

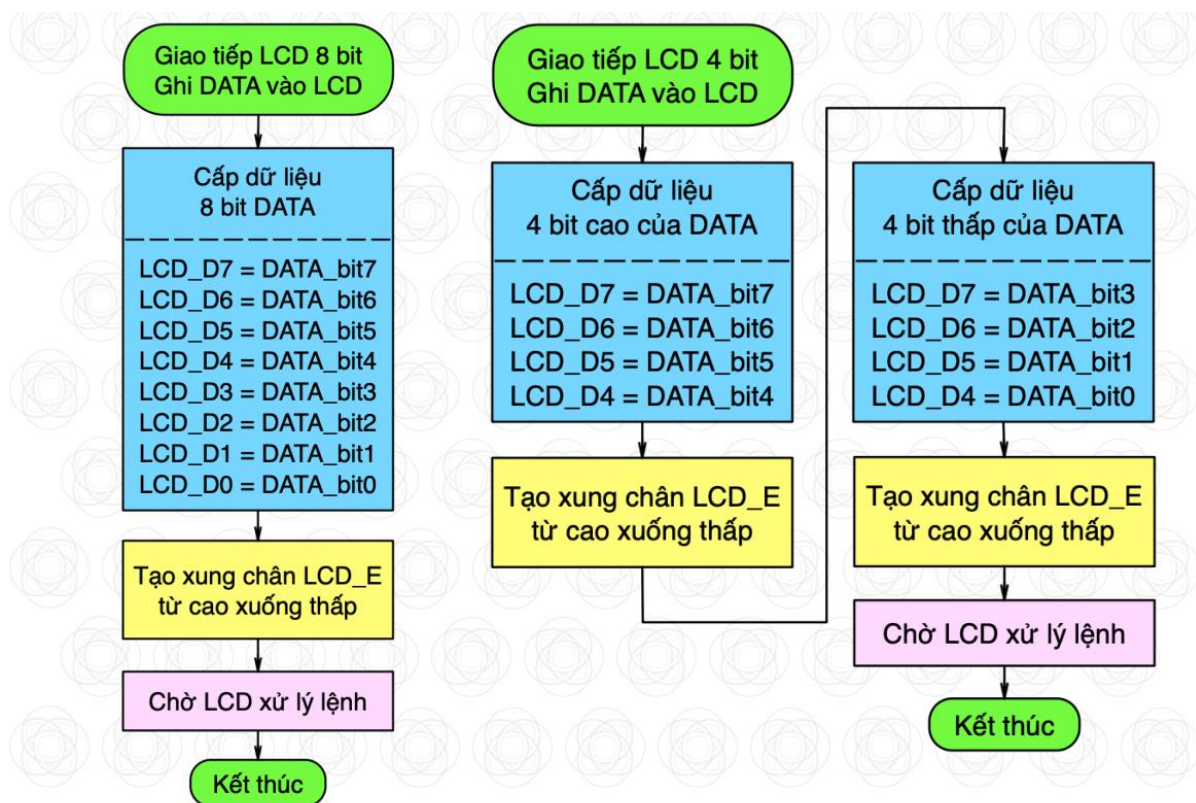


### 3. Nội dung:

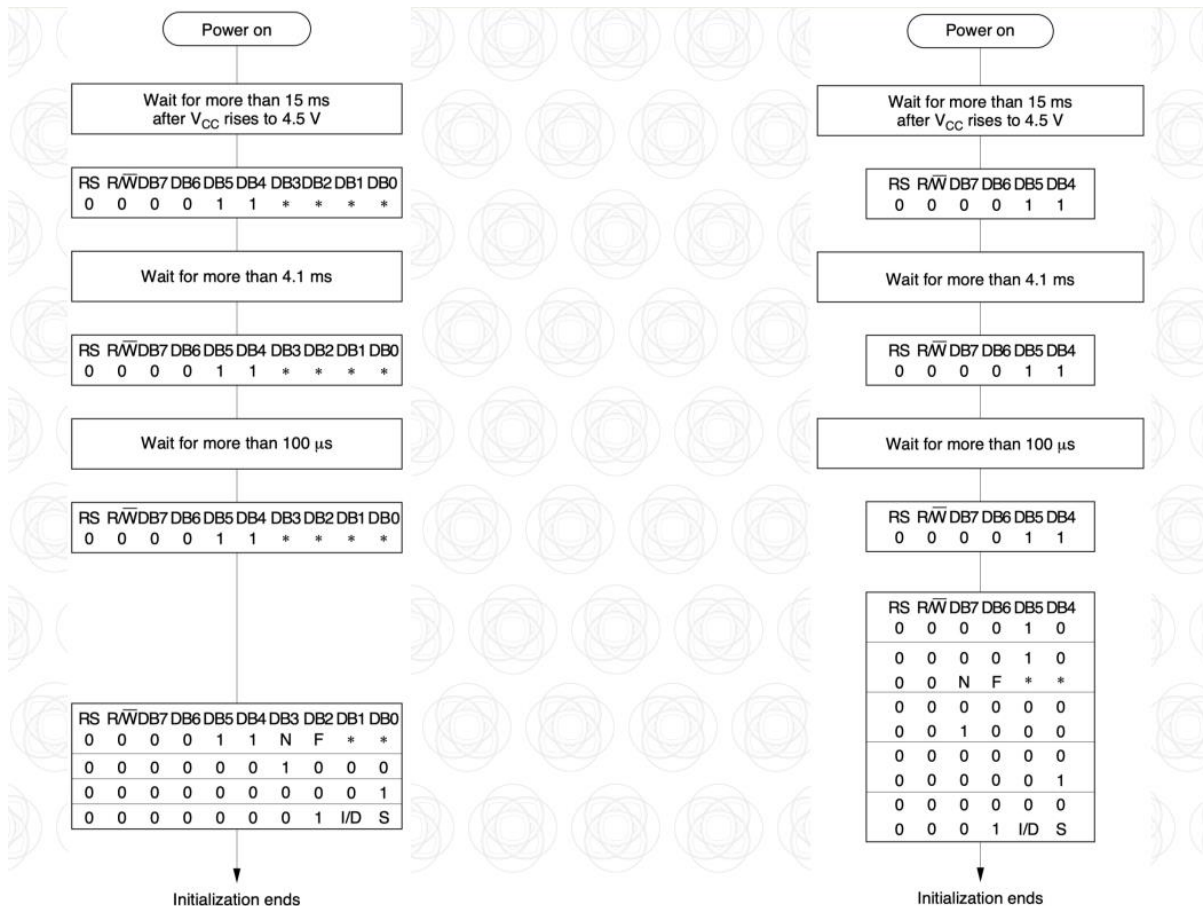
#### 3.1. Phân tích dự án:

- Để hiển thị được trên màn hình LCD ta cần có datasheet của sản phẩm và làm theo hướng dẫn của nhà sản xuất
- Đầu tiên cần viết hàm khởi tạo LCD. Sau đó viết các hàm liên quan để hiển thị được kí tự hay hình đặc biệt lên LCD
- Để xây dựng thư viện ta cần 2 file: 1 file LCD.c để chứa các hàm liên quan đến việc hiển thị trên LCD, 1 file LCD.h chứa tên các hàm (chỉ có tên hàm không có code bên trong hàm). Ngoài ra cần thêm 1 file main.c để test xem thư viện có hoạt động chưa

#### 3.2. Lưu đồ lập trình:



Flowchart để xây dựng hàm LCD\_put() 4bit và 8bit



Trình tự xây dựng hàm LCD\_init() 4bit và 8bit

### 3.3. Mã nguồn chương trình:

- Trong file chi\_LCD.c chứa các hàm
- Lựa chọn giao tiếp LCD 4bit hay 8bit tại đây

```
3  #define LCD_4bit           // Select LCD_8bit or LCD_4bit
```

- Hàm LCD\_put() (4bit)

```
36  #elif defined(LCD_4bit)
37  void LCD_put4bit(uint8_t ch) {
38      HAL_GPIO_WritePin(LCD_RW_GPIO_Port, LCD_RW_Pin, GPIO_PIN_RESET); // write to LCD
39      HAL_GPIO_WritePin(LCD_D7_GPIO_Port, LCD_D7_Pin, ch>>7&1);
40      HAL_GPIO_WritePin(LCD_D6_GPIO_Port, LCD_D6_Pin, ch>>6&1);
41      HAL_GPIO_WritePin(LCD_D5_GPIO_Port, LCD_D5_Pin, ch>>5&1);
42      HAL_GPIO_WritePin(LCD_D4_GPIO_Port, LCD_D4_Pin, ch>>4&1);
43      HAL_GPIO_WritePin(LCD_E_GPIO_Port, LCD_E_Pin, GPIO_PIN_SET); // enable pulse
44      HAL_GPIO_WritePin(LCD_E_GPIO_Port, LCD_E_Pin, GPIO_PIN_RESET);
45  }
46
47  void LCD_put(uint8_t ch) {
48      LCD_put4bit(ch);
49      LCD_put4bit(ch<<4);
50      HAL_Delay(1);
51  }
```

- Hàm LCD\_init() (4bit)

```
52 void LCD_init() {
53     HAL_Delay(15);
54     HAL_GPIO_WritePin(LCD_RS_GPIO_Port, LCD_RS_Pin, GPIO_PIN_RESET); // command
55     LCD_put4bit(0x30);
56     HAL_Delay(5);
57     LCD_put4bit(0x30);
58     HAL_Delay(1);
59     LCD_put4bit(0x30);
60     HAL_Delay(1);
61     LCD_put4bit(0x20);
62     LCD_put(0x28); // 0B00101000 8bit, N=1 (2 lines); F=0 (5x8)
63     LCD_put(0x08);
64     LCD_put(0x01);
65     LCD_put(0x06); // 0B00000110 Increase ID=1, S/C=0 Cursor move
66     // End of LCD initialization procedure
67     LCD_put(0x0D); // 0B00001011 D = 1, C = 0, B = 1 Blink the cursor
68 }
69
```

- Hàm LCD\_put() (8bit)

```
6 void LCD_put(uint8_t ch) {
7     HAL_GPIO_WritePin(LCD_RW_GPIO_Port, LCD_RW_Pin, GPIO_PIN_RESET); // write to LCD
8     HAL_GPIO_WritePin(LCD_D7_GPIO_Port, LCD_D7_Pin, ch>>7&1);
9     HAL_GPIO_WritePin(LCD_D6_GPIO_Port, LCD_D6_Pin, ch>>6&1);
10    HAL_GPIO_WritePin(LCD_D5_GPIO_Port, LCD_D5_Pin, ch>>5&1);
11    HAL_GPIO_WritePin(LCD_D4_GPIO_Port, LCD_D4_Pin, ch>>4&1);
12    HAL_GPIO_WritePin(LCD_D3_GPIO_Port, LCD_D3_Pin, ch>>3&1);
13    HAL_GPIO_WritePin(LCD_D2_GPIO_Port, LCD_D2_Pin, ch>>2&1);
14    HAL_GPIO_WritePin(LCD_D1_GPIO_Port, LCD_D1_Pin, ch>>1&1);
15    HAL_GPIO_WritePin(LCD_D0_GPIO_Port, LCD_D0_Pin, ch&1);
16    HAL_GPIO_WritePin(LCD_E_GPIO_Port, LCD_E_Pin, GPIO_PIN_SET); // enable pulse
17    HAL_GPIO_WritePin(LCD_E_GPIO_Port, LCD_E_Pin, GPIO_PIN_RESET);
18    HAL_Delay(1); // 37 us
19 }
20
```

- Hàm LCD\_init() (8bit)

```
21 void LCD_init() {
22     HAL_Delay(15);
23     HAL_GPIO_WritePin(LCD_RS_GPIO_Port, LCD_RS_Pin, GPIO_PIN_RESET); // command
24     LCD_put(0x30);
25     HAL_Delay(5);
26     LCD_put(0x30);
27     LCD_put(0x30);
28     LCD_put(0x38); // 0B00111100 8bit, N=1 (2 lines); F=0 (5x8)
29     LCD_put(0x08);
30     LCD_put(0x01);
31     LCD_put(0x06); // 0B00000110 Increase ID=1, S/C=0 Cursor move
32     // End of LCD initialization procedure
33     LCD_put(0x0D); // 0B00001011 D = 1, C = 0, B = 1 Blink the cursor
34 }
35
```

- Hàm này dùng để chọn vị trí hiển thị theo tọa độ XY

```
72 void LCD_gotoXY(uint8_t X, uint8_t Y) {
73     HAL_GPIO_WritePin(LCD_RS_GPIO_Port, LCD_RS_Pin, GPIO_PIN_RESET);
74     LCD_put(0x80+0x40*Y + X);
75 }
```

- Hàm dùng để hiển thị kí tự lên LCD

```
77 void LCD_printf(const uint8_t *p) { // *p address
78     HAL_GPIO_WritePin(LCD_RS_GPIO_Port,LCD_RS_Pin,GPIO_PIN_SET); // data
79     uint8_t i = 0;
80     while ( *(p+i) != 0) {
81         LCD_put( *(p+i));
82         i++;
83     }
84 }
85
```

- Hàm này dùng để ghi các kí tự đặc biệt vào ô nhớ còn trống trong thanh ghi

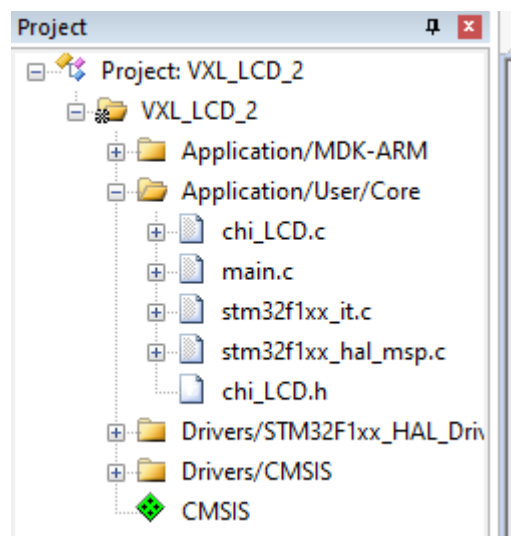
```
86 void LCD_setCGRAM(uint8_t *PatternPointer, uint8_t PatternPosition) {
87     for (uint8_t row = 0; row<8; row++) {
88         HAL_GPIO_WritePin(LCD_RS_GPIO_Port,LCD_RS_Pin,GPIO_PIN_RESET); // command
89         LCD_put(0x40 + (PatternPosition<<3) + row); // address
90         HAL_GPIO_WritePin(LCD_RS_GPIO_Port,LCD_RS_Pin,GPIO_PIN_SET); // data
91         LCD_put(*(PatternPointer + row));
92     }
93 }
```

- Trong file chi\_LCD.h chỉ cần gọi tên các hàm đã xây dựng bên file chi\_LCD.c

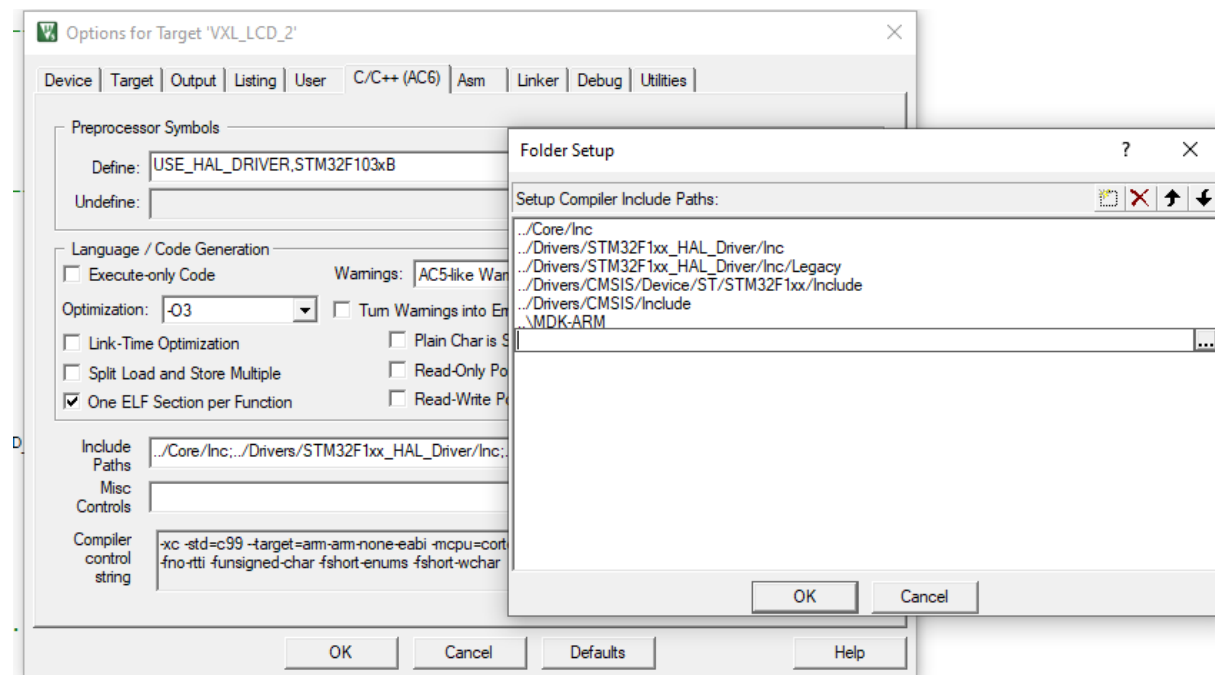
```
1  #include <stdint.h>
2
3  void LCD_put4bit(uint8_t ch);
4  void LCD_put(uint8_t ch);
5  void LCD_init();
6  void LCD_printf(const uint8_t *p);
7  void LCD_gotoXY(uint8_t X, uint8_t Y);
8  void LCD_setCGRAM(uint8_t *PatternPointer, uint8_t PatternPosition);
```

- Với các bước trên ta đã viết xong thư viện chi\_LCD. Bây giờ cần thực hiện 1 số bước để sử dụng được thư viện

- Thêm 2 file đó vào thư mục chứa file main.c



- Thêm đường dẫn tới thư mục chứa file chi\_LCD.h theo hình bên dưới (hiện tại đang nằm trong thư mục MDK-ARM)



- Include thêm thư viện trong code main.c

```
26 #include "chi_LCD.h"
27
```

- Đến đây đã hoàn thành các bước kết nối. Bây giờ chỉ cần gọi hàm muốn sử dụng
- Khai báo bộ mã kí tự đặc biệt (dựa vào datasheet để xây dựng)

```
47 /* USER CODE BEGIN PV */
48 const uint8_t CG1[] = {0,10,31,31,31,14,4,0};
49 const uint8_t CG2[] = {0x0,0xA,0x0,0x0,0x11,0xE,0x0,0x0};
50 const uint8_t CG3[] = {31,31,31,31,31,31,31,31};
51 /* USER CODE END PV */
```

- Gọi hàm và hiển thị

```
106  /* USER CODE BEGIN 2 */
107  LCD_init();
108  LCD_setCGRAM((uint8_t *)CG1,0);
109  LCD_setCGRAM((uint8_t *)CG2,1);
110  LCD_setCGRAM((uint8_t *)CG3,2);
111
112  LCD_gotoXY(0,0);
113  LCD_printf((uint8_t *) "NGUYEN HUU CHI");
114
115  LCD_gotoXY(1,1);
116  HAL_GPIO_WritePin(LCD_RS_GPIO_Port,LCD_RS_Pin,GPIO_PIN_SET);
117  LCD_put(0);
118  LCD_printf((uint8_t *) " 20146479 ");
119  LCD_put(1);
120
```



- Chúng ta cũng có thể sử dụng phương pháp retarget printf để hiển thị lên LCD

```
28  #include <stdio.h>

63  /*struct __FILE
64  {
65      int handle;
66  };*/
67  FILE __stdout;
68  int fputc(int ch, FILE *f)
69  {
70      /* Your implementation of fputc(). */
71      HAL_GPIO_WritePin(LCD_RS_GPIO_Port,LCD_RS_Pin,GPIO_PIN_SET); // command
72      LCD_put(ch);
73      return ch;
74  }
75

109  printf("hello");
```



#### 4. Ghi chú khác (nếu có)