

Niveau : spécialité première	Prévoir le résultat de la superposition de lumières colorées La lumière : partie A : images et couleurs
Couleur des objets. Synthèse additive.	Prévoir le résultat de la superposition de lumières colorées. <i>Illustrer les notions de synthèse additive, de synthèse soustractive et de couleur des objets.</i>
Capacités expérimentales.	<i>Mettre en œuvre un dispositif pour illustrer la synthèse additive ou la synthèse soustractive. Commander la production d'un signal grâce à un microcontrôleur.</i>

Dossier TP Plug'Uino® : Synthèse additive RVB Programmation en langage Arduino



1. Résumé de l'activité

Cette activité expérimentale a pour but de produire un signal lumineux en utilisant la synthèse additive de lumière à l'aide d'un microcontrôleur et d'une matrice de DELs RVB.

3 activités basées sur ce principe sont proposées :

- Un exemple simple, affichant les trois valeurs R, V et B, ainsi que leur somme, à l'aide d'une programmation très basique ;
- Le même exemple, mais cette fois-ci en colonne et abordant une programmation un peu plus technique (utilisation de boucle for)
- La réalisation d'un dessin 5x5 en couleurs, à l'aide d'une autre technique de programmation.

Les trois activités sont indépendantes, basées sur divers niveaux de programmation, et sont systématiquement proposées avec 3 niveaux de programmation : simple, avancé et expert.

2. Thème du programme abordé

Niveau : spécialité première	Prévoir le résultat de la superposition de lumières colorées La lumière : partie A : images et couleurs
Couleur des objets. Synthèse additive.	Prévoir le résultat de la superposition de lumières colorées. <i>Illustrer les notions de synthèse additive, de synthèse soustractive et de couleur des objets.</i>
Capacités expérimentales.	<i>Mettre en œuvre un dispositif pour illustrer la synthèse additive ou la synthèse soustractive.</i> <i>Commander la production d'un signal grâce à un microcontrôleur.</i>

3. Matériel mis en œuvre

Une interface Plug'Uino®

Une matrice de DELs RVB 5x5 Plug'Uino® Réf. 651 050.

4. Présentation des expériences réalisées

4.1. Exemple simple

On allume différents pixels en entrant directement leurs valeurs R, V, B. On utilise la matrice ligne par ligne pour représenter les valeurs de Rouge, Vert et Bleu, ainsi que le résultat de la synthèse additive sur la dernière colonne, de la manière suivante :



Ici, la programmation est très basique : on entre directement les valeurs que l'on veut, sans utilisation de variables ou de boucles.

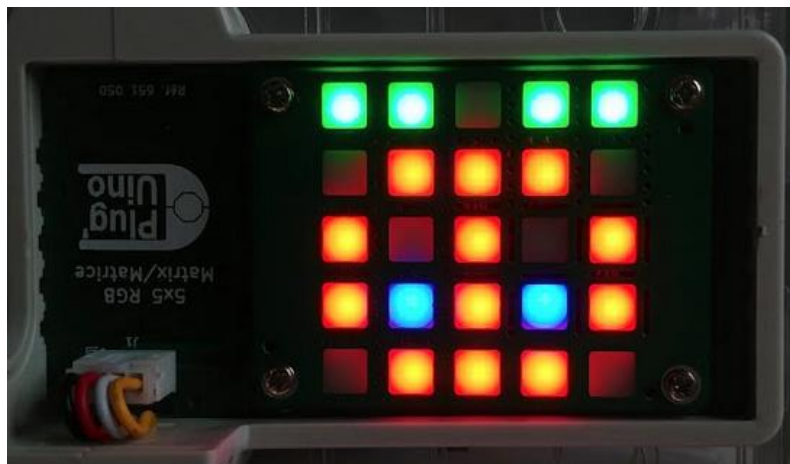
4.2. Représentation en colonnes

Cette fois-ci, la représentation est la même, mais sous forme de colonne, et en utilisant une programmation un peu plus technique (utilisation d'une boucle, de variables) :



4.3. Réalisation d'un dessin en couleurs

Ce programme permet de dessiner un motif en couleur sur la matrice de LED, à partir d'instructions plus évoluées. Le résultat peut être le suivant :



5. Programmes Arduino

5.1. Exemple simple

Voici le programme Arduino fonctionnel, permettant l'exemple simple est le suivant.

La mise en œuvre est simple :

- relier le module Matrice RVB au Plug'Uino® (ici sur la prise D2).
- lancer le programme et observer ;
- changer des paramètres pour comprendre le fonctionnement de la synthèse.

SyntheseRVB_Exemple_Simple

```

1 //
2 //     Sciencéthic
3 //     Synthèse RVB
4 //     Exemple simple
5 //
6 //     Cette utilisation nécessite l'installation de la bibliothèque
7 //     Adafruit NeoPixel
8 //     disponible dans "gérer les bibliothèques"
9 //
10
11 # include<Adafruit_NeoPixel.h>    // Inclut la bibliothèque Adafruit_NeoPixel
12                                   // Qui permet de gérer l'afficheur 5x5 RVB
13                                   // Plug'Uino
14
15 Adafruit_NeoPixel matriceLED = Adafruit_NeoPixel(25,2, NEO_GRB + NEO_KHZ800);
16                                   // Initialise la matrice : 25 est le nombre de pixels
17                                   // 2 est la sortie utilisée (ici, il faut donc brancher
18                                   // la matrice sur D2)
19
20 void setup()                      // Initialisation du programme
21 {
22     matriceLED.begin();           // Initialisation de la matrice
23 }
24
25 void loop()                      // Boucle principale
26 {
27     // La syntaxe est la suivante :
28     // matriceLED.setPixelColor(numéro de la LED, matriceLED.Color(valeur Rouge, valeur Vert, Valeur Bleu)
29     // Les valeurs des couleurs vont de 0 (couleur éteinte) à 100 (intensité la plus forte).
30     // La LED 0 est la première LED en haut à gauche, puis la numérotation se fait en ligne : la LED 4 est
31     // la dernière LED de la première ligne, la LED 5 la première LED de la deuxième ligne etc...
32
33     // On va afficher ici sur les colonnes 0, 1, 2 les trois couleurs primaires R V B
34     // et sur la dernière colonne la somme des trois premières colonnes
35
36     // Affichage de Rouge + Vert = Jaune en ligne 0
37     matriceLED.setPixelColor(0, matriceLED.Color(100,0,0));
38     matriceLED.setPixelColor(1, matriceLED.Color(0,100,0));
39     matriceLED.setPixelColor(4, matriceLED.Color(100,100,0));
40
41     // Affichage de Rouge + Bleu = Magenta en ligne 1
42     matriceLED.setPixelColor(5, matriceLED.Color(100,0,0));
43     matriceLED.setPixelColor(7, matriceLED.Color(0,0,100));
44     matriceLED.setPixelColor(9, matriceLED.Color(100,0,100));
45
46     // Affichage de Vert + Bleu = Cyan en ligne 2
47     matriceLED.setPixelColor(11, matriceLED.Color(0,100,0));
48     matriceLED.setPixelColor(12, matriceLED.Color(0,0,100));
49     matriceLED.setPixelColor(14, matriceLED.Color(0,100,100));
50
51     // Affichage de Rouge + Bleu + Vert = Blanc en ligne 3
52     matriceLED.setPixelColor(15, matriceLED.Color(100,0,0));
53     matriceLED.setPixelColor(16, matriceLED.Color(0,100,0));
54     matriceLED.setPixelColor(17, matriceLED.Color(0,0,100));
55     matriceLED.setPixelColor(19, matriceLED.Color(100,100,100));
56
57     matriceLED.show();             // Affiche la matrice
58 }

```

5.2. Exemple en colonne

Voici le programme Arduino fonctionnel, permettant l'exemple en colonne.

La mise en œuvre est simple :

- relier le module Matrice RVB au Plug'Uino® (ici sur la prise D2).
- lancer le programme,
- modifier des paramètres pour comprendre le fonctionnement du programme et l'étude de la synthèse RVB.

SyntheseRVB_en_colonne\$

```

1 //
2 //      Sciencéthic
3 //      Synthèse RVB
4 //      Exemple en colonnes
5 //
6 //      Cette utilisation nécessite l'installation de la bibliothèque
7 //      Adafruit NeoPixel
8 //      disponible dans "gérer les bibliothèques"
9 //
10
11 # include<Adafruit_NeoPixel.h>    // Inclut la bibliothèque Adafruit_NeoPixel
12                                   // Qui permet de gérer l'afficheur 5x5 RVB
13                                   // Plug'Uino
14
15 Adafruit_NeoPixel matriceLED = Adafruit_NeoPixel(25,2, NEO_GRB + NEO_KHZ800);
16                                   // Initialise la matrice : 25 est le nombre de pixels
17                                   // 2 est la sortie utilisée (ici, il faut donc brancher
18                                   // la matrice sur D2)
19
20 void setup()                      // Initialisation du programme
21 {
22     matriceLED.begin();           // Initialisation de la matrice
23 }
24
25 void loop()                      // Boucle principale
26 {
27     // La syntaxe est la suivante :
28     // matriceLED.setPixelColor(numéro de la LED, matriceLED.Color(valeur Rouge, valeur Vert, Valeur Bleu)
29     // Les valeurs des couleurs vont de 0 (couleur éteinte) à 100 (intensité la plus forte).
30     // La LED 0 est la première LED en haut à gauche, puis la numérotation se fait en ligne : la LED 4 est
31     // la dernière LED de la première ligne, la LED 5 la première LED de la deuxième ligne etc...
32
33     int Rouge = 100 ;
34     int Vert = 100;
35     int Bleu = 00;
36
37     for (int i = 0; i <=20; i=i+5){
38         matriceLED.setPixelColor(i, matriceLED.Color(Rouge,0,0));
39         matriceLED.setPixelColor(i+1, matriceLED.Color(0,Vert,0));
40         matriceLED.setPixelColor(i+2, matriceLED.Color(0,0,Bleu));
41         matriceLED.setPixelColor(i+4, matriceLED.Color(Rouge,Vert,Bleu));
42     }
43 }
44
45 matriceLED.show();               // Affiche la matrice
46 }

```

5.2. Dessin en couleurs

Voici le programme Arduino fonctionnel, permettant l'exemple en colonne.

La mise en œuvre est simple :

- relier le module Matrice RVB au Plug'Uino® (ici sur la prise D2).
- lancer le programme,
- modifier des paramètres pour comprendre le fonctionnement du programme et l'étude de la synthèse RVB.

SyntheseRVB_Dessin_Couleur\$

```

1 //
2 //     Sciencéthic
3 //     Synthèse RVB
4 //     Réalisation d'un dessin en couleurs
5 //
6 //     Cette utilisation nécessite l'installation de la bibliothèque
7 //     Adafruit NeoPixel
8 //     disponible dans "gérer les bibliothèques"
9 //
10
11 # include<Adafruit_NeoPixel.h>    // Inclut la bibliothèque Adafruit_NeoPixel
12                                   // Qui permet de gérer l'afficheur 5x5 RVB
13                                   // Plug'Uino
14
15 Adafruit_NeoPixel matriceLED = Adafruit_NeoPixel(25,2, NEO_GRB + NEO_KHZ800);
16                                   // Initialise la matrice : 25 est le nombre de pixels
17                                   // 2 est la sortie utilisée (ici, il faut donc brancher
18                                   // la matrice sur D2)
19
20 // Définition du dessin à afficher
21 int matrice[5][5] = {
22 {0,1,1,1,0},           // Le code est le suivant :
23 {1,3,1,3,1},           // 0 : pixel éteint
24 {1,0,1,0,1},           // 1 : pixel rouge
25 {0,1,1,1,0},           // 2 : pixel vert
26 {2,2,0,2,2},           // 3 : pixel bleu
27 };
28 void setup()                // Initialisation du programme
29 {
30   matriceLED.begin();       // Initialisation de la matrice
31 }
32
33 void loop()                  // Boucle principale
34 {
35   // La syntaxe est la suivante :
36   // matriceLED.setPixelColor(numéro de la LED, matriceLED.Color(valeur Rouge, valeur Vert, Valeur Bleu)
37   // Les valeurs des couleurs vont de 0 (couleur éteinte) à 100 (intensité la plus forte).
38   // La LED 0 est la première LED en haut à gauche, puis la numérotation se fait en ligne : la LED 4 est
39   // la dernière LED de la première ligne, la LED 5 la première LED de la deuxième ligne etc...
40
41   int Rouge;
42   int Vert;
43   int Bleu;
44
45   for(int y = 0; y < 5; y++)    // Parcourt chaque colonne du tableau
46     for(int x = 0; x < 5; x++) { // Parcourt chaque ligne du tableau
47       {
48         switch(matrice[x][y])    // selon la valeur lue dans le tableau dessin(x,y)
49         {
50           case 0: Rouge=0; Vert=0; Bleu=0; break;    // si la valeur vaut 0, ne pas l'allumer
51           case 1: Rouge=100; Vert=0; Bleu=0; break;  // si la valeur vaut 1, l'allumer en rouge
52           case 2: Rouge=0; Vert=100; Bleu=0; break;  // si la valeur vaut 2, l'allumer en vert
53           case 3: Rouge=0; Vert=0; Bleu=100; break;  // si la valeur vaut 2, l'allumer en bleu
54         }
55       }
56       matriceLED.setPixelColor(x*5+y, matriceLED.Color(Rouge,Vert,Bleu)); // Allume le pixel en correspondance
57     }
58   }
59   matriceLED.show();           // Affiche la matrice
60 }

```

6. Exploitation

6.1. Programmation

En fonction des capacités des différents élèves on peut leur demander diverses choses :

- à un niveau **d'initiation**, on peut enlever certains commentaires du ou des programmes utilisés et demander aux élèves à quoi correspondent ces lignes de programmation ;
- à un niveau **plus avancé**, on peut effacer certaines lignes et demander aux élèves de programmer eux-mêmes ces lignes :

Exemple sur le programme simple : supprimer les lignes de code 41 à 55 en laissant les commentaires, et en demandant aux élèves de réécrire ces lignes de code.

Exemple sur le programme en colonnes : supprimer les lignes de code 39 à 41, en demandant aux élèves de réécrire ces lignes de code, et en leur montrant le résultat voulu à l'aide de l'illustration 4.2.

Exemple sur le programme de dessin en couleurs : supprimer les lignes de code 51 à 53,, en demandant aux élèves de réécrire ces lignes de code, et en leur montrant le résultat voulu à l'aide de l'illustration 4.3.

- à un niveau **expert** on peut demander aux élèves de programmer le microcontrôleur en partie.

Exemple sur le programme simple : supprimer les lignes de code 38 à 55 en laissant les commentaires, et en demandant aux élèves de réécrire ces lignes de code.

Exemple sur le programme en colonnes : supprimer les lignes de code 37 à 41, en demandant aux élèves de réécrire ces lignes de code en utilisant une boucle for, et en leur montrant le résultat voulu à l'aide de l'illustration 4.2.

Exemple sur le programme de dessin en couleurs : modifier/ajouter des lignes de codes de façon à pouvoir obtenir des couleurs supplémentaires (par exemple programmer pour que le numéro 4 corresponde à l'affichage de pixels en jaune, que le numéro 5 permette d'afficher des pixels en magenta, etc.) et réaliser ainsi un dessin possédant des couleurs supplémentaires.

6.2. Étude de la synthèse additive

Dans le cadre de l'étude sur la synthèse additive, on peut faire utiliser les programmes fournis aux élèves pour :

- étudier les résultats obtenus lors de la synthèse additive,
- comprendre le principe de la synthèse additive,
- modifier des valeurs au sein des programmes de façon à observer les changements lors des synthèses.

7. Pistes d'exploration possible

Nous n'avons utilisé ici que des valeurs de 100 pour l'intensité des LEDs rouges, vertes et bleues. Il est possible de poursuivre la séquence en faisant varier les valeurs d'intensité afin de faire apparaître d'autres couleurs : on peut demander aux élèves de faire apparaître de l'orange, du rose, du violet, du marron...