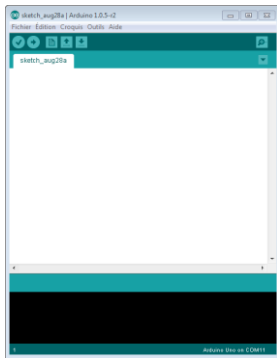




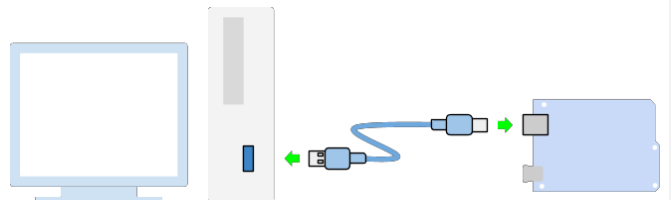
## A- Démarrage

### VOTRE LOGICIEL ARDUINO OUVERT



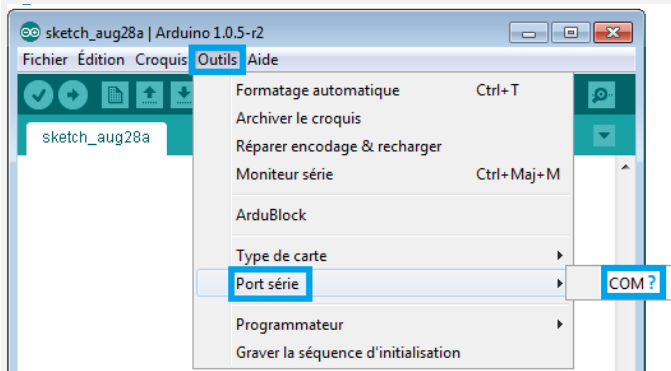
**01 - Lancez votre logiciel Arduino sur votre ordinateur**

### CABLAGE



**02 - Branchez votre carte Arduino sur votre ordinateur**

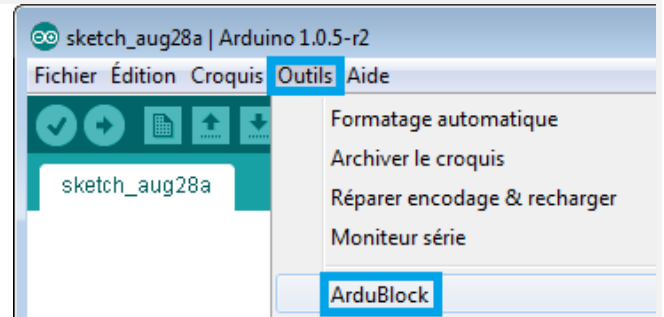
### CONNEXION DE LA CARTE A L'ORDINATEUR



**03 – Configurez le port de connexion de votre carte Arduino**

-> Outils puis Port série, vérifiez la présence de COMxx

### LANCEMENT D'ARDUBLOCK



**04- Cliquez sur Outils puis Ardublock**

**Cette partie est consacrée aux rudiments de la programmation graphique avec le plug'in libre Ardublock.**

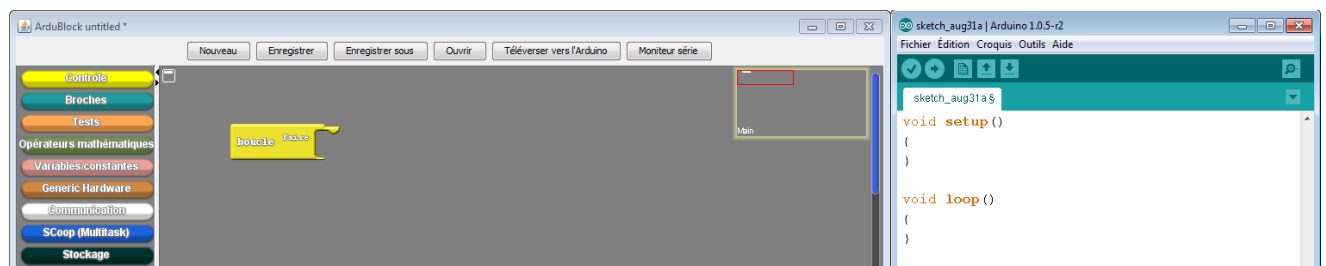
Arduino doit être programmé avec un langage appelé langage Arduino. C'est un dérivé des langages C/C++ un peu simplifié. Même si ce langage n'est pas très compliqué, il est possible de faire beaucoup plus simple ! C'est là qu'intervient le Plug'in Ardublock. C'est un petit programme s'intégrant directement dans Arduino. Rassurez-vous, nous avons intégré Ardublock pour vous et l'avons même augmenté pour faciliter son utilisation pour certains modules.

Retenez que le rôle d'Ardublock est de générer le code Arduino à votre place.

Principe :

1. faites glisser les blocs des bibliothèques dans la zone travail.
2. le programme doit apparaître automatiquement dans le logiciel Arduino
2. il doit compiler (un message vous le confirme)
3. le programme compilé doit ensuite se téléverser dans la carte Arduino
4. ensuite, le message " Téléversement terminé" doit apparaître
5. le programme s'exécute dans la carte Arduino aussi longtemps que la carte Arduino est alimentée.

ARDUBLOCK a généré automatiquement le programme dans ARDUINO



Le logiciel Arduino vous dit ce qui se passe.

1. **Téléversement terminé** = votre programme est opérationnel !
2. **Compilation terminée** = votre programme est correct mais n'est pas dans la carte
3. **Vous avez un message orange** = il y a 1 erreur de syntaxe dans Arduino

## B-Mon Premier programme : Faire clignoter une led

Les outils maintenant installés, nous pouvons enfin programmer. La démarche classique est la suivante.



Démarche classique pour programmer avec Ardublock

1. Réfléchir au résultat attendu
2. **Brancher les modules** sur le shield de connexion par des câbles
3. **Placez vos blocs** (image de votre programme)

Retenez ceci :



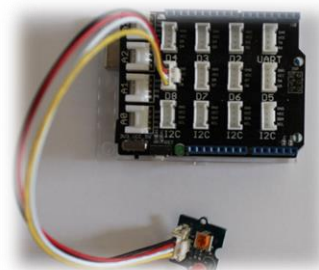
Pour supprimer un bloc :

- faites-le glisser vers le menu de gauche (il retournera d'où il vient)
- pour placer un bloc "Boucle" allez dans Contrôles"

### B.1 - Nous allons **ajouter un module LED**

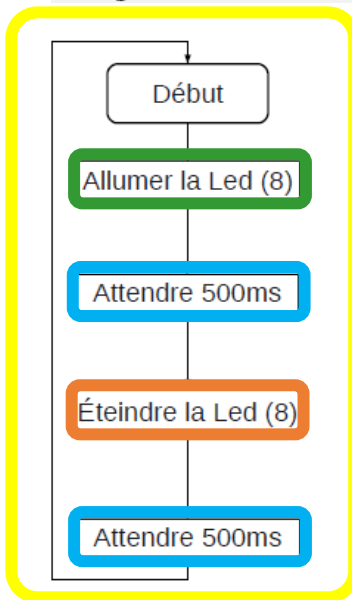
Le branchement est très simple : prenez votre module LED, prenez un câble Grove et réalisez le branchement suivant.

**Reliez** votre **module LED** au shield de connexion **par un câble Grove** sur une des bornes **D2 à D8**, sur la photo la borne D8 a été choisie.

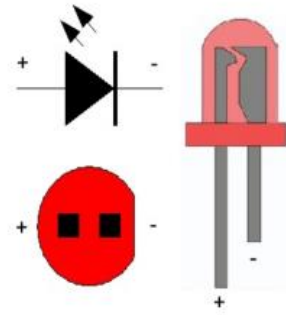
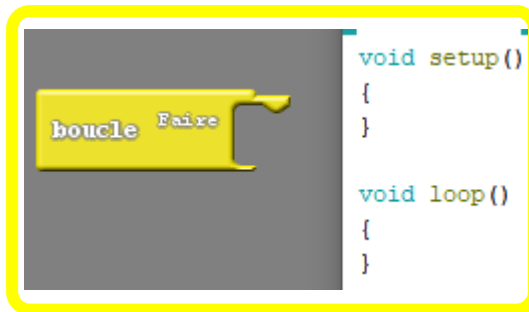
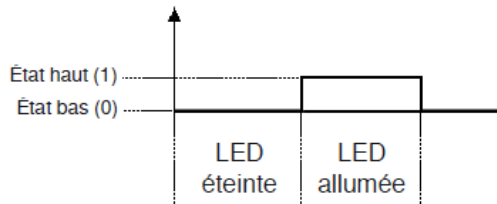


## Objectif du programme

1. La led s'allume et s'éteint toutes les 500ms à l'infini ...



Algorithme



**Attention :** Une DEL est un composant *polarisé*. Il faut respecter le sens de fonctionnement !

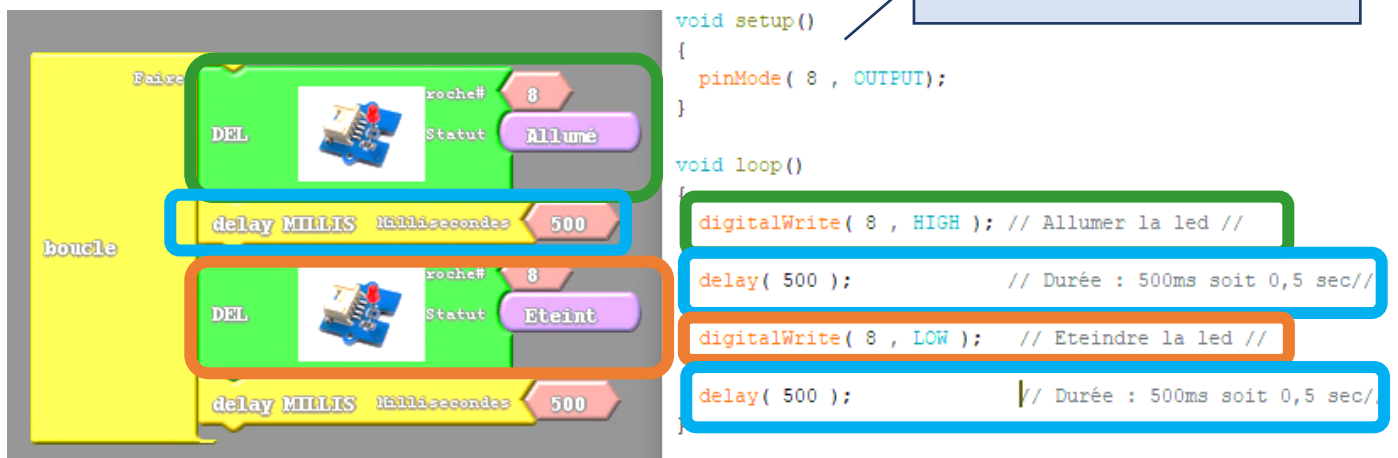
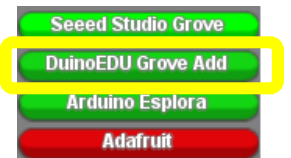
## Élément de base pour débiter tout programme

Void setup () : déclaration de tous les composants participant au programme.  
Void loop () : programme décomposé en différentes étapes

## B2 - Nous allons ajouter un module LED dans Ardublock

Pour cela, c'est assez simple :

1. cliquez sur le bouton DuinoEDU (menu de gauche) => un menu apparaît
2. faites glisser l'objet LED du menu vers votre espace de travail
3. emboîtez ensuite votre bloc LED dans la mâchoire boucle (ou loop)
4. écrivez ensuite au clavier "8" ou "D8" selon la version pour l'option "Broche" qui par défaut est à 1 ou allumé ou high.



Déclaration du composant

Led sur l'emplacement D8 soit 8 en Sortie (OUTPUT)

```
void setup()
{
  pinMode( 8 , OUTPUT);
}

void loop()
{
  digitalWrite( 8 , HIGH ); // Allumer la led //
  delay( 500 ); // Durée : 500ms soit 0,5 sec//
  digitalWrite( 8 , LOW ); // Eteindre la led //
  delay( 500 ); // Durée : 500ms soit 0,5 sec//
}
```

ArduBlock

Code

- Changer la durée d'éclairement de la led à 1 seconde et sa connexion sur D2 sur l'image ci-dessous.

```

void setup()
{
  pinMode( 8 , OUTPUT);
}

void loop()
{
  digitalWrite( 8 , HIGH ); // Allumer la led //
  delay( 1000 );           // Durée : 1000ms soit 1 sec//
  digitalWrite( 8 , LOW ); // Eteindre la led //
  delay( 1000 );           // Durée : 1000ms soit 1 sec//
}

```

- Testez le résultat.

Pour éviter de modifier plusieurs paramètres à chaque fois, on peut utiliser une variable pour fixer le délai de clignotement allumé Ta et éteint Te.

**led-rouge-variable**  

```

int _ABVAR_1_Ta = 0 ;
int _ABVAR_2_Te = 0 ;

```

```

void setup()
{
  pinMode( 8 , OUTPUT);
}

void loop()
{
  _ABVAR_1_Ta = 500 ;
  _ABVAR_2_Te = 1000 ;
  digitalWrite( 8 , HIGH );
  delay( _ABVAR_1_Ta );
  digitalWrite( 8 , LOW );
  delay( _ABVAR_2_Te );
}

```

Pour éviter d'avoir à répéter plusieurs fois la même séquence, il est possible d'utiliser le bloc « **Répète** »

Ici le programme va répéter 5 fois la commande qui sera insérée ci-contre

Objectif du programme

2. Réaliser le signal en morse du sigle SOS.



**B.2 - Vous allez maintenant réaliser le programme** simulant en morse le signal lumineux **SOS**

- Sur feuille réaliser l'/les algorithme(s).
- Appeler votre professeur pour la validation.
- Utiliser Ardublock pour réaliser le programme.



## C-Mon Deuxième programme : Éclairage des parties communes d'un immeuble

**C.1 - Le résultat attendu est le suivant :**



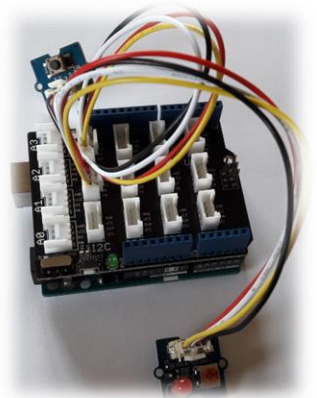
### Objectif du programme

3. Si l'interrupteur est actionné, la LED doit s'allumer
4. Sinon, la LED doit rester éteinte
5. L'éclairage s'éteint après 5 sec.

Nous arrivons à un point essentiel en programmation : les tests. Ici, notre test est très simple mais permet déjà de faire nombre de programmes.

**C.2 - Nous allons donc ajouter un module Bouton**

Le branchement est aussi simple que pour la LED. Mais comme vous débutez, nous allons prendre une habitude. Nous n'allons brancher nos modules que sur les prises paires pour l'instant. Il est tout à fait possible d'utiliser aussi les prises impaires, ici au hasard D4.



Reliez votre **module Bouton** au shield de connexion **par un câble Grove**

**C.3 - Nous allons maintenant ajouter un bloc "Si - Sinon"**

Le bloc "Si - Sinon" est justement fait pour introduire une condition du type : s'il pleut, ouvre ton parapluie, Sinon, ferme ton parapluie.



Procédez comme ceci :

1. **insérez** le bloc "Boucle"
2. cliquez sur le bouton "**Contrôles**" (menu de gauche)
3. faites glisser le bloc "**Si - Sinon**" du menu vers votre espace de travail.
4. faites glisser le bloc "**Bouton**" du menu vers votre espace de travail
5. réaliser le programme
6. appeler le professeur

