Librairies disponibles pour piloter un écran OLED SSD1306

Pour gérer l'affichage sur votre écran OLED, il y a plusieurs librairies Open Source à notre disposition. Pour cet article, je vais vous présenter les deux principales. La librairie développée par Adafruit et celle de Sparkfun.

Librairies **Adafruit_GFX et** Adafruit_SSD1306

Adafruit a développé une librairie très puissante qui va nous permettre de gérer l'affichage de notre mini écran mais aussi de tracer plein de chose très facilement grâce à la librairie dédiée, GFX Library. Vous pouvez récupérer la librairie Adafruit SSD1306 sur le github https://github.com/adafruit/Adafruit_SSD1306 et la librairie GFX ici https://github.com/adafruit/Adafruit-GFX-Library/archive/master.zip.

> *Attention. Il est nécessaire d'inclure les 2 librairies dans votre projet. Pour gagner quelques précieux octets, vous pouvez utiliser la librairie Micro_OLED de Sparkfun décrite dans le prochain paragraphe.*

Liste des fonctions de la librairie disponibles

Fonctions de la librairie Adafruit_SSD1306	
Adafruit_SSD1306 display(OLED_RESET)	Initialise l'objet display(Pin pour le reset)
display()	Actualise l'affichage
clearDisplay()	Efface l'écran et le buffer
invertDisplay(bool)	Inverse l'affichage (true ou false)

Fonctions Adafruit_GFX	
drawPixel(uint16_t x, uint16_t y, uint16_t color)	Dessine un pixel en X,Y de la couleur color
drawLine(uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1, uint16_t color)	Dessine une ligne de X1,Y1 à x2,Y2 de la couleur color
drawFastVLine(uint16_t x0, uint16_t y0, uint16_t length, uint16_t color) drawFastHLine(uin86_t x0, uin86_t y0, uint8_t length, uint16_t color);	Tracé optimisé de lignes verticales et horizontales
drawRect(uint16_t x0, uint16_t y0, uint16_t w, uint16_t h, uint16_t color) fillRect(uint16_t x0, uint16_t y0, uint16_t w, uint16_t h, uint16_t color)	Dessine un rectangle depuis X,Y de largeur w et hauteur h Idem mais plein

drawCircle(uint16_t x0, uint16_t y0, uint16_t r, uint16_t color) fillCircle(uint16_t x0, uint16_t y0, uint16_t r, uint16_t color)	Dessine un cercle de centre X,Y et de rayon r Idem mais le cercle est plein
drawRoundRect(uint16_t x0, uint16_t y0, uint16_t w, uint16_t h, uint16_t radius, uint16_t color) fillRoundRect(uint16_t x0, uint16_t y0, uint16_t w, uint16_t h, uint16_t radius, uint16_t color)	Idem rectangle mais avec un arrondi de rayon raduis aux angles Idem mais le rectangle est plein
drawTriangle(uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1, uint16_t x2, uint16_t y2, uint16_t color) fillTriangle(uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1, uint16_t x2, uint16_t y2, uint16_t color)	Dessine un triangle en spécifiant les coordonnées de chaque sommets (x0,y1), (x2,y2), (x3,y3) Idem mais le triangle est plein
drawChar(uint16_t x, uint16_t y, char c, uint16_t color, uint16_t bg, uint8_t size)	Dessine un caractère en x,y
drawBitmap(int16_t x, int16_t y, uint8_t *bitmap, int16_t w, int16_t h, uint16_t color)	Affiche un bitmap en x,y de largeur w et hauteur t
fillScreen(uint16_t color);	Colorie entièrement l'écran dans la couleur spécifiée
setRotation(uint8_t rotation)	Rotation de l'affichage : 0 -> 0°, 1 -> 90°, 2 -> 180°, 3 -> 270°

Le paramètre color permettant de définir la couleur d'affichage uniquement sur les écrans couleur est disponible dans toutes les fonctions graphiques de la librairie GFX. Pour plus d'informations, vous pouvez consulter ce document de formation en anglais.
La librairie **Adafruit_GFX** est utilisée par d'autres librairies dédiées à chaque (microcontroleur) écran.

Afficher du texte

L'affichage d'un texte demande un peu plus de travail. Il est nécessaire de modifier les paramètres d'affichage paramètre par paramètre. Voici un petit exemple pour afficher Hello Word en 0,0:

```
display.setTextSize(1);                 // setTextSize applique un facteur d'échelle qui permet d'agrandir ou réduire la font

display.setTextColor(WHITE);            // Définir la couleur du texte
```

```
display.setCursor(0,0);             // Positionner le curseur en x=0, y=0

display.println(« Hello, world! »);          // un println comme pour écrire sur le port série

display.setTextColor(BLACK, WHITE);         // On inverse les couleurs, le fond devient noir

display.println(« Hello, world! »);

// Vous pouvez changer à la volée de Font (Librairies disponibles pour piloter un écran OLED SSD1306.docxpour cela vous devez la déclarer comme une librairie en début de projet, par exemple #include <Fonts/FreeMono9pt7b.h>)

display.setFont(&FreeMono9pt7b);

display.setTextColor(WHITE);

display.println(« Hello, world! »);

display.setFont();                          // Pour revenir à la Font par défaut
```

Par contre, vous pouvez oublier les accents, ils ne sont pas gérés dans les Fonts disponibles.