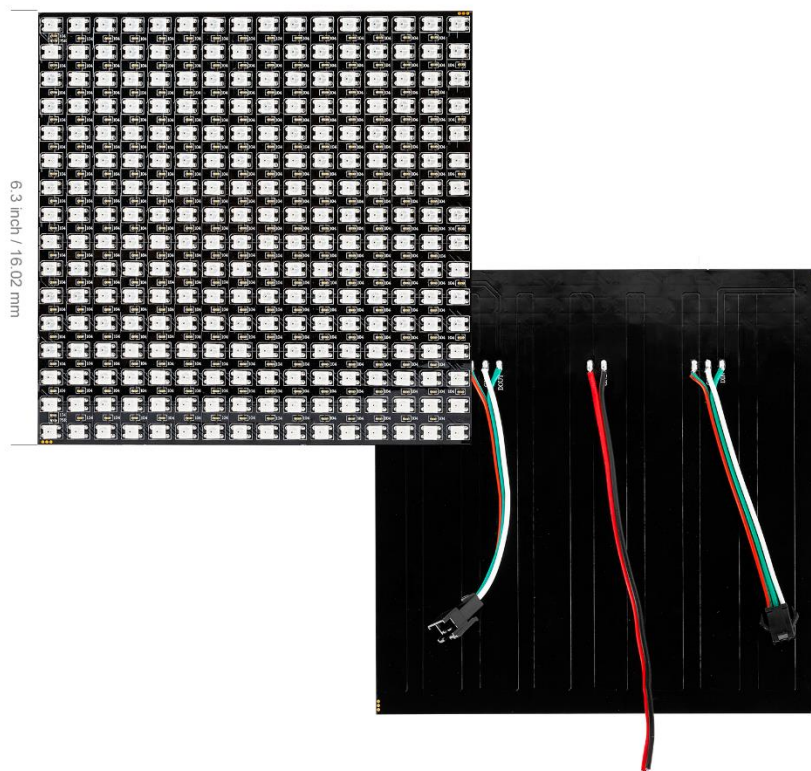


# ARDUINO: MATRICE LED 16X16

## WS2812B PARTIE C++



16\*16 PIXELS



## SOMMAIRE

Introduction.....	3
Composants :.....	3
Matrice : .....	3
Carte électronique : .....	3
Schéma électronique : .....	3
Code Arduino C++.....	4
Initialisation bibliothèque FastLed : .....	4
Initialisation des defines : .....	4
Initialisation Variable : .....	5
Void setup : .....	5
Void loop : .....	6
Fonction Matrice 16x16 : .....	6
Fonction papillon (exemple) : .....	7
Information complémentaire.....	9
Erreur possible : .....	9
Remerciement : .....	9
Fin de rapport : .....	9



## INTRODUCTION

Ce rapport est consacré au code C++ dans l'arduino nano à l'électronique du panneau led et comment le système fonctionne.

### Composants :

- ❖ Carte Arduino nano
- ❖ Matrice Led 16x16 WS2812B

Pour ce projet, on utilise une bibliothèque pour contrôler la matrice de led, la bibliothèque est FastLed.

### Matrice :

La matrice est composée de 256 leds qui sont reliées comme un ruban de la façon suivante :

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
112	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97
113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128
144	143	142	141	140	139	138	137	136	135	134	133	132	131	130	129
145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
176	175	174	173	172	171	170	169	168	167	166	165	164	163	162	161
177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192
208	207	206	205	204	203	202	201	200	199	198	197	196	195	194	193
209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224
240	239	238	237	236	235	234	233	232	231	230	229	228	227	226	225
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256

L'on peut voir que les leds sont relié en zigzag et non en ligne.

### Carte électronique :

Pour ce projet, on utilise une carte arduino nano et le 5V sert à alimenter le Vin de la matrice la data est assurée par le pin 4 et on connecte le ground de la matrice à la carte.

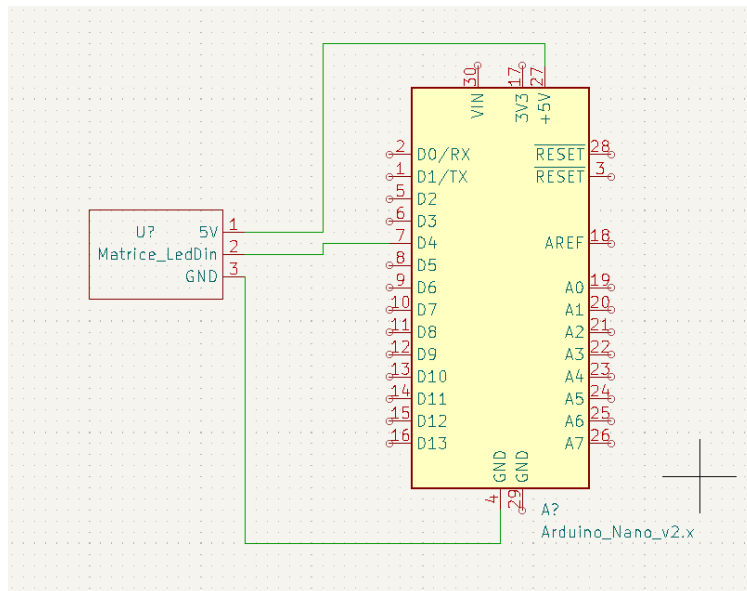
### Schéma électronique :

Voici le schéma où l'on connecte la carte Arduino à la matrice de led16x16 :

- ❖ Le 5V au Vin



- ❖ Le pin 4 au Din
- ❖ Le GND au GND



(Schéma fait via Kicad)

## CODE ARDUINO C++

Cette partie va traiter les différentes fonctions et parties du code et les expliquer.

### Initialisation bibliothèque FastLed :

Si vous ne savez pas installer une bibliothèque, voici un tutoriel ([https://fablabutc.fr/wp-content/uploads/2021/01/Tutoriel\\_Installer-une-bibliotheque-pour-Arduino.pdf](https://fablabutc.fr/wp-content/uploads/2021/01/Tutoriel_Installer-une-bibliotheque-pour-Arduino.pdf))

Dans un premier temps il faut inclure la bibliothèque via la commande `#include` :

```
2 //-----bibliothèque-----//
3 #include <FastLED.h>
4 //-----//
```

### Initialisation des defines :

Dans un second temps, après avoir inclus la bibliothèque, on définit le nombre de leds et le pin qui servirait de data pour la matrice via un `#define`.

```
6 //-----Define-----//
7 #define NUM_LEDS 256 //Nombre de Led
8 #define DATA_PIN 4 //Pin de la data
9 //-----//
```



## Initialisation Variable :

Dans un troisième temps, on crée et on initialise les différentes variables qui serviront dans tous le programme

```
11 //-----Variable-----//
12 CRGB leds[NUM_LEDS]; //tableau led réelle
13 CRGB leds_16x16[16][16]; //tableau led application
14 int i,k,t; //variable boucle
15 int carreaux; //variable position led matrice réel
16 //-----//
```

Il est important de savoir que la matrice de led est un ruban pour pouvoir éventuellement créer des animations de déplacement ou autre, on utilise une variable temporaire la variable leds\_16x16 et une fonction fera le lien avec cette variable et la variable qui sert pour l'affichage la variable leds.

CRGB est une variable crée par la bibliothèque Fastled, cette variable est un tableau et chaque case contient le code couleur de 1 led. En réalité chaque case du tableau contient un tableau de trois cases et chaque case du deuxième tableau contient des valeurs entières de 0 à 255. La première case est pour le Rouge, la deuxième est pour le Vert et la troisième est pour le Bleu.

CRGB leds[NUM\_LEDS] : est la variable réelle qui contient les différentes couleurs de leds de la matrice de leds qui sert à afficher les couleurs.

CRGB leds\_16x16[16][16] : est la variable temporaire qui sert avant la conversion dans la variable dans la variable leds. L'application C# donne le code de cette variable.

i,k,t : ces variables servent aux différentes boucle for

carreaux : cette variable est utilisée dans la conversion entre la variable leds\_16x16 et la variable leds, elle donne la position de la led actuelle.

## Void setup :

Dans un quatrième temps on code la fonction d'initialisation void setup

```
18 void setup() {
19
20     //-----Initialisation-----//
21     FastLED.addLeds<NEOPIXEL, DATA_PIN>(leds, NUM_LEDS); //initialisation Bandeau et pin
22     FastLED.setBrightness(12); //reglage luminosité global
23     //-----//
24
25 }
```

Dans cette fonction la première ligne sert à déclarer que le pin 4 sert a une matrice de led et que il y a 256 leds.

La deuxième ligne sert à définir la luminosité globale de toute les leds.



### Void loop :

Dans un quatrième temps on programme la fonction void loop

```
27 void loop() {  
28  
29     papillon();  
30  
31 }
```

On utilise juste une fonction ici la fonction exemple papillon qui va afficher un papillon.

### Fonction Matrice 16x16 :

Cette fonction est celle qui fait le lien entre la variable temporaire leds\_16x16 et la variable réelle leds.

```
33 void Matrice_16x16()  
34 {  
35     /*  
36     * Fonction conversion tableau 2D  
37     * en 1D représentation réel  
38     * Ruban du tableau  
39     *  
40     * Jérémy Clément 11/03/2022  
41     */  
42  
43     //variable  
44     carreaux = 0;  
45  
46     //Début  
47     for (k = 0; k<16; k = k + 1)  
48     {  
49         if (k % 2 == 0) {  
50             for (i=15; i>=0; i = i - 1)  
51             {  
52                 leds[carreaux] = leds_16x16[k][i];  
53                 carreaux = carreaux + 1 ;  
54             }  
55         }  
56         else  
57         {  
58             for (i=0; i<16; i = i + 1)  
59             {  
60                 leds[carreaux] = leds_16x16[k][i];  
61                 carreaux = carreaux + 1 ;  
62             }  
63         }  
64     }  
65     //Fin  
66 }
```



La fonction de conversion est composée de plusieurs parties, une première boucle for qui est les différentes lignes puis une condition si sinon. Il sert à déterminer si dans la ligne le sens des carreaux de gauche à droite ou alors de droite à gauche. Ensuite, c'est juste une conversion entre les deux fonctions.

Cette fonction est utilisée pour chaque code générer par l'application C# que l'on souhaitera afficher. Il faut juste rajouter deux lignes le nom de la fonction et l'allumage des leds, ceci n'est pas inclus dans la fonction, au cas du changement de nom de la fonction et si l'on ne souhaite pas afficher les leds tout de suite.

### Fonction papillon (exemple) :

Le code a été coupé du as sa longueur et surtout sa répartition. Voici un exemple de code généré grâce à l'application C#.

```
68 void papillon()
69 {
70     /*
71      * Code générer via l'application
72      * dans ce cas c'est le dessin d'un
73      * papillon
74      *
75      * Jérémy Clément 11/03/2023
76      */
77
78     //Début
79
80     //-----Noir-----//
81     leds_16x16[0][0] = CRGB::Black;
82     leds_16x16[0][1] = CRGB::Black;
83     leds_16x16[0][2] = CRGB::Black;
84     leds_16x16[0][3] = CRGB::Black;
85     leds_16x16[0][4] = CRGB::Black;
86     leds_16x16[0][5] = CRGB::Black;
87     leds_16x16[0][6] = CRGB::Black;
88     leds_16x16[0][7] = CRGB::Black;
89     leds_16x16[0][8] = CRGB::Black;
90     leds_16x16[0][9] = CRGB::Black;
91     leds_16x16[0][10] = CRGB::Black;
92     leds_16x16[0][11] = CRGB::Black;
93     leds_16x16[0][12] = CRGB::Black;
94     leds_16x16[0][13] = CRGB::Black;
95     leds_16x16[0][14] = CRGB::Black;
96     leds_16x16[0][15] = CRGB::Black;
97     leds_16x16[1][0] = CRGB::Black;
98     leds_16x16[1][1] = CRGB::Black;
99     leds_16x16[1][2] = CRGB::Black;
```



```

312 leds_16x16[3][11] = CRGB::Yellow;
313 leds_16x16[4][2] = CRGB::Yellow;
314 leds_16x16[4][3] = CRGB::Yellow;
315 leds_16x16[4][4] = CRGB::Yellow;
316 leds_16x16[4][5] = CRGB::Yellow;
317 leds_16x16[4][9] = CRGB::Yellow;
318 leds_16x16[4][10] = CRGB::Yellow;
319 leds_16x16[4][11] = CRGB::Yellow;
320 leds_16x16[4][12] = CRGB::Yellow;
321 leds_16x16[5][2] = CRGB::Yellow;
322 leds_16x16[5][5] = CRGB::Yellow;
323 leds_16x16[5][9] = CRGB::Yellow;
324 leds_16x16[5][12] = CRGB::Yellow;
325 leds_16x16[6][2] = CRGB::Yellow;
326 leds_16x16[6][6] = CRGB::Yellow;
327 leds_16x16[6][8] = CRGB::Yellow;
328 leds_16x16[6][12] = CRGB::Yellow;
329 leds_16x16[7][2] = CRGB::Yellow;
330 leds_16x16[7][3] = CRGB::Yellow;
331 leds_16x16[7][6] = CRGB::Yellow;
332 leds_16x16[7][8] = CRGB::Yellow;
333 leds_16x16[7][11] = CRGB::Yellow;
334 leds_16x16[7][12] = CRGB::Yellow;
335 leds_16x16[8][3] = CRGB::Yellow;
336 leds_16x16[8][4] = CRGB::Yellow;
337 leds_16x16[8][5] = CRGB::Yellow;
338 leds_16x16[8][9] = CRGB::Yellow;
339 leds_16x16[8][10] = CRGB::Yellow;
340 leds_16x16[8][11] = CRGB::Yellow;
341 //-----//
342
343 Matrice_16x16();//Conversion tableau 2d théorique en matrice réel
344 FastLED.show();//après conversion affichage des leds selon leurs couleurs
345
346 //Fin
347 }

```

Le code est composé de plusieurs parties, chaque pixel possède une couleur propre. Même celles qui ne doivent pas s'allumer, le code est trié selon les couleurs. Après avoir ajouté le code généré par l'application, il faut rajouter deux lignes `Matrice_16x16()` ; nom de la fonction qui permet la conversion et aussi `FastLED.show()` ; commande permet l'affichage. La fonction papillon est un exemple donné qui marche sur la matrice. Le programme a été mis dans une fonction pour éviter d'avoir beaucoup de lignes dans le loop et dans le cas d'affichage de plusieurs images différentes et pour faciliter le débogage aussi.

Il existe plusieurs méthodes pour coder les lumières, soit on utilise les raccourcis `Red` `Orange` (exemple : `leds_16x16[0][0] = CRGB ::Red ;`) sinon on peut définir manuellement le Rouge Vert et Bleu (exemple : `leds_16x16[0][0] = CRGB(156,255,80)`)





ou alors on peut aussi la couleur en Hexadécimal (exemple : leds\_16x16[0][0] = CRGB(0x7FFFD4)).

## INFORMATION COMPLÉMENTAIRE

### Informations importantes :

- ❖ Il est possible que si on utilise cette fonction pour une autre matrice différente de celle que j'ai utilisé, cela ne peut pas marcher, ceci peut être due au schéma électronique ou autre chose.
- ❖ Si la bibliothèque change ou cause de mise à jour de Fastled et que les commandes changent, le programme peut ne plus fonctionner.
- ❖ Si l'on prend une matrice différente comment la 8x8 ou la 8x32, il faut changer le programme et aussi l'exemple ne marchera pas.
- ❖ La matrice de led est RGB et non RGBW ou d'autre genre dans le cas d'une matrice RGBW le code ne marchera sûrement pas.

### Erreur possible :

- ❖ Si l'alimentation de la matrice est effectuée par une carte secondaire que la carte qui envoie la data et que les GND des deux cartes cela peut potentiellement créer des problèmes et des « Glitch » sur les images.
- ❖ Il est préférable de donner à chaque led quelle couleur elle doit prendre (même le noir) car si le motif est trop petit cela peut créer des « glitches ».
- ❖ Définir chaque couleur de led permet d'éviter que si on superpose 2 images il peut avoir des restes de l'image précédente.
- ❖ Une autre erreur possible est d'essayer d'afficher le résultat sans utiliser la fonction de conversion ou alors de convertir et de ne pas afficher le résultat.

### Remerciement :

Je remercie Mr Paulmier (professeur d'électronique) qui m'a aidé dans l'interprétation de la matrice de led et qui m'a donné l'idée de créer la fonction de conversion pour permettre de potentiel animation de dessins.

### Fin de rapport :

Ceci est la fin du rapport de la partie C++ du code physique de la matrice led je conseille vivement de jeter un coup d'œil au rapport sur la partie de l'application C#. Les datasheets des composants utilisés sont dans le dossier documentation.

