

Manuel d'utilisation du shield matrice 40 leds pour Arduino

Cher client,

Merci d'avoir acheté notre produit.

Veuillez observer les instructions ci-dessous avant la première utilisation :



Branchements du module

Ce shield n'exige aucun branchement particulier, il s'insère simplement dans les connecteurs latéraux de l'Arduino (ou compatible). Il est également alimenté par celui-ci



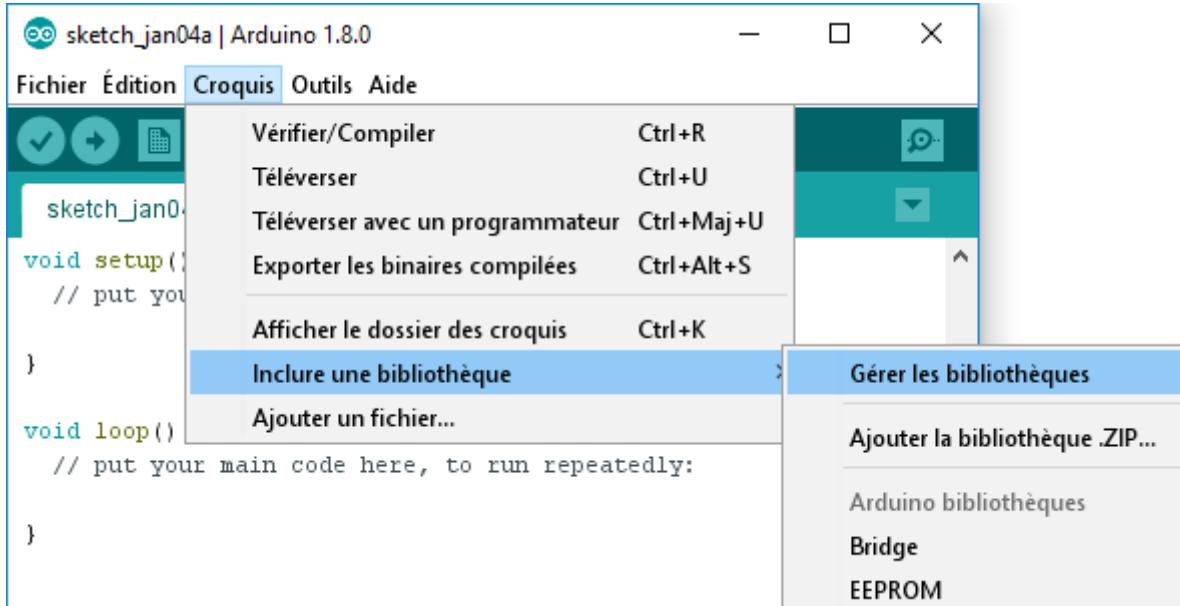
La matrice est composée de leds adressables et n'utilise qu'une seule sortie de l'Arduino pour les commander. L'accès au reste des broches de l'Arduino reste disponible sur le shield.

On définit la couleur des leds adressables en leur envoyant une commande contenant les intensités des trois couleurs : rouge, vert et bleu. Par exemple, on enverra le code (255, 0, 0) pour obtenir la couleur rouge (voir l'exemple de programme en page 3).

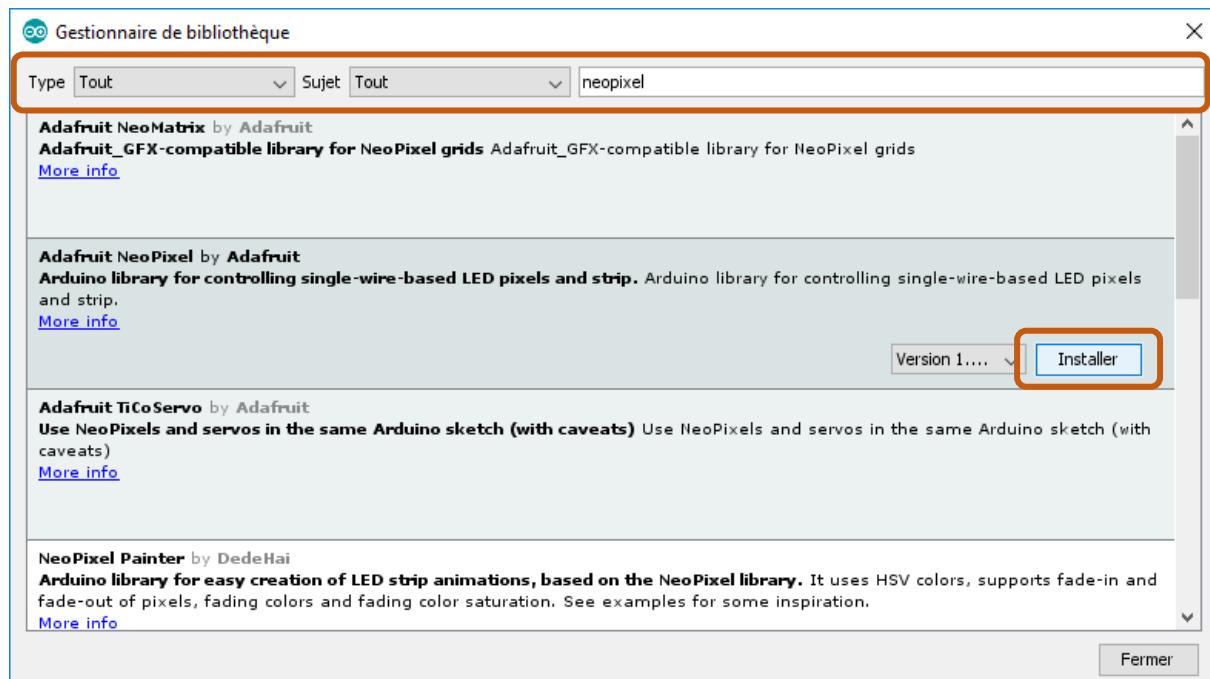
Installation de la bibliothèque

Pour pouvoir utiliser ce module, la bibliothèque *NeoPixel* de chez Adafruit doit être installée :

Ouvrez l'IDE Arduino et aller dans *Croquis* → *Inclure une bibliothèque* → Gérer les bibliothèques



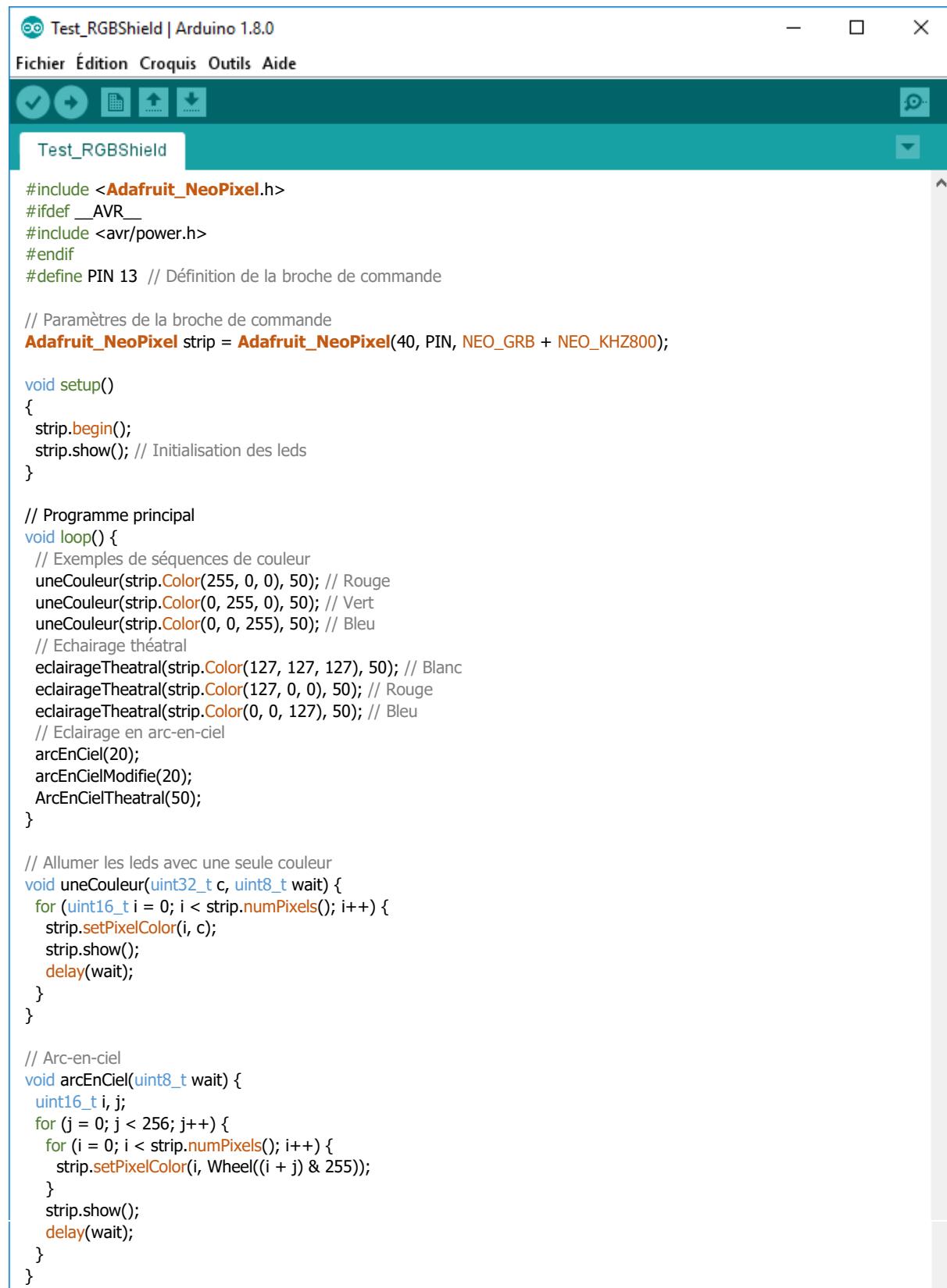
Dans le gestionnaire de bibliothèques, recherchez « neopixel » et installez *Adafruit NeoPixel*:



Fermez la fenêtre une fois l'installation terminée.

Exemple de programme

L'exemple de code ci-dessous allume les leds suivant différentes séquences :



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** Test_RGBShield | Arduino 1.8.0
- Menu Bar:** Fichier Édition Croquis Outils Aide
- Toolbar:** Includes icons for Open, Save, Upload, and Download.
- Sketch Name:** Test_RGBShield
- Code Area:** Displays the C++ code for the RGB shield example. The code includes imports for Adafruit_NeoPixel.h, defines a pin, initializes a NeoPixel strip, and implements setup() and loop() functions for various lighting effects like color sequences and rainbow patterns.

```
#include <Adafruit_NeoPixel.h>
#ifndef __AVR__
#include <avr/power.h>
#endif
#define PIN 13 // Définition de la broche de commande

// Paramètres de la broche de commande
Adafruit_NeoPixel strip = Adafruit_NeoPixel(40, PIN, NEO_GRB + NEO_KHZ800);

void setup()
{
    strip.begin();
    strip.show(); // Initialisation des leds
}

// Programme principal
void loop() {
    // Exemples de séquences de couleur
    uneCouleur(strip.Color(255, 0, 0), 50); // Rouge
    uneCouleur(strip.Color(0, 255, 0), 50); // Vert
    uneCouleur(strip.Color(0, 0, 255), 50); // Bleu
    // Eclairage théâtral
    eclairageTheatral(strip.Color(127, 127, 127), 50); // Blanc
    eclairageTheatral(strip.Color(127, 0, 0), 50); // Rouge
    eclairageTheatral(strip.Color(0, 0, 127), 50); // Bleu
    // Eclairage en arc-en-ciel
    arcEnCiel(20);
    arcEnCielModifie(20);
    ArcEnCielTheatral(50);
}

// Allumer les leds avec une seule couleur
void uneCouleur(uint32_t c, uint8_t wait) {
    for (uint16_t i = 0; i < strip.numPixels(); i++) {
        strip.setPixelColor(i, c);
        strip.show();
        delay(wait);
    }
}

// Arc-en-ciel
void arcEnCiel(uint8_t wait) {
    uint16_t i, j;
    for (j = 0; j < 256; j++) {
        for (i = 0; i < strip.numPixels(); i++) {
            strip.setPixelColor(i, Wheel((i + j) & 255));
        }
        strip.show();
        delay(wait);
    }
}
```

```
// Arc-en-ciel légèrement modifié
void arcEnCielModifie(uint8_t wait) {
    uint16_t i, j;
    for (j = 0; j < 256 * 5; j++) {
        for (i = 0; i < strip.numPixels(); i++) {
            strip.setPixelColor(i, Wheel(((i * 256 / strip.numPixels()) + j) & 255));
        }
        strip.show();
        delay(wait);
    }
}

// Éclairage de théâtre
void eclairageTheatral(uint32_t c, uint8_t wait) {
    for (int j = 0; j < 10; j++) { // 10 répétitions
        for (int q = 0; q < 3; q++) {
            for (int i = 0; i < strip.numPixels(); i = i + 3) {
                strip.setPixelColor(i + q, c); // Allumer les 3eme leds
            }
            strip.show();
            delay(wait);
            for (int i = 0; i < strip.numPixels(); i = i + 3) {
                strip.setPixelColor(i + q, 0); // Eteindre les 3eme leds
            }
        }
    }
}

// Éclairage de théâtre avec effet arc en ciel
void ArcEnCielTheatral(uint8_t wait) {
    for (int j = 0; j < 256; j++) { // Toutes les couleurs
        for (int q = 0; q < 3; q++) {
            for (int i = 0; i < strip.numPixels(); i = i + 3) {
                strip.setPixelColor(i + q, Wheel( (i + j) % 255)); // Allumer les 3eme leds
            }
            strip.show();
            delay(wait);
            for (int i = 0; i < strip.numPixels(); i = i + 3) {
                strip.setPixelColor(i + q, 0); // Eteindre les 3eme leds
            }
        }
    }
}

uint32_t Wheel(byte WheelPos) {
    WheelPos = 255 - WheelPos;
    if (WheelPos < 85) {
        return strip.Color(255 - WheelPos * 3, 0, WheelPos * 3);
    }
    if (WheelPos < 170) {
        WheelPos -= 85;
        return strip.Color(0, WheelPos * 3, 255 - WheelPos * 3);
    }
    WheelPos -= 170;
    return strip.Color(WheelPos * 3, 255 - WheelPos * 3, 0);
}
```

GO TRONIC

ROBOTIQUE ET COMPOSANTS ÉLECTRONIQUES

Si vous rencontrez des problèmes, merci de nous contacter par courriel à :

sav@gotronic.fr



Coordonnées du fabricant :

joy-it[®]

service@joy-it.net

+49 (0)2845 9360 – 50