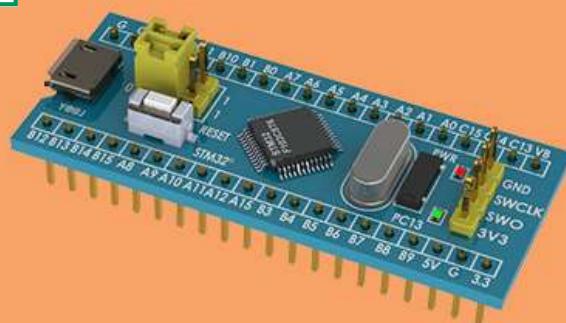


**LẬP TRÌNH STM32**

Lập trình STM32 với Oled LCD SSD1306

POSTED ON 01/08/2021 BY KHUÊ NGUYỄN

01
Th8**Khuê Nguyễn Creator**

Lập trình STM32 điều khiển LCD SSD1306 chuẩn I2C

Màn hình Oled LCD SSD1306 được sử dụng rất nhiều trong các ứng dụng nhúng. Hiển thị đẹp mắt, tiết kiệm năng lượng và hỗ trợ chuẩn kết nối I2C rất phù hợp với những ứng dụng nhỏ.

Bài này nằm trong [Serie Học STM32 từ A tới Z](#)

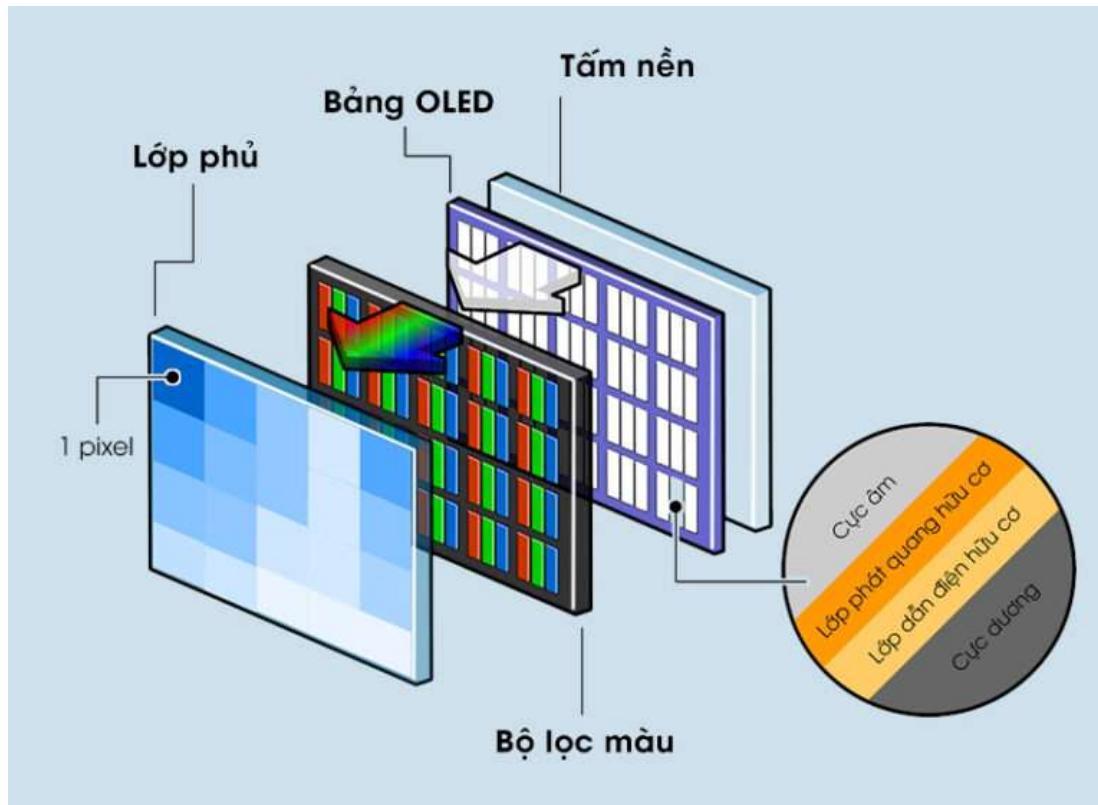
[Mục Lục](#)



1. Màn hình Oled là gì?
 2. Thông số cơ bản về LCD Oled SSD1306
 3. Cách điều khiển LCD Oled SSD1306
 - 3.1. Sơ đồ khối
 - 3.2. Các giao thức điều khiển
 - 3.3. Giao thức I2C điều khiển ssd1306
 - 3.4. Display RAM và cách hiển thị của ssd1306
 - 3.5. Các lệnh và khởi tạo oled ssd1306
 4. Lập trình STM32 điều khiển LCD ssd1306
 - 4.1. Cấu hình trên cubemx
 - 4.2. Download thư viện và lập trình
 - 4.3. Hiển thị văn bản và các hình cơ bản trong LCD
 - 4.4. Hiển thị ảnh với ssd1306
 - 4.5. Hiển thị ảnh động GIF với ssd1306
 - 4.6. Cách chuyển ảnh sang mảng để hiển thị
 5. Kết
- 5.1. Related posts:

Màn hình Oled là gì?

Màn hình OLED (**O**rganic **L**ight **ED**iodes) được biết đến là một loại diode phát sáng (LED). Sử dụng một lớp phát xạ điện quang (màng thuốc hoặc film), với vật liệu bán dẫn, nó cho phép màn hình phát sáng khi có dòng điện chạy qua.



Cấu tạo của màn hình OLED

Về cơ bản, một màn hình OLED sẽ bao gồm 4 thành phần chính sau:

Tấm nền

Được chế tạo bằng thủy tinh hay nhựa, có chức năng chống đỡ cho các bộ phận khác của OLED.

Anode

Tạo ra những khoảng trống để chứa điện tích dương khi dòng điện xuất hiện.

Cathode

Trái ngược với Anode, Cathode sẽ chịu trách nhiệm tạo ra các điện tích âm hay electron nếu có dòng điện đi qua.

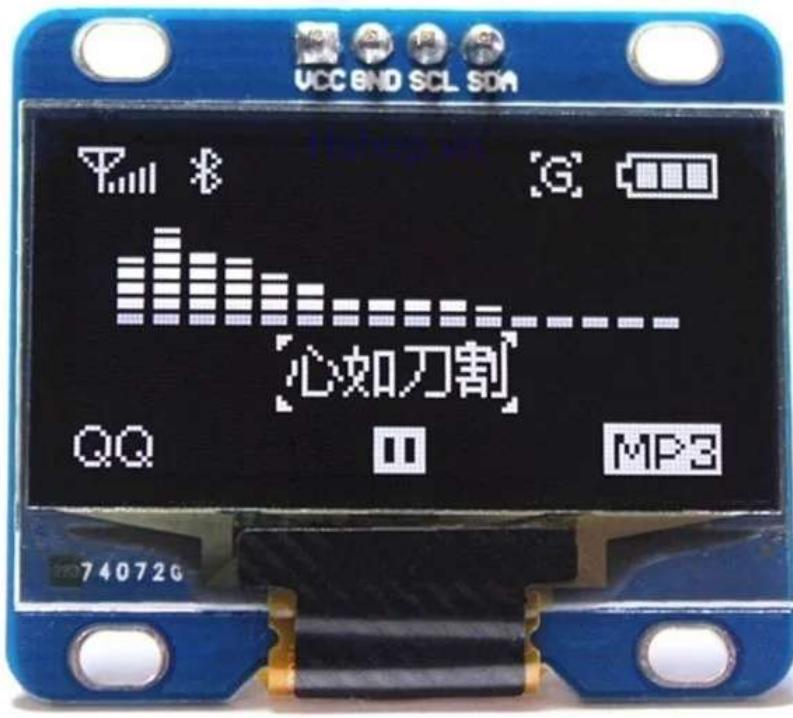
Lớp dẫn hữu cơ

Bao gồm hai thành phần chính với chức năng riêng biệt:*Lớp dẫn*: Cấu tạo từ các phân tử hữu cơ dẻo để giúp vận chuyển các chỗ trống từ Anode.*Lớp phát sáng* : Electron từ Cathode sẽ được truyền tải thông qua lớp này.

Thông số cơ bản về LCD Oled SSD1306

Màn hình OLED SSD1306 với kích thước 0.96 inch, cho khả năng hiển thị hình ảnh tốt với khung hình 128×64 pixel. Ngoài ra, màn hình còn tương thích với hầu hết các **vi điều khiển** hiện nay thông qua giao tiếp I2C. Màn hình sử dụng driver SSD1306 cùng thiết kế nhỏ gọn sẽ giúp bạn phát triển các sản phẩm DIY hoặc các ứng dụng khác một cách nhanh chóng.

Lưu ý: Trên thị trường có loại LCD sh1106 tương đối giống ssd1306 về hình thức. Nhưng độ phân giải là 132×64 và có một số tập lệnh khác ssd1306 nên các bạn lưu ý khi mua nhé.



Thông số kỹ thuật

- Driver: **SSD1306**
- Tương thích với Arduino, 51 Series, MSP430 Series, STM32 / 2, CSR IC,...

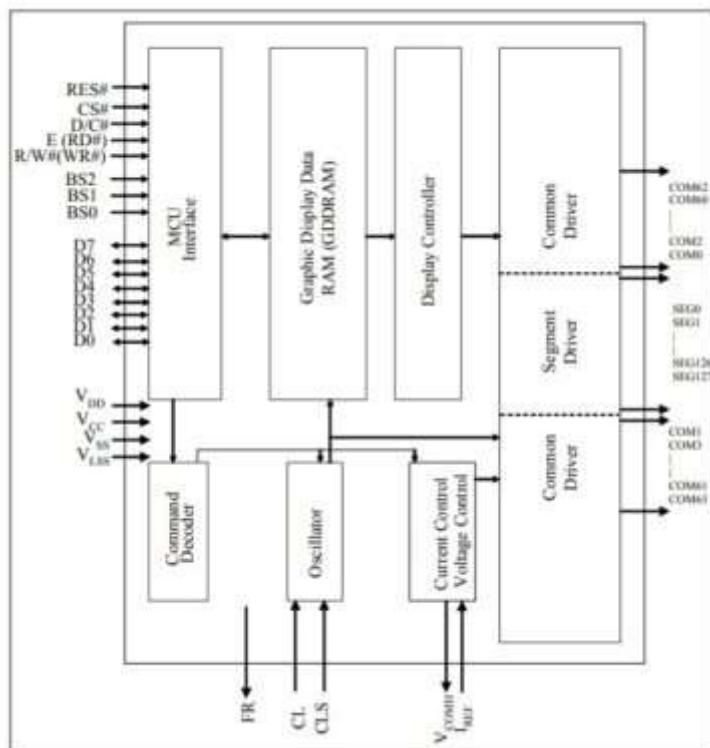
- Tiêu thụ điện năng thấp: 0.08W (fullscreen)
- Có thể điều chỉnh độ sáng và độ tương phản
- Chuẩn giao tiếp: I2C (thông qua 2 chân SCL, SDA)
- Điện áp hoạt động: 3V-5V DC
- Nhiệt độ hoạt động: -30°C-70°C
- Kích thước màn hình: 0.96 inch (128×64 pixel)
- Kích thước module: 26.70* 19.26* 1.85mm (1.030.760.07 inch)

Trong bài viết này chúng ta có module Oled SSD1306 sử dụng giao thức I2C để giao tiếp với STM32

Cách điều khiển LCD Oled SSD1306

Sơ đồ khối

Figure 4-1 SSD1306 Block Diagram



Các giao thức điều khiển

Để chuẩn bị cho việc sử dụng LCD Oled SSD1306, cần download datasheet của nó về tại: <https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf>

Giao thức điều khiển của SSD1306 bao gồm 8 chân dữ liệu D0-D7 và 4 chân điều khiển: RES, CS, DC, RD, WR.

Có 5 giao thức điều khiển ssd1306 được quy định bởi các bus BS[0:2]. Lần lượt là:

- Giao thức parallel 8 bit: 8080
- Giao thức parallel 8 bit: 6800
- Giao thức SPI 3 dây:
- Giao thức SPI 4 dây
- Giao thức I2C

Pin Name Bus Interface	Data/Command Interface								Control Signal					
	D7	D6	D5	D4	D3	D2	D1	D0	E	R/W#	CS#	D/C#	RES#	
8-bit 8080								D[7:0]		RD#	WR#	CS#	D/C#	RES#
8-bit 6800								D[7:0]		E	R/W#	CS#	D/C#	RES#
3-wire SPI	Tie LOW				NC	SDIN	SCLK	Tie LOW		CS#	Tie LOW	RES#		
4-wire SPI	Tie LOW				NC	SDIN	SCLK	Tie LOW		CS#	D/C#	RES#		
I ² C	Tie LOW				SDA _{OUT}	SDA _{IN}	SCL	Tie LOW		SA0		RES#		

Trong bài này mình chỉ đề cập tới giao thức I2C, các giao thức khác các bạn thao khảo trong datasheet nhé.

Giao thức I2C điều khiển ssd1306

Trong giao thức này chân D2 sẽ là SDA out, D1 là SDA in, D0 là SCL. Nếu các bạn chỉ ghi vào LCD thì chỉ cần sử dụng chân SDA in và SCL là đủ. Các chân I2C này bắt buộc sẽ phải có trờ kéo lên vcc theo đúng chuẩn I2C.

Khi truyền địa chỉ để giao tiếp I2C chúng ta sẽ truyền 1 byte như sau:

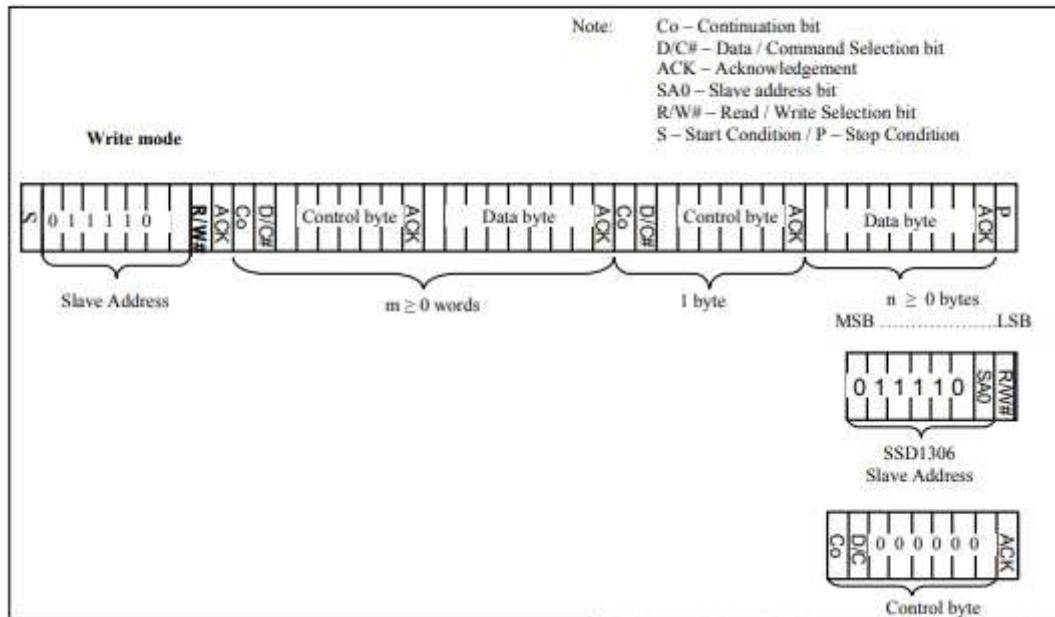
b7 b6 b5 b4 b3 b2 b1 b0

0 1 1 1 1 0 SA0 R/W#

Chân SA0 sẽ định địa chỉ cho SSD1306 vì vậy chúng ta có thể giao tiếp với 2 Oled LCD trên cùng 1 bus I2C.

Bit b0 sẽ quy định việc đọc hoặc ghi vào LCD.

Figure 8-7 : I²C-bus data format



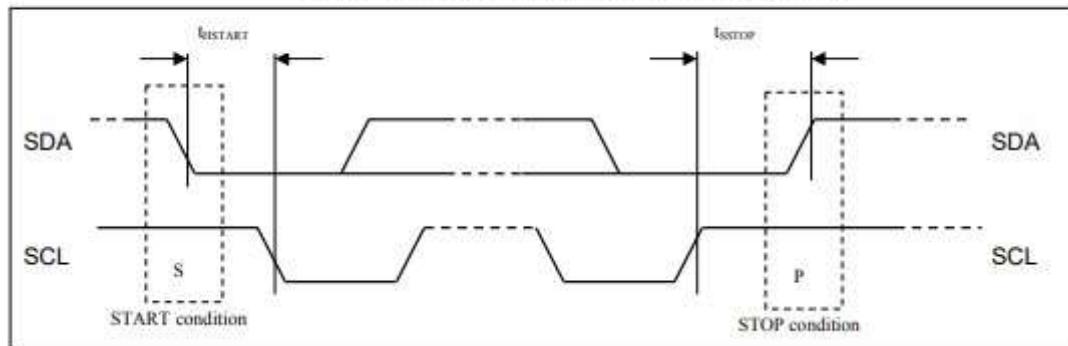
Các bước giao tiếp như sau:

- Điều kiện bắt đầu được thiết lập bằng cách kéo SDA từ CAO đến THẤP trong khi SCL vẫn CAO
- Ngay sau đó Master sẽ gửi 8bit bao gồm 7 bit địa chỉ + 1 bit R/W để quy định việc đọc hoặc ghi LCD. Chế độ ghi bit R/W = 0
- Nếu SSD1306 nhận được đúng địa chỉ sẽ kéo SDA xuống 0 để trả lời. (bit ACK)
- Sau khi nhận ACK, master truyền 1 byte sau đó gồm bit Co và D/C + 6 bit 0. Trong đó nếu Co = 0 thì các bit sau đó chỉ chưa dữ liệu. Nếu bit D/C = 0 truyền lệnh cho lcd. Nếu D/C = 1 truyền dữ liệu cho GDDRAM (hiển thị ra màn hình). Ô nhớ của GDDRAM tự động tăng khi có dữ liệu.
- Bit ACK sẽ được tạo ra mỗi khi truyền xong 1 byte dữ liệu

6. Cuối cùng điều kiện dừng được thiết lập bằng cách kéo “SDA in” từ THẤP đến CAO trong khi “SCL” vẫn ở mức CAO

Bảng sau đây sẽ định nghĩa điều kiện bắt đầu/kết thúc

Figure 8-8 : Definition of the Start and Stop Condition



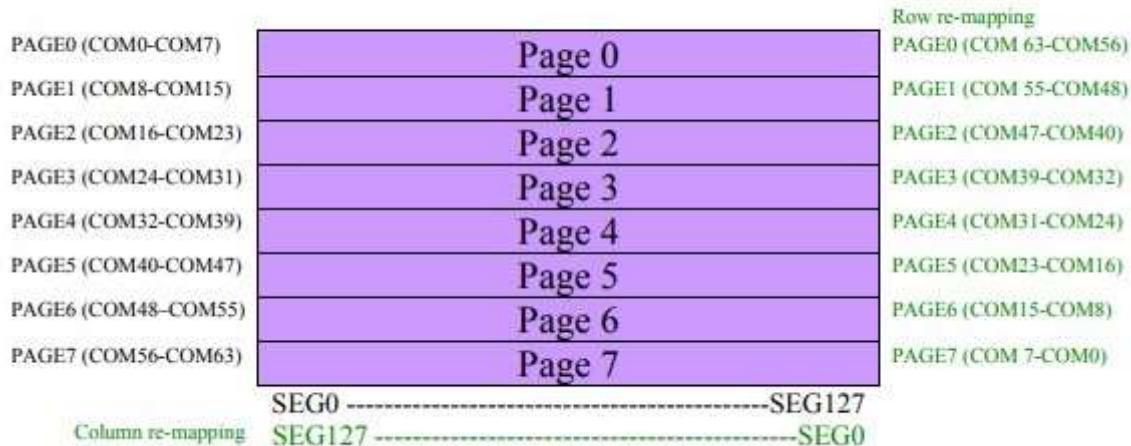
Nhìn chung phương thức giao tiếp của SSD1306 theo chuẩn I2C mà thôi, quan trọng là các bạn biết phải truyền gì vào trong đó.

Display RAM và cách hiển thị của ssd1306

Màn hình LCD oled SSD1306 có độ phân giải 128×64 dot. Được chia thành 8 Page Page0 – Page7.

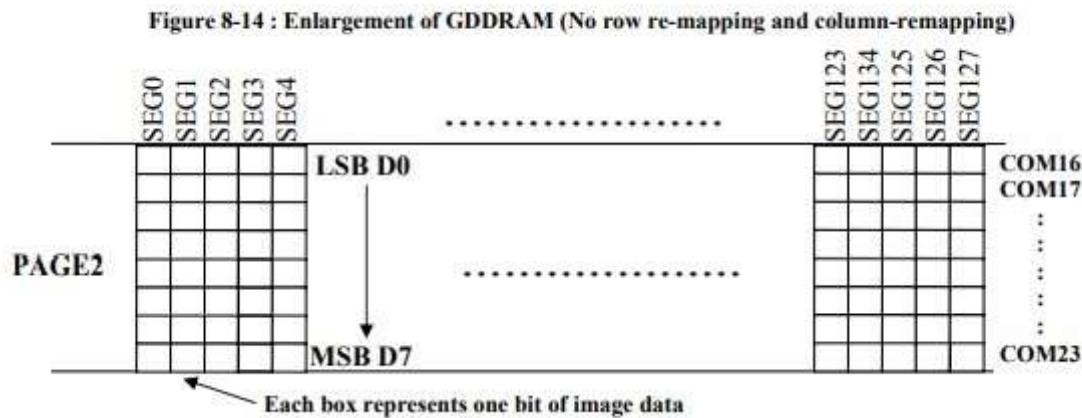
Mỗi page lại được chia thành các Segment SEG0 – SEG127.

Figure 8-13 : GDDRAM pages structure of SSD1306



Khi bắt đầu ghi vào LCD, con trỏ sẽ ở SEG0 và PAGE0 (nếu không có lệnh xoay lcd).

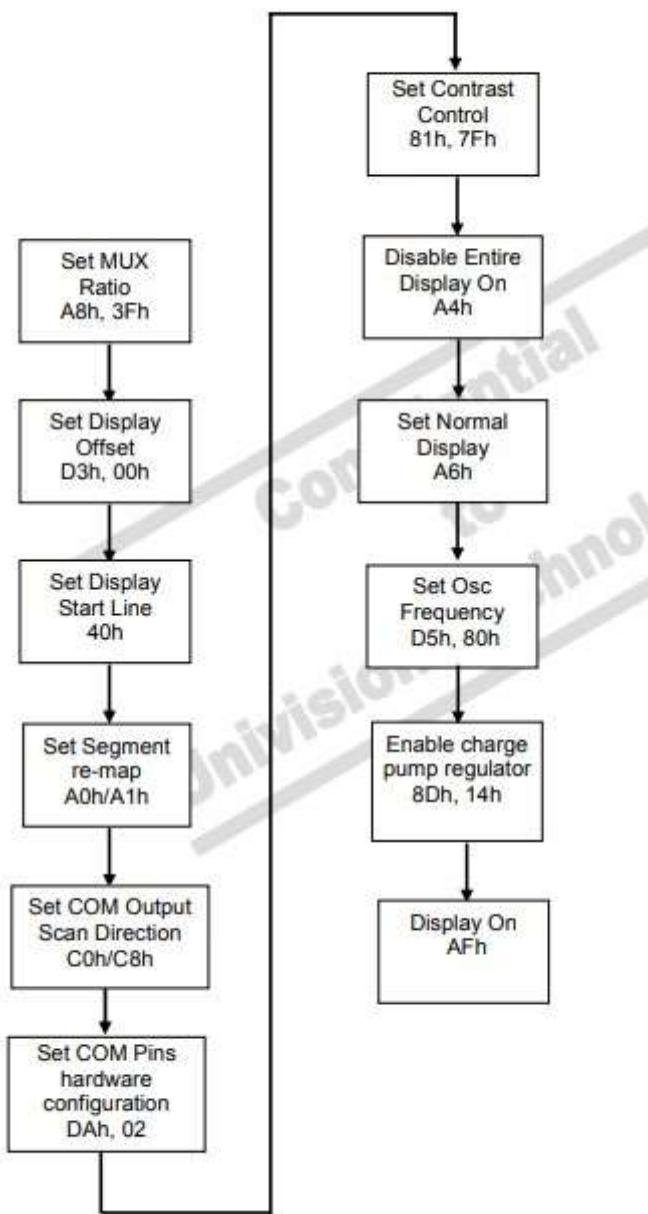
Mỗi khi một Byte được ghi vào GDDRAM con trỏ (Cursor) sẽ được tăng lên 1 đơn vị khi hết một Page con trỏ sẽ tự động nhảy xuống page phía dưới. Vậy nên phương pháp quét mặc định của LCD là từ Trên xuống dưới và từ trái qua phải.



Các lệnh và khởi tạo oled ssd1306

Chi tiết về tập lệnh cho ssd1306 các bạn tham khảo mục 10 **COMMAND DESCRIPTIONS** trong datasheet nhé.

Để khởi tạo cho ssd1306 hoạt động, chúng ta follow the flow chart sau:

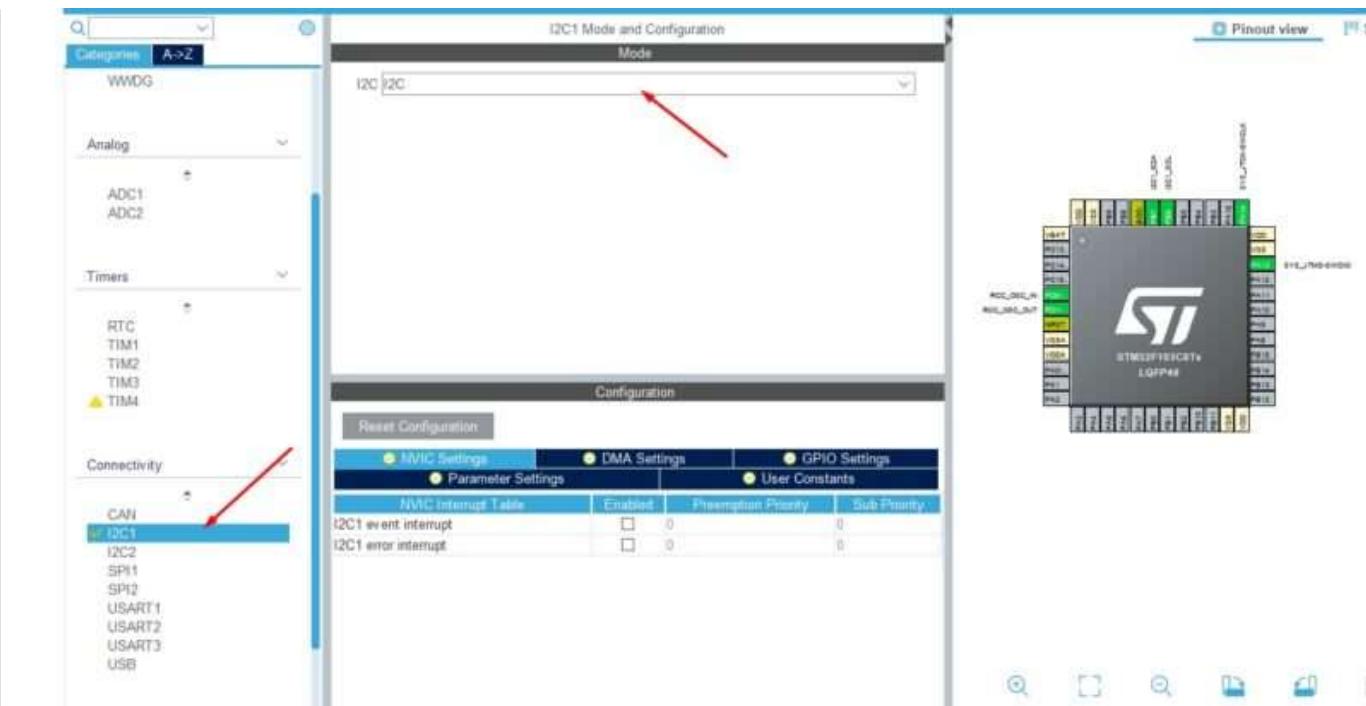
Figure 2 : Software Initialization Flow C

Lập trình STM32 điều khiển LCD ssd1306

Cấu hình trên cubemx

Trong cube chọn stm32f103c8, tab sys chọn debug serial wire. RCC chọn thạch anh ngoài và trong Clock các bạn để 72MHz tần số fmaster nhé.

Trong tab I2C enable lên.

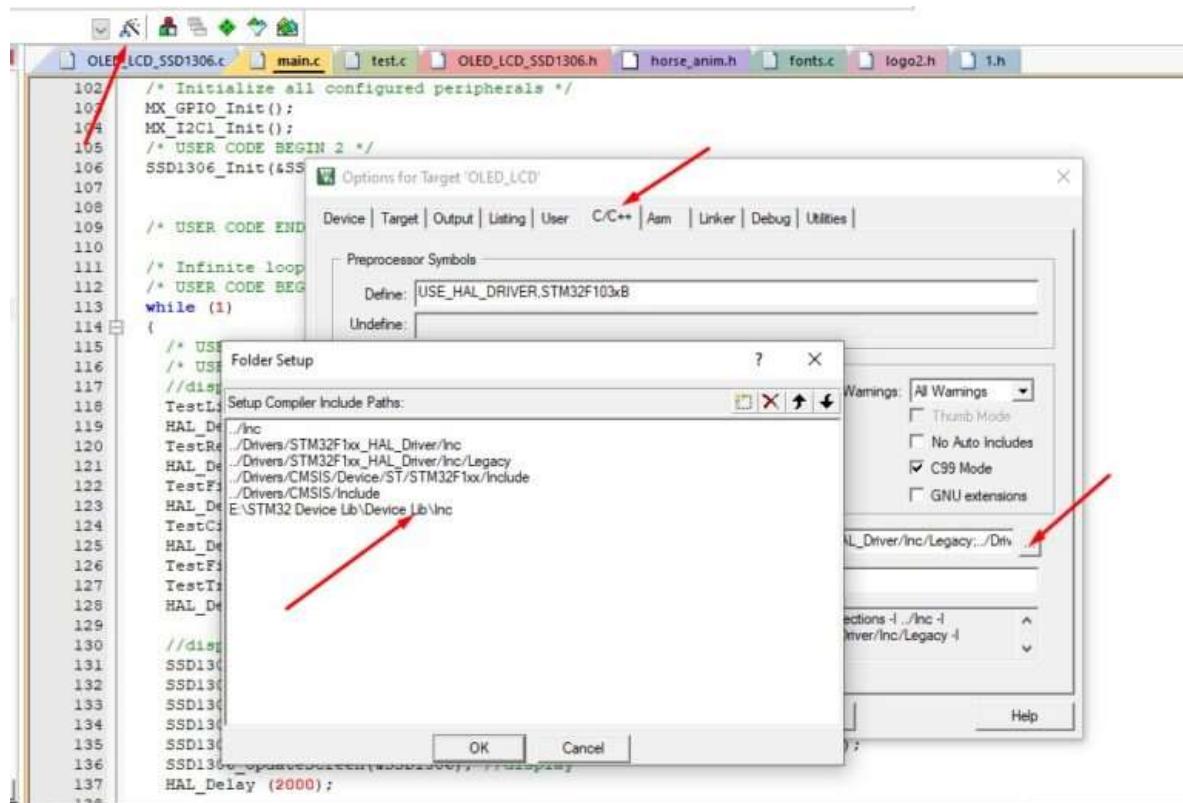


Sau đó gen code như các bài trước.

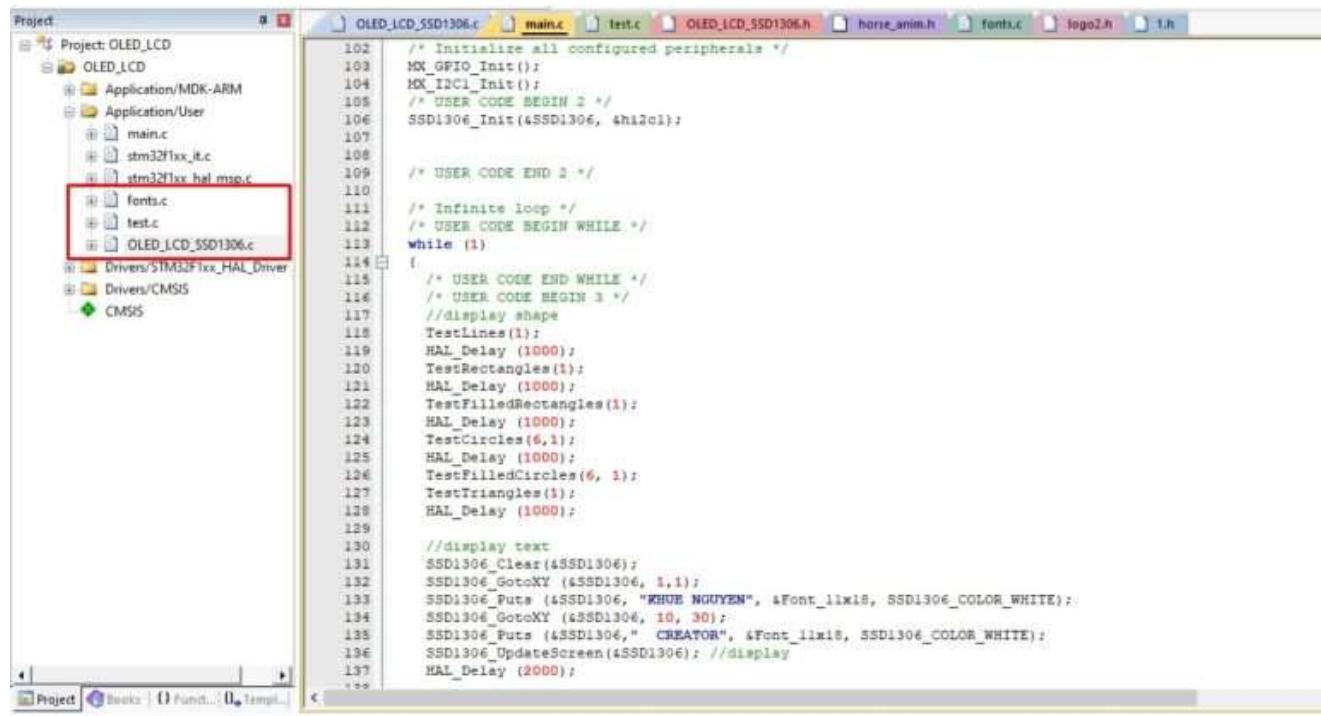
Download thư viện và **Lập trình**

Sau khi Gen code chúng ta sẽ mở bằng Keil C. Để download thư viện các bạn hãy đọc bài viết: [Hướng dẫn download thư viện STM32](#)

Sau khi down về, các bạn chỉnh đường dẫn tới thư viện Device.



Trong project click chuột phải -> add file. Thêm các file ssd1306 chứa thư viện, fonts chứa bộ fonts cho lcd, và test chứa các hàm test cho LCD.



Thêm các thư viện vào main.c gồm:

- Thư viện SSD1306

- Font chữ cho lcd
- File logo nike đã convert
- Ảnh động 1 – 9.

```

24 /* Private includes -----
25 /* USER CODE BEGIN Includes */
26 //#include "ssd1306.h"
27 #include "OLED_LCD_SSD1306.h"
28 #include "test.h"
29 #include "logo2.h"
30 #include "1.h"
31 #include "2.h"
32 #include "3.h"
33 #include "4.h"
34 #include "5.h"
35 #include "6.h"
36 #include "7.h"
37 #include "8.h"
38 #include "9.h"
39 /* USER CODE END Includes */
40
41 /* Private typedef -----
42 /* USER CODE BEGIN PTD */
43
44 /* USER CODE END PTD */
45

```

Tạo một biến Struct SSD1306

```

54 /* Private variables -----
55 I2C_HandleTypeDef hi2c1;
56
57 /* USER CODE BEGIN PV */
58 //SSD1306_Name SH1106;
59 SSD1306_Name SSD1306;
60 /* USER CODE END PV */
61
62 /* Private function prototypes -----
63 void SystemClock_Config(void);
64 static void MX_GPIO_Init(void);
65 static void MX_I2C1_Init(void);
66 /* USER CODE BEGIN PFP */
67
68 /* USER CODE END PFP */
69
70 /* Private user code -----
71 /* USER CODE BEGIN O */
72
73 /* USER CODE END O */
74
75

```

Khởi tạo LCD SSD1306 với phương thức giao tiếp là I2C1

```

99
100    /* USER CODE END SysInit */
101
102    /* Initialize all configured peripherals */
103    MX_GPIO_Init();
104    MX_I2C1_Init();
105    /* USER CODE BEGIN 2 */
106    SSD1306_Init(&SSD1306, &hi2c1); 
107
108
109    /* USER CODE END 2 */
110
111    /* Infinite loop */
112    /* USER CODE BEGIN WHILE */
113    while (1)
114    {

```

Hiển thị văn bản và các hình cơ bản trong LCD

Để vẽ các hình cơ bản chúng ta sử dụng các hàm đồ họa như: Drawline, Drawcircle, Drawrectangle....

Để viết chữ chúng ta sử dụng hàm Puts và truyền vào Text, Loại font và Màu sắc

Lưu ý: Mỗi khi sử dụng hàm thì dữ liệu sẽ chỉ được ghi vào bộ nhớ đệm. Để hiển thị lên LCD chúng ta sử dụng hàm UpdateScreen nhé!

```

113    /* USER CODE BEGIN WHILE */
114    {
115        /* USER CODE END WHILE */
116        /* USER CODE BEGIN 3 */
117        //display shape
118        TestLines(1);
119        HAL_Delay (1000);
120        TestRectangles(1);
121        HAL_Delay (1000);
122        TestFilledRectangles(1);
123        HAL_Delay (1000);
124        TestCircles(6,1);
125        HAL_Delay (1000);
126        TestFilledCircles(6, 1);
127        TestTriangles(1);
128        HAL_Delay (1000);
129
130        //display text
131        SSD1306_Clear(&SSD1306);
132        SSD1306_GotoXY (&SSD1306, 1,1);
133        SSD1306_Puts (&SSD1306, "KHUE NGUYEN", &Font_11x18, SSD1306_COLOR_WHITE);
134        SSD1306_GotoXY (&SSD1306, 10, 30);
135        SSD1306_Puts (&SSD1306, " CREATOR", &Font_11x18, SSD1306_COLOR_WHITE);
136        SSD1306_UpdateScreen(&SSD1306); //display
137        HAL_Delay (2000);
138

```

Hiển thị ảnh với ssd1306

Tương tự với hiển thị ảnh chúng ta sử dụng hàm Drawbitmap. Truyền vào tọa độ bắt đầu, tọa độ kết thúc. mảng chứa dữ liệu bitmap và màu sắc.

```

138
139 //display image
140 SSD1306_Clear(&SSD1306);
141 SSD1306_DrawBitmap(&SSD1306, 0,0,gImage_logo2, 132, 64, SSD1306_COLOR_WHITE);
142 SSD1306_UpdateScreen(&SSD1306);
143 HAL_Delay(5000);
144

```

Hiển thị ảnh động GIF với ssd1306

Ảnh động thực ra là chuỗi các ảnh được hiển thị liên tục, chúng ta sẽ cho nó vào 1 vòng while với số lần lặp lại được quy định trong loop.

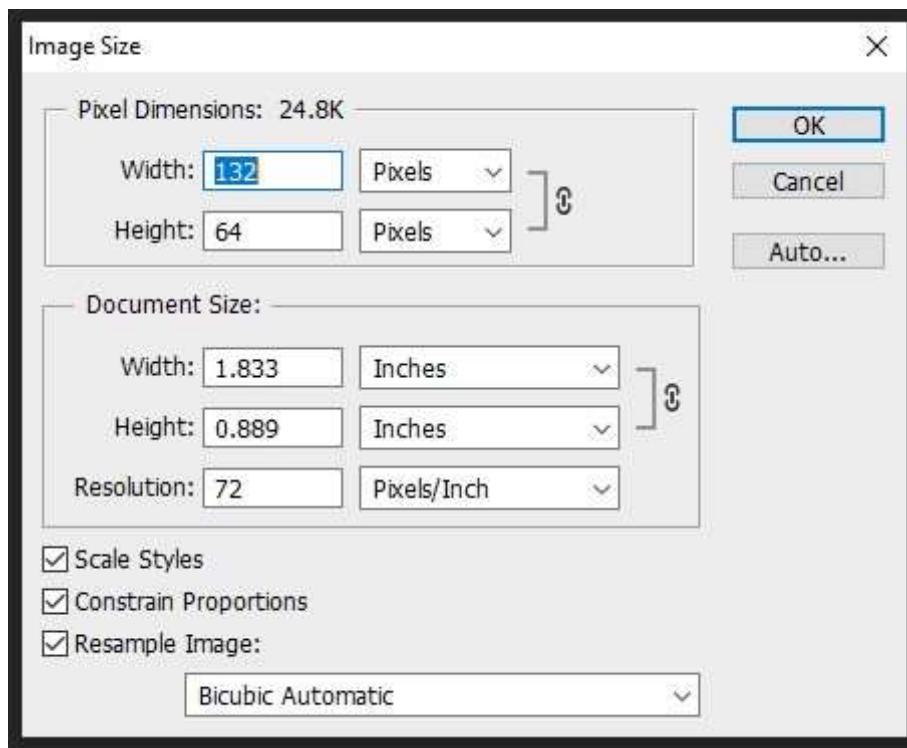
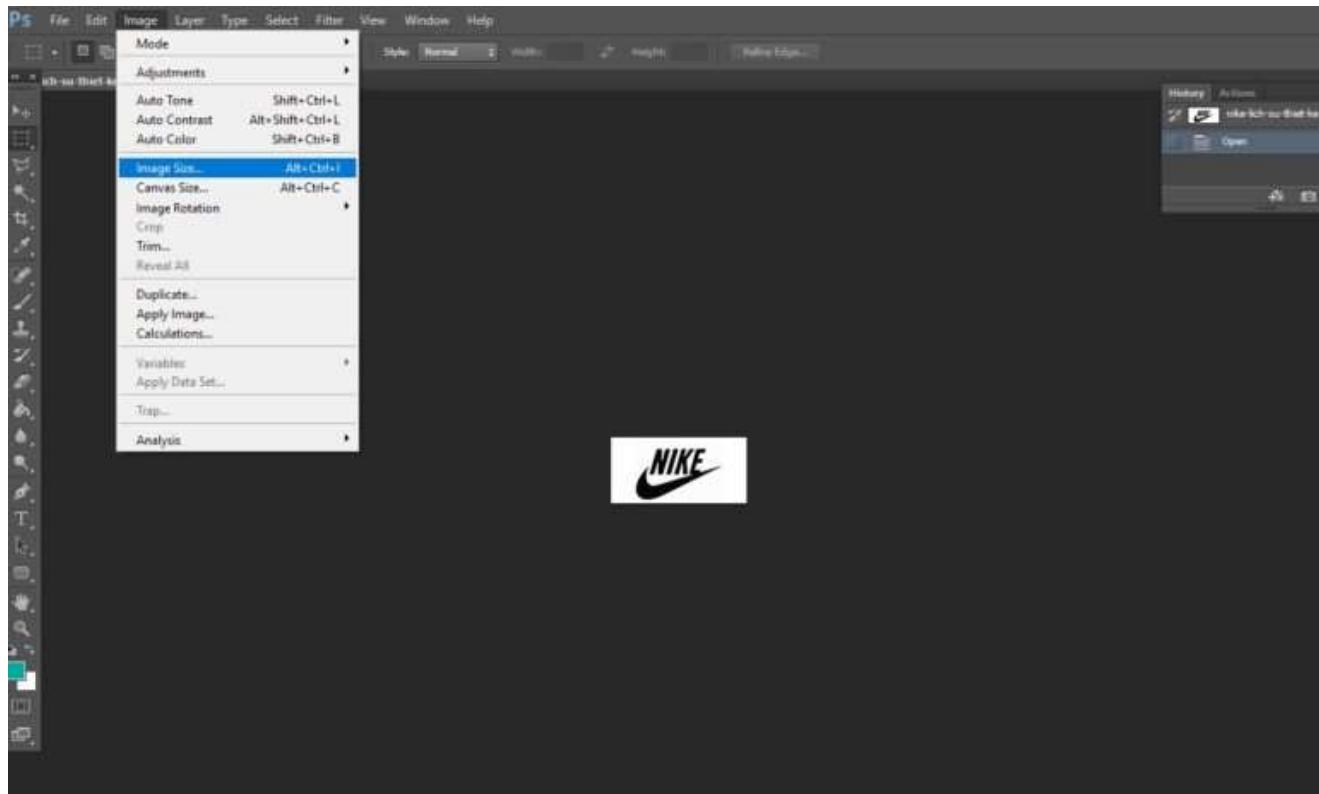
```

161     uint8_t loop = 5;
162     while(loop--)
163     {
164         SSD1306_Clear(&SSD1306);
165         SSD1306_DrawBitmap(&SSD1306,0,0,gImage_1,132,64,SSD1306_COLOR_WHITE);
166         SSD1306_UpdateScreen(&SSD1306);
167         SSD1306_Clear(&SSD1306);
168         SSD1306_DrawBitmap(&SSD1306,0,0,gImage_2,132,64,SSD1306_COLOR_WHITE);
169         SSD1306_UpdateScreen(&SSD1306);
170         SSD1306_Clear(&SSD1306);
171         SSD1306_DrawBitmap(&SSD1306,0,0,gImage_3,132,64,SSD1306_COLOR_WHITE);
172         SSD1306_UpdateScreen(&SSD1306);
173         SSD1306_Clear(&SSD1306);
174         SSD1306_DrawBitmap(&SSD1306,0,0,gImage_4,132,64,SSD1306_COLOR_WHITE);
175         SSD1306_UpdateScreen(&SSD1306);
176         SSD1306_Clear(&SSD1306);
177         SSD1306_DrawBitmap(&SSD1306,0,0,gImage_5,132,64,SSD1306_COLOR_WHITE);
178         SSD1306_UpdateScreen(&SSD1306);
179         SSD1306_Clear(&SSD1306);
180         SSD1306_DrawBitmap(&SSD1306,0,0,gImage_6,132,64,SSD1306_COLOR_WHITE);
181         SSD1306_UpdateScreen(&SSD1306);
182         SSD1306_Clear(&SSD1306);
183         SSD1306_DrawBitmap(&SSD1306,0,0,gImage_7,132,64,SSD1306_COLOR_WHITE);
184         SSD1306_UpdateScreen(&SSD1306);
185         SSD1306_Clear(&SSD1306);
186         SSD1306_DrawBitmap(&SSD1306,0,0,gImage_8,132,64,SSD1306_COLOR_WHITE);
187         SSD1306_UpdateScreen(&SSD1306);
188         SSD1306_Clear(&SSD1306);
189         SSD1306_DrawBitmap(&SSD1306,0,0,gImage_9,132,64,SSD1306_COLOR_WHITE);
190         SSD1306_UpdateScreen(&SSD1306);
191     }

```

Cách chuyển ảnh sang mảng để hiển thị

Download ảnh cần convert. Chỉnh lại kích thước theo đúng mong muốn bằng photoshop. Chúng ta có thể dùng công cụ Image size (phóng to thu nhỏ theo kích thước) và Canvas Size(Thêm hoặc cắt ảnh theo kích thước)



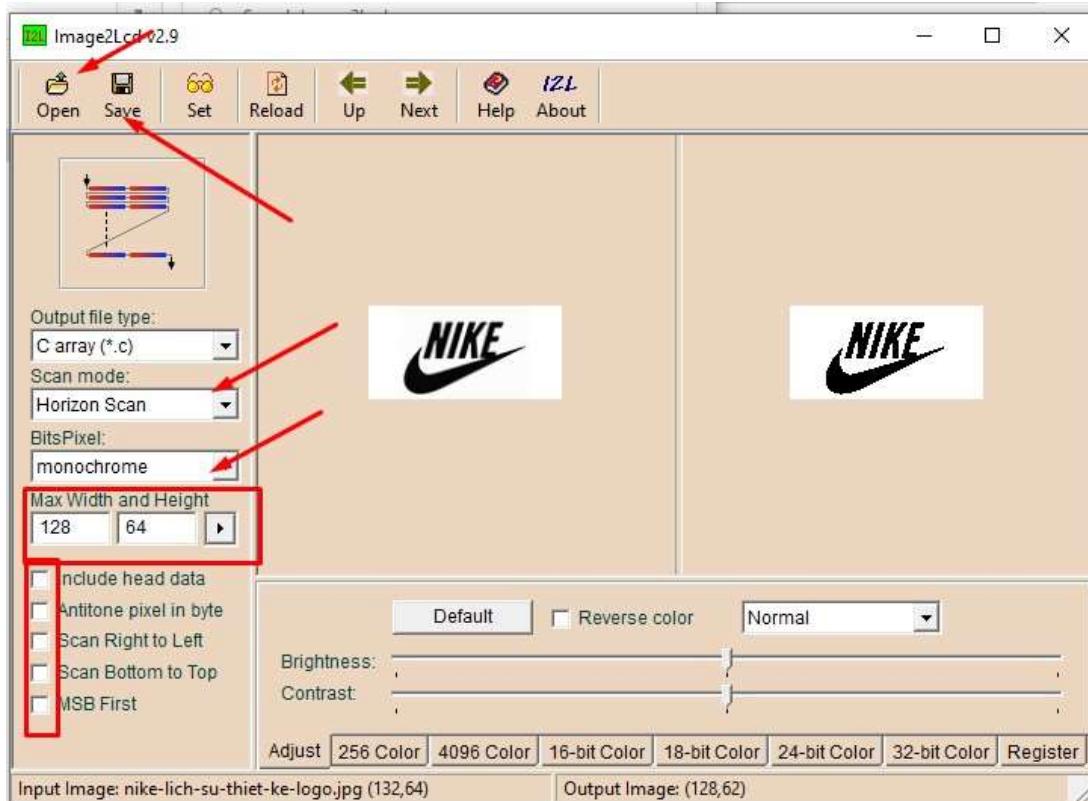
Download công cụ Image2LCD tại đây:

<https://www.fshare.vn/file/Q8NDJBO9ZU2F>

Mở phần mềm.

1. Nhấn open mở ảnh vừa chỉnh trong photoshop

2. Chọn các tùy chọn như hình gồm: kiểu quét, chế độ màu, độ phân giải, các tùy chọn khác
3. Nhấn Save sau đó save thành file .h nhớ trỏ vào trong thư mục Inc trong project của bạn nhé.



Trong main.c thêm #include “Tên_File.h” rồi có thể sử dụng được rồi.

Đây là kết quả của mình.



Facebook Watch

Kết

Màn hình Oled LCD SSD1306 rất phù hợp dùng trong các ứng dụng nhỏ gọn, Độ phân giải và hiển thị của nó cũng tương đối ổn. Ngoài I2C ssd1306 còn có thể giao tiếp theo các phương thức khác, giúp cho tốc độ được nâng cao hơn rất nhiều.

Nếu thấy bài viết này hay, hãy chia sẻ tới những người bạn học hay đồng nghiệp của mình. Và nếu thắc mắc điều gì, hãy để lại bình luận nhé

Và cùng gia nhập những người nghiên lập trình tại đây nhé: [Hội anh em nghiên lập trình](#)

4.5/5 - (4 bình chọn)

Related Posts:

1. [Lập trình STM32 đọc nhiệt độ với DS18b20 giao tiếp OneWire](#)
2. [Hướng dẫn download và sử dụng tài liệu Lập trình STM32](#)

3. **Bài 14: Sử dụng STM32 IWDG Independent Watchdog Timer chống treo vi điều khiển**
4. **Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231**
5. **Bài 3: Lập trình STM32 GPIO điều khiển Led và nút nhấn**
6. **Bài 2: Tổng quan về KIT STM32F103C8T6 Blue Pill**



KHUÊ NGUYỄN

Chỉ là người đam mê điện tử và lập trình. Làm được gì thì viết cho anh em xem thôi. :D

Trả lời

Email của bạn sẽ không được hiển thị công khai. Các trường bắt buộc được đánh dấu *

Bình luận *

Tên *

Email *



Trang web



PHẢN HỒI

Fanpage

Khuê Nguyễn Creator - Họ...
2.754 lượt thích

Đã thích **Chia sẻ**

**Khuê Nguyễn Creator - Học
Lập Trình Vi Điều Khiển**
khoảng một tháng trước

Lý do thời gian gần đây mình không viết bài
và làm thêm gì cả là đây 😊)
Chính thức ra mắt sản phẩm định vị thông
minh vTag.

Đây là một sản phẩm định vị đa năng với
3 công nghệ định vị WIFI, GPS, LBS kết
hợp với sóng NB-IOT dành riêng cho các
sản phẩm IOT.

Chỉ với 990.000đ chúng ta đã có thể có
sản phẩm để:

- Định vị trẻ em, con cái... Xem thêm

Bài viết khác

Lập trình 8051 - AT89S52

Khuê Nguyễn Creator

Lập trình 8051 - AT89S52



PROTEUS

Bài 1: Tổng quan về 8051 và chip AT89S51 - 52

Tổng quan về 8051

8051 là một dòng chip nhập môn cho lập trình viên nhúng, chúng được sử...

[ĐỌC THÊM](#)

Lập trình STM32 và CubeMX



Khuê Nguyễn Creator



Lập trình STM32 HID Host giao tiếp với chuột và bàn phím

Lập trình STM32 USB HID Host giao tiếp với chuột và bàn phím máy tính

Trong bài này, chúng ta sẽ cùng học STM32 HID Host hiển STM32 giấu nhẹm

[ĐỌC THÊM](#)

Lộ trình học lập trình nhúng từ A tới Z

Lập trình nhúng là một ngành có cơ hội nhưng cũng đòi hỏi nhiều kiến...

[3 COMMENTS](#)[ĐỌC THÊM](#)

Lập trình STM32F407 SDIO đọc dữ liệu thẻ nhớ

Lập trình STM32 SDIO đọc ghi dữ liệu vào thẻ nhớ SD card

Trong bài này chúng ta cùng học cách lập trình STM32 SDIO, một chuẩn giao...

[ĐỌC THÊM](#)

Lập trình STM32 và CubeMX



Khuê Nguyễn Creator



Lập trình STM32F407 DAC chuyển đổi số sang tương tự

Lập trình STM32 DAC tạo sóng hình Sin trên KIT STM32F407 Discovery

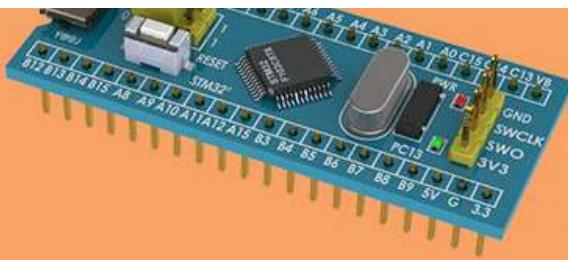
Trong bài này chúng ta sẽ cùng nhau tìm hiểu STM32 DAC với KIT STM32F407VE...

[ĐỌC THÊM](#)

Lập trình STM32 và CubeMX



Khuê Nguyễn Creator



Sử dụng hàm printf để in Log khi Debug trên STM32

Hướng dẫn sử dụng printf với STM32 Uart để in Log trên Keil C

Trong bài này chúng ta sẽ học cách retarget hàm printf của thư viện stdio...

3 COMMENTS

[ĐỌC THÊM](#)

ESP32 và Platform IO



Khuê Nguyễn Creator



Bài 9 WIFI: Lập trình ESP32 OTA nạp firmware trên Internet

Lập trình ESP32 FOTA nạp firmware qua mạng Internet với OTA Drive

Trong bài này chúng ta sẽ học cách sử dụng ESP32 FOTA (Firmware Over The...

4 COMMENTS

[ĐỌC THÊM](#)

Lập trình Nuvoton



Khuê Nguyễn Creator



Cài đặt SDC Complier và Code:Blocks IDE

Hướng dẫn cài đặt SDCC và Code::Blocks lập trình Nuvoton

Ở bài này chúng ta sẽ cài đặt các công cụ cần thiết cho việc...

[ĐỌC THÊM](#)

Blog này làm ra để lưu trữ tất cả những kiến thức, những câu chuyện của mình. Đôi khi là những ý tưởng nhất thời, đôi khi là các dự án tự mình làm. Chia sẻ cho người khác cũng là niềm vui của mình, kiến thức mỗi người là khác nhau, không hẳn quá cao siêu nhưng sẽ có lúc hữu dụng.

Liên Kết

Nhóm: Nghiên Lập Trình

Fanpage: Khuê Nguyên Creator

My Shop

Thông Tin

Tác Giả

Chính Sách Bảo Mật



Copyright 2022 © Khuê Nguyễn