

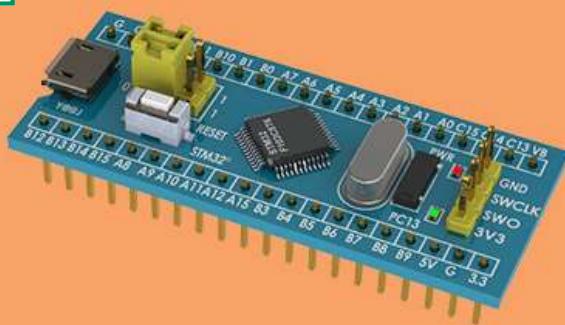
LẬP TRÌNH STM32

Lập trình STM32 điều khiển LCD1602 chế độ 8bit và 4bit

POSTED ON 25/07/2021 BY KHUÊ NGUYỄN

25
Th7

Lập trình STM32 và CubeMX

**Khuê Nguyễn Creator**

Lập trình STM32 điều khiển LCD1206 chế độ 8bit và 4bit

LCD1602 được sử dụng rất nhiều trong lập trình, tuy rằng nó ít được sử dụng trong các sản phẩm thương mại thế nhưng để làm quen về LCD thì sử dụng LCD1602 rất phù hợp.

Trong bài này chúng ta sẽ học các sử dụng STM32 điều khiển LCD1602 chế độ 8bit và 4bit.

Bài viết nằm trong Serie [Lập trình STM32 từ A tới Z](#)

Mục Lục

1. Tổng quan về LCD1602

1.1. LCD là gì?

1.2. LCD1602 overview

1.3. Sơ đồ chân – Pin Out

2. Cách điều khiển LCD1602

2.1. Cấu tạo và cách hoạt động của LCD1602

2.2. Cách đọc và ghi vào LCD1602

2.3. Các lệnh trong LCD1602

2.4. Quy trình khởi tạo LCD1602

2.5. Cách kết nối phần cứng STM32 với LCD1602

3. Lập trình STM32 điều khiển LCD1602 chế độ 8bit

3.1. Khởi tạo LCD1602 chế độ 8bit trên CubeMX

3.2. Lập trình STM32 LCD1602 chế độ 8bit

3.3. Kết quả

3.4. Lập trình STM32 LCD1602 chế độ 4bit

4. Kết

4.1. Related posts:



Tổng quan về LCD1602

LCD là gì?

LCD là một loại vật chất phản xạ ánh sáng khi điện thế thay đổi. Nó hoạt động dựa trên nguyên tắc ánh sáng nền (Back Light). Nó bao gồm một lớp chất lỏng nằm giữa 2 lớp kính phân cực ánh sáng. Bình thường, khi không có điện áp, các tinh thể này được xếp thẳng hàng giữa hai lớp cho phép ánh sáng truyền qua theo hình xoắn ốc. Hai bộ lọc phân cực, 2 bộ lọc màu và 2 bộ cân chỉnh sẽ xác định cường độ ánh sáng đi qua và màu nào được tạo ra trên một pixel.

Khi có điện áp cấp vào, lớp canh chỉnh sẽ tạo một vùng điện tích, canh chỉnh lại các tinh thể lỏng đó. Nó không cho phép ánh sáng đi qua để hiện thị lên hình ảnh tại vị trí điểm ảnh đó. Các điểm ảnh trong màn hình LCD là một transistor cực nhỏ ở một trong 2 chế độ: cho phép ánh sáng đi qua hoặc không.

Điểm ảnh bao gồm 3 yếu tố màu: đỏ, xanh lá, xanh dương. Các màn hình LCD trước đây thường tiêu thụ điện năng nhiều, độ tương phản thấp cho đến khi các nhà khoa học người Anh tìm ra “Biphenyl” – vật liệu chính của tinh thể lỏng, thì LCD mới thực sự phổ biến. LCD xuất hiện đầu tiên trong các máy tính cầm tay, trò chơi điện tử cầm tay, đồng hồ điện tử,...

Theo [Wikimedia](#)

LCD1602 overview

- Điện áp hoạt động là 5 V.
- Kích thước: 80 x 36 x 12.5 mm
- Chữ đen, nền xanh lá
- Khoảng cách giữa hai chân kết nối là 0.1 inch tiện dụng khi kết nối với Breadboard.
- Tên các chân được ghi ở mặt sau của màn hình LCD hỗ trợ việc kết nối, đi dây điện.
- Có đèn led nền, có thể dùng biến trở hoặc PWM điều chỉnh độ sáng để sử dụng ít điện năng hơn.
- Có thể được điều khiển với 6 dây tín hiệu



Sơ đồ chân – Pin Out

1. VSS: tương đương với GND – cực âm
2. VDD: tương đương với VCC – cực dương (5V)
3. Contrast Voltage (Vo): điều khiển độ sáng màn hình
4. Register Select (RS): điều khiển địa chỉ nào sẽ được ghi dữ liệu
5. Read/Write (RW): Bạn sẽ đọc (read mode) hay ghi (write mode) dữ liệu? Nó sẽ phụ thuộc vào bạn gửi giá trị gì vào.
6. Enable pin: Cho phép ghi vào LCD
7. D0 – D7: 8 chân dữ liệu, mỗi chân sẽ có giá trị HIGH hoặc LOW nếu bạn đang ở chế độ đọc (read mode) và nó sẽ nhận giá trị HIGH hoặc LOW nếu đang ở chế độ ghi (write mode)
8. Backlight (Backlight Anode (+) và Backlight Cathode (-)): Tắt bật đèn màn hình LCD.

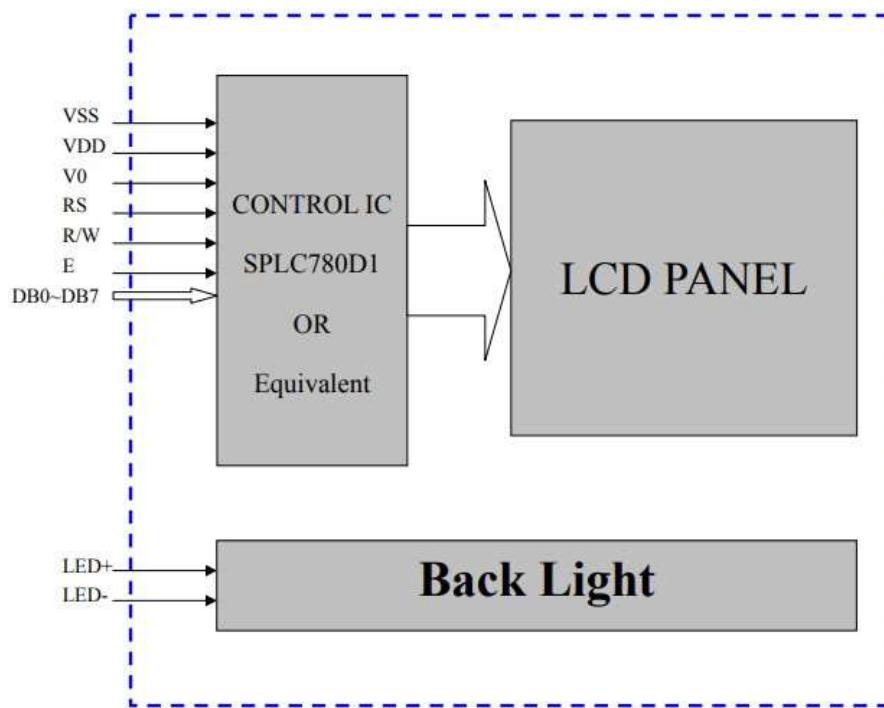
Cách điều khiển LCD1602

Cấu tạo và cách hoạt động của LCD1602

Tham khảo datasheet: <https://cdn-shop.adafruit.com/datasheets/TC1602A-01T.pdf>

Màn hình LCD1602 cấu tạo từ 3 phần:

- Chip driver: Điều khiển LCD, giao tiếp với vi điều khiển theo interface LCD
- LCD Panel: Hiển thị ra bên ngoài
- Back Light: Đèn nền LCD



Sơ đồ khối LCD1602

DISPLAY DATA RAM (DD RAM): Bộ nhớ hiển thị dữ liệu

Điều khiển LCD1602 chính là thay đổi giá trị của DD RAM, mỗi ô trên DD RAM tương ứng với một vị trí của màn hình.

Với LCD1602 chúng có 2 line

- Line 1: từ 0x80 tới 0x8F
- Line 2: từ 0xC0 tới 0xCF

Ví dụ: Nếu bạn muốn hiển thị ở Line 1 ô đầu tiên, Chúng ta sẽ thay đổi giá trị của ô nhớ địa chỉ 0x80. Giá trị được ghi sẽ so sánh với bảng mã trong CG ROM, từ đó hiển thị ra đúng kí tự được lưu trên đó.

Lệnh để nhảy giữa các ô nhớ là Set cursor (con trỏ)

Character Generator ROM (CG ROM): Bộ nhớ kí tự chỉ đọc

Đây là bộ nhớ được ghi sẵn của LCD, trong đó chứa các kí tự mà lcd hỗ trợ. Có hai mẫu Character mà LCD1602 hỗ trợ đó là 5x8 và 5x10

Character Generator RAM (CG RAM): Bộ nhớ kí tự có thể lập trình

Đây là bộ nhớ để người sử dụng có thể tự tạo ra các font chữ riêng trên LCD của mình.

2 LINES X 16 CHARACTERS PER LINE																
Char.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Line 1	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
Line 2	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF

Sơ đồ Display RAM LCD1602

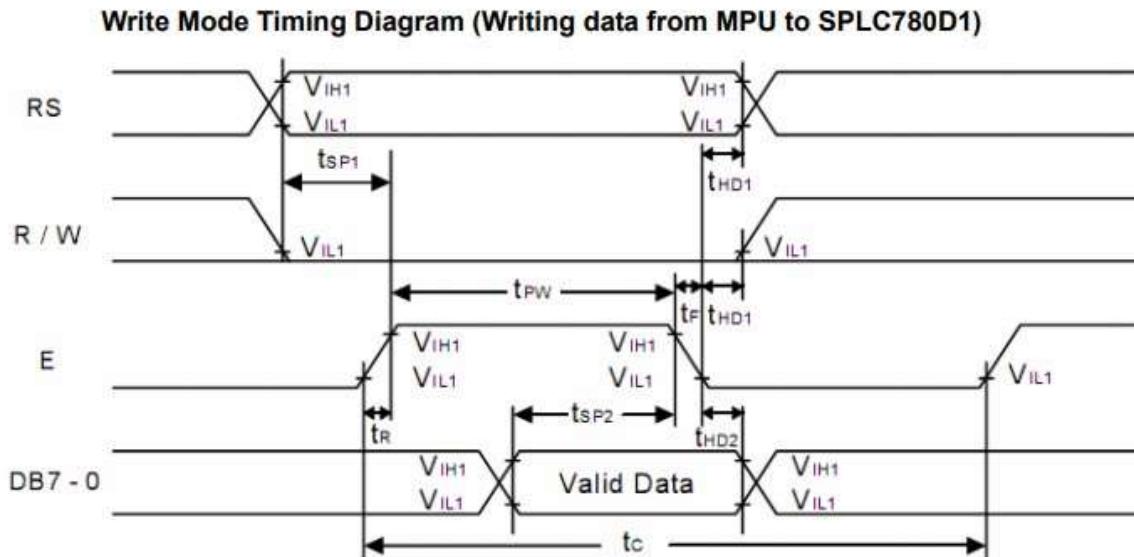
Cách đọc và ghi vào LCD1602

Quy trình ghi vào LCD1602 như sau:

1. Chân RS kéo xuống 0 nếu gửi Lệnh (Command) điều khiển LCD, lên 1 gửi data (ghi vào DD RAM)
2. Chân R/W: Kéo xuống 0 ghi dữ liệu

3. Các chân D0 – D7: Khi ghi dữ liệu, các chân D0-D7 của LCD sẽ ở chế độ Input, chân MCU kết nối vào sẽ ở chế độ OutPut
4. Chân EN sẽ được kéo lên 1 để chốt dữ liệu vào LCD, sau đó lại nhả về 0

Khi lập trình chúng ta sẽ sử dụng bảng Timing để delay thời gian cho phù hợp.



Timing Diagram Write Mode – Cách ghi dữ liệu vào LCD

Characteristics	Symbol	Limit			Unit	Test Condition
		Min.	Typ.	Max.		
E Cycle Time	t _c	400	-	-	ns	Pin E
E Pulse Width	t _{PW}	150	-	-	ns	Pin E
E Rise/Fall Time	t _R , t _F	-	-	25	ns	Pin E
Address Setup Time	t _{SP1}	30	-	-	ns	Pins: RS, R/W, E
Address Hold Time	t _{HD1}	10	-	-	ns	Pins: RS, R/W, E
Data Setup Time	t _{SP2}	40	-	-	ns	Pins: DB0 - DB7
Data Hold Time	t _{HD2}	10	-	-	ns	Pins: DB0 - DB7

Bảng Timing ghi vào LCD1602

Trong STM32 chúng ta sẽ lập trình như sau,

Lưu ý: Trong thư viện này mình chỉ sử dụng chế độ Write, vậy nên chân R/W mặc định đã được nối xuống 0 rồi nhé!

```

static void CLCD_Write8(CLCD_Name* LCD, uint8_t Data, uint8_t Mode)
{
    if(Mode == CLCD_COMMAND)
    {
        HAL_GPIO_WritePin(LCD->RS_PORT, LCD->RS_PIN, GPIO_PIN_RESET); // RS = 0, write cmd
    }
    else if(Mode == CLCD_DATA)
    {
        HAL_GPIO_WritePin(LCD->RS_PORT, LCD->RS_PIN, GPIO_PIN_SET); // RS = 1, write data
    }
    HAL_GPIO_WritePin(LCD->D0_PORT, LCD->D0_PIN, Data&0x01?GPIO_PIN_SET:GPIO_PIN_RESET);
    HAL_GPIO_WritePin(LCD->D1_PORT, LCD->D1_PIN, Data>>1&0x01?GPIO_PIN_SET:GPIO_PIN_RESET);
    HAL_GPIO_WritePin(LCD->D2_PORT, LCD->D2_PIN, Data>>2&0x01?GPIO_PIN_SET:GPIO_PIN_RESET);
    HAL_GPIO_WritePin(LCD->D3_PORT, LCD->D3_PIN, Data>>3&0x01?GPIO_PIN_SET:GPIO_PIN_RESET);
    HAL_GPIO_WritePin(LCD->D4_PORT, LCD->D4_PIN, Data>>4&0x01?GPIO_PIN_SET:GPIO_PIN_RESET);
    HAL_GPIO_WritePin(LCD->D5_PORT, LCD->D5_PIN, Data>>5&0x01?GPIO_PIN_SET:GPIO_PIN_RESET);
    HAL_GPIO_WritePin(LCD->D6_PORT, LCD->D6_PIN, Data>>6&0x01?GPIO_PIN_SET:GPIO_PIN_RESET);
    HAL_GPIO_WritePin(LCD->D7_PORT, LCD->D7_PIN, Data>>7&0x01?GPIO_PIN_SET:GPIO_PIN_RESET);

    HAL_GPIO_WritePin(LCD->EN_PORT, LCD->EN_PIN, GPIO_PIN_RESET);
    CLCD_Delay(1);
    HAL_GPIO_WritePin(LCD->EN_PORT, LCD->EN_PIN, GPIO_PIN_SET);
    CLCD_Delay(1);
    HAL_GPIO_WritePin(LCD->EN_PORT, LCD->EN_PIN, GPIO_PIN_RESET);
    CLCD_Delay(1);
}

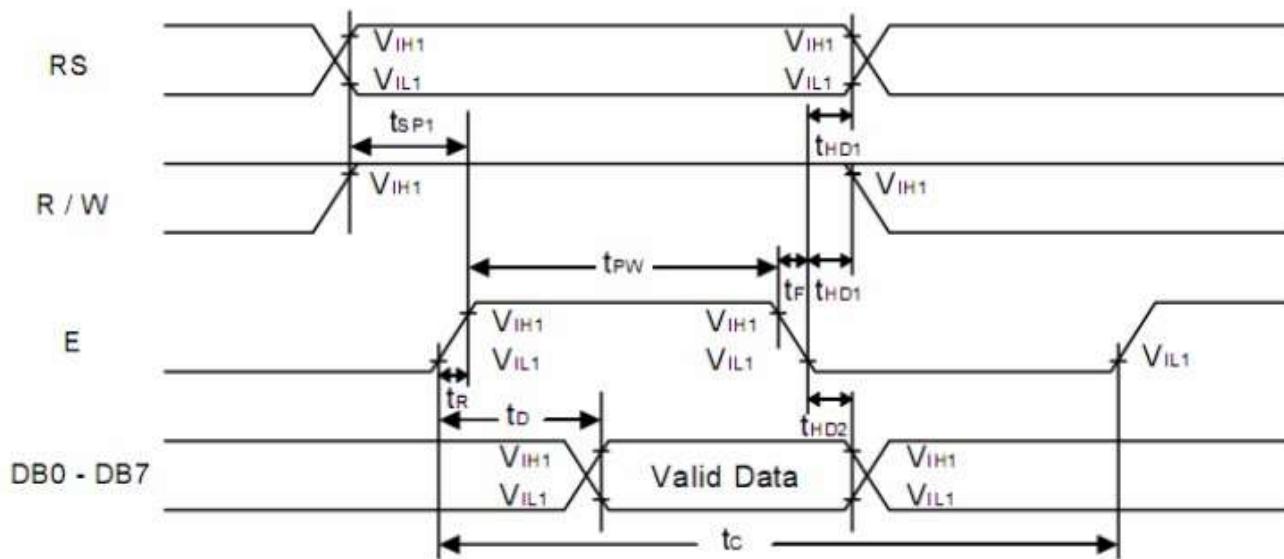
```

Quy trình đọc LCD1602:

- Chân RS kéo xuống 0 nếu gửi Lệnh (Command) điều khiển LCD, lên 1 gửi data (ghi vào DD RAM)
- Chân R/W: Kéo lên 1 để đọc dữ liệu
- Các chân D0 – D7: Khi đọc dữ liệu, các chân D0-D7 của LCD sẽ ở chế độ Output, chân MCU kết nối vào sẽ ở chế độ Input
- Chân EN sẽ được kéo lên 1 để chốt dữ liệu vào LCD, sau đó lại nhả về 0

Khi lập trình chúng ta sẽ sử dụng bảng Timing để delay thời gian cho phù hợp.

Read Mode Timing Diagram (Reading data from SPLC780D1 to MCU)



Timing Diagram Read Mode – Cách đọc dữ liệu từ LCD

(2) Read Mode (Reading data from SPLC780D1 to MPU)

Characteristics	Symbol	Limit			Unit	Test Condition
		Min.	Typ.	Max.		
E Cycle Time	t_c	400	-	-	ns	Pin E
E Pulse Width	t_W	150	-	-	ns	Pin E
E Rise/Fall Time	t_R, t_F	-	-	25	ns	Pin E
Address Setup Time	t_{SP1}	30	-	-	ns	Pins: RS, R/W, E
Address Hold Time	t_{HD1}	10	-	-	ns	Pins: RS, R/W, E
Data Output Delay Time	t_D	-	-	100	ns	Pins: DB0 - DB7
Data hold time	t_{HD2}	5.0	-	-	ns	Pin DB0 - DB7

Các lệnh trong LCD1602

Để ghi các lệnh vào LCD1602 chúng ta sử dụng bảng lệnh sau:

Các bit 1 tương ứng với lệnh, và các bit sau bit 1 tương ứng với tham số.

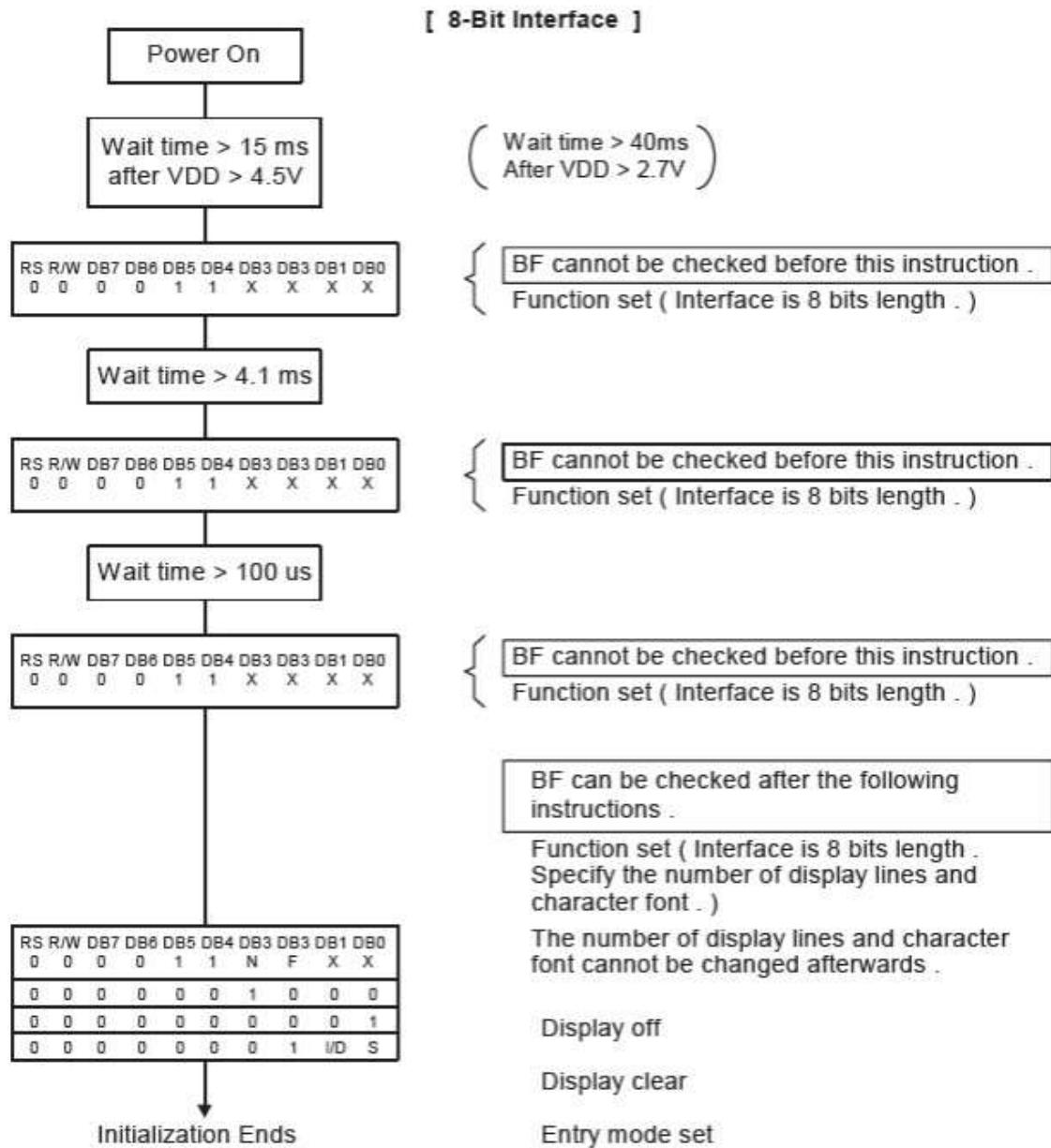
VD: Khi gửi lệnh INPUT SET chúng ta sẽ sử dụng lệnh 0x40, sau đó AND với 2bit I/D và S và set các chân RS và R/W về 0.

COMMAND	COMMAND CODE											COMMAND CODE	E-CYCLE $f_{osc}=250\text{KHz}$					
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0								
SCREEN CLEAR	0	0	0	0	0	0	0	0	0	1	Screen Clear, Set AC to 0 Cursor Reposition	1.64ms						
CURSOR RETURN	0	0	0	0	0	0	0	0	1	*	DDRAM AD=0, Return, Content Changeless	1.64ms						
INPUT SET	0	0	0	0	0	0	0	1	I/D	S	Set moving direction of cursor, Appoint if move	40us						
DISPLAY SWITCH	0	0	0	0	0	0	1	D	C	B	Set display on/off,cursor on/off, blink on/off	40us						
SHIFT	0	0	0	0	0	1	S/C	R/L	*	*	Remove cursor and whole display,DDRAM changeless	40us						
FUNCTION SET	0	0	0	0	1	DL	N	F	*	*	Set DL,display line,font	40us						
CGRAM AD SET	0	0	0	1	ACG						Set CGRAM AD, send receive data	40us						
DDRAM AD SET	0	0	1	ADD							Set DDRAM AD, send receive data	40us						
BUSY/AD READ CT	0	1	BF	AC							Executing internal function, reading AD of CT	40us						
CGRAM/ DDRAM DATA WRITE	1	0	DATA WRITE								Write data from CGRAM or DDRAM	40us						
CGRAM/ DDRAM DATA READ	1	1	DATA READ								Read data from CGRAM or DDRAM	40us						
	I/D=1: Increment Mode; I/D=0: Decrement Mode S=1: Shift S/C=1: Display Shift; S/C=0: Cursor Shift R/L=1: Right Shift; R/L=0: Left Shift DL=1: 8D DL=0: 4D N=1: 2R N=0: 1R F=1: 5x10 Style; F=0: 5x7 Style BF=1: Execute Internal Function; BF=0: Command Received											E-cycle changing with main frequency. Example: If fcp or $f_{osc}=270\text{KHz}$ 40us x 250/270 =37us						
	DDRAM: Display data RAM CGRAM: Character Generator RAM ACG: CGRAM AD ADD: DDRAM AD & Cursor AD AC: Address counter for DDRAM & CGRAM																	

Tập lệnh trong LCD1602

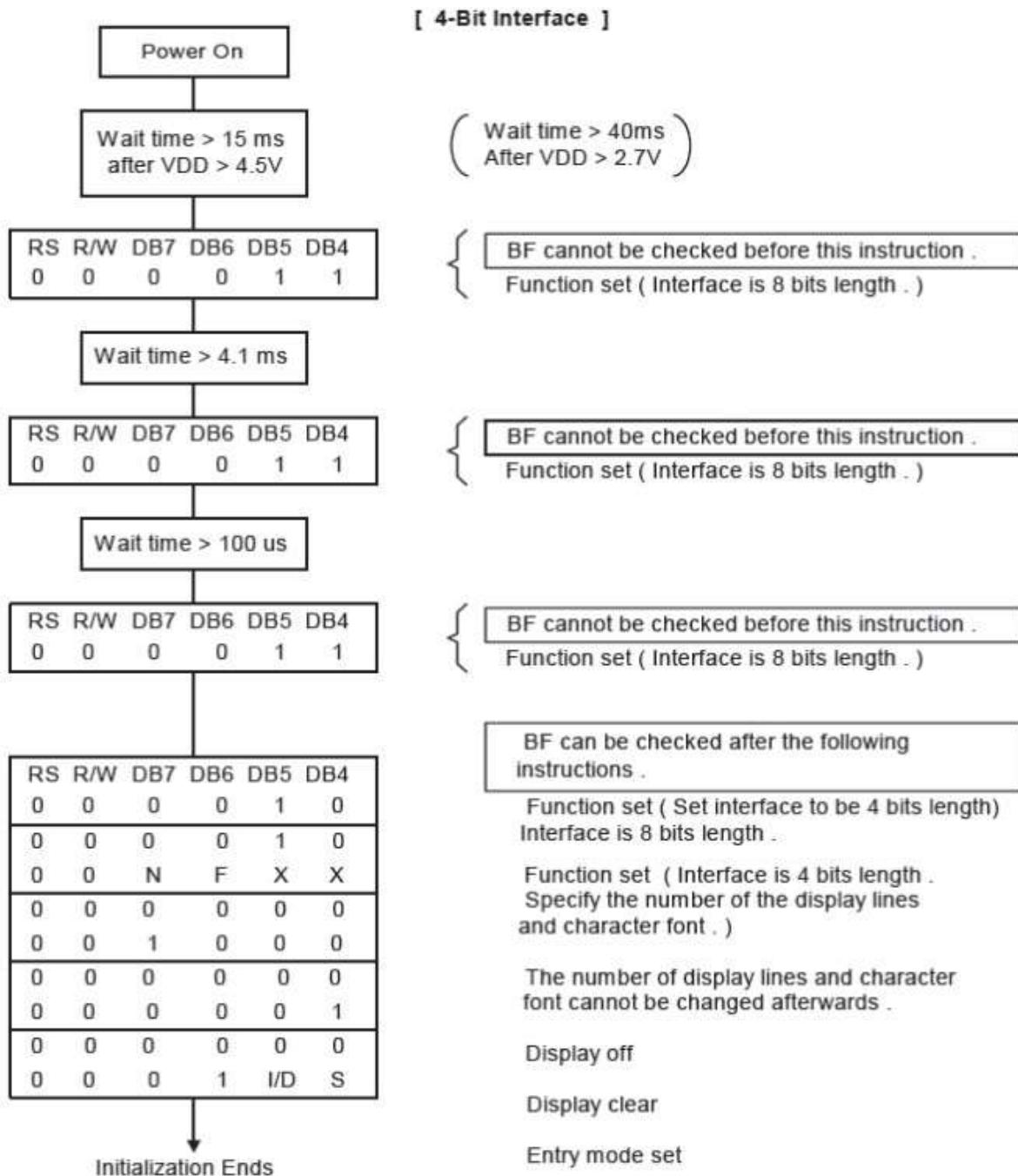
Quy trình khởi tạo LCD1602

Để khởi tạo cho LCD chế độ 8Bit, chúng ta sẽ lập trình theo trình tự sau:



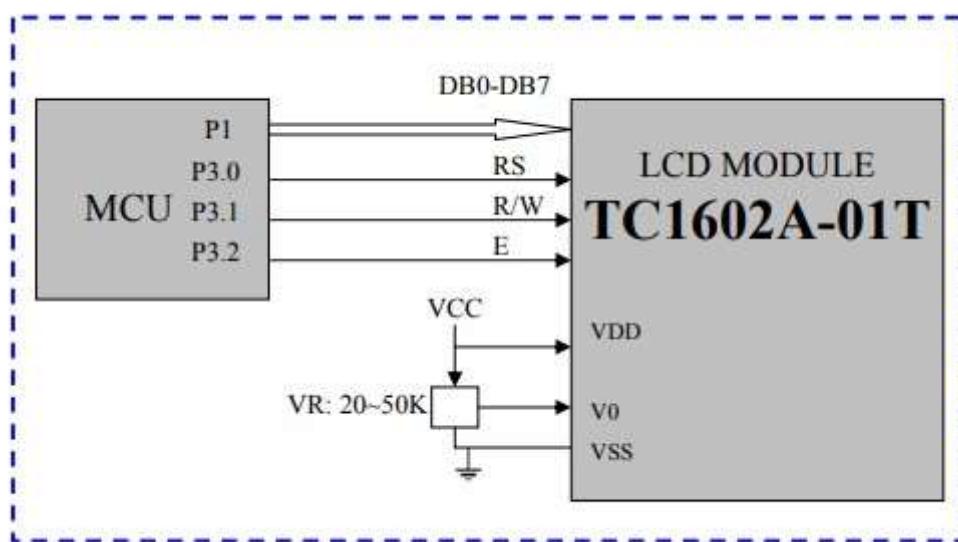
Để khởi tạo LCD1602 chế độ 4 bit chúng ta sẽ lập trình theo trình tự sau:

Với chế độ 4bit, để gửi 1byte (8bit) chúng ta sẽ gửi 2 lần 4bit.



Cách kết nối phần cứng STM32 với LCD1602

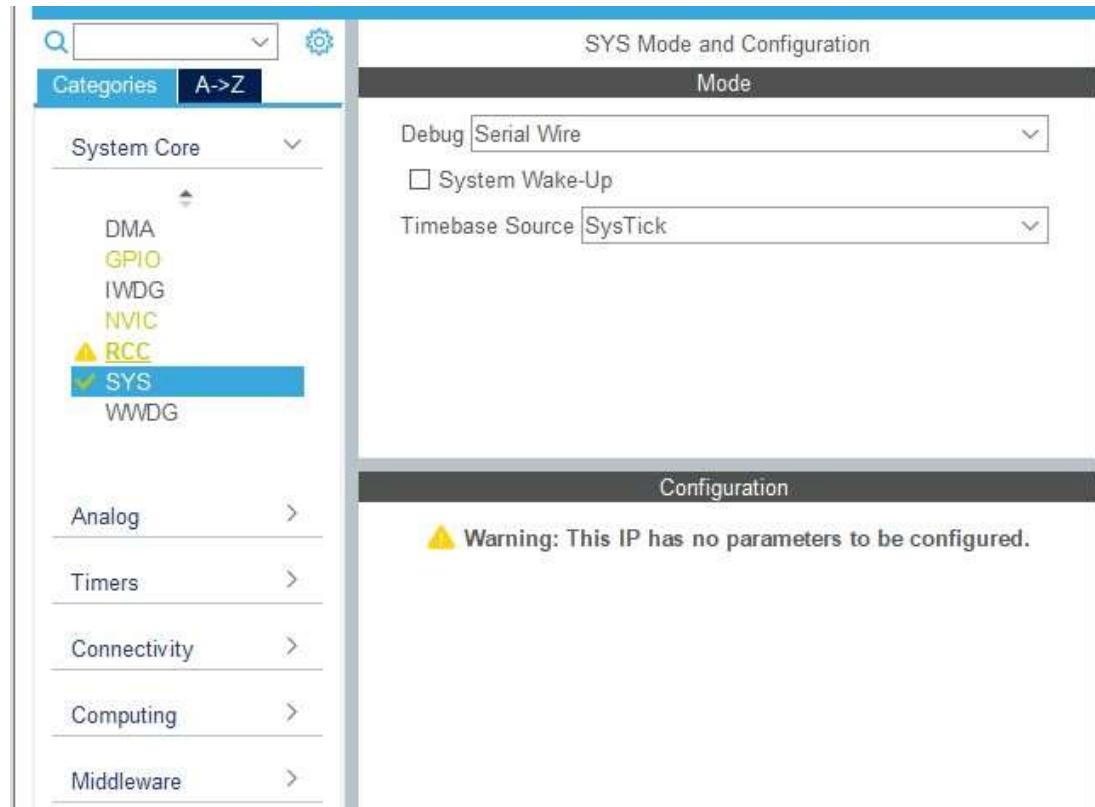
Để điều khiển độ tương phản của LCD chúng ta bắt buộc phải nối chân V0 vào biến trờ hoặc trờ để điều khiển độ tương phản.



Lập trình STM32 điều khiển LCD1602 chế độ 8bit

Khởi tạo LCD1602 chế độ 8bit trên CubeMX

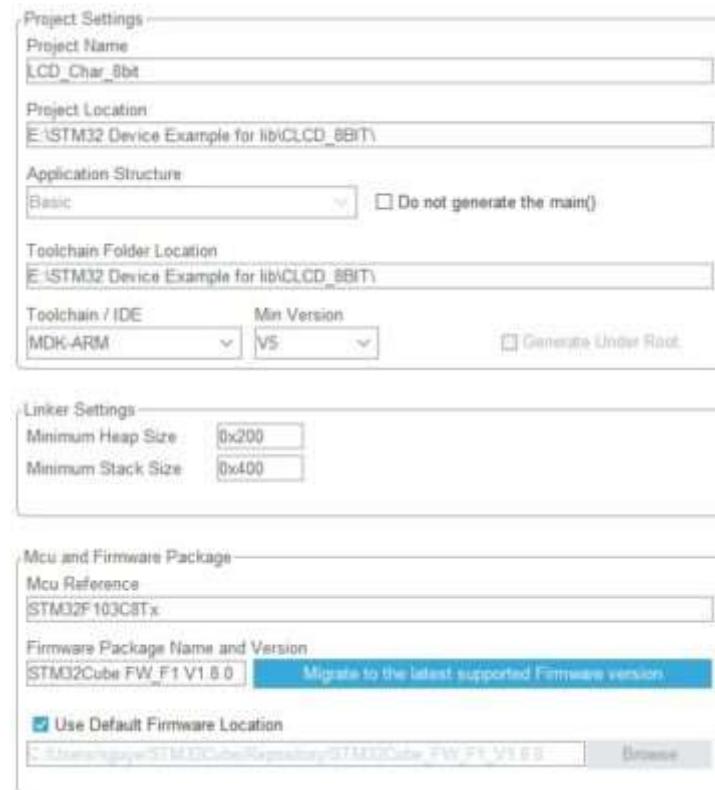
Trong STM32 CubeMx khởi tạo với MCU STM32f103c8t6, trong SYS chọn debug – Serial Wire.



Trong GPIO khởi tạo các chân chế độ Output, đặt tên tương ứng với các chân trên LCD1602.



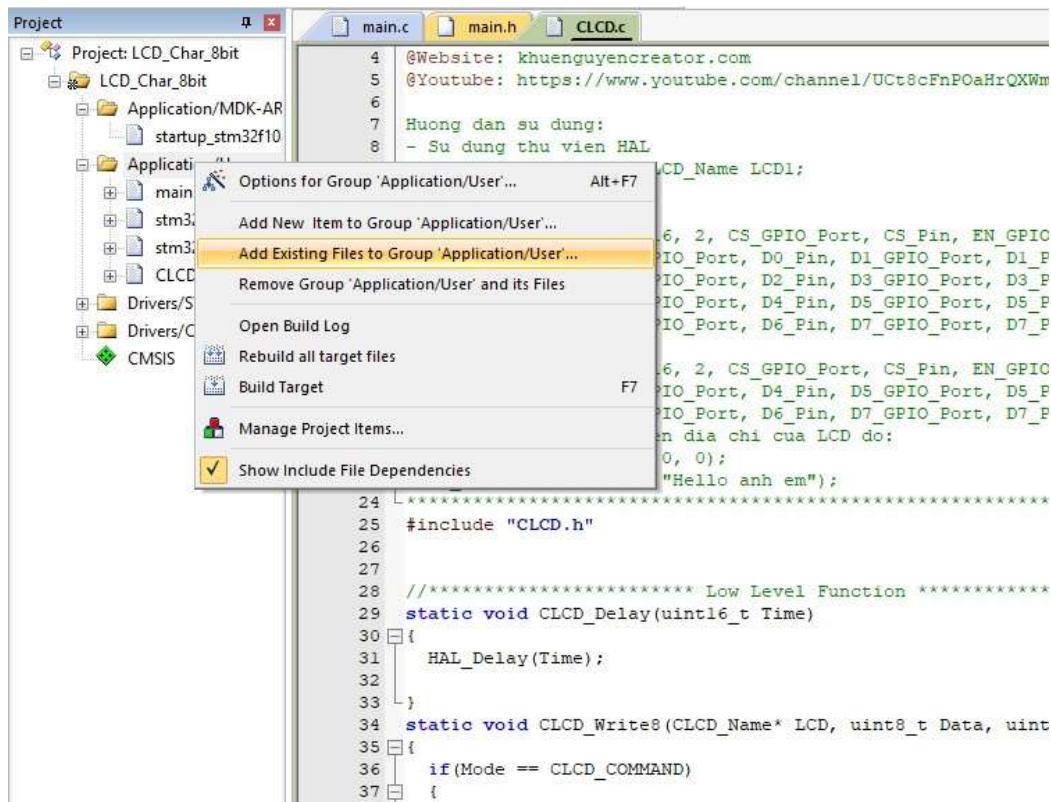
Khởi tạo chọn tools chain là MDK-ARM V5.



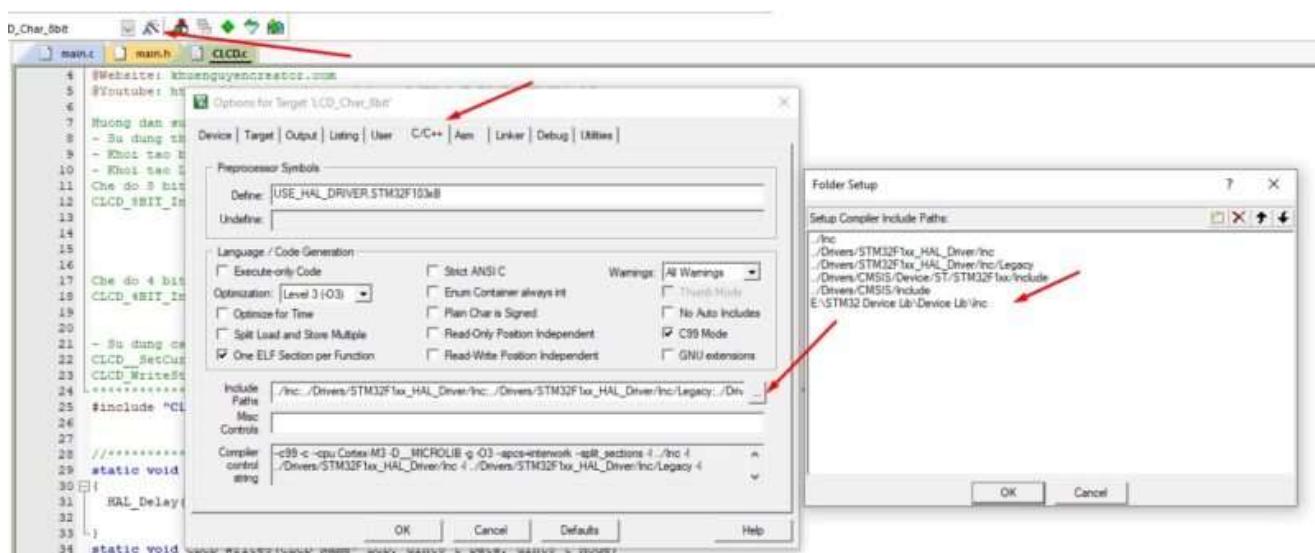
Lập trình STM32 LCD1602 chế độ 8bit

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](#)

Add file CLCD.c vào Project bằng cách nhấn chuột phải -> Add Existing



Thêm đường dẫn vào thư viện mà bạn down về. Nhấn vào biểu tượng Option -> C/C++ -> Inclue Path. Sau đó trỏ tới nơi lưu file CLCD.h



Add CLCD.h vào main.c

```

21  /* Includes -----
22 #include "main.h"
23
24 /* Private includes -----
25 /* USER CODE BEGIN Includes */
26 // #include "CLCD_8BIT.h"
27 #include "CLCD.h"
28 #include <stdio.h>
29 /* USER CODE END Includes */
30
31 /* Private typedef -----
32 /* USER CODE BEGIN PTD */
33
34 /* USER CODE END PTD */
35
36 /* Private define -----
37 /* USER CODE BEGIN PD */
38 /* USER CODE END PD */
39

```

Khởi tạo một LCD tên là LCD1.

Các bạn có thể khởi tạo nhiều LCD trong project của bạn.

Khởi tạo 1 buffer để hiển thị lên LCD và biến Count.

```

47  /* USER CODE BEGIN PV */
48 CLCD_Name LCD1;
49 uint8_t Count;
50 char LCD_send[16];
51 /* USER CODE END PV */
52
53 /* Private function prototypes -----
54 void SystemClock_Config(void);
55 static void MX_GPIO_Init(void);
56 /* USER CODE BEGIN PFP */
57
58 /* USER CODE END PFP */
59
60 /* Private user code -----
61 /* USER CODE BEGIN 0 */
62
63 /* USER CODE END 0 */
64

```

Truyền vào LCD1 các tham số setup như: &LCD1 địa chỉ LCD, số hàng và cột 16x2. Các chân kết nối với LCD

```

 90  /* USER CODE END SYSTEM */
91  /* Initialize all configured peripherals */
92  MX_GPIO_Init();
93  /* USER CODE BEGIN 2 */
94  CLCD_8BIT_Init(&LCD1, 16, 2, CS_GPIO_Port, CS_Pin, EN_GPIO_Port, EN_Pin,
95                  D0_GPIO_Port, D0_Pin, D1_GPIO_Port, D1_Pin,
96                  D2_GPIO_Port, D2_Pin, D3_GPIO_Port, D3_Pin,
97                  D4_GPIO_Port, D4_Pin, D5_GPIO_Port, D5_Pin,
98                  D6_GPIO_Port, D6_Pin, D7_GPIO_Port, D7_Pin);
99
100 /* USER CODE END 2 */
101
102 /* Infinite loop */
103 /* USER CODE BEGIN WHILE */
104 while (1)
105 {

```

Trong While

Đầu tiên nhảy con trỏ tới nơi cần ghi, sau đó ghi giá trị vào ô 0,0

Lệnh sprintf sẽ ghi giá trị của Count vào chuỗi. Sau đó ghi vào LCD.

Tiếp tới tăng giá trị Count và delay 1s.

```

104
105 while (1)
106 {
107     /* USER CODE END WHILE */
108
109     /* USER CODE BEGIN 3 */
110     CLCD_SetCursor(&LCD1, 0, 0);
111     CLCD_WriteString(&LCD1, "Khue Nguyen Creator");
112     sprintf(LCD_send, "LCD Dem : %d", Count);
113     CLCD_SetCursor(&LCD1, 0, 1);
114     CLCD_WriteString(&LCD1, LCD_send);
115     Count++;
116     HAL_Delay(1000);
117 }
/* USER CODE END 3 */

```

Kết quả



Facebook Watch

Lập trình STM32 LCD1602 chế độ 4bit

Với chế độ 4bit, chúng ta chỉ cần khởi tạo lại LCD với chế độ 4bit

Sau đó ghi LCD như bình thường

```

90  /* Initialize all configured peripherals */
91  MX_GPIO_Init();
92  /* USER CODE BEGIN 2 */
93  CLCD_4BIT_Init(&LCD1, 16, 2, CS_GPIO_Port, CS_Pin, EN_GPIO_Port, EN_Pin,
94  D4_GPIO_Port, D4_Pin, D5_GPIO_Port, D5_Pin,
95  D6_GPIO_Port, D6_Pin, D7_GPIO_Port, D7_Pin);
96
97  /* USER CODE END 2 */
98
99  /* Infinite loop */
100 /* USER CODE BEGIN WHILE */
101 while (1)
102 {
103     /* USER CODE END WHILE */
104
105     /* USER CODE BEGIN 3 */
106     CLCD_SetCursor(&LCD1, 0, 0);
107     CLCD_WriteString(&LCD1, "Khue Nguyen Creator");
108
109     sprintf(LCD_send, "LCD Dem: %d", Count);
110     CLCD_SetCursor(&LCD1, 0, 1);
111     CLCD_WriteString(&LCD1, LCD_send);
112     Count++;
113     HAL_Delay(1000);
114 }
```

Kết

Đến đây bạn đã có thể điều khiển LCD1602, sử dụng LCD1602 để hiển thị một số kết quả đo đạc từ sensor hay các linh kiện khác rất đơn giản. Hi vọng bạn đã hiểu rõ cách điều khiển LCD.

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 cứu lập trình tại đây nhé: [Hội anh em nghiên cứu lập trình](#)

5/5 - (4 bình chọn)

Related Posts:

1. [Bản đồ bộ nhớ \(Memory map\) vi điều khiển STM32F103](#)
2. [Bài 16: Lập trình STM32 USB CDC truyền nhận dữ liệu qua cổng COM ảo](#)
3. [Bài 15: Lập trình STM32 Window Watchdog Timer kiểm lỗi phần mềm](#)
4. [Lập trình STM32 từ A tới Z](#)
5. [Bài 12: Lập trình STM32 với giao thức SPI](#)
6. [Bài 11: Lập trình STM32 với Giao thức UART](#)



KHUÊ NGUYỄN

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

3 THOUGHTS ON “LẬP TRÌNH STM32 ĐIỀU KHIỂN LCD1602 CHẾ ĐỘ 8BIT VÀ 4BIT”



Bao.N says:

Em có làm theo a hướng dẫn cả 2 trường hợp thì lcd của em chỉ sáng nền mà không xuất chữ , em sử dụng stm32f4 black board . Em đã đổi thư viện về f4. Có những nguyên do nào không sáng vậy anh .

14/09/2021 AT 8:00 CHIỀU

TRẢ LỜI



Khuê Nguyễn says:

Em xem con biến trở đã vặn chưa, chỉnh độ tương phản ấy

14/09/2021 AT 9:58 CHIỀU

TRẢ LỜI



dương hoàng says:

a có thể nào chỉ cho em vẽ lưu đồ giải thuật cho ct stm32 điều khiển lcd 16.2 hiển thị tên với mssv đc k ạ

25/09/2021 AT 4:34 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 - Học Lập Trình Vi Điều Khiển
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



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 chung ta sẽ cùng học STM32 HID Host, biến STM32 giống như...

[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 STM32 và CubeMX



Khuê Nguyễn Creator





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

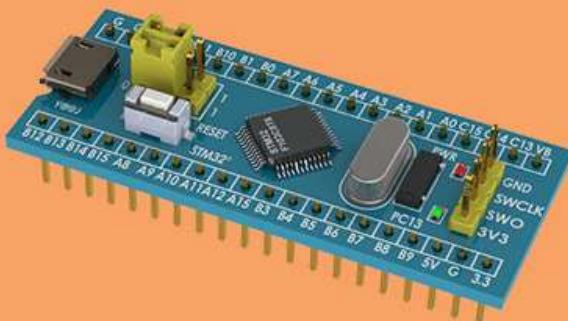
Trong bài này chúng ta sẽ cùng nhau tìm hiểu STM32 DAC với KIT STM32F407VE...

[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