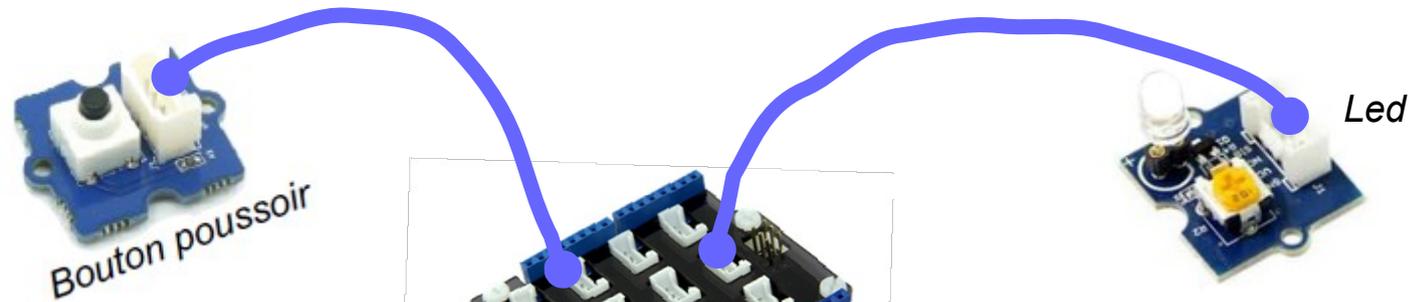


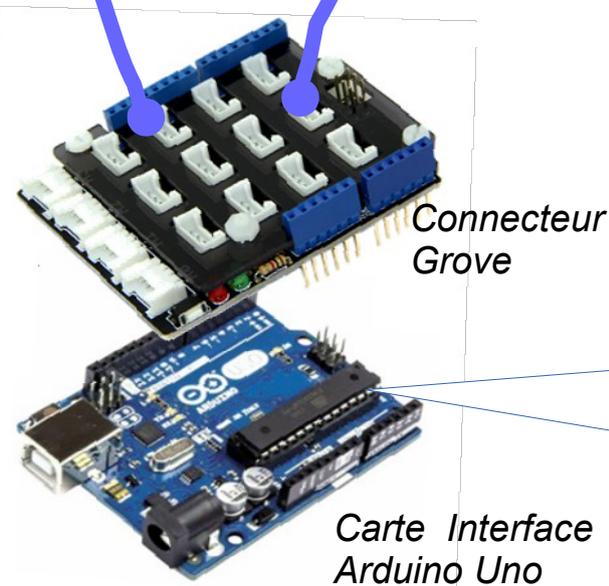
# Découvrir mBlock et Arduino



## Réalisation d'un système d'éclairage autonome :

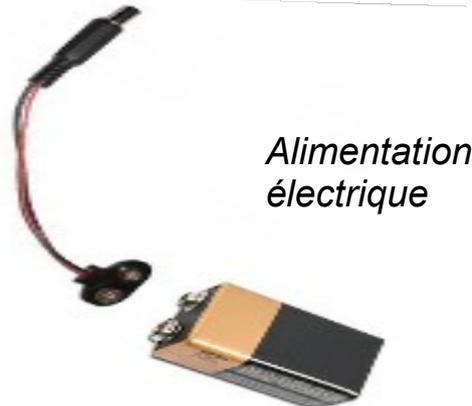


Lorsqu'on appuie sur le bouton, la lampe doit s'allumer pendant 5 secondes, puis s'éteindre.



```
0x00000000 > ff 35 04 09 01 1a 02  
0x00000001 > ff 35 04 02 00 00 00 3f 04  
0x00000002 > ff 35 04 09 01 1a 02  
0x00000003 > ff 35 04 02 00 00 00 3f 04  
0x00000004 > ff 35 04 09 01 1a 02  
0x00000005 > ff 35 04 09 01 1a 02  
0x00000006 > ff 35 04 02 00 00 00 3f 04  
0x00000007 > ff 35 04 09 01 1a 02  
0x00000008 > ff 35 04 09 01 1a 02
```

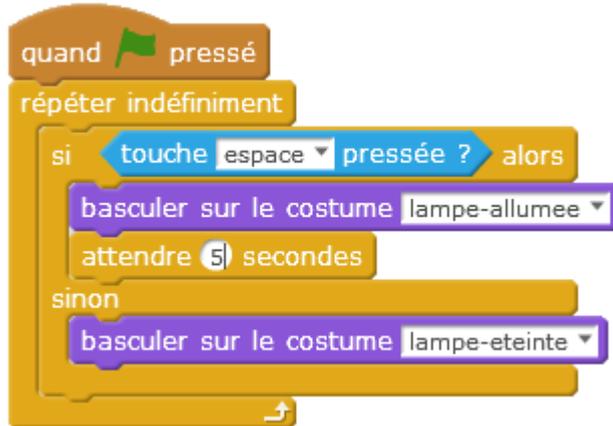
Programme mémorisé dans le microcontrôleur



# Découvrir mBlock et Arduino

## Réalisation du programme et simulation à l'écran :

Le programme est réalisé à l'aide du logiciel mBlock qui permet d'écrire simplement le programme par assemblage de blocs.



La structure du programme peut, dans un premier temps, être testée en simulation sur l'ordinateur :

- une touche du clavier (barre espace) remplace le bouton
- des éléments graphique (costumes) affichés à l'écran, simulent le fonctionnement de la lampe.



**Lors de l'appui sur la touche « espace » du clavier, le lutin Lampe change de costume (du costume 1 : lampe éteinte ; au costume 2 : Lampe allumée).**

# Découvrir mBlock et Arduino

## A. Préparation des éléments graphiques pour la simulation à l'écran :

1

Lancer le logiciel mBlock



3

Créer et nommer les costumes

2

Créer ou ouvrir le lutin « lampe »



x: -93 y: -180

Lutins

Nouveau lutin :



lampe-e...

Scène  
1 arrière-plan

Scripts

Costumes

Sons

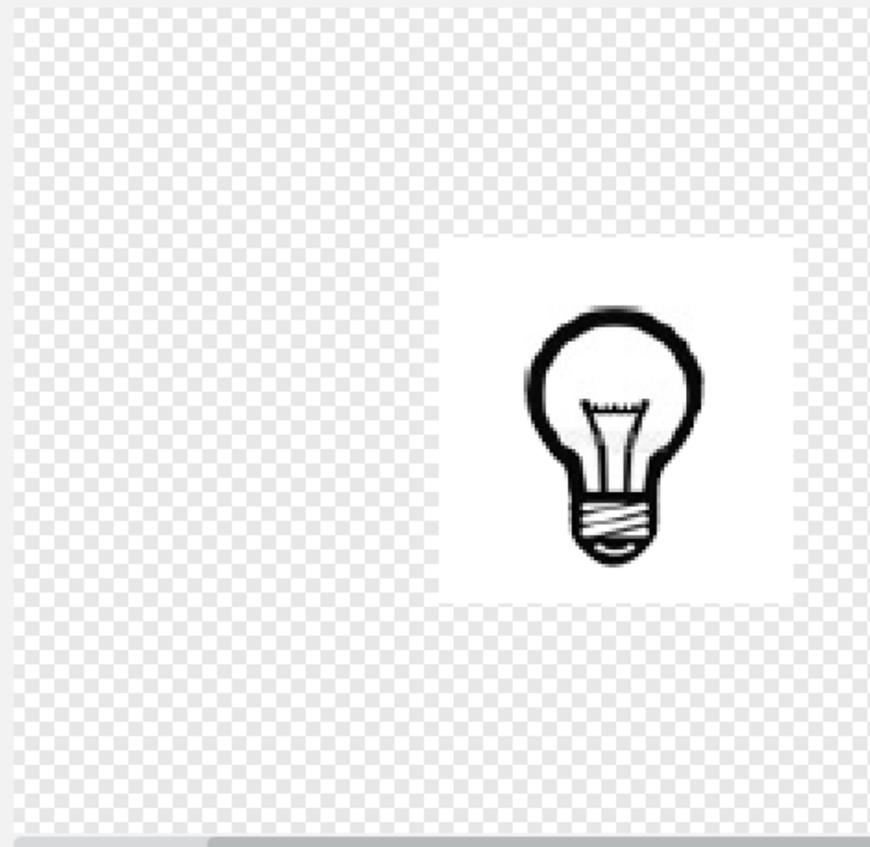


Nouveau costume :

lampe-eteinte



Effacer Importer



# Découvrir mBlock et Arduino

## B. Écriture du script :

5

Lancer le programme

4

Réaliser le script en utilisant les blocs d'instructions.

The screenshot displays the mBlock software interface. On the left, a stage window titled 'lampe' shows a lightbulb icon. Below it, a 'Lutins' (sprites) window shows a lightbulb sprite named 'lampe-e...'. The main workspace is divided into three panels: 'Scripts', 'Costumes', and 'Sons'. The 'Scripts' panel contains a list of categories: Mouvement, Apparence, Son, Stylo, Blocs & variables, Evénements, Contrôle, Capteurs, Opérateurs, and Pilotage. The script area shows a sequence of blocks: 'montrer', 'cacher', 'basculer sur le costume lampe-allumee', 'costume suivant', 'basculer sur l'arrière-plan arrière-plan1', 'ajouter à l'effet couleur 25', 'mettre l'effet couleur à 0', 'annuler les effets graphiques', 'ajouter 10 à la taille', and 'mettre à 100 % de la taille initiale'. A script is also visible on the right side of the workspace, starting with 'quand flag pressé', followed by a 'répéter indéfiniment' loop containing a 'si touche espace pressée ? alors' block with 'basculer sur le costume lampe-allumee' and 'attendre 5 secondes' blocks, and a 'sinon' block with 'basculer sur le costume lampe-eteinte'. Blue arrows point from the numbered text boxes to the 'Lancer le programme' button and the script blocks.

# Découvrir mBlock et Arduino

## C. Test du programme en simulation sur l'ordinateur :

6

Utiliser la barre d'espace du clavier



7

Observez le résultat à l'écran



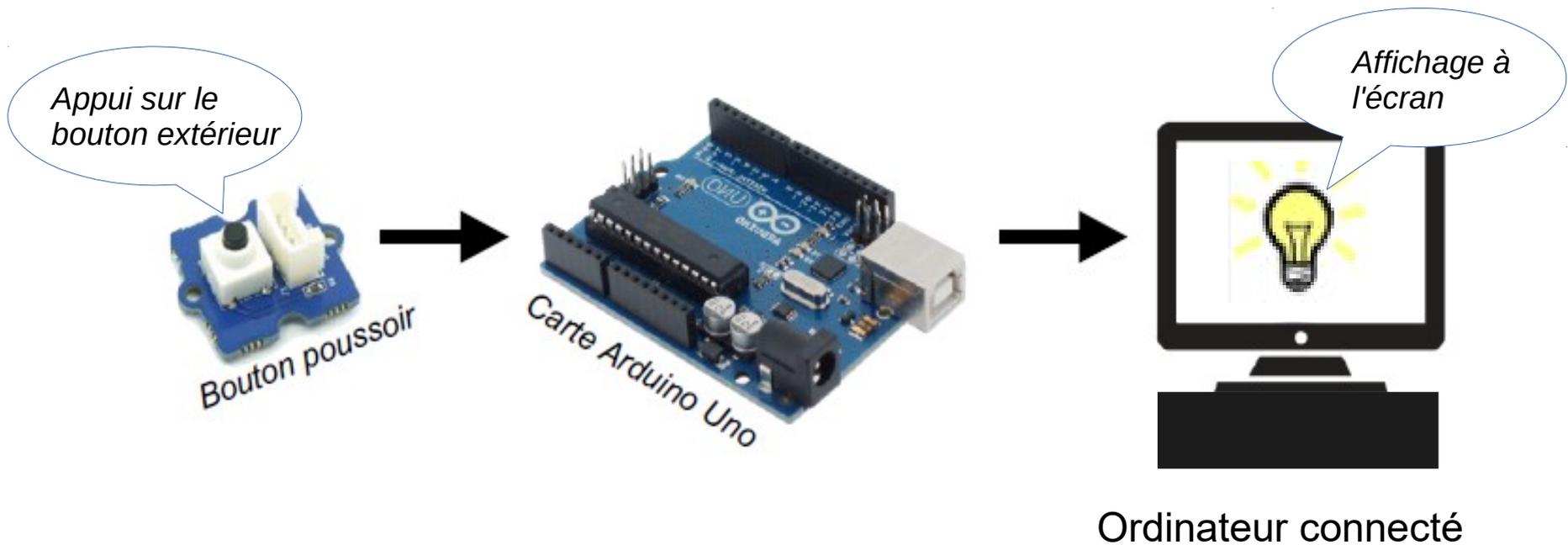
Affichage du costume du lutin

The screenshot shows the mBlock software interface. On the left, a window titled 'lampe' displays a glowing lightbulb icon. A speech bubble above it says 'Affichage du costume du lutin'. Below the window, a blue box contains the text '7 Observez le résultat à l'écran'. At the bottom left, a 'Lutins' (Sprites) panel shows a lightbulb icon labeled 'lampe-e...'. The main area is the script editor, which is divided into 'Scripts', 'Costumes', and 'Sons' tabs. The 'Scripts' tab is active, showing a script for a 'quand flag pressé' event. The script includes a 'répéter indéfiniment' loop with a 'si' (if) condition: 'si touche espace pressée ? alors' followed by 'basculer sur le costume lampe-allumee', 'attendre 5 secondes', and 'sinon' followed by 'basculer sur le costume lampe-eteinte'. Other visible script blocks include 'montrer', 'cacher', 'basculer sur l'arrière-plan', 'ajouter à l'effet couleur 25', 'mettre l'effet couleur à 0', 'annuler les effets graphiques', 'ajouter 10 à la taille', and 'mettre à 100 % de la taille initiale'.

# Découvrir mBlock et Arduino

## Usage d'un élément extérieur :

Le but est d'obtenir le même fonctionnement mais depuis un bouton extérieur.  
Pour cela on utilise une carte interface entre le bouton et l'ordinateur.



# Découvrir mBlock et Arduino

## A. Configuration de mBlock :

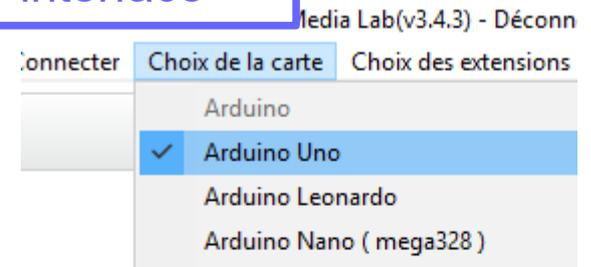
1

Connecter la carte au poste informatique avec le câble USB



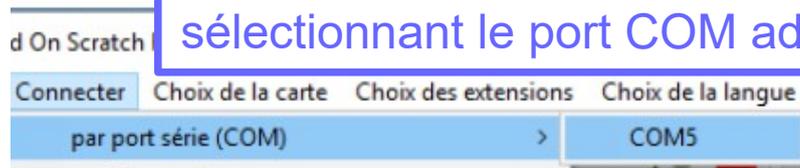
2

Sélectionner le type de carte interface

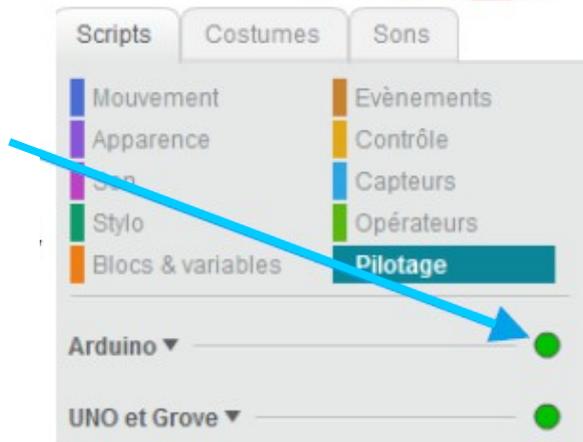


3

Connecter la carte Arduino en sélectionnant le port COM adapté

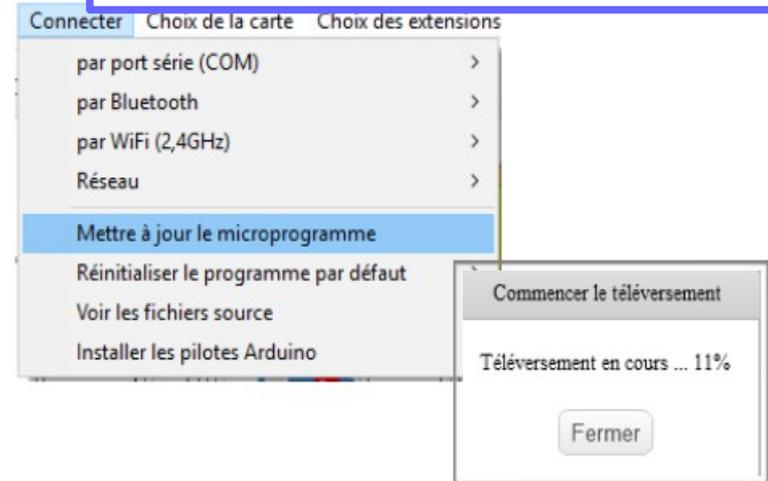


Si l'interface Arduino est bien connectée et reconnue par le poste informatique, le voyant rouge passe au vert.



4

Mettre à jour le microprogramme



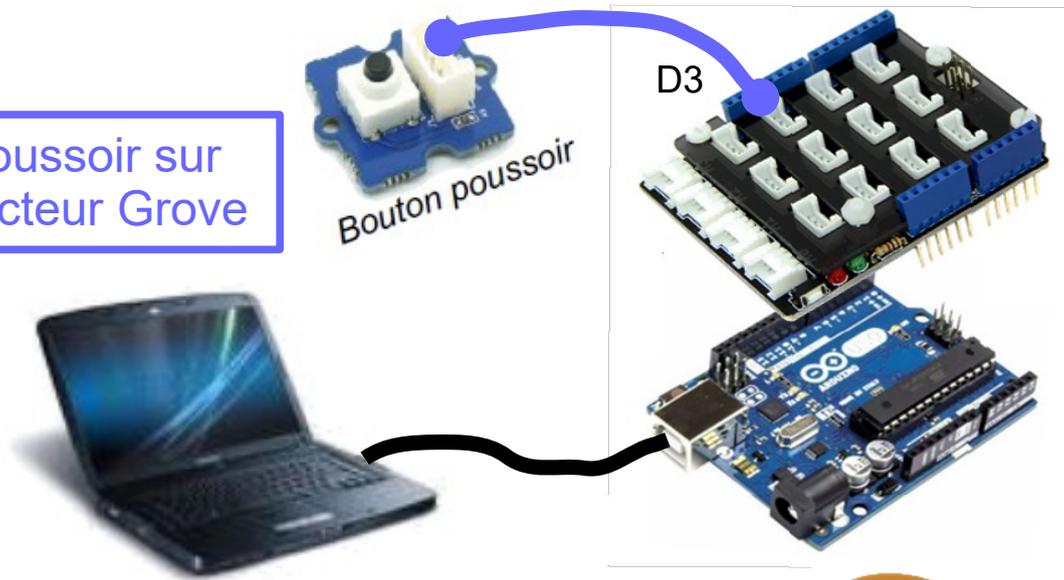
Cela permet d'enregistrer dans l'interface Arduino un programme pour communiquer avec le poste informatique

# Découvrir mBlock et Arduino

## B. Adaptation du script :

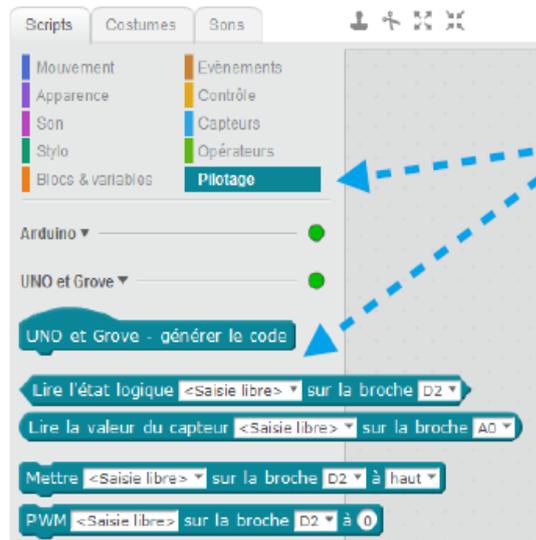
5

Connecter le bouton poussoir sur la broche D3 du connecteur Grove

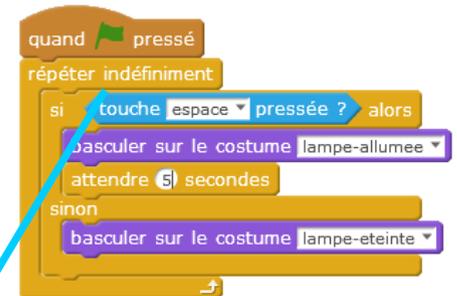


6

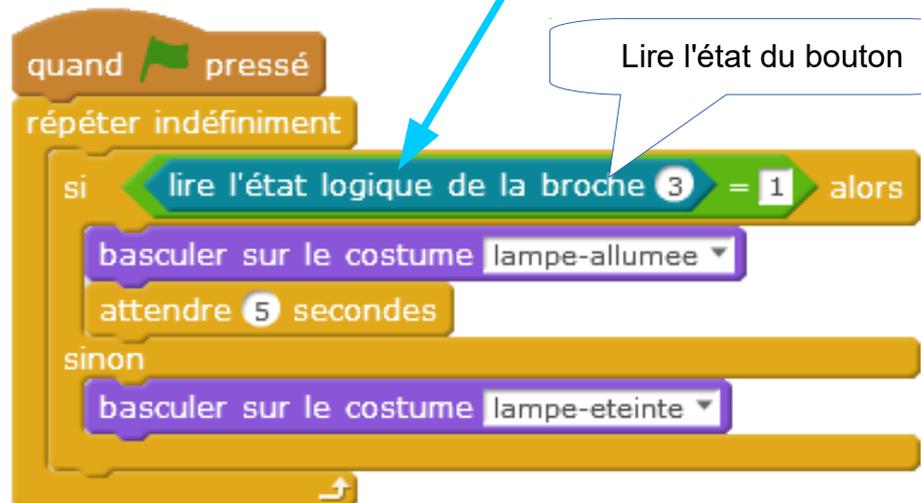
Adapter le programme au câblage en utilisant une instruction de pilotage



Librairies  
spécifiques  
Arduino



Lire l'état du bouton



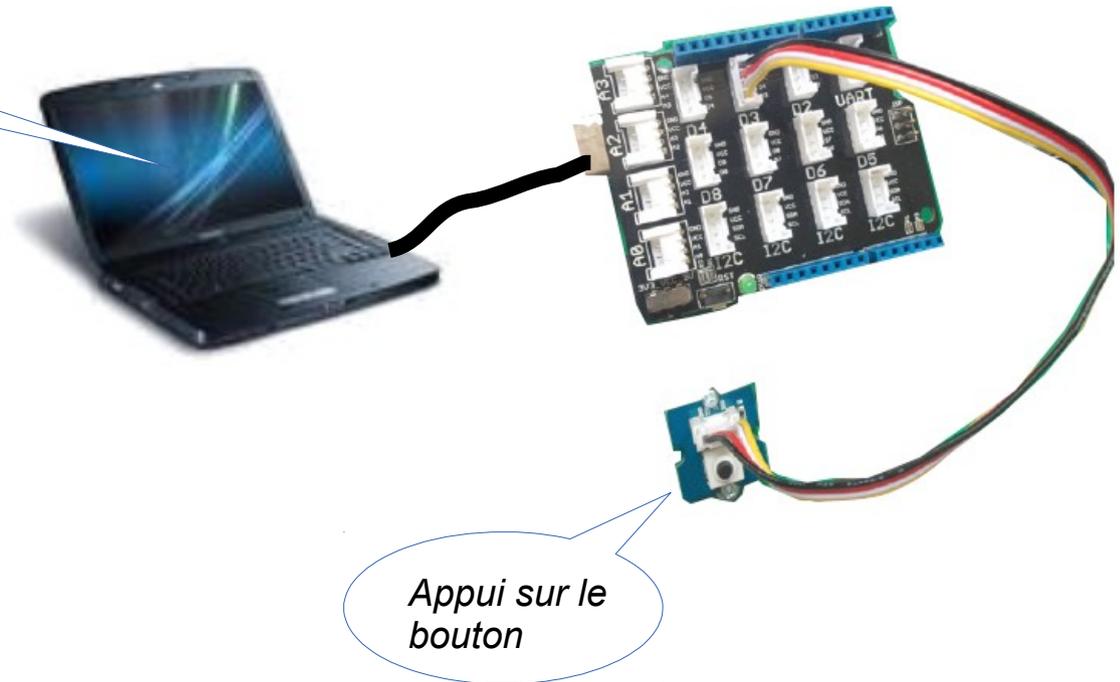
# Découvrir mBlock et Arduino

## C. Test du fonctionnement :



7

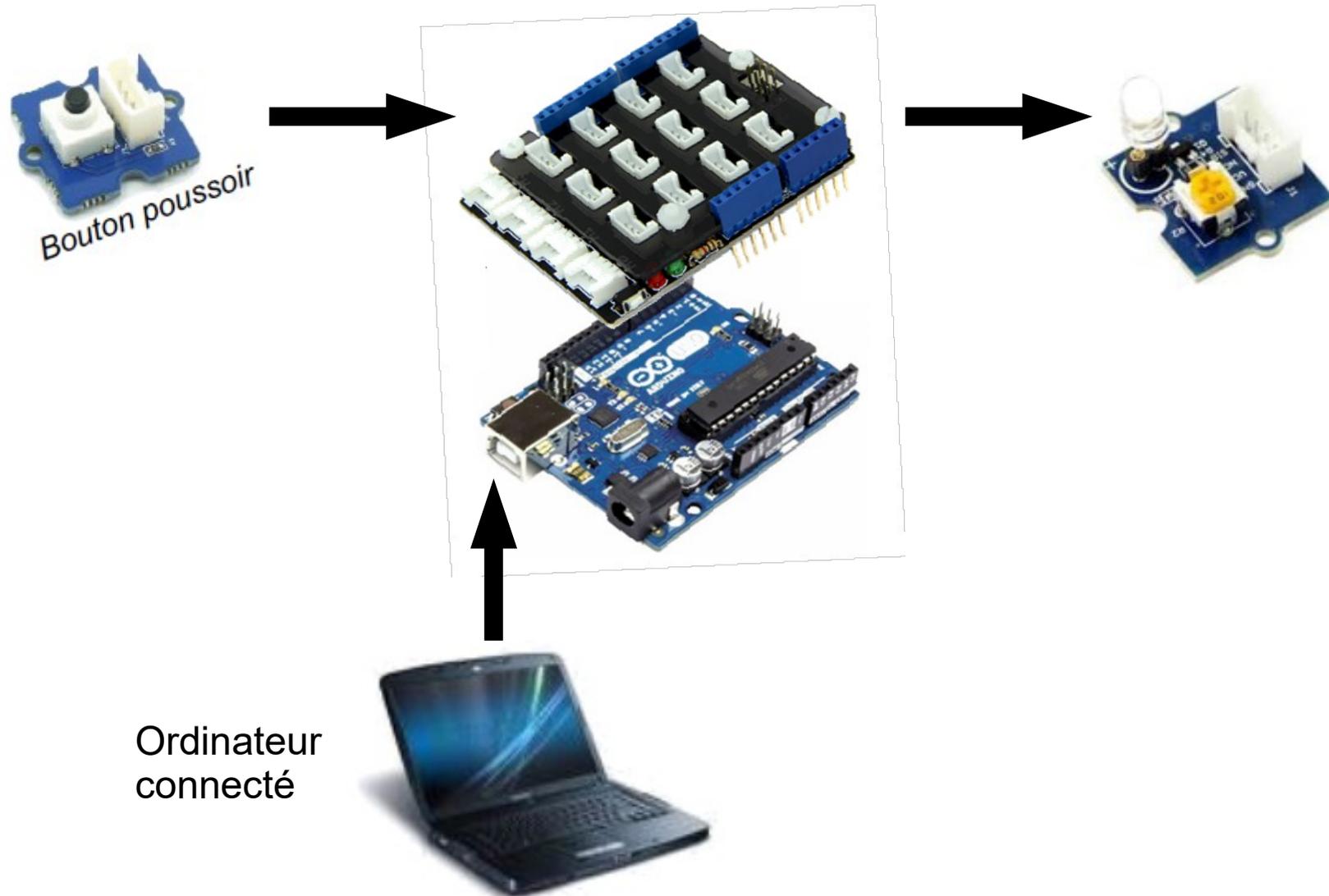
Lancer le programme et vérifier le bon fonctionnement.



# Découvrir mBlock et Arduino

## Pilotage des éléments réels en mode connecté:

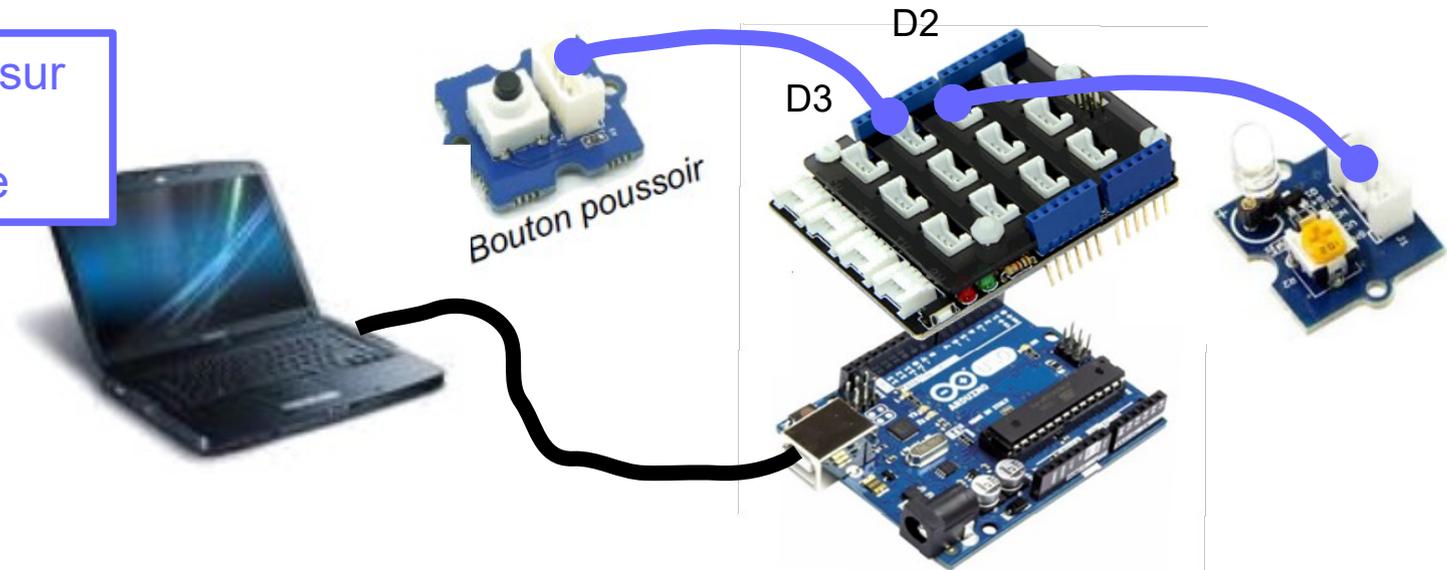
Le système réel est connecté à l'interface et va être testé.



# Découvrir mBlock et Arduino

1

Connecter la Del sur la broche D2 du connecteur Grove



2

Adapter le programme en utilisant les instructions de pilotage

```
quand flag pressé
répéter indéfiniment
  si lire l'état logique de la broche 3 = 1 alors
    mettre l'état logique de la broche 2 à haut
    attendre 5 secondes
  sinon
    mettre l'état logique de la broche 2 à bas
```

Mettre D2 à l'état 1 logique

Mettre D2 à l'état 0 logique

3

Tester le fonctionnement sur le matériel en mode connecté



Bouton poussoir  
Relâché

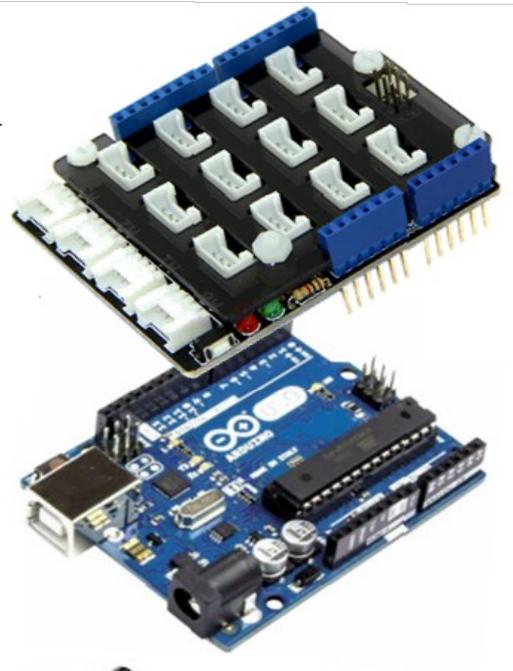
Led éteinte

Bouton poussoir  
Appuyé

Led allumée

# Découvrir mBlock et Arduino

## Passage en mode déconnecté (système embarqué) :



Le programme sera téléversé dans la mémoire du microcontrôleur de la carte arduino.

L'ordinateur sera déconnecté

Le système fonctionnera en autonomie grâce à l'énergie de la pile

# Découvrir mBlock et Arduino

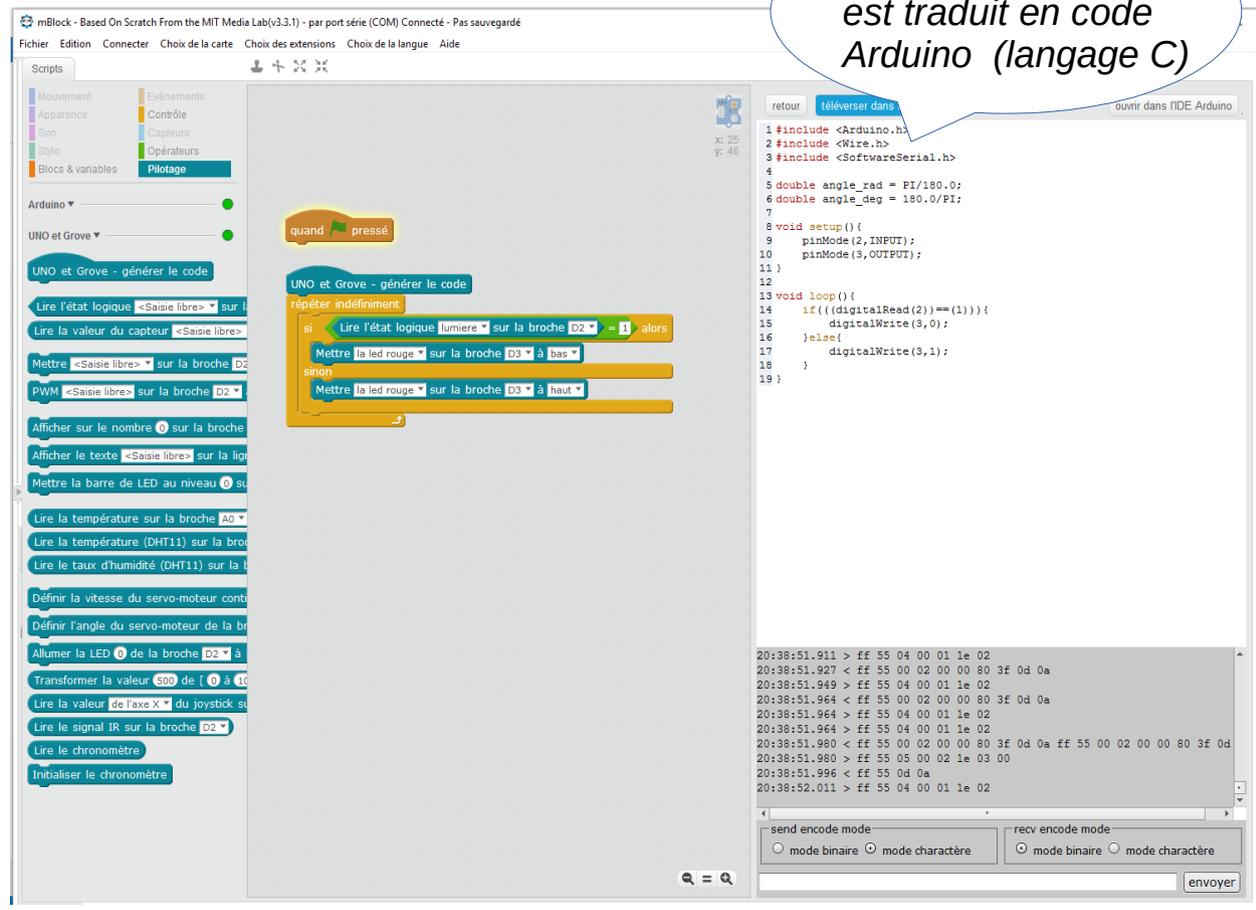
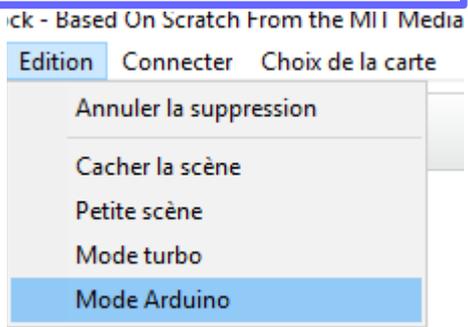
1 Modifiez la tête du programme

quand  pressé



Le programme block est traduit en code Arduino (langage C)

2 Passage au mode Arduino



```
1 #include <Arduino.h>
2 #include <Wire.h>
3 #include <SoftwareSerial.h>
4
5 double angle_rad = PI/180.0;
6 double angle_deg = 180.0/PI;
7
8 void setup() {
9   pinMode(2, INPUT);
10  pinMode(3, OUTPUT);
11 }
12
13 void loop() {
14   if ((digitalRead(2)) == (1)) {
15     digitalWrite(3, 0);
16   } else {
17     digitalWrite(3, 1);
18   }
19 }
```

Serial output:

```
20:38:51.911 > ff 55 04 00 01 1e 02
20:38:51.927 < ff 55 00 02 00 00 80 3f 0d 0a
20:38:51.949 > ff 55 04 00 01 1e 02
20:38:51.964 < ff 55 00 02 00 00 80 3f 0d 0a
20:38:51.964 > ff 55 04 00 01 1e 02
20:38:51.964 > ff 55 04 00 01 1e 02
20:38:51.980 < ff 55 00 02 00 00 80 3f 0d 0a ff 55 00 02 00 00 80 3f 0d
20:38:51.980 > ff 55 00 02 1e 03 00
20:38:51.996 < ff 55 0d 0a
20:38:52.011 > ff 55 04 00 01 1e 02
```

# Découvrir mBlock et Arduino

```
1 #include <Arduino.h>
2 #include <Wire.h>
3 #include <SoftwareSerial.h>
4
5 double angle_rad = PI
6 double angle_deg = 180
7
8 void setup() {
9   pinMode(2, INPUT);
10  pinMode(3, OUTPUT)
11 }
12
13 void loop() {
14   if ((digitalRead(2) == 1)) {
15     digitalWrite(3, 0);
16   } else {
17     digitalWrite(3, 1);
18   }
19 }
```

20:38:51.911 > ff 55 04 00 01 1e 02  
20:38:51.927 < ff 55 00 02 00 00 80 3f 0d 0a  
20:38:51.949 > ff 55 04 00 01 1e 02  
20:38:51.964 < ff 55 00 02 00 00 80 3f 0d 0a  
20:38:51.964 > ff 55 04 00 01 1e 02  
20:38:51.964 < ff 55 04 00 01 1e 02  
20:38:51.980 < ff 55 00 02 00 00 80 3f 0d 0a ff 55 00 02 00 00  
20:38:51.980 > ff 55 05 00 02 1e 03 00  
20:38:51.996 < ff 55 0d 0a  
20:38:52.011 > ff 55 04 00 01 1e 02

retour téléverser dans l'Arduino

3

Transférer le programme dans l'interface arduino

Commencer le téléversement

Téléversement fini

Fermer

4

Attendre la fin du téléversement

Le programme est compilé en codes binaires

5

Tester le programme sur le matériel (Alimenté par une pile, le système est autonome et le câble USB peut être retiré)