3,337,751 members and growing! Email Password Sign in ☑ Remember me? Lost your pass

Home MFC/C++ C# ASP.NET .NET VB.NET All Topics Help! Articles Message Boards StoreFront Lounge

All Topics, C#, .NET >> C# Programming >> General

I/O Ports Uncensored Part 2 - Controlling LCDs (Liquid Crystal Displays) and VFDs (Vacuum Fluorescent Displays) with Parallel Port

By Levent Saltuklaroglu.

C#
Windows (WinXP, Win2
Win95, Win98, WinME
.NET
Win32, VS
Dev

Posted	: **29 Sep 200**
Views	: **421,139**

Controlling LCDs (Liquid Crystal Displays) and VFDs (Vacuum Fluorescent Displays) with Parallel Port

Programming books

Ultimate Combo MFC tools $449

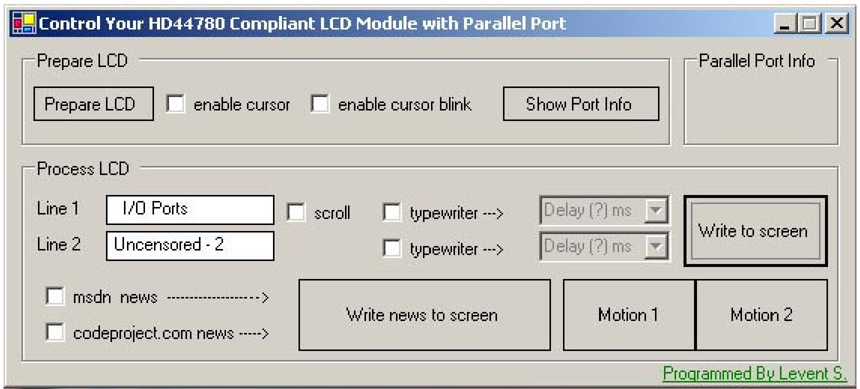
Send to a friend

Search Articles Go! Advanced Search Sitemap

Print Broken Article? Bookmark Discuss

305 votes for this article.
Popularity: 12.12. Rating: **4.88** out

- LCD Demo Application (Initialization and Execution) - 18.1 Kb
- LCD App Source Files - 17.8 Kb



Control Your HD44780 Compliant LCD Module with Parallel Port

Prepare LCD

Prepare LCD □ enable cursor □ enable cursor blink Show Port Info

Parallel Port Info

Process LCD

Line 1: I/O Ports □ scroll □ typewriter ---> Delay (?) ms

Line 2: Uncensored - 2 □ typewriter ---> Delay (?) ms

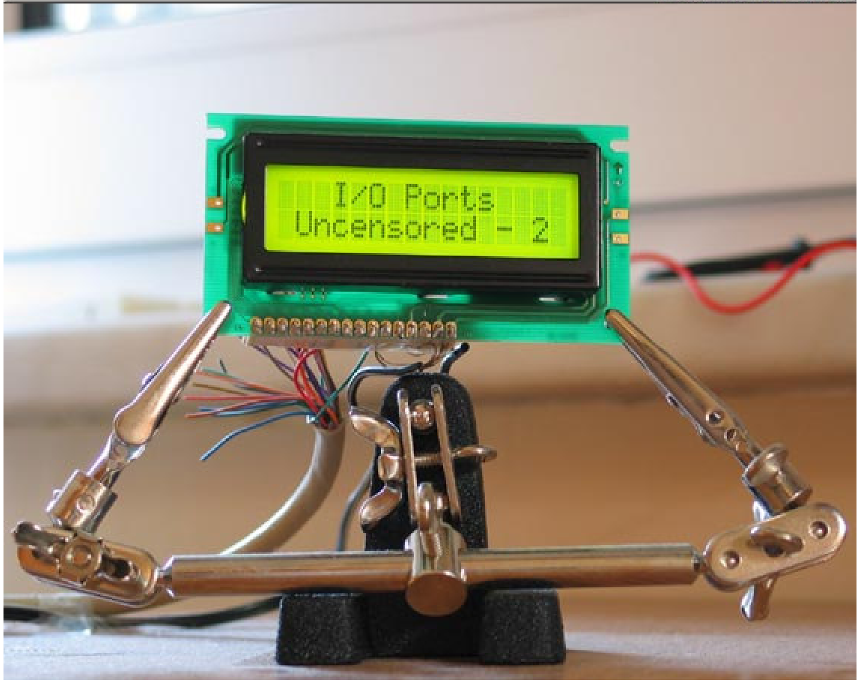
Write to screen

□ msdn news ------------------->

□ codeproject.com news ----->

Write news to screen Motion 1 Motion 2

Programmed By Levent S.



Before You Start

If you want to understand the logic of these I/O ports and want to make a quick start to LCD you have to check out my first article. If you don't need the logic and only need a quick live work, do the circuit in part 2 and download my program to make LCD live.

Part 1 - What is an LCD and what is it for?

Don't be afraid, I won't give a history lesson like every other article out there does but I want to say why we are using all these LCDs from the time when it was first invented. A first we used LEDs and then 7SDs appeared now we are using LCDs because of their capability of showing not only numbers also showing letters and all kind of symbols. I us a character based LCD see it's datasheet here. It has a HD44780(Hitachi) compatible KS0066U controller. Samsung made this controller and if you are interested in the electronic stuff, you can download the controller info from here These controllers are not much complicated like today's graphic LCDs' (Palms, New Generation Digital Cameras, Nokia and other cell phones) controllers but they have been used in commercial and industrial products around the world. Because of their basic interface they can talk to the old Z80, 6502 and/or the PIC microcontrollers which are considered as one of the most popular microcontrollers by the hobbyists. You can find additional info on microcontroller at http://www.microchip.com/

Let me explain what VFDs are: VFDs (Vacuum Fluorescent Displays) are also like LCDs. These are commonly used in microwaves and some pocket calculators. See a sample picture of a VFD here. The key is controllers so there are HD44780 compliant VFDs arou I like VFDs more than LCDs because they are brighter than LCDs but they are also more expensive than LCDs so I used an LCD in my article.

What are the shapes of these LCDs? I used a 2x16 (2 row(line) x 16 column) LCD in my article as you can see in the above picture. There are lots of LCD types such as 1x8, 2x8 1x16, 1x20, 2x20, 2x20, 2x10, 1x40 and there are also many different vendors. And if there are more than 80 characters, mostly they use 2 controllers on the LCD (Mine has o - This is important). I don't give any lcd structure information due to codeproject's conce if somebody wants to learn about how these LCD structures are check these two urls:

1. Samsung's LCD structure and type article
2. Sharp's The Principles of LCD Technology article

Part 2 - LCD Connections, Pinout Descriptions and Circuit

LCD Connections

Mostly an LCD has a 14 pin connection which are D0-D7 (Data Bits), E (Enable), R/W (Read/Write), RS (Register Select), Vee or Vo (Contrast), Vdd or Vcc (+5V supply), Vss (Ground/Earth connection) so there are 8 data, 3 power and 3 control lines for a standar non-backlight LCD. If an LCD has a backlight it has additional two lines for that which ar 15th (VB+) and 16th(VB- or Ground) lines. Below there is a table of the pin matches an pictures of my LCD's backsides:

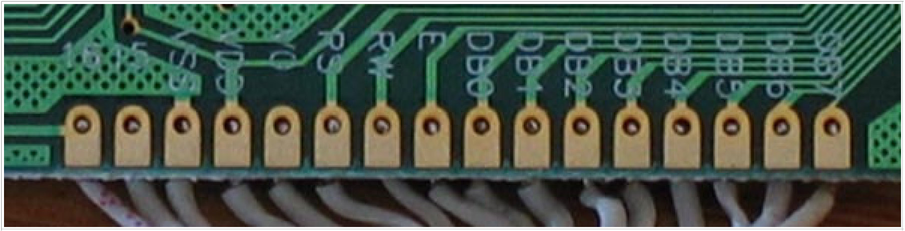
Standard LCD Pin Matches (Character number <80)

Pin	Symbol/Alternate Symbol	Possibility	Function
1	Vss	-	Power supply (GND)
2	Vdd/Vcc	-	Power supply (+5V)
3	Vee/Vo	-	Contrast adjust
4	RS	0/1	0 = Instruction input / 1 = Data input
5	R/W	0/1	0 = Write to LCD module / 1 = Read from LCD module
6	E	1, 1-->0	Enable signal
7	DB0	0/1	Data pin 0
8	DB1	0/1	Data pin 1
9	DB2	0/1	Data pin 2
10	DB3	0/1	Data pin 3
11	DB4	0/1	Data pin 4
12	DB5	0/1	Data pin 5
13	DB6	0/1	Data pin 6
14	DB7	0/1	Data pin 7

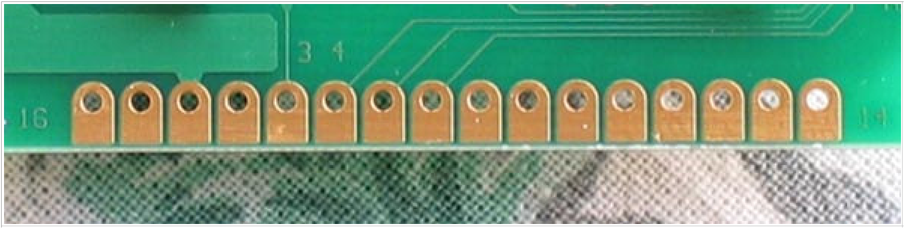
If your LCD has more than 80 characters (like 4x40)

15	E2	1, 1->0	Enable signal row 2 & 3
16			Not used mostly

HY1602B (Hyper 1602B) with KS0065 controller (compatible with HD44780) and backlight. Click for a bigger picture



1602-04 with KS0066U controller (compatible with HD44780) and backlight Click for a bigger picture



These are 2 different 2x16 LCDs as in the pictures the 15th and 16th pins are for backlight as I mentioned above.

Pinout Descriptions

Pin 1,2,3

According to the table, I call Pin 1(Vss),2(Vdd/Vcc),3(Vee/Vo) power pins because they the gates to power. Pin 1 is for ground so you have to connect to ground/earth and Pin for the +5V power supply. 6V or 4,5V is mostly acceptable in few amperes and also 3V i acceptable on some of the LCD modules (You can also power these modules with a batte in a very economical way). In my application I get the voltage from the molex cable of t pc which is inside the case. And pin 3 is for the LCD's contrast adjustment. I did not but you could use a potentiometer(10K pot will be ok) for changing the contrast of your LCD See the schematics below

Pin 4,5,6

I call Pin4(RS),5(R/W),6(E) the control buddies because these pins are the arms of your controller inside your LCD module. Pin 4(RS) is registration select if this pin is low the da perceived by the data pins taken as commands by the LCD and if this pin is high the LCD can receive/send 8 or 4 bit character data. I call Pin 5(R/W) clerk because when this pin low you can write character to the LCD, if the pin is high you can read character data or status information from the LCD. I didn't make any read operations in my app so I solde this pin to the ground (with soldering this to the ground I made this pin low - "0" see the below circuits). Pin 6(E) which I call the guardian, is used to initiate the actual transfer c commands or character data between the LCD module and the data pins.

Pin 7,8,9,10,11,12,13,14

The eight pins which are DB0-DB7 are the data pins which I call them the workers. The data can be transferred or fetched from the LCD by 8 or 4 bits. Which one is better? This up to you, by the way if you are using a microcontroller and you have few pins you can your module in 4 bit mode(by using DB4-DB7). I used 8 bit mode in my LCD because I used the parallel port which already have 8 bit data lines (remember my first article, pa D0-D7)

Pin 15,16

These two pins are for the backlight of the LCD module. 15th pin goes to the power supply(VB+) and 16th pin goes to the ground(VB-). Backlight is very useful in dim environments but some LCD modules don't have backlights. There are multicolored LCD around as well.

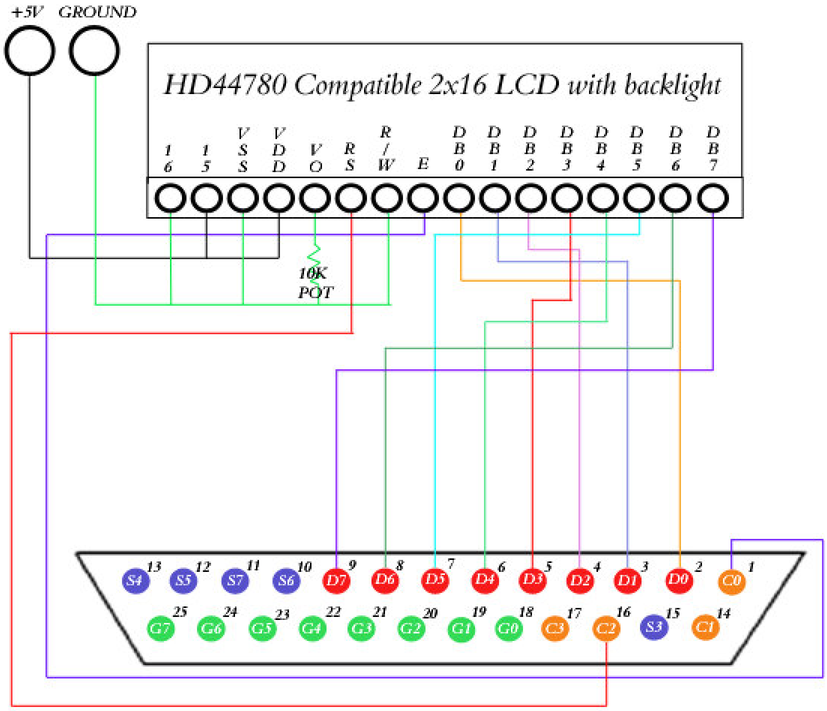
Circuit

What we need to supply for our circuit? Below is a list of that:

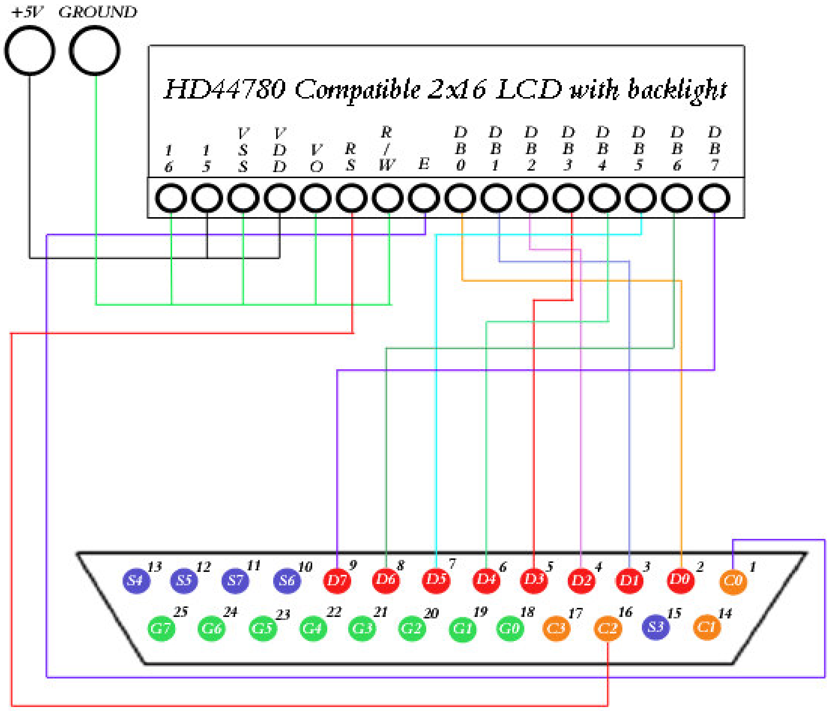
No:	Description
1.	2x16 Paralell LCD must be HD44780 compatiable
2.	Normal parallel printer cable
3.	Normal twin power cable
4.	10 way housing and the PCB header
5.	6 way housing and the PCB header
6.	Hard drive type power connector (Molex)
7.	16 PCB terminals
8.	A digital multimeter(must measure few ampers!), a solder pen with some soldering iron
9.	Some patience, and my program :)
*	10 K potentiometer (Not required - needed when you want to adjust the contrast of your LCD - see circuit with potentiometer)

Make the circuit as its shown in the pictures below.

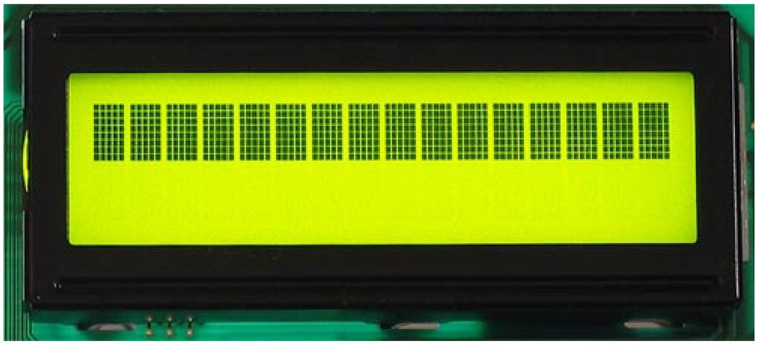
The circuit with potentiometer



The circuit without potentiometer



If your soldering goes well you get a typical test screen of a character based LCD as its shown below:



Part 3 - Before Coding

Before coding you have to know some of the basic instructions of a HD44780 controller. Below is a table from the Hitachi's old web site which explains the instructions for HD44 compatible LCD. Note: I add some extra info for easy understanding.

Instruction	Code										Description	Execut 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 display and returns cursor to the home position (address 0).	1.64m
Cursor home	0	0	0	0	0	0	0	0	1	*	Returns cursor to home position (address 0). Also returns display being shifted to the original position. DDRAM contents remains unchanged.	1.64m
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction (I/D), specifies to shift the display (S). These operations are performed during data read/write. I/D = 0 --> cursor is in decrement	40u

											position. I/D = 1 --> cursor is in increment position. S = 0 --> Shift is invisible. S = 1 --> Shift is visible	
Display On/Off control	0	0	0	0	0	0	1	D	C	B	Sets On/Off of all display (D), cursor On/Off (C) and blink of cursor position character (B). D = 0 --> Display off. D = 1 --> Displan on. C = 0 --> Cursor off. C = 1 --> Cursor on. B = 0 --> Cursor blink off. B = 1 --> Cursor blink on.	40uS
Cursor/display shift	0	0	0	0	0	1	S/C	R/L	*	*	Sets cursor-move or display-shift (S/C), shift direction (R/L). DDRAM contents remains unchanged. S/C = 0 --> Move cursor. S/C = 1 --> Shift display. R/L = 0 --> Shift left. R/L = 1 --> Shift right	40uS
Function set	0	0	0	0	1	DL	N	F	*	*	Sets interface data length (DL),	40uS

						number of display line (N) and character font(F). DL = 0 --> 4 bit interface. DL = 1 --> 8 bit interface. N = 0 --> 1/8 or 1/11 Duty (1 line). N = 1 --> 1/16 Duty (2 lines). F = 0 --> 5x7 dots. F = 1 --> 5x10 dots.	
Set CGRAM address	0	0	0	1	CGRAM address	Sets the CGRAM address. CGRAM data is sent and received after this setting.	40uS
Set DDRAM address	0	0	1	DDRAM address		Sets the DDRAM address. DDRAM data is sent and received after this setting.	40uS
Read busy-flag and address counter	0	1	BF	CGRAM / DDRAM address		Reads Busy-flag (BF) indicating internal operation is being performed and reads CGRAM or DDRAM address counter contents (depending on previous instruction). I used some delay functions in my code	0uS

				which are ThreadSleep if you don't want to use these you can check the Busy Flag and make your LCD speedy. BF = 0 --> Can accept instruction. BF = 1 --> Internal operation in progress no additional operation can be accepted.	
Write to CGRAM or DDRAM	1	0	write data	Writes data to CGRAM or DDRAM.	40uS
Read from CGRAM or DDRAM	1	1	read data	Reads data from CGRAM or DDRAM.	40uS

* = Not important, Can be "1" or "0"
** = Execution Time is a time needed which the LCD needs for the operation.

CGRAM is character generator RAM, this ram can hold user defined graphic characters. This capability gains these modules popularity because of this you can make bargraphs and your own language's special characters(Maybe Chinese, Korean, Turkish, Greek, etc

DDRAM is the Display Data RAM for these modules which represents the hexadecimal Display data adresses. See below:

Char.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Line 1	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
Line 2	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF

If you decide to develop your own program after this article, this instruction table will co in handy. If it comes a bit complicated see Part 4 then you can understand the logic beh the instructions with coding.

Part 4 - Coding Time

Now we know a lot of things about LCDs so we can write some codes to make our LCD li There are three important functions in my app which are `Prepare_LCD` | `move_to_speci` | `button_Write_to_screen_Click`

The below code(`Prepare_LCD`) Prepares the LCD after you saw the test screen. I made t comments as clear as possible so you can examine with the guiding of the instruction co above.

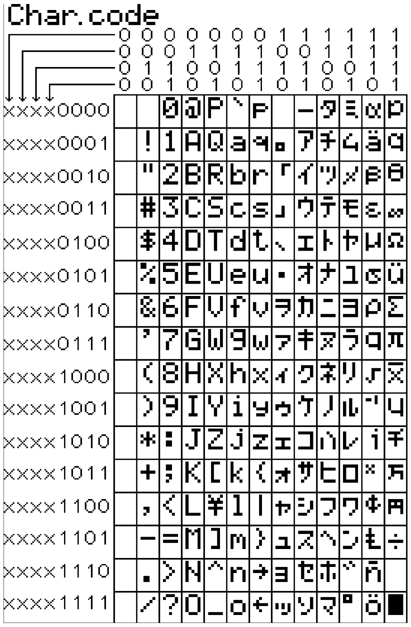
⊟ Collapse

```
private void Prepare_LCD(int cursor_status)
{
 /* Look at the instruction table to make these comments make sense */

 /* Thread.Sleep() function is not needed for some type of LCD instructions
  * and also this is changeable from an LCD
  * to another so tryout the best for your module */

  /* Sends 12(d) = 1100 binary to open the entire display and
   * makes a delay that LCD needs for execution */

  if(cursor_status == 0)
    PortAccess.Output(data, 12); //Thread.Sleep(1);
//The delays can be smaller, Check Busy Flag info in the article

  /* Sends 14(d) = 1110 binary to open the entire display
   * and makes the cursor active and also
   *  makes a delay that LCD needs for execution */

  if(cursor_status == 1)
    PortAccess.Output(data, 14); //Thread.Sleep(1);
//The delays can be smaller, Check Busy Flag info in the article

  /* Sends 15(d) = 1111 binary to open the entire display, makes
   * the cursor active and blink and also
   *  makes a delay that LCD needs for execution */

  if(cursor_status == 2)
    PortAccess.Output(data, 15); //Thread.Sleep(1);
//The delays can be smaller, Check Busy Flag info in the article

  /* Makes the enable pin high and register pin low */
  PortAccess.Output(control, 8); Thread.Sleep(1);
//The delays can be smaller, Check Busy Flag info in the article

  /* Makes the enable pin low for LCD to read its
   * data pins and also register pin low */
  PortAccess.Output(control, 9); Thread.Sleep(1);
//The delays can be smaller, Check Busy Flag info in the article

  /* Clears entire display and sets DDRAM address
   *  0 in address counter */
  PortAccess.Output(data, 1); //Thread.Sleep(1);
//The delays can be smaller, Check Busy Flag info in the article

  /* Makes the enable pin high and register pin low */
  PortAccess.Output(control, 8); Thread.Sleep(1);
//The delays can be smaller, Check Busy Flag info in the article

  /* Makes the enable pin low for LCD to read its
   * data pins and also register pin low */
  PortAccess.Output(control, 9); Thread.Sleep(1);
//The delays can be smaller, Check Busy Flag info in the article

  /* We are setting the interface data length to 8 bits
   * with selecting 2-line display and 5 x 7-dot character font.
   * Lets turn the display on so we have to send */
  PortAccess.Output(data, 56); //Thread.Sleep(1);
//The delays can be smaller, Check Busy Flag info in the article

  /* Makes the enable pin high and register pin low */
  PortAccess.Output(control, 8); Thread.Sleep(1);
//The delays can be smaller, Check Busy Flag info in the article

  /* Makes the enable pin low for LCD to read its
   * data pins and also register pin low */
  PortAccess.Output(control, 9); Thread.Sleep(1);
//The delays can be smaller, Check Busy Flag info in the article
}
```

`move_to_specific` moves the cursor anywhere on the LCD (anywhere from the 32 optic in a 2x16 LCD). See the code below:

```
private void move_to_specific(int line, int column)
{
  /* Makes the RS pin low */
  PortAccess.Output(control, 8); //Thread.Sleep(1);

  if(line == 1)
  {
    /* Sets RAM address so that the cursor is positioned
     * at a specific column of the 1st line. */
    PortAccess.Output(data, 127+column); //Thread.Sleep(1);
  }
  if(line == 2)
  {
    /* Sets RAM address so that the cursor
     * is positioned at a specific column of the 2nd line. */
    PortAccess.Output(data, 191+column); //Thread.Sleep(1);
  }
}
```

See the below table for the ascii code binary equivalent for understanding the button_Write_to_screen_Click function:



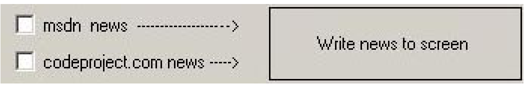
Char.code	0	0	0	0	0	0	0	1	1	1	1	1	1
	0	0	0	1	1	1	1	0	0	1	1	1	1
	0	1	1	0	0	1	1	1	1	0	0	1	1
	0	0	1	0	1	0	1	0	1	0	1	0	1
xxxx0000			0	@	P	`	p		ー	タ	ミ	α	p
xxxx0001		!	1	A	Q	a	q	。	ア	チ	ム	ä	q
xxxx0010		"	2	B	R	b	r	「	イ	ツ	メ	β	θ
xxxx0011		#	3	C	S	c	s	」	ウ	テ	モ	ε	∞
xxxx0100		$	4	D	T	d	t	、	エ	ト	ヤ	μ	Ω
xxxx0101		%	5	E	U	e	u	・	オ	ナ	ユ	σ	ü
xxxx0110		&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111		'	7	G	W	g	w	ァ	キ	ヌ	ラ	g	π
xxxx1000		(	8	H	X	h	x	ィ	ク	ネ	リ	√	x̄
xxxx1001		)	9	I	Y	i	y	ゥ	ケ	ノ	ル	⁻¹	y
xxxx1010		*	:	J	Z	j	z	ェ	コ	ハ	レ	j	千
xxxx1011		+	;	K	[	k	{	ォ	サ	ヒ	ロ	ˣ	万
xxxx1100		,	<	L	¥	l	|	ャ	シ	フ	ワ	¢	円
xxxx1101		-	=	M	]	m	}	ュ	ス	ヘ	ン	£	÷
xxxx1110		.	>	N	^	n	→	ョ	セ	ホ	゛	ñ	
xxxx1111		/	?	O	_	o	←	ッ	ソ	マ	゜	ö	█

Now you can look at the button_Write_to_screen_Click function in the code, I am no putting it here because I don't want to make the article hard to read but I commented o in the app so you can check it out.

Note: You can enable Thread.Sleep functions in the code if your LCD's execution time i very quick

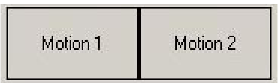
Part 5 - Before Final Note

Some people say "what will I benefit from an extra LCD?" For those, I made a tiny thing which grabs codeproject or the msdn rss headlines from the net and displays them on th LCD screen. Now you can check out the news while you are playing or watching full scre TV on your computer.


☐ msdn news -------------------->
☐ codeproject.com news ---->
Write news to screen

- Here you can find the video of that.

As you can see in the video the first line is fixed and the second line scrolls with the new after a period of time the second news scrols and the third, fourth and goes on. If you h online site which has some member systems or forums on it, you could send informatior using a php or an asp file to a xml or txt file and then fetch the info from there and senc the screen of your LCD or maybe even your area's forecast. You can make some periodi e-mail checks and display them on your LCD screen. You can also interface with your winamp for some spectrum visualization like a plugin but this will be outer world plugin f your winamp :) so see the motion1 and motion2 buttons for info how to develop these t of solutions:



You can make lots of things with an extra LCD on your desktop and these type of things bounded by your imagination so I can say here "*Enjoy your imagination*" :)

Final Note

Finally I come to the end of my second article in codeproject. I hope I was able to put a smile on the faces of those waiting for my second article. After my first article I get 35-4 e-mails from various people who are interested in these type of things. People send thei works also their circuits about various projects, for ex: some people sent their year(s) lasting problems on similar projects. So I want to thank you all for e-mails and also the threads in codeproject. I will be expecting your new suggestions and comments on this article. Don't abstain to mail at ls@izdir.com if you have problems.

What will be the third?

I don't want to talk about my next article so I didn't put a picture of it like I did in the er of my first article because it is after all a mystery.

Levent Saltuklaroglu

Click here to view Levent Saltuklaroglu's online profile.

Other popular C# Programming articles:

- A flexible charting library for .NET
  Looking for a way to draw 2D line graphs with C#? Here's yet another charting class library with a high degree of configurability, that is also easy to use.
- I/O Ports Uncensored - 1 - Controlling LEDs (Light Emiting Diodes) with Parallel Port
  Controlling LEDs (Light Emiting Diodes) with Parallel Port
- 3D Pie Chart
  A class library for drawing 3D pie charts.
- Events and Delegates simplified
  This article shows you how to design events for your classes.

[Top] **Sign in** to vote for this article: *Poor* ○ ○ ○ ○ ○ *Excellent* Vo

Dundas
Data Visualization
Dundas Chart for .NET
• Visual Studio 2005 features
• Full AJAX support
• Advanced interactivity for ASP.NET

Note: You must **Sign in** to post to this message board.

FAQ Message score threshold 3.0 Search comments **Set Optic**

View Message View Per page 25

Msgs 1 to 25 of 245 (Total: 245) (Refresh) First Prev Next

Subject	Author	
Strange LCD	**Fmcatalan**	**12:58 7 Sep**
Windows XP - doesn't work	**zopostyle**	**21:38 24 Aug**
VB	**VB King!**	**9:28 24 Aug**
selamlar	**harunbelet@mynet.com**	**16:39 23 Aug**
Merhaba	**harunbelet@mynet.com**	**16:39 23 Aug**
Programs shows nothing when started	**alek66**	**17:14 4 Jul**
Wiring Ok, programs not	**alek66**	**13:00 3 Jul**
Re: Wiring Ok, programs not	alek66	13:36 3 Ju
Borland C++ Builder source for LCD	**GuLrs**	**13:38 19 Jun**
Re: Borland C++ Builder source for LCD	ubivetz	7:52 1 Sep
inpout32.dll	**TOFRAGUZBEL**	**8:22 22 May**
What about PCI?	**Valery Vitko**	**5:24 25 Apr**
Levent Bey merhaba	**NightBLuex**	**8:37 12 Apr**
slm levent s.	**EfsanE_1925**	**17:15 23 Mar**
merhaba	**EfsanE_1925**	**20:24 22 Mar**
Connect bar code component with computer?	**Duong Nhu Y**	**2:28 15 Mar**
Problems with similar LCD type	**BlindSummit**	**16:46 5 Jan**
Re: Problems with similar LCD type	Roy_w	15:56 12 Jar
What about VC++ 6.0	**DeanOakes**	**10:21 25 Dec**
Great Article (LCD Controller KS0065)	**RichTeel**	**1:45 18 Dec**
Re: Great Article (LCD Controller KS0065)	BlindSummit	9:23 7 Jar
blocking read from parallel port	**rspr**	**12:44 14 Dec**
display chronometer	**el_aristo**	**8:52 20 Nov**
USB	**chuahh**	**10:05 9 Nov**
Re: USB	jkersch	8:16 4 Aug

Last Visit: 10:17 Sunday 10th September, 2006 First Prev Next

General comment News / Info Question Answer Joke / Game Admin message

Updated: 29 Sep 2003