

Ecran OLED

Par Benoit THOMAS

L'écran OLED permet d'afficher tous types d'information, images ou textes. Il existe des écrans OLED adaptés pour une utilisation avec un arduino. Nous allons voir dans ce tutoriel comment les utiliser. A travers ce tutoriel l'objectif est de savoir utiliser les écrans OLED pour afficher du texte ou un logo.

Afin de comprendre le fonctionnement des écrans OLED nous allons étudier 2 exemples. Dans le premier exemple nous allons voir comment afficher du texte, et ensuite nous verrons comment afficher une image/logo.

Pour ce tutoriel, il faut installer plusieurs librairies : selon votre écran soit la librairie Adafruit_SSD1306.h, soit Adafruit_SH1106.h. Egalement la librairie Adafruit_GFX.h sera nécessaire dans les deux cas.

Matériel :

Pour ce tutoriel, nous allons utiliser un arduino UNO, un écran OLED et des câbles.

Fonctionnement :

Nous allons écrire un code pour afficher ce que l'on souhaite, puis nous allons envoyer le code sur l'arduino. Enfin l'arduino va communiquer par 2 pins pour afficher l'information souhaité sur l'écran.

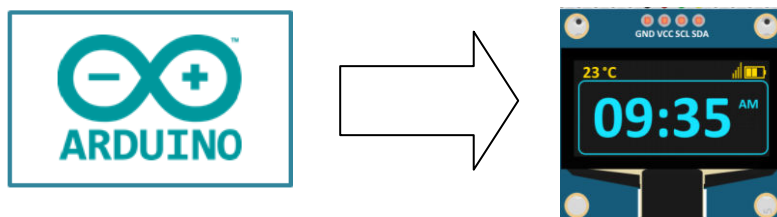
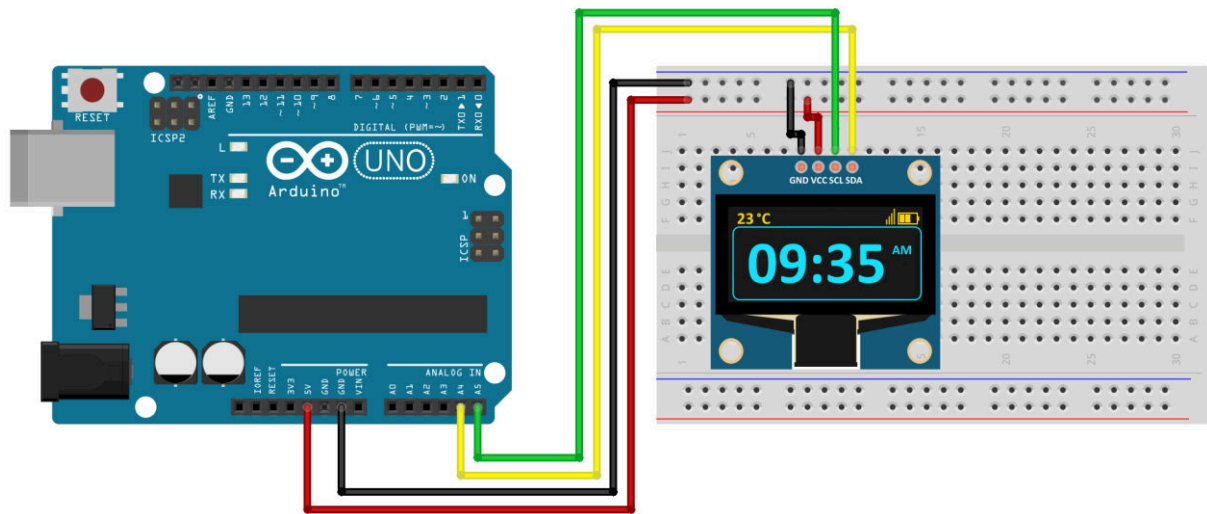


Schéma et câblage:

Pour commencer, nous allons câbler comme ci-dessous :



Il est également possible de relier SCL avec SCL et SDA avec SDA.

1- Afficher du texte

Pour afficher du texte c'est relativement simple, il faut utiliser la commande :

```
display.print(« texte à afficher »);
```

Cependant il est nécessaire de configurer l'écran avant de pouvoir afficher le texte. Pour cela il faut mettre toutes les commandes expliquées ci-dessous :

<code>display.clearDisplay();</code>	Pour vider l'écran
<code>display.setTextSize(9);</code>	Choisir la taille du texte (ici 9 Pixels)
<code>display.setTextColor(WHITE);</code>	Choisir la couleur du texte
<code>display.setCursor(0,0);</code>	Mettre le curseur en haut à gauche
<code>display.print(i);</code>	Afficher la valeur de i
<code>display.display();</code>	Mettre à jour l'écran et afficher le texte

Pour mettre en application l'affichage de texte nous allons faire un exemple très simple : afficher un chronomètre de 60 secondes.

Pour cela il suffit de créer une boucle for et à chaque seconde changer l'affichage de n+1. Voici le code (fichier : « aff_texte »):

```

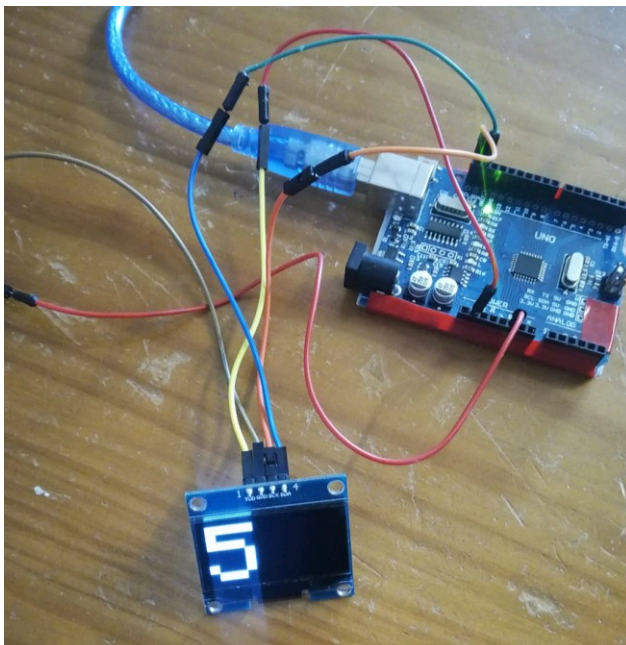
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SH1106.h>
#define OLED_RESET 4
Adafruit_SH1106 display(OLED_RESET);

//----- initialisation de l'ecran -----//

void setup()  {
  Serial.begin(9600);
  display.begin(SH1106_SWITCHCAPVCC, 0x3C);
}
//-----Affiche le chronometre-----//
void loop() {
  display.clearDisplay();           // vide l'ecran
  display.setTextSize(9);          //modifie la taille du texte
  display.setTextColor(WHITE);     // choisir la couleur

  for (int i=1;i<60; i++){         // boucle for
    display.setCursor(0,0);        //mettre le curseur en haut à gauche
    display.print(i);              //afficher la valeur de i (les secondes)
    display.display();             //mettre à jour l'écran et afficher le texte
    delay(1000);                  //attendre 1000 millisecondes càd 1 seconde
    display.clearDisplay();        //vider l'écran
    display.display();             //mettre à jour l'écran
  }
}

```



2- Afficher un logo

Pour afficher une image ou un logo, il faut convertir l'image en un code compréhensible pour l'arduino afin qu'il affiche l'image.

Pour cela nous allons utiliser le site suivant : <https://diyusthad.com/image2cpp> qui permet de convertir une image en format Arduino.

1. Select image



Ajouter l'image/logo que vous souhaitez ajouter.

Il faut ensuite mettre l'image avec une définition de 128 X 64 pixels. C'est la définition « classique » des écrans OLED, mais si votre écran est différent il faut mettre les dimensions de votre écran.

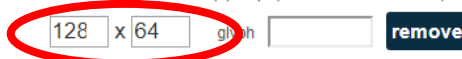
Les autres paramètres sont optionnels.

Vous pouvez ensuite voir si le rendu vous satisfait.

2. Image Settings

Canvas size/s:

binary logo 128x64 (1).bmp (file resolution: 128 x 64)



Background color:

White Black

Invert image colors

Brightness threshold:

0 - 255; pixels with brightness above become white, below become black.

Scaling

scale to fit, keeping proportions ▼

Center:

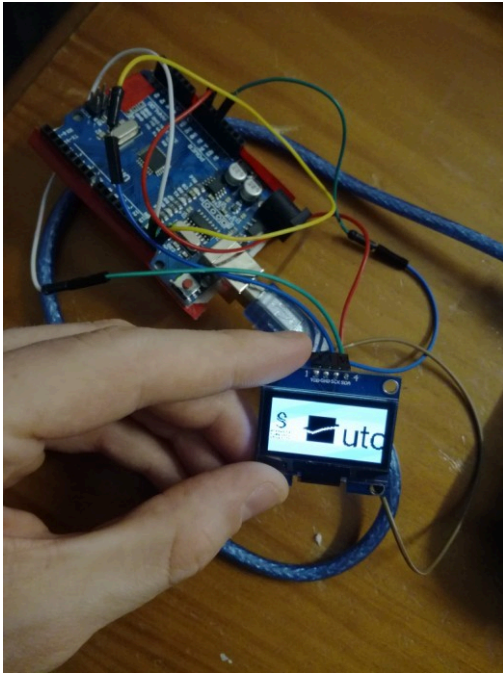
horizontally vertically

NOTE: Centering the image only works when using a canvas larger than the selected image.


```
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff  
};
```

```
void setup() {  
  Serial.begin(9600);  
  display.begin(SH1106_SWITCHCAPVCC, 0x3C);  
  display.clearDisplay();  
  
  // -----Affiche le logo-----//  
  
  display.drawBitmap(1, 1, logo16_glcd_bmp, 128, 64, 1); //affiche le logo16 à la position 1, 1 au format 126 x 64 px  
  display.display(); //met à jour l'ecran, affiche l'image  
  
}  
  
void loop () {}
```

On obtient alors :



Pour aller plus loin :

Il est possible de créer des fonctions afin d'afficher des animations. Par exemple dans le code d'exemple de la librairie, il a plusieurs fonctions qui permettent de créer des animations semblables à un écran de veille.

De plus il est également possible de créer un menu est de mettre en variable certaines informations, pour les faire varier en fonction du temps.

Sources :

DIY USTHAD, Displaying Images in OLED Display [En ligne],
<https://www.youtube.com/watch?v=H174EUG2kLO>, [Consulté le : 15/12/2019]

DIY USTHAD, Displaying Images in OLED Display |BLOG| [En ligne],
, <https://diyusthad.com/2019/02/display-images-in-oled-display.html> [Consulté le : 15/12/2019]

DIY USTHAD, Interfacing OLED 128x64 I2C with Arduino Uno [En ligne],
<https://www.youtube.com/watch?v=9xZmWZ4Hh6M&feature=youtu.be>, [Consulté le :
15/12/2019]

Crazy Couple, Tutorial on I2C OLED Display with Arduino/NodeMCU [En ligne],
https://www.youtube.com/watch?v=e_0HJY0ulo&t=5s, [Consulté le : 17/12/2019]