



Afficher une image sur un écran ST7789

Les écrans ST7789 ne coûtent rien (3€) et permettent d'afficher des images et des photos. Même sans carte SD.

 Difficulté Moyen

 Durée 2 heure(s)

 Catégories Électronique

 Coût 7 EUR (€)

Sommaire

Introduction

Video d'introduction

Étape 1 - Installer / Paramétrer TFT_eSPI

Étape 2 - Branchement de l'écran

Étape 3 - Essayer l'écran

Étape 4 - Convertir l'image en code

Étape 5 - Fabriquer notre mini écran

Notes et références

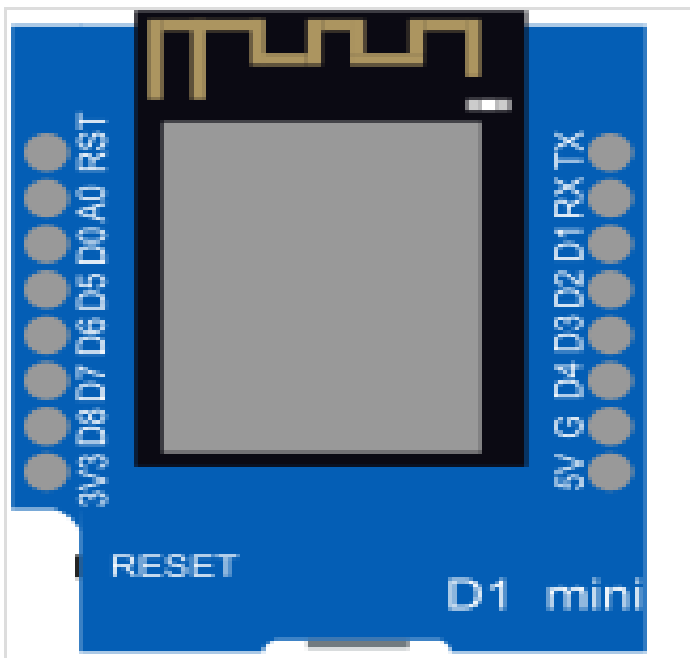
Commentaires

Introduction

J'ai acheté cet écran (ST7789) pour l'utiliser avec un ESP8266 afin de créer un afficheur connecté low cost.

Malheureusement, j'ai eu beaucoup de mal à trouver la bonne bibliothèque pour l'exploiter, mais avec mon aide vous devriez y arriver sans problème !

Voyons comment afficher une image sans avoir à utiliser une carte SD.

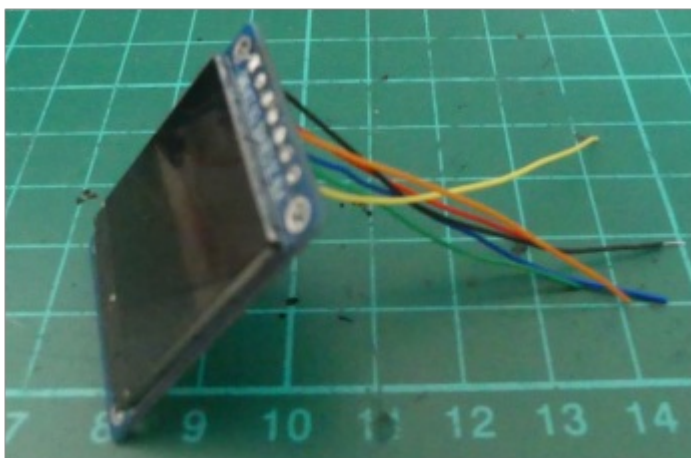


Matériaux

- Wemos Mini Pro (clone) ou autre ESP8266
- ST7789 240x240

Outils

- Fer à souder
- Imprimante 3D



<https://www.thingiverse.com/thing:3509423>

https://github.com/maditnerd/st7789_bitmap

Étape 1 - Installer / Paramétrer TFT_eSPI

Après avoir essayé la bibliothèque d'Adafruit (`adafruit_st7789`) qui ne **marche pas** avec mon écran et une bibliothèque non disponible dans le gestionnaire de bibliothèque (`arduino_st7789`) qui était **instable** et lente c'est sans équivoque **TFT_eSPI** qui **marche le mieux**.

TFT_eSPI doit être paramétré pour fonctionner avec cet écran.

Pour cela nous devons modifier `User_Setup.h` à l'intérieur de la bibliothèque, une fois que cela est fait, la bibliothèque marchera avec tous les croquis et exemples **sans modification**.

Télécharger mon code ici : https://github.com/maditnerd/st7789_bitmap

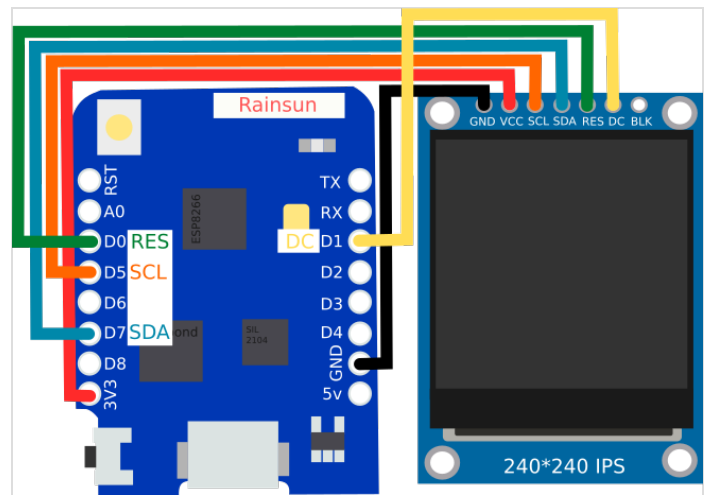
Dans le dossier `User_Setup`, vous trouverez le fichier configuré pour un ST7789.

- Installer **TFT_eSPI** à l'aide du **Gestionnaire de bibliothèque**
- Aller dans le dossier **libraries** (sous Windows : **Documents/Arduino/libraries**)
- Aller dans le dossier **TFT_eSPI**
- Remplacer `User_Setup.h` avec ma version.

Étape 2 - Branchement de l'écran

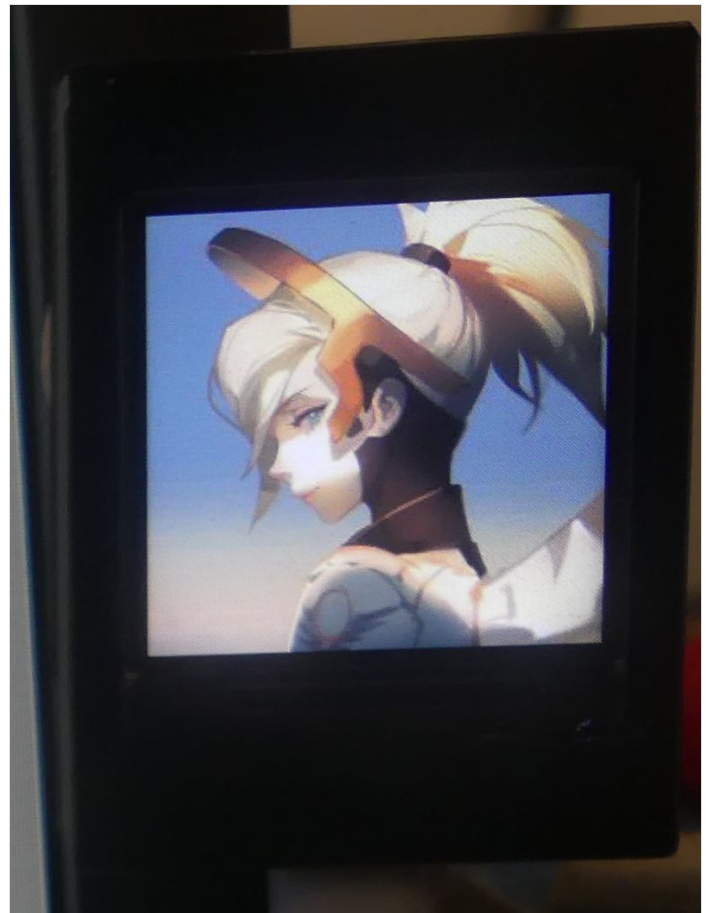
L'écran est branché en SPI sur l'esp8266

- D5 --> SCL (Hardware SPI)
- D7 --> SDA (Hardware SPI)
- D0 --> RES (n'importe quelle broche)
- D1 --> DC (n'importe quelle broche)



Étape 3 - Essayer l'écran

Vous pouvez essayer de faire fonctionner l'écran avec les **exemples disponibles** dans `tft_eSPI` ou **utiliser mon code** pour afficher une image.



Étape 4 - Convertir l'image en code

Dans mon code (https://github.com/maditnerd/st7789_bitmap), l'image est sauvegardée dans **bitmap.h**.

Pour pouvoir afficher une image, Il nous faut la **convertir en code** et le **copier** dans un **tableau**.

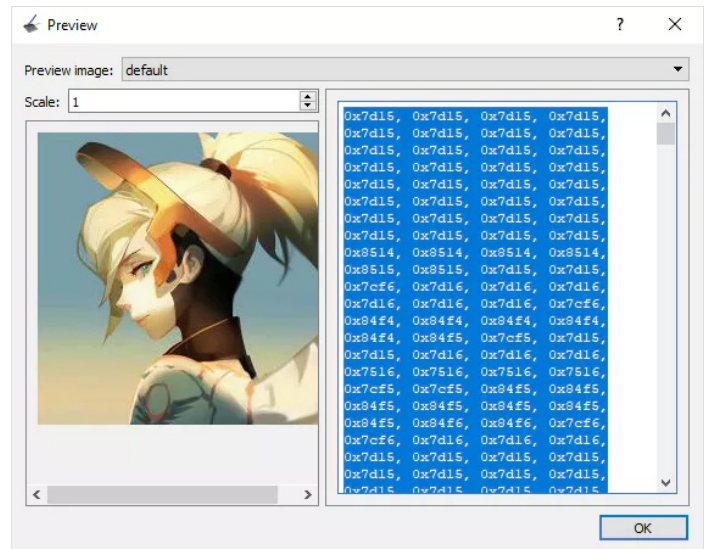
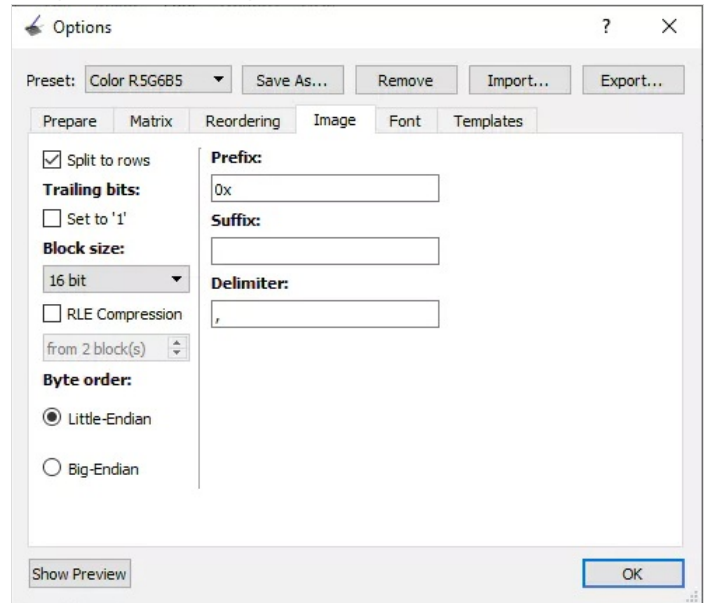
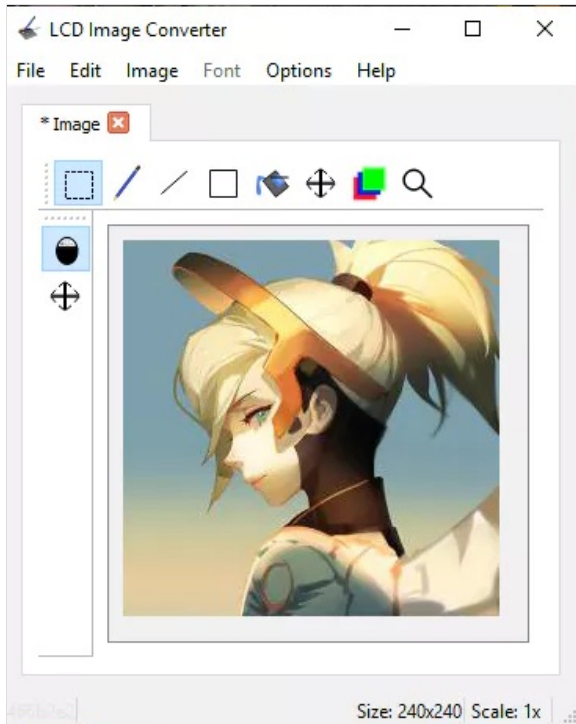
Pour cela, nous allons utiliser **LCD image converter**, vous pouvez trouver ce programme ici: <https://sourceforge.net/projects/lcd-image-converter/>

- Redimensionner l'image à la taille de votre écran (240x240)
- Cliquer sur **New Image**
- Aller sur **Image / Import** et sélectionner votre image
- Aller dans **Options / Conversion**
- Choisissez le **Preset : Color R5G6B5**
- Aller dans l'onglet **Image**
- Changer **Block Size** à **16bit**
- Cliquer sur **Show Preview**
- Copier le code
- Copier-coller le **dans le tableau**

```
const uint16_t mercy [] PROGMEM = { // --> PASTE HERE};
```

Et voilà, téléverser le code et vous devriez voir l'image s'affichait sur l'écran.

Note: Il y a un exemple dans TFT_eSPI pour décoder des fichiers JPEG

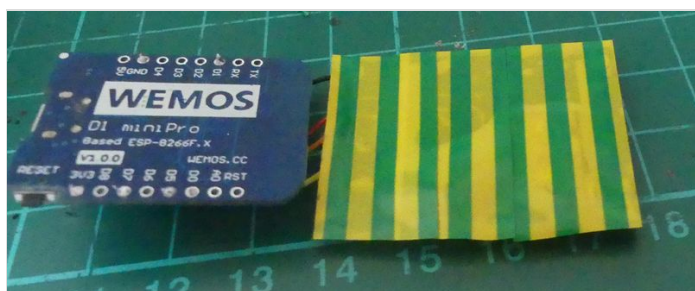
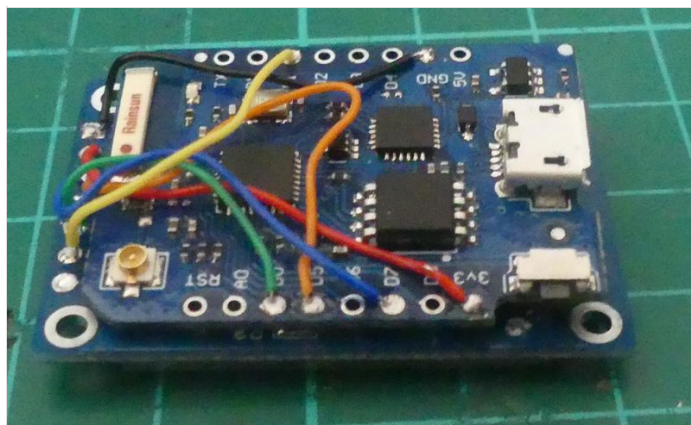
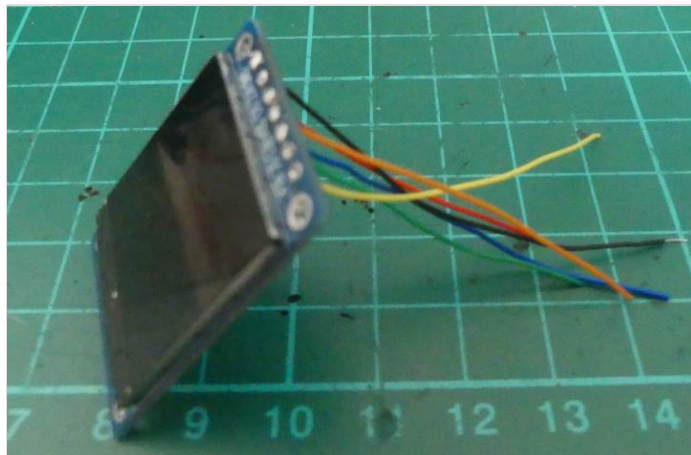


Étape 5 - Fabriquer notre mini écran

Vous pouvez trouver les fichiers STL ici : <https://www.thingiverse.com/thing:3509423>

Merci à *Olivier Sarrailh* pour le boîtier.

- Afin de réduire au plus la taille, j'ai retiré les broches de l'écran.
- Je l'ai est remplacé par du **Wrapping Cable 30 AWG**
- Après ça j'ai soudé l'écran à mon clone de wemos mini pro
- Ensuite, j'ai ajouté du **ruban électrique** pour éviter les courts-circuits
- Pour finir, j'ai **imprimé le boîtier**



Notes et références

Suivez moi sur twitter si vous voulez être au courant des prochains tutoriels en avance : <https://twitter.com/m4dnerd>

