

## Simuler le fonctionnement d'un écran de smartphone

Un écran de smartphone est constitué de millions de pixels. Ces petits carrés dont l'ordre de grandeur d'environ 0,1 millimètre sont constitués d'une source lumineuse et de trois filtres de couleur (vert, bleu et rouge) contrôlés électroniquement.

**Comment peut-on créer un grand nombre de couleurs sur un écran de smartphone ?**



### 1. Eléments de correction et éléments de programmes associés

Eléments de programme abordés : Programme de Première Spécialité Physique-Chimie

#### A) Images et couleurs

Couleur blanche, couleurs complémentaires. Couleur des objets. Synthèse additive, synthèse soustractive. Absorption, diffusion, transmission. Vision des couleurs et trichromie.

Choisir le modèle de la synthèse additive ou celui de la synthèse soustractive selon la situation à interpréter. Séances 2 et 3  
Interpréter la couleur perçue d'un objet à partir de celle de la lumière incidente ainsi que des phénomènes d'absorption, de diffusion et de transmission. Séance 3  
Prévoir le résultat de la superposition de lumières colorées et l'effet d'un ou plusieurs filtres colorés sur une lumière incidente. Séances 2 et 3  
*Illustrer les notions de synthèse additive, de synthèse soustractive et de couleur des objets. Séances 2 et 3*

On peut adapter cette séance au programme de Sciences Numérique et Technologie (SNT) en classe de Seconde générale et technologique. Dans ce cas, on peut supprimer la question 13 et simplifier la séance 3, très axée sur les compétences de la classe de première spécialité.

Cette séquence, assez longue à mettre en œuvre, peut être découpée et raccourcie au besoin des objectifs visés par l'enseignant. La séance 1 ayant pour but d'évaluer diagnostiquement l'aisance des élèves à utiliser un microcontrôleur.

#### Eléments de correction :

Pour les séances 1 et 2, un exemple de cahier multimédia complété par un binôme est présenté comme élément de correction pour les dix premières questions :

## Utiliser un microcontrôleur pour faire clignoter un DEL ( Séance 1 )

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(2, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(2, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(5000);           // wait for a second
  digitalWrite(2, LOW); // turn the LED off by making the voltage LOW
  delay(1000);          // wait for a second
}
```

**Question 1 :** A la place de "LED\_BUILTIN", il faut indiquer la numéro de la borne où l'on a branché le câble, ici le chiffre 2.

**Question 2 :** - "void setup ()" → Notre programme porte le nom de "setup ()".

- "pinMode ( 2 , OUTPUT );" → On indique à notre carte qu'il y a un composant digital branché à la borne 2 et qu'il s'agit d'un actionneur.

- "void loop ()" → On crée une boucle qui va se répéter encore et pour toujours.

- " {} " permet d'indiquer les fonctions qui vont se répéter dans notre boucle.

- " digitalWrite ( 2 , HIGH );" → On modifie l'état du composant branché sur la borne 2. L'état devient ici *allumé*.

- " delay ( 5000 );" → L'instruction précédent va avoir une durée de 5000 ms, soit 5 secondes.

- " digitalWrite ( 2 , LOW );" → L'état du composant branché à la borne 2 est de nouveau modifié. Il est alors *éteint*.

- " delay ( 1000 );" → L'instruction précédent va avoir une durée de 1 seconde.

- " } " → Fin de la boucle qui va se répéter

**Question 3 :** On remarque que notre DEL s'allume, pendant 5 secondes, puis s'éteint pendant 1 seconde, sans arrêt

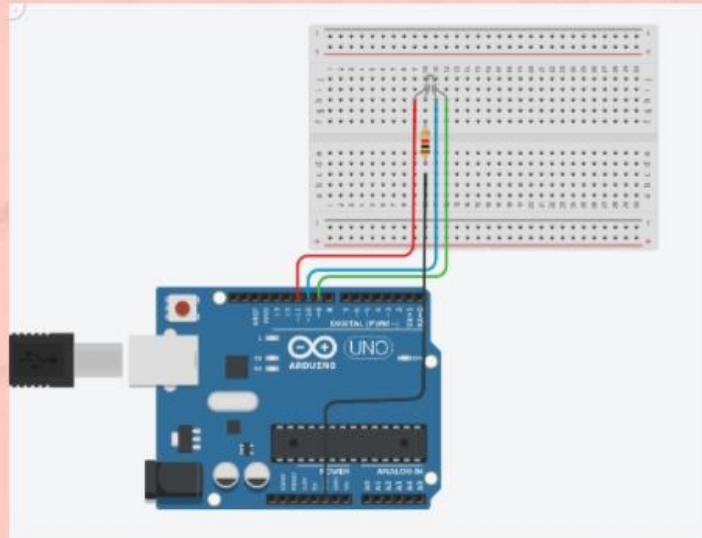
**Question 4 :** Pour que notre DEL clignote plus rapidement il faut le temps qu'elle soit allumé ou éteint soit réduit. Ce temps pour ainsi pour de 1000 ou 5000 ms à 100 ms ou bien pour un clignote encore plus rapide à 25 ms.

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(2, OUTPUT);
}

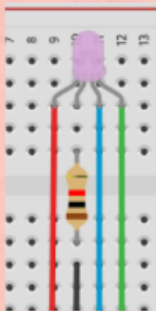
// the loop function runs over and over again forever
void loop() {
  digitalWrite(2, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(25);             // wait for a second
  digitalWrite(2, LOW); // turn the LED off by making the voltage LOW
  delay(25);            // wait for a second
}
```

## Modélisation d'un pixel par une DEL RGB ( Séance 2 )

**Question 5 :** A l'inverse de la DEL utilisée dans la première partie, la DEL RGB est composée de 4 pôles : une cathode reliée à la résistance puis à la borne GND et 3 broches de couleurs chacune reliées à une borne du côté "digital".



3 / 11



**Question 6 :** Ici on allume et éteint les LEDs rouge et bleue. On remarque que la LED RGB clignote de couleur rose/violet

```
// C++ code
//
void setup()
{
  pinMode(11/10/9, OUTPUT); // Il y a des composants aux bornes 11,10 et 9
}

void loop() {
  digitalWrite(11 , HIGH); // Allumer la LED rouge reliée à la borne 11
  digitalWrite(10 , HIGH); // Allumer la LED bleue reliée à la borne 10
  delay(1000); // Attendre une seconde
  digitalWrite(11 , LOW); // Eteindre la LED rouge reliée à la borne 11
  digitalWrite(10, LOW); // Eteindre la LED bleue reliée à la borne 10
  delay(1000); // Attendre une seconde
}
```

4 / 11



**Question 7 :** Ici on allume et éteint les LEDs rouge et verte. On remarque que la LED RGB clignote de couleur jaune

```
// C++ code
//
void setup()
{
  pinMode(11/10/9, OUTPUT); // Il y a des composants aux bornes 11,10 et 9
}

void loop() {
  digitalWrite(11, HIGH); // Allumer la LED rouge reliée à la borne 11
  digitalWrite(9, HIGH); // Allumer la LED verte reliée à la borne 9
  delay(1000); // Attendre une seconde
  digitalWrite(11, LOW); // Eteindre la LED rouge reliée à la borne 11
  digitalWrite(9, LOW); // Eteindre la LED verte reliée à la borne 9
  delay(1000); // Attendre une seconde
}
```

5 / 11



**Question 8 :** Ici on allume et éteint les LEDs bleue et verte. On remarque que la LED RGB clignote de couleur cyan

```
// C++ code
//
void setup()
{
  pinMode(11/10/9, OUTPUT); // Il y a des composants aux bornes 11,10 et 9
}

void loop() {
  digitalWrite(10, HIGH); // Allumer la LED bleue reliée à la borne 10
  digitalWrite(9, HIGH); // Allumer la LED verte reliée à la borne 9
  delay(1000); // Attendre une seconde
  digitalWrite(10, LOW); // Eteindre la LED bleue reliée à la borne 10
  digitalWrite(9, LOW); // Eteindre la LED verte reliée à la borne 9
  delay(1000); // Attendre une seconde
}
```

6 / 11





**Question 8 :** Ici on allume et éteint les LEDs des 3 couleurs. On remarque que la DEL RGB est blanche.

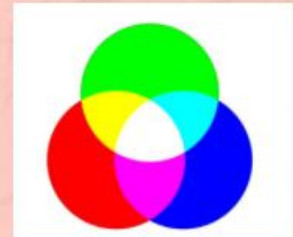
```
// C++ code
//
void setup()
{
  pinMode(11/10/9, OUTPUT); // Il y a des composants aux bornes 11,10 et 9
}

void loop() {
  digitalWrite(10 , HIGH); // Allumer la LED bleue reliée à la borne 10
  digitalWrite(9 , HIGH); // Allumer la LED verte reliée à la borne 9
  digitalWrite(11 , HIGH); // Allumer la LED rouge reliée à la borne 11
  delay(1000); // Attendre une seconde
  digitalWrite(10 , LOW); // Eteindre la LED bleue reliée à la borne 10
  digitalWrite(9 , LOW); // Eteindre la LED verte reliée à la borne 9
  digitalWrite(11 , LOW); // Eteindre la LED rouge reliée à la borne 11
  delay(1000); // Attendre une seconde
}
```

7 / 11



**Question 9 :** Entre le bleu et le vert : cyan  
Entre le rouge et le bleu : rose  
Entre le rouge et le vert : jaune  
Entre les 3 couleurs : blanc



**Question 10 :** Sans résistance, l'intensité dans le circuit est trop élevée et la DEL ne fonctionne plus dans ces conditions.

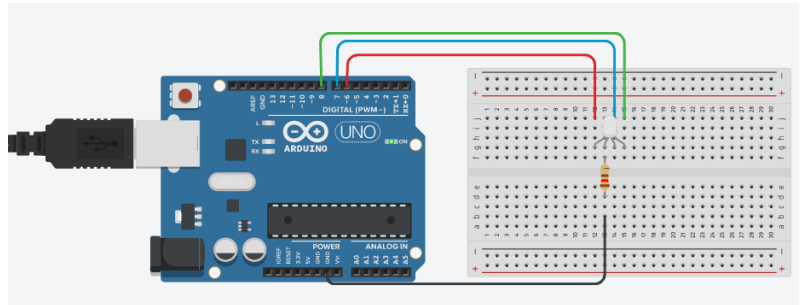
8 / 11

**Question 11 :** Reprendre le programme permettant d'activer toutes les broches de la DEL (question 8) et remplacer les fonctions digitalWrite par AnalogWrite afin d'obtenir le même résultat. Copier votre résultat sur le cahier multimédia.

```

1 // C++ code
2 //
3 void setup()
4 {
5   pinMode(6/7/8, OUTPUT);
6 }
7
8 void loop()
9 {
10  analogWrite(6, 255);
11  analogWrite(7, 255);
12  analogWrite(8, 255);
13  delay(1000); // Wait for 1000 millisecond(s)
14  analogWrite(6, 0);
15  analogWrite(7, 0);
16  analogWrite(8, 0);
17  delay(1000); // Wait for 1000 millisecond(s)
18 }

```

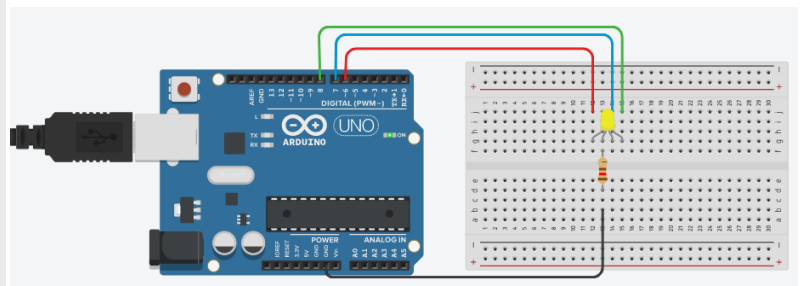


**Question 12 :** Modifier ce programme pour obtenir une couleur jaune.

```

1 // C++ code
2 //
3 void setup()
4 {
5   pinMode(6/7/8, OUTPUT);
6 }
7
8 void loop()
9 {
10  analogWrite(6, 255);
11  analogWrite(7, 0); // On éteint la DEL bleue
12  analogWrite(8, 255);
13  delay(1000);
14  analogWrite(6, 0);
15  analogWrite(7, 0);
16  analogWrite(8, 0);
17  delay(1000);
18 }

```



**Attention :** l'animation sur Tinkercad a parfois du mal à rendre les couleurs tertiaires comme l'orange. Il est recommandé de vérifier sur l'animation avant de le demander.

**Question 13 :** En modifiant la valeur du pas, on obtient une couleur différente. Ainsi en connaissant la valeur du pas pour chacune de ces trois couleurs dites **PRIMAIRES**, on peut obtenir toutes les autres couleurs qui possèdent chacun un code appelé **RGB** comportant le pas pour le rouge, celui pour le vert puis celui pour le bleu. Calculer alors le nombre de couleurs que l'on peut réaliser si le **pas est de 256 (ce qui correspond à un codage par un octet pour chaque couleur primaire en informatique)**.

On réalise le calcul suivant :

$256 \times 256 \times 256 = 16\,777\,216$  couleurs soit environ 17 millions de couleurs différentes.

**Question bonus :** Un téléphone portable moderne comporte un écran de définition 2400 pixels x 1800 pixels. Combien d'octets faut-il pour afficher une image en couleur (si le pas est bien de 256) ?

Calcul du nombre de pixels sur un écran :  $2400 \times 1800 = 4\,320\,000$  pixels

Pour chaque pixel, il faut 3 octets (un pour le rouge, un pour le vert et un pour le bleu). On peut donc calculer le nombre d'octet pour tous les pixels :

$3 \times 4\,320\,000 = 12\,960\,000$  octets = 12 656 ko = 12,4 Mo (en utilisant la conversion 1 ko =  $2^{10}$  octets)

C'est beaucoup pour une image. C'est pourquoi il existe des formats compressés.

### Éléments de corrections sur la Séance 3 ( Couleur d'un objet éclairé par la DEL RGB )

L'objectif de cette partie est de capitaliser les acquis sur la synthèse additive et les couleurs des deux séances précédentes pour aborder les notions de couleur d'un objet, synthèse soustractive et couleur complémentaire d'un objet. Une situation déclenchante sur l'éclairage dans un stade permet de faciliter la mise en activité des élèves .

Il est tout à fait possible d'espacer la séance 3 des deux premières séances au besoin.

### A – Partie expérimentale

On souhaite étudier la couleur d'un objet éclairé par une DEL RGB. Pour cela vous disposez d'un microcontrôleur (carte Arduino), de câbles électriques, d'une DEL RGB, d'un conducteur ohmique de résistance égale à 220 ohms.

### Travail à réaliser :

- Noter la couleur des objets donnés éclairés par la lumière du jour
- A l'aide de votre travail, réaliser un montage et un programme pour allumer une DEL RGB dans la couleur primaire demandée par votre professeur. Ne pas brancher le câble USB au microcontrôleur.

*Appeler le professeur pour vérification de votre montage.*

- Eteindre toute source de lumière autre que la DEL RGB allumée et observer la couleur des différents objets éclairés par la DEL RGB.

Le programme utilisé à la question 12 est remobilisé par les élèves afin de réaliser le protocole précédent illustrant la synthèse soustractive.

**Question 14 :** Modifier la couleur émise par la DEL. Choisir une des couleurs secondaires (obtenues par synthèse additive de deux couleurs primaires).

On réutilise les codes créés dans les questions 6 à 8.

**Question 15 :** Rassembler dans un tableau, la couleur des objets donnés éclairés par la lumière du jour et leur couleur éclairé par deux couleurs différentes issues de la DEL RGB.

Couleur de l'objet en lumière blanche	Couleur de l'objet en lumière jaune	Couleur de l'objet en lumière cyan
Objet blanc	jaune	cyan
Objet rouge	rouge	noir

### B – Couleur diffusée et absorbée par un objet

Pour interpréter la couleur que l'on perçoit, on modélise la lumière d'une couleur secondaire comme constitué des couleurs primaires (rouge, vert, bleu) qui la constituent.

**Question 16 :** Rappeler les couleurs primaires qui permettent d'obtenir une lumière blanche ? Même question pour la couleur secondaire que vous avez choisi.

Pour obtenir une lumière blanche, les couleurs primaires nécessaires sont le bleu, le rouge et le vert.

Pour obtenir une lumière jaune, les couleurs primaires nécessaires sont le rouge et le vert.

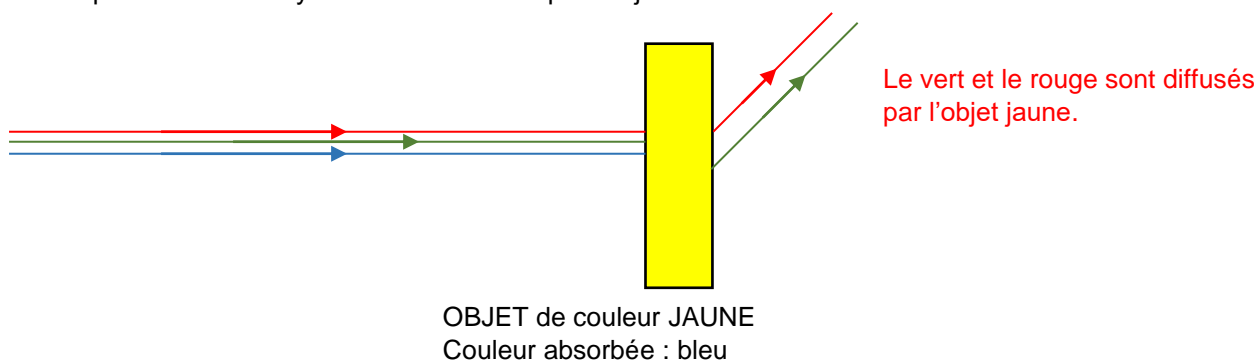
Pour obtenir une lumière cyan, les couleurs primaires nécessaires sont le vert et le bleu.

Pour obtenir une lumière magenta, les couleurs primaires nécessaires sont le rouge et le bleu.

**Question 17 :** Sachant qu'un objet diffuse uniquement les rayonnements qu'il renvoie éclairé en lumière blanche/à la lumière du jour, prévoir les couleurs primaires que renvoie un objet jaune. En déduire la couleur primaire qu'absorbe cet objet.

Un objet jaune renvoie les couleurs primaires rouge et vertes qui le constituent en synthèse additive. On en conclut qu'un objet jaune absorbe la couleur primaire qui ne la constitue pas, c'est-à-dire le bleu.

**Question 18 :** On représente ci-dessous le schéma suivant expliquant la couleur d'un objet éclairé en lumière blanche (que l'on modélise par les trois couleurs primaires). Pour un objet de la couleur de votre choix, le reproduire et compléter avec les rayonnements diffusés par l'objet dans toutes les directions :



**Question 19 :** Rappeler les couleurs primaires dont est constituée la couleur secondaire que vous avez choisie pour votre DEL RGB. Reproduire le schéma du dessus pour le même objet mais pour les couleurs émises par la LED RGB.

On attend le même type de schéma qu'à la question 18.

**Question 20 :** Vérifier que la couleur diffusée par l'objet est cohérente avec vos résultats obtenus à la partie précédente. Si ce n'est pas le cas, proposer une explication.

L'explication que l'on peut proposer est que la couleur en lumière blanche de l'objet ne contient pas exactement les proportions des trois couleurs primaires qu'attendu. On peut aussi évoquer la différence de perception des couleurs entre chaque individu.

### C – Couleur complémentaire

La couleur absorbée par l'objet est appelée **couleur complémentaire** du jaune et se trouve à l'opposé du jaune dans le schéma que vous avez complété à la question 9.

**Question 21 :** Prévoir la couleur complémentaire des deux couleurs avec laquelle vous avez éclairé votre DEL RGB.

Pour cela, on exploite les schémas introduits à la question 18 sur l'absorption et la diffusion de couleurs.

On utilise la DEL RGB de la couleur secondaire que vous avez choisie pour éclairer un objet de la couleur complémentaire à celle émise par la DEL.

**Question 22 :** Réaliser l'expérience et noter vos observations.

**Question 23 :** A l'aide d'un schéma, expliquer la couleur perçue pour cet objet. On s'aidera des schémas faits aux questions 18 et 19.

**Question 24 :** A l'aide des éléments étudiés, proposer une explication simple mais rigoureuse de cette confusion.

Le drapeau français est constitué de trois couleurs disposées selon des bandes verticales : Bleu, blanc et rouge.

Eclairée par la lumière jaune des projecteurs du stade avant le début du match, la couleur de chaque bande n'est plus la même car le jaune n'est constitué que de deux couleurs primaires : le vert et le rouge.

- La bande bleue ne peut diffuser que du bleu et non le rouge et le vert issu du projecteur. Elle ne diffuse donc aucune couleur et paraîtra noire.

- La bande blanche diffuse toutes les couleurs et paraîtra jaune sous l'éclairage des projecteurs.

- La bande rouge ne peut que diffuser du rouge. Le vert issu des projecteurs sera absorbé alors que le rouge sera diffusé : la bande paraîtra rouge.

On constate donc que l'on percevrait le drapeau français comme étant constitué de bandes verticales noire, jaune et rouge, ce qui correspond exactement au drapeau belge.

On ne recommande donc pas lors d'une rencontre France-Belgique, d'éclairer le stade avec des projecteurs jaunes !

### **Précisions des réglages :**

Il convient de vérifier toutes les DEL RGB : en particulier si la réponse au vert, au bleu et au rouge est la même ainsi que de s'assurer qu'une des 3 DEL n'est pas dysfonctionnante.

## **2. Grille d'évaluation**

---

Cette grille peut être utilisée directement par le professeur ou en auto-évaluation par les pairs (en enlevant la compétence en rouge) pour les dix premières questions uniquement.

L'évaluation laisse une place importante aux compétences S'APPROPRIER et REALISER.

### **Critères de réussite :**

Je sais....	Acquis	En cours d'acquisition	Non acquis
AUTONOMIE Faire preuve d'autonomie en suivant les instructions demandées (Séance 1)			



REALISER Modifier un programme fourni pour qu'il fonctionne correctement (q.1)			
S'APPROPRIER Commenter clairement un programme pour un lecteur non expert (q.2)			
REALISER Modifier un programme pour l'adapter à un besoin (q.4)			
COMMUNIQUER Créer un document collaboratif soigné et riche de contenu (Distanciel)			
REALISER Réaliser un schéma électrique propre et ordonné (q.5)			
REALISER Modifier le programme pour répondre aux questions demandées (q.6-7-8)			
ANALYSER Compléter un schéma en accord avec les résultats obtenus (q.9)			
VALIDER Raisonner de manière critique (q.10)			

### **3. Réflexion sur les sources d'erreur et les incertitudes**

---

Sur cette séance, les incertitudes sont difficilement quantifiables.

Par contre, on peut faire comprendre quelques sources d'erreurs qualitatives aux élèves :

- Erreur systématique due à la perception de l'œil (la sensibilité de l'œil dépend des longueurs d'ondes des lumières reçues)
- Erreur systématique due à l'intensité lumineuse de chaque DEL contenue dans la DEL RGB

### **4. Témoignages d'élèves**

---

#### **Retours sur la première séance :**

Les élèves étaient plus ou moins à l'aise mais les instructions les ont bien aidés.

#### **Retours sur la deuxième séance (distanciel) :**

- Aucune difficulté sur la réalisation d'un montage sur Tinkercad.
- Il y a eu quelques problèmes techniques sur Tinkercad (code impossible à modifier, quelques difficultés à s'inscrire) qui justifient de passer une partie de la séance 1 à faire l'inscription sur le site en classe.
- Quelques difficultés pour certains élèves pour tordre les fils sur le montage
- Les étapes claires et le fait de faire le montage arduino « blink » en séance 1 aide beaucoup les élèves pour la deuxième séance et leur permet de le retravailler tranquillement, à leur rythme !
- Les élèves y ont passé de 1h30 à 2h.

#### **Retours sur la troisième séance :**

- Aucun problème noté par les élèves : ils se sentent à l'aise.

#### **Utilisation de l'outil « cahier multimédia » de l'ENT NEO HDF :**

- Outil facile à utiliser même si les élèves regrettent qu'ils ne puissent pas l'utiliser sur téléphone
- Aide importante à la collaboration pour la plupart des élèves. Pour certains, cela n'a pas eu l'effet escompté.

#### **Retours généraux des élèves sur la séquence :**

- Très intéressant. La séance en distanciel a rassuré certains élèves et leur ont permis de gagner en confiance sur le microcontrôleur.
- Les élèves intéressés par l'informatique et/ou l'électronique ont particulièrement apprécié la séance.

### **5. Nature des plus-values**

---

- La mise en activité des élèves est grandement facilitée par l'aspect expérimental. La première séance permet ainsi de gérer l'hétérogénéité des savoir-faire des élèves au sujet des microcontrôleurs.

- L'outil numérique permet aux élèves de s'approprier davantage l'utilisation du microcontrôleur hors de la classe, ce qui serait impossible sans l'outil numérique. C'est un gain de temps conséquent en première spécialité où chaque heure d'enseignement est précieuse.
- L'utilisation du cahier numérique (ou de tout autre outil fourni par l'ENT académique) permet de travailler à distance et en collaboration. Utiliser un outil où l'on peut voir les modifications réalisés par les élèves est un vrai plus et aide à la collaboration dans les binômes où le travail n'est pas réparti équitablement entre les élèves.