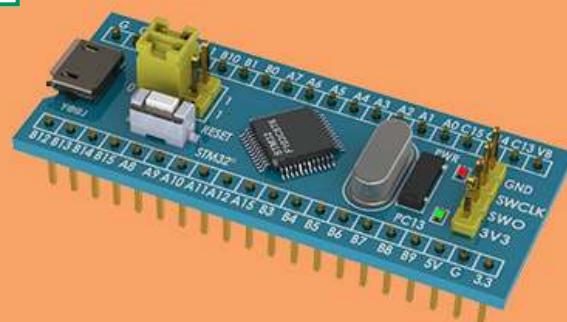


**LẬP TRÌNH STM32**

Lập trình STM32 với TFT LCD ST7735 giao thức SPI

POSTED ON 02/08/2021 BY KHUÊ NGUYỄN

02
Th8**Khuê Nguyễn Creator**

Lập trình STM32 hiển thị ảnh với LCD TFT ST7735

TFT LCD ST7735 là một trong những dòng LCD hiển thị màu phổ biến hiện nay. Với giao thức **SPI** đơn giản, đỡ tốn tài nguyên nhưng tốc độ cũng vẫn rất ổn định rất phù hợp với các dự án nhỏ.

Bài này nằm trong serie **Học STM32 từ A tới Z**

Mục Lục



1. Tổng quát về màn hình TFT LCD ST7735
2. Hướng dẫn điều khiển TFT LCD ST7735
 - 2.1. Sơ đồ khối TFT LCD ST7735
 - 2.2. Chuẩn giao tiếp SPI – 3 wire Serial
 - 2.3. Cấu tạo điểm ảnh TFT LCD ST7735
 - 2.4. Bản đồ bộ nhớ DRAM
 - 2.5. Phương thức quét
 - 2.6. Lệnh điều khiển
3. Lập trình STM32 điều khiển TFT LCD ST7735
 - 3.1. Kết nối STM32 Blue Pill với LCD ST7735
 - 3.2. Cấu hình cubemx
 - 3.3. Lập trình STM32 với Keil C
 - 3.4. Cách chuyển ảnh sang mảng để hiển thị
 - 3.5. Kết quả
4. Kết
 - 4.1. Related posts:

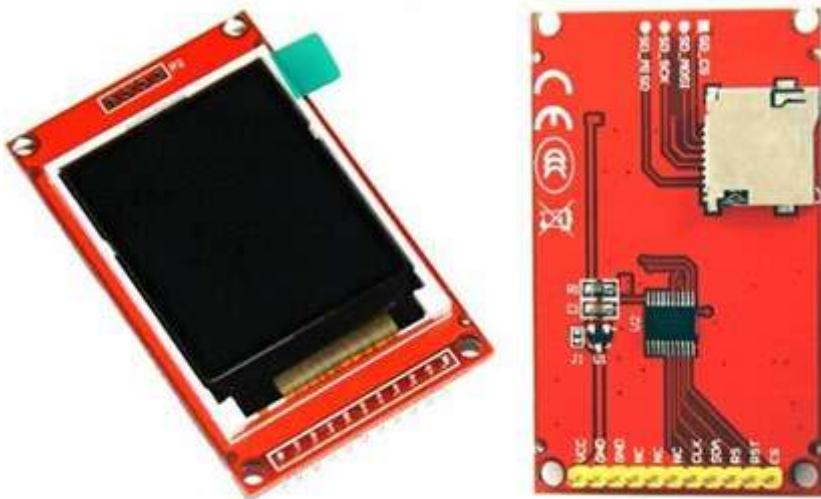
Tổng quát về màn hình TFT LCD ST7735

Màn hình LCD TFT 1.8 inch ST7735 giao tiếp SPI còn được tích hợp IC chuyển nguồn và IC chuyển mức Logic nên có thể tương thích với cả hai mức điện áp Logic giao tiếp thông dụng là 3.3VDC và 5VDC, 1 điểm đặc biệt nữa của **màn hình LCD** này là có sơ đồ chân tương thích để có thể cắm trực tiếp với Arduino mà không cần thêm dây bus.

THÔNG SỐ KỸ THUẬT MÀN HÌNH TFT LCD ST7735

- Điện áp sử dụng: 3.3~5VDC
- Điện áp giao tiếp: TTL 3.3~5VDC
- IC Driver hiển thị: ST7735
- Cỡ màn hình: 2.0 inch
- Độ phân giải: 128 x 160 pixels

- Tích hợp khe thẻ nhớ MicroSD giao tiếp SPI.
- Hỗ trợ nhiều MCU trong dòng mà không cần bất kỳ hệ thống dây điện
- Dự trữ mạch chữ SPI FLASH để tạo điều kiện cho ứng dụng mở rộng

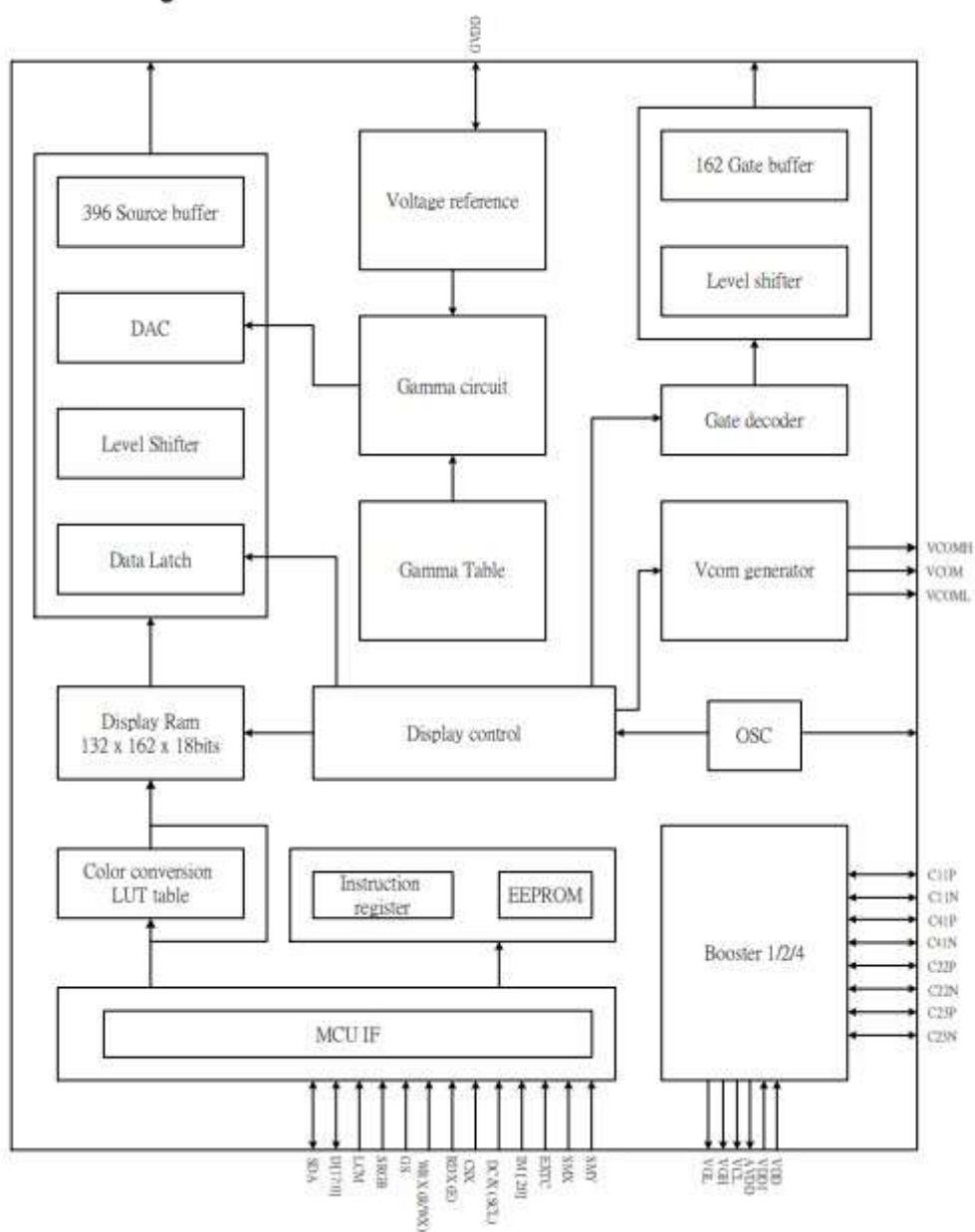


Hướng dẫn điều khiển TFT LCD ST7735

Để bắt đầu chúng ta tham khảo Datasheet:

<https://www.displayfuture.com/Display/datasheet/controller/ST7735.pdf>

Sơ đồ khối TFT LCD ST7735



Chuẩn giao tiếp SPI – 3 wire Serial

LCD ST7735 hỗ trợ 5 phương thức giao tiếp được quy định bởi các bit IM[2:0]

IM2	IM1	IM0	Interface	Read back selection
0	-	-	3-line serial interface	Via the read instruction
1	0	0	8080 MCU 8-bit parallel	RDX strobe (8-bit read data and 8-bit read parameter)
1	0	1	8080 MCU 16-bit parallel	RDX strobe (16-bit read data and 8-bit read parameter)
1	1	0	8080 MCU 9-bit parallel	RDX strobe (9-bit read data and 8-bit read parameter)
1	1	1	8080 MCU 18-bit parallel	RDX strobe (18-bit read data and 8-bit read parameter)

Trong bài này chúng ta sử dụng chuẩn 3 Wire hay SPI Transmit only (Do Module đã thiết kế sẵn như vậy).

Chân CS – Chip Select: Chọn chip trong giao tiếp SPI

Chân RST – Reset chip

Chân RS – Register Select (Hoặc D/C trong datasheet) Chân này có tác dụng chọn dữ liệu truyền qua serial sẽ là Lệnh (Command) hay Dữ liệu (Data)

Chân CLK – Clock: Xung đồng hồ trong giao tiếp SPI

Chân SDA – Data: Ở mode này chân SDA chính là Chân D0

Vì điều khiển phải ghi các mức 1 0 vào các chân này để có thể điều khiển được LCD ST7735

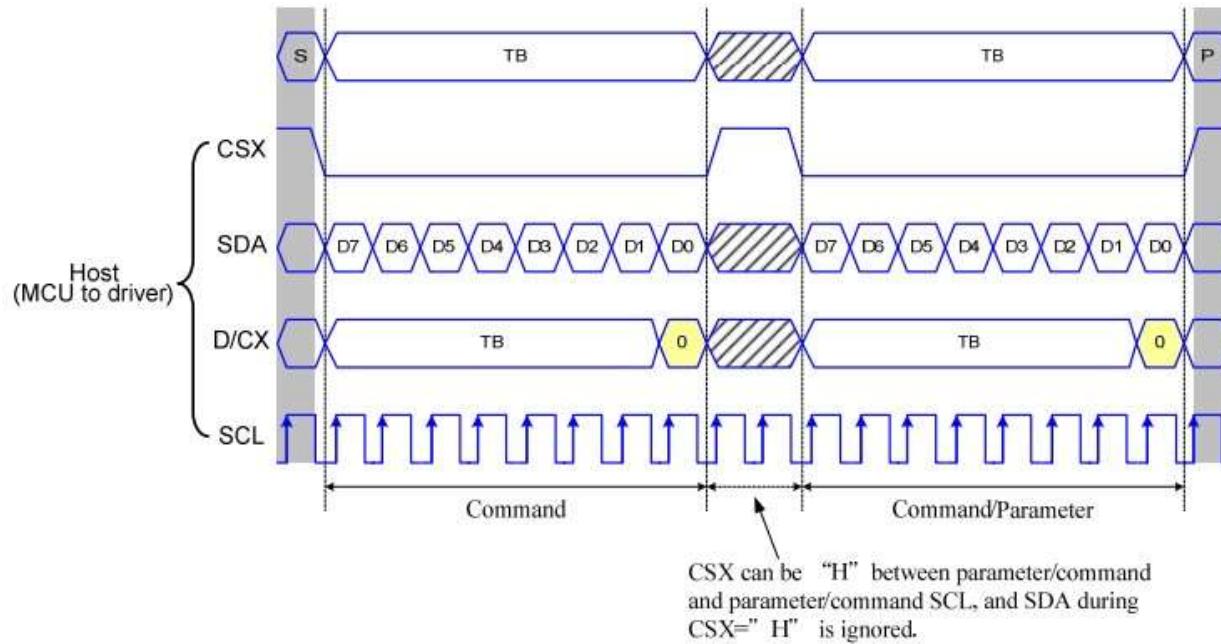
IM2	IM1	IM0	Interface	RDX	WRX	D/CX	Read back selection
0	-	-	3-line serial	Note1	Note1	SCL	D[17:1]: unused, D0: SDA
1	0	0	8080 8-bit parallel	RDX	WRX	D/CX	D[17:8]: unused, D7-D0: 8-bit data
1	0	1	8080 16-bit parallel	RDX	WRX	D/CX	D[17:16]: unused, D15-D0: 16-bit data
1	1	0	8080 9-bit parallel	RDX	WRX	D/CX	D[17:9]: unused, D8-D0: 9-bit data
1	1	1	8080 18-bit parallel	RDX	WRX	D/CX	D17-D0: 18-bit data

Module của chúng ta sử dụng Mode Serial 4 line vậy nên chân SPI4W phải được kéo lên 1.

- Với chế độ 3 Line: Bit D/CX hay data/command sẽ được chỉ định bởi dữ liệu ghi vào. Gồm 8bit dữ liệu và 1 bit D/CX
- Với chế độ 4 Line: But D/CX sẽ được chọn bởi phần cứng. Điều khiển qua 1 GPIO khác

IM2	SPI4W	Interface	Read back selection
0	0	3-line serial interface	Via the read instruction (8-bit, 24-bit and 32-bit read parameter)
0	1	4-line serial interface	Via the read instruction (8-bit, 24-bit and 32-bit read parameter)

Dưới đây là bảng phương thức giao tiếp kiểu Seriel 4 Wire. Nói chung nó là SPI thôi. Chỉ cần khi giao tiếp bạn chọn chân RS (D/CX) để gửi Command hay Data là được,



Cấu tạo điểm ảnh TFT LCD ST7735

Data Color Coding

LCD ST7735 hỗ trợ 3 định dạng màu sắc khác nhau:

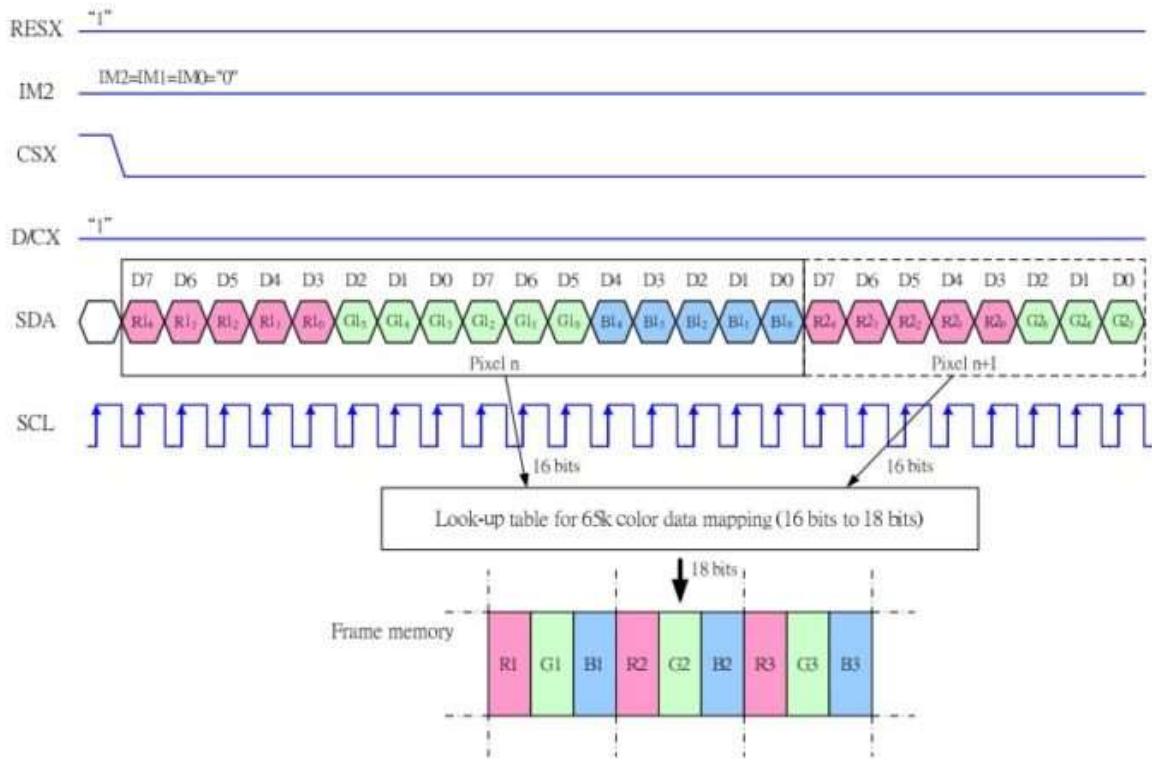
- 4k colors, RGB 4,4,4 – bit input.
- 65k colors, RGB 5,6,5 – bit input.
- 262k colors, RGB 6,6,6 – bit input.

Định dạng chúng ta sử dụng sẽ là RGB 565. Đây cũng là một định dạng rất phổ biến thường gặp trong LCD. Bởi vì chúng tròn 16bit. Truyền cho nó đơn giản 😊

Mỗi điểm ảnh sẽ được hiển thị bởi 16bit (2 Byte) trong đó:

- R: Red 5 bit
- G: Green 6 bit

- B: Blue 5 bit



Bản đồ bộ nhớ DRAM

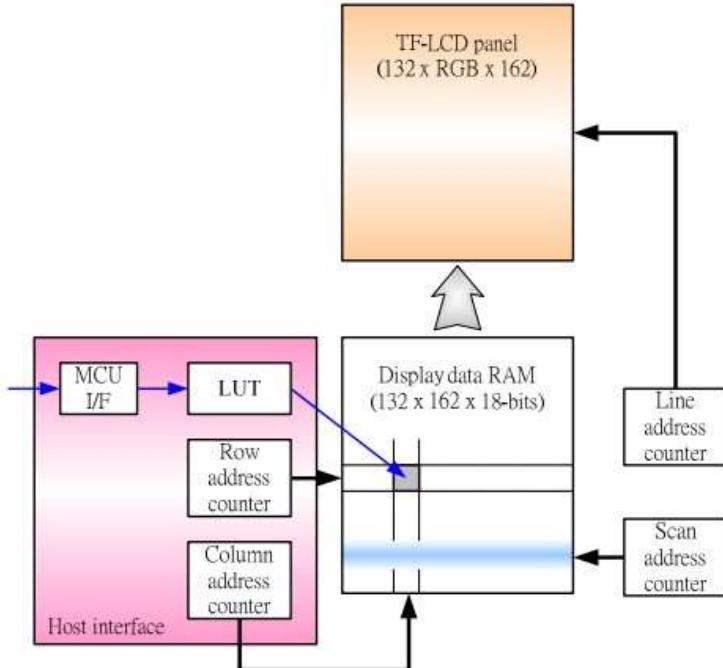


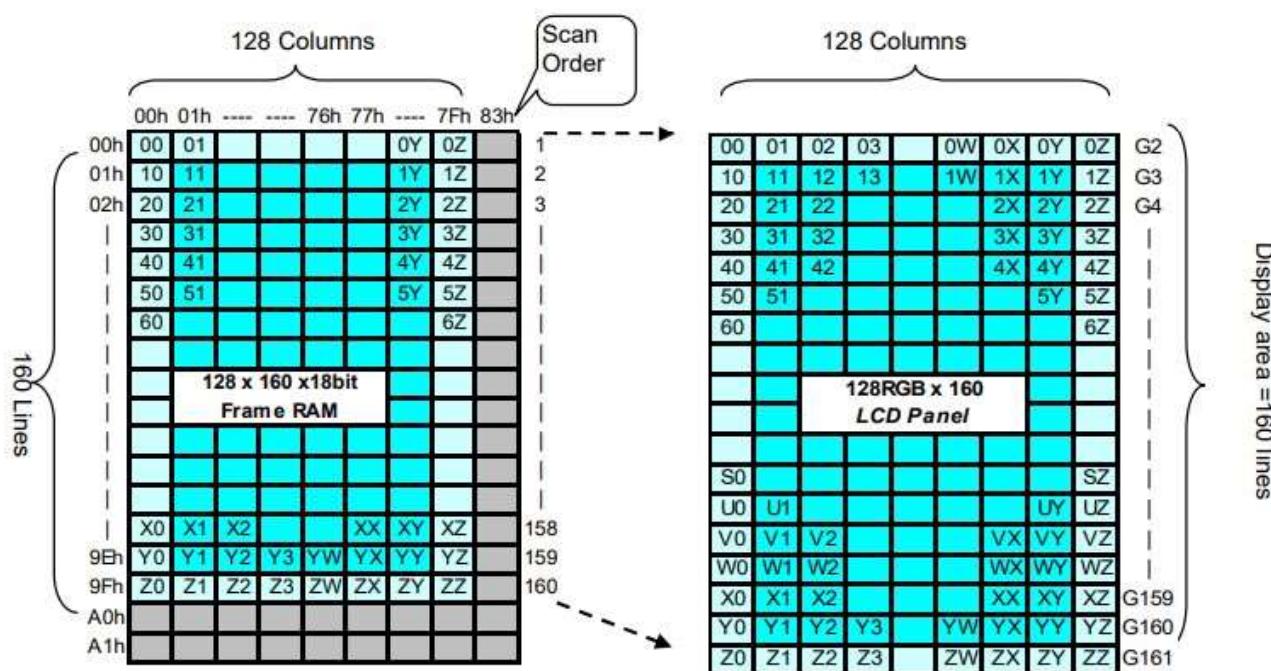
Fig. 9.8.1 Display data RAM organization

LCD ST7735 tích hợp bộ nhớ Static RAM có kích thước $132 \times 162 \times 18$ bits. Nghĩa là lưu được một hình ảnh kích thước 132×162 pixel với độ phân giải 18bit/pixel.

		Pixel 1						Pixel 2						Pixel 131						Pixel 132					
Gate Out	Source Out	S1	S2	S3	S4	S5	S6	—	S391	S392	S393	S394	S395	S396	—	SA	ML='0'	ML='1'							
		RA						RGB Order						SA											
		MY='0'	MY='1'																						
1	0	161		R0	G0	B0	R1	G1	B1	—	R131	G131	B131	R132	G132	B132	0	161							
2	1	160															1	160							
3	2	159															2	159							
4	3	158															3	158							
5	4	157															4	157							
6	5	156															5	156							
7	6	155															6	155							
8	7	154															7	154							
155	154	7															154	7							
156	155	6															155	6							
157	156	5															156	5							
158	157	4															157	4							
159	158	3															158	3							
160	159	2															159	2							
161	160	1															160	1							
162	161	0															161	0							
CA		MX='0'	0			1			130			131													
		MX='1'	131			130			1			0													

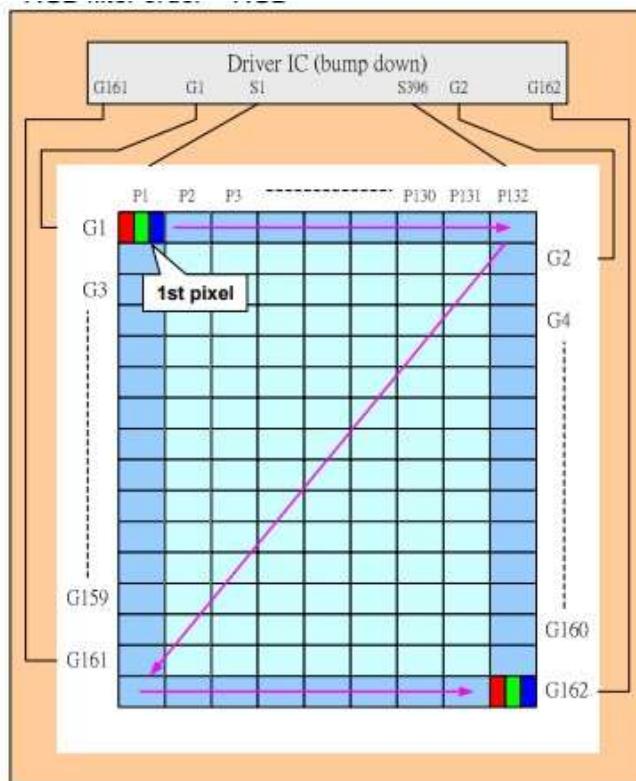
Địa chỉ hàng và cột được đánh thứ tự bởi các biến RA (Row Address) và CA (Collum Address). Thứ tự này sẽ thay đổi khi có các lệnh Mirror. Giúp chúng ta xoay LCD theo các chiều khác nhau

Với Module màn hình TFT LCD ST7735 chúng ta sử dụng độ phân giải 160×128 . Vậy nên bản đồ bộ nhớ DRAM như sau:



Phương thức quét

Chiều mặc định của ST7735 là Từ trái sang phải và từ trên xuống dưới. Chiều quét có thể được **lập trình** lại bởi các câu lệnh.



Lệnh điều khiển

Các lệnh điều khiển các bạn tham khảo mục **Command** trong Datasheet nhé.

Lập trình STM32 điều khiển TFT LCD ST7735

Kết nối STM32 Blue Pill với LCD ST7735

SDA — A5

SCL — A7

RS — B8

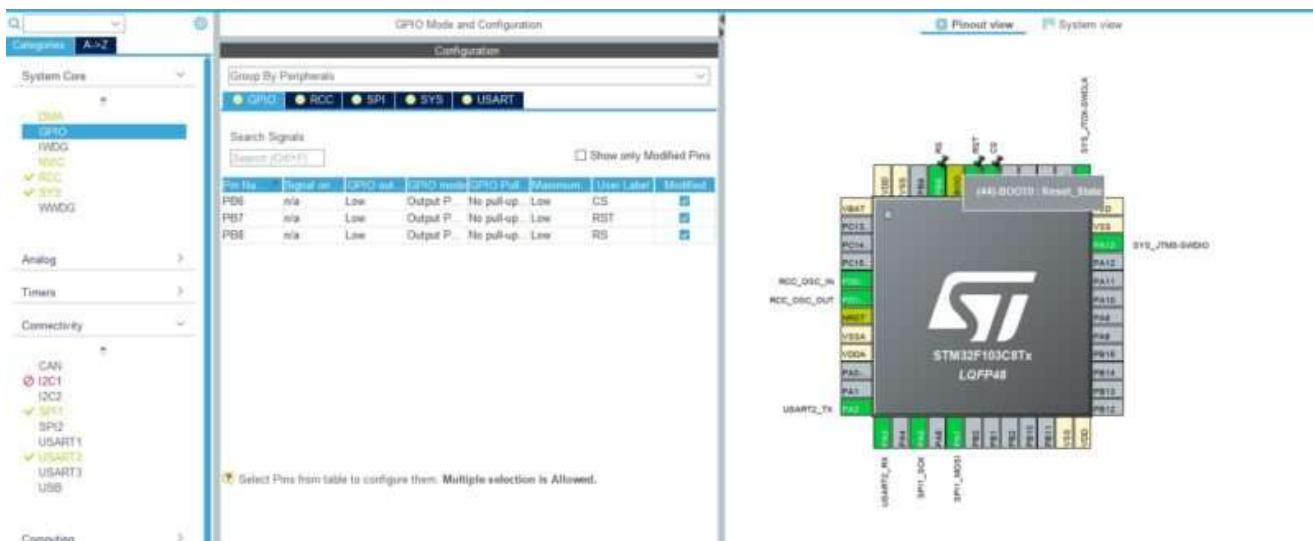
RST — B7

CS — B6

Cấu hình cubemx

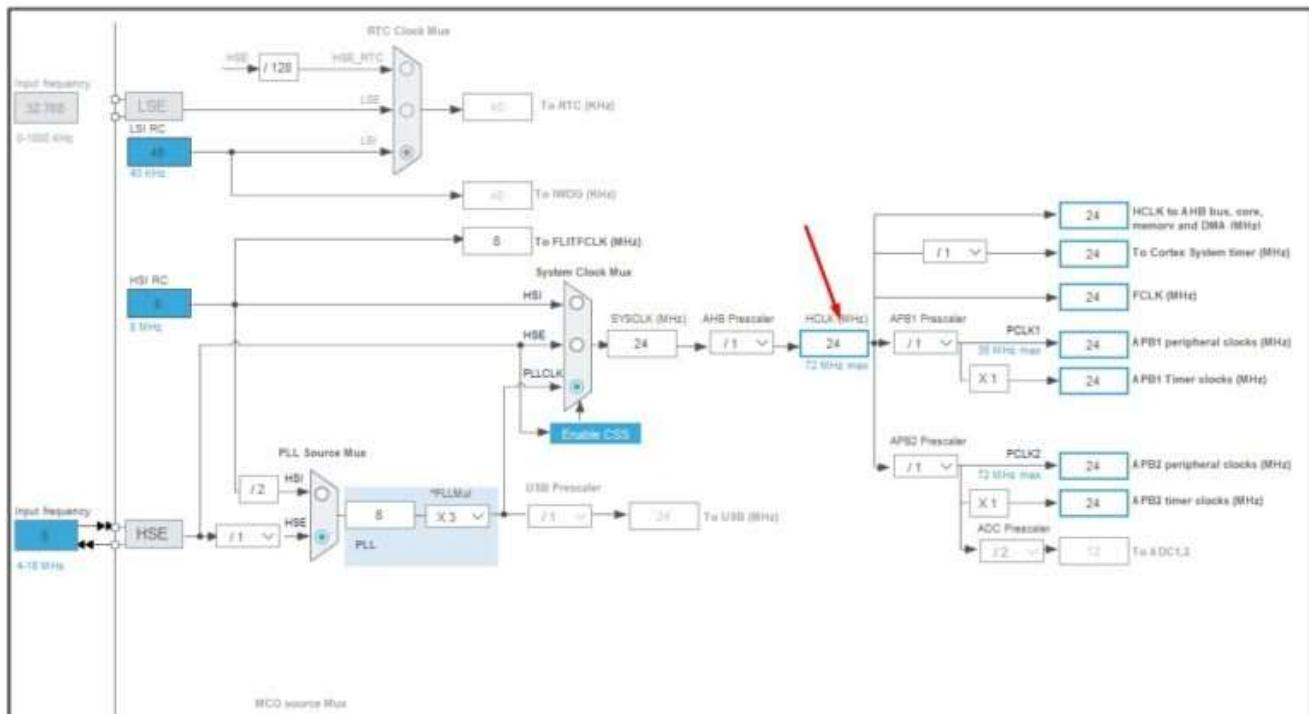
Mở cubemx, chọn chip stm32f103c8. SYS chọn Debug Serial Wire. RCC chọn Crystal. Quá quen thuộc rồi anh em nhỉ.

GPIO chúng ta tạo ra 3 chân CS, RS, RST để điều khiển LCD



SPI chế độ Trasmit Only master và tham số chọn như ảnh sau. Lưu ý: **First Bit** phải là **MSB** nhé

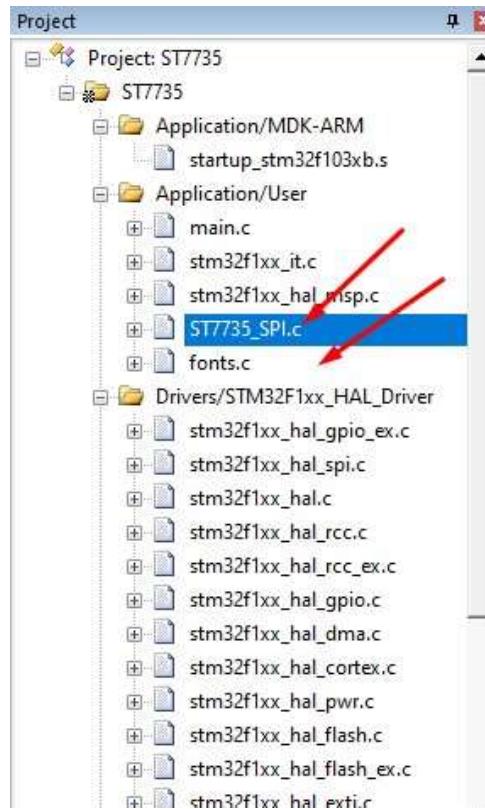
Clock chọn xung là 24Mhz.



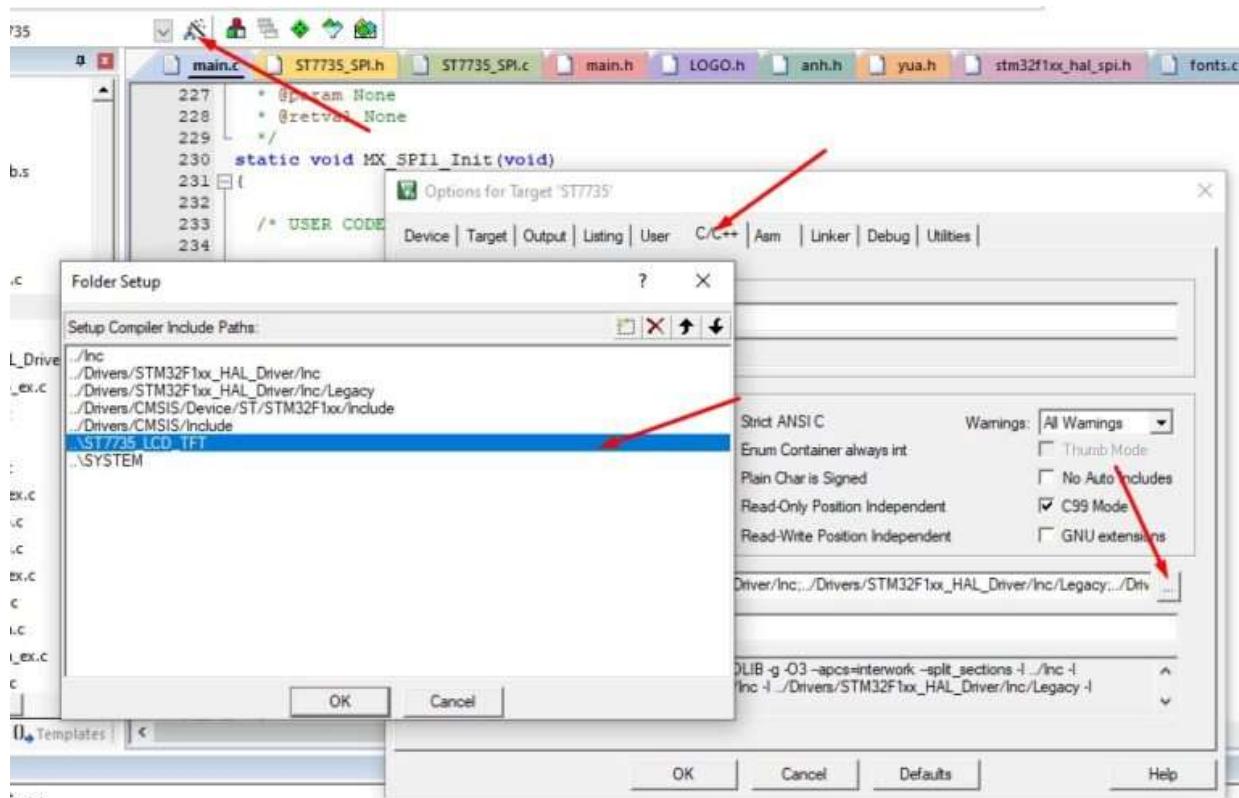
Lập trình STM32 với Keil C

Download thư viện LCD ST7735 theo bài viết: [Hướng dẫn download tài liệu lập trình STM32](#)

Thêm các thư viện vào project.



Chỉnh đường dẫn vào thư viện.



Include thư viện và ảnh trong main.c

```

24  /* Private includes -----
25  /* USER CODE BEGIN Includes */
26 #include <string.h>
27 #include "ST7735_SPI.h"
28 #include "fonts.h"
29 #include "LOGO.h"
30 #include "anh.h"
31 #include "yua.h"
32 /* USER CODE END Includes */
33
34 /* Private typedef -----
35 /* USER CODE BEGIN PTD */
36
37 /* USER CODE END PTD */

```

Khởi tạo ST7735

```

98
99  /* Initialize all configured peripherals */
100 MX_GPIO_Init();
101 MX_DMA_Init();
102 MX_SPI1_Init();
103 MX_USART2_UART_Init();
104 /* USER CODE BEGIN 2 */
105 ST7735_Init();
106 /* USER CODE END 2 */
107
108 /* Infinite loop */
109 /* USER CODE BEGIN WHILE */
110 while (1)

```

Trong While chúng ta sẽ hiển thị ảnh, text, vẽ các đường thẳng...

Full Code

```

02  ST7735_DrawImage(0,0,128, 160,(uint8_t*)gImage_yua);
03  HAL_Delay(3000);
04
05  ST7735_WriteString(20, 100, "Censored", Font_11x18, ST7735_
06  HAL_Delay(3000);
07  ST7735_FillScreen(ST7735_BLACK);
08
09 for(int x = 0; x < ST7735_WIDTH; x++) {
10     ST7735_DrawPixel(x, 0, ST7735_RED);
11     ST7735_DrawPixel(x, ST7735_HEIGHT-1, ST7735_RED);
12 }
13
14 for(int y = 0; y < ST7735_HEIGHT; y++) {
15     ST7735_DrawPixel(0, y, ST7735_RED);
16     ST7735_DrawPixel(ST7735_WIDTH-1, y, ST7735_RED);
17 }
18
19 HAL_Delay(1000);

```

```

20
21 // Check fonts
22 ST7735_FillScreen(ST7735_BLACK);
23 ST7735_WriteString(0, 0, "Font_7x10, red on black, lorem ipsum");
24 ST7735_WriteString(0, 3*10, "Font_11x18, green, lorem ipsum");
25 ST7735_WriteString(0, 3*10+3*18, "Font_16x26", Font_16x26, ST7735_BLACK);
26 HAL_Delay(2000);
27
28 // Check colors
29 ST7735_FillScreen(ST7735_BLACK);
30 ST7735_WriteString(0, 0, "BLACK", Font_11x18, ST7735_WHITE, ST7735_BLACK);
31 HAL_Delay(500);
32
33 ST7735_FillScreen(ST7735_BLUE);
34 ST7735_WriteString(0, 0, "BLUE", Font_11x18, ST7735_BLACK, ST7735_WHITE);
35 HAL_Delay(500);
36
37 ST7735_FillScreen(ST7735_RED);
38 ST7735_WriteString(0, 0, "RED", Font_11x18, ST7735_BLACK, ST7735_WHITE);
39 HAL_Delay(500);
40
41 ST7735_FillScreen(ST7735_GREEN);
42 ST7735_WriteString(0, 0, "GREEN", Font_11x18, ST7735_BLACK, ST7735_WHITE);
43 HAL_Delay(500);
44
45 ST7735_FillScreen(ST7735_CYAN);
46 ST7735_WriteString(0, 0, "CYAN", Font_11x18, ST7735_BLACK, ST7735_WHITE);
47 HAL_Delay(500);
48
49 ST7735_FillScreen(ST7735_MAGENTA);
50 ST7735_WriteString(0, 0, "MAGENTA", Font_11x18, ST7735_BLACK, ST7735_WHITE);
51 HAL_Delay(500);
52
53 ST7735_FillScreen(ST7735_YELLOW);
54 ST7735_WriteString(0, 0, "YELLOW", Font_11x18, ST7735_BLACK, ST7735_WHITE);
55 HAL_Delay(500);
56
57 ST7735_FillScreen(ST7735_WHITE);
58 ST7735_WriteString(0, 0, "WHITE", Font_11x18, ST7735_BLACK, ST7735_WHITE);
59 HAL_Delay(500);

```



Fill màn hình: **ST7735_FillScreen**

Hiển thị text: **ST7735_WriteString**

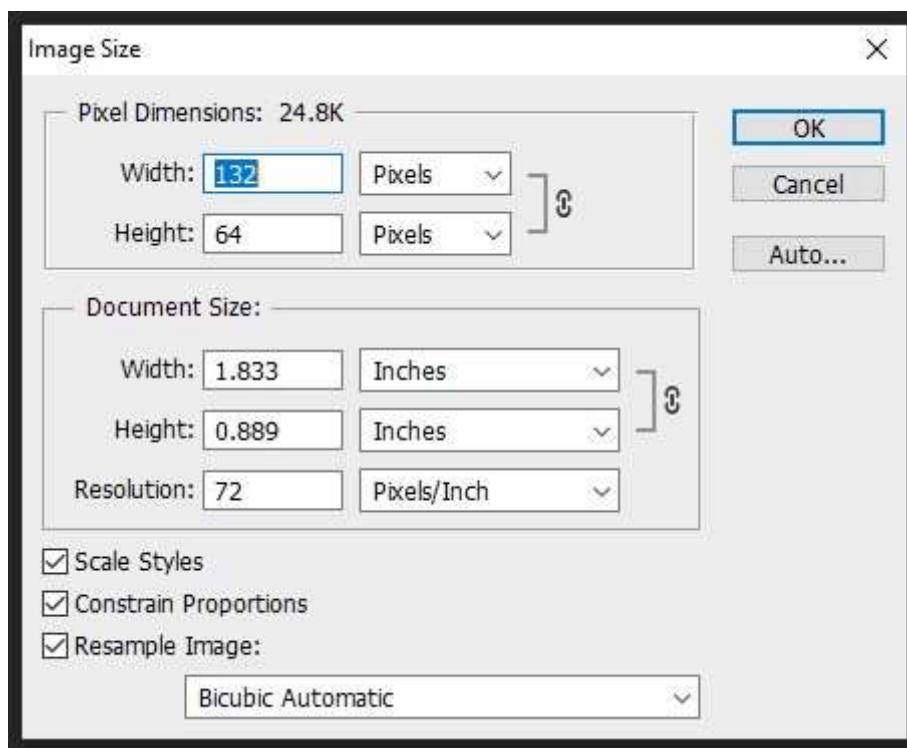
Vẽ điểm ảnh: **ST7735_DrawPixel**

Hiển thị ảnh: **ST7735_DrawImage**

Cách chuyển ảnh sang mảng để hiển thị

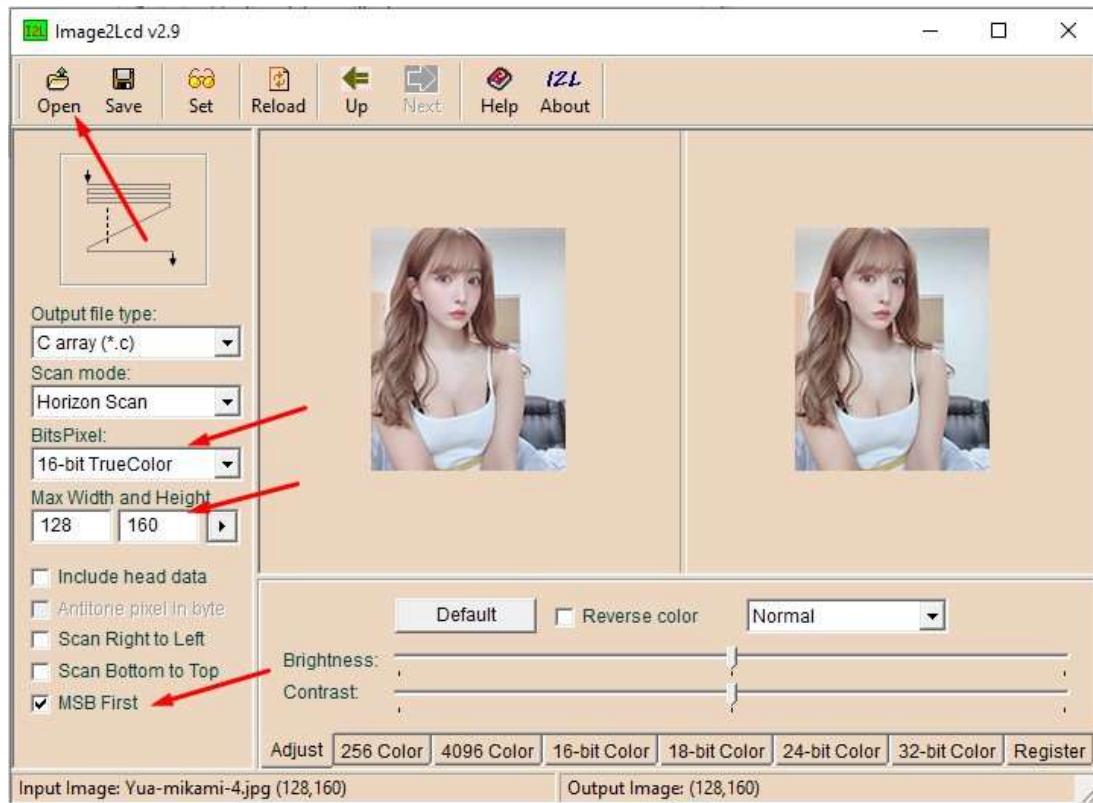
Để thay đổi ảnh chúng ta sử dụng công cụ Image2LCD mà mình cũng đã đề cập tại bài LCD OLED SSD1306. Link: <https://www.fshare.vn/file/Q8NDJBO9ZU2F>

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)



Mở tools Image2LCD

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 trả 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.

Lưu ý: Phải tích chọn vào MSB cho phù hợp với kiểu truyền dữ liệu của SPI nhé

Kết quả



Lập trình STM32 điều khiển tft lcd st7735

Khuê Nguyễn Creator - Học Lập Trình Vi Điều Khiển

Chia sẻ

Facebook Watch

Kết

Viết Driver cho màn hình TFT LCD ST7735 cũng khá phức tạp, do có rất nhiều lệnh. Nhưng nguyên lý chung trong lập trình LCD thì không đổi. Bạn chỉ cần nắm rõ những nguyên lý đó. Thì việc lập trình cũng trở nên đơn giản hơn rất nhiều.

Nếu bạn thấy bài viết này có ích hãy để lại bình luận và đừng quên ra nhập **Hội Anh Em Nghiện Lập trình** nhé.

5/5 - (1 bình chọn)

Related Posts:

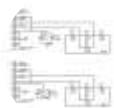
1. [Lập trình STM32 đọc nhiệt độ với DS18b20 giao tiếp OneWire](#)
2. [Hướng dẫn download và sử dụng tài liệu Lập trình STM32](#)
3. [Bài 14: Sử dụng STM32 IWDG Independent Watchdog Timer chống treo vi điều khiển](#)
4. [Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231](#)
5. [Bài 3: Lập 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

5 THOUGHTS ON “LẬP TRÌNH STM32 VỚI TFT LCD ST7735 GIAO THỨC SPI”



chung luong says:

hay quá bạn !

02/08/2021 AT 4:20 CHIỀU

TRẢ LỜI



Khuê Nguyễn says:

Cám ơn bạn

02/08/2021 AT 9:16 CHIỀU

TRẢ LỜI



phạm lê trí tuệ says:

nhấn reset chạy lại từ đầu bị treo vì điều khiển do nguyên nhân vì sao anh

30/08/2021 AT 9:14 CHIỀU

TRẢ LỜI



Khuê Nguyễn says:

bạn vào chế độ debug, run từng lệnh sẽ biết nó bị treo ở đâu

04/09/2021 AT 4:07 CHIỀU

TRẢ LỜI

Bao.N says:

Hay quá anh ơi , nhưng màn hình của em nó cứ nháy liên tục vậy , làm sao fix vậy anh

18/09/2021 AT 8:54 CHIỀU

TRẢ LỜI

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



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



Khuê Nguyễn Creator



PROTEUS

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

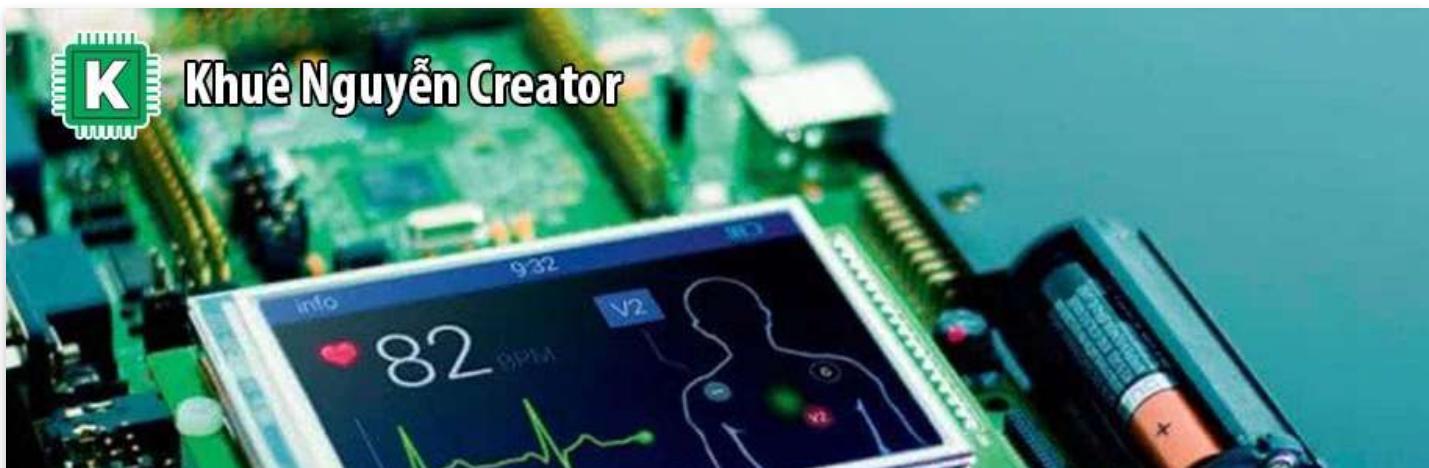


Lập trình STM32 HID Host giao tiếp với chuột và bàn phím

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, biến STM32 giống như...

[ĐỌC THÊM](#)





Lộ trình học lập trình nhúng từ A tới Z

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 STM32 và CubeMX



Lập trình STM32F407 SDIO đọc dữ liệu thẻ nhớ



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

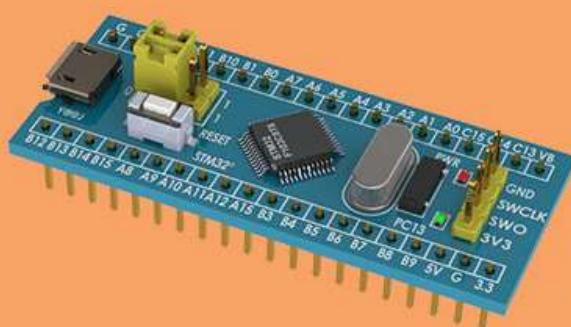
Trong bài này chúng ta sẽ cùng nhau tìm hiểu STM32 DAC với KIT STM32F407VE...

[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