

## 1- Création du projet

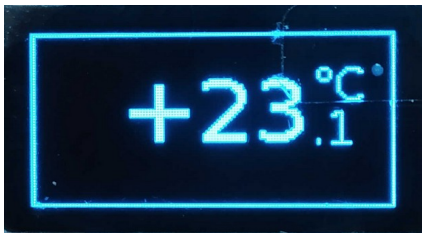
Pour commencer, créez un projet intitulé 'TestAfficheur' en utilisant **PlatformIO** pour l'environnement NetBeans. Une fois le projet ouvert dans **NetBeans**, créez les fichiers suivants : **main.cpp**, **afficheur.cpp**, **afficheur.h** et **fonts.h**. Le fichier '**main.cpp**' contiendra le programme de test de l'afficheur, tandis que les fichiers '**afficheur.cpp**' et '**afficheur.h**' regrouperont tous les éléments tels que les fonctions, les constantes nécessaires à l'affichage. Le dernier fichier, '**fonts.h**', stockera les polices de caractères utilisées.

Vous ajouterez également le fichier **esp32\_snir.h** disponible en ressource à votre projet.

Dans le fichier '**afficheur.cpp**', assurez-vous de déclarer en tant que variable globale une instance de la classe **SSD1306** nommée '**AfficheurOLED**', pour une utilisation dans l'ensemble du fichier.

## 2- Création de nouvelles polices de caractères

Pour réaliser la figure suivante :



À partir du lien <https://oledisplay.squix.ch/> vous devez créer de nouvelles polices de caractères.

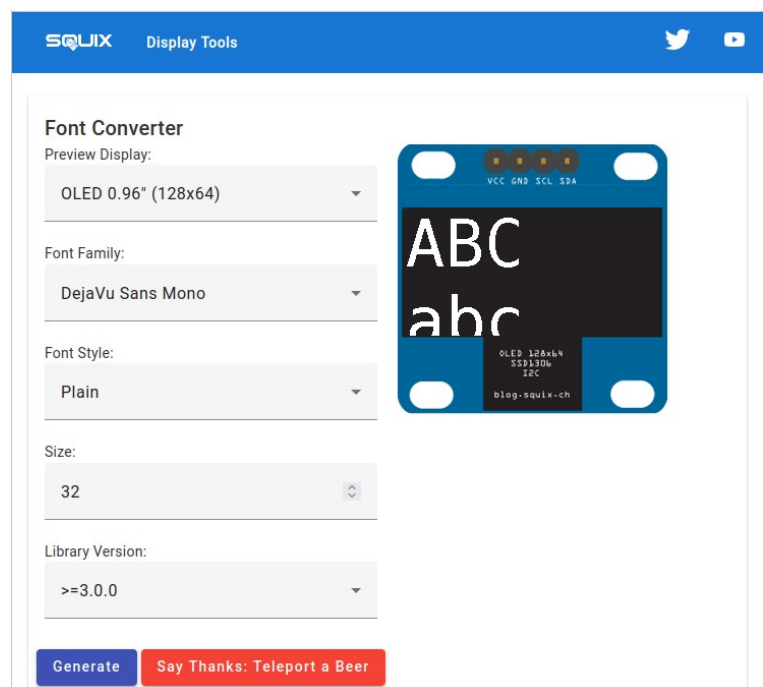
Réalisez dans un premier temps la police **DejaVu\_Sans\_Mono\_32**

Cliquez sur **Generate** et copiez l'ensemble du code généré dans le fichier **fonts.h**.

Vous remplacerez **char** par **uint8\_t** dans le fichier.

Vous ajouterez également l'inclusion du fichier **Arduino.h**

Vous procéderez de la même manière pour la police : **DejaVu\_Sans\_Mono\_16**

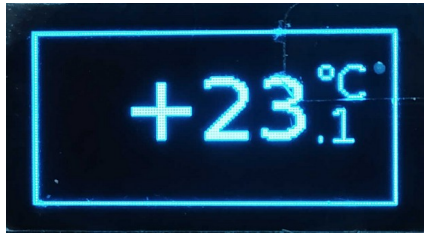


```
#include <Arduino.h>

const uint8_t DejaVu_Sans_Mono_32[] PROGMEM = {
    0x13, // Width: 19
    0x26, // Height: 38
    0x20, // First Char: 32
    0xE0, // Numbers of Chars: 224
    ...
}
```

### 3- Description de l'affichage attendu

L'application à réaliser consiste, dans un premier temps, à mettre au point une fonction chargée d'afficher la température en degré Celsius sur l'afficheur OLED SSD1306 comme le montre la figure ci-dessous :



La partie entière de la température utilise la police **DejaVu\_Sans\_32** avec affichage systématique du signe.

L'unité « °C » et les dixièmes de degrés utilisent la police **DejaVu\_Sans\_16**.

### 4- Création des fonctions

Une première fonction est chargée d'initialiser l'afficheur pour notre application en tenant compte de l'orientation de l'afficheur.

Codez la fonction dont le prototype est le suivant ; **void InitialiserAfficheur(void)** ;

Codez la fonction **AfficherTemperature()**. Cette fonction reçoit 2 paramètres : le premier **\_temperature**, un entier représente la valeur entière de la température, le second **\_dixiemes** les dixièmes de degré.

Voici les étapes à suivre pour l'affichage :

#### Étape 1

- Effacer l'écran
- Dessiner dans un premier temps le rectangle, il doit occuper la largeur et la hauteur de l'écran de l'afficheur.

#### Étape 2

- Sélectionner la police **DejaVu\_Sans\_Mono\_16**.
- Fixer l'alignement du texte à droite.
- Écrire les unités à partir des coordonnées du point de référence (120,10).
- Afficher les dixièmes de degrés à partir de la même abscisse, l'ordonnée est augmenté de la hauteur de la police de caractères.
- Calculer la largeur du texte en pixels de « °C » avec la police actuelle.

#### Étape 3

- Sélectionner la police **DejaVu\_Sans\_Mono\_32**.
- Écrire la valeur entière de la température en tenant compte du signe. La position est calculée à partir de la position point de référence, de la largeur du texte précédemment écrit.

### 5- Test de la fonction

A l'aide des boutons-poussoirs présents sur votre carte ESP32-SNIR, vous allez réaliser le test de votre fonction **AfficherTemperature()**.

En fonction de la position de l'interrupteur **SW**

- **BP1** → augmente la partie entière d'une unité, ou augmente les dixièmes de degré [0 → 9]
- **BP2** → décrémente la partie entière d'une unité, ou décrémente les dixièmes de degré [9 → 0]

Appellez la fonction **InitialiserAfficheur()** dans la fonction **setup()** puis codez le corps de la fonction **loop()**.

Attendre 200 ms avant chaque scrutation des poussoirs et de l'interrupteur.

Effectuer les corrections si nécessaire dans votre fonction pour que l'affichage soit satisfaisant.