

Microcontrôleur

Projet : OLEDBITMAP – pilotage d'un écran OLED par bus SPI

Ce projet a pour objectif de concevoir une bibliothèque de fonctions permettant la gestion de l'écran PmodOLEDRgb de chez Digilent. L'intérêt est de permettre à un utilisateur d'utiliser cet écran avec un minimum d'investissement en apprentissage.

1) Ressources

Le Pmod OLEDRgb est particulièrement bien documenté sur le centre de ressources de Digilent : <https://digilent.com/reference/pmod/pmodoledrgb/reference-manual>

Le composant principal est le SSD1331 donc la datasheet est disponible là : https://cdn-shop.adafruit.com/datasheets/SSD1331_1.2.pdf

Le module communique sur le bus SPI, mais a besoin de quelques autres broches de contrôle. Ces dernières seront manipulées comme des I/Os classiques. Les deux phases les plus délicates du projet sont les suivantes :

- définir la séquence d'allumage de l'écran
- utiliser les ressources matérielles pour limiter l'usage du processeur

2) Fonctionnalités attendues

a) fonctions de base

Les fonctions indispensables à fournir sont les suivantes :

- OLEDRgb_init : initialisation de l'écran et mise en place de l'environnement (selon l'avancement du projet, cette fonction peut récupérer des paramètres, ces derniers sont laissés libres)
- mise à disposition d'un pointeur vers la mémoire graphique (tableau représentant l'image affichée à l'écran)

Par défaut, l'affichage sera en couleurs codées sur 16 bits. L'utilisation de modes différents sera bienvenu pour éventuellement réduire la taille de la mémoire graphique. Idée de modes utiles :

- 65536 couleurs RGB (mode de base)
- 256 couleurs RGB
- 256 couleurs indexées
- 16 couleurs indexées (chaque octet contient la valeur de deux pixels)
- 4 couleurs indexées (chaque octet contient la valeur de 4 pixels)
- 2 couleurs indexées (1 bit par pixel)

b) fonctions secondaires

Les fonctions suivantes sont bienvenues pour permettre l'usage par un utilisateur débutant :

- `set_pix_val` : définit la couleur d'un pixel
- `set_palette` : définit une éventuelle table de correspondance pour les couleurs indexées

3) Performances

La performance de l'ensemble de l'architecture est un critère important. Notamment, la fourniture de fonctions bloquantes est un minimum, mais dans l'idéal, l'usage du processeur doit être réduit au strict nécessaire.

De même, l'écriture d'un fichier de bibliothèque `.c` et de son fichier d'en-tête `.h` est un plus intéressant (tant pédagogiquement que techniquement).

Un bilan des ressources matérielles utilisées doit être tenu à jour.