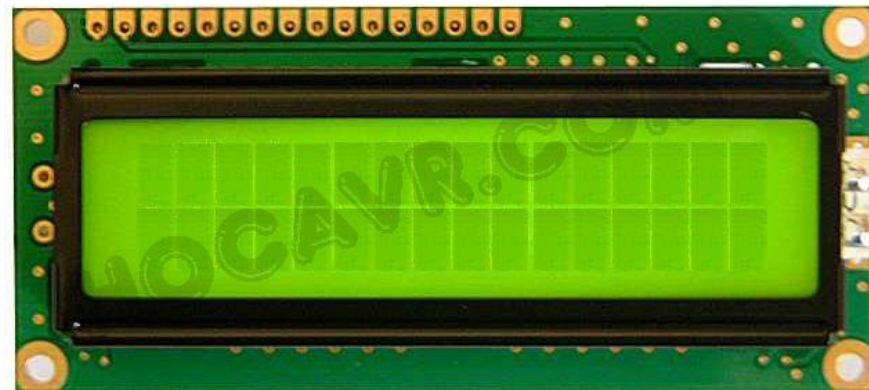


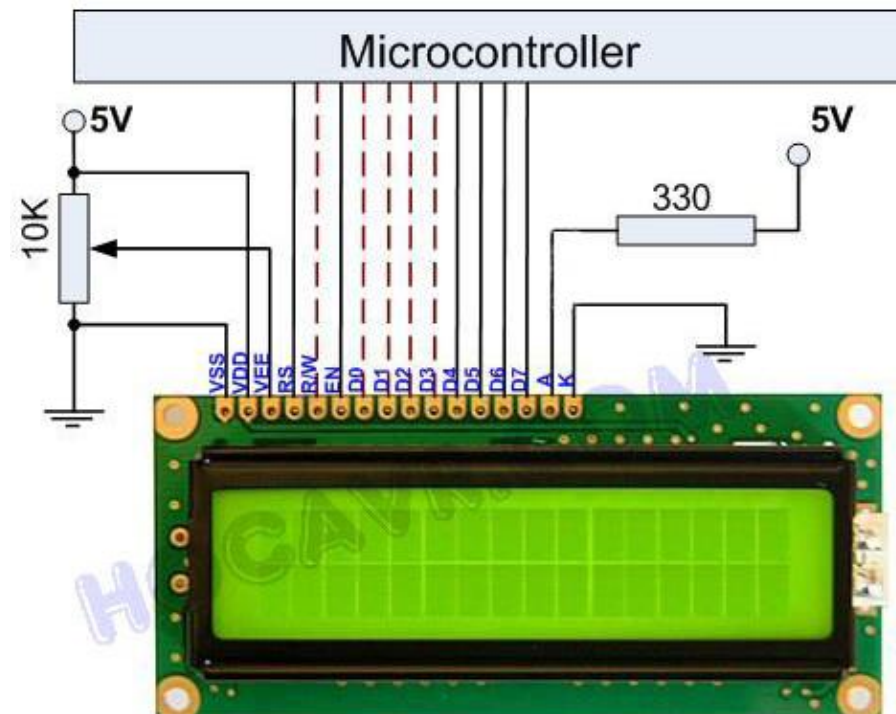
# Basic I/O Functions: LCD

- **Text LCD** là các loại màn hình tinh thể lỏng nhỏ dùng để hiển thị các dòng chữ hoặc số trong bảng mã ASCII.
- Text LCD được chia sẵn thành từng ô và ứng với **mỗi ô chỉ có thể hiển thị một ký tự ASCII** nên được gọi là Text LCD (phân biệt Graphic LCD có thể hiển thị hình ảnh)
- **Kích thước Text LCD** được định nghĩa bằng số ký tự có thể hiển thị trên 1 dòng và tổng số dòng mà LCD có.  
Ví dụ LCD 16x2 là loại có 2 dòng và mỗi dòng có thể hiển thị tối đa 16 ký tự. Một số kích thước Text LCD thông thường gồm 16x1, 16x2, 16x4, 20x2, 20x4...



# Basic I/O Functions: LCD

Chức năng	Pins	Kí hiệu	Giá trị	Mô tả
Ground	1	Vss (GND)		0 V
Nguồn LCD	2	Vdd (VCC)		+5 V
Tương phản	3	Vee		0 - Vdd
Điều khiển	4	RS	0	D0 – D7: lệnh
			1	D0 – D7: dữ liệu
	5	R/W	0	Ghi từ VXL vào LCD
			1	Đọc từ LCD ra VXL
	6	E	0	Vô hiệu hóa LCD
			1	LCD hoạt động
Dữ liệu/lệnh			Từ 1 xuống 0	Bắt đầu ghi/đọc LCD
	7	D0	0/1	Bit 0 LSB
	8	D1	0/1	Bit 1
	9	D2	0/1	Bit 2
	10	D3	0/1	Bit 3
	11	D4	0/1	Bit 4
	12	D5	0/1	Bit 5
	13	D6	0/1	Bit 6
Đèn nền	14	D7	0/1	Bit 7
	15	A (Anode)		+5 V
	16	K (Cathode)		0 V





# Basic I/O Functions: LCD

## ❖ LCD là có 2 thanh ghi 8 bit

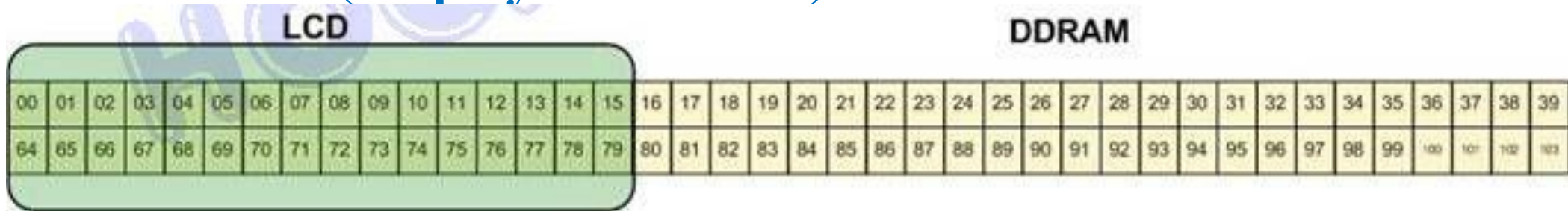
- IR (Instruction Register): Thanh ghi chứa mã lệnh (chỉ ghi)
- DR (Data Register): thanh ghi chứa dữ liệu data

## ❖ LCD có 3 loại bộ nhớ

- DDRAM (Display Data RAM): chứa dữ liệu cần hiển thị
- CGROM (Character Generator ROM): bộ nhớ ROM chứa bộ font tạo ra ký tự
- CGRAM (Character Generator RAM): bộ nhớ RAM chứa bộ font tạo ra các symbol tùy chọn

# Basic I/O Functions: LCD

## ❖ DDRAM (Display Data RAM)



DDRAM là bộ nhớ tạm chứa các ký tự cần hiển thị lên LCD, bộ nhớ này gồm có 80 ô được chia thành 2 hàng, mỗi ô có độ rộng 8 bit và được đánh số từ 0 đến 39 cho dòng 1; từ 64 đến 103 cho dòng 2. Mỗi ô nhớ tương ứng với 1 ô trên màn hình LCD

LCD loại 16x2 có thể hiển thị tối đa 32 ký tự (có 32 ô hiển thị), vì thế có một số ô nhớ của DDRAM không được sử dụng làm các ô hiển thị

Loại LCD LM016L trong Proteus có 2 hàng: hàng trên có 16 ô nhớ bắt đầu từ địa chỉ 0x80; hàng dưới có 16 ô nhớ bắt đầu từ địa chỉ 0xC0

Khi viết mã ASCII ký tự vào 1 trong 32 địa chỉ trên sẽ hiển thị ký tự ra LCD



# Basic I/O Functions: LCD

## ❖ CGROM (Character Generator ROM)

Lower 4 Bits	Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)				0	1	P		F				-	9	3	α	p
xxxx0001	(2)			!	1	A	Q	a	4				.	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	Ω
xxxx0101	(6)			%	5	E	U	e	u				・	オ	ナ	υ	ϖ
xxxx0110	(7)			&	6	F	V	f	v				ヲ	カ	ニ	ρ	Σ
xxxx0111	(8)			'	7	G	W	g	w				ア	キ	ヌ	q	π
xxxx1000	(1)			(	8	H	X	h	x				ィ	ク	ネ	Ј	⌘
xxxx1001	(2)			)	9	I	Y	i	y				ゥ	ケ	ル	ʹ	ϣ
xxxx1010	(3)			*	:	J	Z	j	z				エ	コ	レ	j	〒
xxxx1011	(4)			+	;	K	[	k	{				オ	サ	ヒ	⌘	⌘
xxxx1100	(5)			,	<	L	¥	l					カ	シ	フ	⌘	⌘
xxxx1101	(6)			-	=	M	]	m	}				ユ	ズ	ン	⌘	÷
xxxx1110	(7)			.	>	N	^	n	~				ヨ	セ	ホ	ゝ	⌘
xxxx1111	(8)			/	?	O	_	o	+				ッ	ソ	マ	〇	■

**CGROM** là vùng nhớ cố định chứa định nghĩa font cho các ký tự. Chúng ta không trực tiếp truy xuất vùng nhớ này mà chip xử lý của LCD sẽ tự thực hiện khi có yêu cầu đọc font để hiện thị. Một điều đáng lưu ý là địa chỉ vùng nhớ CGROM chính là mã ASCII của ký tự đó. Ví dụ ký tự ‘a’ có mã ASCII là 97, tham khảo tổ chức của vùng nhớ CGROM trong hình 4 bạn sẽ nhận thấy địa chỉ font của ‘a’ có 4 bit thấp là 0001 và 4 bit cao là 0110, địa chỉ tổng hợp là  $01100001 = 97$





# Basic I/O Functions: LCD

## ❖ CGRAM (Character Generator RAM):

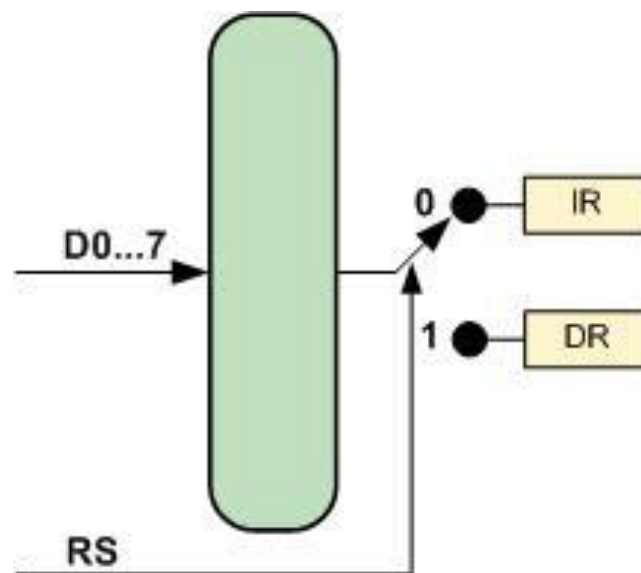
**CGRAM** là vùng nhớ chứa các symbol do người dùng tự định nghĩa, mỗi symbol được có kích thước 5x8 và được dành cho 8 ô nhớ 8 bit. Các symbol thường được định nghĩa trước và được gọi hiển thị khi cần thiết. Vùng này có tất cả 64 ô nhớ nên có tối đa 8 symbol có thể được định nghĩa. (Cách sử dụng bộ nhớ CGRAM có thể tham khảo datasheet của LCD)



# Basic I/O Functions: LCD

## ĐIỀU KHIỂN HIỂN THỊ TEXT RA LCD

**Chân điều khiển RS (chân số 3):** nếu muốn gọi 1 mã lệnh đến LCD thì chân RS phải được reset về 0. Ngược lại, khi muốn ghi mã ASCII của ký tự cần hiển thị lên LCD thì chúng ta sẽ set RS=1 để chọn thanh ghi DR





# Basic I/O Functions: LCD

## ĐIỀU KHIỂN HIỂN THỊ TEXT RA LCD

**Chân điều khiển R/W (chân số 4):** Chân lựa chọn giữa việc đọc và ghi

- **Nếu  $R/W=0$**  thì dữ liệu sẽ được ghi từ bộ điều khiển ngoài vào LCD
- **Nếu  $R/W=1$**  thì dữ liệu sẽ được đọc từ LCD ra ngoài

**Chú ý:** Phải đọc trạng thái LCD để biết LCD có đang bận hay không (cờ Busy Flag - BF): Nếu  $BF=1$  thì chúng ta phải chờ cho LCD xử lí xong nhiệm vụ hiện tại, đến khi nào  $BF=0$  một thao tác mới sẽ được gán cho LCD. Có 2 cách thực hiện

- Cách 1 là đọc bit BF về kiểm tra và chờ  $BF=0$
- Cách 2 là viết một hàm delay một khoảng thời gian cố định nào đó (tốt nhất là trên 1ms). Khi đó kết nối chân R/W của LCD xuống GND





# Basic I/O Functions: LCD

## ĐIỀU KHIỂN HIỂN THỊ TEXT RA LCD

**Chân điều khiển EN (chân số 5):** Chân cho phép LCD hoạt động (Enable), chân này cần được kết nối với bộ điều khiển để cho phép thao tác LCD. Để **đọc và ghi data** từ LCD chúng ta cần tạo một “**xung cạnh xuống**” trên **chân EN**, nói theo cách khác, muốn ghi dữ liệu vào LCD trước hết cần đảm bảo rằng chân  $EN=0$ , tiếp đến xuất dữ liệu đến các chân  $D0:7$ , sau đó set chân EN lên 1 và cuối cùng là xóa EN về 0 để tạo 1 xung cạnh xuống



# Basic I/O Functions: LCD

## ĐIỀU KHIỂN HIỂN THỊ TEXT RA LCD

Instruction	Code										Description	Execution Time (max) (when $f_{cp}$ or $f_{osc}$ is 270 kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address 0 in address counter.	
Return home	0	0	0	0	0	0	0	0	0	1	Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged.	1.52 ms
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	37 $\mu$ s
Display on/off control	0	0	0	0	0	0	1	D	C	B	Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B).	37 $\mu$ s
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	—	—	Moves cursor and shifts display without changing DDRAM contents.	37 $\mu$ s
Function set	0	0	0	0	1	DL	N	F	—	—	Sets interface data length (DL), number of display lines (N), and character font (F).	37 $\mu$ s
Set CGRAM address	0	0	0	1	ACG	ACG	ACG	ACG	ACG	ACG	Sets CGRAM address. CGRAM data is sent and received after this setting.	37 $\mu$ s
Set DDRAM address	0	0	1	ADD	ADD	ADD	ADD	ADD	ADD	ADD	Sets DDRAM address. DDRAM data is sent and received after this setting.	37 $\mu$ s
Read busy flag & address	0	1	BF	AC	AC	AC	AC	AC	AC	AC	Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents.	0 $\mu$ s

Write data to CG or DDRAM	1	0	Write data	Writes data into DDRAM or CGRAM.	37 $\mu$ s $t_{ADD} = 4 \mu s^*$
Read data from CG or DDRAM	1	1	Read data	Reads data from DDRAM or CGRAM.	37 $\mu$ s $t_{ADD} = 4 \mu s^*$
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: 5 $\times$ 10 dots, F = 0: 5 $\times$ 8 dots BF = 1: Internally operating BF = 0: Instructions acceptable				DDRAM: Display data RAM CGRAM: Character generator RAM ACG: CGRAM address ADD: DDRAM address (corresponds to cursor address) AC: Address counter used for both DD and CGRAM addresses	Execution time changes when frequency changes Example: When $f_{cp}$ or $f_{osc}$ is 250 kHz, $37 \mu s \times \frac{270}{250} = 40 \mu s$
Note: — indicates no effect. * After execution of the CGRAM/DDRAM data write or read instruction, the RAM address counter is incremented or decremented by 1. The RAM address counter is updated after the busy flag turns off. In Figure 10, $t_{ADD}$ is the time elapsed after the busy flag turns off until the address counter is updated.					



# Basic I/O Functions: LCD

## ĐIỀU KHIỂN HIỂN THỊ TEXT RA LCD

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*





# Basic I/O Functions: LCD

```
void LCD_Enable(void)
{
    output_high(LCD_EN);
    delay_us(3);
    output_low(LCD_EN);
    delay_us(50);
}
```

```
//Ham Gui 4 Bit Du Lieu Ra LCD
void LCD_Send4Bit( unsigned char
Data )
{
    output_bit(LCD_D4,Data&0x01);
    output_bit(LCD_D5,(Data>>1)&1);
    output_bit(LCD_D6,(Data>>2)&1);
    output_bit(LCD_D7,(Data>>3)&1);
}
```



# Basic I/O Functions: LCD

```
// Ham Gui 1 Lenh Cho LCD
void LCD_SendCommand
(unsigned char command )
{
LCD_Send4Bit ( command >>4 );
/* Gui 4 bit cao */
LCD_Enable () ;
LCD_Send4Bit ( command );
/* Gui 4 bit thap*/
LCD_Enable () ;
}
```

```
void LCD_Init ( void )
{
output_low(LCD_RS);
output_low(LCD_RW);
delay_ms(20);
LCD_Send4Bit(0x03);
LCD_Enable();
delay_ms(10);
LCD_Send4Bit(0x02);
LCD_Enable();
delay_ms(10);
}
```





# Basic I/O Functions: LCD

```
LCD_SendCommand( 0x28 );    // giao thuc 4  
bit, hien thi 2 hang, ki tu 5x8
```

```
//Function Set: 8-bit, 1 Line, 5x7 Dots  0x30
```

```
//Function Set: 8-bit, 2 Line, 5x7 Dots  0x38
```

```
//Function Set: 4-bit, 1 Line, 5x7 Dots  0x20
```

```
//Function Set: 4-bit, 2 Line, 5x7 Dots  0x28
```

```
LCD_SendCommand( 0x0F );    //Display on  
Cursor on: 0x0E  0x0F
```

```
//Display on Cursor off: 0x0C
```

```
//Display on Cursor blinking: 0x0F
```

```
LCD_SendCommand( 0x06 );    // tang ID,  
khong dich khung hinh      Entry Mode
```

```
LCD_SendCommand( 0x01 );    // Clear Display  
(also clear DDRAM content)  
}
```

```
void LCD_PutChar ( unsigned char Data )  
{  
    output_high(LCD_RS);  
    LCD_SendCommand( Data );  
    output_low(LCD_RS);  
}
```

```
void LCD_Puts (char *s)  
{  
    while (*s)  
    {  
        LCD_PutChar(*s);  
        s++;  
    }  
}
```



# Basic I/O Functions: LCD

```
void LCD_Clear()
{
    LCD_SendCommand(0x01);
    delay_ms(10);
}
```

```
void LCD_Gotoxy(unsigned char
row, unsigned char col)
{
    unsigned char address;
    if(!row)
        address = (0x80 + col);
    else
        address = (0xC0 + col);
    delay_us(1000);
    LCD_SendCommand(address);
    delay_us(50);
}
```