

8.3 Ghép nối với LCD

TS Nguyễn Hồng Quang



Electrical Engineering

1

Giới thiệu LCD

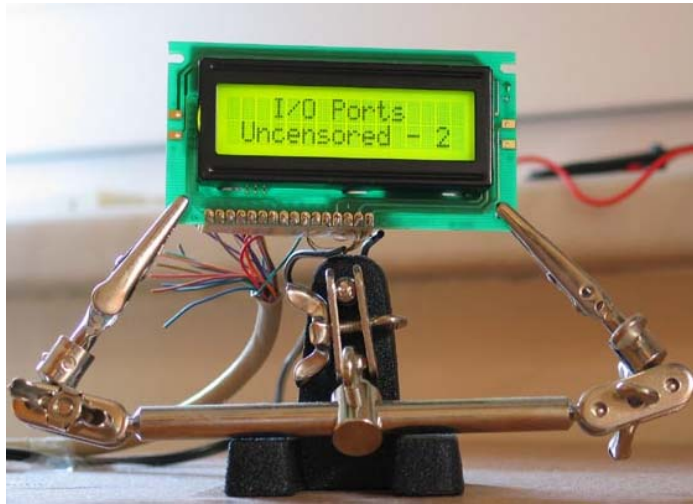
- 8.3.1 Phần cứng
- 8.3.2 Giao tiếp phần mềm
- 8.3.3 Graphic LCD



Electrical Engineering

2

Mạch hiển thị LCD



Electrical Engineering

3

8.3.1 Các loại LCD

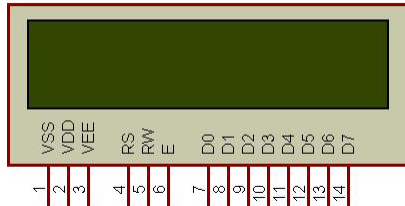
- Màn hình hiển thị tinh thể lỏng
 - Tương thích với mạch HD44780(Hitachi) -KS0066U controller
 - Có thể dễ dàng điều khiển các ký tự trên màn hình sử dụng các từ điều khiển
 - Hiển thị mọi ký tự trong bảng ASCII
 - Có loại 2 x 16 , 4 x 16, 4 x 20
- Loại hiển thị đồ họa (graphic-based)
 - Tương thích với chip Hitachi HD61202, Samsung KS107
 - Kích thước 128x64 hoặc hơn nữa



Electrical Engineering

4

8.3.1 Kết cấu phần cứng



- Kết nối loại 8 bit hoặc 4 bit
- Có đèn backlight hiển thị màu nền
- Lập trình cho LCD là ví dụ về nói chuyện giữa hai VXL với nhau thông qua protocol xác định trước

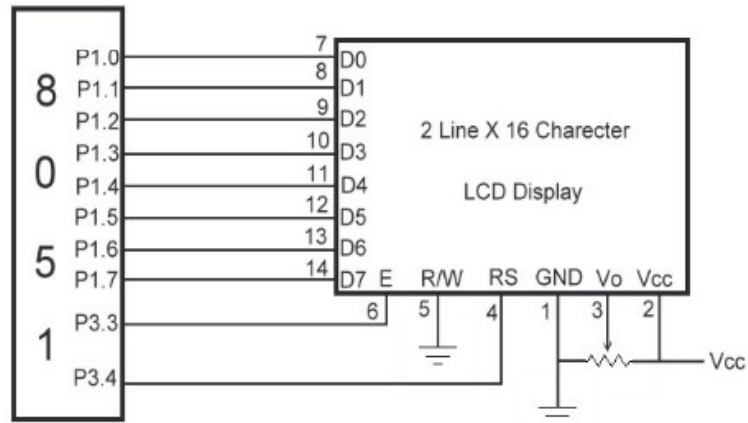


8.3.1 Mô tả chân LCD

Pin No.	Name	Description
Pin no. 1	VSS	Power supply (GND)
Pin no. 2	VCC	Power supply (+5V)
Pin no. 3	VEE	Contrast adjust
Pin no. 4	RS	0 = Instruction input 1 = Data input
Pin no. 5	R/W	0 = Write to LCD module 1 = Read from LCD module
Pin no. 6	EN	Enable signal
Pin no. 7	D0	Data bus line 0 (LSB)
Pin no. 8	D1	Data bus line 1
Pin no. 9	D2	Data bus line 2
Pin no. 10	D3	Data bus line 3
Pin no. 11	D4	Data bus line 4
Pin no. 12	D5	Data bus line 5
Pin no. 13	D6	Data bus line 6
Pin no. 14	D7	Data bus line 7 (MSB)



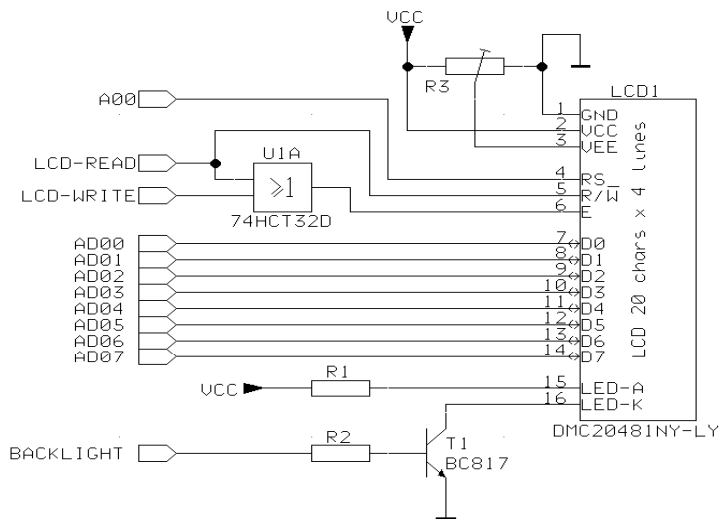
8.3.1 Ví dụ về ghép nối 8bit



Electrical Engineering

7

8.3.1 Ví dụ ghép nối LCD theo địa chỉ bộ nhớ



Electrical Engineering

8

8.3.1 DDRAM (display data RAM)

00	01	02	03	04	05	06	07	32	33	34	35	36	37	38	39	← Character position (dec.)
00	01	02	03	04	05	06	07	20	21	22	23	24	25	26	27	← Row0 DDRAM address (hex)

DDRAM Address for 1 Line LCD

00	01	02	03	04	05	06	07	32	33	34	35	36	37	38	39	← Character position (dec.)
00	01	02	03	04	05	06	07	20	21	22	23	24	25	26	27	← Row0 DDRAM address (hex)
40	41	42	43	44	45	46	47	60	61	62	63	64	65	66	67	← Row1 DDRAM address (hex)

DDRAM Address for 2 Line LCD

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	← Character position (dec.)
00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	← Row0 DDRAM address (hex)
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	← Row1 DDRAM address (hex)
14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27	← Row2 DDRAM address (hex)
54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	66	67	← Row3 DDRAM address (hex)

DDRAM Address for 4 Line LCD



Electrical Engineering

9

8.3.1 CGROM - Character Generator ROM

Address	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000			0	1	A	P	\	P			-	9	3	α	p	
xxxx0001 (2)			!	1	A	Q	a	a			7	7	4	ä	q	
xxxx0010 (3)			"	2	B	R	b	r			"	イ	ツ	β	θ	
xxxx0011 (4)			#	3	C	S	c	s			」	ウ	テ	ε	ω	
xxxx0100 (5)			\$	4	D	T	d	t			、	エ	ト	μ	ρ	
xxxx0101 (6)			%	5	E	U	e	u			・	オ	ナ	1	σ	Ü
xxxx0110 (7)			&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111 (8)			'	7	G	W	g	w			ヲ	キ	ヲ	ラ	g	π
xxxx1000 (1)			<	8	H	X	h	x			イ	ク	ネ	リ	J	Σ
xxxx1001 (2)			>	9	I	Y	i	y			ウ	ケ	ル	リ	Y	
xxxx1010 (3)			*	:	J	Z	j	z			エ	コ	ハ	レ	j	7
xxxx1011 (4)			+	;	K	[k	[オ	サ	ヒ	ロ	*	π
xxxx1100 (5)			,	<	L	¥	l	¥			ハ	シ	フ	ワ	φ	π
xxxx1101 (6)			-	=	M]m]m	>			ユ	ズ	ハ	ン	も	÷
xxxx1110 (7)			.	>	N	^	n	^			ヨ	セ	ホ	°	π	
xxxx1111 (8)			/	?	O	_	o	_			ッ	ソ	マ	°	ö	■

Bảng ký tự mã hóa dạng 5x8 bit



Electrical Engineering

10

8.3.2 Nguyên tắc điều khiển LCD

- Đường EN
 - Khởi động thì đặt mức 1
 - Sau khi kết thúc việc gửi dữ liệu thì trở về mức 0
- Đường RS
 - Khi RS mức 0, gửi lệnh đặc biệt, ví dụ như xóa màn hình, trả con trỏ về đầu dòng
 - Khi RS mức 1, gửi dữ liệu cần hiển thị sẽ gửi vào bus dữ liệu
- Đường R/W
 - Lệnh điều khiển đọc ghi
 - Khi R mức cao thì đọc từ LCD
 - Khi R mức thấp thì ghi vào LCD



8.3.2 Nguyên tắc xử lý cờ bận (busy flag)

- $BF = 1$ khi LCD đang xử lý dữ liệu và $BF = 0$ khi kết thúc quá trình đó
- Để đọc cờ Busy Flag, đặt $RS = 0$ and $R/W = 1$ và bit cao nhất D7 của LCD nhận trạng thái của cờ.
- $BF = 0$ có nghĩa là LCD đã sẵn sàng cho nhận lệnh tiếp theo



8.3.2 Các lệnh cơ bản đối với LCD

- Thực hiện hàm cơ bản như định dạng dữ liệu, loại dữ liệu ...
- Đặt địa chỉ RAM trong
- Truyền dữ liệu với RAM trong
- Các hàm khác



8.3.2 Bảng lệnh

Command	Code										Description	Execution Time	
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0			
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears the display and returns the cursor to the home position (address 0).	82µs-1.64ms	
Return Home	0	0	0	0	0	0	0	0	0	1 *	Returns the cursor to the home position (address 0). Also returns a shifted display to the home position. DD RAM contents remain unchanged.	40µs-1.64ms	
Entry Mode Set	0	0	0	0	0	0	0	0	1	I/D	S	Sets the cursor move direction and enables/disables the display.	40µs
Display ON/OFF Control	0	0	0	0	0	0	0	1	D	C	B	Turns the display ON/OFF (D), or the cursor ON/OFF (C), and blink of the character at the cursor position (B).	40µs
Cursor & Display Shift	0	0	0	0	0	0	1	S/C	R/L	*	*	Moves the cursor and shifts the display without changing the DD RAM contents.	40µs
Function Set	0	0	0	0	0	1	DL	N\$	F	*	#	Sets the data width (DL), the number of lines in the display (L), and the character font (F).	40µs
Set CG RAM Address	0	0	0	0	1	A _{CG}						Sets the CG RAM address. CG RAM data can be read or altered after making this setting.	40µs
Set DD RAM Address	0	0	0	1	A _{DD}						Sets the DD RAM address. Data may be written or read after making this setting.	40µs	
Read Busy Flag & Address	0	1	BF	AC						Reads the BUSY flag (BF) indicating that an internal operation is being performed and reads the address counter contents.		1µs	
Write Data to CG or DD RAM	1	0	Write Data						Writes data into DD RAM or CG RAM.		46µs		
Read Data from CG or DD RAM	1	1	Read Data						Reads data from DD RAM or CG RAM.		46µs		
<div>I/D = 1: Increment I/D = 0: Decrement S = 1: Accompanies display shift. S/C = 1: Display shift S/C = 0: cursor move R/L = 1: Shift to the right. R/L = 0: Shift to the left. DL = 1: 8 bits DL = 0: 4 bits N = 1: 2 lines N = 0: 1 line F = 1: 5x10 dots F = 0: 5 x 7 dots BF = 1: Busy BF = 0: Can accept data # Set to 1 on 24x4 modules \$ With KS0072 is Address Mode.</div>												DD RAM: Display data RAM CG RAM: Character generator RAM A _{CG} : CG RAM Address A _{DD} : DD RAM Address AC: Address counter Used for both DD and CG RAM address.	Execution times are typical. If transfers are timed by software and the busy flag is not used, add 10% to the above times.



8.3.2 Các lệnh thông dụng

No.	Instruction	Hex	Decimal
1	Function Set: 8-bit, 1 Line, 5x7 Dots	0x30	48
2	Function Set: 8-bit, 2 Line, 5x7 Dots	0x38	56
3	Function Set: 4-bit, 1 Line, 5x7 Dots	0x20	32
4	Function Set: 4-bit, 2 Line, 5x7 Dots	0x28	40
5	Entry Mode	0x06	6
6	Display off Cursor off (clearing display without clearing DDRAM content)	0x08	8
7	Display on Cursor on	0x0E	14
8	Display on Cursor off	0x0C	12
9	Display on Cursor blinking	0x0F	15
10	Shift entire display left	0x18	24
12	Shift entire display right	0x1C	30
13	Move cursor left by one character	0x10	16
14	Move cursor right by one character	0x14	20
15	Clear Display (also clear DDRAM content)	0x01	1
16	Set DDRAM address or cursor position on display	0x80+add*	128+add*
17	Set CGRAM address or set pointer to CGRAM location	0x40+add**	64+add**



Electrical Engineering

15

8.3.2 Lệnh reset LCD

- Xóa màn hình
- Hàm
 - DL = 1; 8-bit interface data
 - N = 0; 1-line display
 - F = 0; 5 x 8 dot character font
- Display on/off control:
 - D = 0; Display off
 - C = 0; Cursor off
 - B = 0; Blinking off
- Phụ thuộc vào nguồn với yêu cầu Ton, Toff



Electrical Engineering

16

8.3.2 Khởi tạo LCD

- 1) Gửi lệnh 0x30 - Using 8-bit interface
- 2) Delay 20ms
- 3) Gửi lệnh 0x30 - 8-bit interface
- 4) Delay 20ms
- 5) Gửi lệnh 0x30 - 8-bit interface
- 6) Delay 20ms
- 7) Gửi hàm điều khiển
- 8) Gửi lệnh xóa màn hình
- 9) Đặt entry mode



8.3.2 Mode nhập liệu (entry mode)

- a) I/D - Increment/Decrement bit
- b) S - Display shift.
- Tổ hợp 2 bit này tạo ra trạng thái lệnh 0x04, 0x05, 0x06, 0x07
- Bình thường, mode 0x06 được dùng có nghĩa là No shift và auto increment.



8.3.2 Ví dụ sử dụng 8051

```

LCD_data equ P2      ;LCD Data port
LCD_D7   equ P2.7    ;LCD D7/Busy Flag
LCD_rs   equ P1.0    ;LCD Register Select
LCD_rw   equ P1.1    ;LCD Read/Write
LCD_en   equ P1.2    ;LCD Enable

LCD_init:
    mov     LCD_data,#38H ;Function set: 2 Line, 8-bit, 5x7 dots
    clr     LCD_rs        ;Selected command register
    clr     LCD_rw        ;We are writing in instruction register
    setb    LCD_en        ;Enable H->L
    clr     LCD_en
    acall   LCD_busy      ;Wait for LCD to process the command
    mov     LCD_data,#0FH ;Display on, Cursor blinking command
    clr     LCD_rs        ;Selected instruction register
    clr     LCD_rw        ;We are writing in instruction register
    setb    LCD_en        ;Enable H->L
    clr     LCD_en
    acall   LCD_busy      ;Wait for LCD to process the command
    mov     LCD_data,#01H ;Clear LCD
    clr     LCD_rs        ;Selected command register
    clr     LCD_rw        ;We are writing in instruction register
    setb    LCD_en        ;Enable H->L
    clr     LCD_en
    acall   LCD_busy      ;Wait for LCD to process the command
    mov     LCD_data,#06H ;Entry mode, auto increment with no shift
    clr     LCD_rs        ;Selected command register
    clr     LCD_rw        ;We are writing in instruction register
    setb    LCD_en        ;Enable H->L
    clr     LCD_en
    acall   LCD_busy      ;Wait for LCD to process the command
    ret           ;Return from routine

```



Electrical Engineering

19

8.3.2 Đọc cờ chờ

```

LCD_busy:
    setb    LCD_D7        ;Make D7th bit of LCD data port as i/p
    setb    LCD_en        ;Make port pin as o/p
    clr     LCD_rs        ;Select command register
    setb    LCD_rw        ;we are reading

check:
    clr     LCD_en        ;Enable H->L
    setb    LCD_en
    jb      LCD_D7,check  ;read busy flag again and again till it
becomes 0
    ret           ;Return from busy routine

```

```

LCD_busy:
    mov     r7,#50H
back:
    mov     r6,#FFH
    djnz    r6,$
    djnz    r7,back
    ret

```



Electrical Engineering

20

8.3.2 Gửi lệnh tới LCD

```
LCD_command:
    mov    LCD_data,A      ;Move the command to LCD port
    clr    LCD_rs          ;Selected command register
    clr    LCD_rw          ;We are writing in instruction register
    setb   LCD_en          ;Enable H->L
    clr    LCD_en
    acall  LCD_busy        ;Wait for LCD to process the command
    ret                    ;Return from busy routine

mov    a,#01H            ;01H is command for clearing LCD
acall  LCD_command        ;Send the command
```



8.3.2 Đặt vị trí con trỏ

Bit7	6	5	4	3	2	1	0
1	AD6	AD5	AD4	AD3	AD2	AD1	AD0

- DDRAM bắt đầu từ địa chỉ 0x80.
- Ví dụ cho 2 x 16 LCD. Địa chỉ là 0x80 - 0x8F và 0xC0 to 0xCF là nhìn thấy
- Tập hợp 80 ký tự đều có thể hiển thị thông qua lệnh cuộn LCD

```
;We are placing the cursor on the 4th position
;so the DDRAM address will be 0x03
;and the command will be 0x80+0x03 = 0x83
mov    a,#83H            ;load the command
acall  LCD_command        ;send command to LCD
```



8.3.2 Gửi dữ liệu ra LCD

```
LCD_senddata:
    mov     LCD_data,A      ;Move the command to LCD port
    setb    LCD_rs          ;Selected data register
    clr     LCD_rw          ;We are writing
    setb    LCD_en          ;Enable H->L
    clr     LCD_en
    acall   LCD_busy        ;Wait for LCD to process the data
    ret                          ;Return from busy routine

mov     a,#'A'              ;Ascii value of 'A' will be loaded in
acall   LCD_senddata        ;Send data
```




8.3.2 Ví dụ về gửi xâu dữ liệu

```
LCD_sendstring:
    clr     a                ;clear Accumulator for any previous
data      movc    a,@a+dptr    ;load the first character in
accumulator
    jz      exit            ;go to exit if zero
    acall   lcd_senddata     ;send first char
    inc     dptr            ;increment data pointer
    sjmp    LCD_sendstring   ;jump back to send the next character
exit:
    ret                      ;End of routine

;
; mov     dptr,#my_string ;
; acall   LCD_sendstring   ;Send string
;
; DB      "LCD Tutorial", 00H
; 00H indicate that string is finished.
```



8.3.2 Tạo ra ký tự riêng

Custom Pattern	Decimal	Hex
	Row 1: 4 Row 2: 14 Row 3: 14 Row 4: 14 Row 5: 31 Row 6: 0 Row 7: 4	0x04 0x0E 0x0E 0x0E 0x1F 0x00 0x04
Bit: 4 3 2 1 0 - Hex Row1: 0 0 1 0 0 - 0x04 Row2: 0 1 1 1 0 - 0x0E Row3: 0 1 1 1 0 - 0x0E Row4: 0 1 1 1 0 - 0x0E Row5: 1 1 1 1 1 - 0x1F Row6: 0 0 0 0 0 - 0x00 Row7: 0 0 1 0 0 - 0x04 Row8: 0 0 0 0 0 - 0x00		

- Sử dụng CGRAM để lưu trữ ký tự riêng bắt đầu từ địa chỉ 0x40
- Việc tạo ký tự hình thành bởi tập ma trận điểm 5x8
- Hàng cuối cùng dành cho hiển thị con trỏ



8.3.2 Vùng địa chỉ CGRAM

Memory Map	
Pattern No.	CGRAM Address (Acg)
1	0x00 - 0x07
2	0x08 - 0x0F
3	0x10 - 0x17
4	0x18 - 0x1F
5	0x20 - 0x27
6	0x28 - 0x2F
7	0x30 - 0x37
8	0x38 - 0x3F

Command	Code								Description	Execution Time
	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2		
Set CG RAM Address	0	0	0	1	A _{CG}				Sets the CG RAM address. CG RAM data can be read or altered after making this setting.	40μs

- Lệnh điều khiển được gửi tới địa chỉ tuyệt đối là địa chỉ tương đối + 0x40



8.3.2 Nguyên tắc tạo dữ liệu riêng

```
//Usage:
//    pattern[8]={0x04,0x0E,0x0E,0x0E,0x1F,0x00,0x04,0x00};
//    LCD_build(1,pattern);
//
//LCD Ports are same as discussed in previous sections

void LCD_build(unsigned char location, unsigned char *ptr) {
    unsigned char i;
    if(location<8) {
        LCD_command(0x40+(location*8));
        for(i=0;i<8;i++)
            LCD_senddata(ptr[i]);
    }
}
```

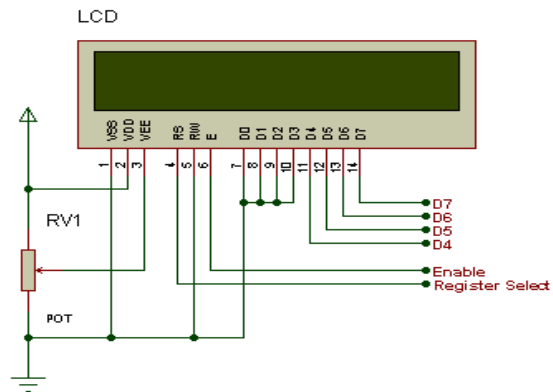


8.3.2 Ví dụ phần mềm assembler

```
LCD_build:
    mov     A,#48H           ;Load the location where we want to store
    acall  LCD_command       ;Send the command
    mov     A,#04H           ;Load row 1 data
    acall  LCD_senddata      ;Send the data
    mov     A,#0EH           ;Load row 2 data
    acall  LCD_senddata      ;Send the data
    mov     A,#0EH           ;Load row 3 data
    acall  LCD_senddata      ;Send the data
    mov     A,#0EH           ;Load row 4 data
    acall  LCD_senddata      ;Send the data
    mov     A,#1FH           ;Load row 5 data
    acall  LCD_senddata      ;Send the data
    mov     A,#00H           ;Load row 6 data
    acall  LCD_senddata      ;Send the data
    mov     A,#04H           ;Load row 7 data
    acall  LCD_senddata      ;Send the data
    mov     A,#00H           ;Load row 8 data
    acall  LCD_senddata      ;Send the data
    ret                     ;Return from routine
```



8.3.2 LCD trong chế độ 4 bit



- Lý do chọn 4 bit để tiết kiệm số chân VXL



8.3.2 Phương pháp khởi tạo chế độ 4 bit

- Trễ 20mS
- Gửi lệnh (0x30)
- Trễ 10mS
- Gửi lệnh (0x30)
- Trễ 1mS
- Gửi lệnh (0x30)
- Trễ 1mS
- Chọn mode (0x30 - for 8-bit and 0x20 for 4-bit)
- Trễ 1mS



8.3.2 Cờ busy

- Trong chế độ 4bit người ta không đọc cờ chờ vì phải đọc 2 lần 4 bit (4bit cao và 4 bit thấp).
- Thay vì đó, ta có thể trễ thời gian chờ từ 300 to 600uS.
- Tùy thuộc vào loại LCD, thời gian chờ có thể khác nhau, tuy vậy 400uS là hợp lý



8.3.2 Ví dụ về khởi tạo LCD 4bit

```
!D4 - P3.0
;D5 - P3.1
;D6 - P3.2
;D7 - P3.3
;EN - P3.7
;RS - P3.5

lcd_port equ P3          ;LCD connected to Port3
en equ P3.7              ;Enable connected to P3.7
rs equ P3.5              ;Register select to P3.5

lcd_reset:               ;LCD reset sequence
    mov lcd_port, #0FFH
    mov delay, #20        ;20ms delay
    acall delays
    mov lcd_port, #83H    ;Data = 30H, EN = 1, First Init
    mov lcd_port, #03H    ;Data = 30H, EN = 0
    mov delay, #15        ;Delay 15ms
    acall delays
    mov lcd_port, #83H    ;Second Init, Data = 30H, EN = 1
    mov lcd_port, #03H    ;Data = 30H, EN = 0
    mov delay, #5         ;Delay 5ms
    acall delays
    mov lcd_port, #83H    ;Third Init
    mov lcd_port, #03H    ;Delay 5ms
    mov delay, #5
    acall delays
    mov lcd_port, #82H    ;Select Data width (20H for 4bit)
    mov lcd_port, #02H    ;Data = 20H, EN = 0
    mov delay, #5         ;Delay 5ms
    acall delays
    ret
```



8.3.2 Tiếp

```
lcd_init:
    acall lcd_reset      ;Call LCD Reset sequence
    mov a,#28H           ;4-bit, 2 line, 5x7 dots
    acall lcd_cmd        ;Call LCD command
    mov a,#0CH           ;Display ON cursor OFF
    acall lcd_cmd        ;Call LCD command
    mov a,#06H           ;Set entry mode (Auto increment)
    acall lcd_cmd        ;Call LCD command
    mov a,#80H           ;Bring cursor to line 1
    acall lcd_cmd        ;Call LCD command
    ret
```



8.3.2 Tương tự về gửi lệnh

```
lcd_cmd:                ;LCD command Routine
    mov temp,a           ;Save a copy of command to temp
    swap a               ;Swap to use higher nibble
    anl a,#0FH           ;Mask the first four bits
    add a,#80H           ;Enable = 1, RS = 0
    mov lcd_port,a       ;Move it to lcd port
    anl a,#0FH           ;Enable = 0, RS = 0
    mov lcd_port,a       ;Move to lcd port

    mov a,temp           ;Reload the command from temp
    anl a,#0FH           ;Mask first four bits
    add a,#80H           ;Enable = 1
    mov lcd_port,a       ;Move to port
    anl a,#0FH           ;Enable = 0
    mov lcd_port,a       ;Move to lcd port

    mov delay,#1         ;delay 1 ms
    acall delayms
    ret
```



8.3.2 Gửi dữ liệu

```
lcd_dat:                                ;LCD data Routine
    mov temp,a                          ;Keep copy of data in temp
    swap a                              ;We need higher nibble
    anl a,#0FH                          ;Mask first four bits
    add a,#0A0H                         ;Enable = 1, RS = 1
    mov lcd_port,a                      ;Move to lcd port
    nop
    clr en                              ;Enable = 0

    mov a,temp                          ;Reload the data from temp
    anl a,#0FH                          ;we need lower nibble now
    add a,#0A0H                         ;Enable = 1, RS = 1
    mov lcd_port,a                      ;Move to lcd port
    nop
    clr en                              ;Enable = 0

    mov delay,#1                        ;Delay 1mS
    acall delayms
    ret
```



8.3.2 Ví dụ sử dụng C

```
void lcd_cmd (char cmd)
{
    lcd_port = ((cmd >> 4) & 0x0F) | LCD_EN;
    lcd_port = ((cmd >> 4) & 0x0F);

    lcd_port = (cmd & 0x0F) | LCD_EN;
    lcd_port = (cmd & 0x0F);

    delayus(200);
    delayus(200);
}

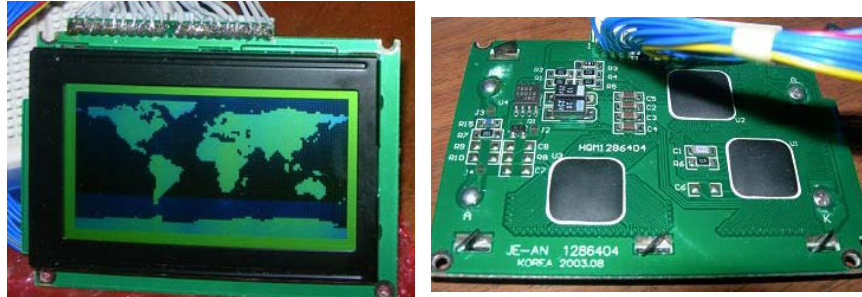
void lcd_data (unsigned char dat)
{
    lcd_port = (((dat >> 4) & 0x0F) | LCD_EN | LCD_RS);
    lcd_port = (((dat >> 4) & 0x0F) | LCD_RS);

    lcd_port = ((dat & 0x0F) | LCD_EN | LCD_RS);
    lcd_port = ((dat & 0x0F) | LCD_RS);

    delayus(200);
    delayus(200);
}
```



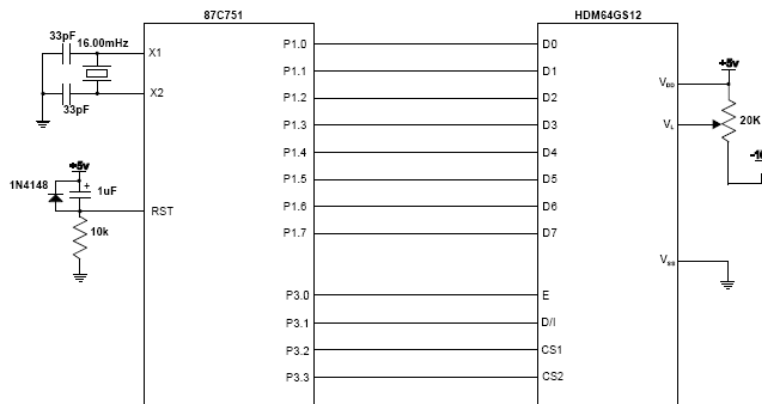
8.3.3 Graphical LCD – 128x64



Electrical Engineering

37


8.3.2 Ghép nối với 8051



Electrical Engineering

38

8.3.3 Chân tín hiệu cơ bản

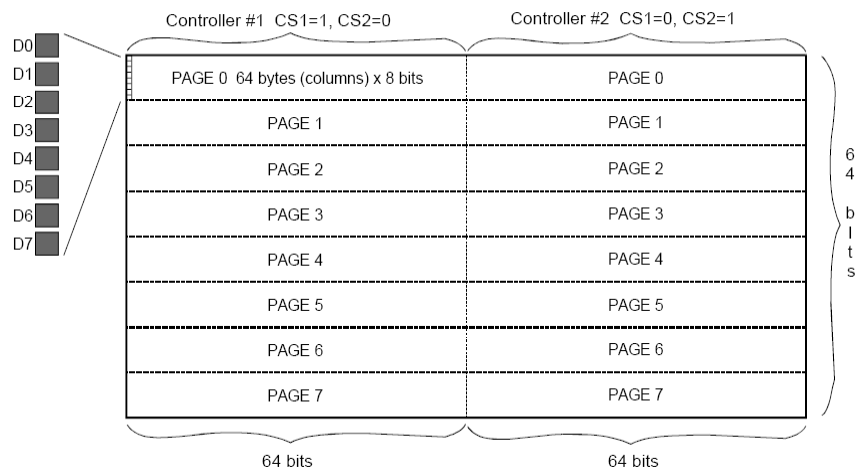
NO.	SYMBOL	LEVEL	FUNCTION
1	V _{ss}		GROUND
2	V _{cc}		POWER SUPPLY FOR LOGIC CIRCUIT
3	V _o		BRIGHTNESS CONTROL
4	RS	H / L 	H : DATA INPUT L : INSTRUCTION CODE INPUT
5	R / W	H / L	H : DATA READ (LCD MODULE → MPU) L : DATA WRITE (LCD MODULE ← MPU)
6	E		ENABLE (SEE CH.5 - TIMING CHARACTERISTICS)
7-14	DB0-DB7	H / L	DATA BUS LINE
15	CS1	H	CHIP SELECT FOR IC1
16	CS2	H	CHIP SELECT FOR IC2
17	RST	L	RESET
18	V _{out}		NEGATIVE VOLTAGE OUTPUT (-5V)
19	+V _{led}		BACKLIGHT
20	-V _{led}		BACKLIGHT



Electrical Engineering

39

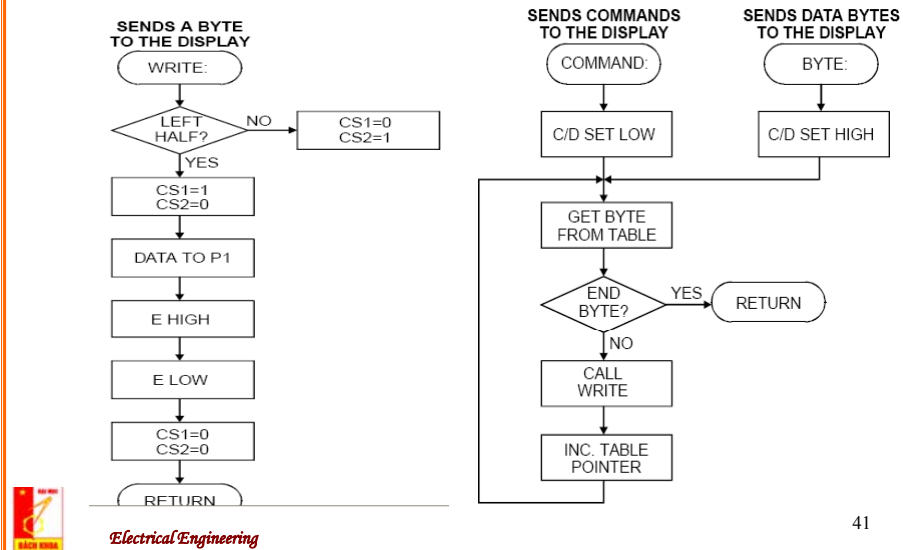
8.3.3 Kết cấu trang màn hình



Electrical Engineering

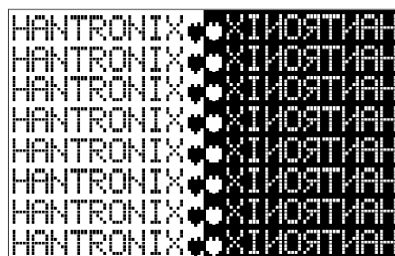
40

8.3.3 Cấu trúc chương trình



41

8.3.3 Ví dụ hiển thị



```

Msg11:
db  0,0feh,10h,10h,10h,0feh,0      ;H
db  0fch,12h,12h,12h,0fch,0        ;A
db  0feh,08h,10h,20h,0feh,0        ;N
db  02h,02h,0feh,02h,02h,0        ;T
db  0feh,12h,32h,52h,8ch,0         ;R
db  7ch,82h,82h,82h,7ch,0          ;O
db  0feh,08h,10h,20h,0feh,0        ;N
db  0,0,82h,0feh,82h,0             ;I
db  0,0c6h,28h,10h,28h,0c6h,0      ;X
db  0,38h,7ch,0f8h,7ch,38h,0       ;heart
db  0,99h

; "Hantronix", right half (reverse video)

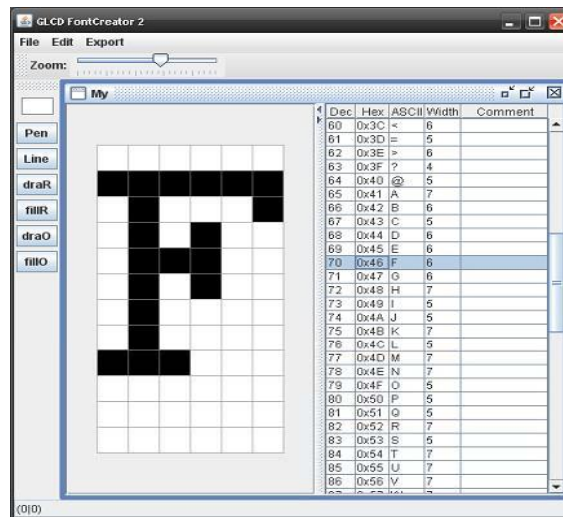
Msg1r:
db  0ffh,0c7h,83h,07h,83h,0c7h,0ffh ;heart
db  0ffh,39h,0d7h,0efh,0d7h,39h,0ffh ;X
db  0ffh,0ffh,7dh,01h,7dh,0ffh        ;I
db  01h,0dfh,0efh,0f7h,01h,0ffh      ;N
db  83h,7dh,7dh,7dh,83h,0ffh         ;O
db  073h,0adh,0cdh,0edh,01h,0ffh     ;R
db  0fdh,0fdh,01h,0fdh,0fdh,0ffh     ;T
db  01h,0dfh,0efh,0f7h,01h,0ffh      ;N
db  03h,0edh,0edh,0edh,03h,0ffh      ;A
db  0ffh,01h,0efh,0efh,0efh,01h,0ffh ;H
db  0ffh,99h
  
```



Electrical Engineering

42

8.3.3 Tạo font sử dụng GLCDFontCreator2



Electrical Engineering

43