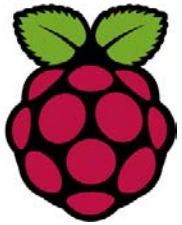
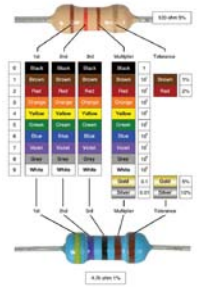
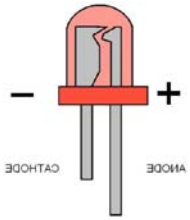
PETIT MAIS COSTAUD - SOS ARDUINO

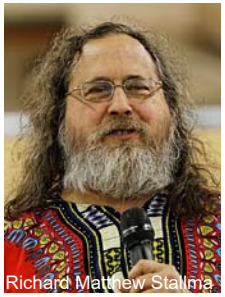
SOS Arduino Uno ... --- ...

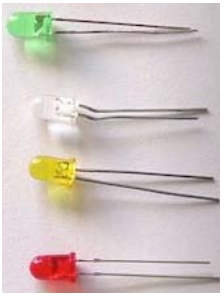
Petit mais costaud l'Arduino Uno !

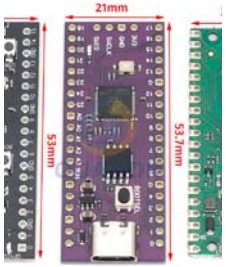


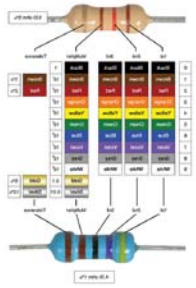


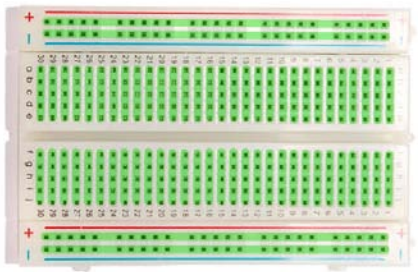












Création maquette et contenu par Laurent Gatto
laurent.gatto@free.fr - http://laurent.gatto.free.fr

SOS Arduino Uno ... --- ...

Objectifs de l'atelier

- Identifier les fonctionnalités d'une carte Arduino Uno.
- Réaliser des montages simples avec des composants électroniques.
- Écrire un programme basique pour contrôler ces composants.

Au cours de cet atelier nous allons donc :

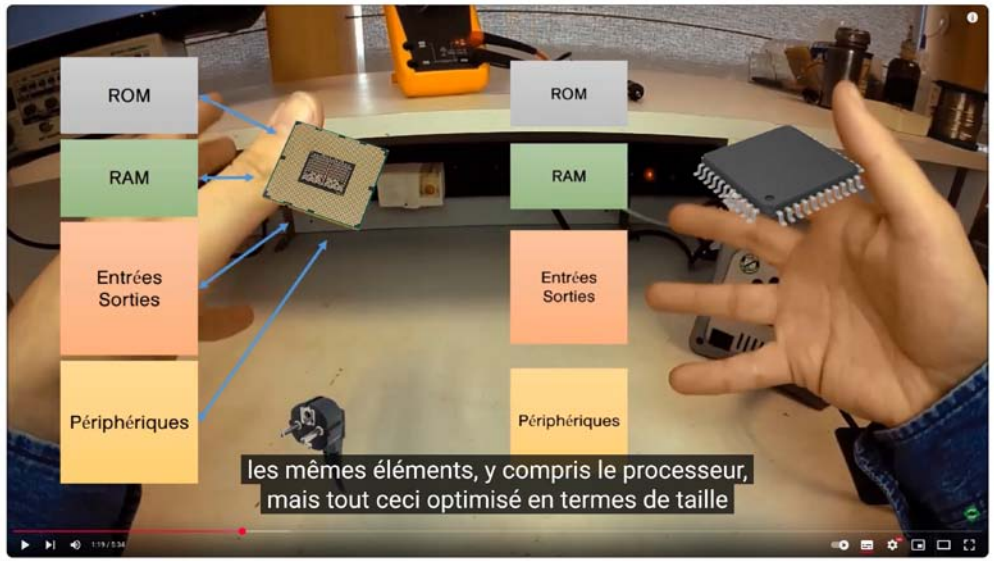
- Découvrir l'Arduino Uno.
- Identifier ses composants et sa connectique.
- Définir ses usages possibles et s'initier à la programmation par blocs.
- Faire nos premiers montages.

SOS Arduino Uno ... --- ...

Re-contextualisation

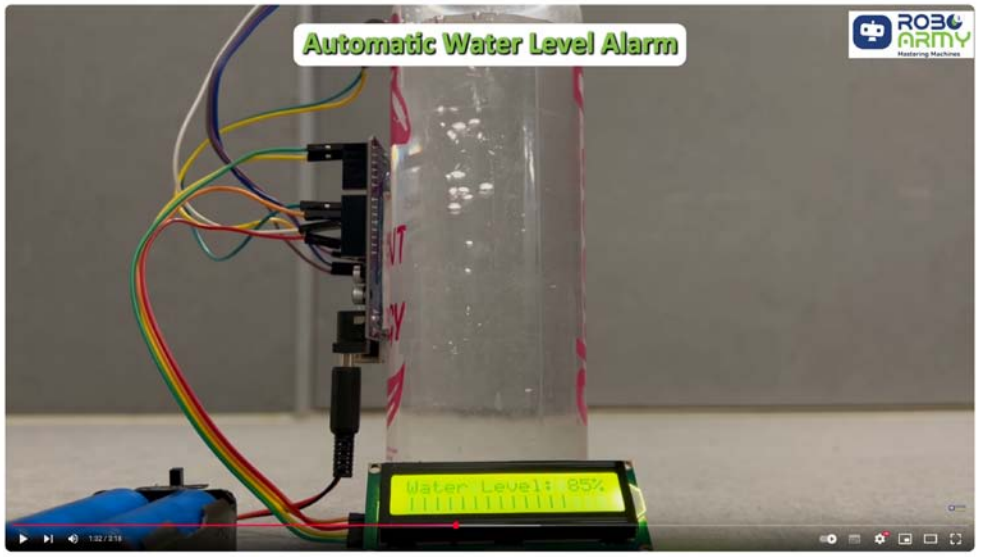
Dans la séance précédente nous avons

- Identifié les principaux micro contrôleurs du marché.
- Identifié des usages possibles d'un micro contrôleur...
- Avec des exemples de projets et d'applications au quotidien.



Microcontrôleur : Comment ça marche ? - SILIS Electronique -

https://www.youtube.com/watch?v=U9T0fjFyVow



Top 20 Arduino Projects for 2025: Science Exhibition & DIY Ideas!

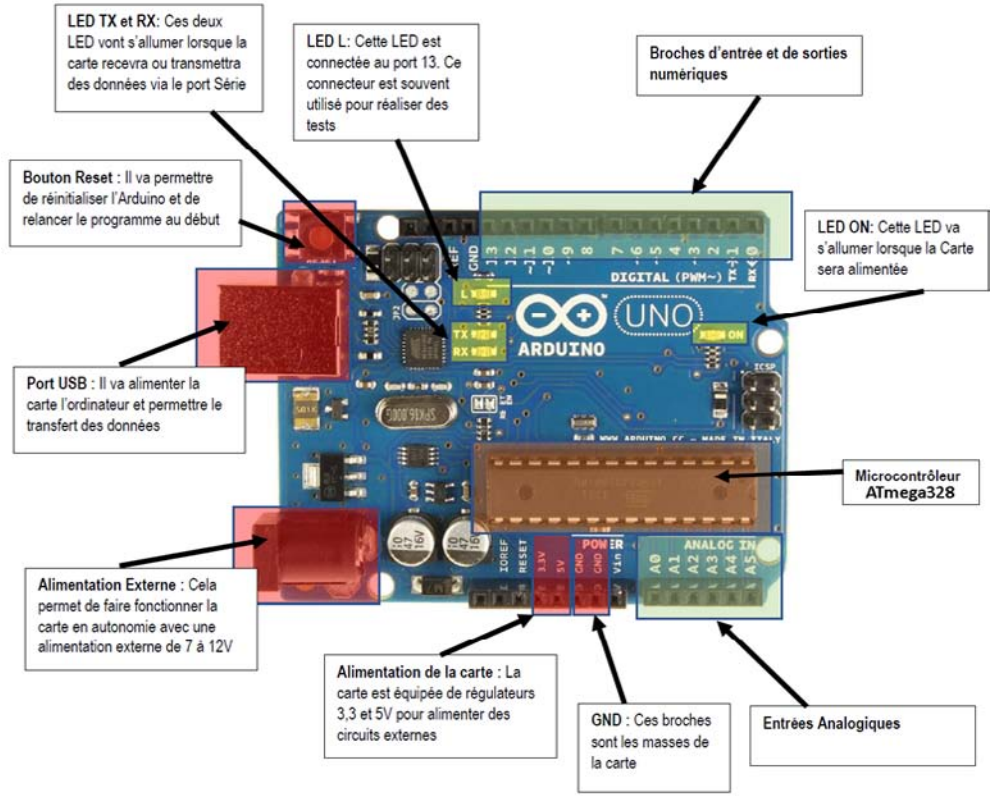
https://www.youtube.com/watch?v=5hYFX2mDNAY

SOS Arduino Uno ... --- ...

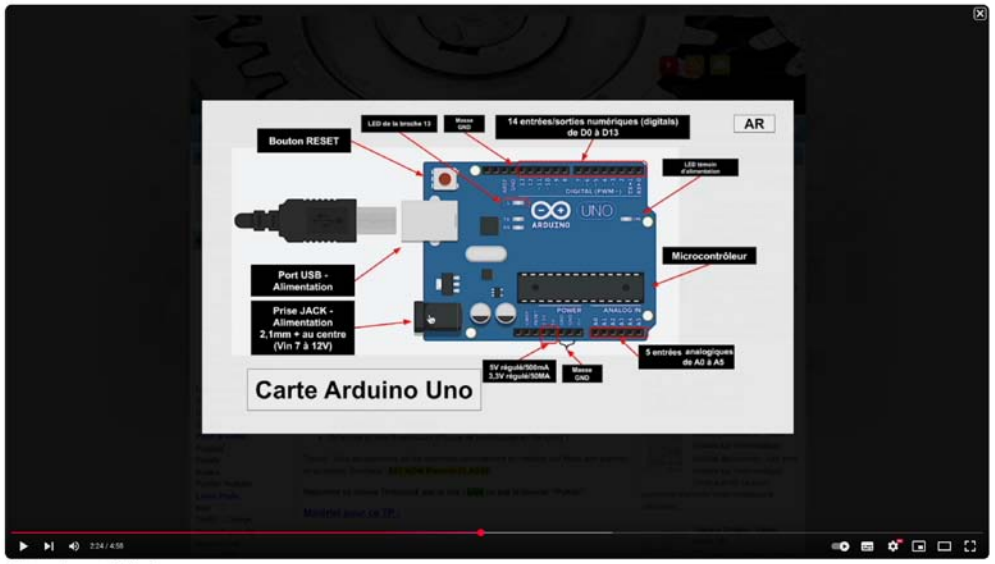
Re-contextualisation

Dans la séance précédente nous avons

- Manipulé la carte Arduino Uno et identifier ses composants : micro contrôleur, broches, alimentation, port USB...
- Vu les langages utilisés (bloc de type « Scratch » et C++).
- Abordé les principes d'un algorithme : nous avons compris qu'à la façon d'une recette de cuisine les ingrédients sont aussi importants que l'ordre d'utilisation.



https://www.ardpylab.fr/?page_id=94



Qu'est ce qu'une carte Arduino ?

https://www.youtube.com/watch?v=hAkWL495-qk

SOS Arduino Uno

Petit mais costaud, l'Arduino !!!

Il nous reste donc à découvrir l'interface de programmation sous Tinkercad :
pour y simuler un montage Arduino.

à commencer à faire ses premiers programmes et montages :
en utilisant des composants comme les Leds, résistances....

On y va ! ;)

Les participants peuvent se mettre en binômes.

Bon atelier !

Le support de cours sera remis et/ou téléchargeable à la fin de la séance.

SOS Arduino Uno ... --- ...

Présentation de l'interface de Tinkercad

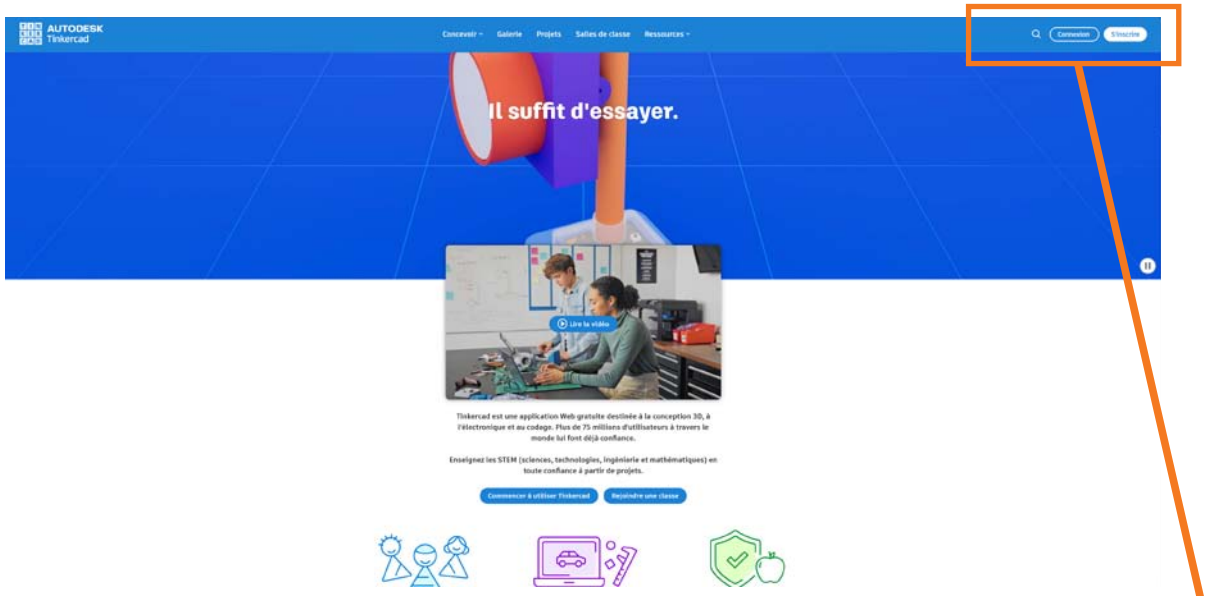
Avant de créer physiquement nos montages nous allons les créer et les tester virtuellement.

Nous avons 2 solutions pour accéder à l'interface de Tinkercad :

1 - Avec le programme : nous Lançons simplement Tinkercad (par le menu Windows ou en double cliquant sur l'icône sur le bureau).

2 - Nous pouvons utiliser le site tinkercad.com qui permet de nombreuses choses dont la création de circuits.

Pour commencer nous allons donc ouvrir ce site dans un navigateur en tapant : tinkercad.com dans la barre de navigation. **Nous allons arriver sur cette page :**



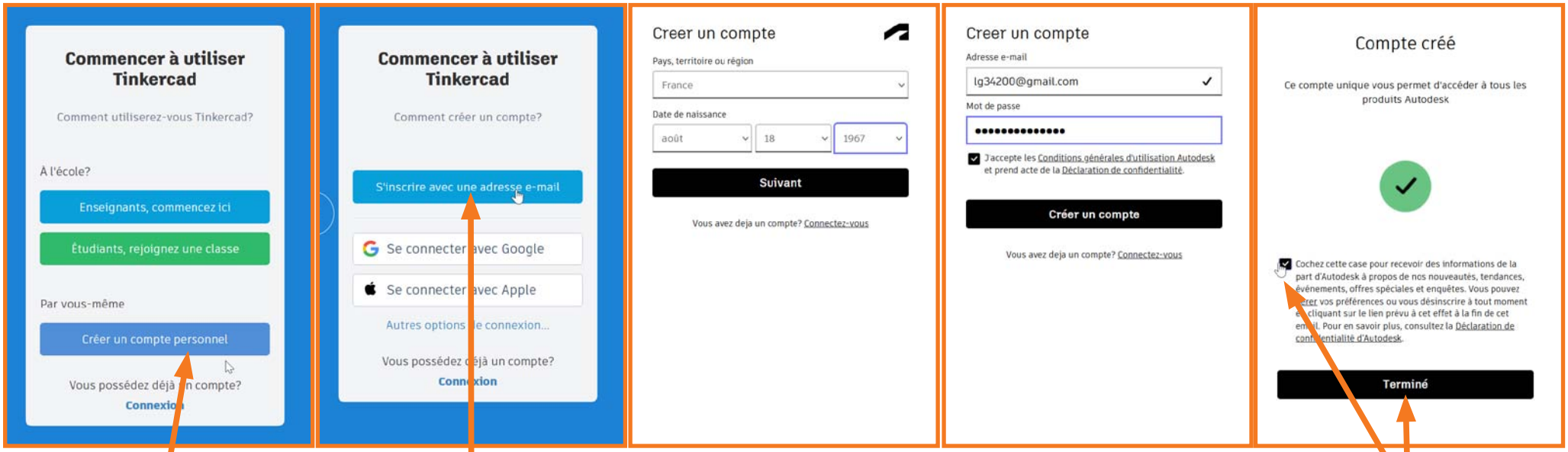
Avant de commencer il faut se connecter ou ouvrir un compte pour pouvoir utiliser le site et stocker en ligne ses projets. **En haut à droite de la page :**

SOS Arduino Uno ... --- ...

Présentation de l'interface de Tinkercad

Nous allons utiliser un compte personnel et nous inscrire avec notre adresse mail :

Puis remplir simplement les informations demandées jusqu'à finaliser l'inscription (en oubliant pas d'utiliser un mot de passe «fort» et de le noter).



1. **Créer un Compte Personnel**
2. **S'inscrire avec une adresse e-mail**
3. **Remplir avec le pays et la date de naissance**
4. **Indiquer son e-mail et le mot de passe**
5. **Le Compte est créé !!!**

Vous pouvez aussi utiliser ce mail pour l'exercice :

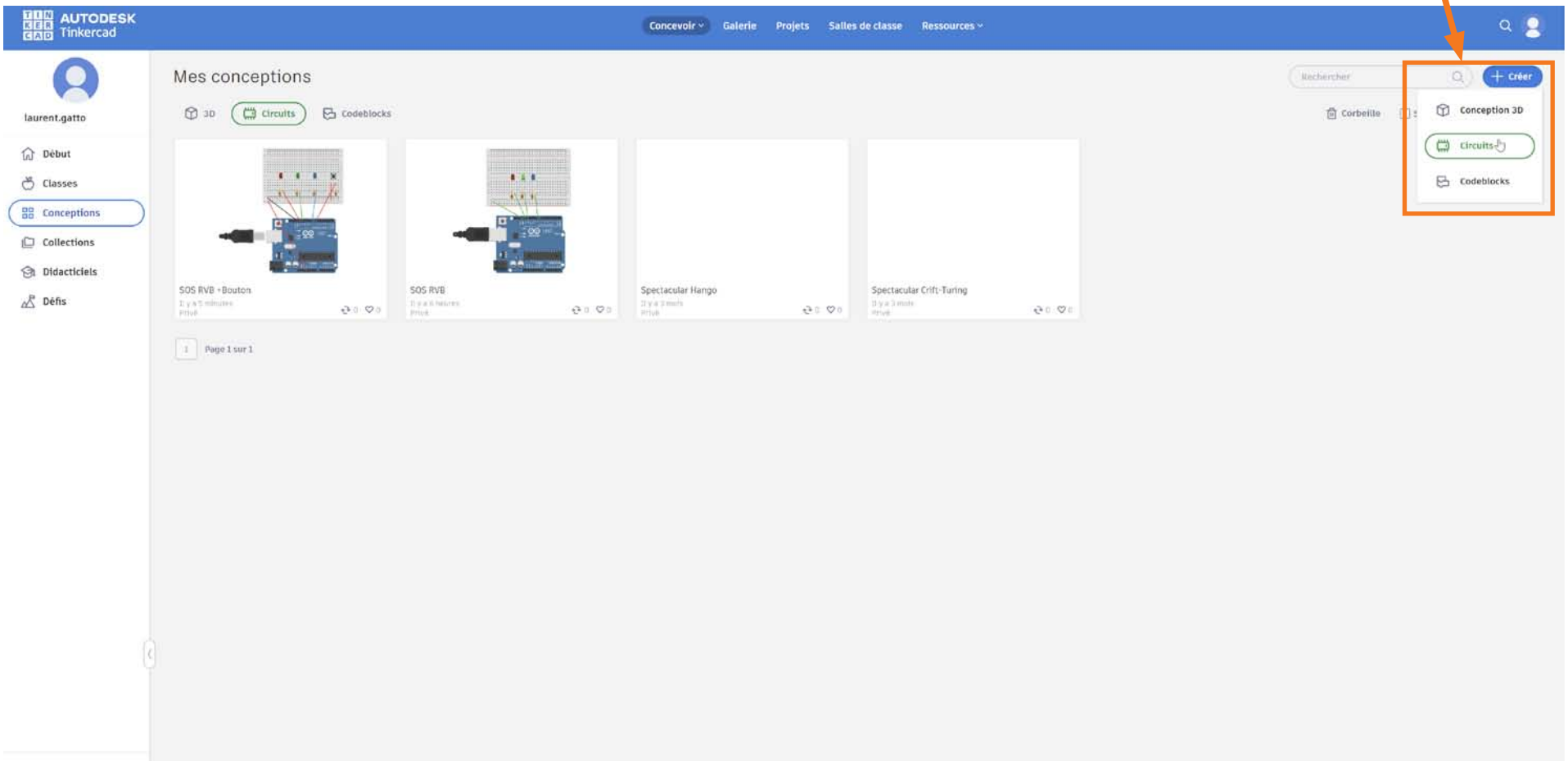
Mail : candidatremn2025@gmail.com

Mot de passe : remn-2025

SOS Arduino Uno ... --- ...

Présentation de l'interface de Tinkercad

Nous allons arriver sur cette page et cliquer sur Circuits

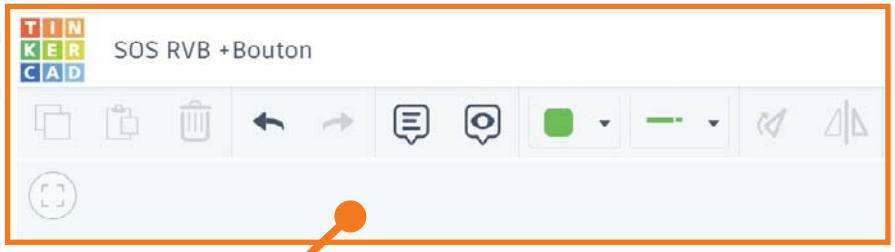


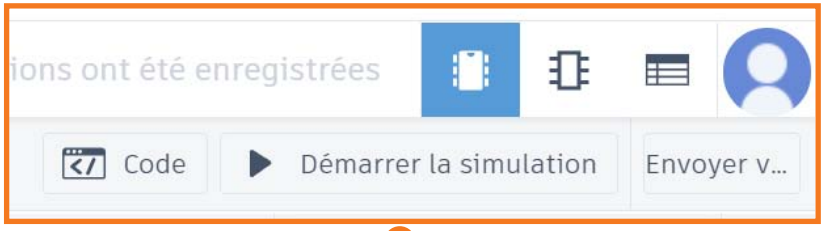
SOS Arduino Uno ... --- ...

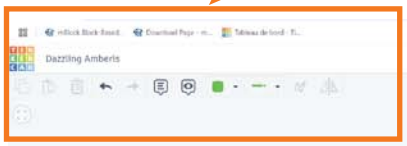
Présentation de l'interface de Tinkercad

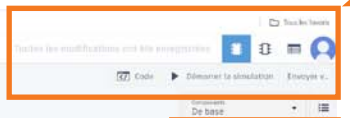
Nous sommes enfin sur le plan de travail de Tinker !

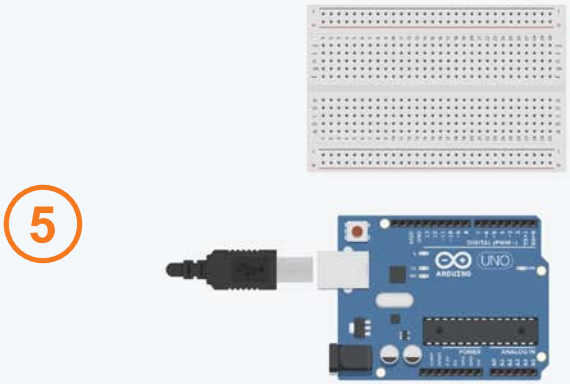
1 : Copier, Coller, Supprimer, Annuler, Notes, Couleurs et type de fils, Symétrie et Rotation.
2 : Passage en mode «Code», Démarrage de la simulation...
3 : Composants divers (Résistances, Leds, Boutons...).
4 : Plan de travail **5 :** Assemblage avec Arduino+Breadboard+Composants+Fils...









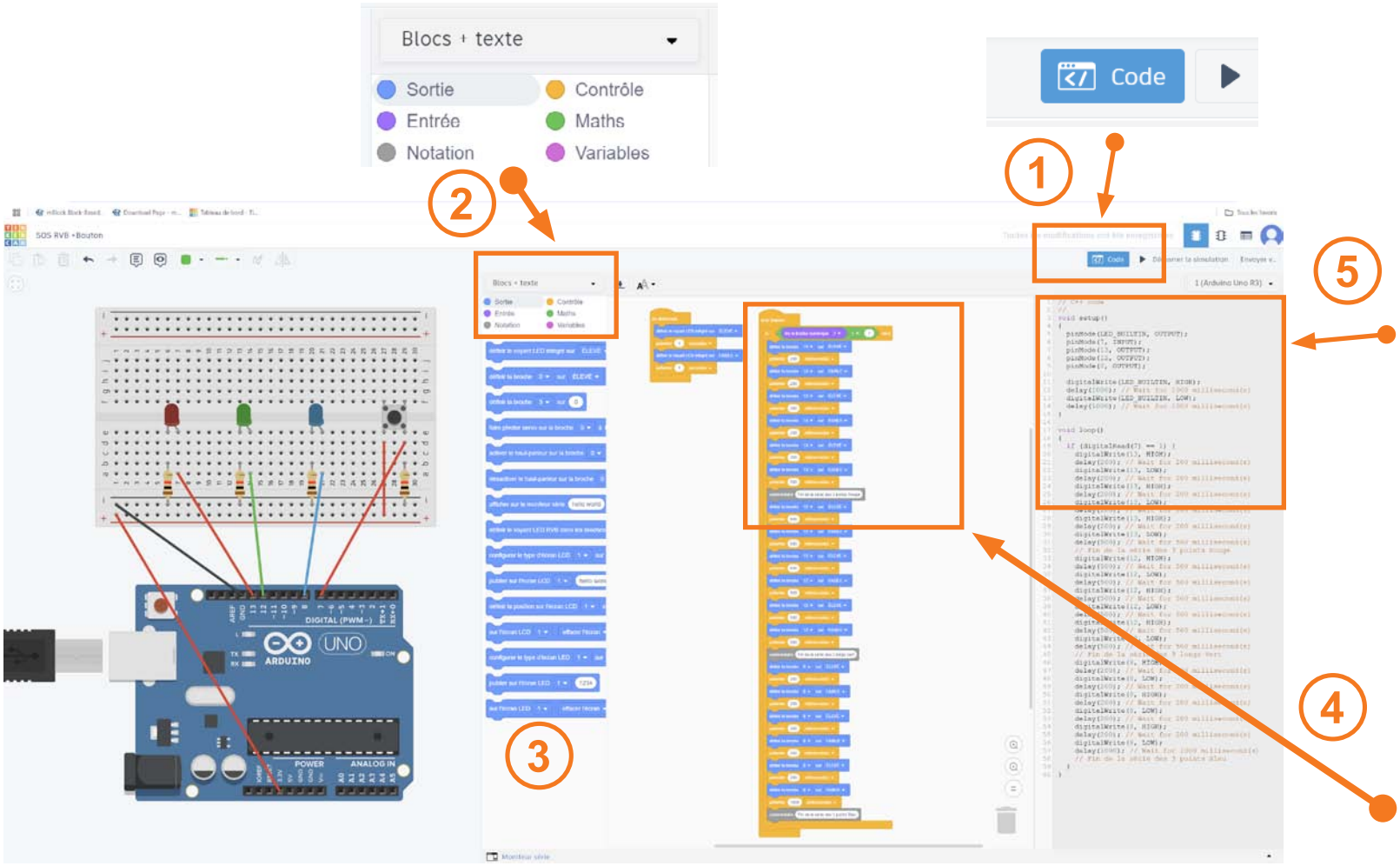


SOS Arduino Uno ... --- ...

Présentation de l'interface de Tinkercad

Nous sommes enfin sur le plan de travail de Tinker !

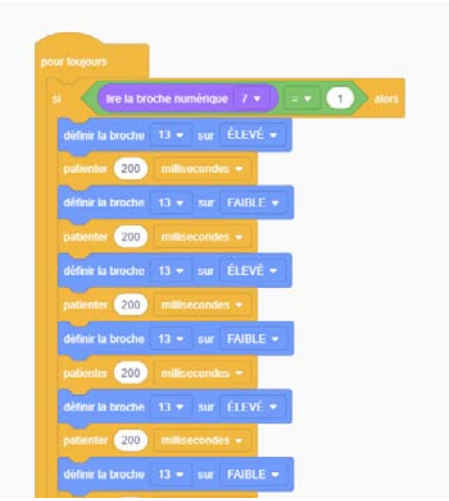
1 : Le passage en mode Code nous permet de passer de l'onglet «Composants» à l'onglet «Programmation».
2 : La zone des blocs par catégories.
3 : La liste des blocs classés par catégorie / couleurs / type d'instruction est alors accessible.
4 : La zone de travail pour assembler les blocs aussi.
5 : un clic sur «Blocs+texte» permet d'accéder à la partie contant le code en langage C++.




```
// C++ code
//
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(7, INPUT);
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(8, OUTPUT);

  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000); // Wait for 1000 millis
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000); // Wait for 1000 millis
}

void loop()
{
  if (digitalRead(7) == 1) {
    digitalWrite(13, HIGH);
    delay(200); // Wait for 200 millis
    digitalWrite(13, LOW);
```



SOS Arduino Uno ... --- ...

Présentation de l'interface de programmation

Il y a en tout 6 catégories de blocs / instructions rangés par thématique / couleur.

1 - Sortie (permet d'utiliser les LEDS, le haut parleur, écran...).

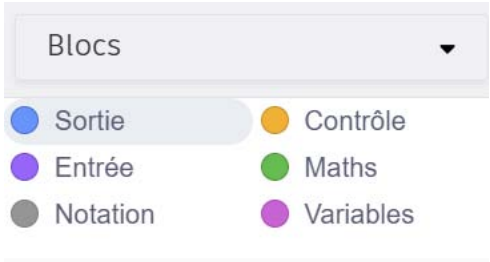
2 - Entrées (permet de lire des valeurs sur les broches de l'Arduino).

3 - Notation (permet de commenter le code afin de le rendre plus compréhensible).

4 - Contrôle (les pauses, conditions, boucles...).

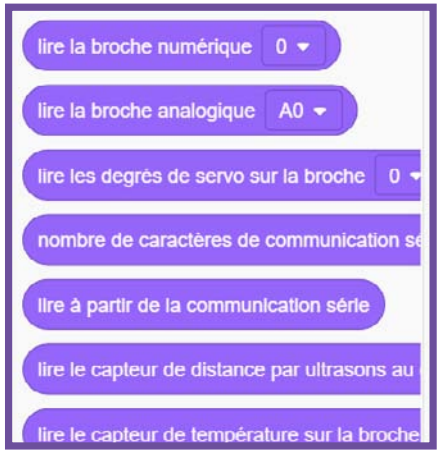
5 - Maths (Opération logiques, comparaisons...).

6 - Variables (personnalisées pour le code...).



1

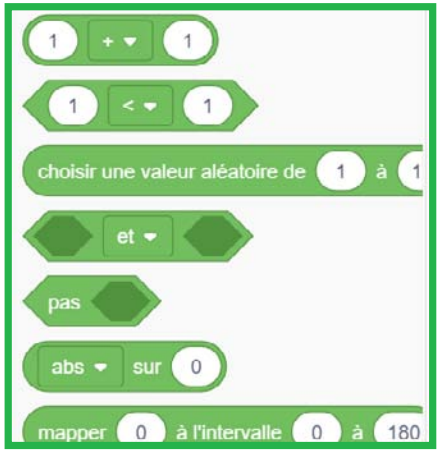
2



3

4

5



6

SOS Arduino Uno ... --- ...

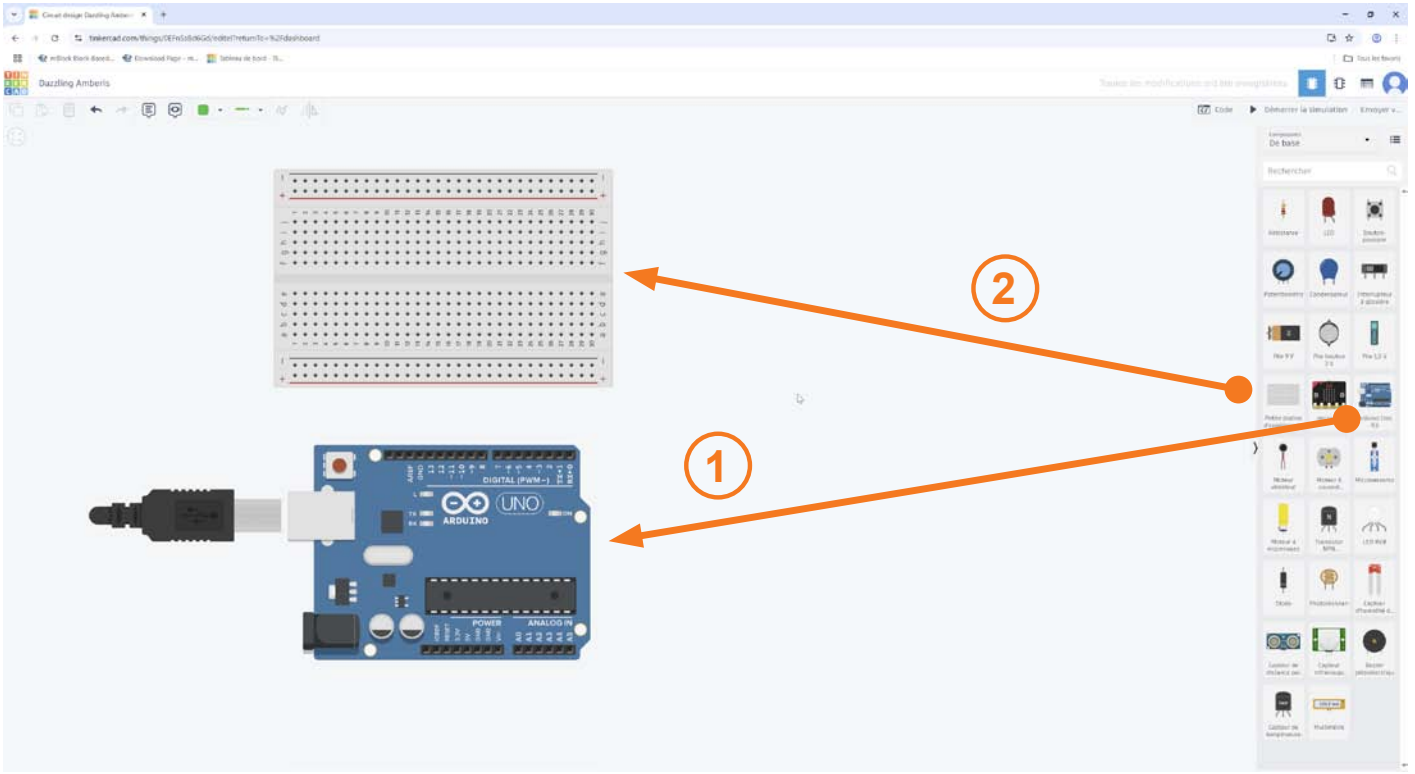
Premier Programme : Faisons clignoter la LED intégrée

**Maintenant que nous avons fait connaissance avec l'interface de programmation,
nous allons pouvoir faire notre premier programme avec comme objectif de faire clignoter la LED !**

Mais avant de commencer nous allons devoir préparer notre plan de travail en y installant l'Arduino et une Breadboard (planche à pain) qui nous servira plus tard pour connecter plus facilement des composants avec l'Arduino et entre eux.

**Nous allons simplement cliquer-glisser l'Arduino (1) puis la Breadboard (2)
de la zone des composants à droite vers le plan de travail.**

A noter : il est à tout moment possible de déplacer le plan de travail avec un cliquer glisser et de zoomer avec Ctrl + Molette de la souris. La touche F ou l'icône en haut à gauche permettent d'ajuster automatiquement le zoom.



SOS Arduino Uno ... --- ...

Premier Programme : Faisons clignoter la LED intégrée

Nous sommes prêt et allons pouvoir faire notre premier programme !

Objectif : faire clignoter la LED intégrée * façon **SOS (... --- ...) !**

Avec un cycle continu de ; 3 clignotements courts, puis 3 longs et enfin 3 courts.

Nous allons cliquer sur le bouton «Code» en haut à droite de la fenêtre.

Par défaut nous avons des blocs prêts à l'usage :

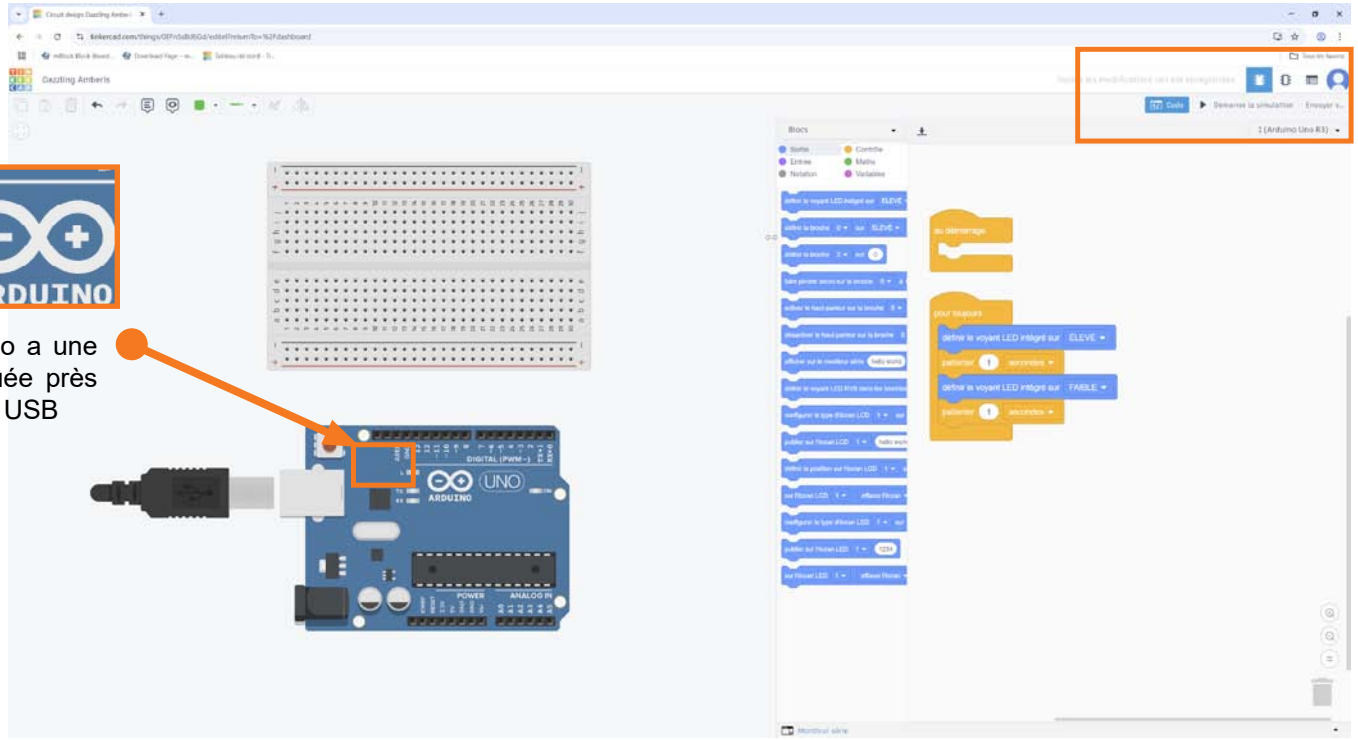
A - Le bloc **«au démarrage»** qui contiendra des blocs / instructions s'exécutant uniquement une fois au démarrage.

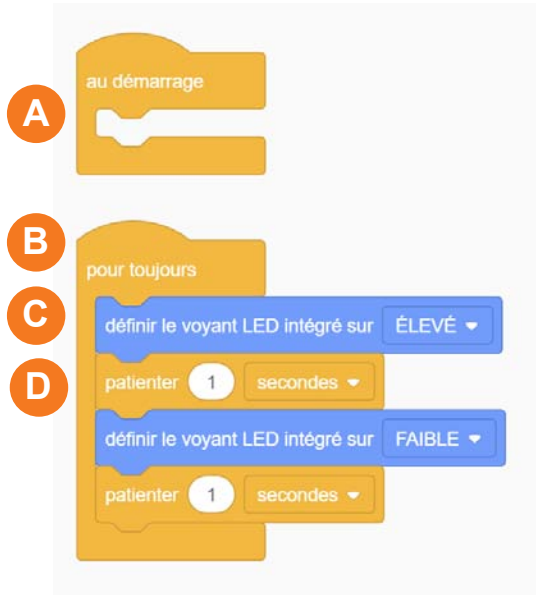
B - Le bloc **«pour toujours»** de la catégorie **«Contrôle»** pour permettre au programme de ne pas s'arrêter.

C - Les bloc **«définir le voyant LED intégré sur...»** de la catégorie **«Sortie»** pour allumer / éteindre la LED intégrée.

D - Les bloc **«patienter ...»** de la catégorie **«Contrôle»** pour faire une pause dans le programme.

* L'Arduino a une petite Led située près du connecteur USB

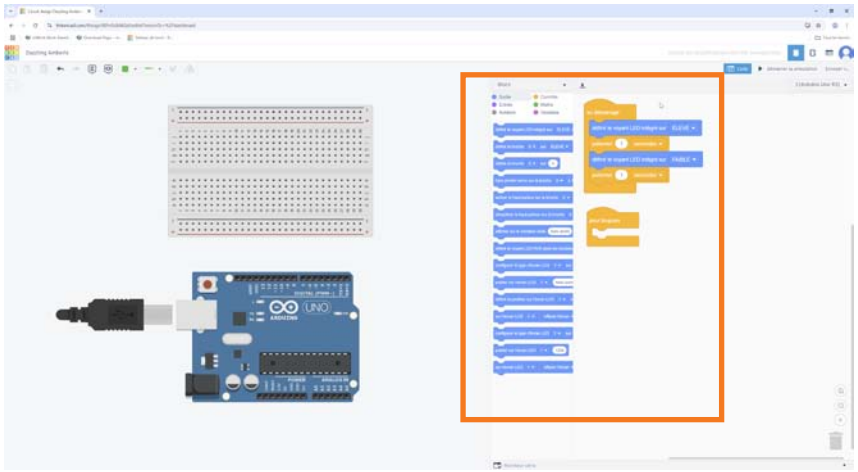




SOS Arduino Uno ... --- ...

Premier Programme : Faisons clignoter la LED intégrée

Nous voulons démarrer notre programmation à partir de zéro.

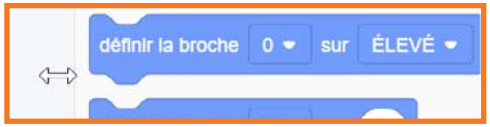


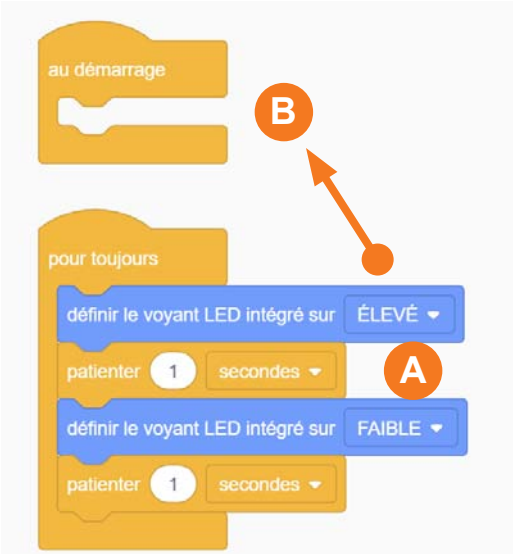
- Nous allons donc déplacer la série de **bloc A** à l'intérieur du **bloc B** avec un cliquer glissé **(1 - 2) pour parvenir au x blocs 3.**

Ce qui signifie que chaque fois que l'Arduino démarrera il exécutera cette séquence une seule fois (la LED intégrée s'allume 1 seconde).

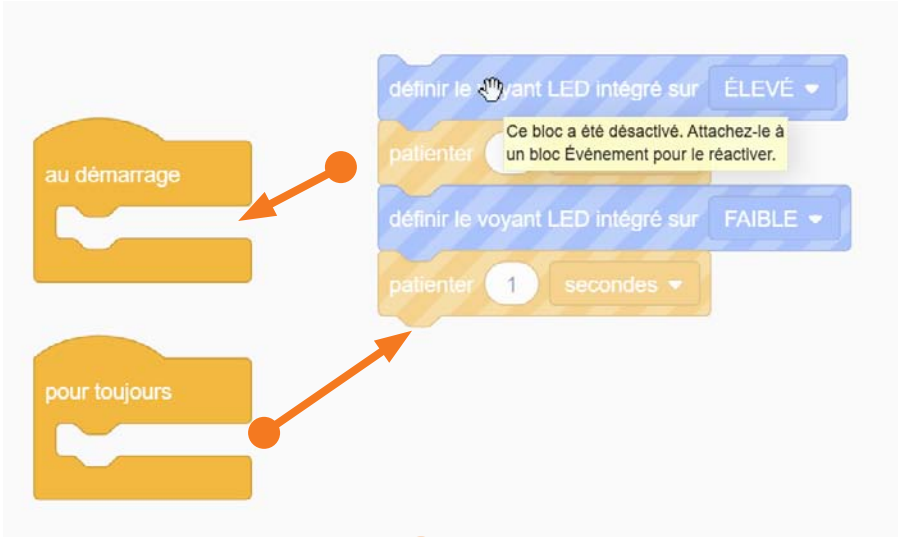
- A partir de maintenant nous allons disposer tous nos bloc à l'intérieur du bloc **« pour toujours »** de façon à ce que le programme s'exécute et tourne en continue.

A noter : Pour avoir plus de place pour ses blocs il est possible d'agrandir la partie Code en cliquant glissant sur le bord gauche de cette partie.





1



2

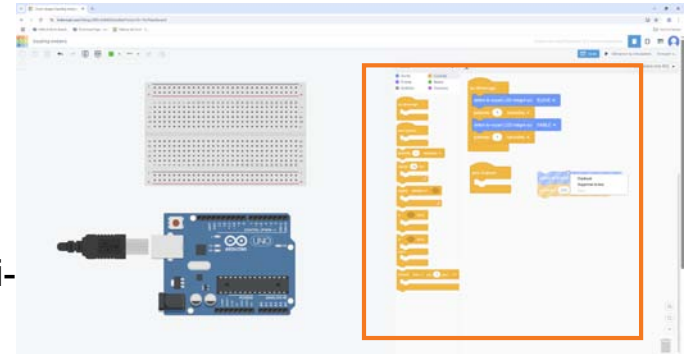
3

SOS Arduino Uno ... ___ ...

Premier Programme : Faisons clignoter la LED intégrée

Nous allons faire la première série de 3 «points» du SOS.

- Pour cela nous allons avoir besoin du bloc **«définir la broche 13 sur...»** de la catégorie **«Sortie»** et du bloc **«patienter ...»** de la catégorie **«Contrôle».**



- Pourquoi ne pas utiliser le bloc **«définir le voyant LED intégré sur...» ?**

La LED intégré est en réalité câblée sur la broche 13 de l'Arduino.
Donc les instructions : **«définir le voyant LED intégré sur...»**
et «défini la broche 13 sur...» ont le même résultat !

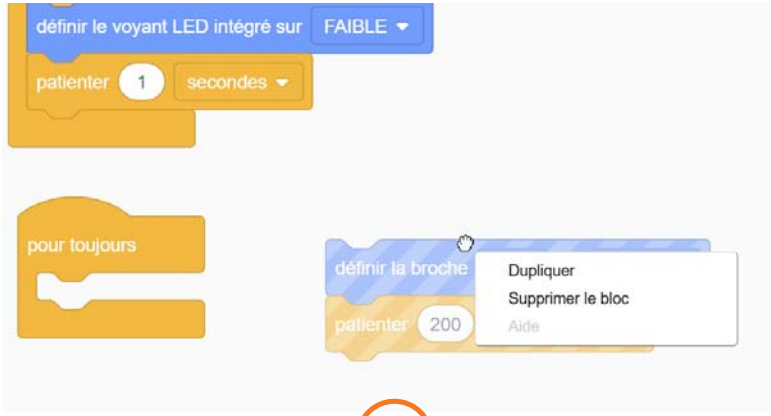
L'avantage de **«définir la broche 13 sur...»** étant que nous pourrons plus tard facilement changée la broche de sortie pour de futures LEDS en changeant juste 13.

- Nous allons provoquer un clignotement rapide (200 ms) de la LED :

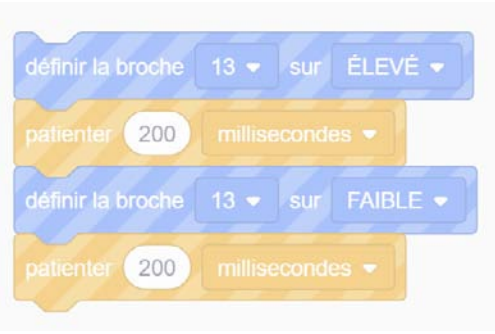
1 - Nous allons créer un premier groupe de blocs en utilisant **«défini la broche 13 sur...»** et **«patienter 200 ms»**
2 - Que nous allons dupliquer avec un clic droit
3 - Pour arriver à ce résultat.

Penser à changer **«élevé»** (qui veut dire allumé) en **«faible»** (éteint) sur le second bloc **«définir la broche 13 sur...»**.
A noter : les blocs sont «ternes» et peu lisible tant qu'ils ne sont pas actif (insérés dans le bloc «pour toujours»).

1



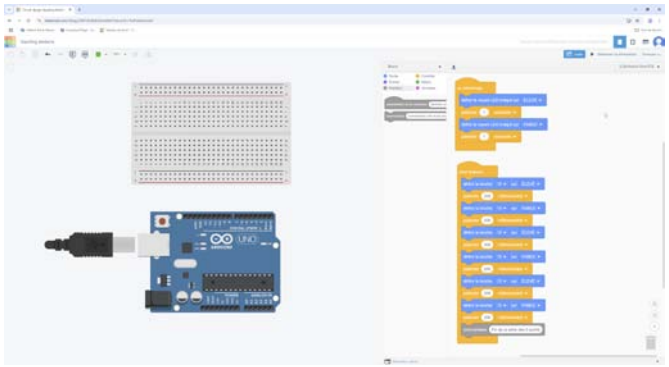
2



3

SOS Arduino Uno ... ___ ...

Premier Programme : Faisons clignoter la LED intégrée

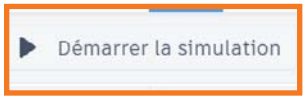


Nous allons faire la première série de 3 «points» du SOS.

1 - Nous allons dupliquer à nouveau notre groupe de 4 blocs.

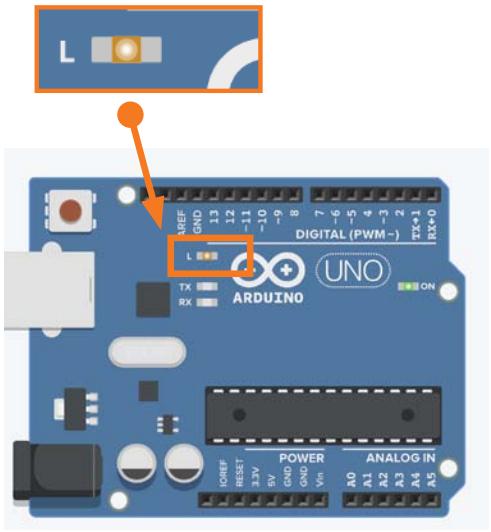
2 - De façon à avoir une **série de 3 x 4 blocs** pour les clignotements courts.
Puis nous rajouterons un commentaire à la fin de la série des 3 clignotements courts de façon à repérer plus facilement cette série (2).

3 - Il ne reste plus qu'à insérer la série de blocs dans le bloc **«pour toujours»** avec un cliquer glisser pour rendre active cette séquence.

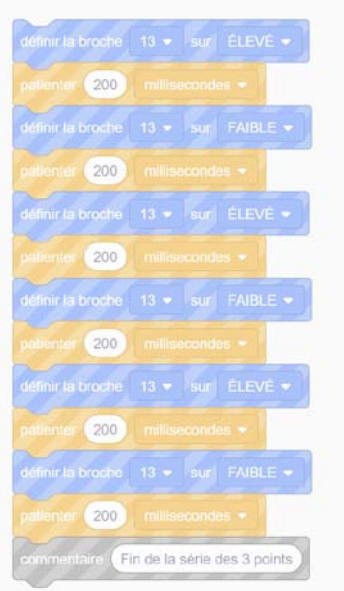


● Nous pouvons désormais tester notre programme en cliquant sur «Démarrer la simulation» en haut à droite.
Nous devrions désormais voir dans la simulation la led intégrée à l'Arduino clignoter.

Bravo !!! ;)



1



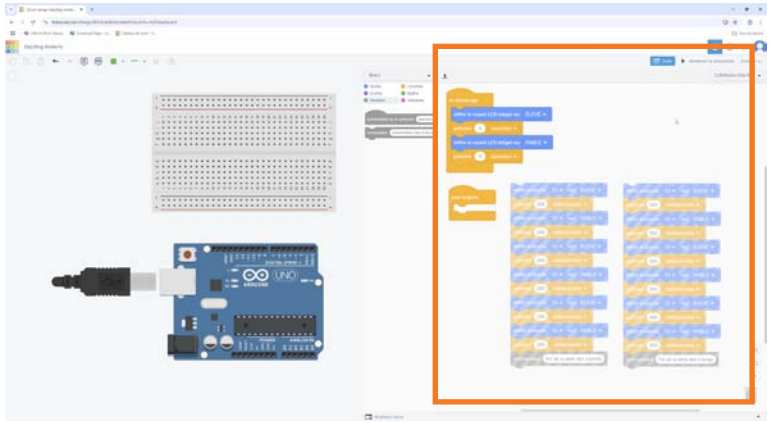
2

3

SOS Arduino Uno ... --- ...

Premier Programme : Faisons clignoter la LED intégrée

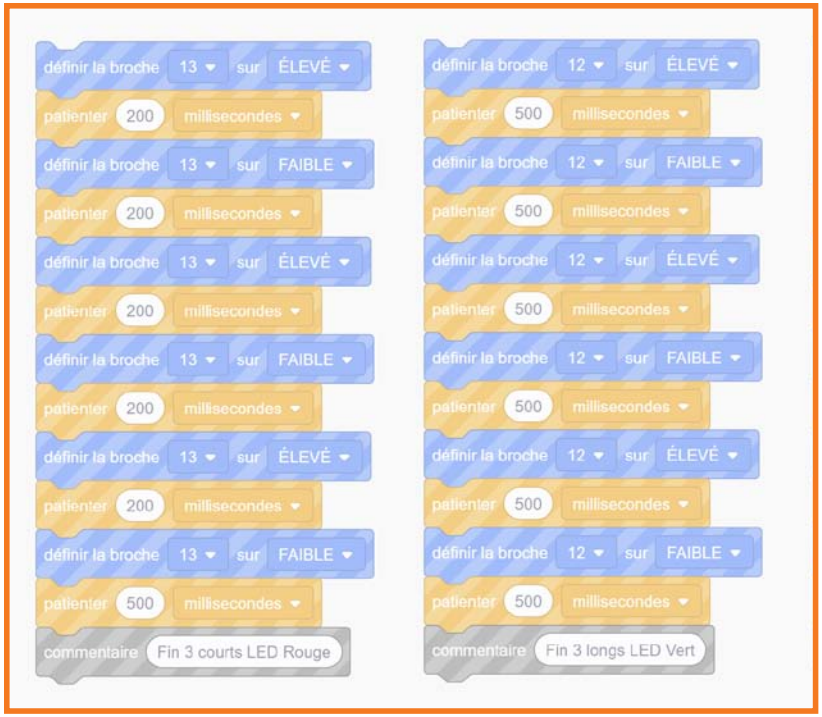
Nous allons désormais faire la première série de 3 «traits» du SOS puis la dernière série de 3 «points».



1 - Nous allons dupliquer à nouveau notre première série

2 - Puis modifier les temps à 500ms de façon à créer les 3 «traits» (clignotement plus long) du SOS (2).
Puis nous allons modifier le commentaire à la fin de la série
des 3 clignotements longs de façon à repérer plus facilement cette série.

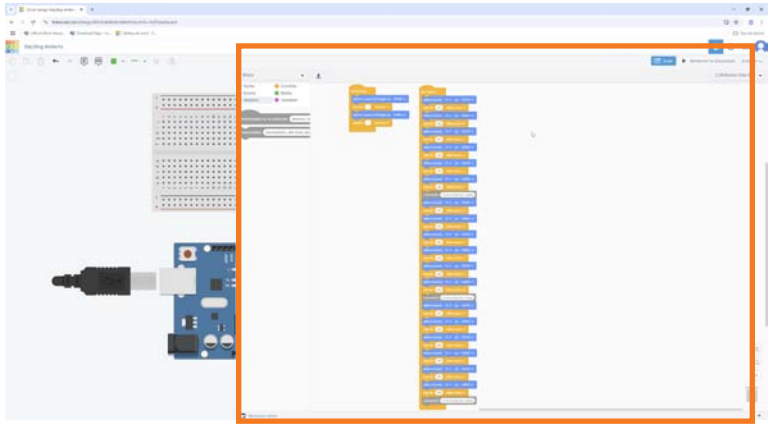
3 - Il ne reste plus qu'à dupliquer la première série de blocs correspondant aux trois points de façon à avoir nos **«... --- ...»** sur 3 séries de blocs.



1 2 3

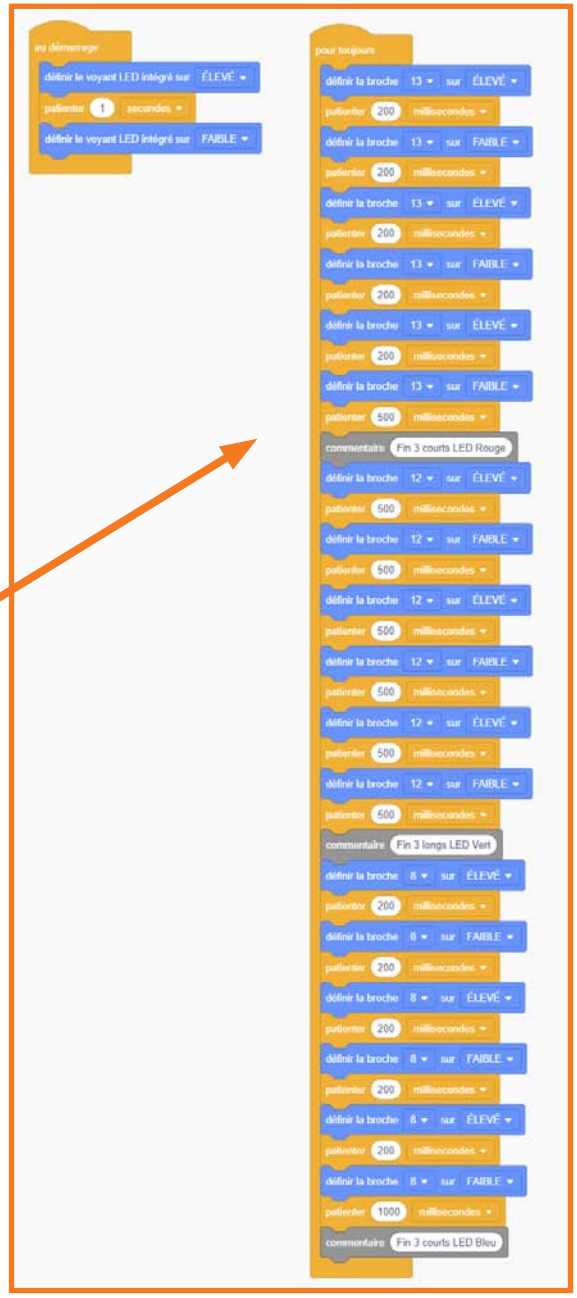
SOS Arduino Uno ... --- ...

Premier Programme : Faisons clignoter la LED intégrée

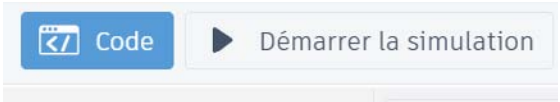


Nous allons désormais «activer» les 3 séries de clignotements du SOS.

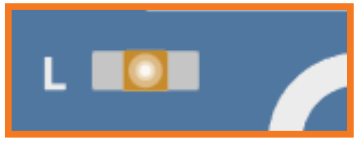
- Il ne reste plus qu'à insérer la série des 3 blocs (dans le bon ordre : 3x200ms - 3x500ms - 3x200ms) dans le bloc **«pour toujours»** avec un cliquer glisser.



- Nous pouvons désormais tester notre programme en cliquant sur «Démarrer la simulation» en haut à droite de la fenêtre.



La LED intégrée de l'Arduino devrait nous faire un joli SOS !



SOS Arduino Uno ... --- ...

Exercice !!!

Modifiez la longueur des clignotements courts et longs en modifiant les valeurs des blocs «patienter» et observez le résultat ! ;)

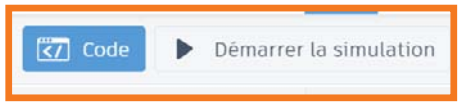
- **Nous les avions fixés à :**
200 ms pour les clignotements courts
et à 500 ms pour pour les clignotements longs.

- **Vous pouvez essayé par exemple de les passer :**
à 400 ms pour les clignotements courts
et à 700 ms pour pour les clignotements longs.

- Avec des copier-coller : Mettez à jour les valeurs de tous les blocs «patientez»

- **Essayez aussi d'augmenter le délai du dernier «patienter»**
de façon à bien marquer la fin du SOS avec 1000ms (soit une seconde).

- Nous pouvons désormais tester notre programme en cliquant sur «Démarrer la simulation» en haut à droite de la fenêtre.
Et pouvoir voir l'impact des timings sur les clignotements.



SOS Arduino Uno ... — — ...

Premier Programme : Faisons clignoter la LED intégrée

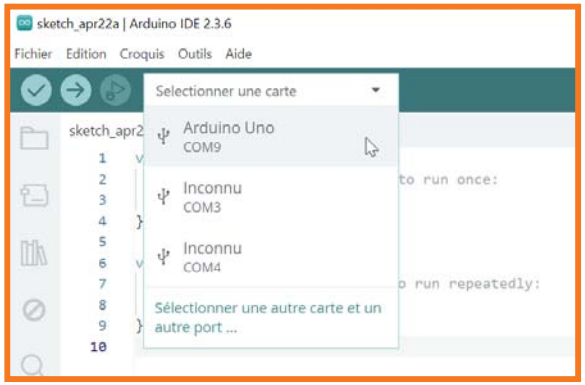
Maintenant que la simulation fonctionne, nous allons désormais pouvoir vérifier notre programme en condition réelle sur l'Arduino :

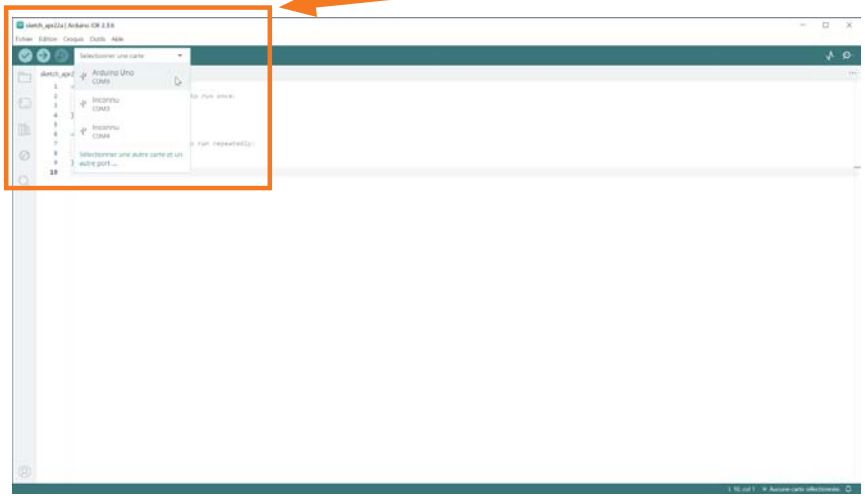
- Il nous faut générer le code C++
- Puis téléverser ce code sur l'Arduino gràçe à l'IDE.

Il faut tout d'abord installer l'IDE (Environnement de Développement Intégré) Arduino :
https://www.arduino.cc/en/software/

Lancer l'IDE puis connecter en USB notre Arduino sur le PC avec son câble USB pour vérifier que tout fonctionne :

- En cliquant sur «Sélectionner une carte» nous devons voir apparaître «Arduino Uno» avec le numéro de port correspondant (ex : COM9).
- Si jamais la carte n'apparaît pas il faut aller la chercher dans «Sélectionner une autre carte et un autre port».





SOS Arduino Uno ... --- ...

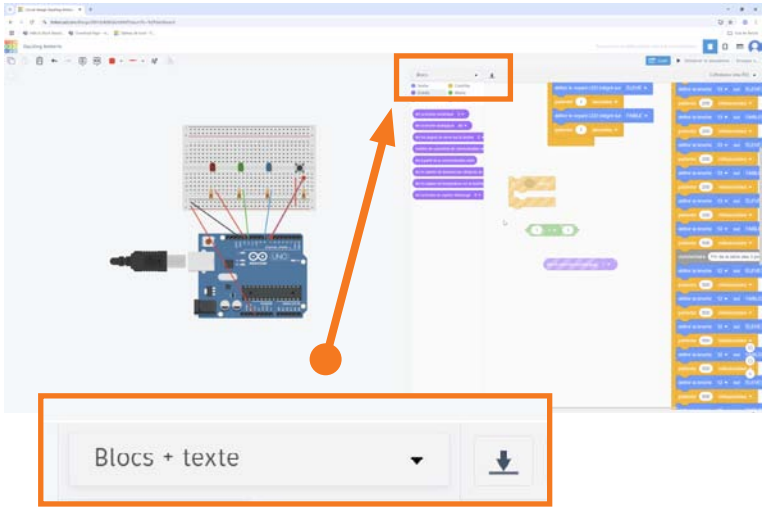
Premier Programme : Faisons clignoter la LED intégrée

Si l'Arduino est reconnu dans l'IDE nous allons pouvoir récupérer le code généré dans Tinkercad et le téléverser sur l'Arduino, il y a deux méthodes :

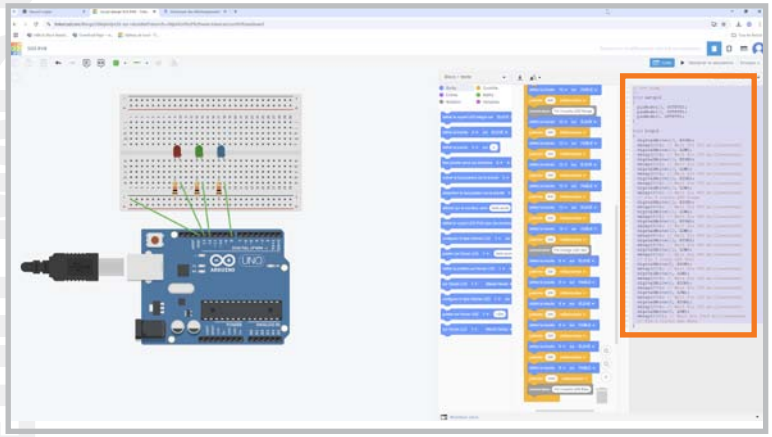
- Télécharger le code du programme à partir de Tinkercad en cliquant sur l'icône : 

Le fichier téléchargé (normalement dans le dossier téléchargement) a une extension .ino :

Nous alors ouvrir directement ce fichier .ino avec l'IDE qui va nous demander la création d'un dossier. Nous validons la création avec «OK» :

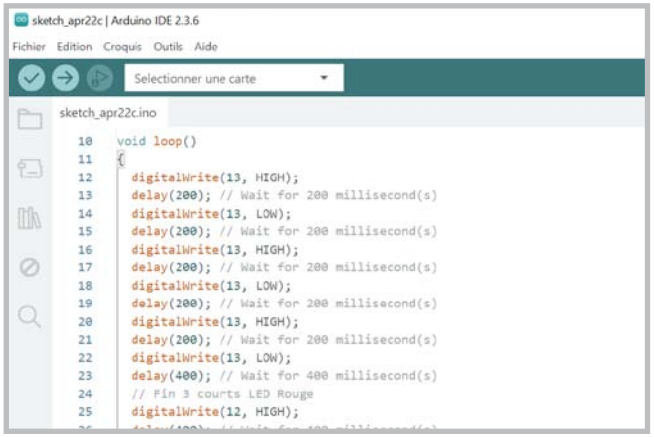


- Ou bien nous pouvons aussi directement copier-coller le code de la partie code de Tinkercad (en mode Bloc + texte) vers l'IDE :



1 **Dans la partie code de Tinkercad :**
Tout Sélectionner : Ctrl + A puis Copier : Ctrl+C

2 **Dans l'IDE de l'Arduino :**
Tout Sélectionner : Ctrl + A



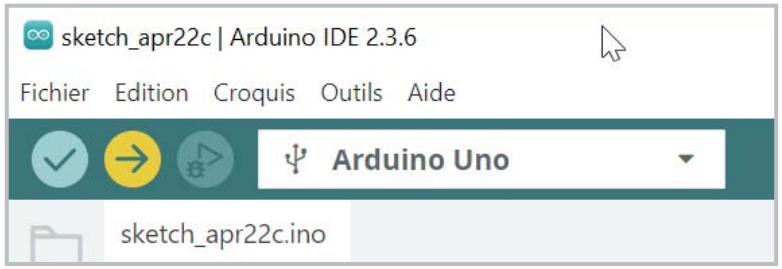
3 **Dans l'IDE de l'Arduino :**
puis Coller : Ctrl+V

SOS Arduino Uno ... --- ...

Premier Programme : Faisons clignoter la LED intégrée

Il ne reste plus qu'à compiler (transformer le langage interprété avec des mots en 0 et 1 pour l'Arduino) puis téléverser (upload / envoyer) le programme du PC vers l'Arduino connecté :

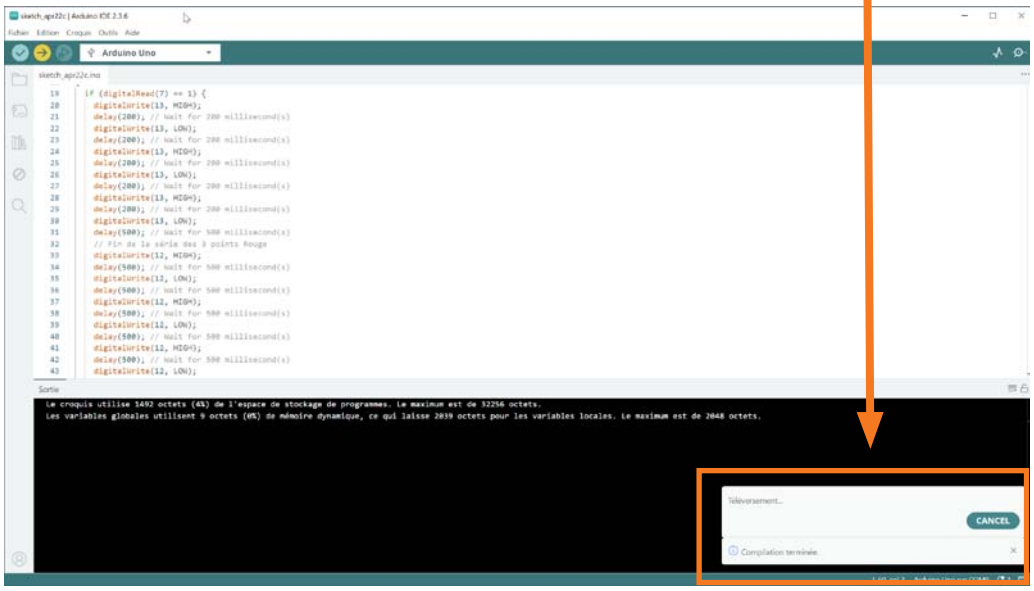
- Nous pouvons cliquer sur  pour vérifier qu'il n'y a pas d'erreurs.



- ou bien directement téléverser le programme compilé sur l'Arduino en cliquant sur 

En bas de la fenêtre de l'IDE, dans la partie noire, nous devons avoir un message qui nous indique l'espace occupé par le «croquis» (programme) ainsi qu'une fenêtre nous indiquant que la compilation et le téléversement sont finis.

Le programme va alors immédiatement s'exécuter sur l'Arduino ! ;)



A noter : il faut que l'Arduino soit branché électriquement (port USB ou Batterie USB pour fonctionner).

En cas de soucis : Réinitialiser l'Arduino avec le petit bouton «Reset».

SOS Arduino Uno ... --- ...

Second Programme : le SOS se fait avec 3 Leds !

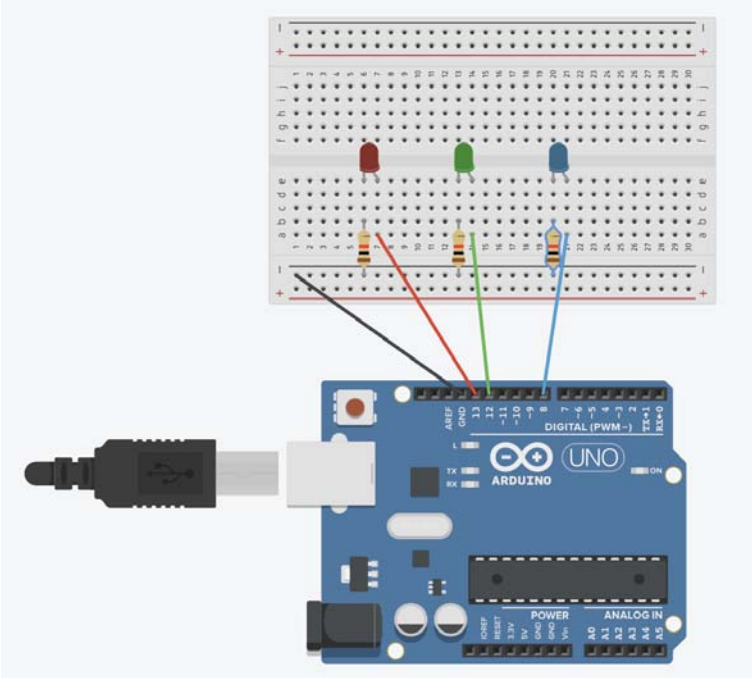
Objectif : Nous allons maintenant essayer de faire clignoter non plus la LED intégrée de l'Arduino mais 3 LEDS externes de couleurs différentes qui chacune afficheront une partie du SOS.

Exemple (vous pouvez choisir vos couleurs) :

La première Led Rouge affichera la première séquence des 3 points **(3 clignotements courts).**

La seconde Led Verte affichera la seconde séquence des 3 traits **(3 clignotements longs).**

La troisième Led Bleue affichera la troisième séquence des 3 points **(3 clignotements courts).**



SOS Arduino Uno ... --- ...

Second Programme : le SOS se fait avec 3 Leds !

3 étapes :

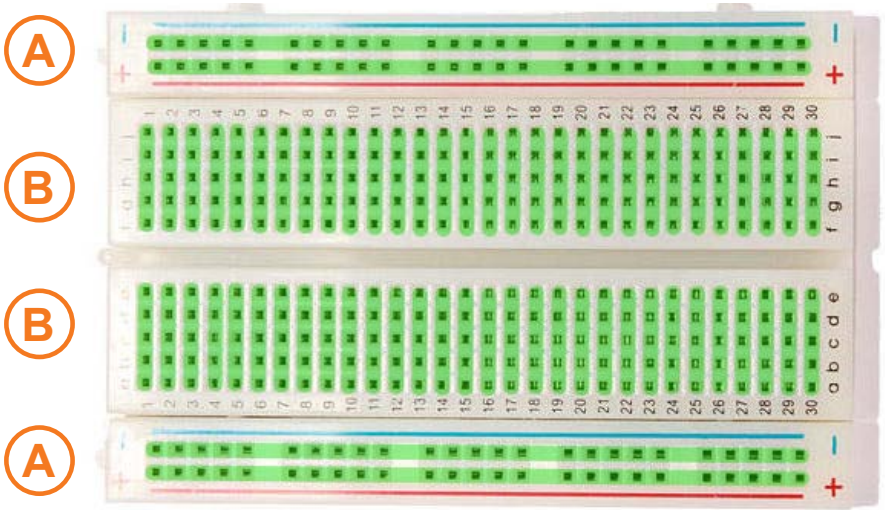
1 - Poser les LEDS sur la breadboard avec un cliquer glissé de la partie composants à droite et commencer à câbler la breadboard et l'Arduino en choisissant la couleur des câbles.

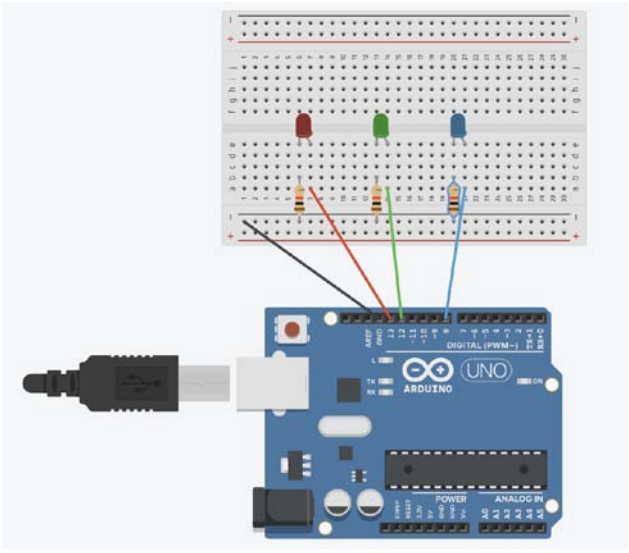
2 - Poser les résistances sur la breadboard avec un cliquer glissé de la partie composants à droite vers la breadboard et faire le câblage.

3 - Modifier le programme pour qu'il allume et éteigne les LEDS externes.

A noter : La breadboard a un câblage interne (non visible) particulier.
A - Les - et les + sont câblés à «l'horizontale» et ne sont bien sur pas reliées entre eux.
B - Toutes les autres partie (repérées par des lettres et chiffres) sont câblées à la «verticale» et ne sont pas reliées entre elles.

Le courant ne peut donc circulé que sur les lignes vertes de ce schéma,
et les composants doivent y être posés en fonction de ce schéma, de ces lignes et des règles de conduction de l'électricité.

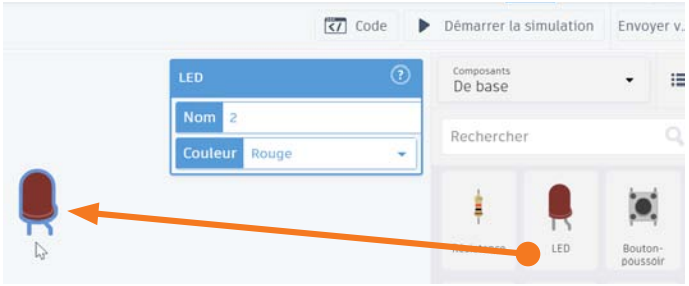




SOS Arduino Uno ... --- ...

Second Programme : le SOS se fait avec 3 Leds !

1 - Poser les Leds et commencer à câbler :

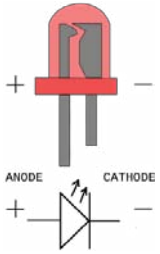


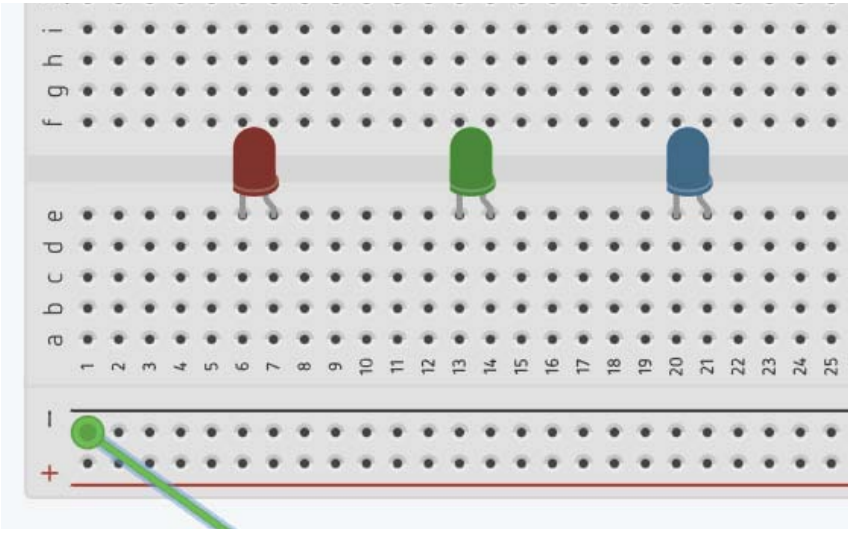
- Nous allons cliquer glisser les Leds de la partie composants à droite vers la breadboard puis choisir les couleurs des Leds.

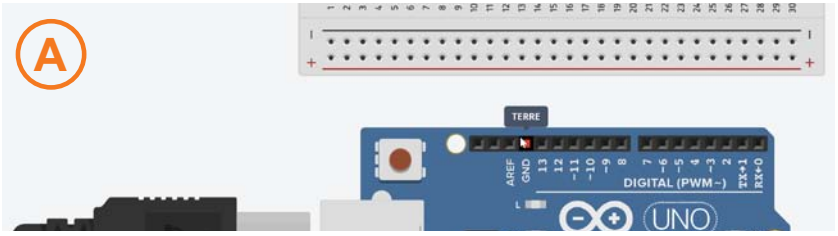
- Nous allons aussi commencer à câbler la breadboard et l'Arduino avec des clics simples sur les broches :

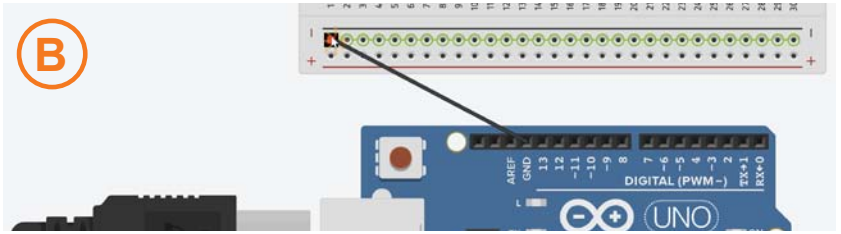
A - du GND (Ground - Masse) de l'Arduino
B - vers le - de la breadboard
C - et choisir la couleur du câble (menu en haut à gauche) : noir (masse en électricité).

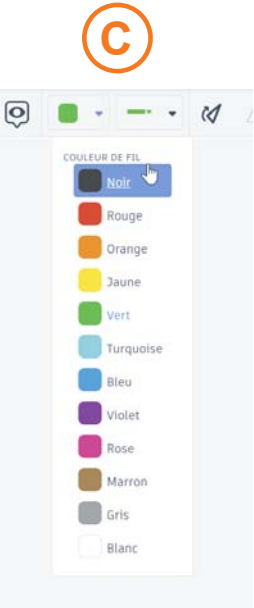
A noter : Les Leds sont «polarisées» elles ont une patte longue (anode) pour le + et une patte courte (cathode) pour le - et doivent donc être branchées dans le bon sens pour fonctionner.







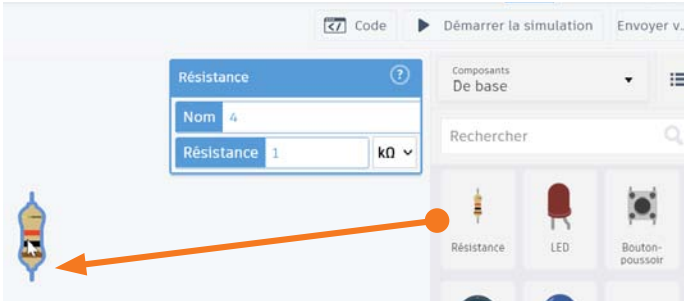




SOS Arduino Uno ... --- ...

Second Programme : le SOS se fait avec 3 Leds !

2 - Poser les résistances et continuer à câbler :

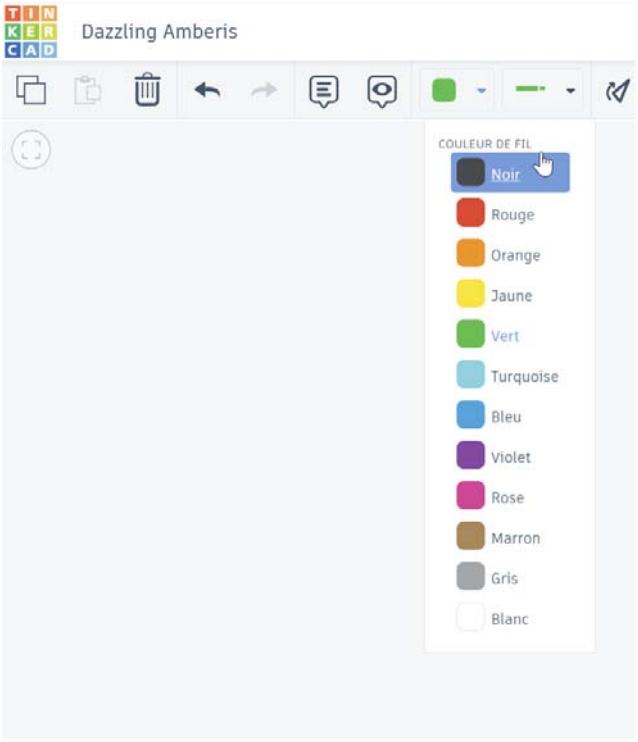


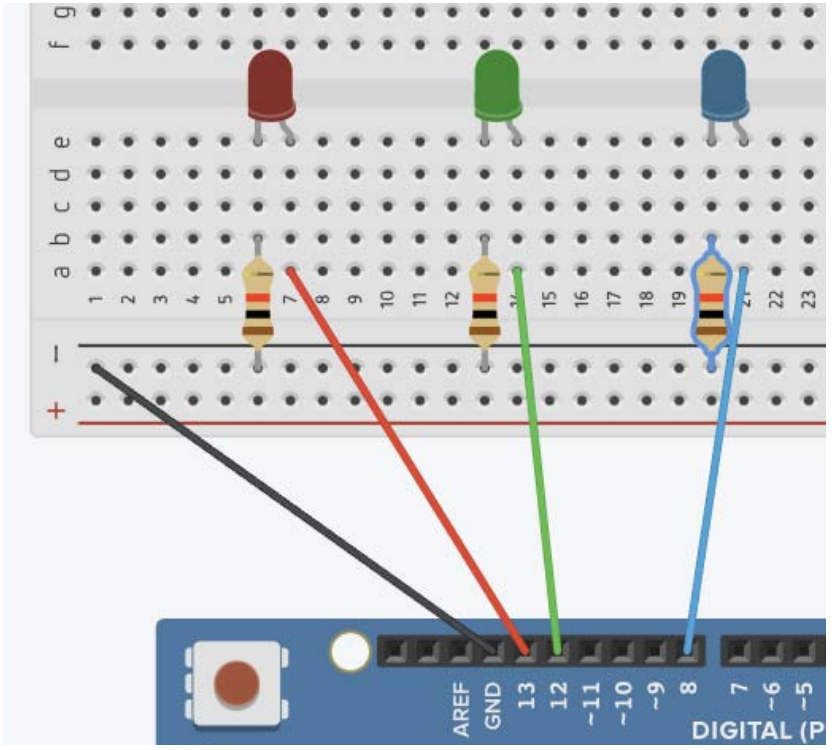
- Nous allons cliquer glisser les résistances de la partie composants à droite vers la breadboard puis vérifier / choisir la valeur des résistances (entre 200 Ohms et 1 k Ohms).

- Nous allons aussi finaliser le câblage entre la breadboard et l'Arduino :

- La Led Rouge est câblée sur **la broche 13** avec **un câble Rouge**.
- La Led Verte est câblée sur **la broche 12** avec **un câble Vert.**
- La Led Bleue est câblée sur **la broche 8** avec **un câble Bleu.**

A noter : Les Leds utilisent environ 3 volt et l'Arduino va jusqu'à 5 volt il faut donc poser une résistance (entre 200 et 1 k Ohms) pour limiter le voltage qui arrive à la Led et ne pas la «griller».





SOS Arduino Uno ... --- ...

Second Programme : le SOS se fait avec 3 Leds !

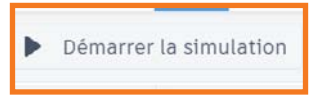
3 - Modifier le programme pour qu'il allume et éteigne les LEDS externes :

- **Dans la partie «Code»** nous allons changer le numéro de la broche pour que celui-ci corresponde à la bonne Led :

1 - La Led Rouge fera les 3 premiers clignotements «courts» : nous ne modifions rien car 13 est la bonne broche.
2 - La Led Verte fera les 3 clignotements «longs» suivants :
nous allons définir toutes les valeurs de la broche (seconde série à 500 ms) à 12.
3 - La Led Bleue fera les 3 derniers clignotements «courts» :
nous allons définir toutes les valeurs de la broche (troisième série à 200 ms) à 8.

- Nous pouvons désormais tester notre programme en cliquant sur «Démarrer la simulation» en haut à droite.

Si nous avons un problème ou un signe qui apparaît :
il faut vérifier la valeur des résistances et le câblage.



1

2

3

SOS Arduino Uno ... --- ...

Second Programme : le SOS se fait avec 3 Leds !

Et maintenant si on testait pour de vrai ?

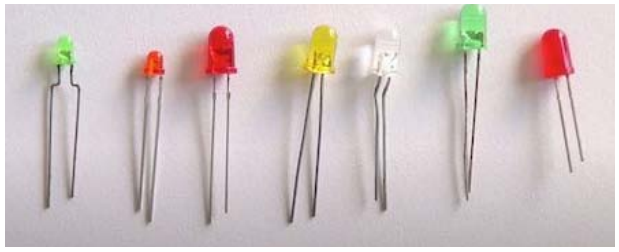
Objectif : basculer de la simulation à la pratique et utiliser l'Arduino Uno, la bread board et les composants.

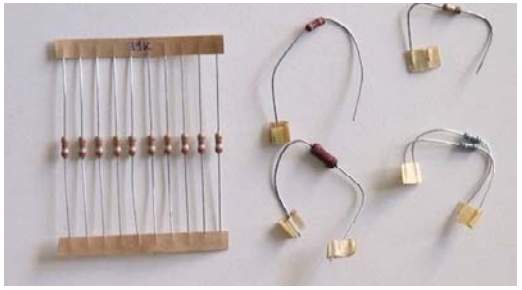
- Nous allons téléverser le programme sur l'Arduino comme nous l'avons déjà fait **(passage de Tinkercad vers l'IDE).**
- Puis nous allons préparer tous nos composants et les câbler à l'identique de la simulation sur Tinkercad.

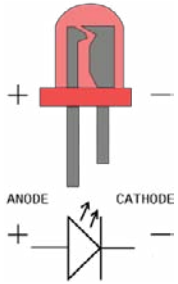
Matériel :

- 3 Leds de couleurs différentes
- 3 résistances dont la valeur va de 200 Ohms à 1 k Ohms
- Des câbles de différentes longueurs.
- Un Arduino Uno et sa breadboard

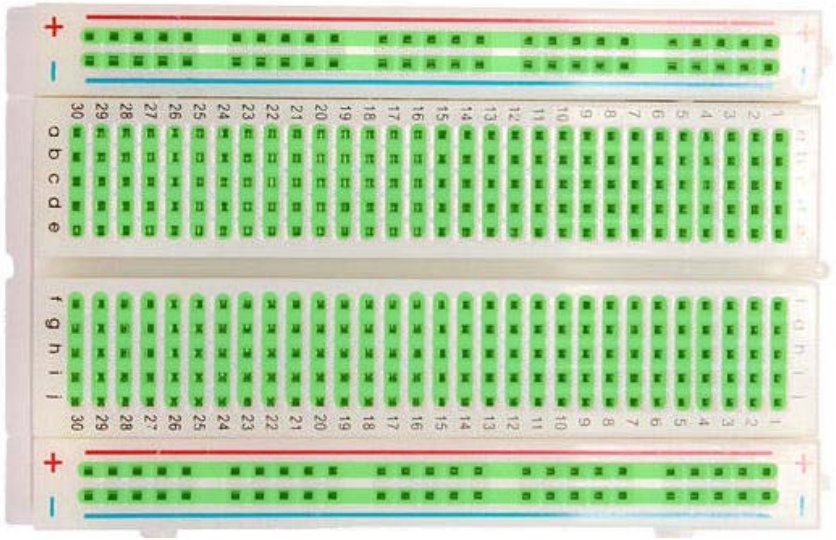
Tout ceci devrait être disponible dans votre kit Arduino.







Led :
Anode +
Cathode -



SOS Arduino Uno ... --- ...

Second Programme : le SOS se fait avec 3 Leds !

Les couleurs des résistances

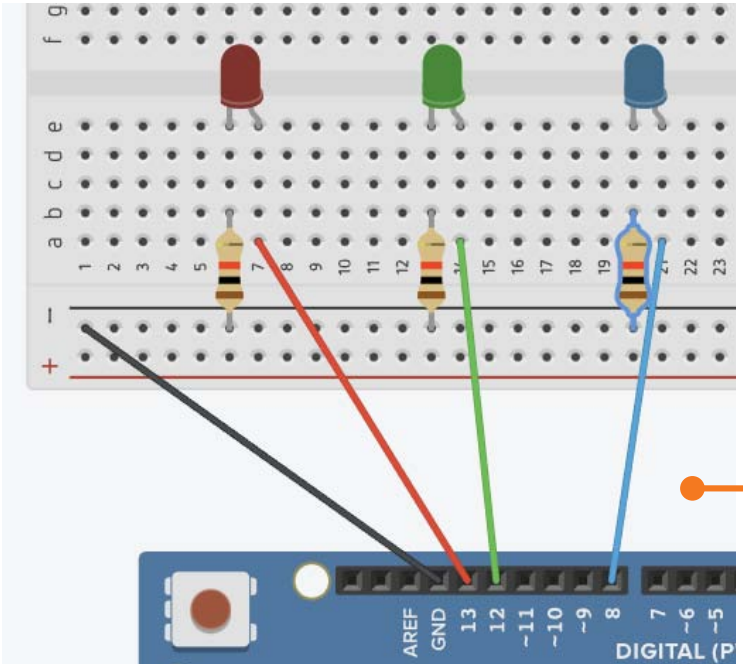
- Vous pouvez vous fier aux couleurs qui apparaissent dans Tinkercad et chercher des résistances avec les mêmes couleurs. **C'est important car votre montage risque de ne pas fonctionner.**

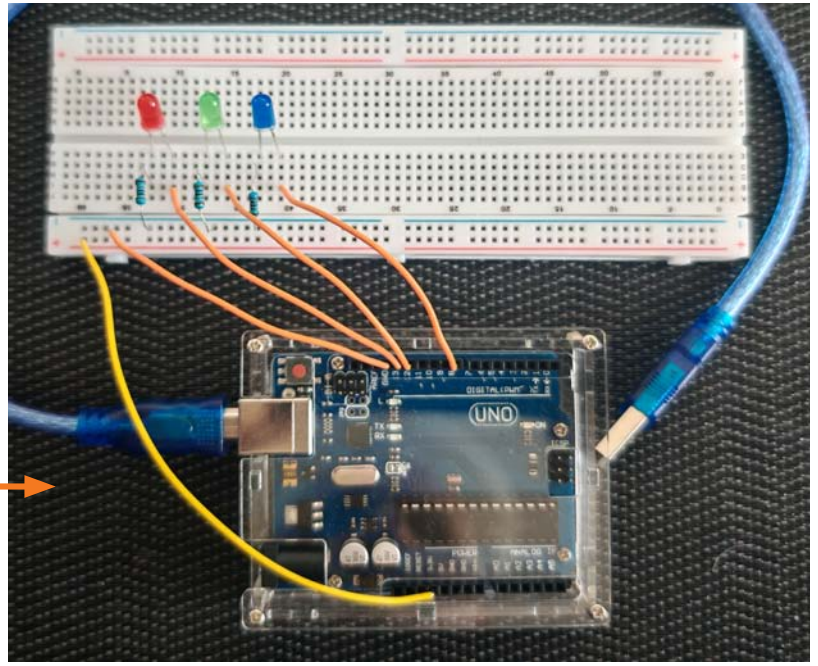
- Les résistances étant très petites il est conseillé d'utiliser l'appareil photo de votre smartphone pour grossir l'image de la résistance et mieux voir les lignes de couleur.

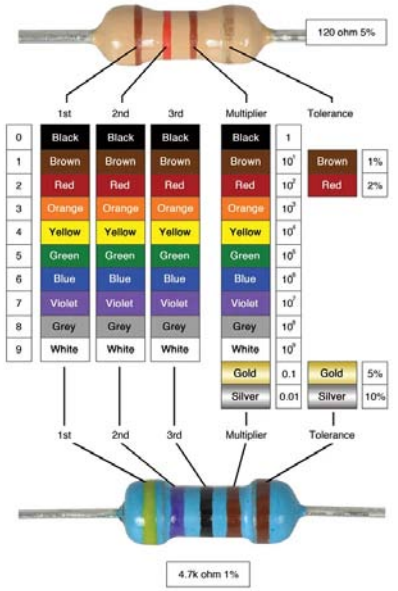
Il existe des sites pour les codes couleurs des résistances :
https://www.digikey.fr/fr/resources/conversion-calculators/conversion-calculator-resistor-color-code

Résistance de 1 k Ohms

En reproduisant le circuit réalisé avec Tinkercad Vous devriez arriver à ce résultat :







SOS Arduino Uno ... --- ...

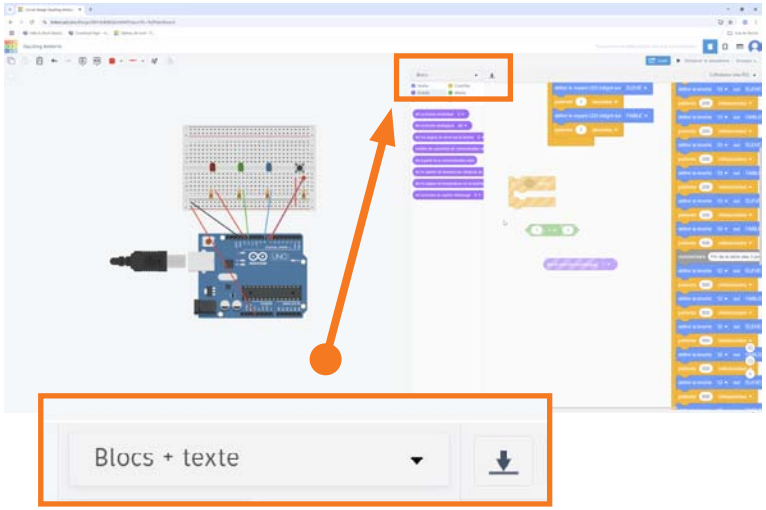
Second Programme : le SOS se fait avec 3 Leds !

Si l'Arduino est reconnu dans l'IDE nous allons pouvoir récupérer le code généré dans Tinkercad et le téléverser sur l'Arduino, il y a deux méthodes :

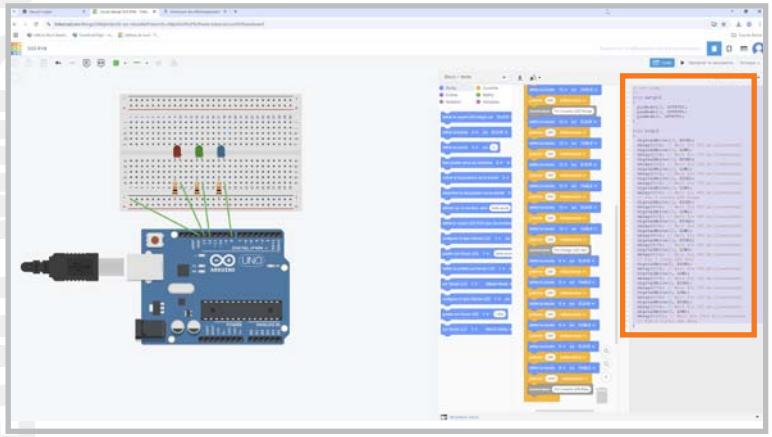
- Télécharger le code du programme à partir de Tinkercad en cliquant sur l'icône : 

Le fichier téléchargé (normalement dans le dossier téléchargement) a une extension .ino :

Nous alors ouvrir directement ce fichier .ino avec l'IDE qui va nous demander la création d'un dossier. Nous validons la création avec «OK» :

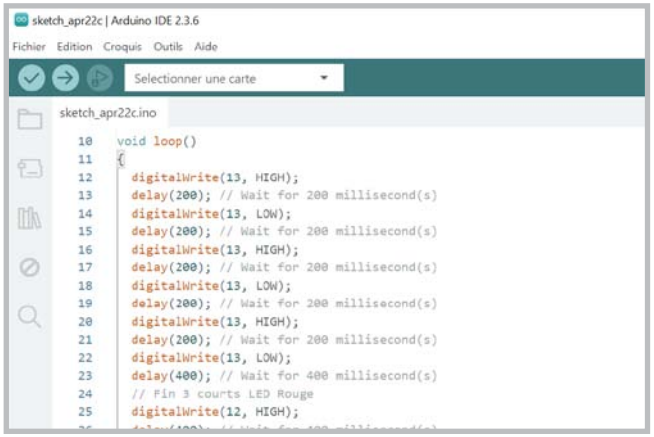


- Ou bien nous pouvons aussi directement copier-coller le code de la partie code de Tinkercad (en mode Bloc + texte) vers l'IDE :



1 **Dans la partie code de Tinkercad :**
Tout Sélectionner : Ctrl + A puis Copier : Ctrl+C

2 **Dans l'IDE de l'Arduino :**
Tout Sélectionner : Ctrl + A



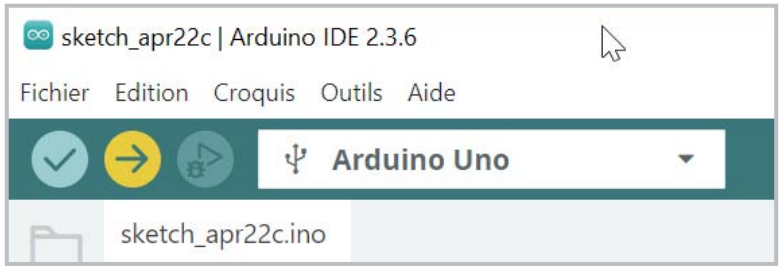
3 **Dans l'IDE de l'Arduino :**
puis Coller : Ctrl+V

SOS Arduino Uno ... --- ...

Second Programme : le SOS se fait avec 3 Leds !

Il ne reste plus qu'à compiler (transformer le langage interprété avec des mots en 0 et 1 pour l'Arduino) puis téléverser (upload / envoyer) le programme du PC vers l'Arduino connecté :

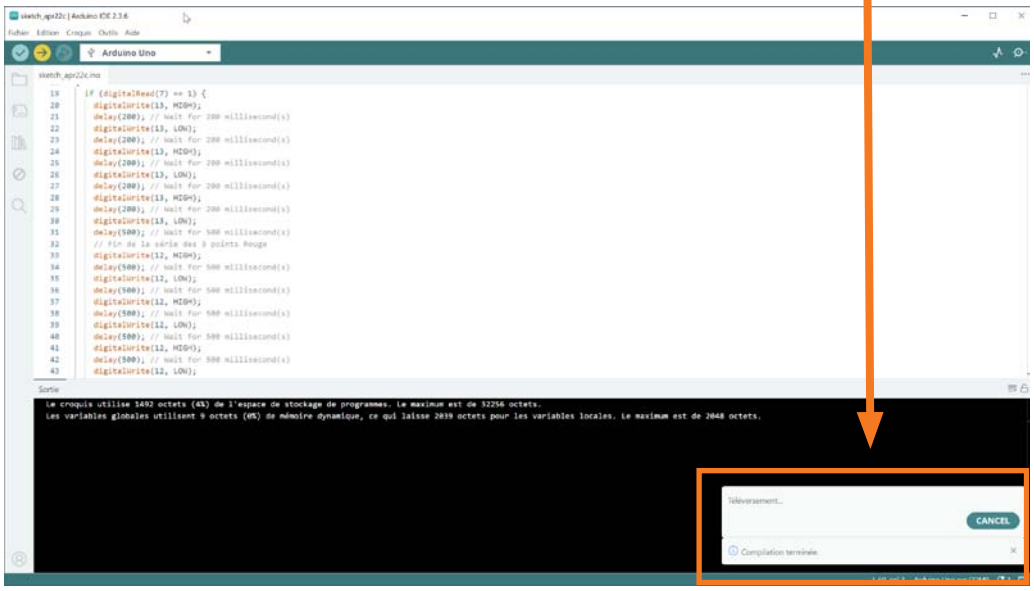
- Nous pouvons cliquer sur  pour vérifier qu'il n'y a pas d'erreurs.



- ou bien directement téléverser le programme compilé sur l'Arduino en cliquant sur 

En bas de la fenêtre de l'IDE, dans la partie noire, nous devons avoir un message qui nous indique l'espace occupé par le «croquis» (programme) ainsi qu'une fenêtre nous indiquant que la compilation et le téléversement sont finis.

Le programme va alors immédiatement s'exécuter sur l'Arduino ! ;)



A noter : il faut que l'Arduino soit branché électriquement (port USB ou Batterie USB pour fonctionner).

En cas de soucis : Réinitialiser l'Arduino avec le petit bouton «Reset».

SOS Arduino Uno ... --- ...

Troisième Programme : et un bouton ! un !

Objectif : déclencher le SOS avec un bouton.

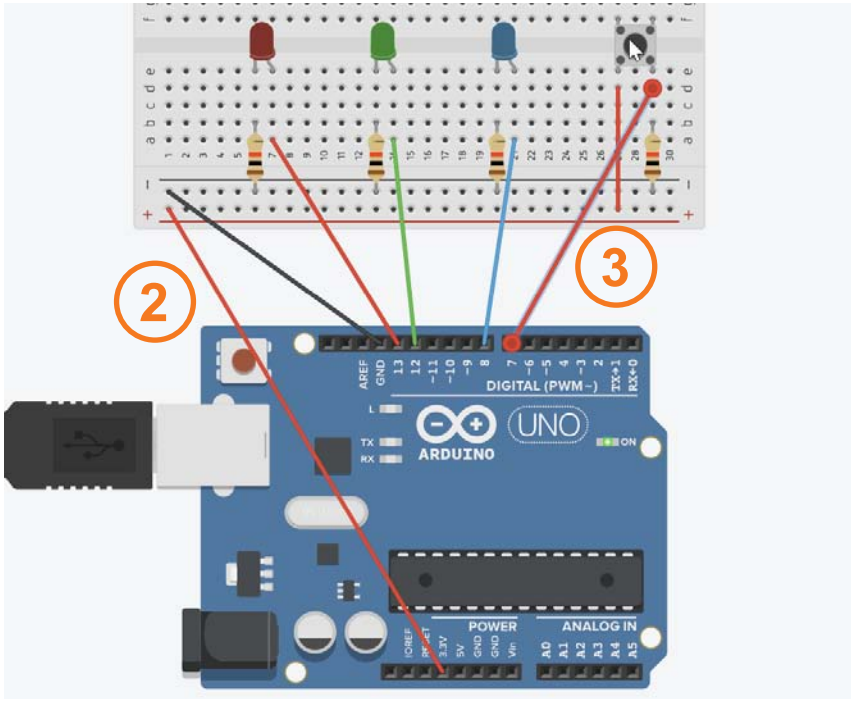
1 - Rajouter le bouton et faire le câblage adéquat :

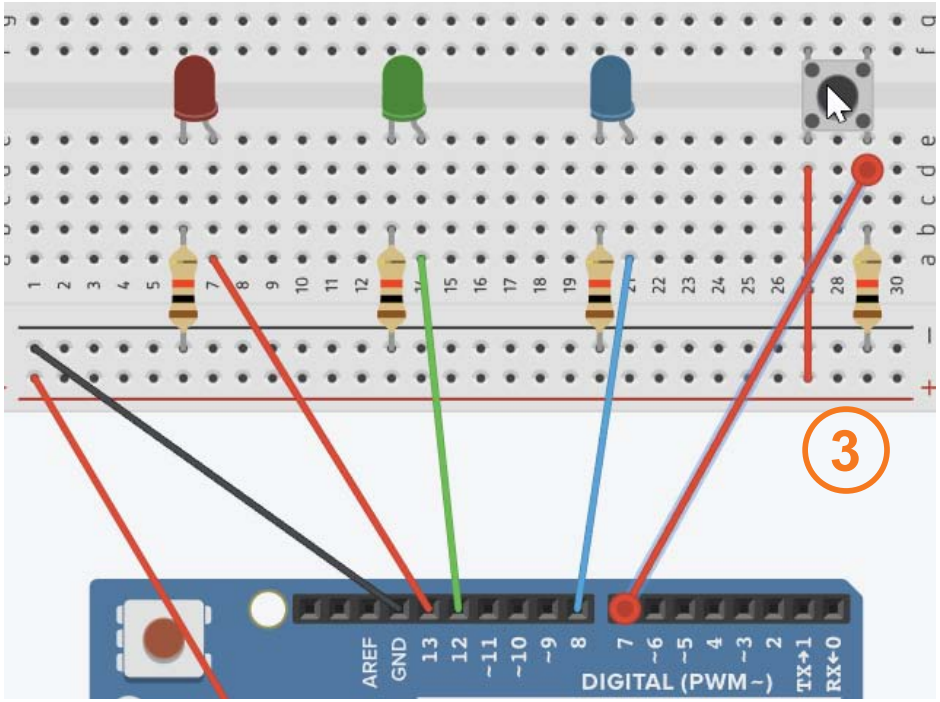
1 - Nous allons reprendre le câblage précédent, y rajouter un bouton et une résistance qui va permettre cette fois non pas d'éviter de griller un composant comme les Leds mais de maintenir un état électrique stable entre le bouton et l'Arduino (0 ou 1).

La résistance permet de mettre l'entrée à LOW quand on n'appuie pas sur le bouton poussoir : C'est une résistance de «pull-down».

3 - Nous allons en plus rajouter un câble entre le 3.3v de l'Arduino et la ligne du + de la breadboard afin d'alimenter une patte du bouton.

2 - Une des pattes du bouton (coté résistance) est câblée sur la broche 7 de l'Arduino.





SOS Arduino Uno ... --- ...

Troisième Programme : et un bouton ! un !

2 - Modifier le programme pour qu'il allume et éteigne les LEDS externes :

A - Dans la partie «Code» nous allons chercher 3 blocs qui vont nous permettre de détecter la pression du bouton et de déclencher le SOS :

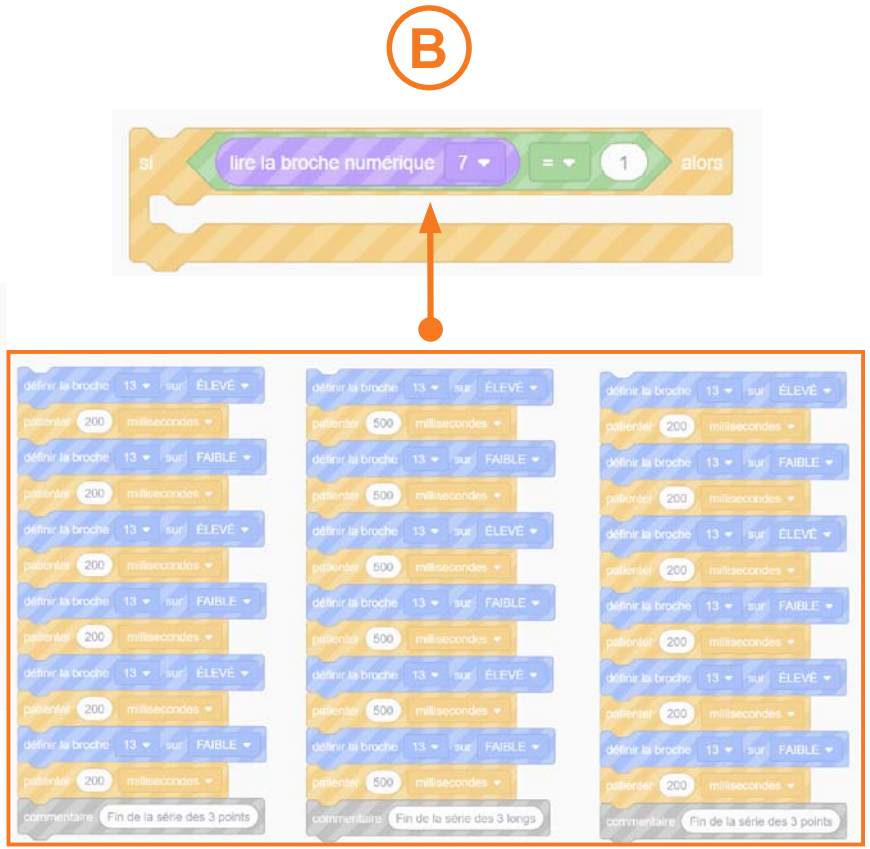
1 - Le bloc « lire la broche numérique... » de la catégorie « Entrée » permettra de savoir si le bouton a été pressé.

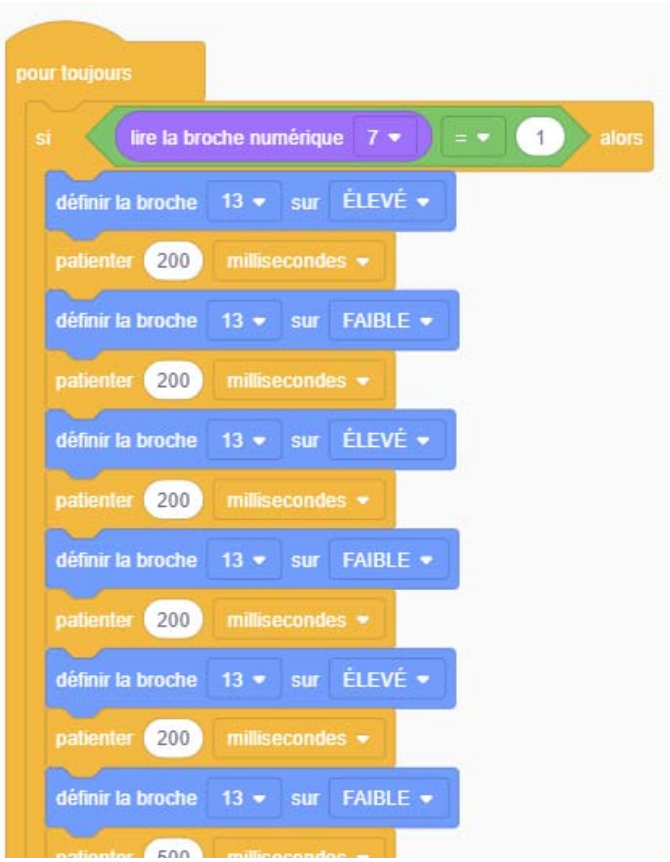
2 - Le bloc « X < = > Y » de la catégorie «Math» permettra de comparer la valeur de la broche avec 1 (bouton préssé)»

3 - Le bloc « Si ... Alors » de la catégorie «Contrôle» permettra de déclencher le SOS si le bouton a été pressé.

B - Nous allons assembler les 3 blocs puis y insérer la série de blocs du SOS.

C - Pour parvenir à ce résultat



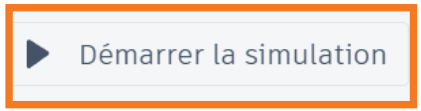


SOS Arduino Uno ... --- ...

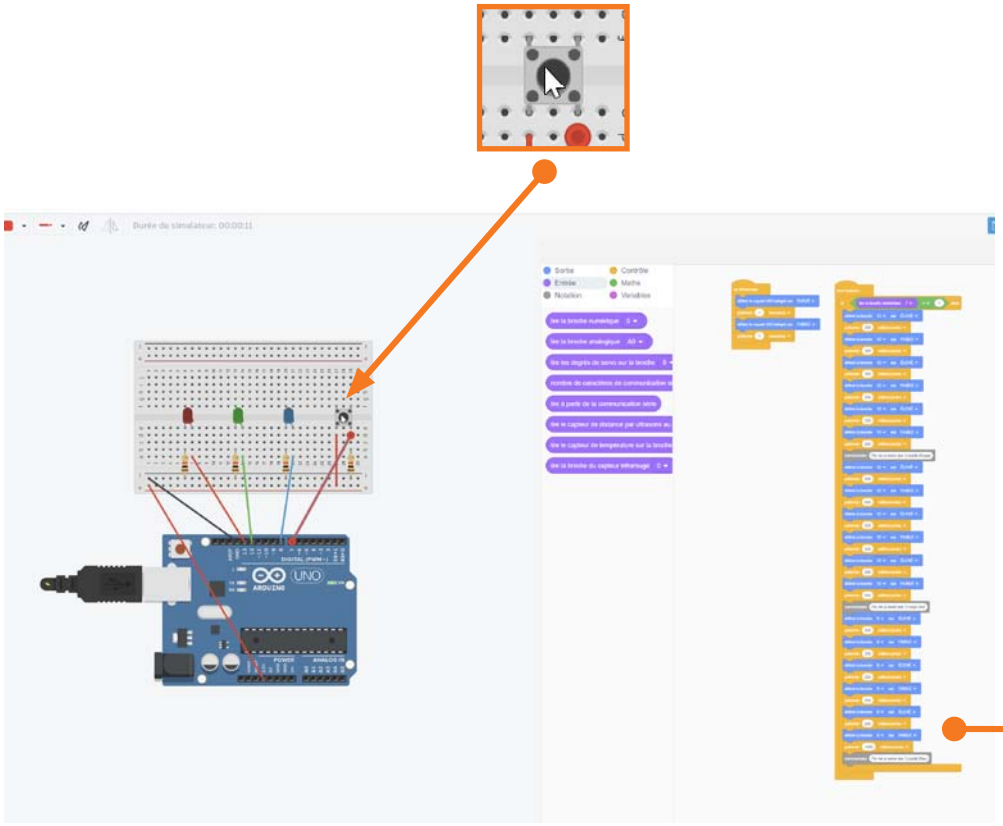
Troisième Programme : et un bouton ! un !

3 - Tester en simulation puis passer au câblage réel :

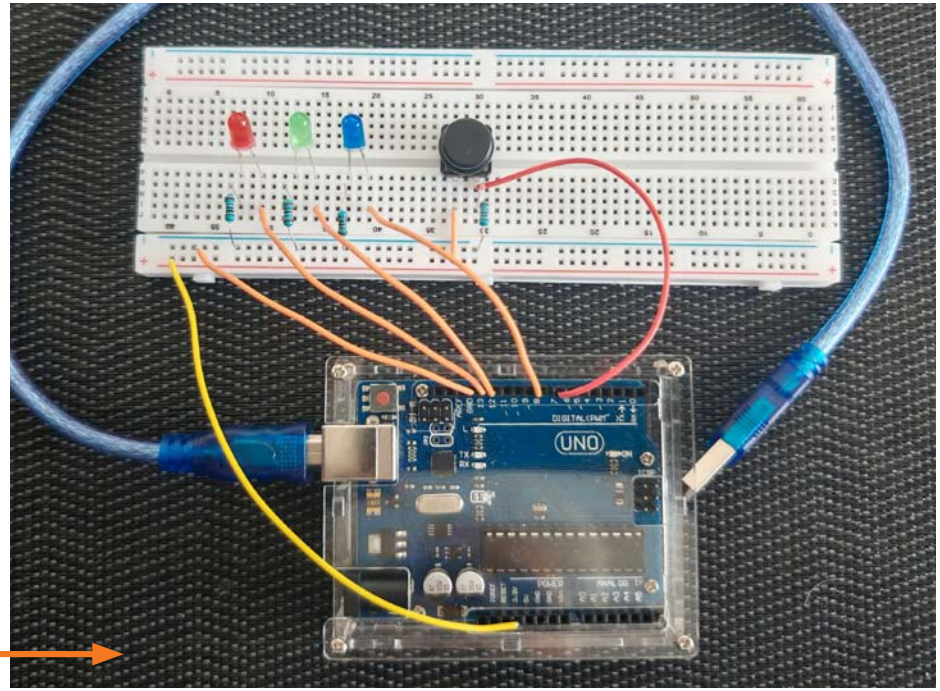
Nous pouvons désormais démarrer la simulation pour voir si tout est OK.



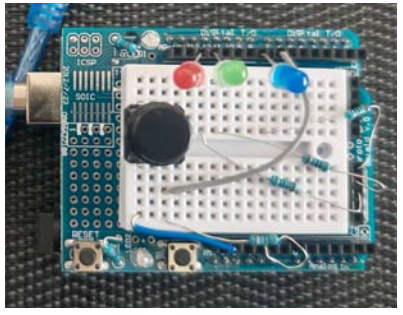
Attention : il faut cliquer sur le bouton dans la simulation pour déclencher le SOS.



Et pour finir : la câblage réel !!!



Et le même en plus petit ! ;)



SOS Arduino Uno ... --- ...

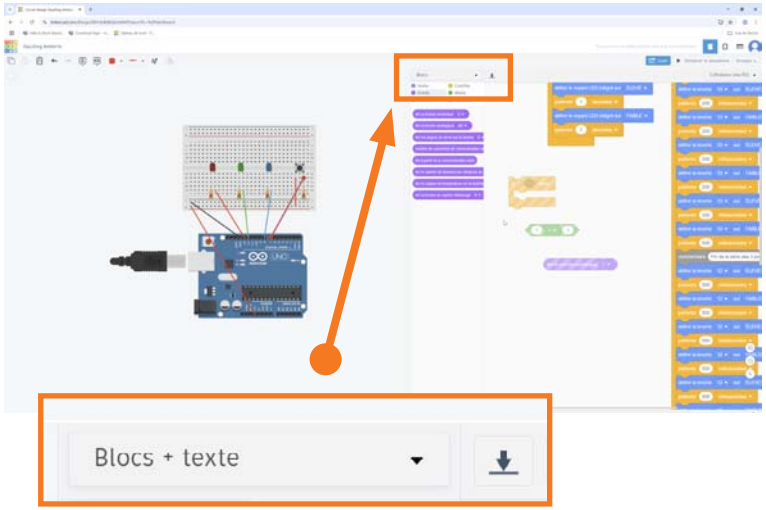
Troisième Programme : et un bouton ! un !

Si l'Arduino est reconnu dans l'IDE nous allons pouvoir récupérer le code généré dans Tinkercad et le téléverser sur l'Arduino, il y a deux méthodes :

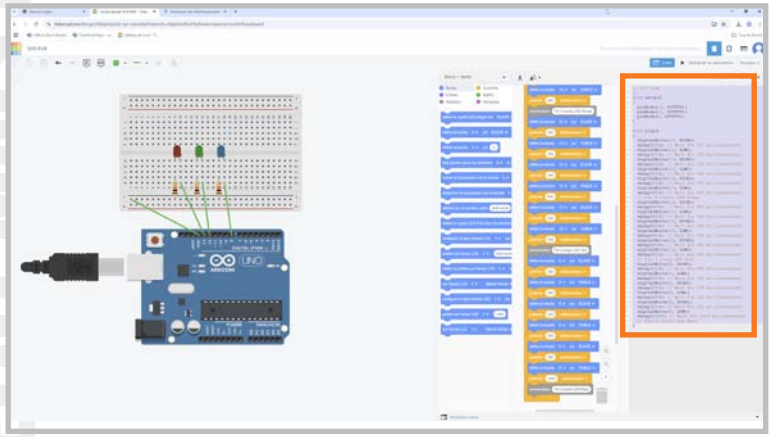
- Télécharger le code du programme à partir de Tinkercad en cliquant sur l'icône : 

Le fichier téléchargé (normalement dans le dossier téléchargement) a une extension .ino :

Nous alors ouvrir directement ce fichier .ino avec l'IDE qui va nous demander la création d'un dossier. Nous validons la création avec «OK» :

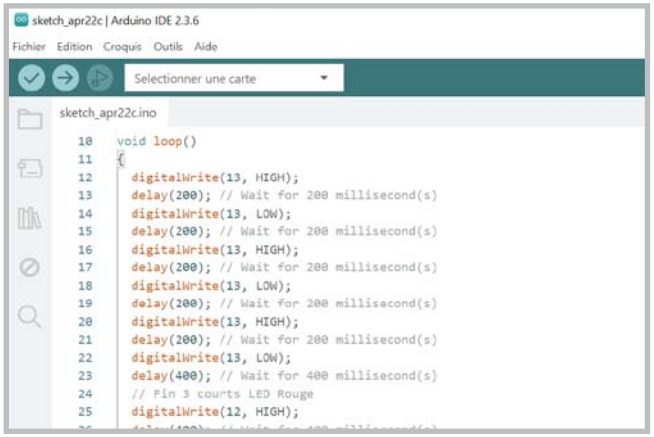


- Ou bien nous pouvons aussi directement copier-coller le code de la partie code de Tinkercad (en mode Bloc + texte) vers l'IDE :



1 Dans la partie code de Tinkercad :
Tout Sélectionner : Ctrl + A puis Copier : Ctrl+C

2 Dans l'IDE de l'Arduino :
Tout Sélectionner : Ctrl + A



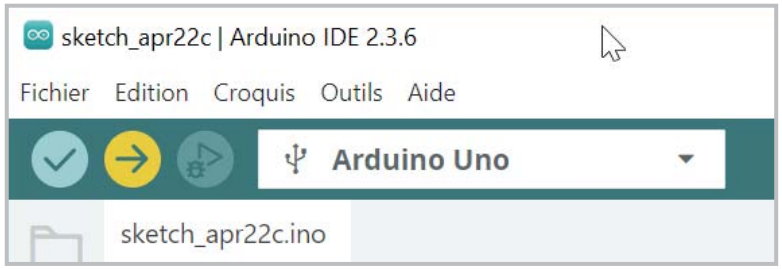
3 Dans l'IDE de l'Arduino :
puis Coller : Ctrl+V

SOS Arduino Uno ... --- ...

Troisième Programme : et un bouton ! un !

Il ne reste plus qu'à compiler (transformer le langage interprété avec des mots en 0 et 1 pour l'Arduino) puis téléverser (upload / envoyer) le programme du PC vers l'Arduino connecté :

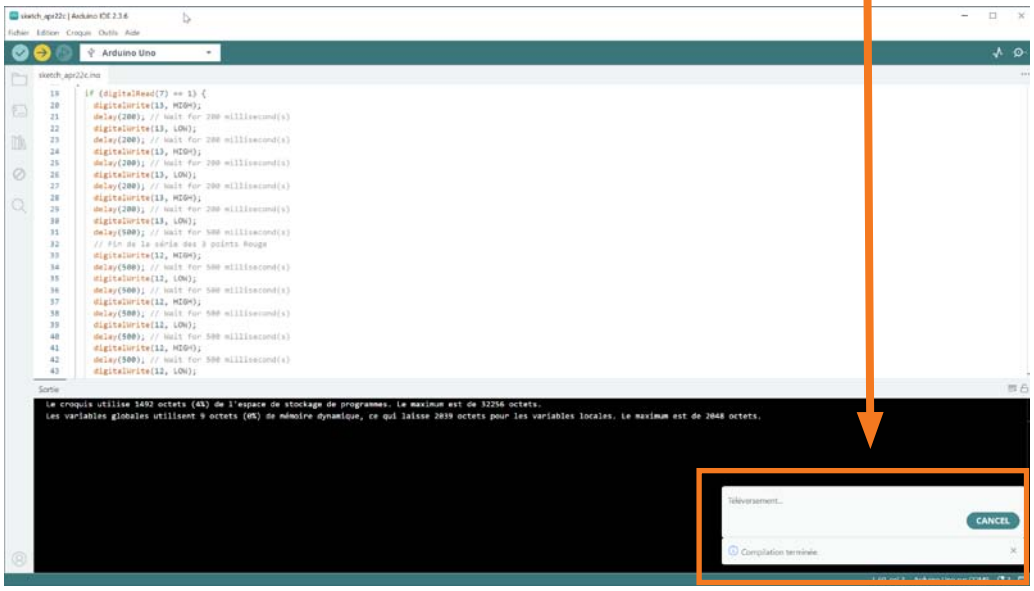
- Nous pouvons cliquer sur  pour vérifier qu'il n'y a pas d'erreurs.



- ou bien directement téléverser le programme compilé sur l'Arduino en cliquant sur 

En bas de la fenêtre de l'IDE, dans la partie noire, nous devons avoir un message qui nous indique l'espace occupé par le «croquis» (programme) ainsi qu'une fenêtre nous indiquant que la compilation et le téléversement sont finis.

Le programme va alors immédiatement s'exécuter sur l'Arduino ! ;)



A noter : il faut que l'Arduino soit branché électriquement (port USB ou Batterie USB pour fonctionner).

En cas de soucis : Réinitialiser l'Arduino avec le petit bouton «Reset».

SOS Arduino Uno ... --- ...

Support de cours téléchargeable ici :

http://laurent.gatto.free.fr/support.html

Prochaine étape :
Moteurs et capteurs !

LAURENT GATTO

Merci pour
votre participation !

N'oubliez pas de finaliser
votre auto-évaluation,
Quizz et questionnaire de
satisfaction.

Petit mais costaud, l'Arduino !!!

ARDUINO

à Bientôt !
;)