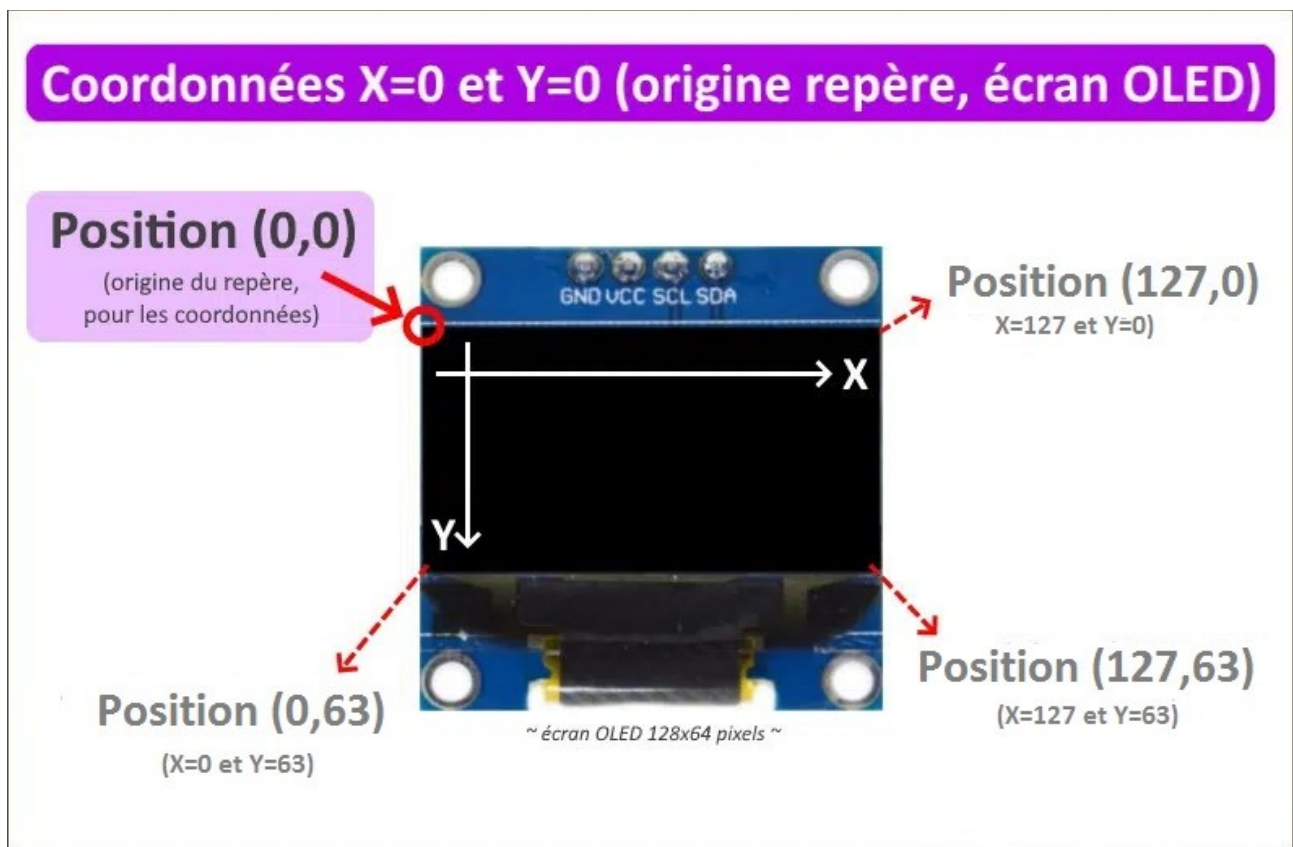


TD Programmation Orienté Objet en C++

Afficheur OLED

Présentation

Les écrans OLED sont des **écrans graphiques**, monochromes ou multi-couleurs, permettant d'afficher du texte, des images, et des figures graphiques. Ces écrans intègrent un contrôleur, permettant de faire l'interface entre l'écran oled et le microcontrôleur ESP32 via un bus i2c.



Préparation du projet

Réalisez un projet nommé **Afficheur** avec **PlatformIO** et votre environnement de développement.

Pour cela, créez un répertoire portant le nom de votre projet **Afficheur**, ouvrez une console dans ce répertoire et lancez la commande suivante pour créer le projet :

```
pio project init --ide netbeans --board lolin32
```

Ouvrez ensuite le projet avec NetBeans. dans les propriétés du projet (Clic droit sur le projet), dans la catégorie Run modifier la commande de run avec la ligne suivante :

```
pio run --target upload
```

Programmation

Pour piloter un écran OLED il existe des librairies toutes faites.

Installation de la librairie avec platformIO

```
pio pkg install --library "thingpulse/ESP8266 and ESP32 OLED driver for SSD1306 displays@^4.3.0"
```

Il suffit simplement d'insérer les lignes suivantes dans le haut de votre programme pour créer un objet **display**

```
#include <SSD1306Wire.h>

SSD1306Wire display(0x3c, SDA , SCL, GEOMETRY_128_64);
```

Une chose importante à savoir, c'est que **l'affichage ne sera pas mis à jour à chaque fois que vous ajouterez du texte, ou élément graphique**. En fait, vous ne ferez que mettre à jour la mémoire tampon, et il sera nécessaire d'envoyer toute cette mémoire tampon à l'écran, pour ce faire vous devrez utiliser la méthode **display()**

```
#include <Arduino.h>
#include <SSD1306Wire.h>

SSD1306Wire display(0x3c, SDA , SCL, GEOMETRY_128_64);

void setup(){

    display.init();
    display.setFont(ArialMT_Plain_24);
    display.flipScreenVertically();
    display.drawString(0, 0, "Test Ecran");
    display.drawString(0, 32, "Ligne 2");
    display.display();
}

void loop(){
}
```