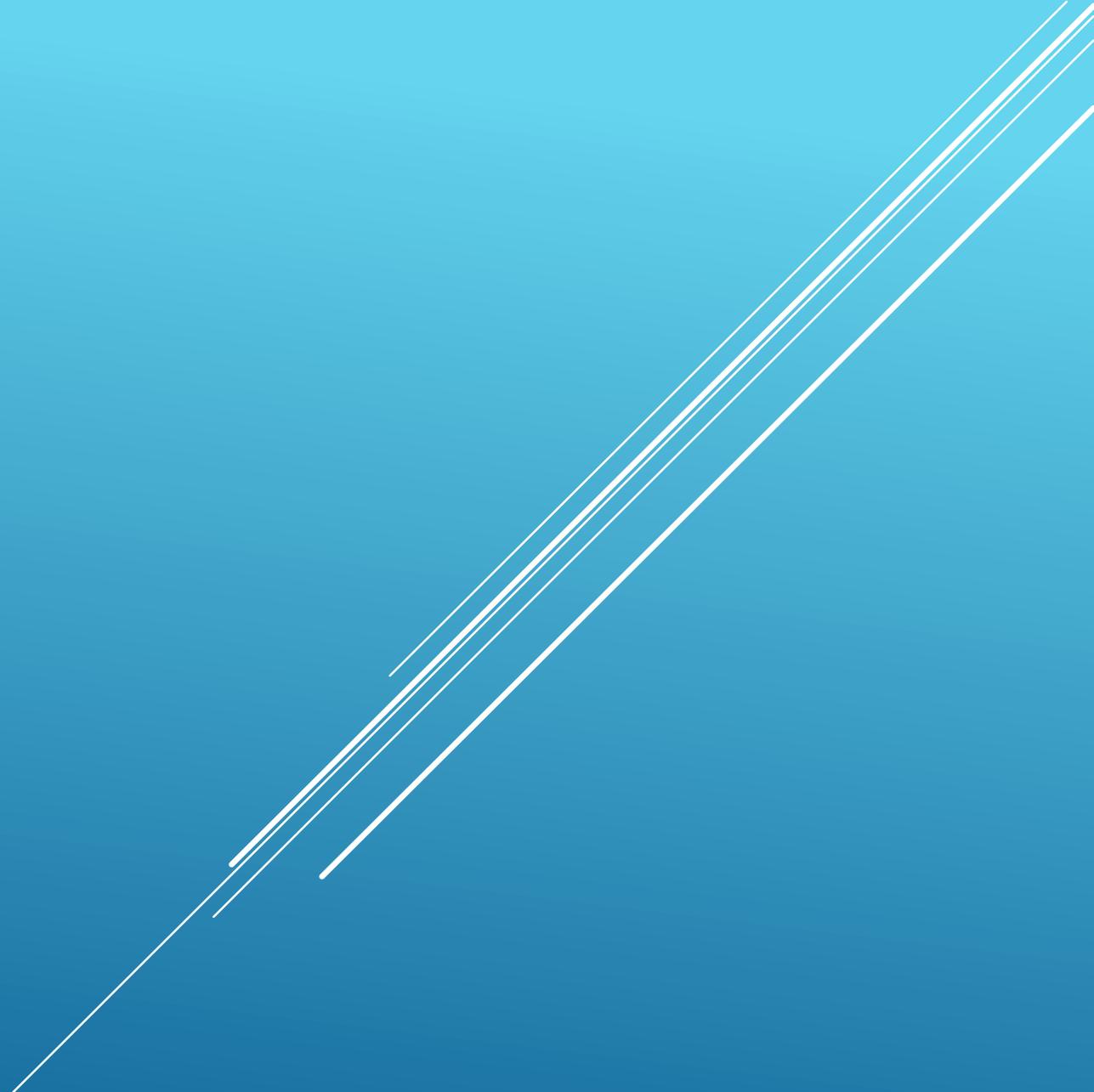
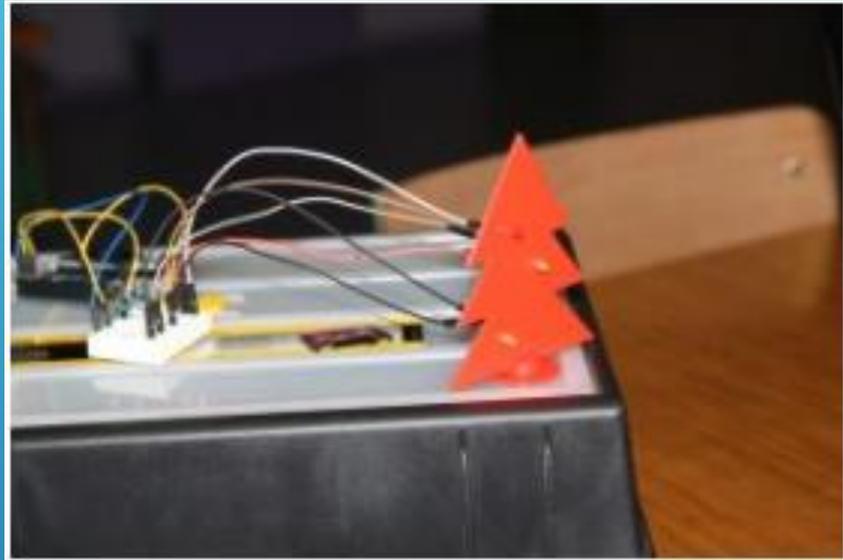


FABLAB 37110

Atelier ARDUINO – JEUNES

Séance du 25/11/2017

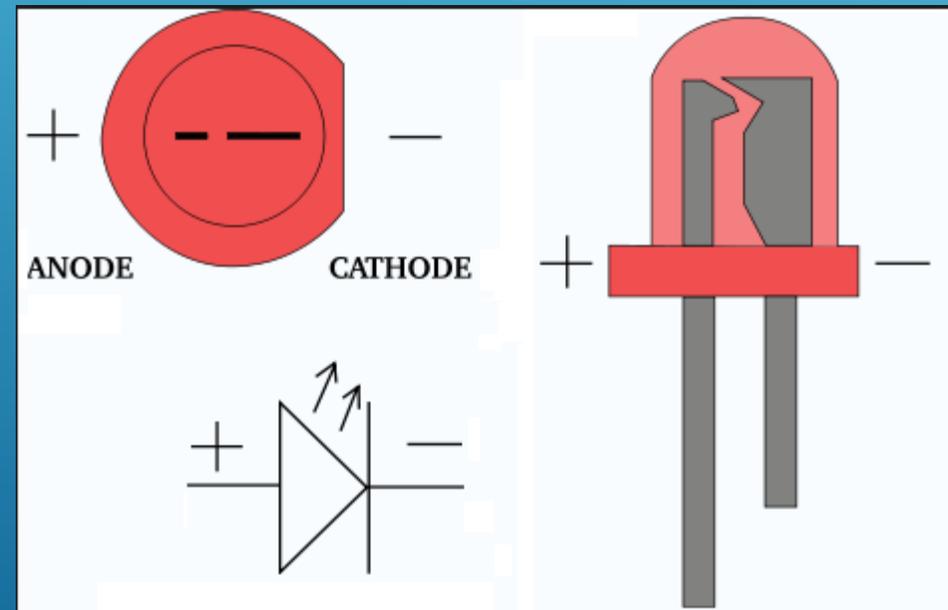
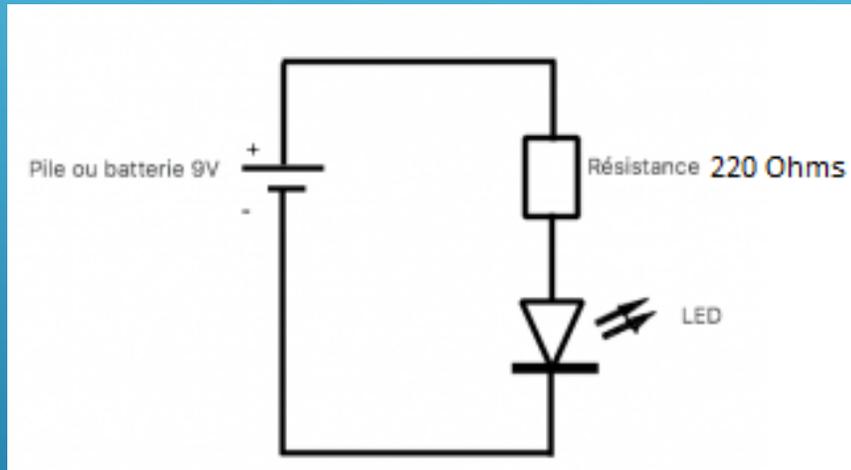


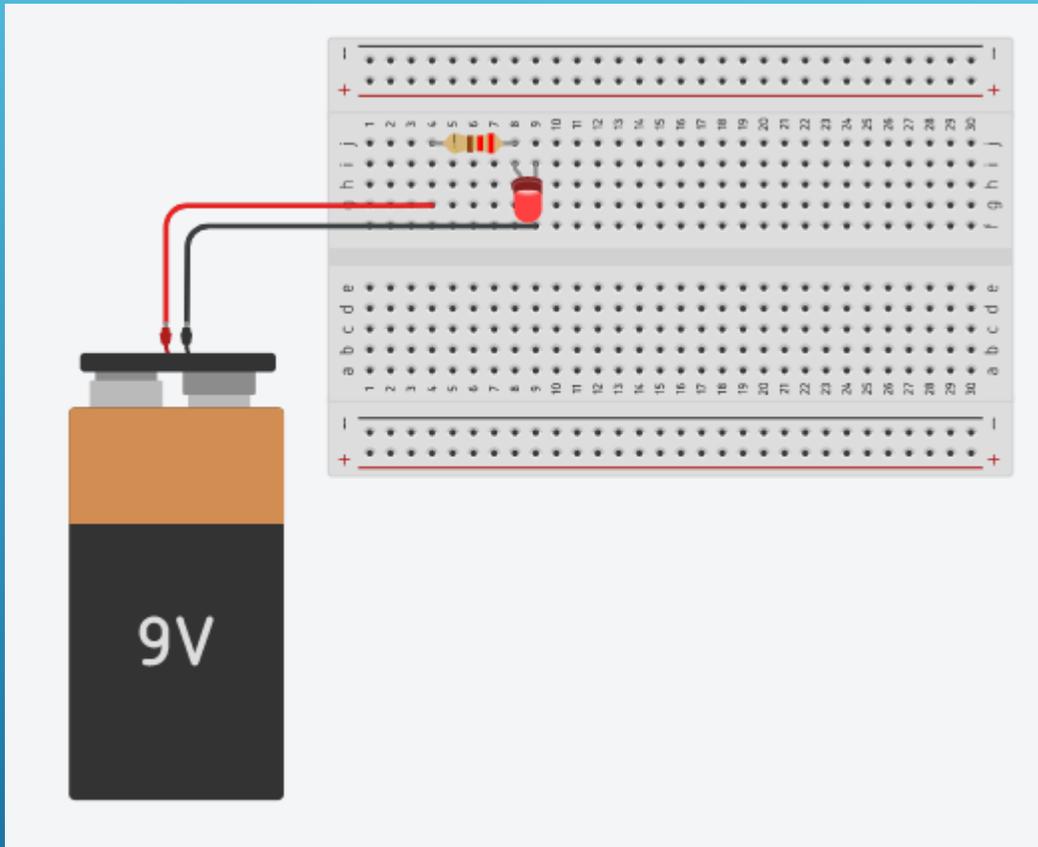


OBJECTIF : PROGRAMMER LA LED QUI
SERA MONTÉE SUR LE SAPIN DE NOËL...

ALLUMER UNE LED

Schéma de principe





ALLUMER UNE LED

Matériel :

1 « breadboard »

1 pile 9V

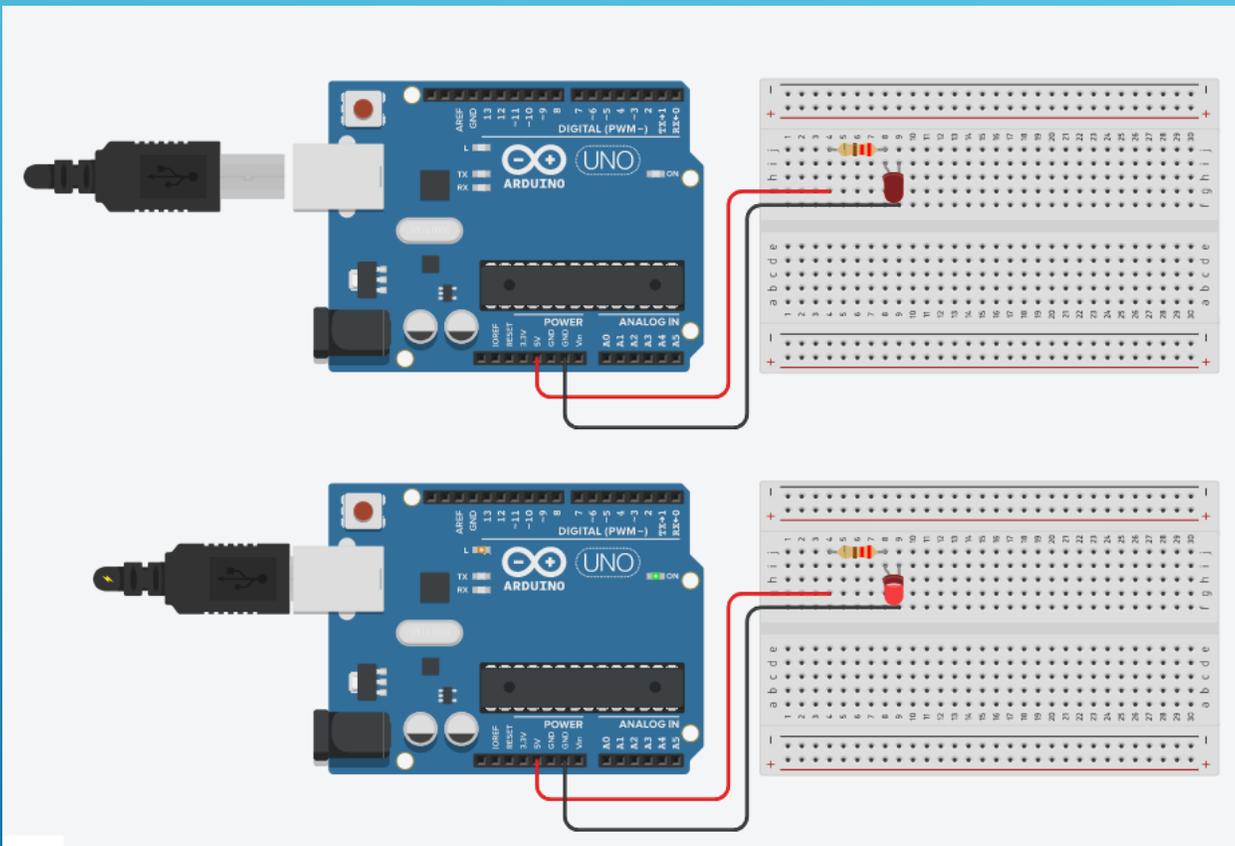
1 Resistance 220 Ohms

1 LED

2 fils

La led s'allume tant que la pile est branchée.

ATTENTION AU SENS DE BRANCHEMENT DE LA LED !



ALLUMER UNE LED

Matériel :

1 « breadboard »

1 Arduino

1 Resistance 220 Ohms

1 LED

2 fils

La led s'allume tant que l'Arduino est sous tension,

GND correspond au « - »

+5V correspond au « + »

ALLUMER UNE LED

Matériel :

1 « breadboard »

1 Arduino

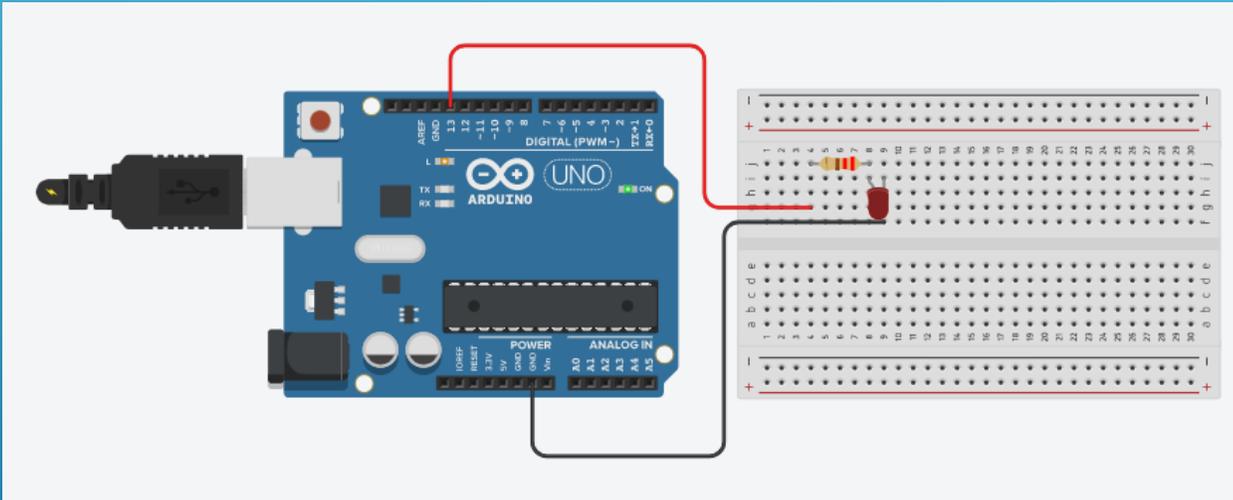
1 Resistance 220 Ohms

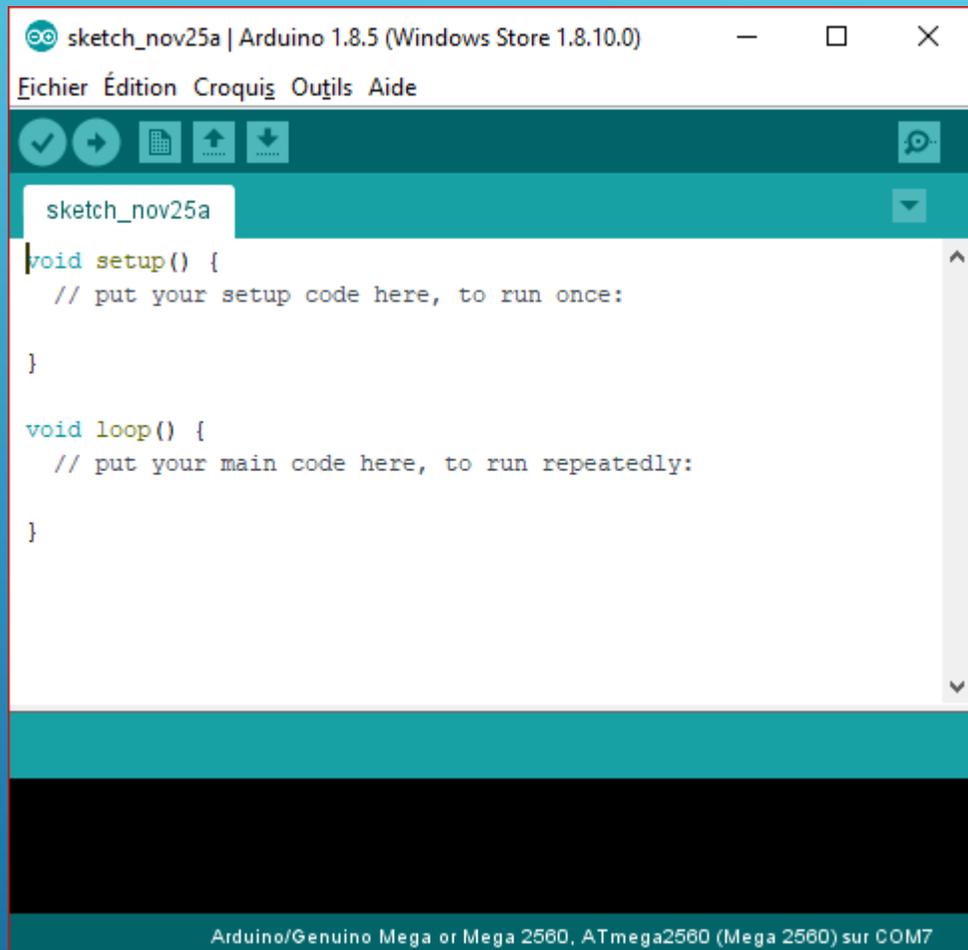
1 LED

2 fils

La led s'allume suivant le programme enregistré dans l'arduino.

Le « + » sera fournit par la borne 13.





sketch_nov25a | Arduino 1.8.5 (Windows Store 1.8.10.0)

Fichier Édition Croquis Outils Aide

```
sketch_nov25a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) sur COM7

PROGRAMMER ETAPE 1

Matériel :

...

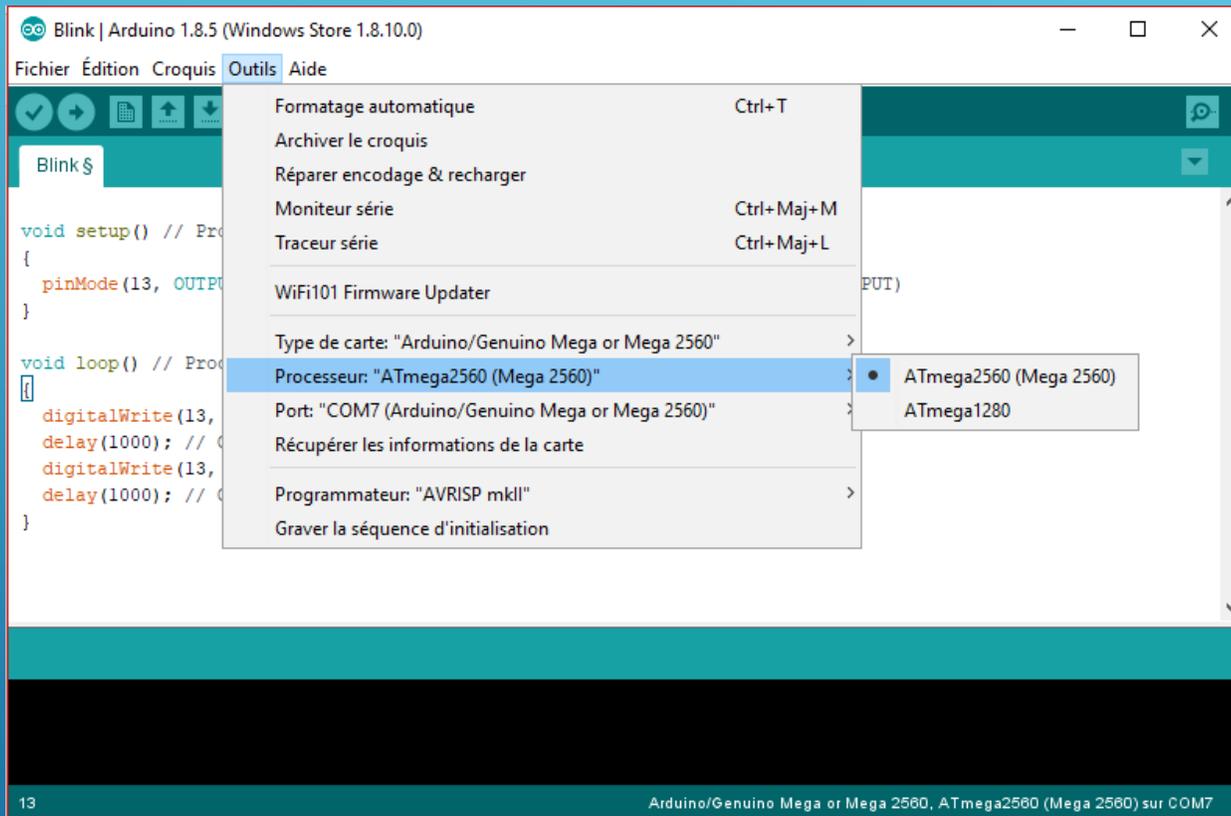
Logiciel Arduino sur ordinateur

La led s'allume suivant le programme enregistré dans l'arduino.

Branchez l'Arduino à l'aide du câble USB.

PROGRAMMER ETAPE 2

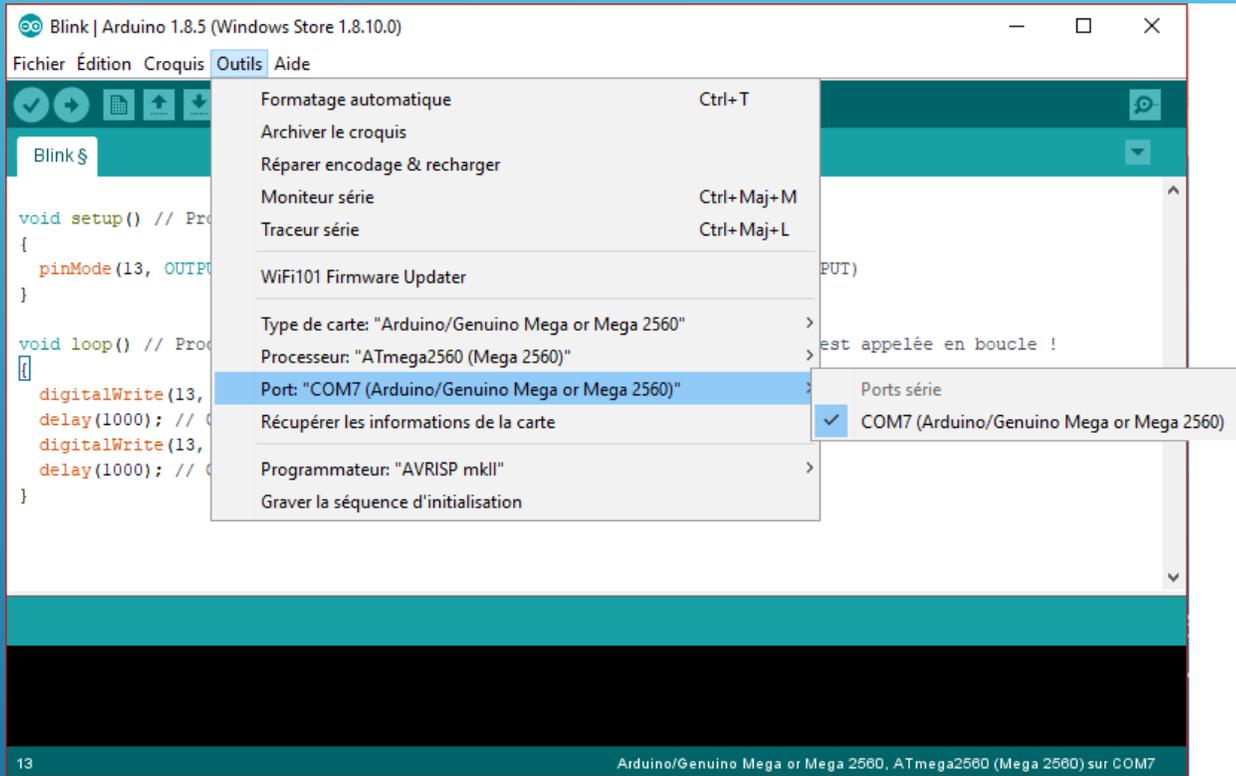
Vérifier que l'Arduino
sélectionné est le bon.

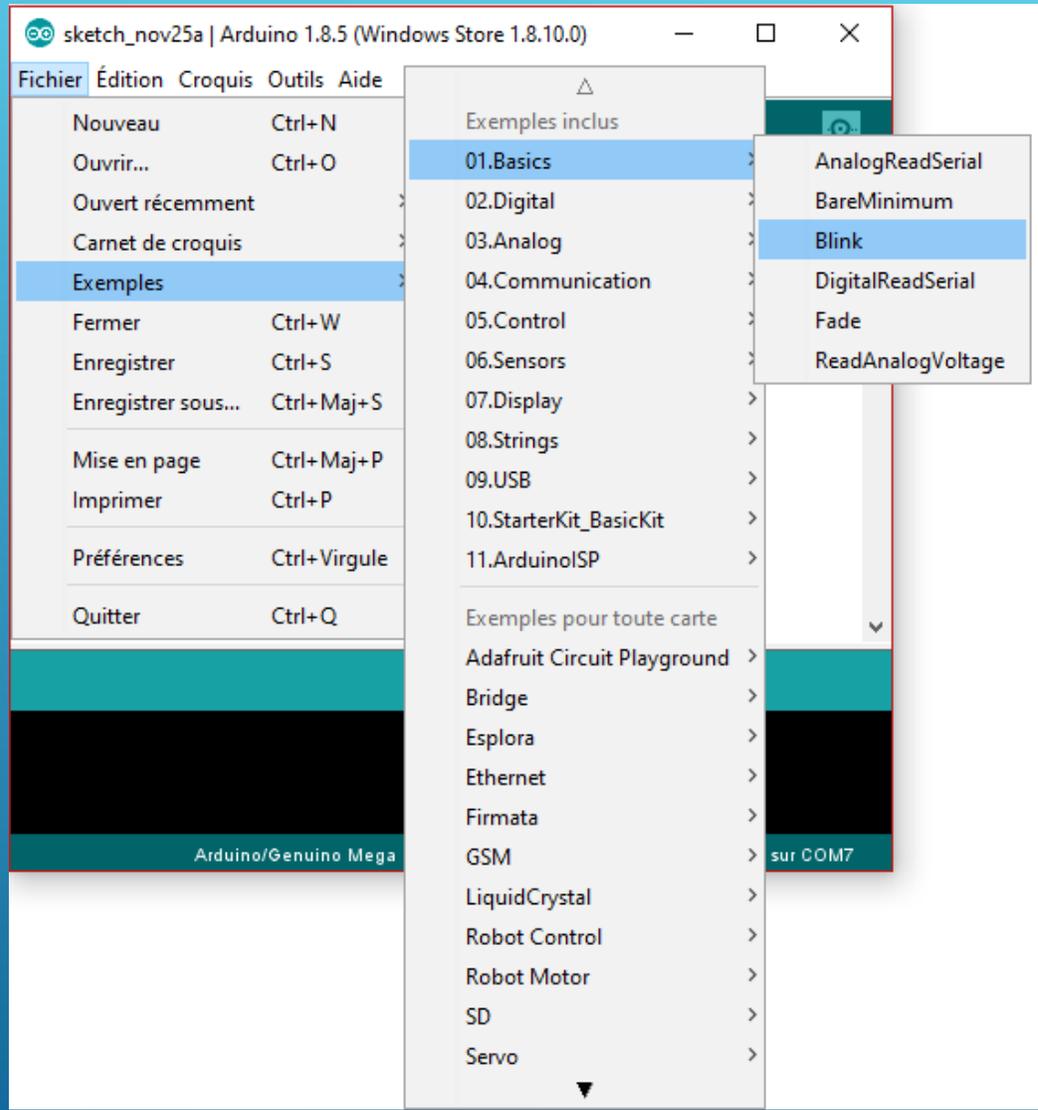


PROGRAMMER ETAPE 3

Vérifier que l'Arduino est
reconnu par Windows.

Pour cela le 'port' doit être
accessible et non vide.



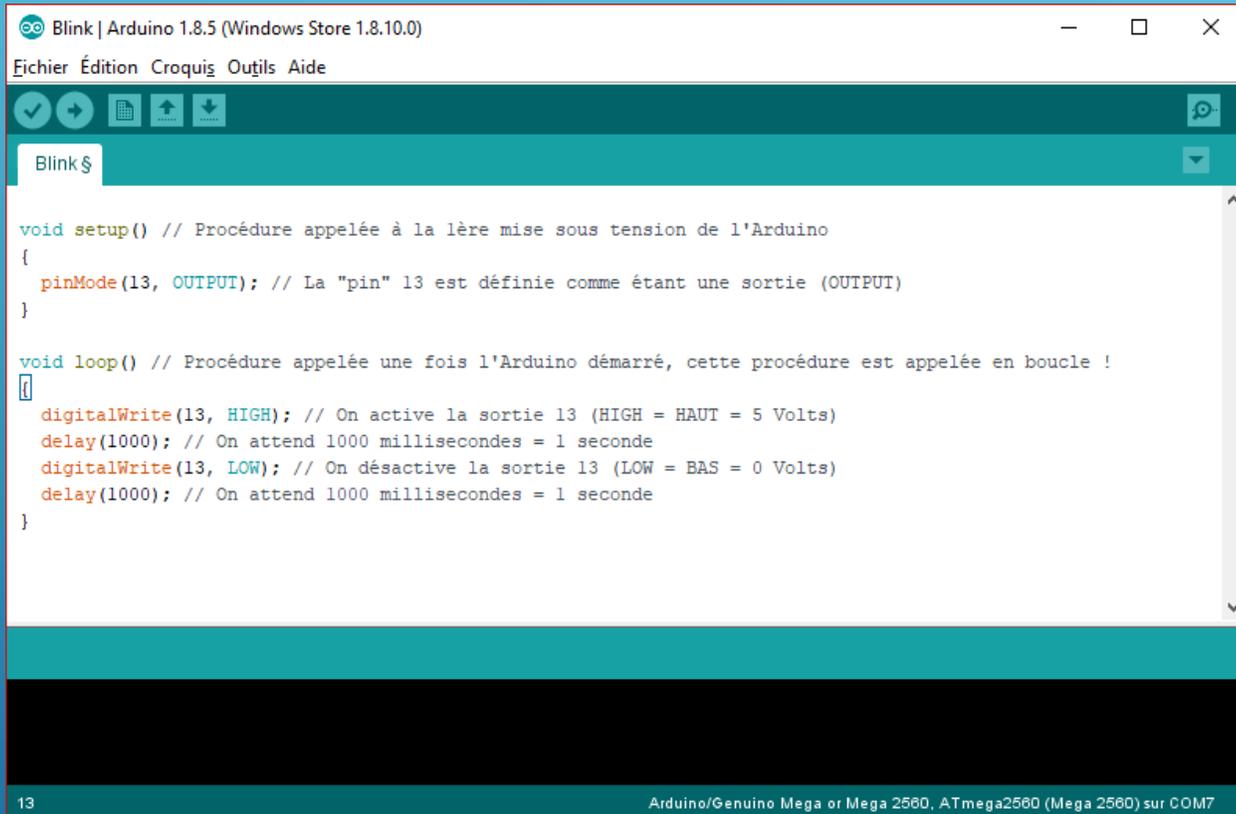


PROGRAMMER ETAPE 4

Ouvrir un nouveau programme
depuis un exemple.

PROGRAMMER ETAPE 5

Le code doit correspondre à l'exemple ci-contre.



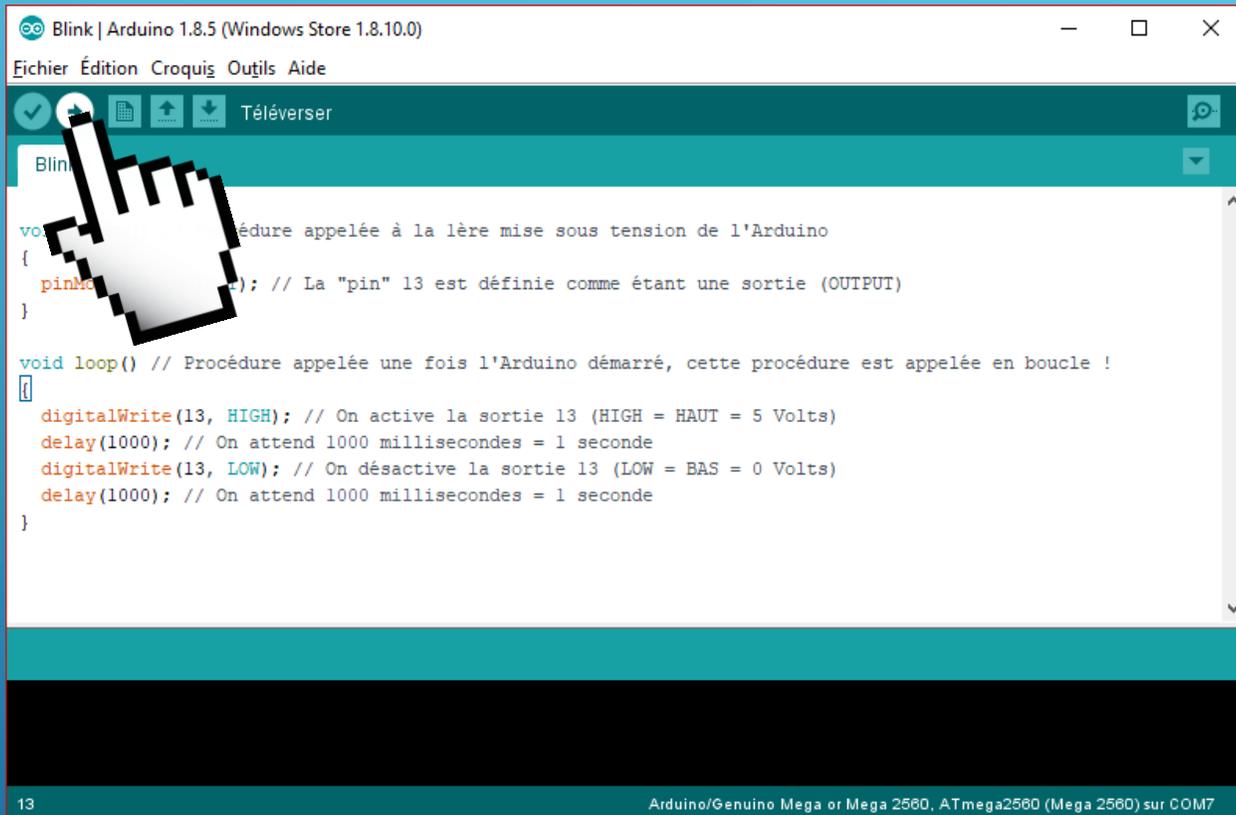
```
void setup() // Procédure appelée à la lère mise sous tension de l'Arduino
{
  pinMode(13, OUTPUT); // La "pin" 13 est définie comme étant une sortie (OUTPUT)
}

void loop() // Procédure appelée une fois l'Arduino démarré, cette procédure est appelée en boucle !
{
  digitalWrite(13, HIGH); // On active la sortie 13 (HIGH = HAUT = 5 Volts)
  delay(1000); // On attend 1000 millisecondes = 1 seconde
  digitalWrite(13, LOW); // On désactive la sortie 13 (LOW = BAS = 0 Volts)
  delay(1000); // On attend 1000 millisecondes = 1 seconde
}
```

13 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) sur COM7

PROGRAMMER ETAPE 6

Téléverser le code dans
l'Arduino.

A screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.8.5 (Windows Store 1.8.10.0)". The menu bar includes "Fichier", "Édition", "Croquis", "Outils", and "Aide". The toolbar shows a green checkmark, a plus icon, a document icon, an upload icon, a download icon, and a "Téléverser" button. A hand cursor is clicking the upload icon. The main text area contains the following code:

```
void setup() // Procédure appelée à la lère mise sous tension de l'Arduino
{
  pinMode(13, OUTPUT); // La "pin" 13 est définie comme étant une sortie (OUTPUT)
}

void loop() // Procédure appelée une fois l'Arduino démarré, cette procédure est appelée en boucle !
{
  digitalWrite(13, HIGH); // On active la sortie 13 (HIGH = HAUT = 5 Volts)
  delay(1000); // On attend 1000 millisecondes = 1 seconde
  digitalWrite(13, LOW); // On désactive la sortie 13 (LOW = BAS = 0 Volts)
  delay(1000); // On attend 1000 millisecondes = 1 seconde
}
```

The status bar at the bottom shows "13" on the left and "Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) sur COM7" on the right.

PROGRAMMER ETAPE 7

Pas d'erreurs ?

```
Téléversement terminé
Le croquis utilise 1462 octets de l'espace de stockage de programmes. Le maximum est de 253952 octets.
Les variables globales utilisent 0 octets (0%) de mémoire dynamique, ce qui laisse 8183 octets pour les variables locales.

13 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) sur COM7
```



```
expected ';' before ')' token
exit status 1
expected ';' before ')' token

13 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) sur COM7
```

Recopier les messages d'erreur

digitalWrite()

DESCRIPTION

Met un niveau logique HIGH (HAUT en anglais) ou LOW (BAS en anglais) sur une broche numérique. Si la broche a été configurée en SORTIE avec l'instruction pinMode(), sa tension est mise à la valeur correspondante : 5V (ou 3.3V sur les cartes Arduino 3.3V) pour le niveau HAUT, 0V (masse) pour le niveau BAS.

SYNTAXE

```
digitalWrite(broche, valeur)
```

[\[Get Code\]](#)

PARAMÈTRES

- broche: le numéro de la broche de la carte Arduino
- valeur : HIGH ou LOW (ou bien 1 ou 0)

VALEUR RETOURNÉE

Aucune

POUR COMPRENDRE...

delay(ms)

DESCRIPTION

Réalise une pause dans l'exécution du programme pour la durée (en millisecondes) indiquée en paramètre. (=Pour mémoire, il y a 1000 millisecondes dans une seconde...!)

SYNTAXE

```
delay (ms);
```

[\[Get Code\]](#)

PARAMÈTRES

- ms (unsigned long): le nombre de millisecondes que dure la pause

VALEUR RENVOYÉE

Aucune

POUR COMPRENDRE...