



# **Embedded Systems Design (630414)**

Lecture 12

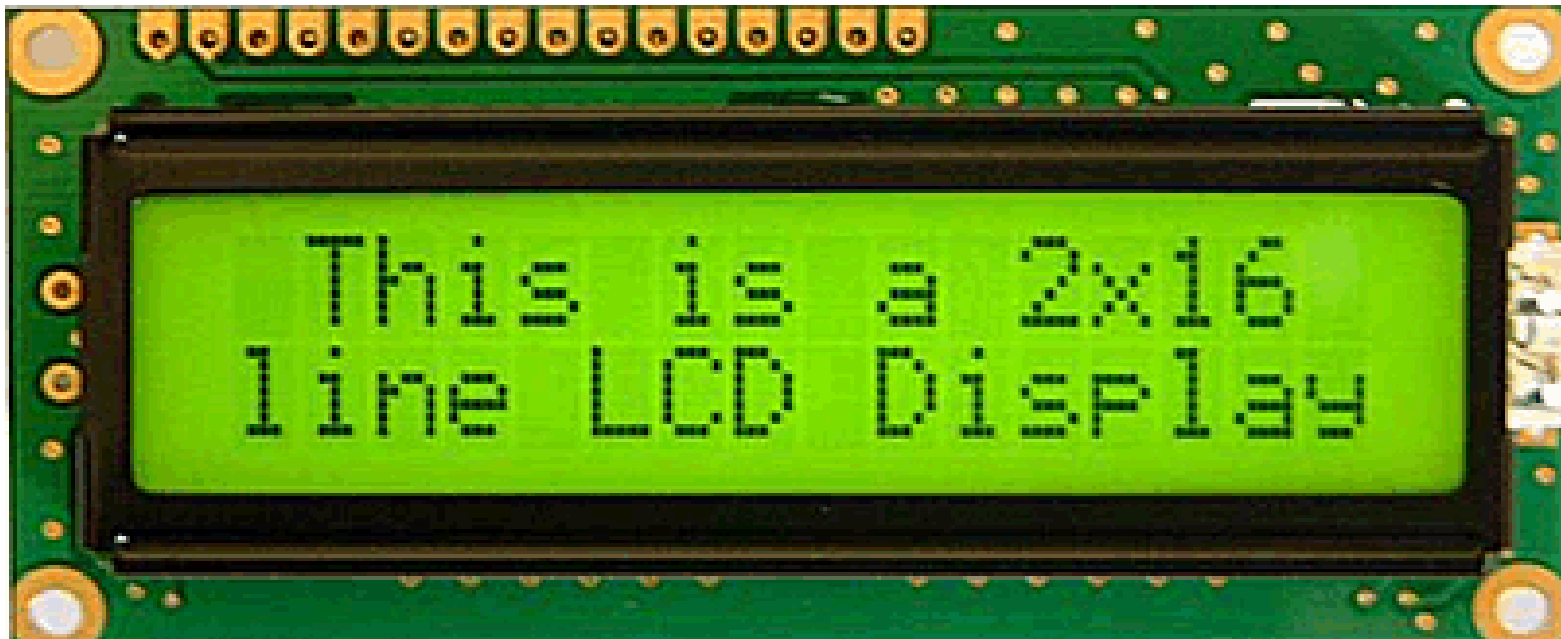
## **LCD Interfacing**

**Prof. Kasim M. Al-Aubidy**

Philadelphia University

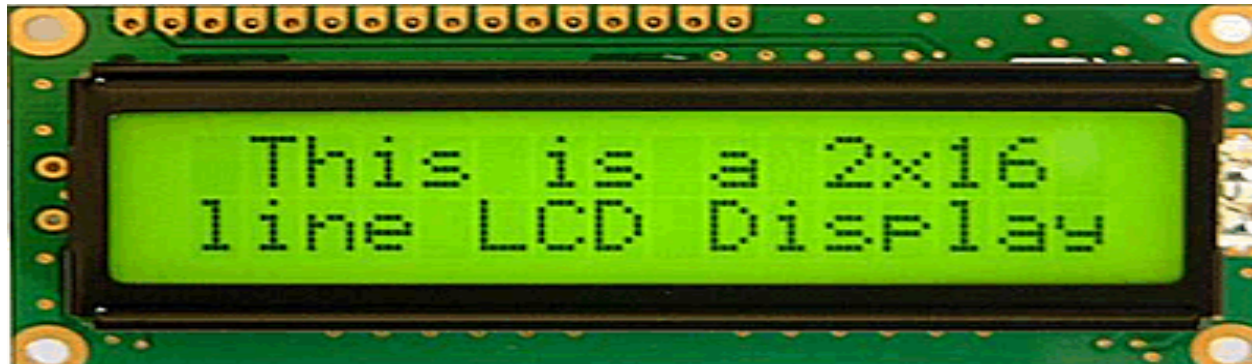
## LCD DISPLAY:

- This component is specifically manufactured to be used with microcontrollers. It is used for displaying different messages on a small liquid crystal display.



## LCD DISPLAY: Simple Model.

- The model described here is for its low price and great capabilities most frequently used in practice:
  - It can display messages in two lines with 16 characters each.
  - It can display all the letters of alphabet, Greek letters, punctuation marks, mathematical symbols etc.
  - It is also possible to display symbols made up by the user.
  - Other useful features include automatic message shift (left and right), cursor appearance, LED backlight etc.

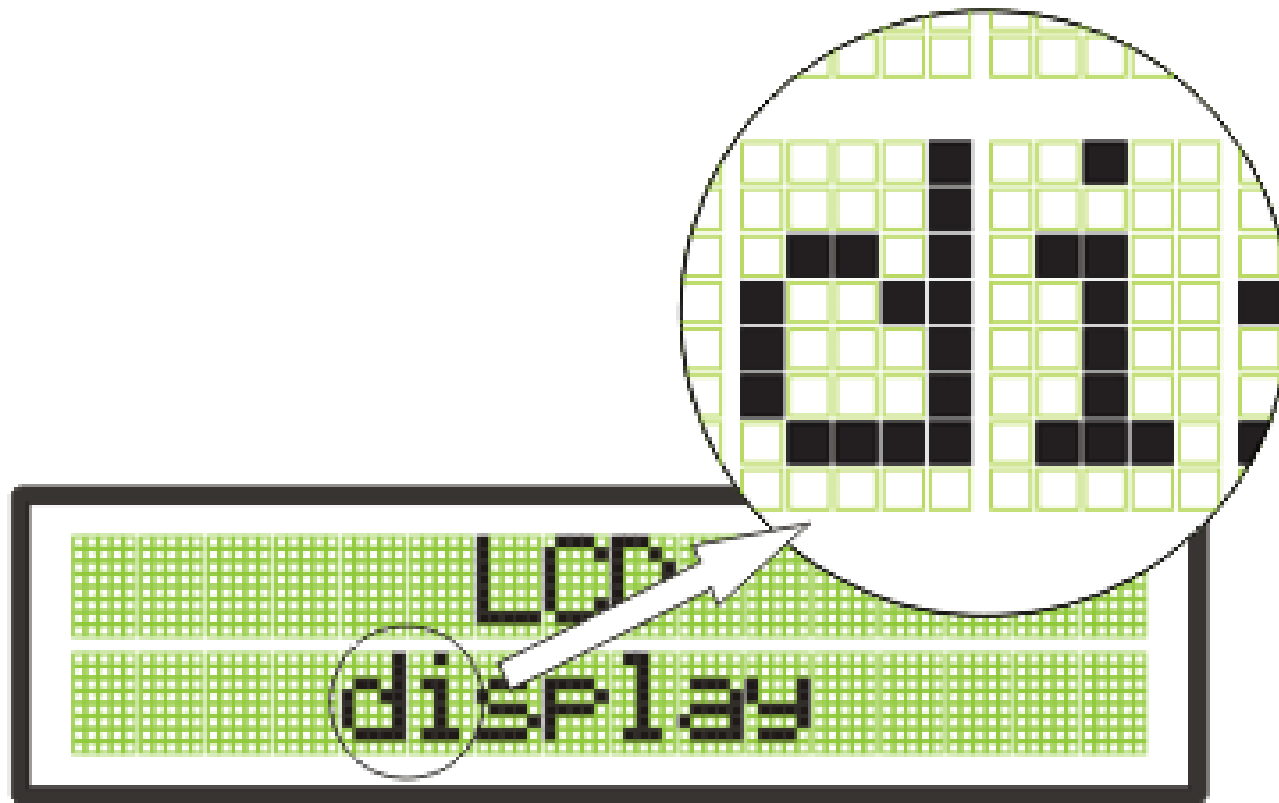


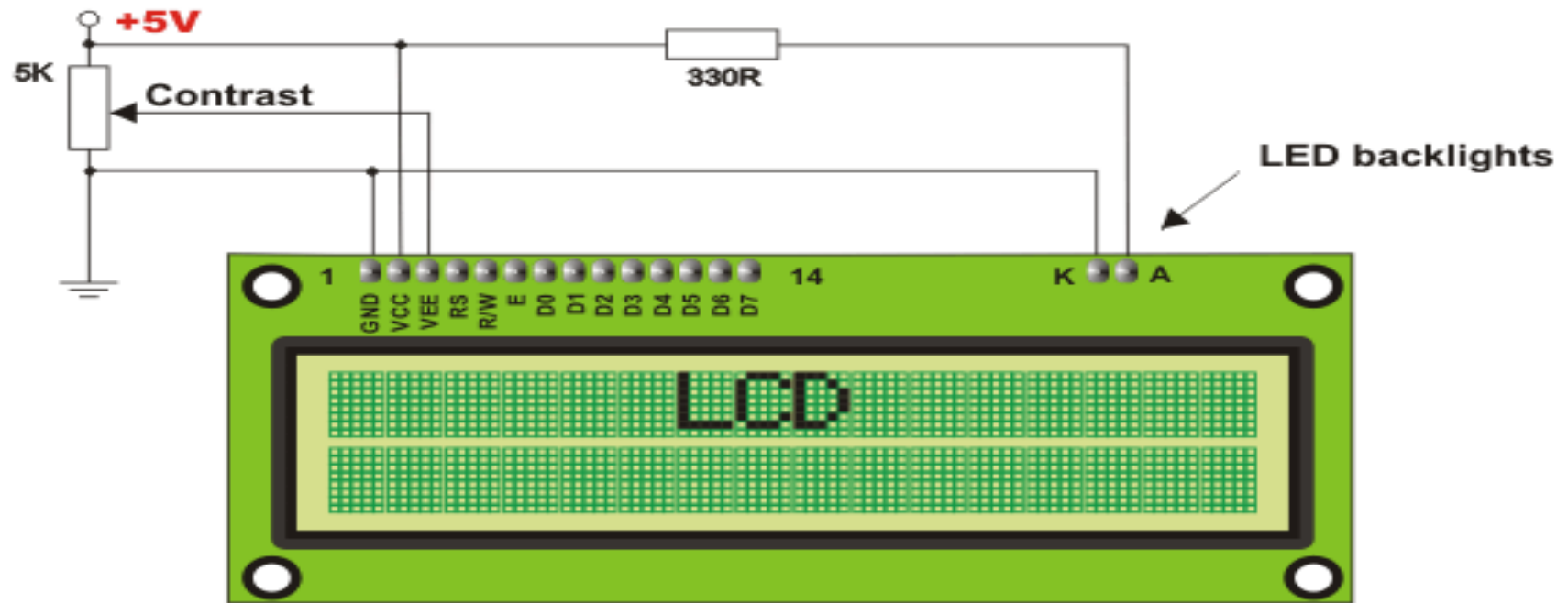
## LCD Display Pins:

Function	Pin No.	Name	Logic State	Description
Ground	1	Vss	-	0V
Power supply	2	Vdd	-	+5V
Contrast	3	Vee	-	0-Vdd
Control Of Operation	4	RS	0 1	D0 – D7 are interpreted as commands D0 – D7 are interpreted as data
	5	R/W	0 1	Write data (from controller to LCD) Read data (from LCD to controller)
	6	E	0 1 from 1 to 0	Access to LCD disabled Normal operating Data/commands are transferred to LCD
Data	7 to 14	D0-D7	0/1	Data Bus

## LCD Screen:

An LCD screen can display two lines with 16 characters each. Every character consists of 5x8 or 5x11 dot matrix. This book covers a 5x8 character display which is most commonly used.





Display contrast depends on the power supply voltage and whether messages are displayed in one or two lines. For this reason, varying voltage 0-V<sub>dd</sub> is applied to the pin marked as V<sub>ee</sub>. A trimmer potentiometer is usually used for this purpose. Some of the LCD displays have built-in backlight (blue or green LEDs). When used during operation, a current limiting resistor should be serially connected to one of the pins for backlight power supply (similar to LED diodes).

## **LCD Memory:**

LCD display contains three memory blocks:

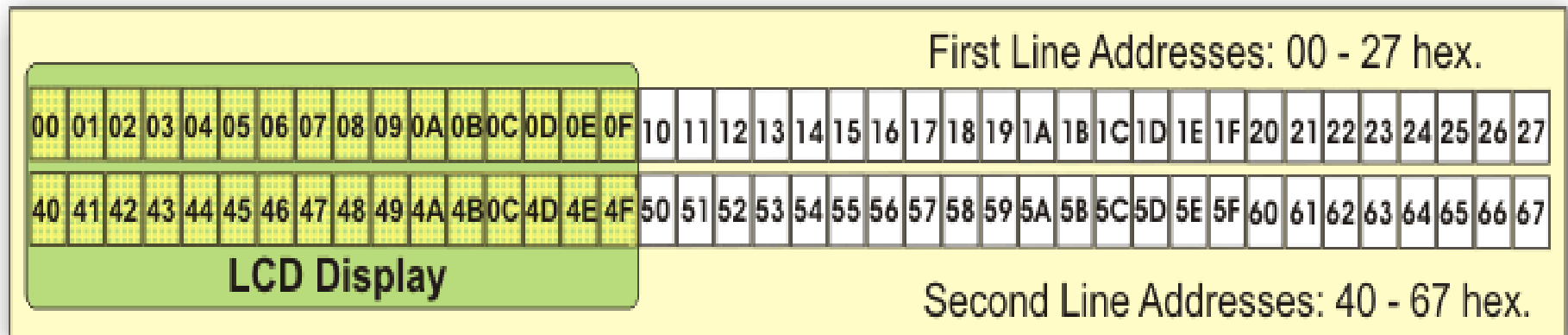
1. DDRAM Display Data RAM;
2. CGRAM Character Generator RAM; and
3. CGROM Character Generator ROM.

### **1. DDRAM Memory:**

- DDRAM memory is used for storing characters to be displayed. The size of this memory is capable of storing 80 characters. Some memory locations are directly connected to the characters on display.
- Everything works quite simply: it is enough to configure the display to increment addresses automatically (shift right) and set the starting address for the message to be displayed (for example 00 hex).

- Afterwards, all characters sent through lines D0-D7 will be displayed in the message format.
- In this case, displaying starts from the first field of the first line because the initial address is 00 hex.
- If more than 16 characters are sent, then all of them will be memorized, but only the first sixteen characters will be visible. In order to display the rest of them, the shift command should be used.

## DDRAM Memory





## 2. CGROM Memory:

- CGROM memory contains a standard character map with all characters that can be displayed on the screen. Each character is assigned to one memory location:

		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	≡	α	ρ		
	xxxx0001	(2)	!	1	A	Q	a	q				。	ア	チ	△	ã	q	
	xxxx0010	(3)	"	2	B	R	b	r				「	イ	ツ	×	β	θ	
	xxxx0011	(4)	#	3	C	S	c	s				」	ウ	テ	モ	ε	∞	
	xxxx0100	(5)	\$	4	D	T	d	t				、	エ	ト	ト	μ	Ω	
	xxxx0101	(6)	%	5	E	U	e	u					・	オ	ナ	1	ε	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					イ	ク	ネ	リ	γ	×
	xxxx1001	(2)	>	9	I	Y	i	y					ウ	ケ	ル	ル	γ	γ
	xxxx1010	(3)	*	:	J	Z	j	z					エ	コ	ハ	レ	j	キ
	xxxx1011	(4)	+	;	K	[	k	[					オ	サ	ヒ	ロ	*	万
	xxxx1100	(5)	,	<	L	¥	l	l					カ	シ	フ	ワ	φ	円
	xxxx1101	(6)	-	=	M	]	m	]					ユ	ズ	ハ	ン	も	÷
	xxxx1110	(7)	.	>	N	^	n	^					ヨ	セ	ホ	°	ん	
	xxxx1111	(8)	/	?	O	_	o	_					ッ	ソ	マ	°	ö	■

## LCD Basic Commands:

- All data transferred to an LCD through the outputs D0-D7 will be interpreted as a command or a data, which depends on the RS pin logic state:
- **RS = 1** - Bits D0 - D7 are addresses of the characters to be displayed. LCD processor addresses one character from the character map and displays it. The DDRAM address specifies location on which the character is to be displayed. This address is defined prior to transferring character or the address of the previously transferred character is automatically incremented.
- **RS = 0** - Bits D0 - D7 are commands for setting the display mode.
- Here is a list of commands recognized by the LCD:

<b>Command</b>	<b>RS</b>	<b>RW</b>	<b>D7</b>	<b>D6</b>	<b>D5</b>	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>	<b>Execution Time</b>
Clear display	0	0	0	0	0	0	0	0	0	1	1.64mS
Cursor home	0	0	0	0	0	0	0	0	1	x	1.64mS
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	40uS
Display on/off control	0	0	0	0	0	0	1	D	U	B	40uS
Cursor/Display Shift	0	0	0	0	0	1	D/C	R/L	x	x	40uS
Function set	0	0	0	0	1	DL	N	F	x	x	40uS
Set CGRAM address	0	0	0	1	CGRAM address						40uS
Set DDRAM address	0	0	1	DDRAM address						40uS	
Read "BUSY" flag (BF)	0	1	BF	DDRAM address						-	
Write to CGRAM or DDRAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	40uS
Read from CGRAM or DDRAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	40uS

**I/D:** 1 = Increment (by 1)  
0 = Decrement (by 1)

**S:** 1 = Display shift on  
0 = Display shift off

**D:** 1 = Display on  
0 = Display off

**U:** 1 = Cursor on  
0 = Cursor off

**B:** 1 = Cursor blink on  
0 = Cursor blink off

**R/L:** 1 = Shift right  
0 = Shift left

**DL:** 1 = 8-bit interface  
0 = 4-bit interface

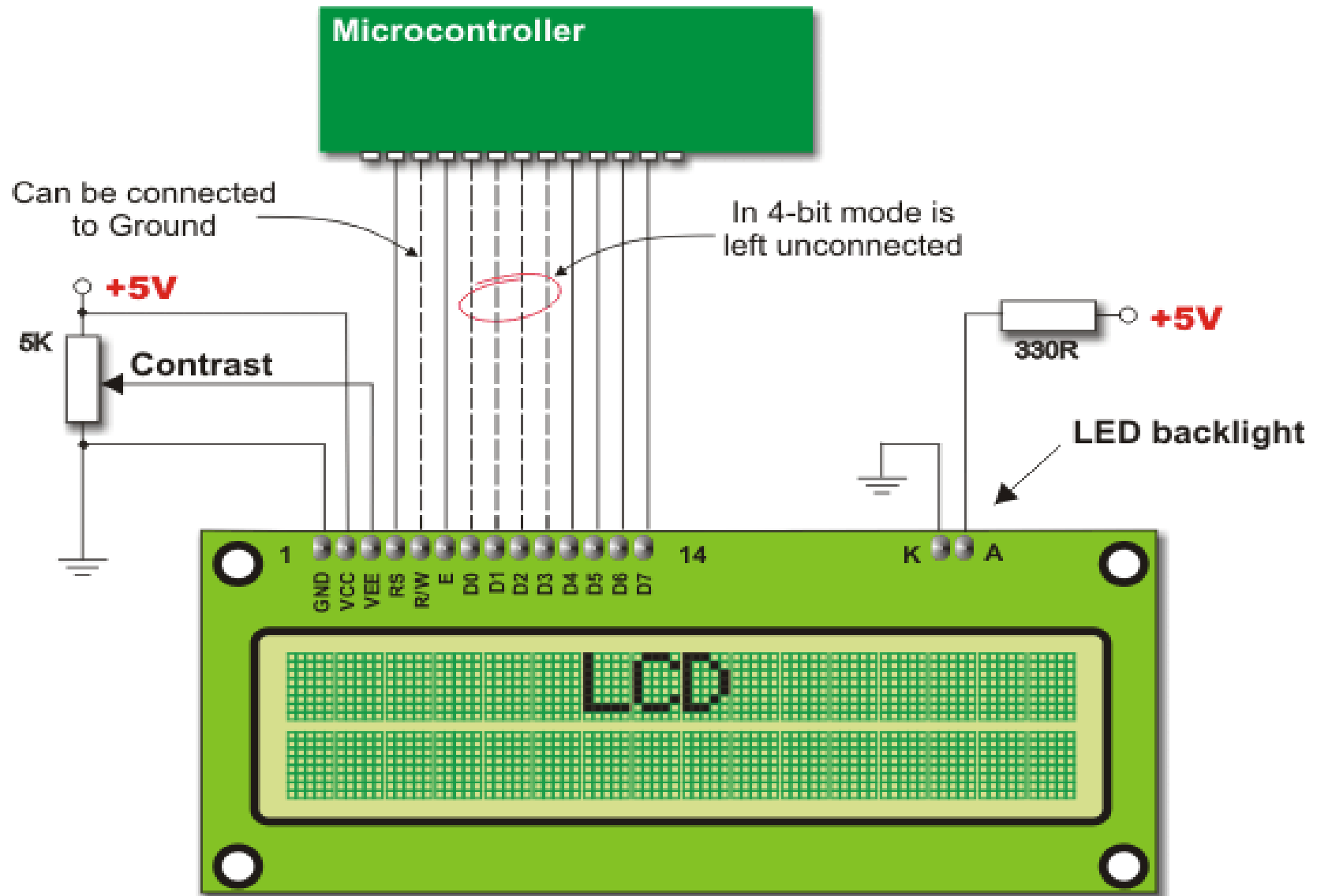
**N:** 1 = Display in two lines  
0 = Display in one line

**F:** 1 = Character format 5x10 dots  
0 = Character format 5x7 dots

**D/C:** 1 = Display shift  
0 = Cursor shift

## LCD Connecting

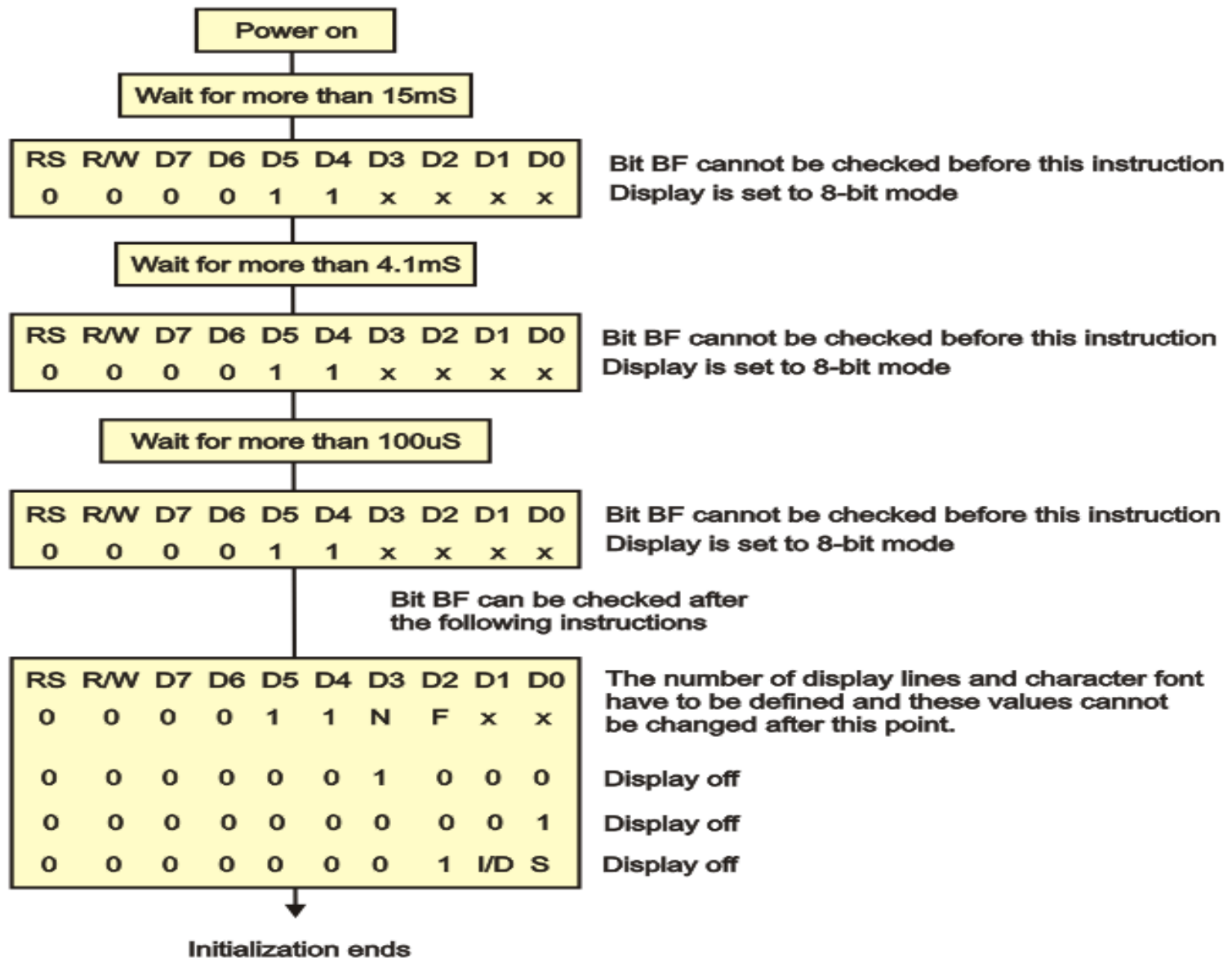
- Depending on how many lines are used for connecting an LCD to the microcontroller, there are 8-bit and 4-bit LCD modes. The appropriate mode is selected at the beginning of the operation in the process called 'initialization'. The 8-bit LCD mode uses outputs D0- D7 to transfer data as explained on the previous page.
- The main purpose of the 4-bit LCD mode is to save valuable I/O pins of the microcontroller. Only 4 higher bits (D4-D7) are used for communication, while others may be left unconnected. Each piece of data is sent to the LCD in two steps- four higher bits are sent first (normally through the lines D4-D7), then four lower bits. Initialization enables the LCD to link and interpret received bits correctly.



## LCD Initialization:

- The LCD is automatically cleared when powered up. It lasts for approximately 15mS. After this, it is ready for operation. The mode of operation is set by default, which means that:
- Display is cleared.
- Mode
  - DL** = 1 - Communication through 8-bit interface
  - N** = 0 - Messages are displayed in one line
  - F** = 0 - Character font 5 x 8 dots
- Display/Cursor on/off
  - D** = 0 - Display off
  - U** = 0 - Cursor off
  - B** = 0 - Cursor blink off
- Character entry
  - ID** = 1 Displayed addresses are automatically incremented by 1
  - S** = 0 Display shift off
- There are two initialization algorithms. Which one is to be performed depends on whether connecting to the microcontroller is through 4- or 8-bit interface. In both cases, all that's left to do after initialization is to specify basic commands and of course - to display messages.

- Refer to figure below for the procedure in 8-bit initialization:





The procedure in 4-bit initialization is as follows:

