

Utilisation de

ARDUBLOCK

PROGRAMMATION ARDUINO

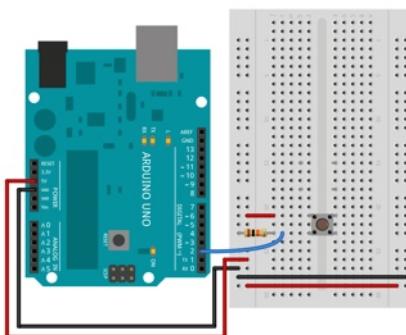


Il y a deux façons de programmer les cartes Arduino.

- Soit avec du code «pur» en ligne de code type langage C ;
- Soit sous forme de blocs type puzzle, nommé Ardublock.

C'est cette deuxième méthode qui est la plus adaptée en collège et que nous allons traiter.

Il est tout à fait possible de partir d'une carte arduino neutre achetée sur n'importe quel fournisseur de composants d'électroniques. Mais ceci vous obligera à adapter l'électronique d'entrée (capteur) et de sortie (actionneur) à la carte via une platine de test ou de montages plus ou moins bricolés, comme le montre la photo ci-contre.



Carte
Arduino Uno
«basique»

COMPATIBILITÉ



AutoProgUno



Shield Grove
DuinoEdu

Autre solution, passer par des fournisseurs qui ont didactisés la carte arduino ainsi que les capteurs/actionneurs. Exemple avec la société A4 ou DuinoEdu.

Ce petit didacticiel, n'est en aucun cas présent pour faire l'apologie d'un fournisseur en particulier, mais juste pour vous permettre de débuter rapidement avec le système Arduino via Ardublock.

The image shows two side-by-side screenshots of Arduino development environments. On the left is the standard Arduino IDE interface, showing a text-based code editor with the following sketch:

```
sketch_nov19a | Arduino 1.0.6
sketch_nov19a §
void setup()
{
pinMode( 3 , OUTPUT);
}

void loop()
{
digitalWrite( 3 , HIGH );
}
```

A callout bubble points to this window with the text: "Interface de programmation Arduino en ligne de code".

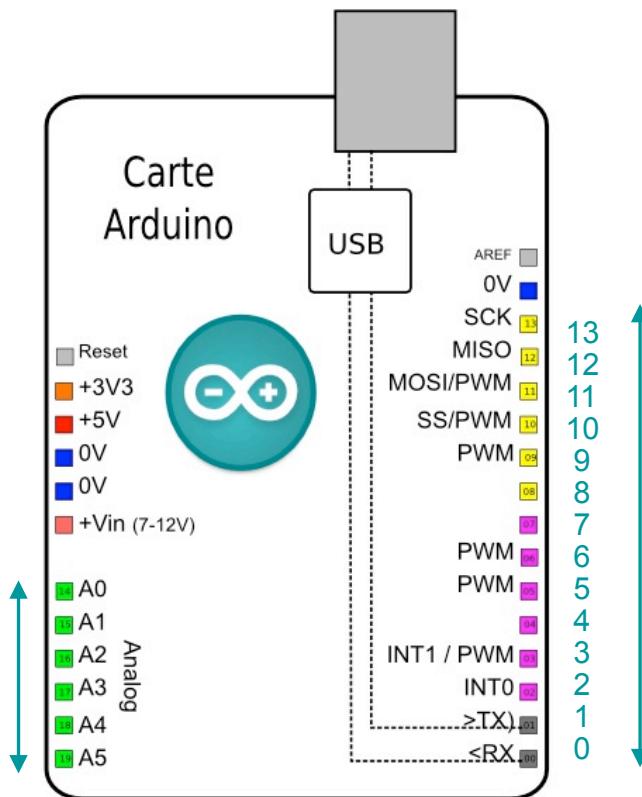
On the right is the Ardublock interface, showing a graphical block-based editor. A specific block is highlighted: "boucle Faire" followed by "Fixe la sortie numérique au niveau 3 HIGH". The Ardublock library browser sidebar lists categories like Contrôles, Broches, Variables/Constantes, Operateurs Math, Utilitaires, Blobs TinkerKit, DFRobot, Seeedstudio Grove, and Linker Kit. A callout bubble points to this window with the text: "Interface de programmation Ardublock".

LA CARTE

Source image :

http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php

6 entrées analogiques
De A0 à A5



INSTALLATION DU LOGICIEL

Téléchargez / Installez l'interface de programmation Arduino directement sur le site officiel :

<http://arduino.cc/en/Main/Software>

Il vous faudra ensuite installer le plugin Ardublock. Téléchargeable ici par exemple :

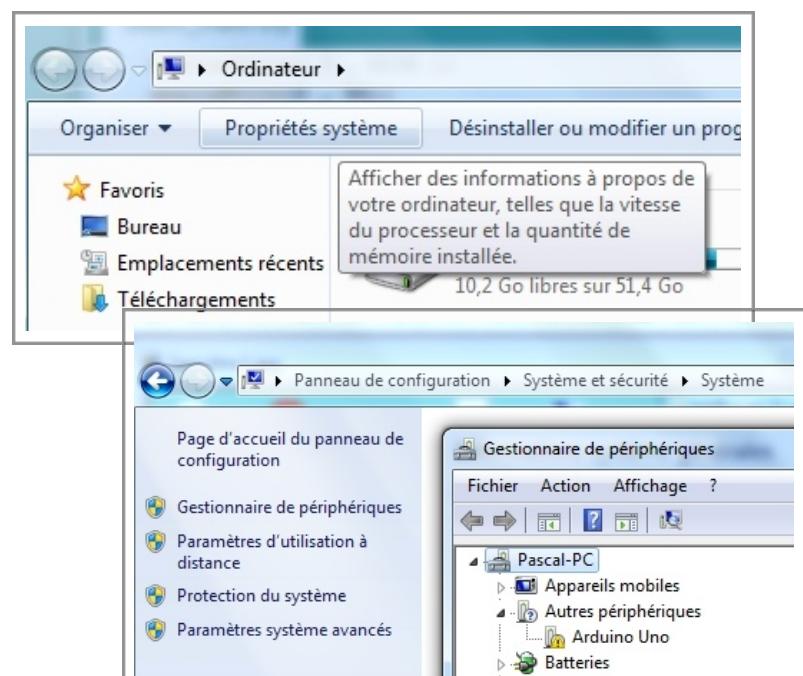
<http://sourceforge.net/projects/ardublock/>

Ou sinon utilisé l'interface augmentée de DuinoEdu qui ne nécessite pas d'installation (possibilité de l'utiliser sur une clé USB par exemple). Le plugin Ardublock est inclus ainsi que les librairies utiles pour l'afficheur, servomoteur par exemple.

<http://duinoedu.com/telechargement.html>

Branchez ensuite votre carte arduino à l'ordinateur via un port USB. La carte n'est pas prise en compte, il faut donc installer le driver adéquat.

Pour cela rendez-vous dans votre vos Propriétés systèmes puis dans votre gestionnaire de périphériques.

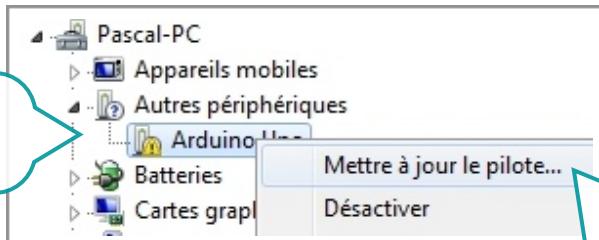


Le site officiel : <http://arduino.cc/>

Avec des exemples de projets, un forum et une communauté très active.

1

Clic droit sur le périphérique inconnu



2

Mettre à jour le pilote

Comment voulez-vous rechercher le pilote ?

- Rechercher automatiquement un pilote mis à jour
Windows va rechercher sur votre ordinateur et sur Internet le pilote le plus récent pour votre périphérique, sauf si vous avez désactivé cette fonctionnalité dans les paramètres d'installation du périphérique.

- Rechercher un pilote sur mon ordinateur
Recherchez et installez manuellement le pilote.

3

Rechercher un pilote sur mon ordinateur



Rechercher le pilote sur votre ordinateur

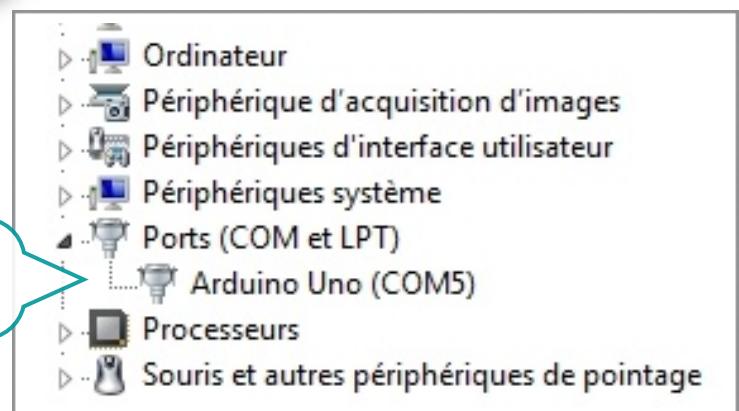


5

Valider et suivre les instructions jusqu'à l'installation correcte du driver.

6

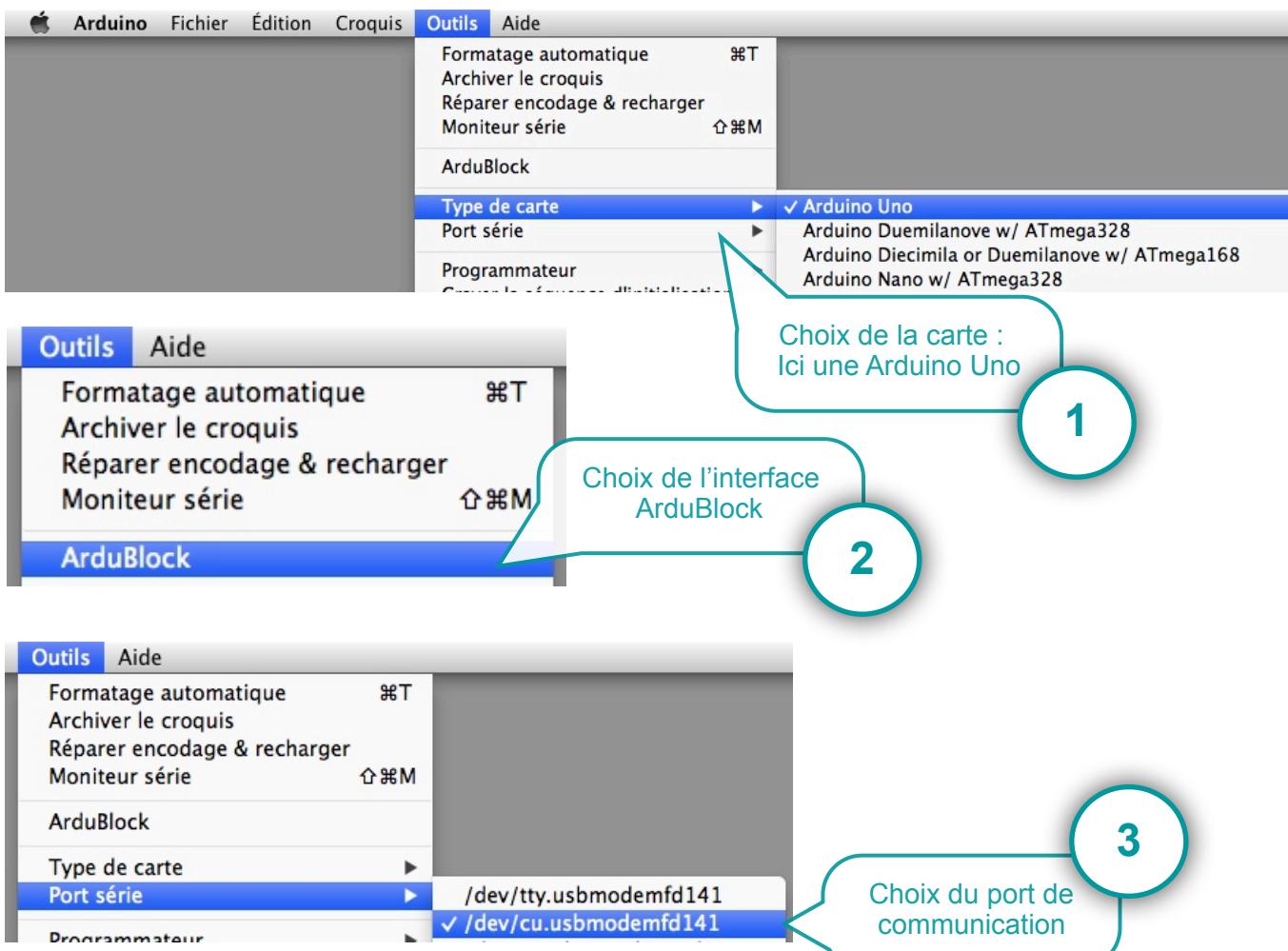
Votre carte Arduino est maintenant reconnue.



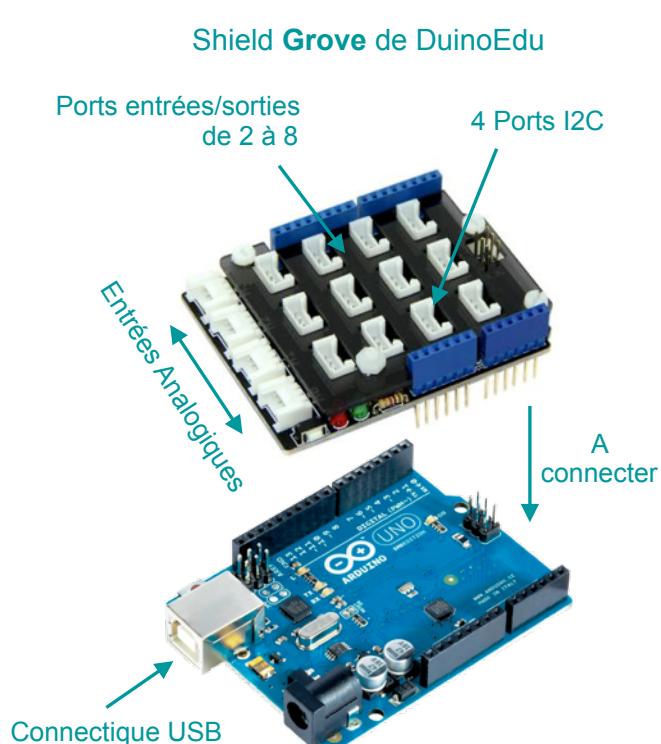
START

Nous pouvons commencer à programmer et utiliser la carte Arduino.

L'INTERFACE DE PROGRAMMATION

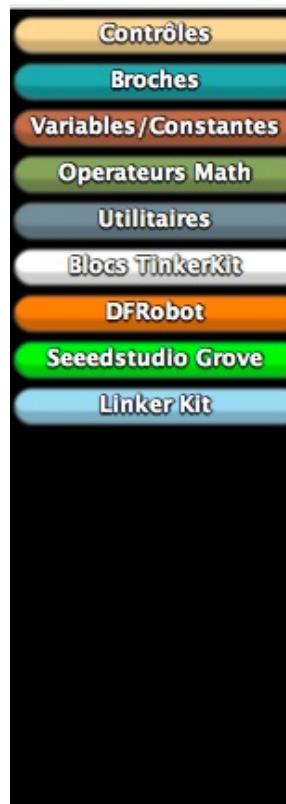


Interface AutoProg UNO de A4



MON 1ER PROGRAMME

Allumer continuellement une diode sur le Port 3.
(Le Port 3 devient alors automatiquement une sortie)



1



2



3

Indiquez le Port 3

Pour allumer la lampe
indiquez l'état HIGH
(état haut ou 1 logique)

4

Contrairement à LOW (état
bas 0 logique)



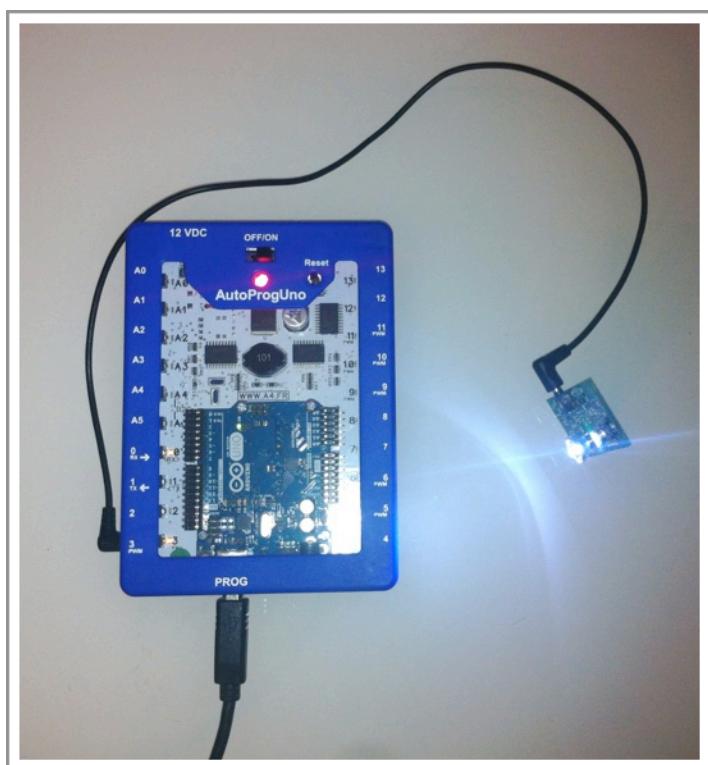
5

Transférez le programme

6

Il se convertit en ligne de
code pour être transféré
dans le microcontrôleur.

```
sketch_nov19a §
void setup()
{
pinMode( 3 , OUTPUT);
}
void loop()
{
digitalWrite( 3 , HIGH );
}
```

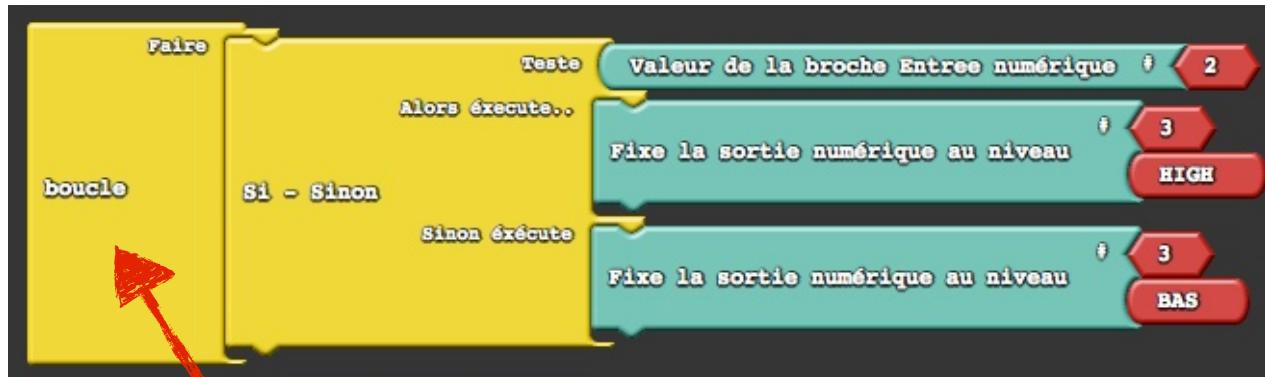


Vérification :

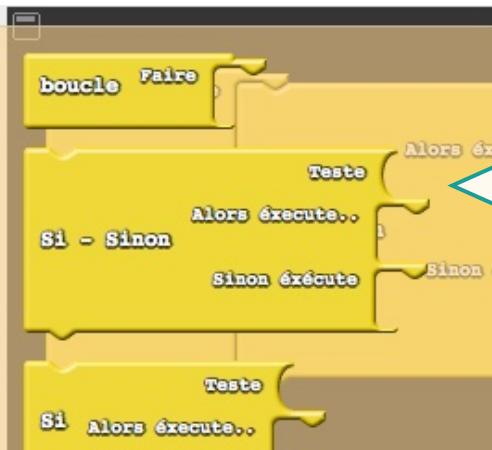
La DEL connectée au Port 3 reste
continuellement allumée.

MON 1ER TEST

Allumer une lampe sur le Port 3 uniquement si le Port 2 est activité.
La DEL est éteinte dans le cas contraire.
(Le Port 2 devient alors une entrée)

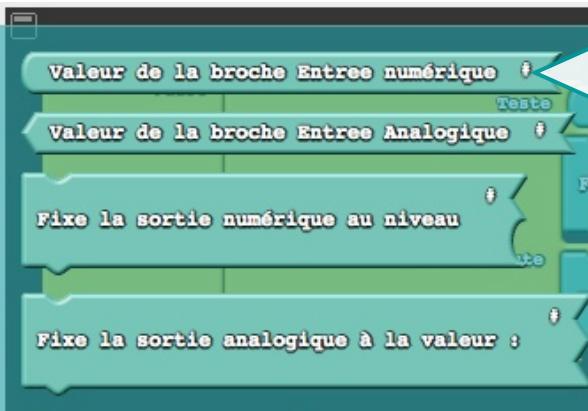


**Ne pas oublier la
boucle de départ**



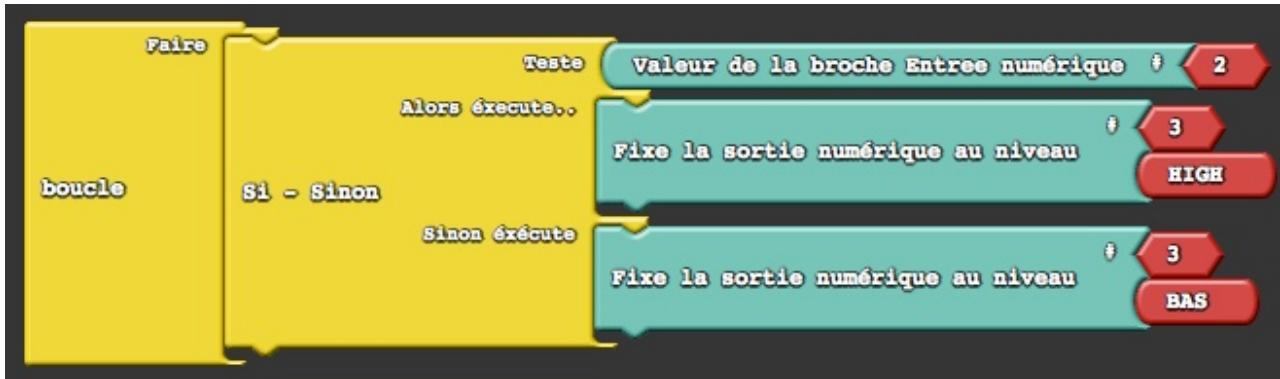
Dans les blocs de
Contrôles :

Glissez / Déposez le bloc
«Si - Sinon»



Dans les blocs de Broches :

Glissez / Déposez le bloc «Valeur
de la broche Entrée numérique»

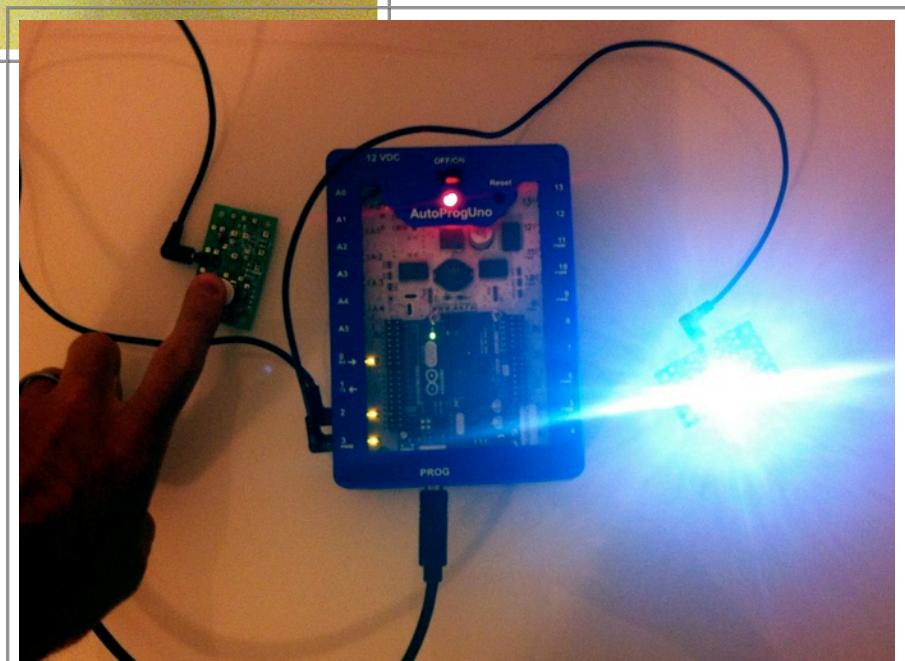
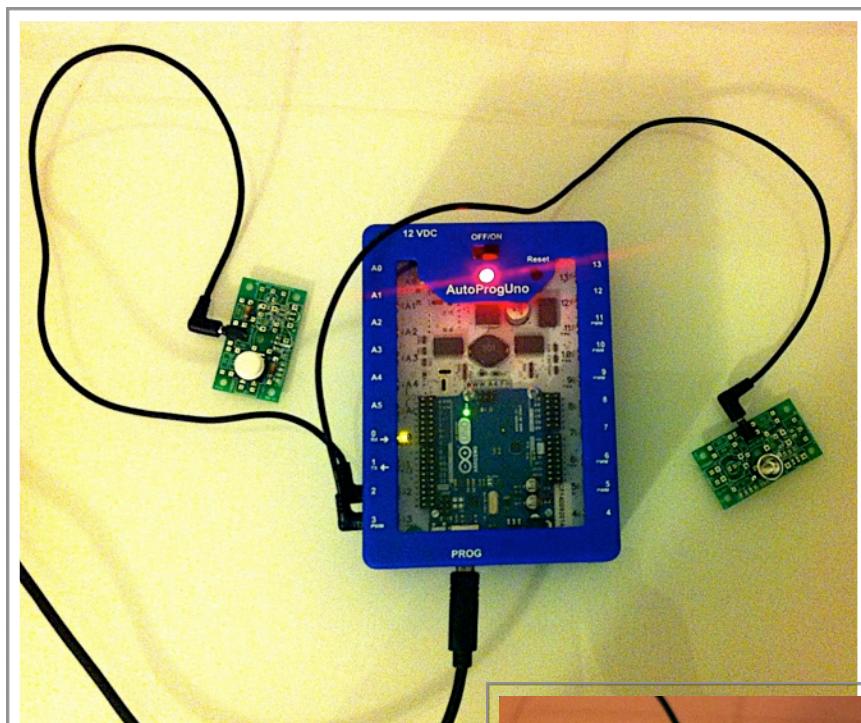


Conversion du programme automatiquement en ligne de code

sketch_nov19b §

```
void setup()
{
pinMode( 3 , OUTPUT);
pinMode( 2 , INPUT);
}

void loop()
{
if (digitalRead( 2))
{
digitalWrite( 3 , HIGH );
}
else
{
digitalWrite( 3 , LOW );
}
}
```



PROGRAMMATION AVEC FONCTION « ET »

Une DEL s'allume si 2 entrées sont activées :
Port 2 **et** Port 12.
Dans le cas contraire la DEL est éteinte (Port 3).

Port 2	Port 12	Port 3
Entrée : Interrupteur	Entrée : Capteur de présence	Sortie : DEL
0	0	0
0	1	0
1	0	0
1	1	1



Ne pas oublier la boucle de départ



Dans les blocs :
Opérateurs Tests

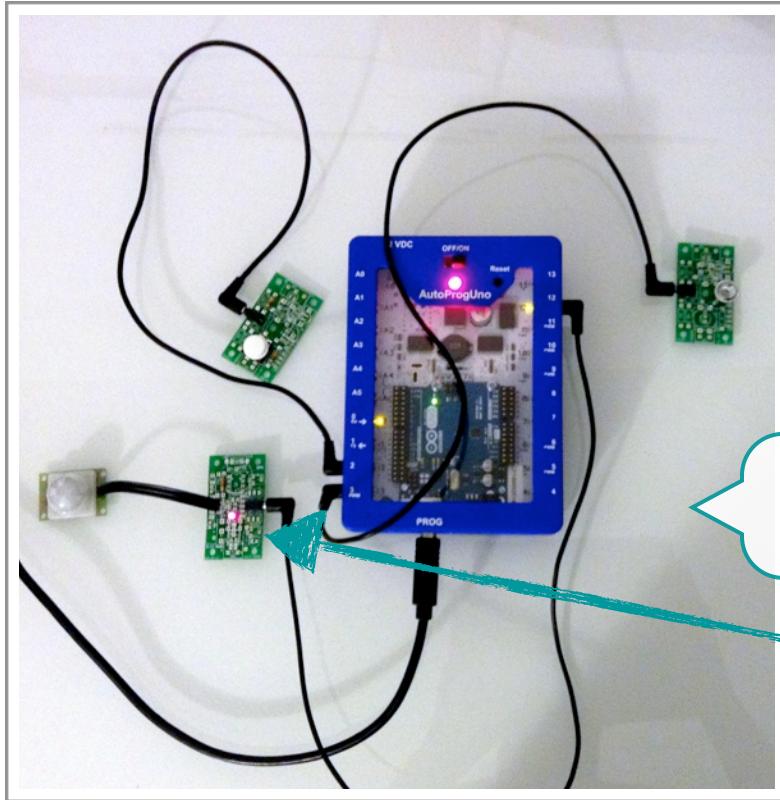
Glissez / Déposez le bloc
«ET»

Conversion du programme automatiquement en ligne de code

sketch_nov19b §

```
void setup()
{
pinMode( 3 , OUTPUT);
pinMode( 12 , INPUT);
pinMode( 2 , INPUT);
}

void loop()
{
if (( digitalRead( 2) && digitalRead( 12 ) ))
{
digitalWrite( 3 , HIGH );
}
else
{
digitalWrite( 3 , LOW );
}
}
```

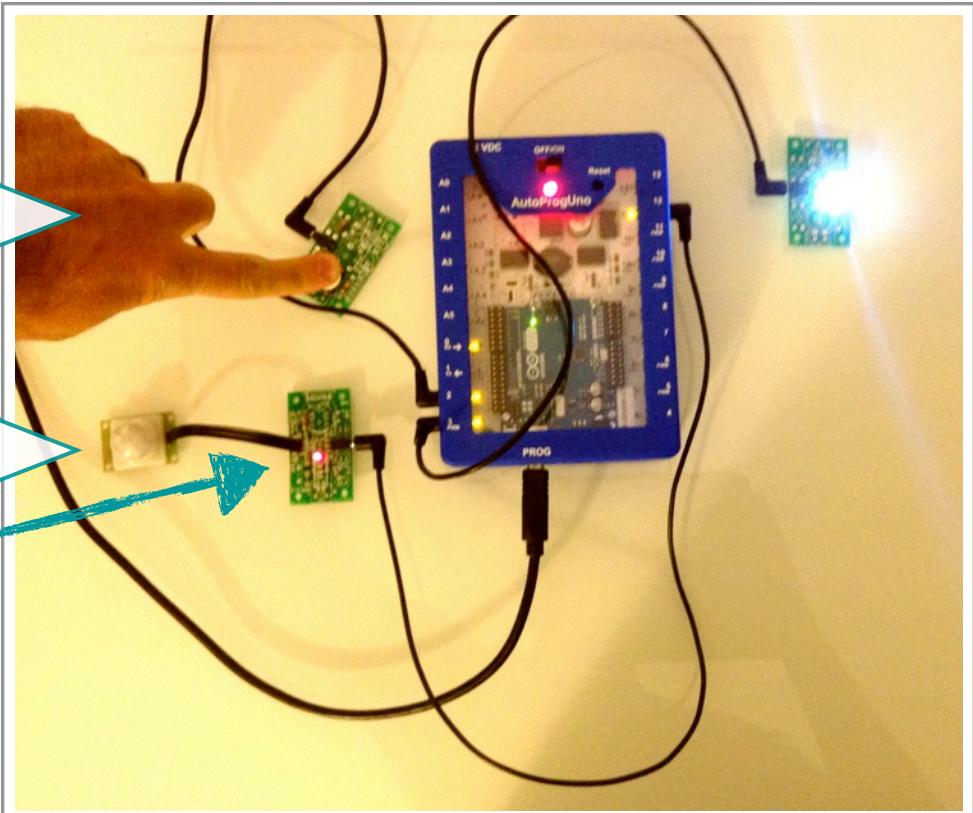


Détection uniquement par le capteur de présence.

Bouton poussoir activé.

ET

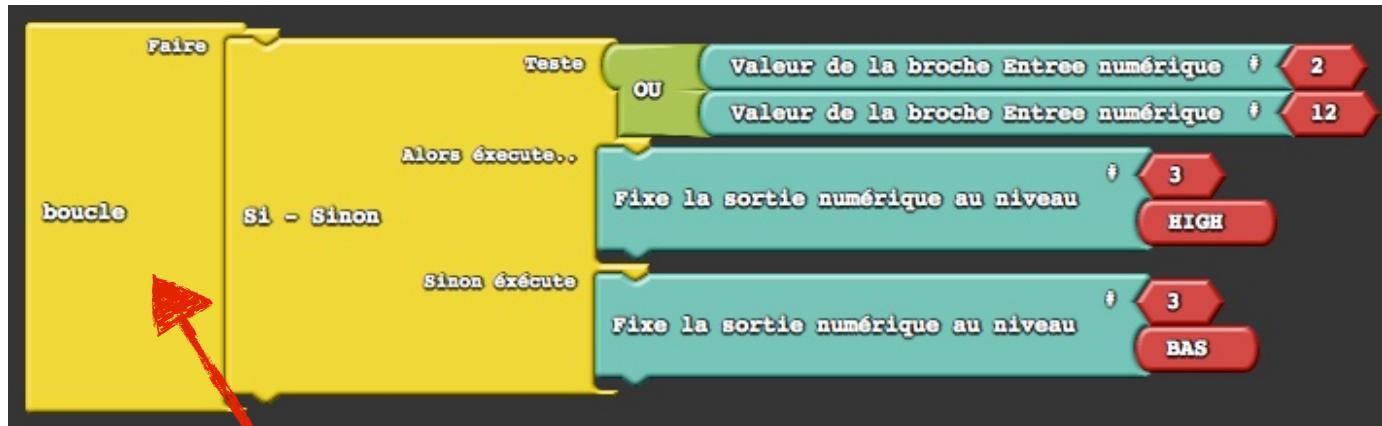
Détection par le capteur de présence.



PROGRAMMATION AVEC FONCTION « OU »

Une lampe s'allume si une des 2 entrées est activée :
Port 2 ou Port 12.
Dans le cas contraire la lampe est éteinte (Port 3).

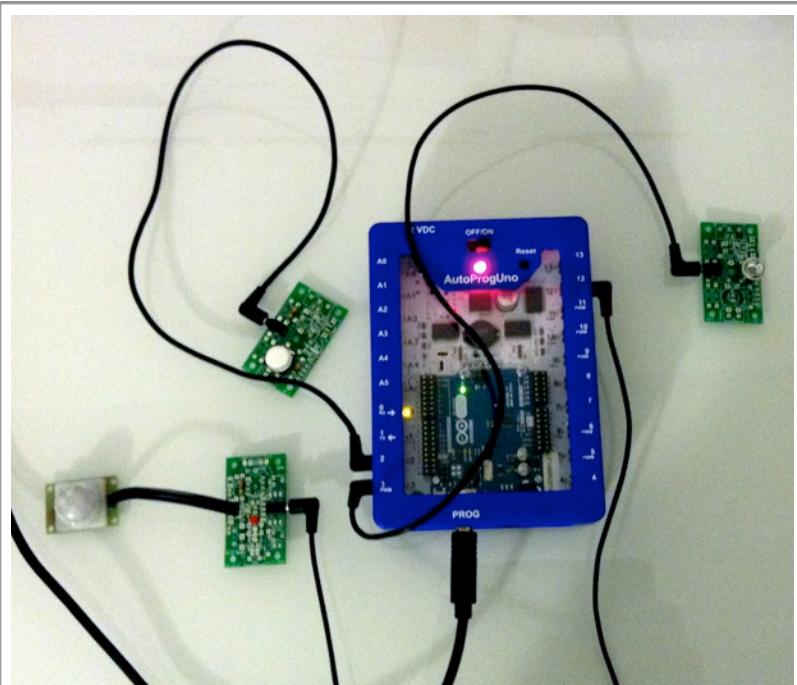
Port 2	Port 12	Port 3
Entrée : Interrupteur	Entrée : Cp de présence	Sortie : Lampe
0	0	0
0	1	1
1	0	1
1	1	1



Ne pas oublier la
boucle de départ



Conversion du programme automatiquement en ligne de code

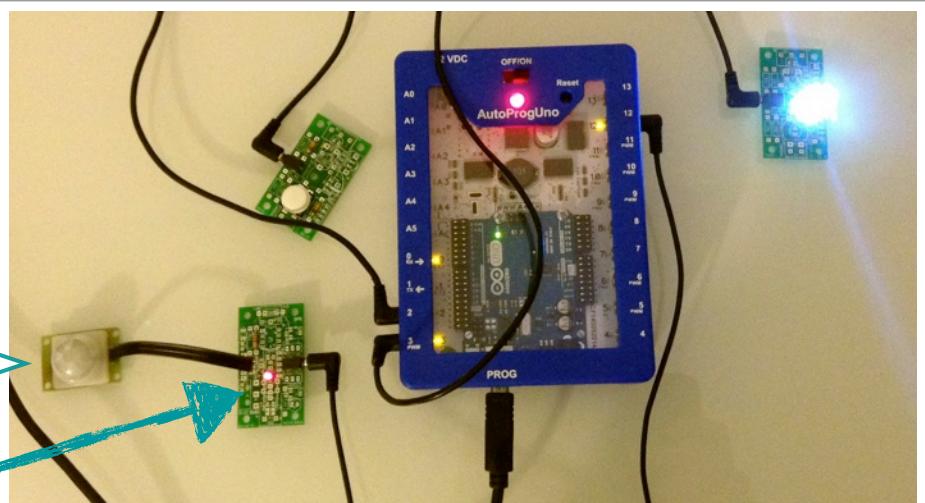


sketch_nov19b §

```
void setup()
{
pinMode( 3 , OUTPUT);
pinMode( 12 , INPUT);
pinMode( 2 , INPUT);
}

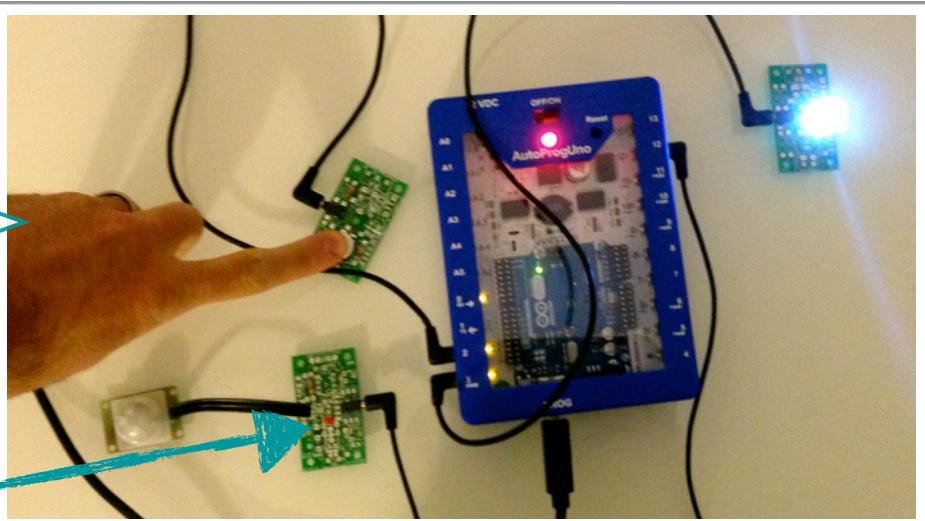
void loop()
{
if (( digitalRead( 2 ) || digitalRead( 12 ) ))
{
digitalWrite( 3 , HIGH );
}
else
{
digitalWrite( 3 , LOW );
}
}
```

Détection uniquement par le capteur de présence.



OU

Bouton poussoir activé uniquement.



UTILISATION DES ENTRÉES ANALOGIQUES

Les valeurs d'entrées sur les ports analogiques sont immédiatement codées en valeurs numériques sur 10 bits.

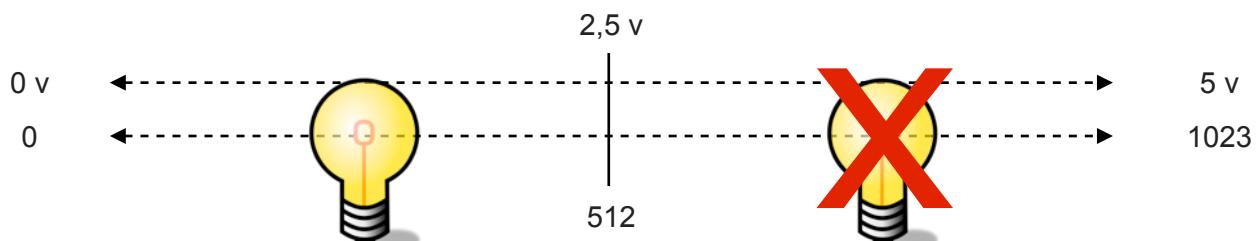
Décimal	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	1	0
...
1023	1	1	1	1	1	1	1	1	1	1



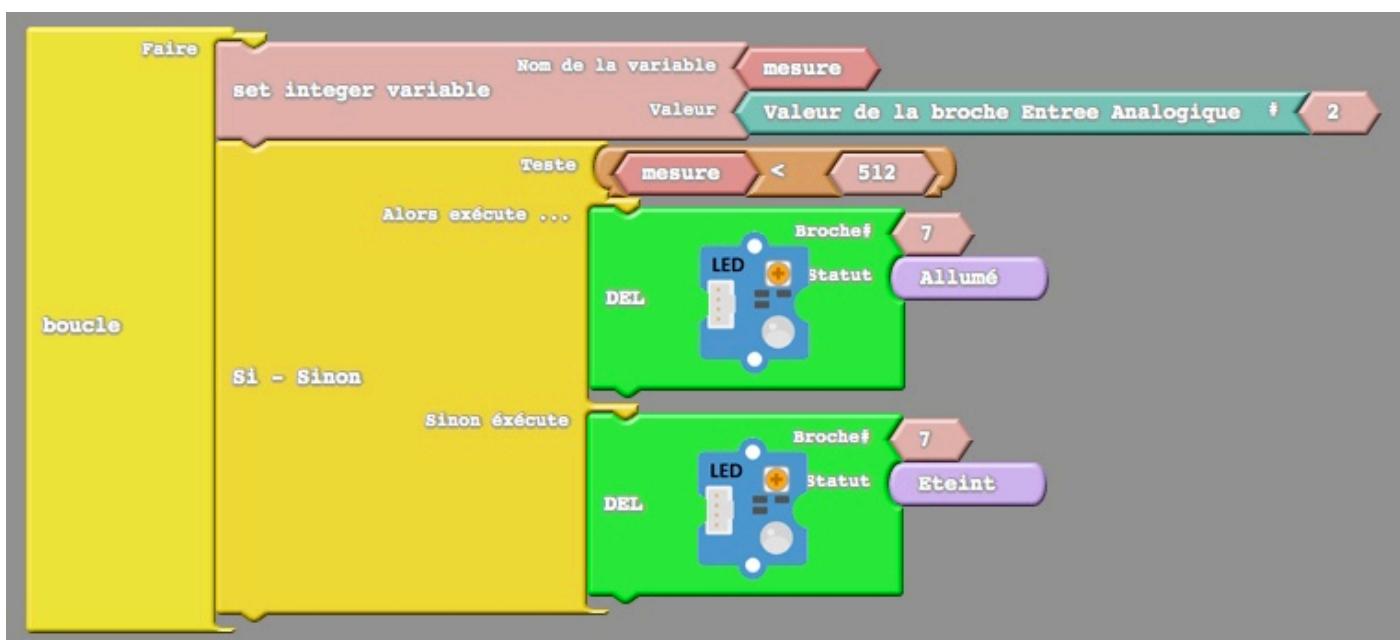
Un capteur de luminosité (photodiode) peut être associé à un potentiomètre.

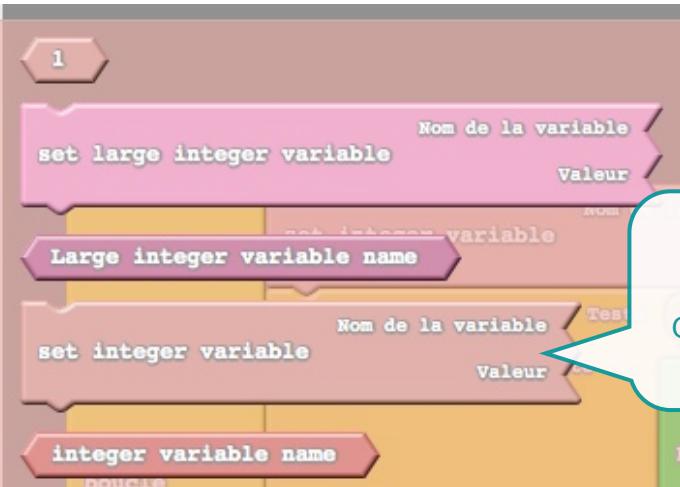
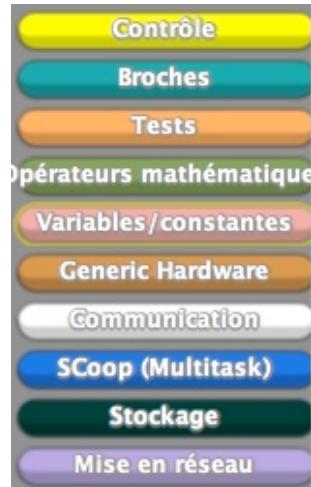
Les 5v que peut fournir le capteur sont découpés en 1024 valeurs.

Notre programme permet d'allumer une DEL si le potentiomètre est dans la première moitié de sa plage d'utilisation, c'est à dire entre 0v et 2,5v, soit entre 0 et 512 en numérique. La DEL est branchée sur le port 7 et le potentiomètre sur le port analogique 2.



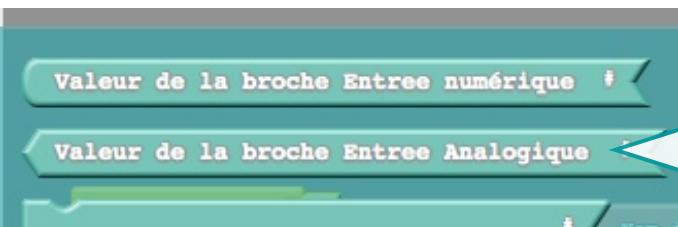
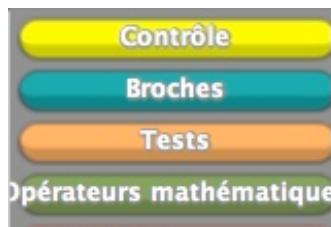
Pour pouvoir utiliser la valeur d'entrée (en numérique) du port analogique sur lequel est branché le potentiomètre, il faut enregistrer cette valeur dans une variable : que nous avons appeler «mesure» (attention éviter les accents, espaces, ...).





Dans les blocs :
Variables / contantes

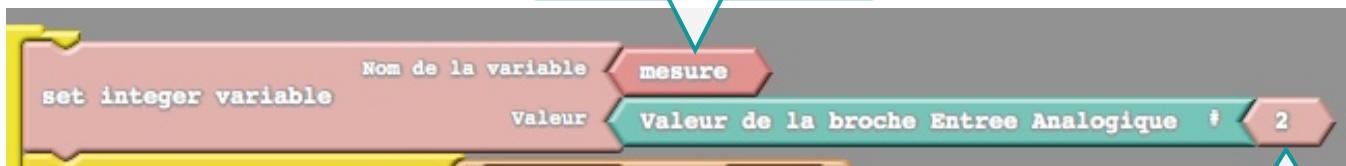
Glissez / Déposez le bloc
«Set interger variable»



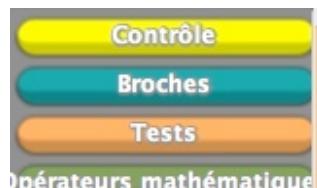
Dans les blocs :
Broches

Glissez / Déposez le bloc
«Valeur ... Analogique»

Renommez le nom de la
variable

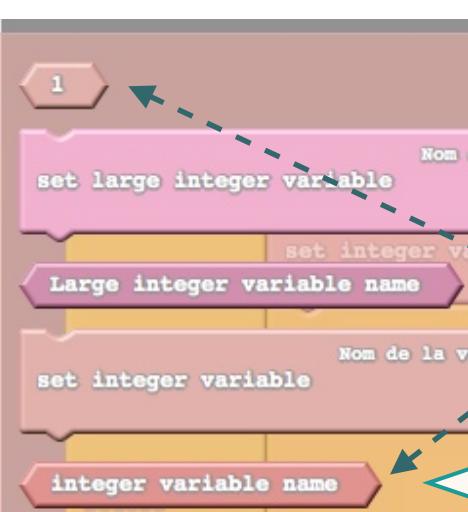


Indiquez le port
analogique utilisé



Dans les blocs :
Test

Glissez / Déposez le bloc
«... < ...»



Dans les blocs :
Variables / Contantes

Glissez / Déposez le bloc
«integer variable name»

Ainsi qu'un bloc numérique
qu'il faudra compléter par
«mesure» et «512»

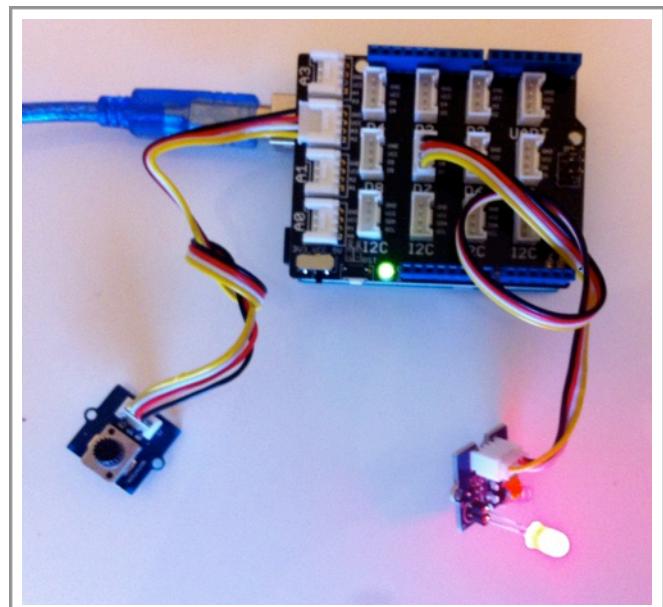
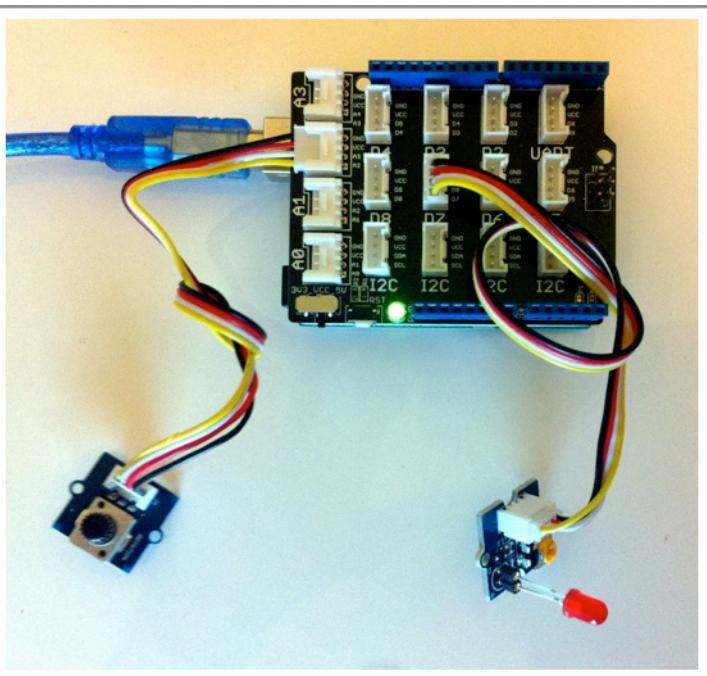
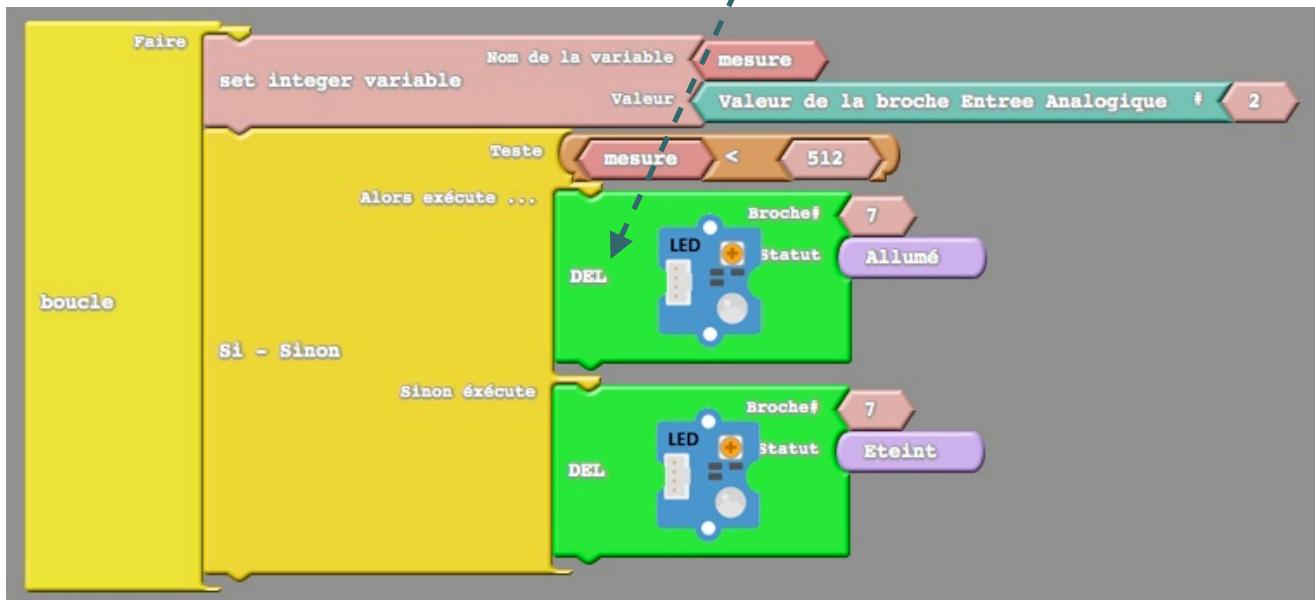
Contrôle
Broches
Tests
Opérateurs mathématique
Variables/constantes
Generic Hardware
Communication
SCoop (Multitask)
Stockage
Mise en réseau
Blocs de code
TinkerKit
DFRobot
Seeed Studio Grove
DuinoEDU Grove Add
Adafruit



Les blocs plus graphiques sont disponibles dans la librairie DuinoEdu Grove Add

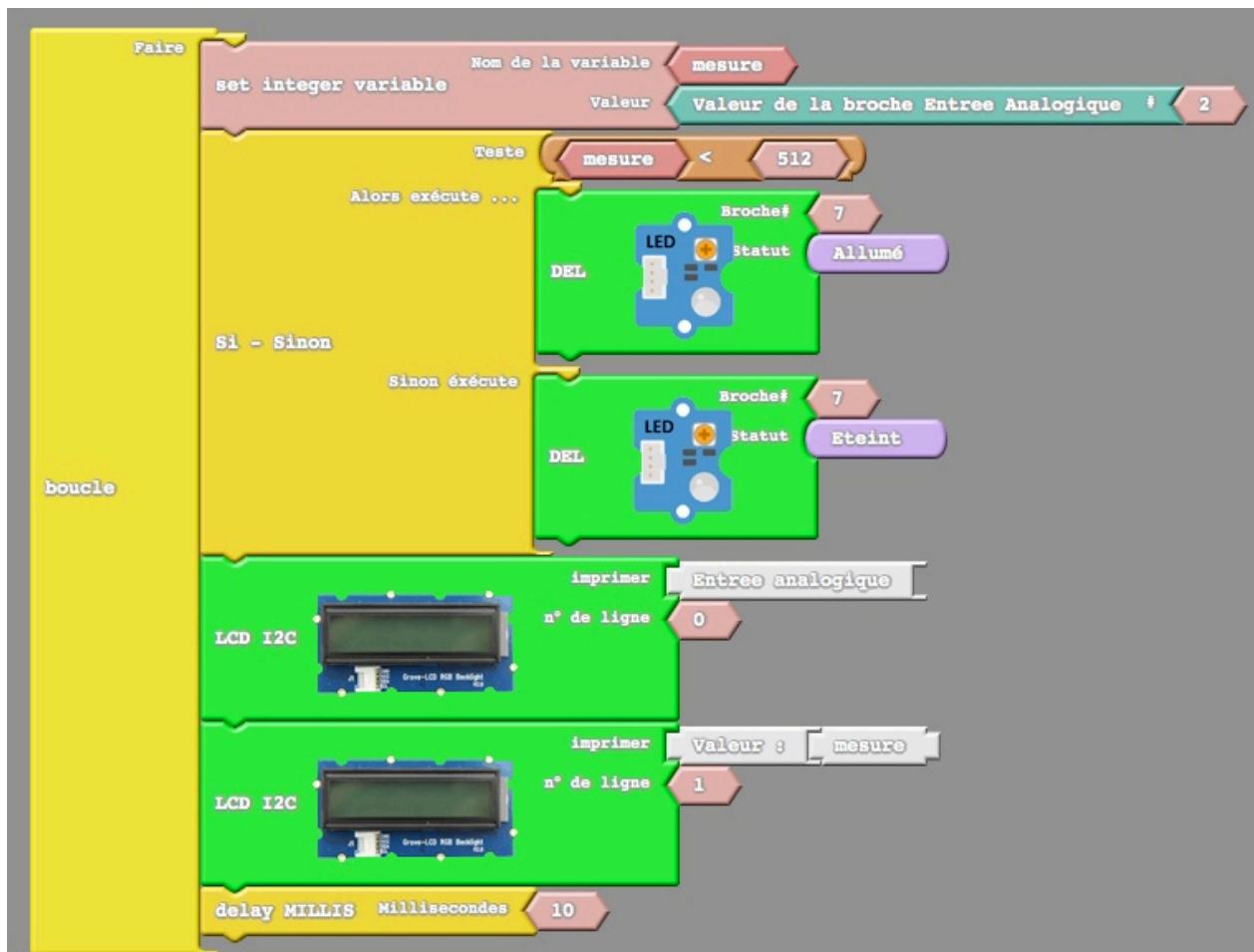
Fixe la sortie numérique au niveau
7 HIGH

Fonction identique

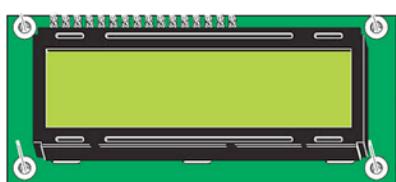


UTILISATION DE L'AFFICHEUR

Gardons le programme précédent mais affichons maintenant la valeur de l'entrée analogique enregistrée dans la variable «mesure». L'afficheur est branché sur un port I2C.



Dans les blocs :
DuinoEDU Grove Add
Glissez / Déposez le bloc
«LCD I2C»



Possibilité de piloter un afficheur LCD :
2 lignes / 16 caractères par ligne





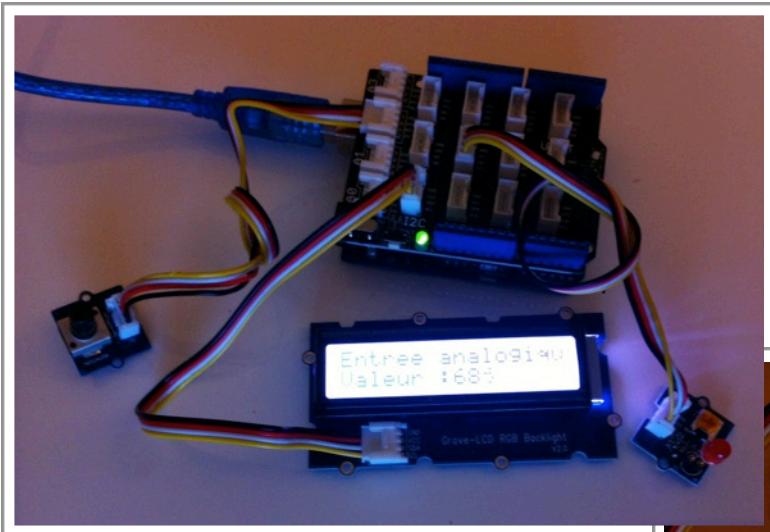
n° de la ligne = 0 alors écriture sur la ligne 1

n° de la ligne = 1 alors écriture sur la ligne 2



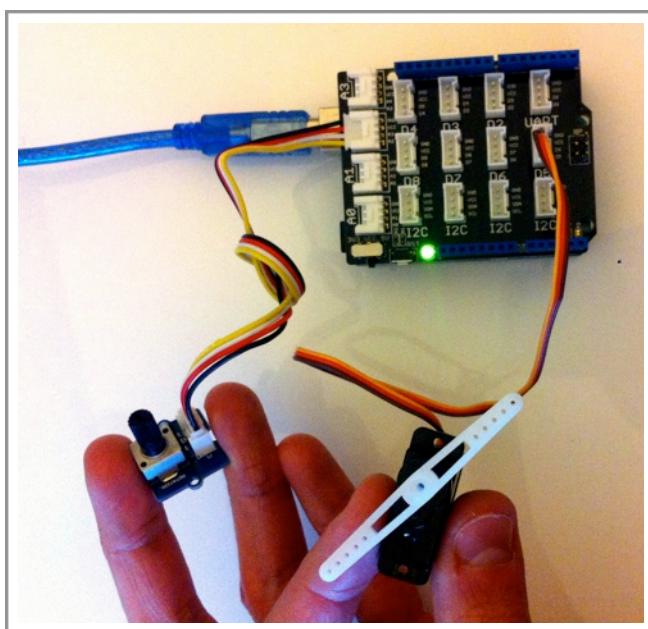
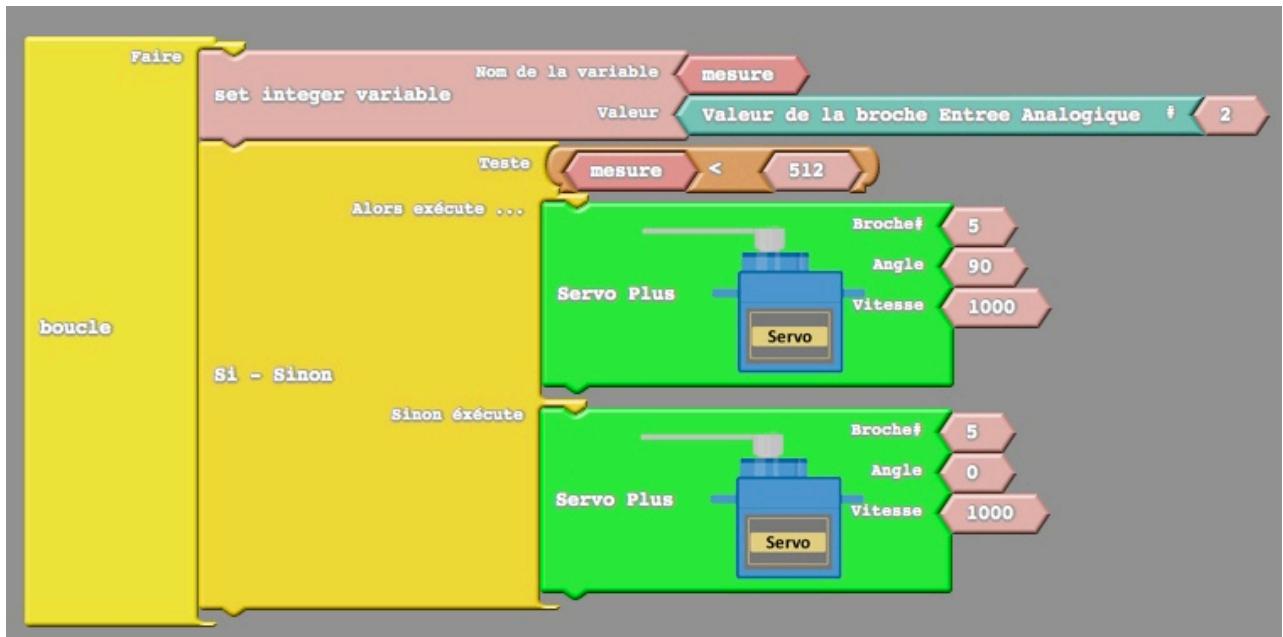
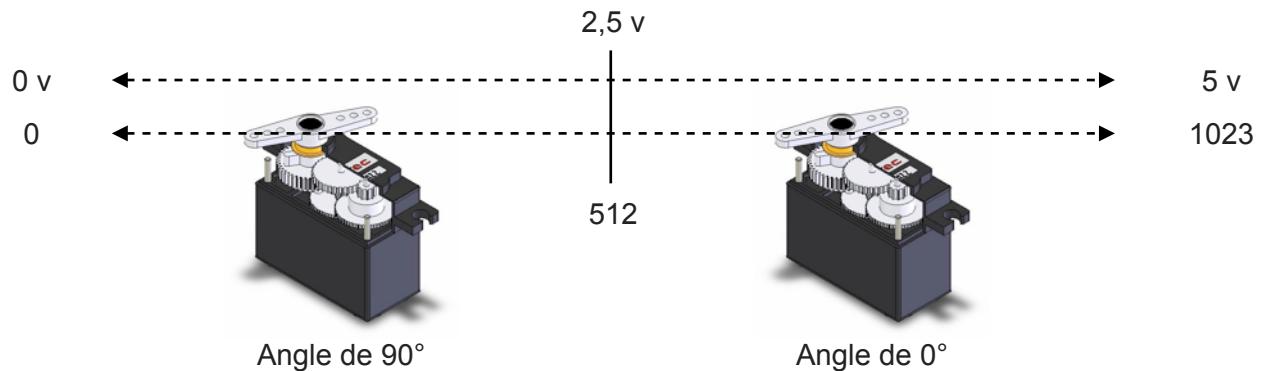
Dans les blocs :
Variable / constantes

Glissez / Déposez le bloc
«texte»
ou «String variable»



UTILISATION DU SERVOMOTEUR

Gardons le programme précédent mais plutôt d'indiquer la valeur analogique sur un afficheur, le potentiomètre pilote un servomoteur



Dans les blocs :
DuinoEDU Grove Add
Glissez / Déposez le bloc
« Servo Plus »

UTILISATION DES SORTIES ANALOGIQUES

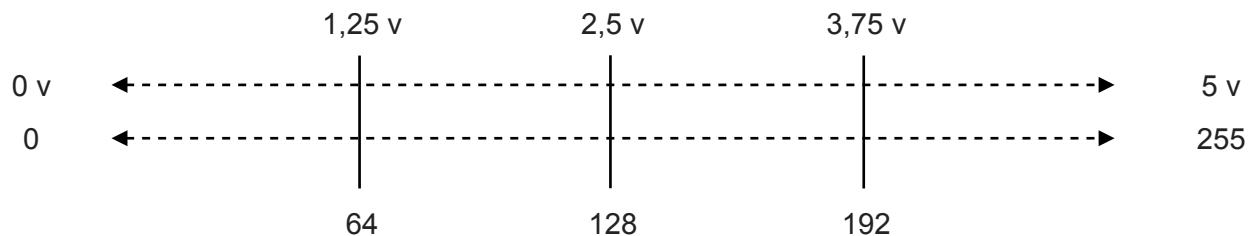
Quelques ports sont configurables en sorties analogiques, ils sont repérés par le symbole :

Mais comme pour les entrées analogiques, ils sont convertis en numériques sur 8 bits (contrairement aux entrées analogiques qui sont en 10 bits).



Donc seuls les **ports 3, 5, 6, 9, 10 et 11** peuvent fournir une sortie analogique.

Décimal	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1	0
...
255	1	1	1	1	1	1	1	1



Faisons le test sur le port 6 avec une tension de 3,75 volts soit 192 en valeur numérique.

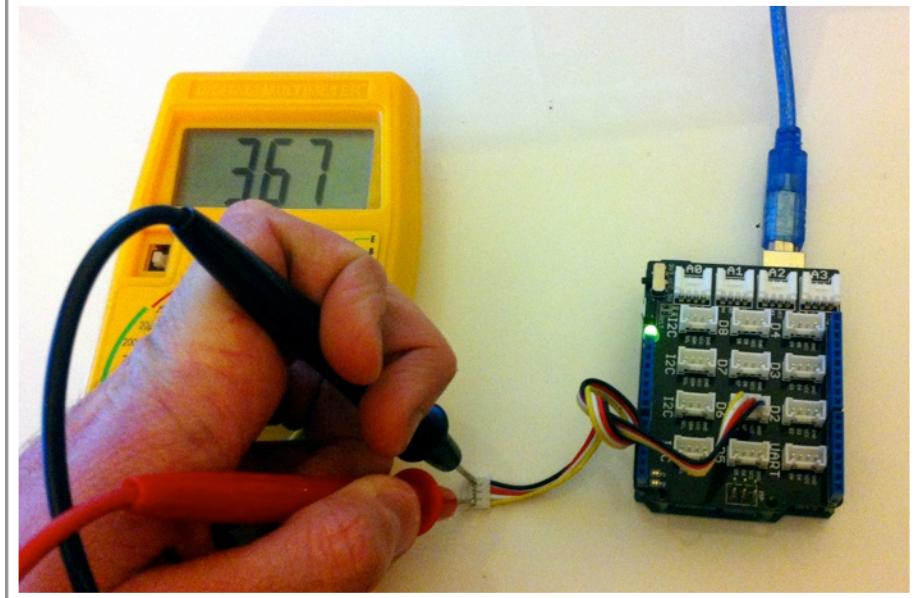


The screenshot shows the ArduBlock library browser sidebar with categories like Broches, Tests, Opérateurs mathématique, Variables/constants, Generic Hardware, Communication, and SCoop (Multitask). The main area displays several blocks related to pins and analog outputs, including 'Valeur de la broche Entrée numérique', 'Valeur de la broche Entrée Analogique', 'Fixe la sortie numérique au niveau', and 'Fixe la sortie analogique à la valeur'. A callout bubble points to the 'Fixe la sortie analogique à la valeur' block.

Dans les blocs de **Broches** :

Glissez / Déposez le bloc
« **Fixe la sortie analogique à la valeur** »

Effectivement la sortie analogique n'est pas exactement à 3,75 mais à 3,67 volts. Ceci est du au fait que l'alimentation n'est pas aussi exactement à 5v mais à 4,88 volts.



Pour les plus électroniciens, il s'agit de ports PWM.

Si vous souhaitez une tension «propre», il faut la filtrer avec un montage R.C de ce type.

Avec $R = 1k$ et $C = 1\mu F$
(temps de réponse rapide et tension correctement filtrée).

