

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 hiển 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,...

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. Constrast 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.

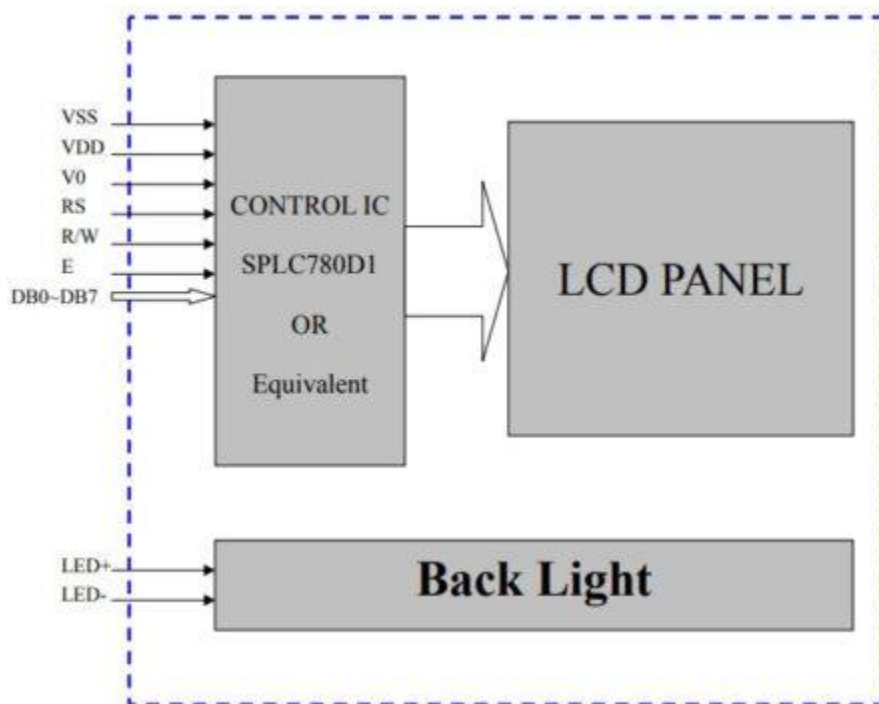
Cách điều khiển LCD1602

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 5×8 và 5×10

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

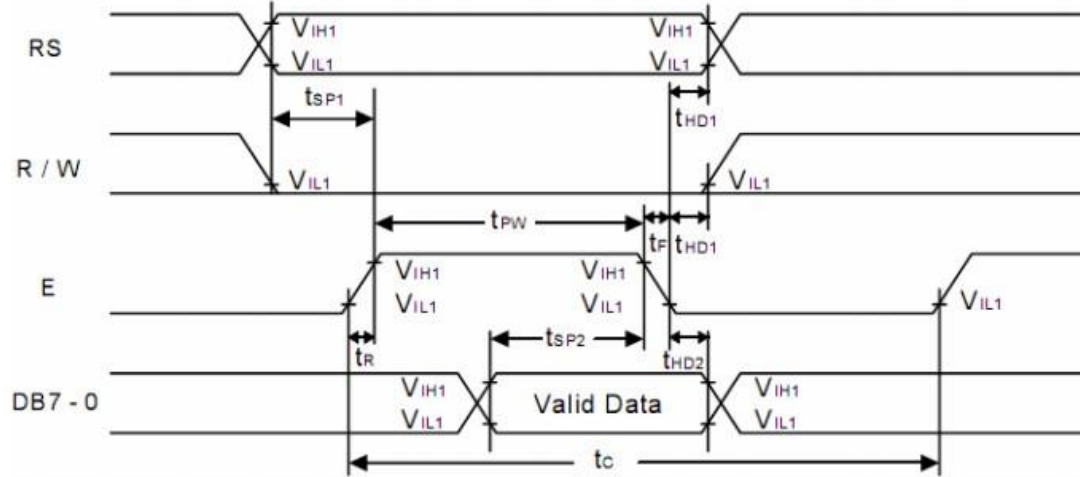
Cách đọc và ghi vào LCD1602

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 thả về 0

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

Write Mode Timing Diagram (Writing data from MPU to SPLC780D1)



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

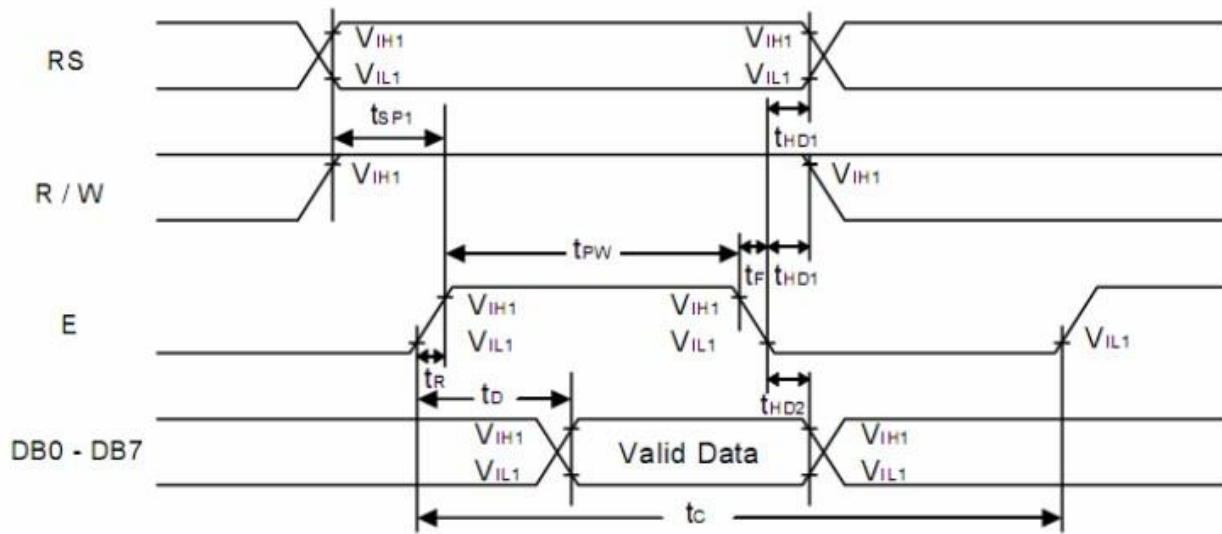
Bảng Timing ghi vào LCD1602

Quy trình đọc LCD1602:

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 lên 1 để đọc dữ liệu
3. 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
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.

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 _{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 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.

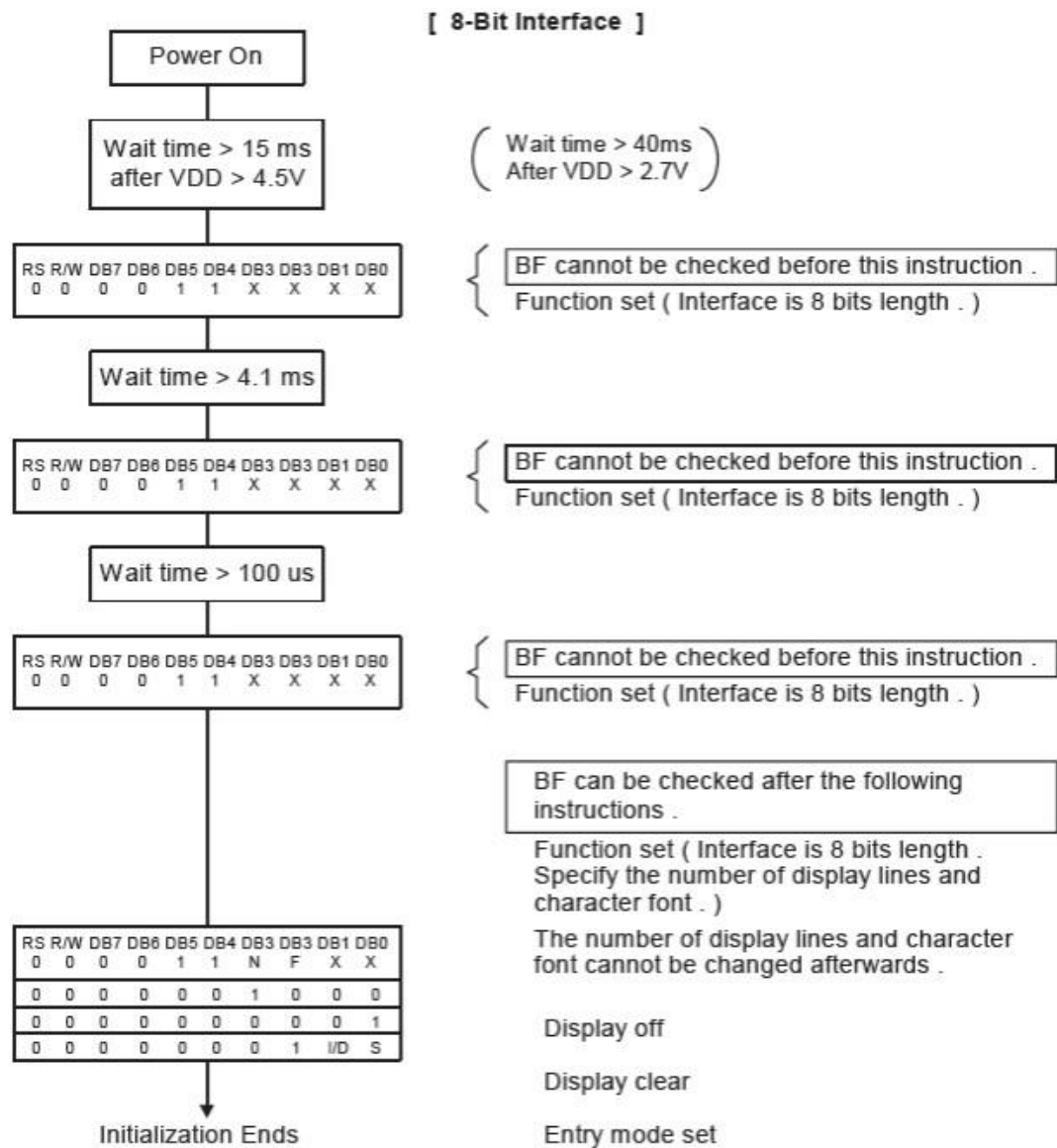
VD: Khi gửi lệnh INPUT SET chúng ta sẽ sử dụng lệnh 0x40, sau đó AND với 2bit I/D và S và set các chân RS và R/W về 0.

COMMAND	COMMAND CODE										COMMAND CODE	E-CYCLE f _{osc} =250KHz
	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										DDRAM: Display data RAM CGRAM: Character Generator RAM ACG: CGRAM AD ADD: DDRAM AD & Cursor AD AC: Address counter for DDRAM & CGRAM	E-cycle changing with main frequency. Example: If fcp or f _{osc} =270KHz 40us x 250/270 =37us

Tập lệnh trong LCD1602

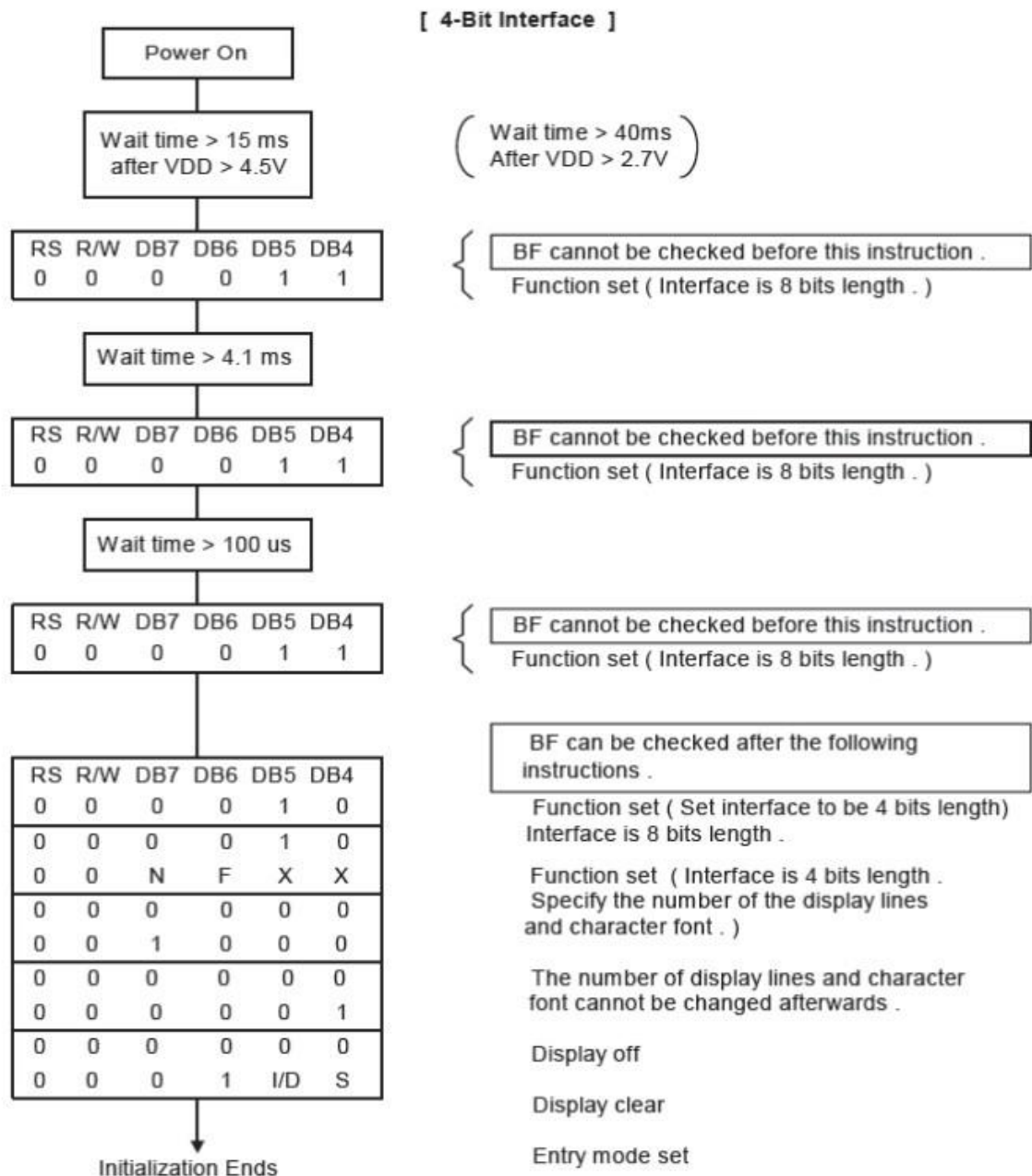
Quy trình khởi tạo LCD1602

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



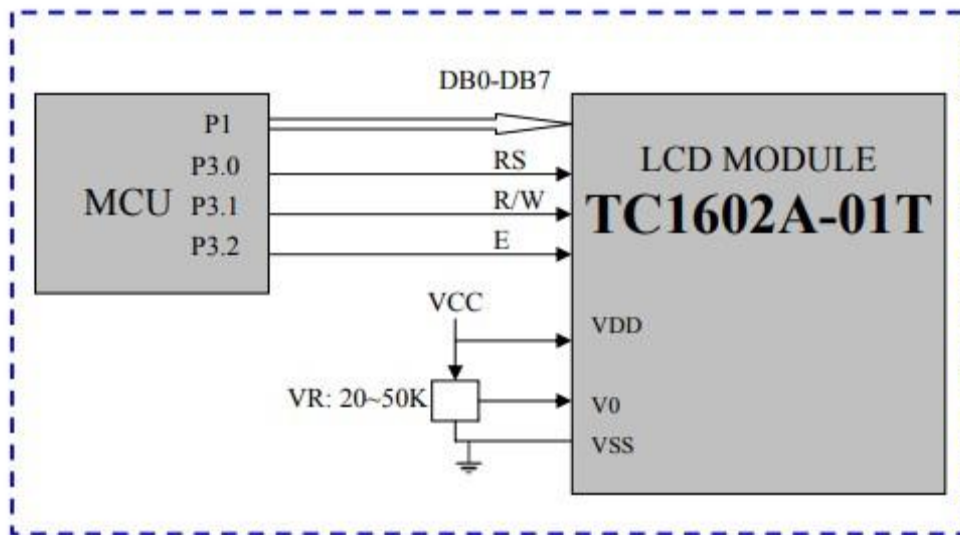
Để khởi tạo LCD1602 chế độ 4 bit chúng ta sẽ lập trình theo trình tự sau:

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



Cách kết nối phần cứng STM32 với LCD1602

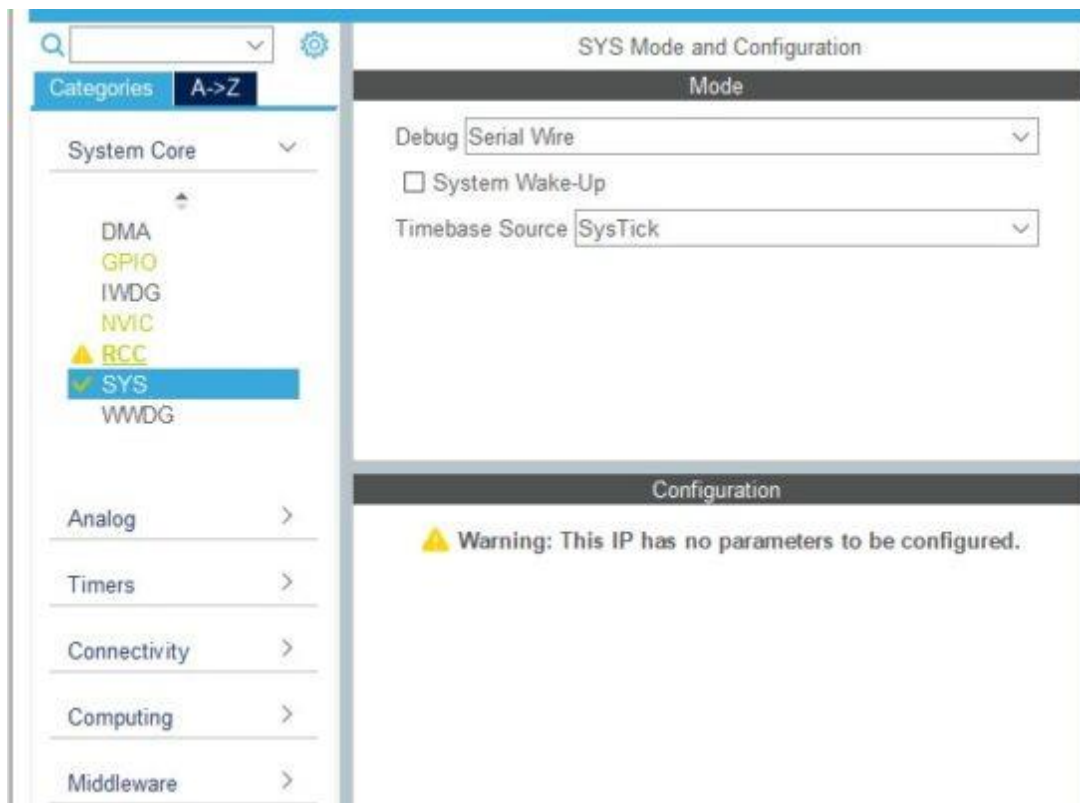
Để điều khiển độ tương phản của LCD chúng ta bắt buộc phải nối chân V0 vào biến trở hoặc trở để điều khiển độ tương phản.



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.



Khởi tạo chọn tools chain là MDK-ARM V5.

Mcu and Firmware Package

Mcu Reference

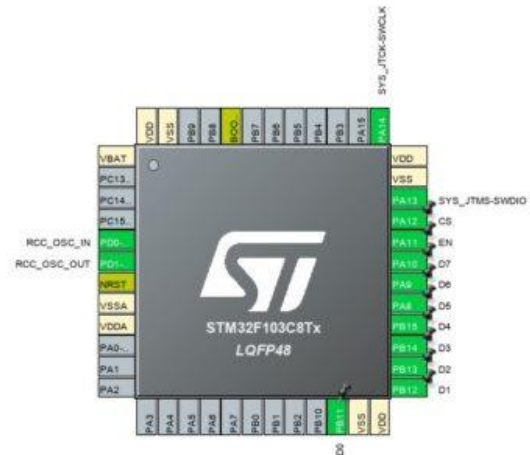
STM32F103C8Tx

Firmware Package Name and Version

STM32Cube_FW_F1_V1.8.0 [Migrate to the latest supported Firmware version](#)

☒ Use Default Firmware Location

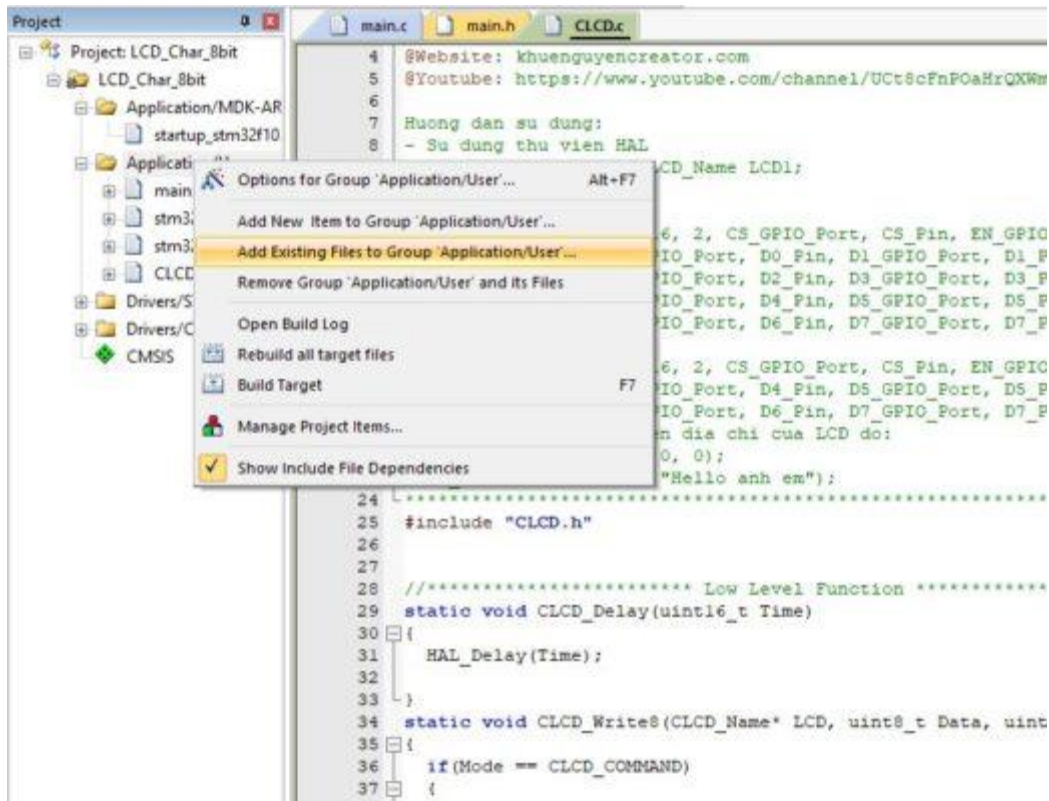
C:\Users\nguyx\STM32Cube\Repository\STM32Cube_FW_F1_V1.8.0 [Browse](#)



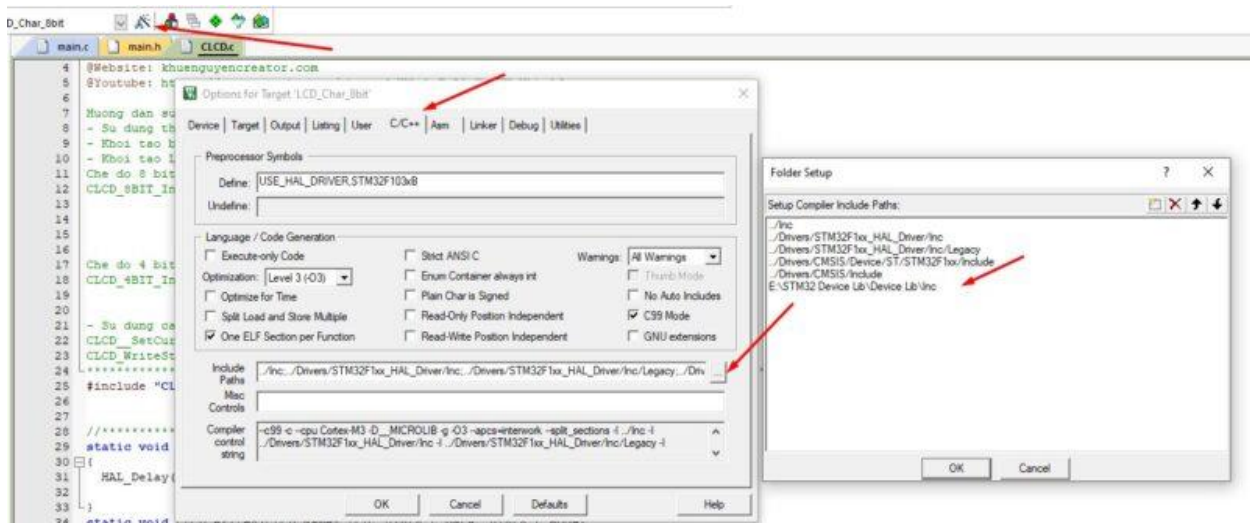
Lập trình STM32 LCD1602 chế độ 8bit

Sau khi Gen code chúng ta sẽ mở bằng Keil C.

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++ -> Include Path. Sau đó trở 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 */
```

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


```

89  /* USER CODE END SystemInit */
90
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 printf sẽ ghi giá trị của Count vào chuỗi. Sau đó ghi vào LCD.

Tiếp tới tăng giá trị Count và delay 1s.

```

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

```

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 }

```