 synthèse additive des couleurs avec ARDUINO

|  |
| --- |
| C1 S’approprier |
|  |

**Nommer** les trois couleurs primaires.

Le rouge, le vert et le bleu en synthèse additive.

**Problématique : Comment obtenir d’autres couleurs à partir des 3 couleurs primaires ?**

|  |
| --- |
| C2 Raisonner |
|  |
| C5 Communiquer |
|  |

**Proposer** un protocole expérimental permettant de répondre à la problématique :

L’activation simultanée de 2 ou 3 diodes produit d’autres couleurs.

On pourra également faire varier l’intensité de 3 diodes pour faire apparaître de nouvelles couleurs.

**Partie 1**

**Synthèse de couleurs à l’aide d’une DEL**

1. **Liste du matériel**

* Une carte Arduino UNO (et son câble USB),
* Une DEL (Diode Electro-Luminescente) RVB (Rouge Vert Bleu)à cathode commune,
* 3 résistances identiques possédant le code couleur suivant : rouge, rouge, marron,
* Une plaque d'essai,
* 4 fils pour le câblage du montage.

1. **Valeur des résistances**

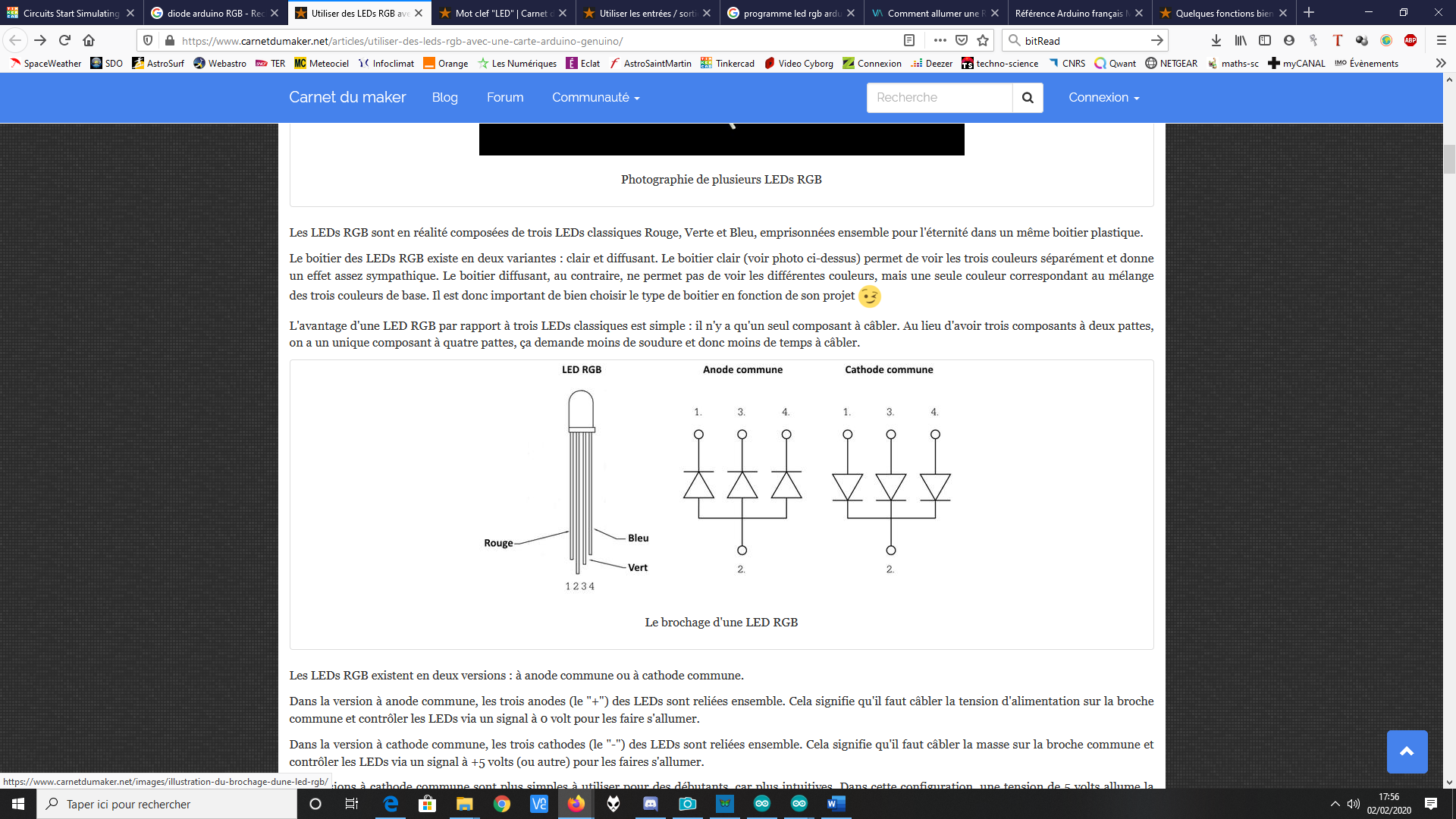
En utilisant le code couleur fourni en annexe, déterminer la valeur des résistances (identiques) :

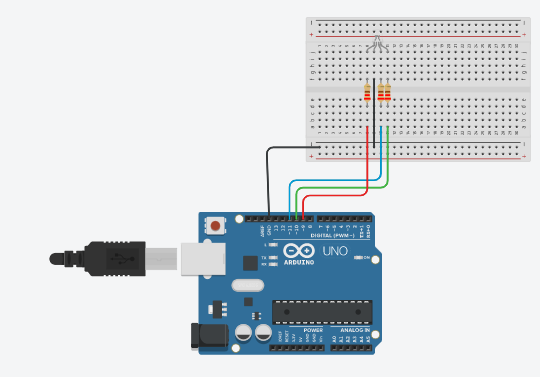
|  |
| --- |
| C1 S’approprier |
|  |

Résistance : rouge=2 rouge=2 marron=10

Donc R = 22 x 10 = 220 Ω

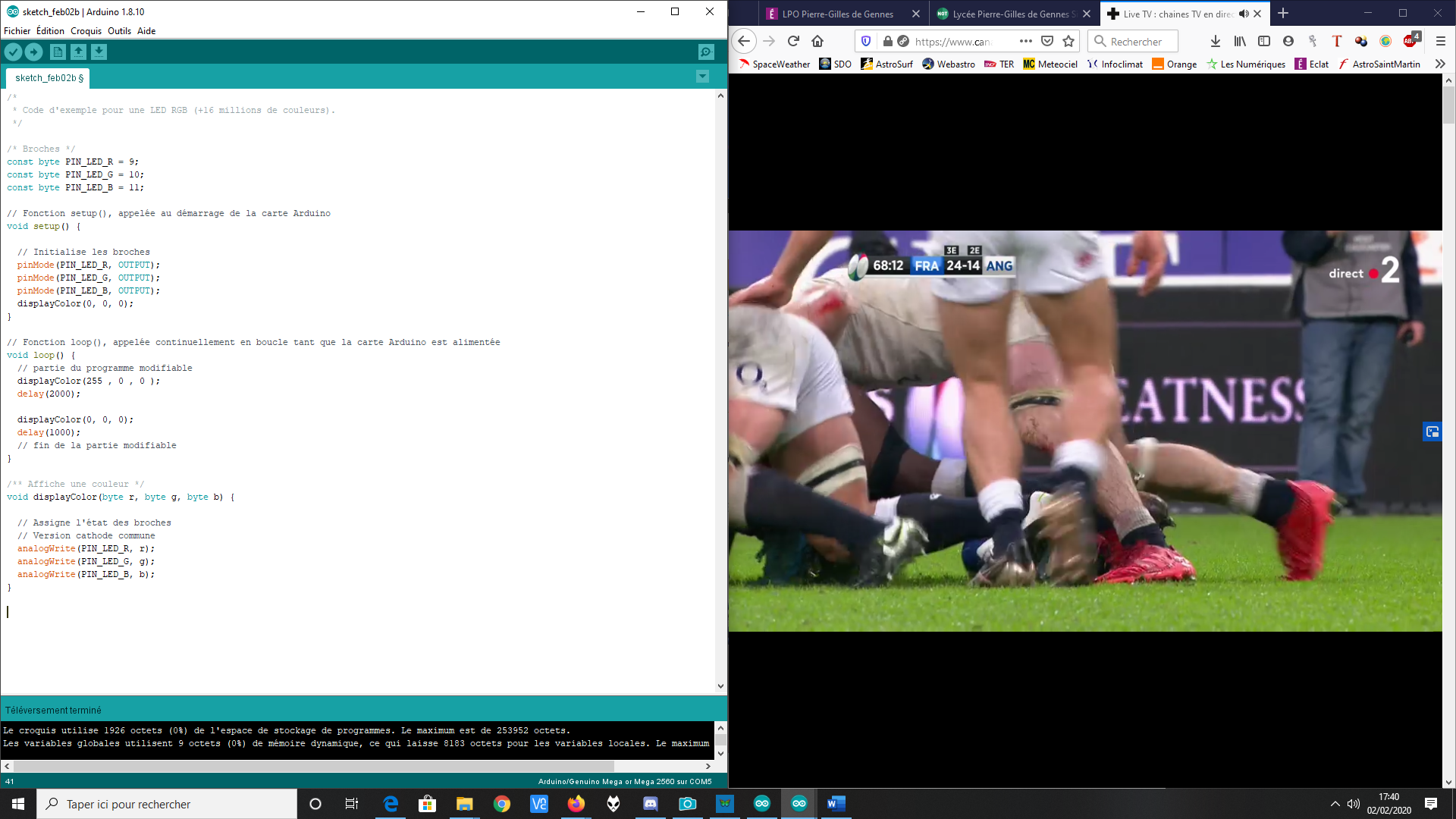
1. **Partie expérimentale**

* **Réaliser** le circuit schématisé ci-dessous. Quelques précisions :
* La plus longue broche de la DEL (n°2) est reliée à la terre (GND) sur Arduino.
* La broche n°1 est reliée à une résistance elle-même reliée à la borne 9 de l’Arduino.
* La broche n°3 est reliée à une résistance, elle-même reliée à la borne 10.
* La broche n°4 est reliée à une résistance, elle-même reliée à la borne 11.



|  |  |
| --- | --- |
| C3 Réaliser | |
| Résultat de recherche d'images pour "appel CCF" | Appel |
|  | |

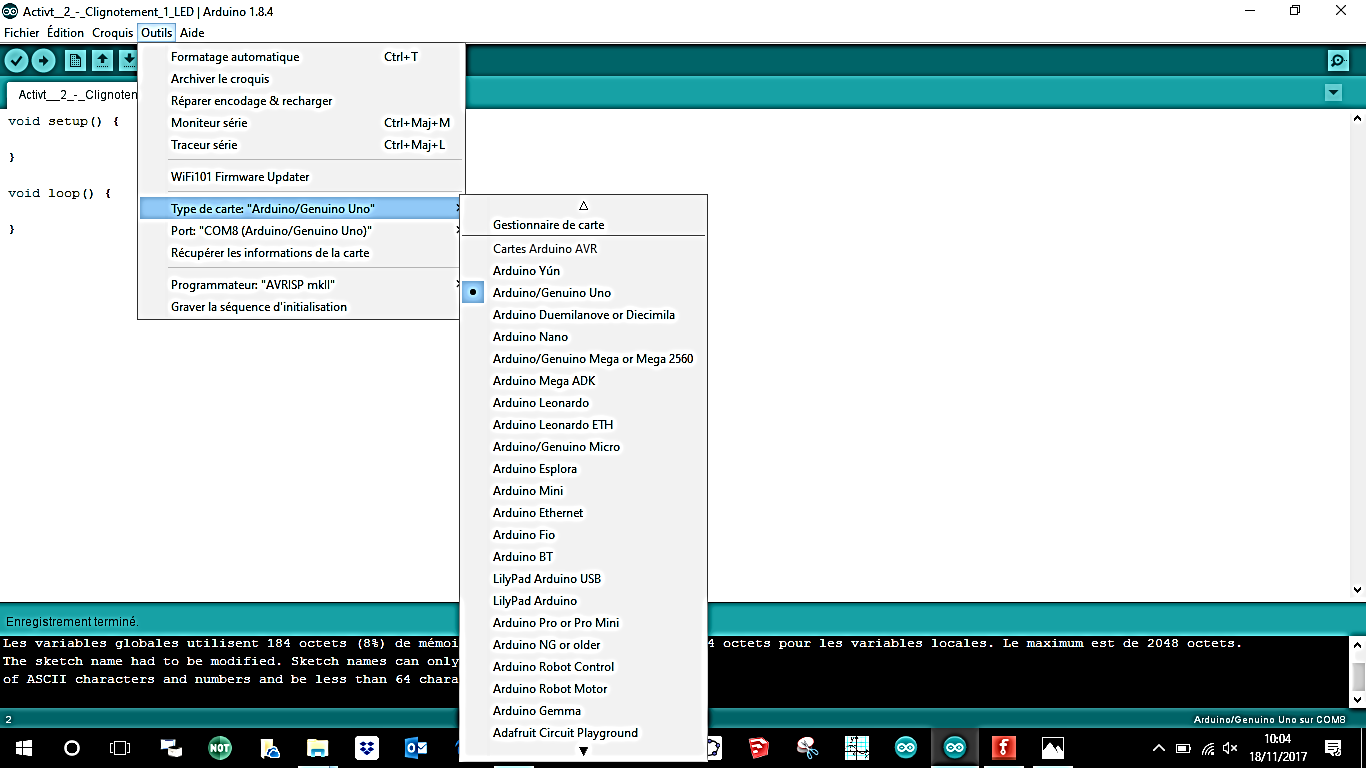
* **Faire vérifier** le montage au professeur**.**



1. **Programmation**

Le réglage de la DEL RVB s’effectue à partir d’un microcontrôleur Arduino. Le programme permettant de contrôler la DEL est présenté ci-contre. Il est accessible dans le logiciel « IDE ».

* **Brancher** la carte Arduino au PC. Dans le menu « Outil », **choisir** la carte « Arduino/Genuino Uno » et le port adapté (« COM 8… »).



|  |
| --- |
| C3 Réaliser |
|  |

* **Téléverser** le programme à l’aide de l’icône :

****

|  |
| --- |
| C1 S’approprier |
|  |

1. **Résultats**

La diode apparaît rouge durant 2 s, soit 2 000 ms, puis s’éteint pendant 1 s, soit 1 000 ms.

1. **Obtention de couleurs différentes**

* **Modifier** le programme (uniquement la zone ci-dessous) de façon à ce que la diode s’allume avec la couleur verte, puis **compléter** la première ligne du tableau ci-après.

// partie du programme modifiable

|  |  |
| --- | --- |
| C3 Réaliser | |
| Résultat de recherche d'images pour "appel CCF" | Appel professeur lorsque la diode s’allume avec la couleur verte. |
|  | |

displayColor(255 , 0 , 0 );

delay(2000);

displayColor(0, 0, 0);

delay(1000);

// fin de la partie modifiable

* **Modifier** le programme de façon à ce que la diode s’allume avec chacune des couleurs indiquées dans le tableau ci-dessous et **renseigner** les modifications apportées.

|  |  |
| --- | --- |
| **Couleur obtenue** | **Modification programme** |
| **Verte** | displayColor( 0 , 255 , 0 ); |
| **Bleue** | displayColor( 0 , 0 , 255 ); |
| **Jaune** | displayColor( 255 , 255 , 0 ); |
| **Magenta** | displayColor( 255 , 0 , 255 ); |
| **Cyan** | displayColor( 0 , 255 , 255 ); |
| **Blanche** | displayColor(255 , 255 , 255 ); |

|  |
| --- |
| C5 Communiquer |
|  |

1. **Conclusion**

En jouant sur le ton, la saturation et la luminosité, on obtient 256 nuances différentes pour chaque couleur primaire (variant entre 0 et 255).

A partir des trois couleurs primaires, on peut donc obtenir 256 x 256 x 256 = 2563 soit 16 777 216 couleurs.

1. **Prolongement de l’activité**

* **Modifier** le programme en remplaçant toutes les lignes de la zone modifiable par :

for (int i=0; i <= 255; i++){

displayColor(0 , i , i );

delay(30);

}

|  |
| --- |
| C2 Raisonner |
|  |
| C5 Communiquer |
|  |

* **Indiquer** vos observations.

La diode prend la couleur cyan, et sa luminosité augmente progressivement.

**Partie 2**

**Synthèse de couleurs à l’aide de l’application « TINKERCAD »**

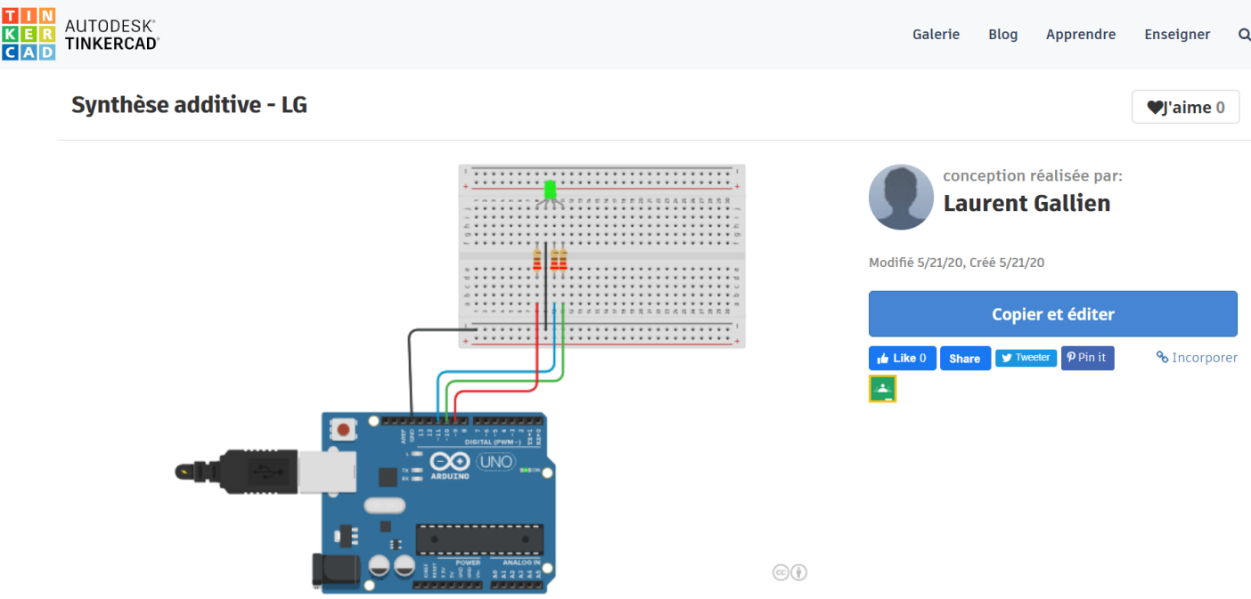
* 1. Découverte de l’application « TINKERCAD »

**TINKERCAD** est un logiciel (gratuit) permettant de reproduire le montage de la partie précédente et de tester le programme afin d’obtenir les différentes couleurs du tableau (page 4).

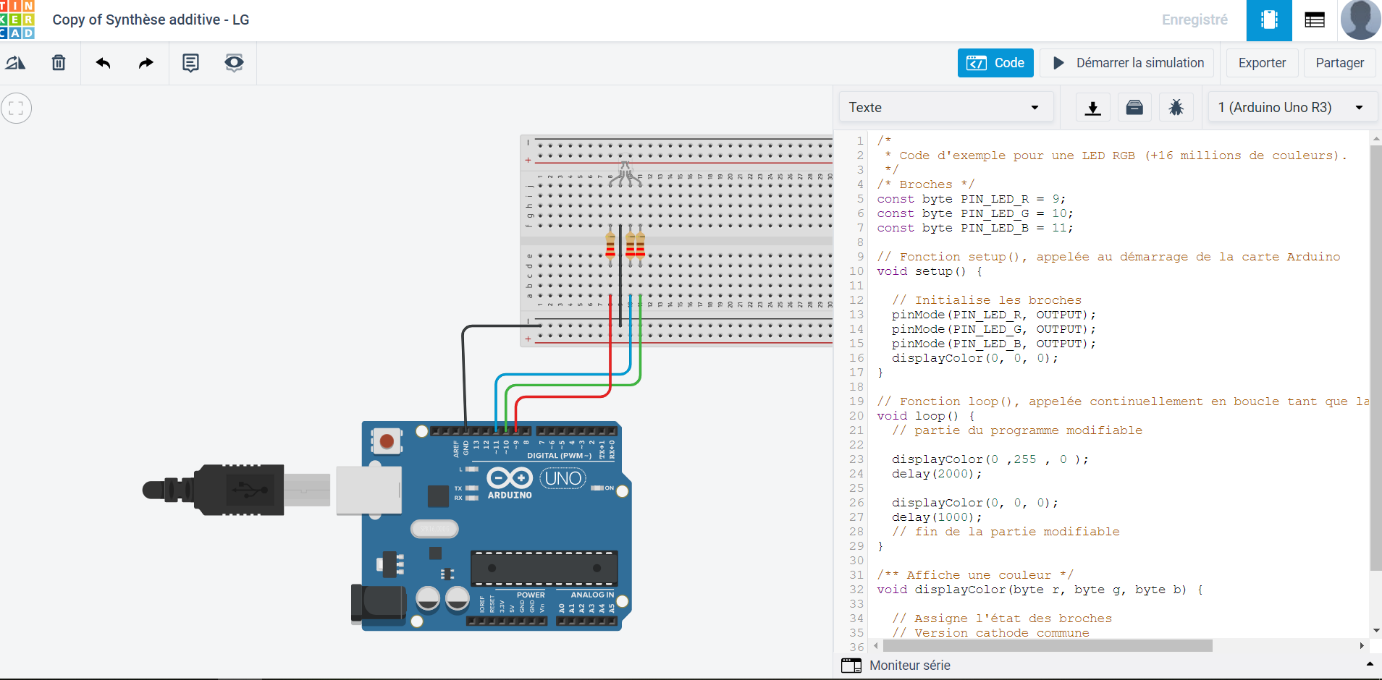
***Remarque : une inscription sur TINKERCAD est requise afin de pouvoir effectuer des modifications sur le montage et/ou sur la programmation.***

**Cliquer** sur le lien suivant :

<https://www.tinkercad.com/things/0B3MHOaE0a6>

La page suivante s’affiche :

En cliquant sur « copier et éditer », puis sur « code », le programme apparait sur la partie droite de votre écran :



* 1. Utilisation de l’application « TINKERCAD »
* **Modifier** le programme (partie du programme modifiable), de façon à ce que la diode s’allume avec chacune des couleurs indiquées dans le tableau ci-dessous :

|  |  |
| --- | --- |
| **Couleur obtenue** | **Modification programme** |
| **Verte** | displayColor( 0 , 255 , 0 ); |
| **Bleue** | displayColor( 0 , 0 , 255 ); |
| **Jaune** | displayColor( 255 , 255 , 0 ); |
| **Magenta** | displayColor( 255 , 0 , 255 ); |
| **Cyan** | displayColor( 0 , 255 , 255 ); |
| **Blanche** | displayColor(255 , 255 , 255 ); |

|  |  |
| --- | --- |
| C3 Réaliser | |
| Résultat de recherche d'images pour "appel CCF" | Appel professeur lorsque le tableau est entièrement complété |
|  | |

|  |
| --- |
| C5 Communiquer |
|  |

* **Conclusion :** A partir de ces 3 couleurs primaires, on peut obtenir plus de 2563 soit 16 777 216 couleurs.
* **Modifier** le programme en remplaçant toutes les lignes de la zone modifiable par :

for (int i=0; i <= 255; i++){

displayColor(i , 0 , i );

delay(30);

}

|  |
| --- |
| C2 Raisonner |
|  |
| C5 Communiquer |
|  |

* **Indiquer** vos observations :

La diode prend la couleur magenta, et sa luminosité augmente progressivement.

* **Modifier** le programme, de façon à obtenir une couleur jaune de plus en plus foncée. **Ecrire** les modifications apportées dans l’encadré ci-dessous.

for (int i=0; i <= 255; i++){

displayColor(i , i , 0 );

delay(30);

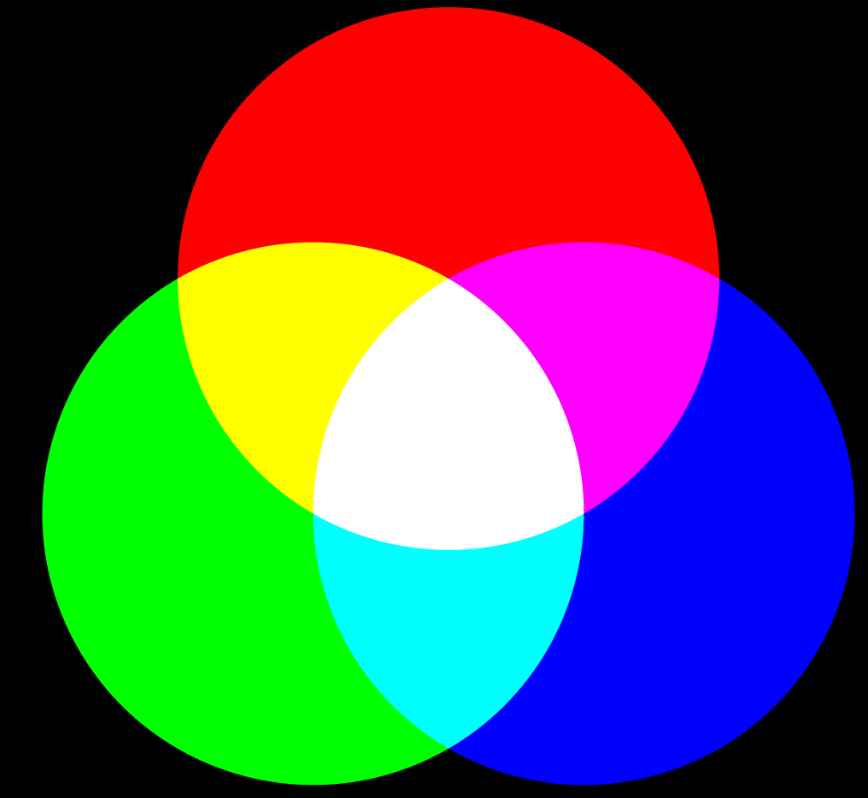
}

**La synthèse additive**

**Bilan**

* **A retenir** :

La synthèse additive des couleurs RVB permet d’obtenir d’autres couleurs en superposant/combinant des sources lumineuses colorées. L’œil réalise une synthèse additive pour percevoir une infinité de couleurs à partir des trois couleurs primaires.

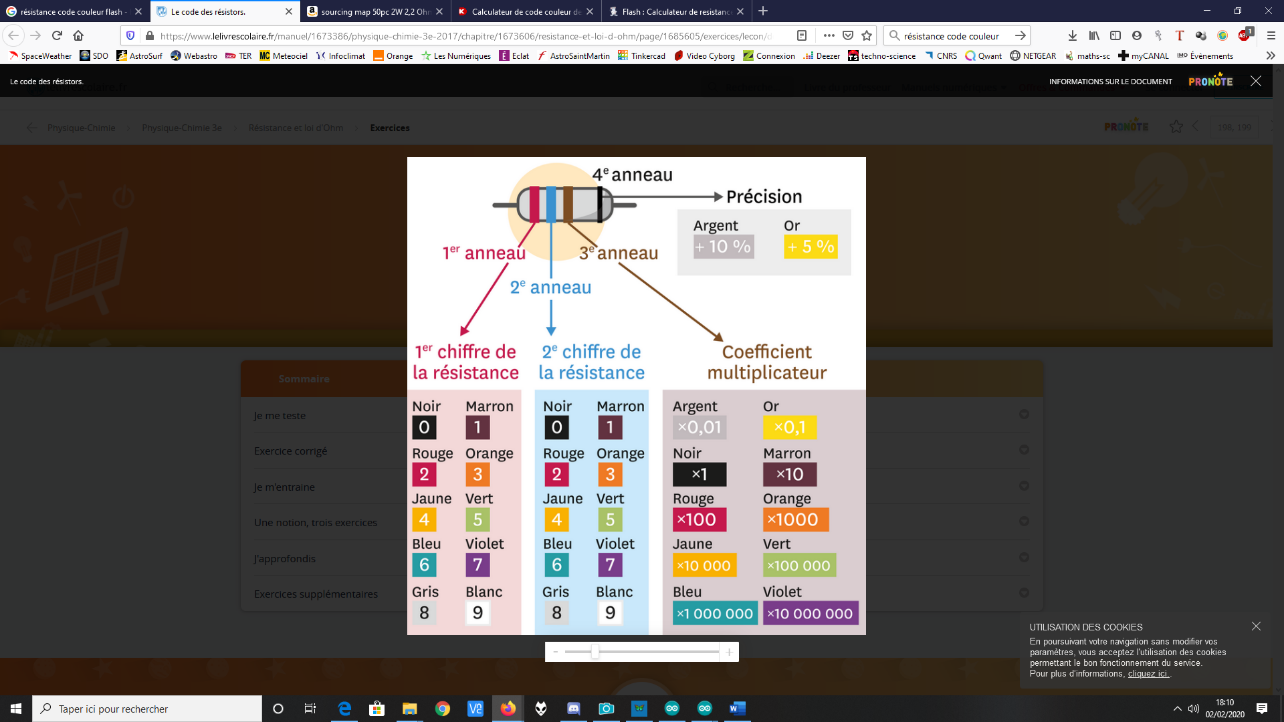
****

* **Application :** les pixels d’un écran

Les images qui s’affichent sur un écran sont composées de pixels. Chaque pixel est composé de lumières rouges, vertes ou bleues, invisibles à l’œil nu : il s’agit de luminophores. Ces éléments peuvent être « éclairés » ou « éteints ». Ils sont si proches les uns des autres que l’œil ne distingue pas les lumières colorées provenant d’un même pixel.

Les pixels sont assemblés sur une dalle plus ou moins grande constituant la base d’un écran ; il existe des dalles de petite taille (smartphone) et des dalles de grande taille (télévision).

**Annexe**

**Code couleur des résistances :**

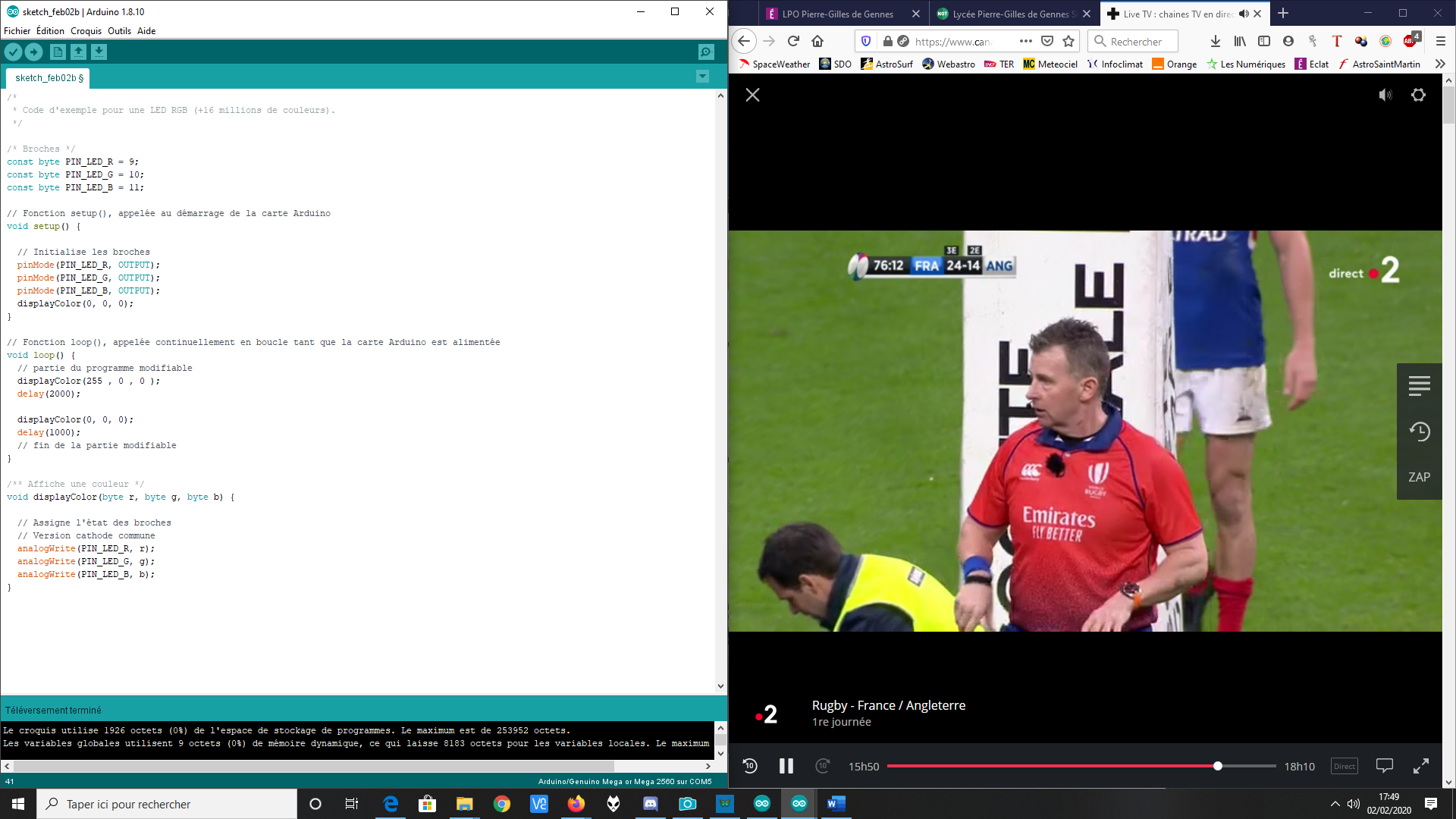
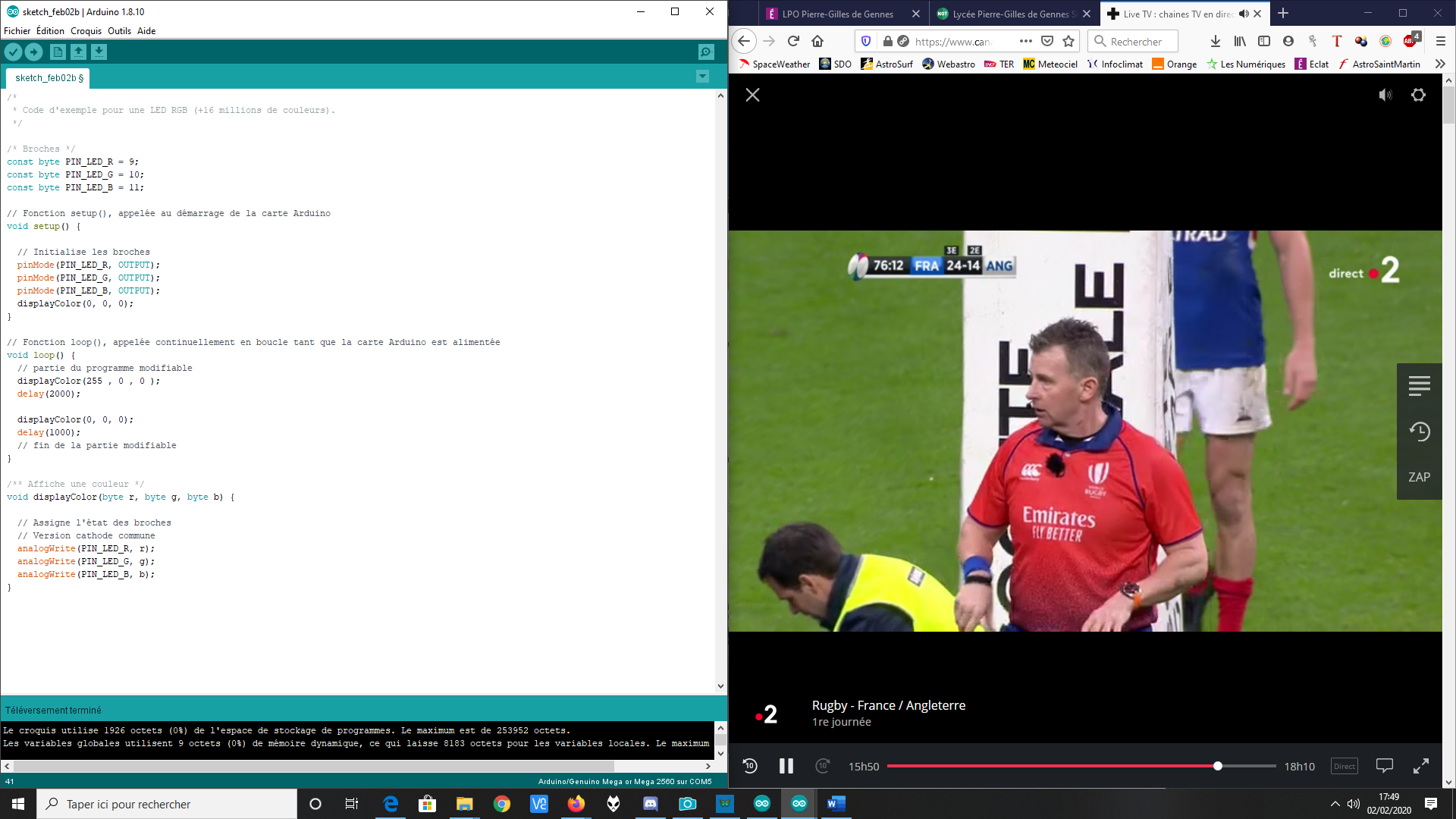
Exemple : 2 6 x 10 = 260 Ω

|  |  |  |  |
| --- | --- | --- | --- |
| 1er anneau | 2ème anneau | 3ème anneau | Valeur de la résistance (Ω) |
| Couleur : Rouge | Couleur : bleu | Couleur : Marron | R = |
| 1er chiffre = 2 | 2ème chiffre = 6 | Coefficient multiplicateur : 10 |

**Présentation d’Arduino :**

Programme de base

**Ordre de mise en service de la carte Arduino**

* Faire vider la carte Arduino par le professeur
* Ne pas brancher le câble USB
* Réalisation du circuit avec branchement sur les ports Entrée/Sortie de l’Arduino
* Ecriture du programme
* Vérification du programme le programme demande l’enregistrement du programme : accepter
* Branchement du câble USB
* Téléverser le programme

// initialisation : partie du programme réalisé 1 seule fois

// le port 13 est en mode « sortie », le port va émettre une tension.

**void setup() {**

pinMode(13, OUTPUT);

**}**

// la boucle infinie (loop)

**void loop() {**

// le port 13 est alimenté en +5V niveau « haut »

digitalWrite(13, HIGH);

// délai de 1000ms soit 1s

delay(1000);

// la boucle (loop) : le port 13 n’est pas alimenté (GND) niveau « bas »

digitalWrite(13, LOW);

delay(1000);

**}**