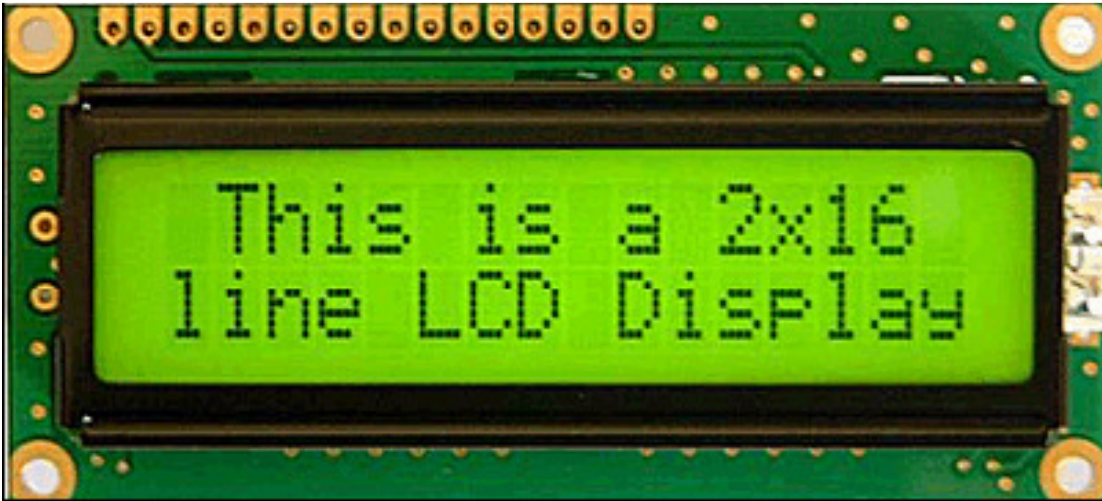
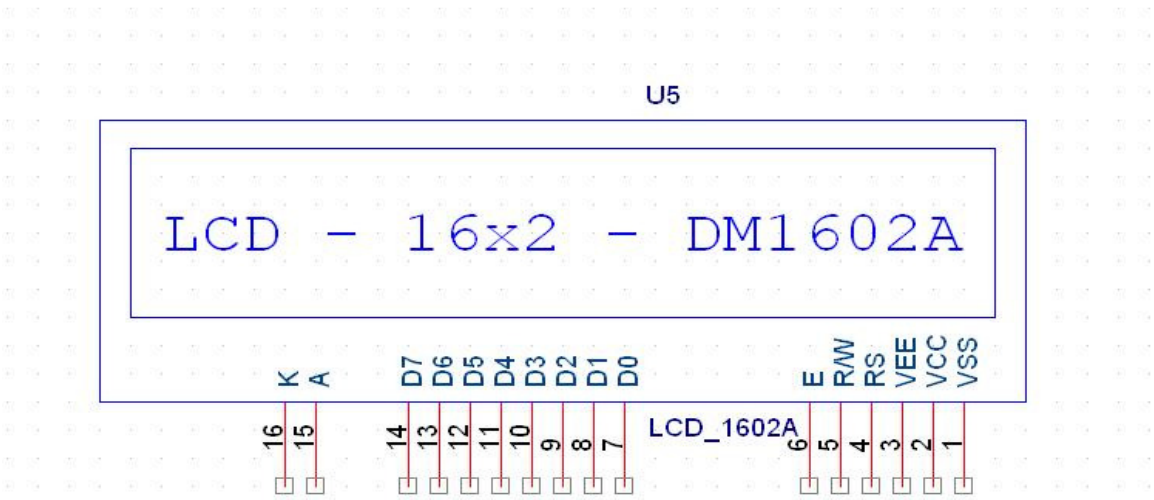


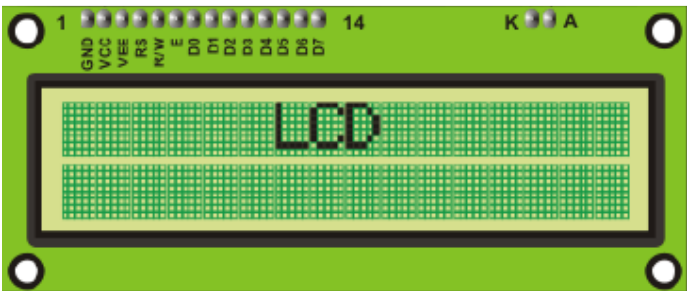
LCD 1602



Hình dáng của LCD 1602



Sơ đồ chân của LCD 1602



Chân	Kí hiệu	Mức Logic	I/O	Chức năng
1	Vss	-	-	Nguồn (GND)
2	Vcc	-	-	Nguồn (+5V)
3	Vee	-	-	Chỉnh độ tương phản
4	RS	0/1	I	0 = Nhập lệnh 1 = Nhập dữ liệu
5	R/W	0/1	I	0 = Ghi dữ liệu 1 = Đọc dữ liệu
6	E	1, 1->0	I	Tín hiệu cho phép
7	DB0	0/1	I/O	Bus dữ liệu 0
8	DB1	0/1	I/O	Bus dữ liệu 1
9	DB2	0/1	I/O	Bus dữ liệu 2
10	DB3	0/1	I/O	Bus dữ liệu 3
11	DB4	0/1	I/O	Bus dữ liệu 4
12	DB5	0/1	I/O	Bus dữ liệu 5
13	DB6	0/1	I/O	Bus dữ liệu 6
14	DB7	0/1	I/O	Bus dữ liệu 7
15	Lamp-	-	-	Đèn LCD
16	Lamp+	-	-	Đèn LCD

Các thanh ghi

- Thanh ghi IR: Mỗi lệnh được nhà sản xuất LCD đánh địa chỉ rõ ràng. Người dùng chỉ việc cung cấp địa chỉ lệnh bằng cách nạp vào thanh ghi IR.

Ví dụ:

Lệnh “hiển thị màn hình và con trỏ” có mã lệnh là 00001110

- Thanh ghi DR : Thanh ghi DR dùng để chứa dữ liệu 8 bit để ghi vào vùng RAM DDRAM hoặc CGRAM (ở chế độ ghi) hoặc dùng để chứa dữ liệu từ 2 vùng RAM này gửi ra cho MPU (ở chế độ đọc).

- **Cờ báo bận BF: (Busy Flag)**

Khi đang thực thi các hoạt động bên trong, LCD bỏ qua mọi giao tiếp với bên ngoài và bật cờ BF(thông qua chân DB7 khi có thiết lập RS=0, R/W=1) lên để cho biết nó đang “bận”.

- **Bộ đếm địa chỉ AC : (Address Counter)**

Khi một địa chỉ lệnh được nạp vào thanh ghi IR, thông tin được nối trực tiếp cho 2 vùng RAM (việc chọn lựa vùng RAM tương tác đã được bao hàm trong mã lệnh).

Sau khi ghi vào (đọc từ) RAM, bộ đếm AC tự động tăng lên (giảm đi) 1 đơn vị.

Bộ nhớ LCD

Vùng RAM hiển thị DDRAM : (Display Data RAM)

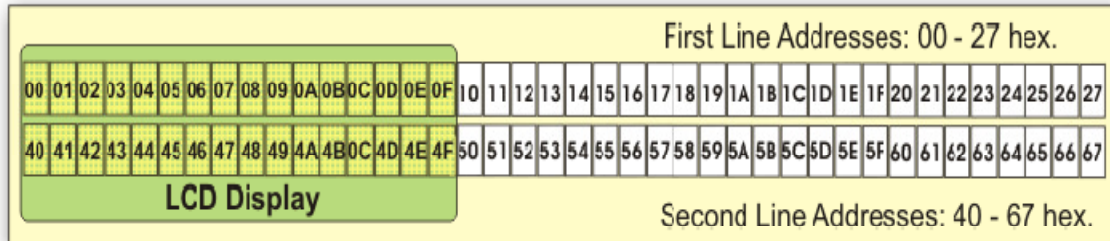
Vùng CGROM: Character Generator ROM

Vùng CGRAM : (Character Generator RAM)

DDRAM

Đây là vùng RAM dùng để hiển thị, nghĩa là ứng với một địa chỉ của RAM là một ô kí tự trên màn hình.

DDRAM Memory



CGROM

Chứa các mẫu kí tự loại 5x7 hoặc 5x10 điểm ảnh/kí tự, và định địa chỉ bằng 8 bit.

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

A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0											LSB 04 03 02 01 00				
											1	0	0	0	0
											1	0	0	0	0
											1	0	1	1	0
											1	1	0	0	1
											1	0	0	0	1
											1	0	0	0	1
											1	1	1	1	0
0	1	1	0	0	0	1	0	0	1	1	0	0	0	0	0

Mẫu kí tự đồ họa riêng.

Table 5 Relationship between CGRAM Addresses, Character Codes (DDRAM) and Character Patterns (CGRAM Data)

For 5 × 8 dot character patterns

Character Codes (DDRAM data)								CGRAM Address						Character Patterns (CGRAM data)												
7	6	5	4	3	2	1	0	5	4	3	2	1	0	7	6	5	4	3	2	1	0					
High				Low				High		Low				High		Low										
0 0 0 0 * 0 0 0								0 0 0				0 0 0				↑ ↓		1 1 1 1 0						} Character pattern (1)		
												0 0 1						1 0 0 0 1								
												0 1 0						1 0 0 0 1								
												0 1 1						1 1 1 1 0								
												1 0 0						1 1 1 1 0								
												1 0 1						1 0 1 0 0								
												1 1 0						1 0 0 1 0								
												1 1 1						1 0 0 0 1								
0 0 0 0 * 0 0 1								0 0 1				0 0 0				↑ ↓		1 0 0 0 1						} Character pattern (2)		
												0 0 1						0 1 0 1 0								
												0 1 0						0 1 0 1 0								
												0 1 1						1 1 1 1 1								
												1 0 0						0 0 1 0 0								
												1 0 1						0 0 1 0 0								
												1 1 0						0 0 1 0 0								
												1 1 1						0 0 0 0 0								
0 0 0 0 * 1 1 1								1 1 1				0 0 0				↑ ↓								} Cursor position		
												0 0 1														
												1 0 0														
												1 0 1														

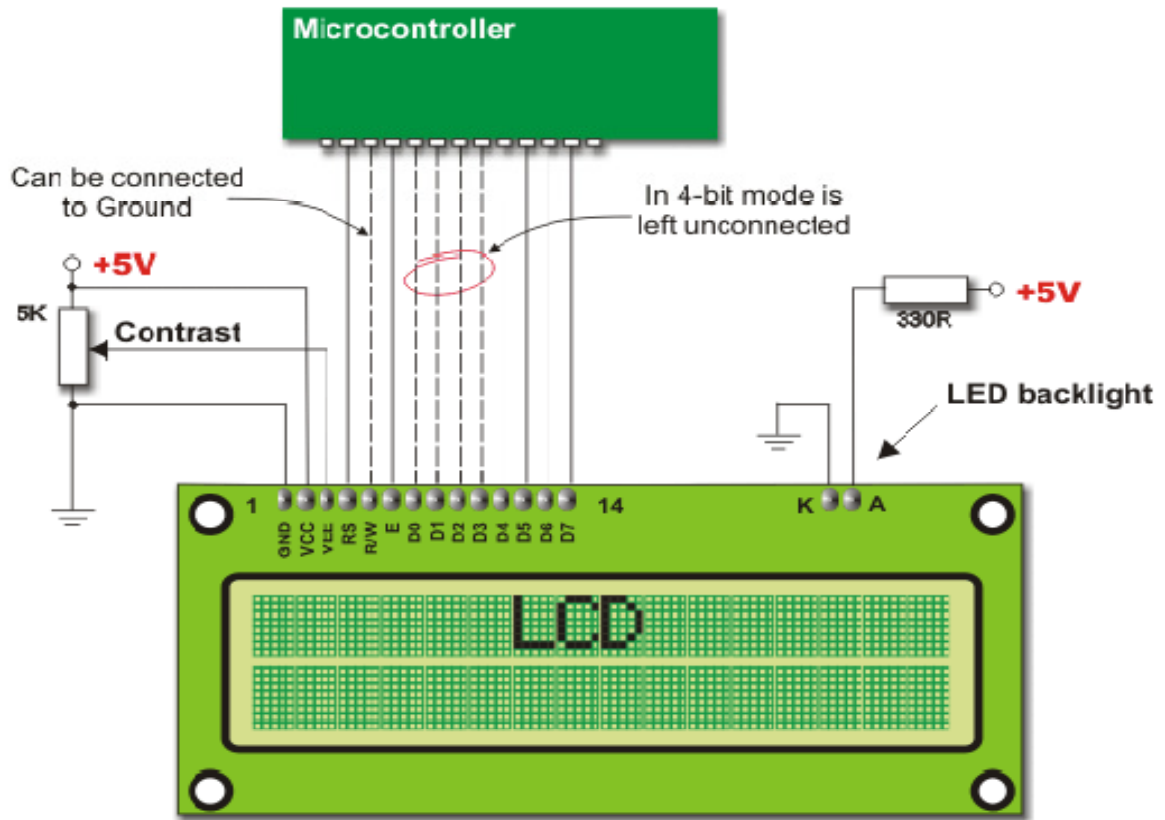
- Notes:
- Character code bits 0 to 2 correspond to CGRAM address bits 3 to 5 (3 bits: 8 types).
 - CGRAM address bits 0 to 2 designate the character pattern line position. The 8th line is the cursor position and its display is formed by a logical OR with the cursor.
 - Character pattern row positions correspond to CGRAM data bits 0 to 4 (bit 4 being at the left).
 - As shown Table 5, CGRAM character patterns are selected when character code bits 4 to 7 are all 0. However, since character code bit 3 has no effect, the R display example above can be selected by either character code 00H or 08H.
 - 1 for CGRAM data corresponds to display selection and 0 to non-selection.
- * Indicates no effect.

Tập lệnh:

Tập lệnh	Mã nhị phân										Mô tả	Thời gian thực thi
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Xoá hiển thị	0	0	0	0	0	0	0	0	0	1	Xoá hiển thị và đưa con trỏ về vị trí ban đầu (địa chỉ 0).	1.64mS
Con trỏ ban đầu	0	0	0	0	0	0	0	0	1	*	Trả con trỏ về vị trí ban đầu (địa chỉ 0). Ngoài ra đưa hiển thị đã bị dịch chuyển về vị trí ban đầu. Nội dung bộ nhớ hiển thị dữ liệu(DDRAM) không thay đổi.	1.64mS
Thiết lập chế độ	0	0	0	0	0	0	0	1	I/D	S	Thiết lập hướng di chuyển của con trỏ tăng/giảm(I/D=0:giảm,I/D=1:tăng), chỉ rõ dịch chuyển hiển thị (S=0:không dịch chuyển hiển thị,S=1:dịch chuyển hiển thị). Hoạt động này được thực hiện trong suốt quá trình đọc/ghi dữ liệu	40uS
Điều khiển hiển thị	0	0	0	0	0	0	1	D	C	B	Bật/tắt hiển thị (D=0:tắt,D=1:bật) nhưng dữ liệu vẫn lưu trong DDRAM, bật/tắt con trỏ(C=0:tắt,C=1:bật) và bật tắt con trỏ nhấp nháy tại vị trí của kí tự (B=0:tắt,B=1:bật).	40uS
Dịch chuyển con trỏ/hiển thị	0	0	0	0	0	1	S/C	R/L	*	*	Dịch chuyển con trỏ/hiển thị qua trái /phải mà không phải đọc/ghi lại dữ liệu (S/C=0:di chuyển con trỏ,S/C=1:di chuyển hiển thị), (R/L=0:dịch trái,R/L=1:dịch phải). nội dung DDRAM không thay đổi.	40uS
Thiết lập chức năng	0	0	0	0	1	DL	N	F	*	*	Khởi tạo giao diện của độ dài dữ liệu (DL=0:độ dài 4 bit,DL=1:8 bit), số hàng hiển thị(N=0:1 hàng,N=1:2 hàng) và phong chữ(F=0:5x7,F=1:5x10).	40uS
Thiết lập địa chỉ CGRAM	0	0	0	1	Địa chỉ CGRAM						Thiết lập địa chỉ bộ nhớ tạo kí tự (CGRAM), dữ liệu được gửi/nhận sau thiết lập này	40uS

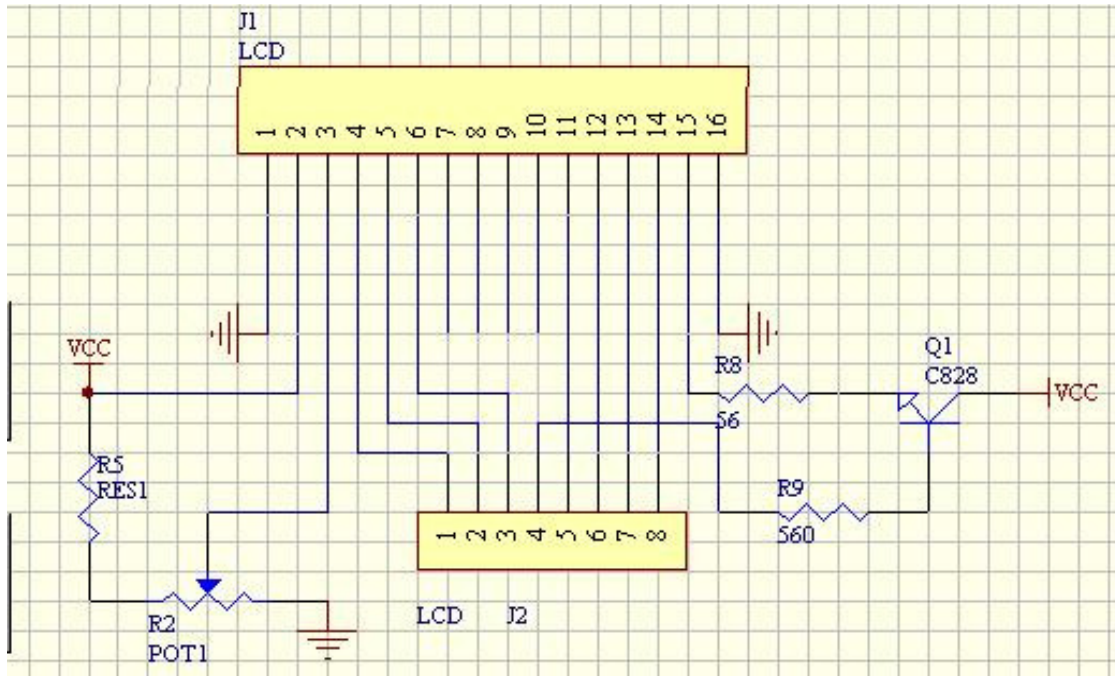
Tập lệnh	Mã nhị phân										Mô tả	Thời gian thực thi
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Thiết lập địa chỉ DDRAM	0	0	1	Địa chỉ DDRAM							Thiết lập địa chỉ bộ nhớ tạo kí tự (DDRAM.), dữ liệu được gửi/nhận sau thiết lập này	40uS
Đọc cờ bận	0	1	BF	Địa chỉ CGRAM / DDRAM							Đọc cờ bận Busy-flag (BF), kiểm tra xem hệ thống có đang thực thi 1 lệnh đã được nhận trước đó không. (BF=1 :hệ thống đang thực hiện tác vụ bên trong, khi BF=0 thì lệnh tiếp theo mới được thực thi)	0uS
Ghi dữ liệu đến CGRAM/ DDRAM.	1	0	Ghi dữ liệu							Ghi dữ liệu đến CGRAM/ DDRAM.	40uS	
Đọc dữ liệu từ CGRAM/ DDRAM.	1	1	Đọc dữ liệu							Đọc dữ liệu từ CGRAM/ DDRAM.	40uS	

Mã (Hex)	Lệnh đến thanh ghi của LCD
1	Xoá màn hình hiển thị
2	Trở về đầu dòng
4	Giả con trỏ (dịch con trỏ sang trái)
6	Tăng con trỏ (dịch con trỏ sang phải)
5	Dịch hiển thị sang phải
7	Dịch hiển thị sang trái
8	Tắt con trỏ, tắt hiển thị
A	Tắt hiển thị, bật con trỏ
C	Bật hiển thị, tắt con trỏ
E	Bật hiển thị, nhấp nháy con trỏ
F	Tắt con trỏ, nhấp nháy con trỏ
10	Dịch vị trí con trỏ sang trái
14	Dịch vị trí con trỏ sang phải
18	Dịch toàn bộ hiển thị sang trái



Copy file "Lcd_4bit.h" vào thư mục C:\Program Files\PICC\Devices

- lcd_init(): Khởi tạo LCD, gọi 1 lần trong hàm main
- lcd_gotoxy(int8 x, int8 y) : Hiện thị tại vị trí cột x , hàng y
- lcd_send_byte(int8 address, int8 n): Gửi 1 byte n đến lcd, address=0: thao tác lệnh, address=1: thao tác dữ liệu
- int8 lcd_read_byte(void) : đọc về 1 byte
- lcd_putc(char c) : Gửi các ký tự lên lcd
- char lcd_getc(int8 x, int8 y) : Đọc ký tự tại cột x, hàng y



Trên đây là cấu trúc cơ bản của LCD1602. Công cụ này rất hữu hiệu trong việc hiển thị cho người quan sát: số, ký tự, hiệu ứng trên đó. Để hiểu cụ thể chúng ta cùng đi vào làm thực tế nhé.