher 16 caractères sur 2 lignes.

La couleur et l'intensité sont réglables.

La ligne 0 est la ligne supérieure de l'écran, et la ligne 1 et la ligne inférieure. La carte Arduino le contrôle par n'importe quelle prise I2C.



## Utilisation avec



# mBlock

**N'ayant pas encore réussi à utiliser l'écran en mode connecté, il faut donc téléverser en faisant bien attention à l'en-tête**

Dans ce programme, on fera apparaître la valeur de la position du potentiomètre placé sur l'entrée analogique A0



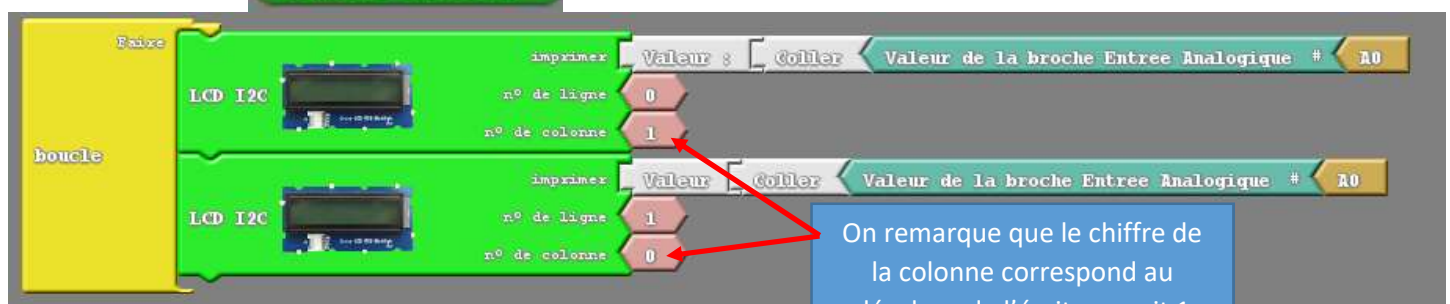
## Utilisation avec

**ARDUINO**  
Augmenté !

Dans le

**DuinoEDU Grove Add**

menu



On remarque que le chiffre de la colonne correspond au décalage de l'écriture, soit 1 pour un caractère



Bug inexpliqué pour le moment ...

Avec l'alim de 5v – 1.2A fourni avec le kit starter, l'écran s'allume mais n'affiche pas les caractères.

Utiliser le bloc d'alimentation car il faut + d'énergie



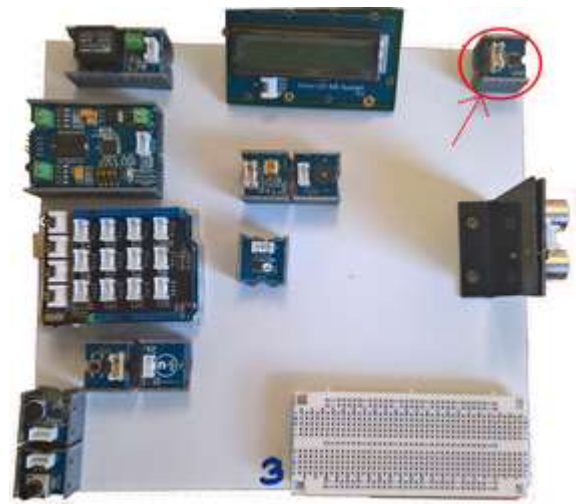


## Le thermomètre



Le capteur de température Grove utilise une thermistance (appelé aussi thermistor) pour détecter la température ambiante.

On connecte ce module les port A0 à A3 sur le shield Grove.



C'est une résistance qui varie en fonction de la température.

Sa valeur augmente lorsque la température ambiante diminue.

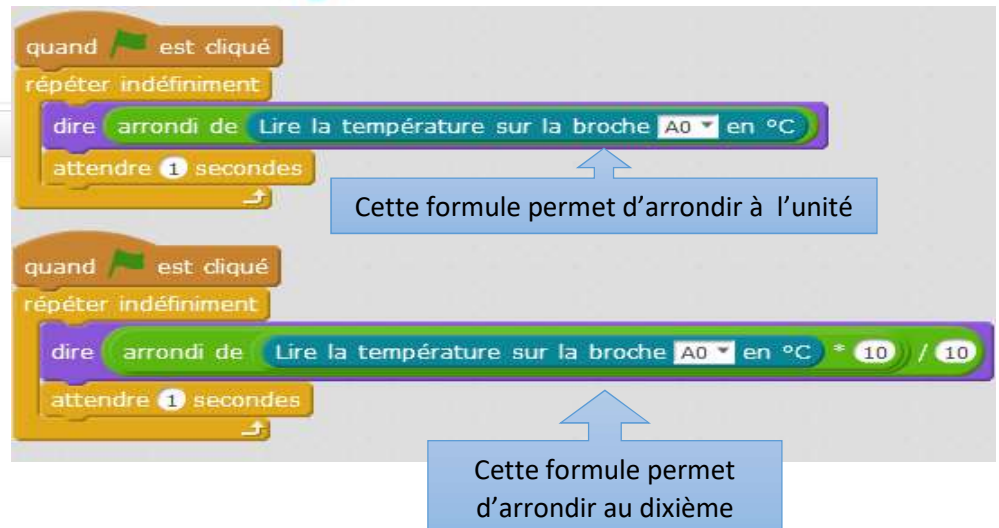
C'est cette caractéristique que nous utilisons pour calculer la température ambiante.

La portée détectable de ce capteur est de -40 - 125°C, et la précision est  $\pm 1.5^\circ\text{C}$ .

## Utilisation avec mBlock

Ce block est disponible depuis les dernières versions.

Untitled



## Utilisation avec

Dans ce programme on utilise l'afficheur LCD pour afficher directement la température du capteur.



ARDUINO  
Augmenté !

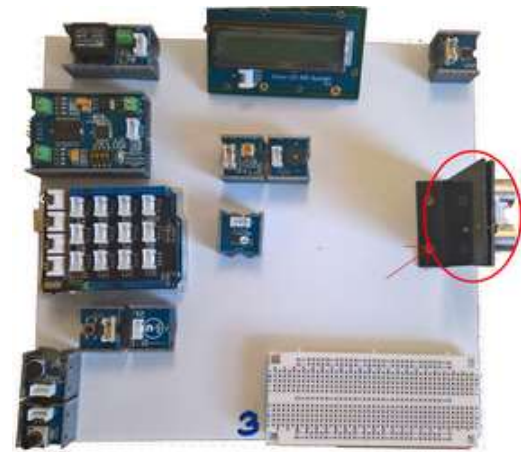
ARDUBLOCK  
ÉDUCATION

## Le capteur Ultrason



Le capteur à ultrasons est un module de mesure de distance sans contact (fonctionnant à 42KHz), adapté aux projets nécessitant une mesure moyenne distance de 3cm à 4 mètres (réellement ?)

La résolution est de 1 cm. On le branche sur une entrée Numérique de Grove (D2 à D8).



Sur ce modèle de capteur ultrason, il y a 4 broches.  
La broche VCC (+5v),  
La broche Gnd ( pour la masse)  
La broche « TRIG » pour « trigger » , déclencheur en français  
La broche « Echo » pour écouter le retour du son émis par la broche TRIGGER

Pour ce modèle il faut donc utiliser 2 broches de la carte Arduino. (exemple le trig sur la broche 13 et l'echo sur la broche 12)

### Utilisation avec



GROVE

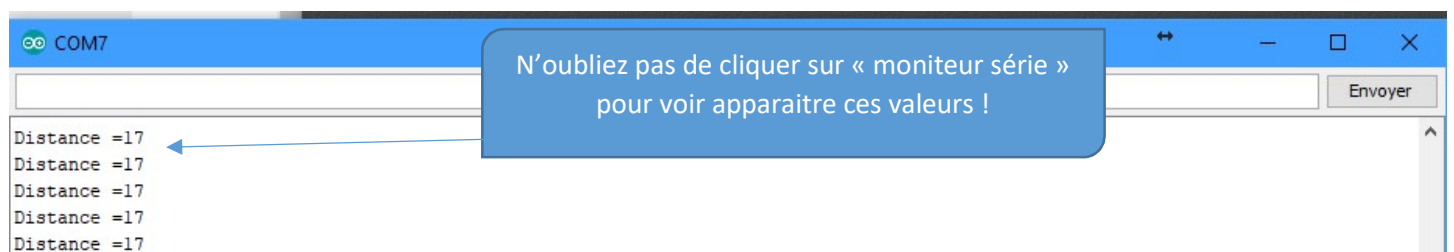


En attendant une mise à jour de mblock, le capteur Ultrason actuellement utilisable est le modèle **standard**

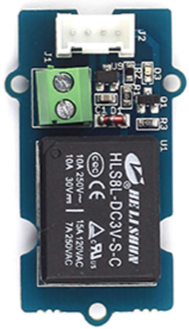
### Utilisation avec



Les 2 modèles se trouve dans le menu



## Le relais Grove

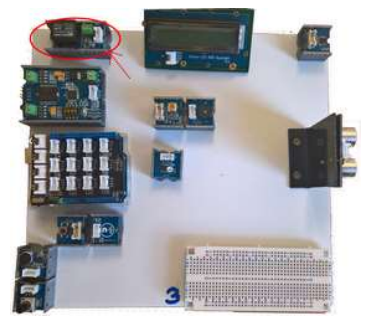


Le relais est un commutateur numérique normalement ouvert capable de commuter des tensions et des courants beaucoup plus élevés que vos cartes Arduino normales.

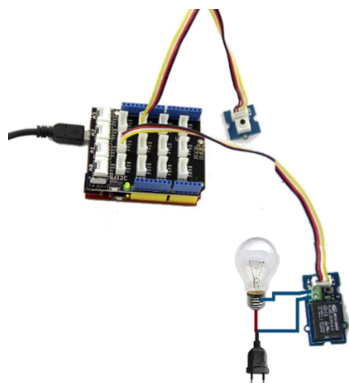
La carte programmable « Arduino UNO » ne fournit en sortie que 5 Volts et 40 mA

En clair, la partie commande de faible tension peut commander une partie plus puissante. Cela permet d'ouvrir ou de fermer un interrupteur 220V (DANGER) avec une commande en 5V (pas DANGER). **C'est un interrupteur commandé électroniquement.** Lorsqu'elle est réglée sur HIGH, la DEL sur le module s'allume et le relais se ferme, ce qui permet au courant de circuler.

La tension de crête est de 250V à 10 ampères.



**Veillez faire preuve de beaucoup de prudence lorsque vous travaillez avec des tensions secteur.**



**Le professeur s'assurera du montage électrique sur le bornier vert, les tensions utilisées ne dépasseront jamais 15V.**

Dans cet exemple, le bornier ne laissera pas passer le courant au repos.

Lorsqu'il sera commandé par le bouton poussoir, il pourra alors laisser l'énergie de la prise secteur vers l'ampoule.

## Utilisation avec mBlock



mBlock

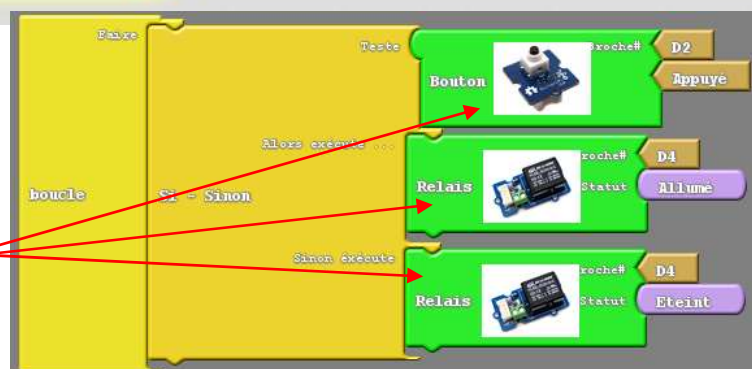
relais commandé par BP



## Utilisation avec

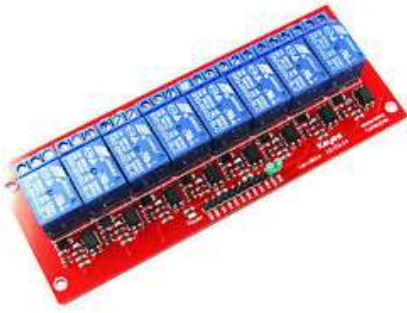
**ARDUINO**  
Augmenté !

**Seeed Studio Grove**





## Le relais d'autre marque



Pour commuter du 220v AC, ces relais ont besoin de 12v.

Avec 5v on peut commuter jusqu'à 110v AC.

**Il est interdit de laisser le câblage aux élèves sur le secteur.**

Il faut séparer l'alimentation de 5v (à 9v) pour la carte Arduino, et l'alimentation de 12v pour les relais si l'on ne veut pas détériorer les cartes Arduino.

Dans ce cas, il faut alors enlever le jumper qui relie le COM au GND.

Cette carte requiert des connaissances et présente un réel danger pour un utilisateur non averti !

**N'hésiter pas à utiliser un différentiel 30mA sur le circuit que l'on veut commuter.**

+ 12v sur Vcc  
0v sur Gnd

On connecte sur ces broches les sorties numériques de la carte Arduino  
Sur IN 1 à 8

Gnd de la carte Arduino sur le Com  
Ne rien câbler sur le Gnd du dessous

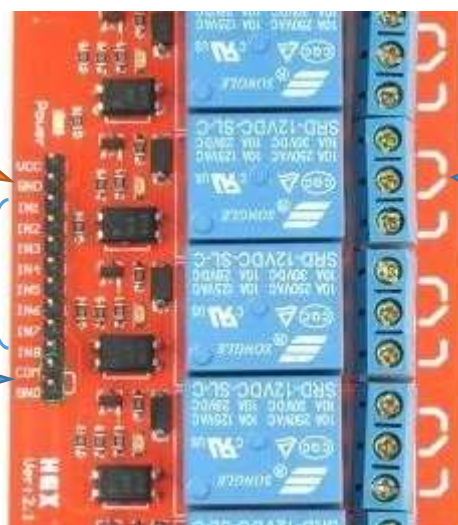


Schéma du fonctionnement  
(Pour la compréhension !!)

Au repos

Relais  
Commuté !



## Capteur de distance IR Grove

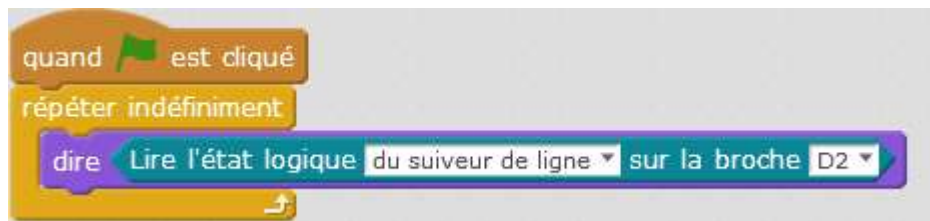


Ce module de détection infrarouge ou permet de détecter un objet réfléchissant à une distance de 4 à 15 mm.

Il permet d'obtenir une réponse « tout ou rien » de la présence de l'absence de l'objet réfléchissant devant le capteur.

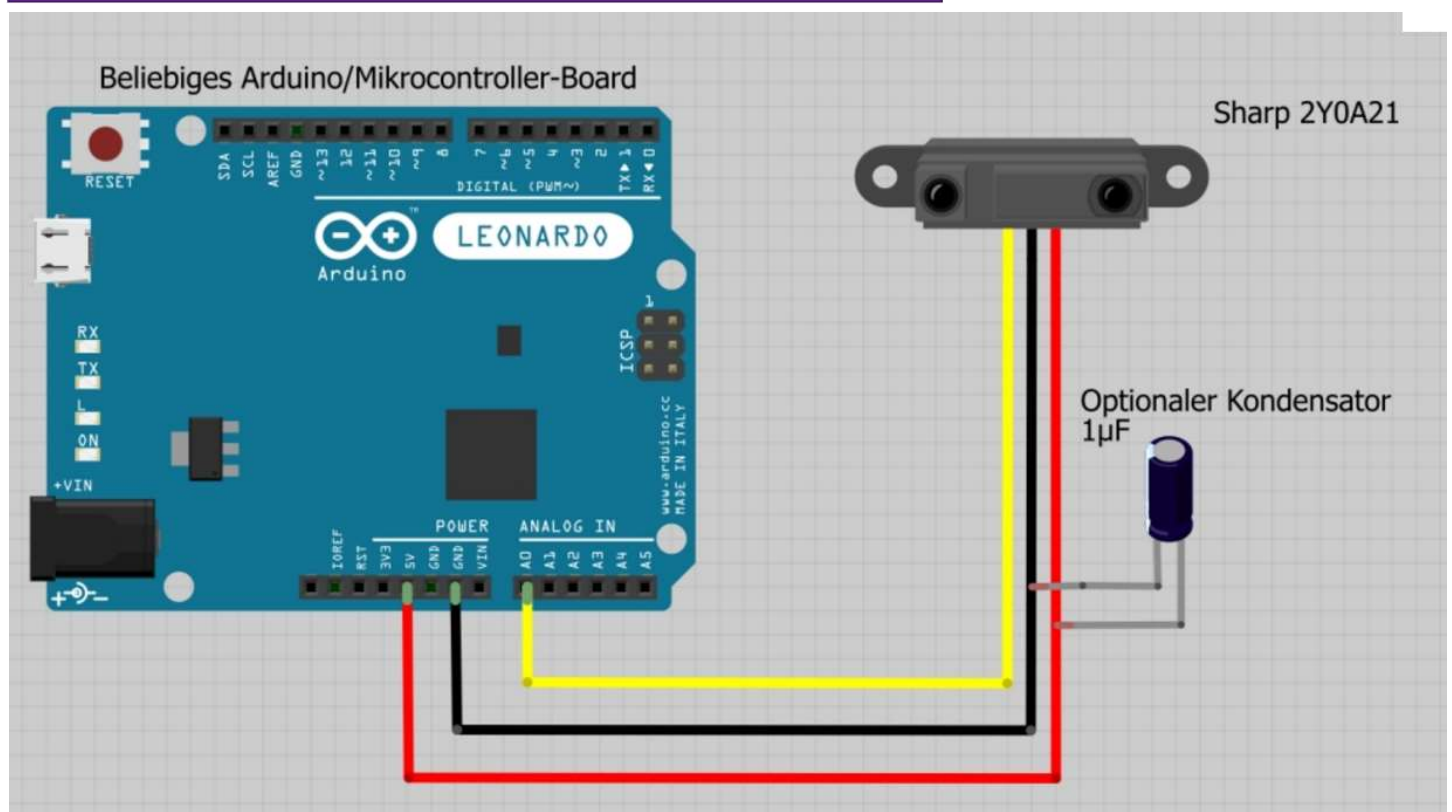
Le module est doté d'un potentiomètre pour ajuster la sensibilité et d'une LED rouge qui permet de vérifier la détection du module. Dès que le module détecte un objet, la LED s'allume.

### Utilisation avec



La valeur lue ainsi prendra la valeur « 0 » car aucun objet se trouve à proximité (<15mm)

## Capteur de distance SHARP (Robot POB)

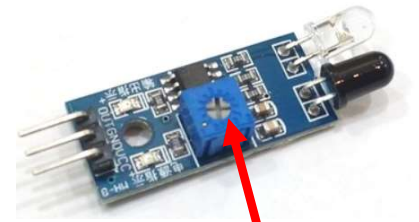


Pour utiliser ce capteur, il faudra faire plusieurs prises de mesures car les valeurs du capteur ne sont pas linéaires. En d'autres termes, il ne retourne pas la valeur 1 pour 1cm, 2 pour 2 cm...

# Le capteur d'obstacle Infrarouge / détecteur de ligne

Ces modules logiques sont à définir comme des

**Entrées numériques (D2 à D8).** La lumière Infrarouge est émise par une DEL IR réfléchiée sur l'obstacle puis reçue par l'autre. On peut régler la sensibilité de la distance en tournant le potentiomètre. Il permet de détecter les obstacle entre 2 à 30 cm.



**!!! Laissez le professeur effectuer ce réglage !!!**

Il ne retourne que des 0 ou des 1.

Si un objet, foncé se trouve devant lui, il ne le détecte pas. En revanche le blanc est facilement détectable. On peut alors s'en servir pour programmer un système qui suivrait une ligne.

Utilisation avec  **mBlock**



## Le capteur de lumière



Il s'agit d'un **module analogique** qui émet plusieurs signaux électriques qui peuvent être convertis en différentes gammes (cela dépend du convertisseur analogique / numérique sur votre carte

contrôleur, par exemple 0-255 pour un ADC 8 bits).  
Il intègre un phototriode hautement sensible et fiable.



**On connecte ce module les port A0 à A3 sur le Shield Grove.**

Ce capteur de lumière compatible Grove utilise le phototransistor LS06-S pour détecter la présence de lumière. L'utilisation d'un phototransistor à la place d'une photorésistance permet d'obtenir un signal plus linéaire.

### Utilisation avec **mBlock**

Sur Mblock les valeurs vont de 0 (quand il fait noir) à 750 en pleine lumière.

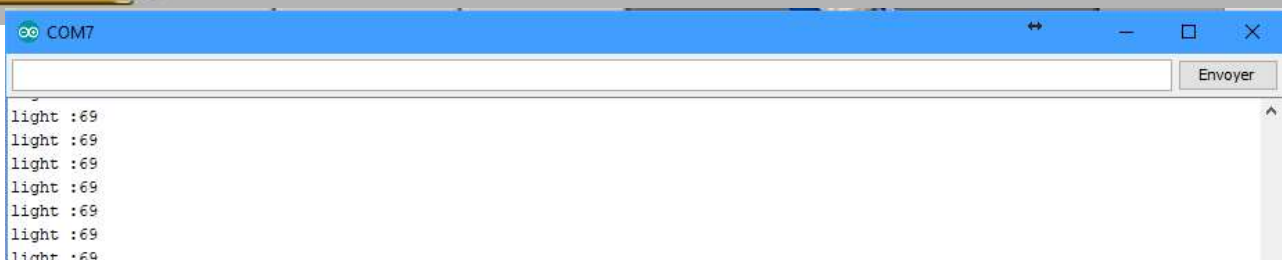
if valeur lumière



### Utilisation avec



Sur Ardublock Education les valeurs vont de 0 (quand il fait noir) à 759 en pleine lumière.



## Le servomoteur à rotation continue



**!!!!Attention au sens de connexion !!!!**

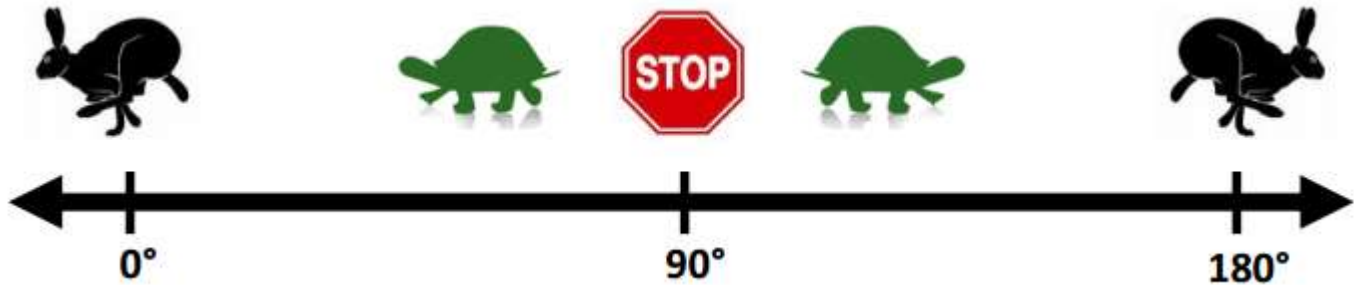
Connexion sur les ports : D3, D5, D6, D9, D10 ou D11. (PWM)

Chaque servomoteur consomme un certain

courant, il faudra faire attention à ne pas dépasser les 500mA.

La valeur de l'angle a une grande importance sur le déplacement :

elle permet de régler le sens et la vitesse de déplacement :

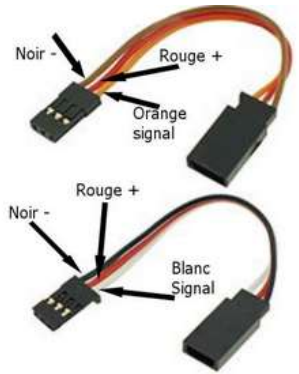


Pour arrêter un moteur il faut connaître son angle de « point mort » ou « d'arrêt ».

Pour ça ouvrir avec Ardublock Augmenté le fichier : « repérage valeur Angle arrêt sevo360 avec moniteur Serie.abp »

Dans l'exemple qui suit, on peut voir que lorsqu'on appuie sur le bouton poussoir sur la broche D2, les 2 servomoteurs à rotation continue s'arrêtent (valeur d'arrêt = 94°).

Sinon, ils tournent.



```
quand [drapeau vert] est cliqué
répéter indéfiniment
  si Lire l'état logique [du bouton poussoir] sur la broche [D2] = 1 alors
    orienter le servo-moteur de la broche [6] à un angle de [94] °
    orienter le servo-moteur de la broche [5] à un angle de [94] °
  sinon
    orienter le servo-moteur de la broche [6] à un angle de [30] °
    orienter le servo-moteur de la broche [5] à un angle de [160] °
```

Ces valeurs de rotations permettent à la maquette de faire tourner le moteur gauche à la même vitesse que le moteur droit, pour aller tout droit.

On obtient ces valeurs en essayant plusieurs valeurs.

Le bloc ci-dessous ne fonctionne pas encore parfaitement dans la version 3.4.11...à vérifier.

Définir la vitesse du servo-moteur continu de la broche [D3] à [255] dans le sens [normal] (trim: [0])

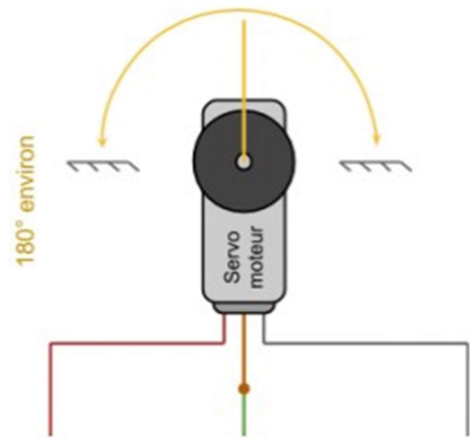


## Le servomoteur à rotation angulaire 180°

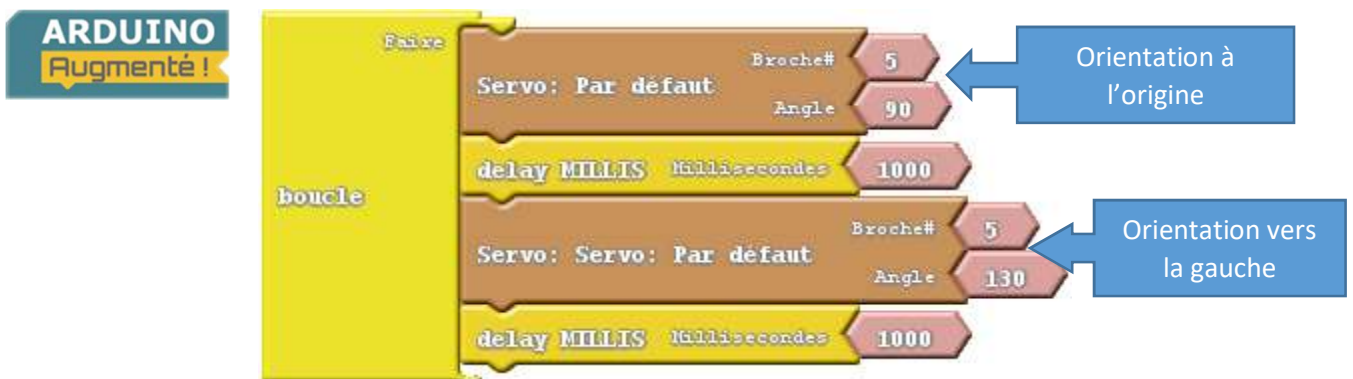
Connexion sur les ports : D3, D5, D6, D9, D10 ou D11. (PWM ou ~)

Chaque servomoteur consomme un certain courant, il faudra faire attention à ne pas dépasser les 500mA.

- Amplitude angle théorique : 0 à 180°
- Amplitude angle réelle : 8~11° à 169~172°



On retiendra les angles suivants : Angle de 15° à 165°




## Le module Bluetooth

Le module Bluetooth agit sur la carte Arduino **en envoyant et en recevant des données par la ligne Tx et Rx**. Il faut donc dire à la carte Arduino sur quelle broche il va recevoir les données du Bluetooth et sur quelle broche il va envoyer des données.

Dans l'exemple :

Tx de la carte Bluetooth est relié/indiqué à Rx sur Ardublock => broche D8

Rx de la carte Bluetooth est relié/indiqué Tx sur Ardublock => broche D9


|   |  |
|---|--|
|  | Après quelques essais avec mBlock 3.4.11 je n'ai pas réussi à utiliser le mode connecté pour faire apparaître les informations reçues.<br>Ceci dit, on utilisera le Bluetooth pour débrancher la carte Arduino et lui envoyer des données <b>pour le commander sans ordinateur !</b> |
|---|--|

### Principe d'utilisation :

On programmera la carte Arduino avec Ardublock. Vous trouverez des exemples avec la version de Technologie Services mais aussi avec celle de DuinoEdu.

Celle-ci attendra certains caractères, par exemple « A » pour allumer.

On programmera ensuite une application avec ApplInventor du MIT pour envoyer ces caractères.

| Coté <b>ARDUINO</b> de DuinoEdu  |  |
|--|--|
| Il ya plus de blocs, ils permettent de tout régler par les blocs : <b>désignation de la variable BT, réglages de la vitesse de réception, lecture des informations reçues.</b> |  |
|   |  |

On peut alors ajouter une action si l'on reçoit un caractère A => allumage d'une DEL,

Le caractère 2 => allumage d'une autre LED...

Il ne faut que ces blocs.



Voir : <http://lara.jteknno.info/ardublock-education-bluetooth-appinventor-une-solution/>

Il faudra cependant modifier la vitesse de réception dans la ligne de code et mettre 9600 sinon ça ne marche pas !

```
sketch_nov29a | Arduino 1.6.11
Fichier Édition Croquis Outils Aide
sketch_nov29a $
#include <TSSerial.h>

TSSerial bt_2_3;

void setup()
{
  Serial.begin(9600);
  bt_2_3.init(2, 3, 115200);
}

void loop()
{
  Serial.print("BT :");
  Serial.print(bt_2_3.receive(2, 3));
  Serial.println();
  delay( 100 );
}
```

```
sketch_nov29a | Arduino 1.6.11
Fichier Édition Croquis Outils Aide
sketch_nov29a $
#include <TSSerial.h>

TSSerial bt_2_3;

void setup()
{
  Serial.begin(9600);
  bt_2_3.init(2, 3, 9600);
}

void loop()
{
  Serial.print("BT :");
  Serial.print(bt_2_3.receive(2, 3));
  Serial.println();
  delay( 100 );
}
```

## Coté application sur Android avec AppInventor



Il est nécessaire d'avoir au préalable un compte google pour se connecter sur les serveurs de création du MIT.

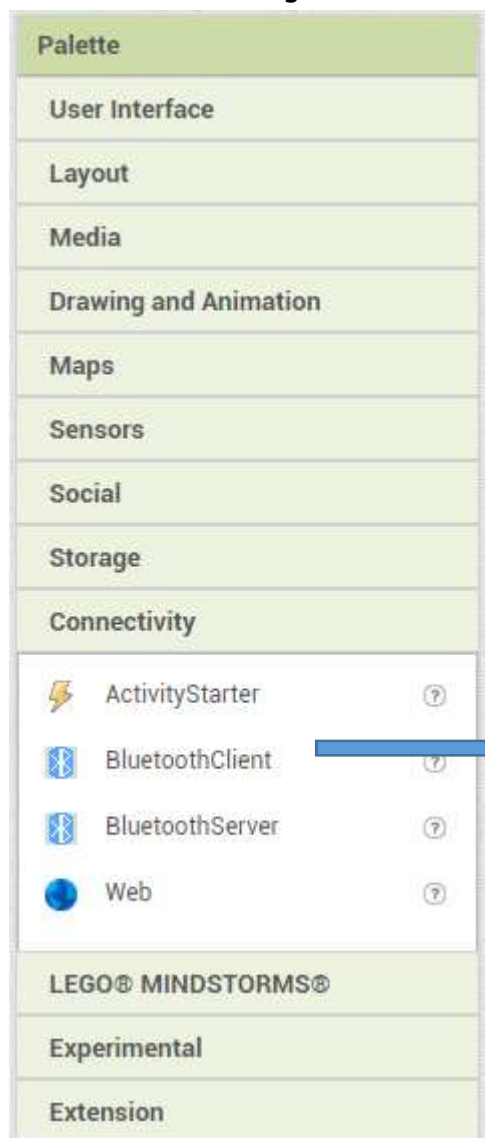
Il existe dans les collèges la possibilité de créer un réseau similaire et qui permet aux professeurs et aux élèves de ne pas avoir à gérer les mails...

Ces blocks permettent d'avoir un menu pour appairer le smartphone et le module de la carte Arduino.

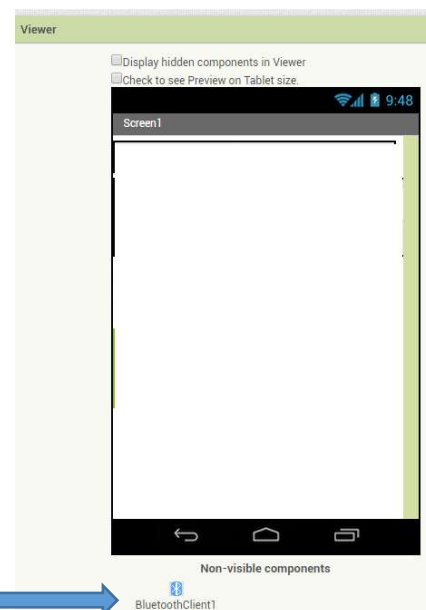
En mode



Dans le menu de gauche (Palette), il faut ajouter le BluetoothClient



Et le faire glisser





En mode

Designer Blocks

Et à partir du menu :

The screenshot displays the Xcode IDE interface. On the left, the 'Component Picker' is visible, showing a hierarchy of components. The 'BluetoothClient1' component is highlighted with a green circle. Below the component picker is a 'Media' section with a file named 'aide.jpg' and an 'Upload File ...' button. On the right, the 'Swift' code editor shows the implementation of the 'BluetoothClient1' component. The code is organized into a 'do' block, which contains several method calls to the 'BluetoothClient1' object. The methods include 'BytesAvailableToReceive', 'Connect', 'ConnectWithUUID', 'Disconnect', 'IsDevicePaired', and various 'Receive' methods for different data types (Signed/Unsigned Byte, 2-Byte, 4-Byte, and Text).

```
when BluetoothClient1 .BluetoothError
  functionName message
do
  call BluetoothClient1 .BytesAvailableToReceive
  call BluetoothClient1 .Connect
    address
  call BluetoothClient1 .ConnectWithUUID
    address
    uuid
  call BluetoothClient1 .Disconnect
  call BluetoothClient1 .IsDevicePaired
    address
  call BluetoothClient1 .ReceiveSigned1ByteNumber
  call BluetoothClient1 .ReceiveSigned2ByteNumber
  call BluetoothClient1 .ReceiveSigned4ByteNumber
  call BluetoothClient1 .ReceiveSignedBytes
    numberOfBytes
  call BluetoothClient1 .ReceiveText
    numberOfBytes
  call BluetoothClient1 .ReceiveUnsigned1ByteNumber
  call BluetoothClient1 .ReceiveUnsigned2ByteNumber
  call BluetoothClient1 .ReceiveUnsigned4ByteNumber
```



fait la liste de bluetooth disponible  
avant le relâchement du bouton à l'écran

```
when Btn_activation .BeforePicking
do
  set Btn_activation .Elements to BluetoothClient1 .AddressesAndNames
```

après avoir lâché, on choisit le  
module BT et on appaire

```
when Btn_activation .AfterPicking
do
  if
    call BluetoothClient1 .Connect
    address Btn_activation .Selection
  then
    set Btn_activation .Enabled to false
    set BTN_desactivation .Enabled to true
    set Btn_activation .TextColor to [red]
    set BTN_desactivation .TextColor to [blue]
    call Avertissement_notif .ShowMessageDialog
    message "Connexion"
    title "Activée"
    buttonText "OK"
```

désactivation du bouton.  
Activation!

```
when BTN_desactivation .LongClick
do
  call BluetoothClient1 .Disconnect
  set Btn_activation .Enabled to true
  set BTN_desactivation .Enabled to false
  set Btn_activation .TextColor to [blue]
  set BTN_desactivation .TextColor to [red]
  call Avertissement_notif .ShowMessageDialog
  message "Déconnexion"
  title "Bluetooth"
  buttonText "OFF"
```



Vous pouvez alors ajouter des actions quand on va pické (touché du doigt les différents boutons préalablement intégré à l'écran « Designer » , par exemple envoi du nombre 1

Avec ceci :

```
when Bt1 .Click
do
  call BluetoothClient1 .Send1ByteNumber
  number "1"
```