

## EXTREME ELECTRONICS

Best free Microcontroller Tutorials and Projects

**Free Shipping !**



Line Following Robot Kit USB AVR Programmer v2.1

xBoard v2.0 GLCD Development Board

[HOME](#)   [STORE](#)   [FORUM](#)   [PRIVACY POLICY](#)

Search Here...

### CATEGORIES

- [32bit ARM Projects](#)
- [AVR Development Board](#)
- [AVR Projects](#)
- [AVR Tutorials](#)
- [Chitchat](#)
- [Code Libraries](#)
- [Code Snippets](#)
- [Electronics](#)
- [GSM Projects](#)
- [Hardware](#)
- [Microchip PIC Tutorials](#)
- [News](#)
- [PIC Development Board](#)
- [PIC16F877A Tutorials](#)
- [Programming in 'C'](#)
- [RF](#)
- [Robotics](#)
- [Software](#)
- [Tools](#)
- [Uncategorized](#)

### NAVIGATION

- [Home](#)
- [Forum](#)
- [Shop](#)
- [Links](#)

### SUBSCRIBE

Get New Articles Delivered To  
Your Inbox!

Email address:

[Subscribe](#)

Delivered by [FeedBurner](#)

1891 readers  
BY FEEDBURNER

### LATEST DISCUSSIONS

- frequency measurement of ac signal
- Program atmega8,168,328 with Arduino Uno board
- LCD custom character problem
- Icd connection to pc

YOU ARE HERE : [EXTREME ELECTRONICS](#) » [AVR PROJECTS](#) » [REMOTE TEMPERATURE MONITORING USING GSM - AVR PROJECT](#)

## LCD Library for PIC – Setup on MPLAB X IDE

Posted By Avinash On June 24th, 2013 09:12 PM. Under [PIC16F877A Tutorials](#)



This article describes the setup and use of the C library for hd44780 based alphanumeric lcd modules. This library is also available for the [AVR family of microcontrollers](#). This part is focused on its usage with [PIC16F](#) series of MCUs from Microchip. Here we describe how to setup a [MPLAB X](#) project with support for lcd related functions. The library is designed for compilation and use with Microchip's [XC8](#) C Compiler.



Fig. LCD Demo

## Creating a New Project in MPLAB X



### RELATED POSTS



#### [PIC16F877A Serial Communication Demo](#)

BY AVINASH - AUGUST 1 2013 06:46 PM

In the last two tutorial we discussed how to make ...



#### [USART Library for PIC – Setup on MPLAB X IDE](#)

BY AVINASH - JULY 27 2013 06:34 PM

This article describes the setup and use of the C ...



#### [Serial Communication with PIC16F877A](#)

BY AVINASH - JULY 24 2013 06:50 PM

This article series aims at teaching serial communication between a ...



#### [Making A Thermometer with PIC16F877A](#)

BY AVINASH - JULY 13 2013 06:06 PM

At basic level on microcontroller you can create this interesting ...



#### [LCD Library for PIC – Setup on MPLAB X IDE](#)

BY AVINASH - JUNE 24 2013 09:12 PM

This article describes the setup and use of the C ...

### HOT TOPICS

[GUI Software for USBasp based USB AVR Programmers.](#)

427 comments received

[Using LCD Module with AVR's](#)

333 comments received

[Welcome](#)

269 comments received

[Programming in C - Tips for Embedded Development.](#)

156 comments received

[RF Communication Between Microcontrollers – Part III](#)

146 comments received

### RECENT COMMENTS

ISP issue with 40 pin AVR dev board  
keypad matrix problem  
External I2C EEPROM  
Interfacing with PIC18F4550  
eXtreme Burner 1.4.2 does not read ATmega2560 flash  
CAN ANY ONE HELP ME PLEASE !!  
Multichannel Remote & Fan Regulator Programme not working.

## LATEST ON AVR TUTORIALS



## Ultrasonic

## Rangefinder HC-SR04 Interfacing with ATmega8

BY AVINASH - AUGUST 12 2013 12:15 PM

In this article I will describe how to interface a ...



## Getting Started with AVR Microcontrollers

BY AVINASH - MAY 24 2013 11:10 AM

The first step in getting started with microcontrollers is to ...



## Introduction to AVR Microcontrollers

BY AVINASH - OCTOBER 31 2012 03:55 PM

Just what is a microcontroller? To ...

## LATEST ON MICROCHIP PIC TUTORIALS



## Door Entry Detection for MCU Based Designs

BY AVINASH - OCTOBER 12 2012 12:43 PM

This article discuss how you can detect the entry of ...



## Thermometer with PIC Microcontroller

BY AVINASH - JUNE 10 2011 07:26 PM

In the last tutorial we learn about the multiplexing technique ...



## Using Multiplexed 7 Segment Displays – PIC Microcontroller Tutorial

BY AVINASH - MAY 31 2011 12:11 PM

You can create a new project using the MPLAB's Start page as shown below.

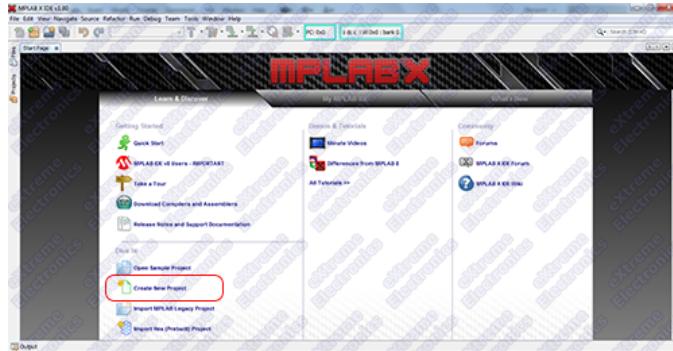


Fig. MPLAB X Start Page

Alternatively you can use *File->New Project*

Fig. Select New Project from File Menu

And for those who love the Keyboard over mouse can hit &lt;Ctrl&gt;+&lt;Shift&gt;+&lt;N&gt;

Any of the three method will launch the New Project Wizard as shown below.

The first step is the selection of project type.

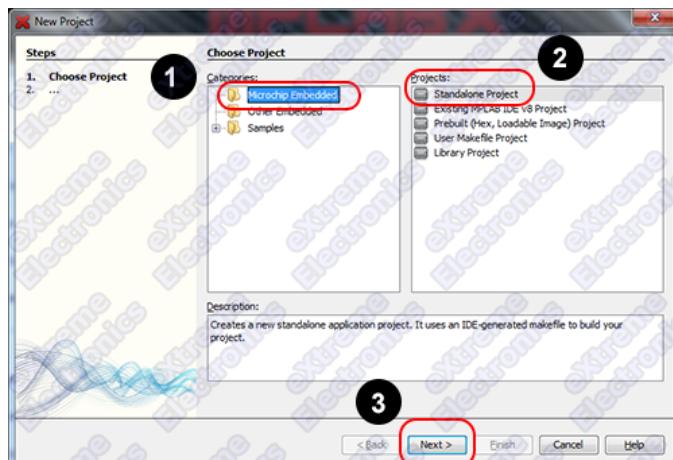
From the **Categories** list select **Microchip Embedded** and from **Projects** select **Standalone Project**.

Fig. Project Type Selection

Second step is the selection of device for which the project is targeted. Select **Mid Range 8-bit MCUs (PIC12/16/MCP)** in **Family** and **PIC16F877A** in **Device**.

TAHER Says:

Hi. It is such a nice project. i request you to ma...



Imran Says:

Dear Avinash, Your effort helping others. Pl...



Hem Says:

Sir, I have found my solution. I make 2 changes in ...



Sundeep Subbaraya Says:

Hi Avinash, I received the board. After playing...



Manjunath K S Says:

Can you plz tell me whats wrong in my code to read...

## FACEBOOK LIKE BOX

Find us on Facebook



ExtremeElectronics

Like

21,240 people like ExtremeElectronics.



Facebook social plugin

## LATEST ON AVR PROJECTS



### Microwave Controller using ATmega8 – AVR Project

BY AVINASH - JANUARY 1 2014  
08:20 PM  
Microwave Controller's User Interface ...



### Remote

### Temperature Monitoring using GSM – AVR Project

BY AVINASH - APRIL 7 2013 01:14 PM  
Temperature monitoring have wide application in daily life. In modern ...

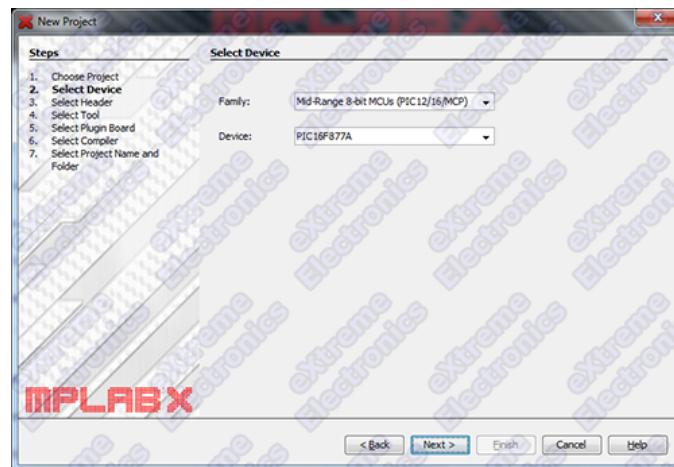


Fig. Device Selection

Third step is the selection of debug tool. For that select **Simulator**.

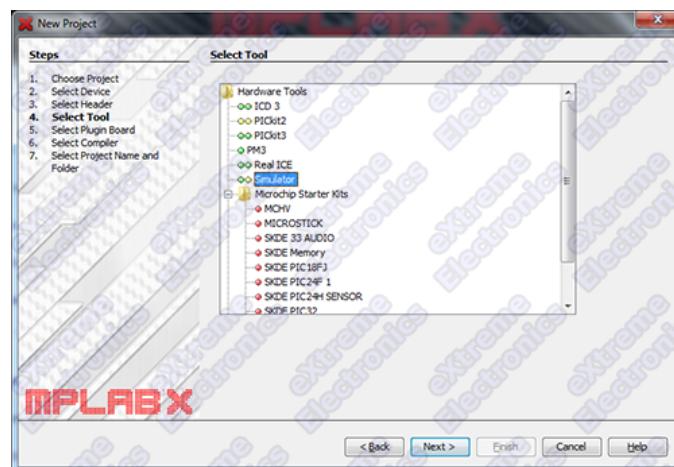


Fig. Tool Selection

Microchip MPLAB lets you install more than one compiler. It also lets you install more than one version of the same compiler. So there is a step to select a compiler for use with your project.

In the image below you can see the compiler XC8 in the list of available compilers, under it is listed all different versions of the XC8 versions available. Since I had only version 1.12, it is the only available option under XC8.

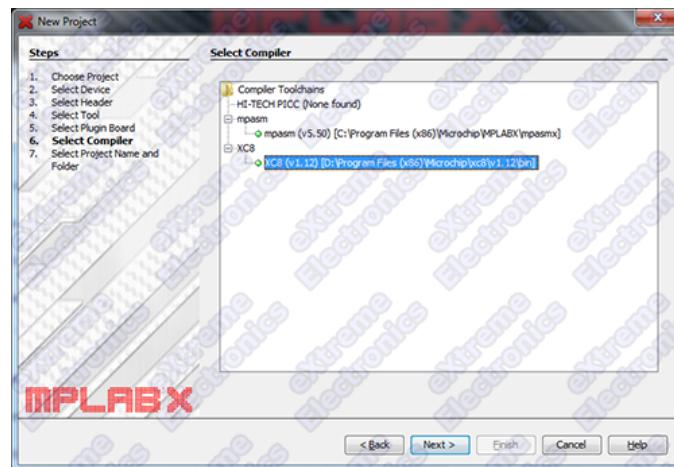
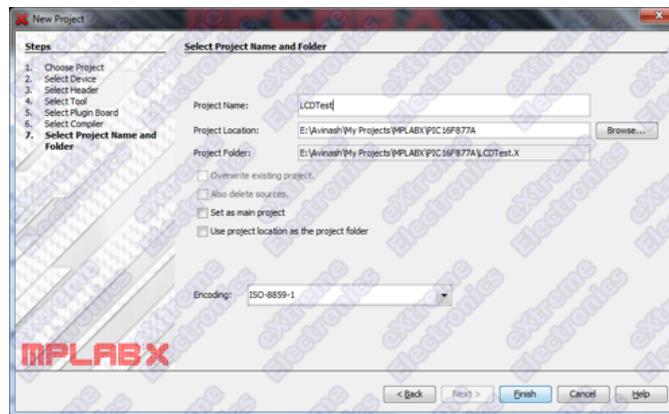


Fig. Compiler Selection

### Project name and location

Finally enter the name of project and choose a location on your computer where you wish to save the project and all related files.



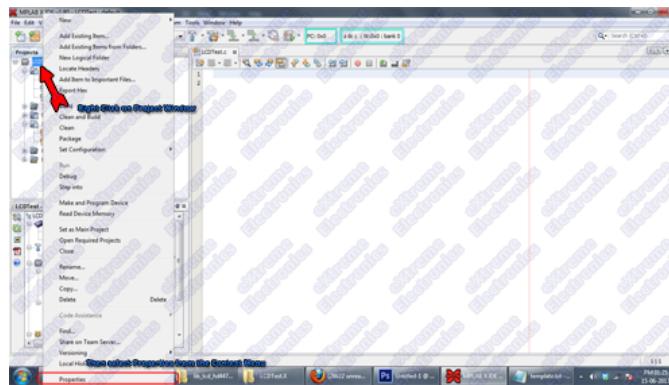
**Fig. Enter Project Name and Location**

Click **Finish**.

## Configuring the new MPLAB Project

A small configuration that is needed on all MPLAB project is the definition of a global C preprocessor symbol (consider it as a project level global constant). This symbol is named **\_XTAL\_FREQ** and it holds a value equal to the frequency at which the PIC MCU will be running. **Note the definition of this symbol is very important ! other wise the codes will not be able to generate accurate timing and the code will fail at runtime!**

On MPLAB Window, right click over the project's name in the **Project** window.



**Fig. Project Context Menu**

This will bring up a context menu as shown above. From the menu select **Properties**. It is the last item in the long menu. The **Project Properties** window will open up as shown below.

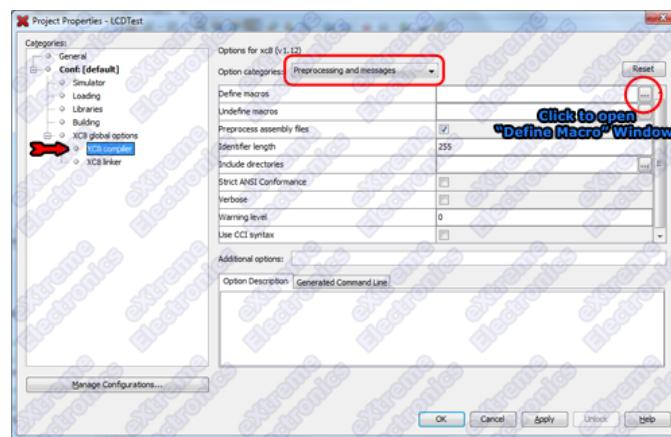


Fig. Project Properties.

In the left hand pane labeled **Categories** select **XC8 Compiler**. It will open up options for the compiler which is displayed on the right hand side of the window. In **Option Categories** select **Preprocessing and messages**.

And then click on the button as shown in the image above to open up the **Define Macro** window.

In this window add a macro name **\_XTAL\_FREQ=20000000**(two followed by seven zeros, i.e. 20 MHz).

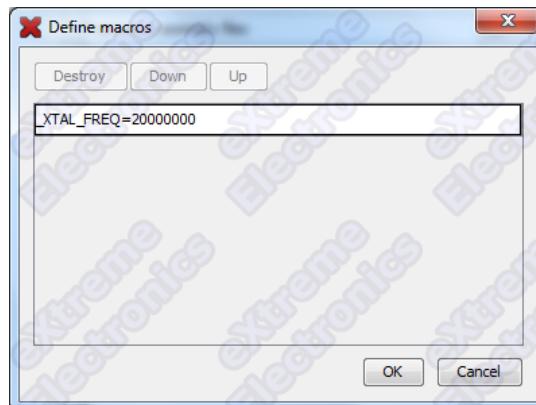


Fig. Define Macros.

## Adding LCD Library Support Files

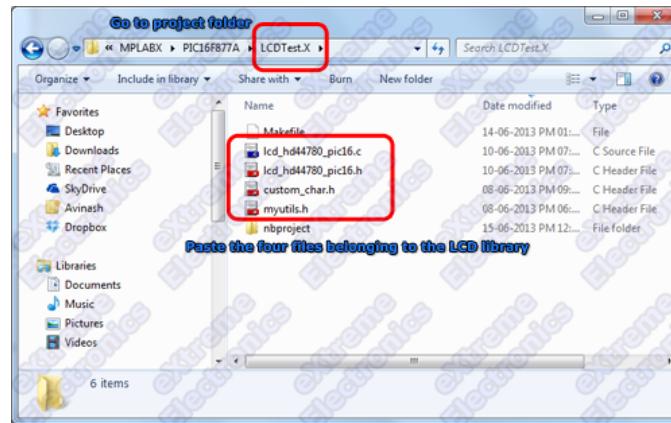
The library to drive hd44780 based alphanumeric LCD modules comes as a set of following files.

- lcd\_hd44780\_pic16.c
- lcd\_hd44780\_pic16.h
- custom\_char.h
- myutils.h

You can download the package from the link given below.

- [Download hd44780 based lcd library for PIC16](#)

Unzip all the files from the package and then copy them to your MPLAB X (**LCDTest.X** in this case) project folder using Windows file manager.

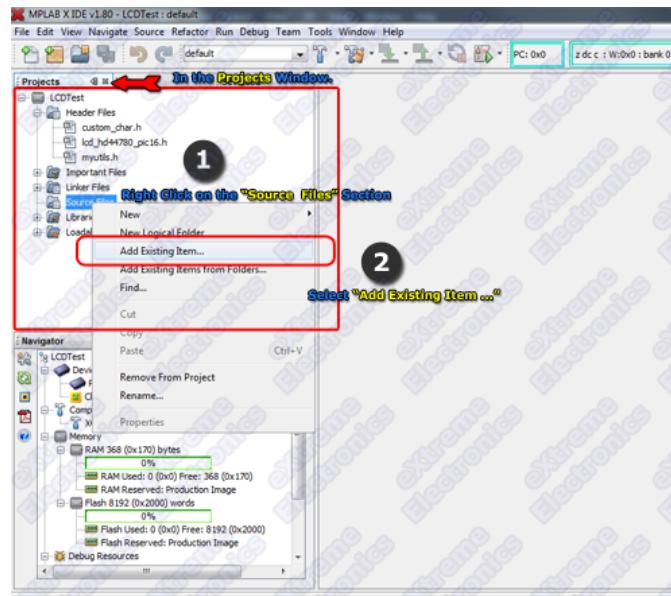


**Fig. Paste the LCD Lib files to project folder.**

Now when the files have been copied to the project folder. Its time to add the files to the project.

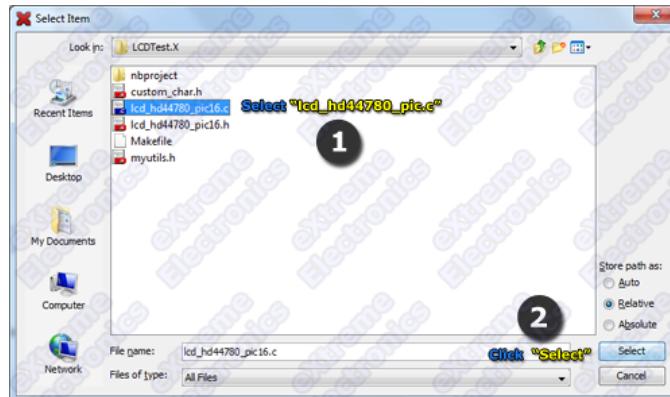
#### Adding the source files.

Please refer to the image given below to add existing source files in MPLAB X.



**Fig. Adding existing source files.**

It will open up the standard file select dialog which lets you select files to add to the project. Select **Lcd\_hd44780\_pic16.c** and it will get added to the project.

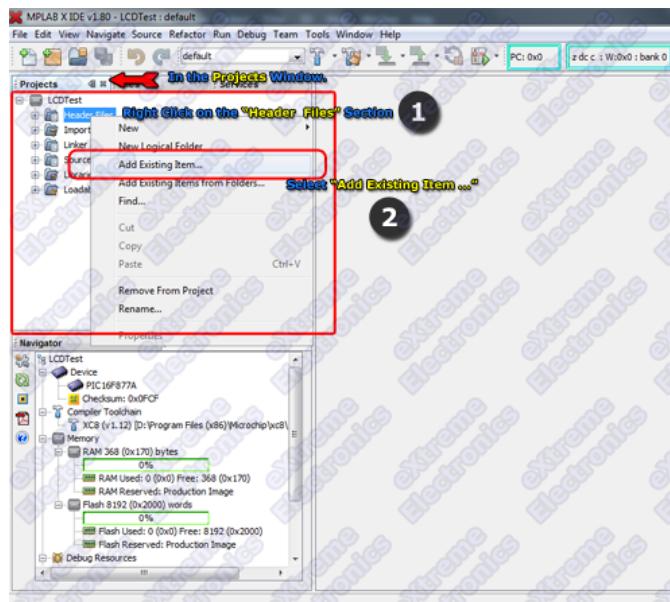
**Fig. Select lcd\_hd44780\_pic16.c**

#### Adding the header files.

As you must have seen that the lcd library has three header files.

- lcd\_hd44780\_pic16.h
- custom\_char.h
- myutils.h

All the three header files must be added to the header files section of the project. The steps are same as described above for source files.

**Fig. Adding existing header files**

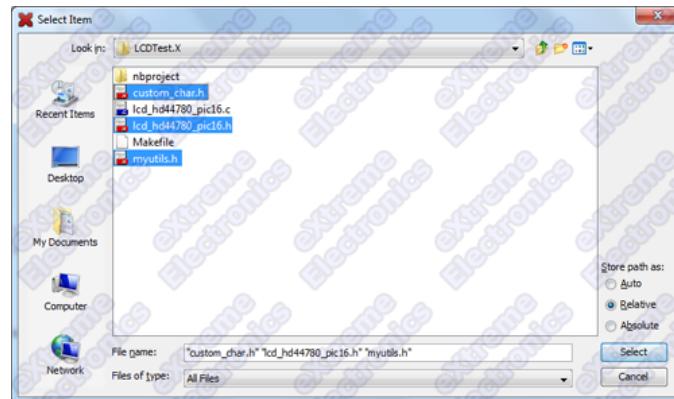


Fig. Selecting LCD header files.

Now the project is ready and has support of using lcd related functions.

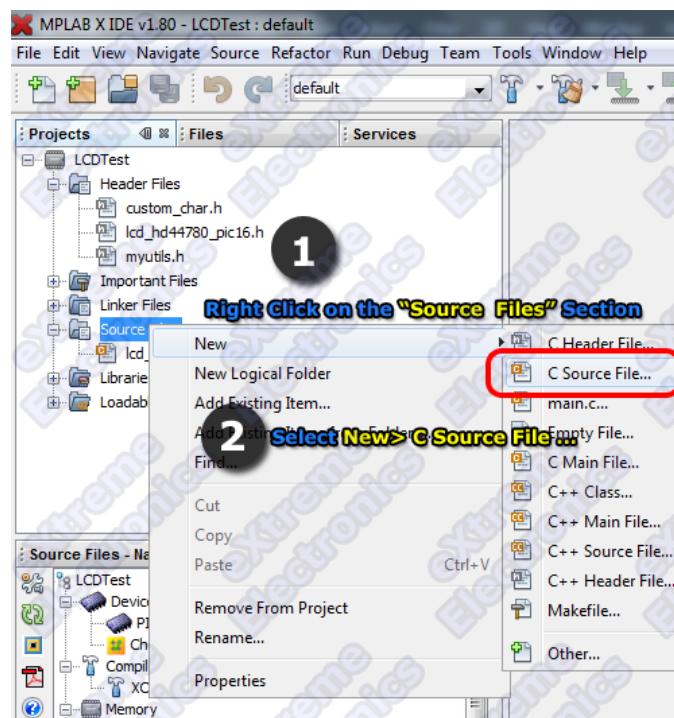
## Writing a "Hello World" Application

We will write a small program that will initialize the LCD module and write a string Hello World on it.

### Adding a new source file to project

You will need to add a new source file to the project, this file will hold our main application code. For this

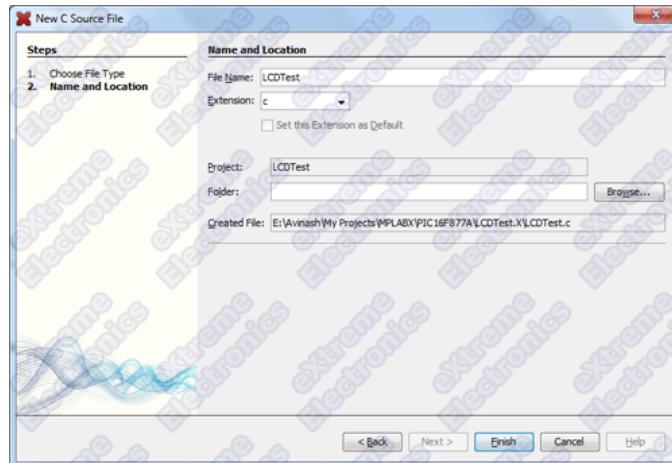
1. Right click on the **Source Files** Section of Project in the **Projects** window.
2. Select **New > C Source File ...**



**Fig. New Source File.**

It will open up the New C Source File dialog.

In this dialog enter the name for this new file. Enter the name "**LCDTest**" and copy paste the program given below to this file.

**Fig. Naming the new file.**

## Sample Program

```
*****
The most basic example program to write a line of text in a
alphanumeric lcd module using our lcd library for pic16 mcu

Compiler: Microchip XC8 v1.12 (http://www.microchip.com/xc)
IDE: Microchip MPLABX

MCU: PIC16F877A
Frequency: 20MHz

NOTICE

NO PART OF THIS WORK CAN BE COPIED, DISTRIBUTED OR PUBLISHED
WRITTEN PERMISSION FROM EXTREME ELECTRONICS INDIA. THE LIBRA
OF IT CAN BE USED IN COMMERCIAL APPLICATIONS. IT IS INTENDED
HOBBY, LEARNING AND EDUCATIONAL PURPOSE ONLY. IF YOU WANT TO
COMMERCIAL APPLICATION PLEASE WRITE TO THE AUTHOR.

WRITTEN BY:
AVINASH GUPTA
me@avinashgupta.com
```

```
*****
#include <xc.h>

#include "lcd_hd44780_pic16.h"

// CONFIG
#pragma config FOSC = HS           // Oscillator Selection bits
#pragma config WDTE = OFF          // Watchdog Timer Enable bit
#pragma config PWRTE = OFF          // Power-up Timer Enable bit
#pragma config BOREN = ON           // Brown-out Reset Enable bit
#pragma config LVP = ON             // Low-Voltage (Single-Supply) I/O
#pragma config CPD = OFF            // Data EEPROM Memory Code Protection
#pragma config WRT = OFF            // Flash Program Memory Write Protection
#pragma config CP = OFF             // Flash Program Memory Code Protection

void main (void)
{
    //Initialize the LCD Module
    LCDInit(LS_NONE);

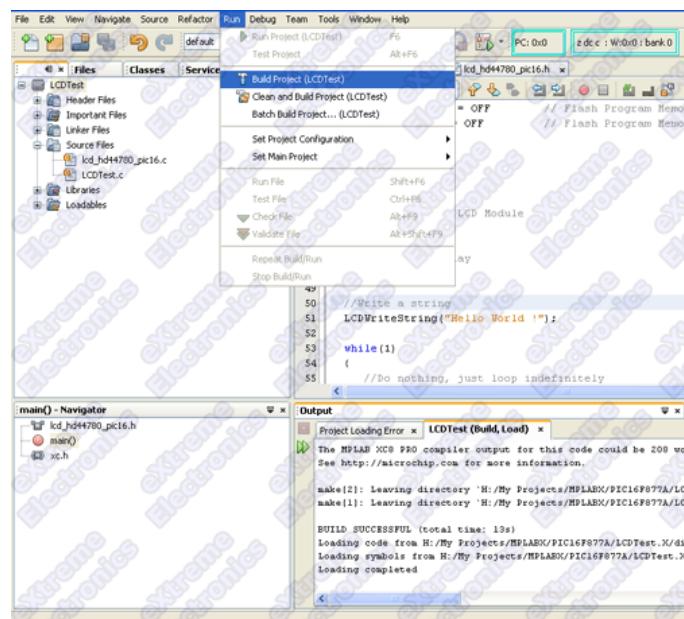
    //Clear the display
    LCDClear();

    //Write a string
    LCDWriteString("Hello World !");

    while(1)
    {
        //Do nothing, just loop indefinitely
    }
}
```

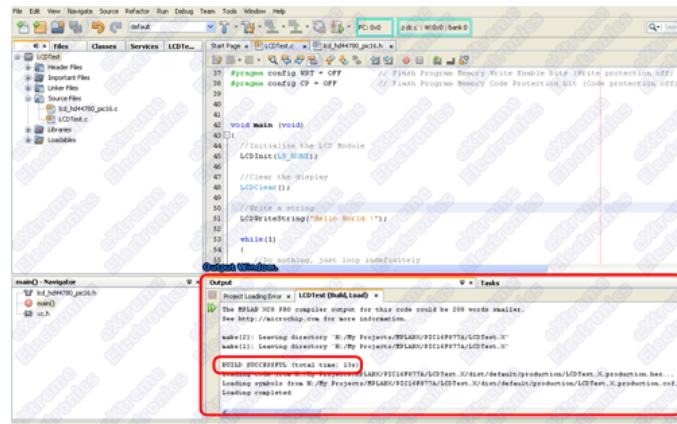
Now its time to build this project. Building is the process of compiling all files in the project and then linking them to build the final executable image. This executable image is in the form of a hex file. The name of this file is **<project name>.X.production.hex**

To build the project select **Build Project** from the **Run** Menu.



**Fig. Building the project**

If you have done everything precisely as described in this tutorial your project should build without any problem and you should get a **BUILD SUCCESSFUL** message in the output window of the MPLAB X IDE.



## Fig. Build Success

The hex file which is actually the executable program in machine language, is generated in the folder "`\dist\default\production`" which is in your project folder.

The name of the file should be ***LCDTest.X.production.hex***

## Burning the hex file to MCU

Burning is the process of transferring the data contained in this hex file to appropriate memory locations on the chip so that chip can execute this program. To burn a chip you need a device programmer. A device programmer consists of a hardware and a driving software. You can purchase a device programmer for PIC16F MCUs from our online store. Please note that we ship through out India and to many other countries.

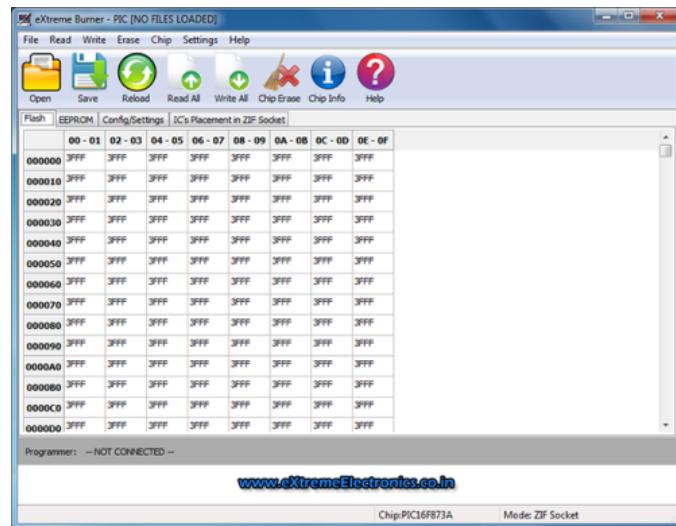
International customers can pay safety using **Paypal** while Indian customer can use **Credit Card, Debit Card or Online Banking** for purchase.

- Purchase USB Port based device programmer for PIC16F MCUs.

How to burn the hex file to PIC MCU using this programmer is described in this article.

- eXtreme Burner – PIC 16 user guide

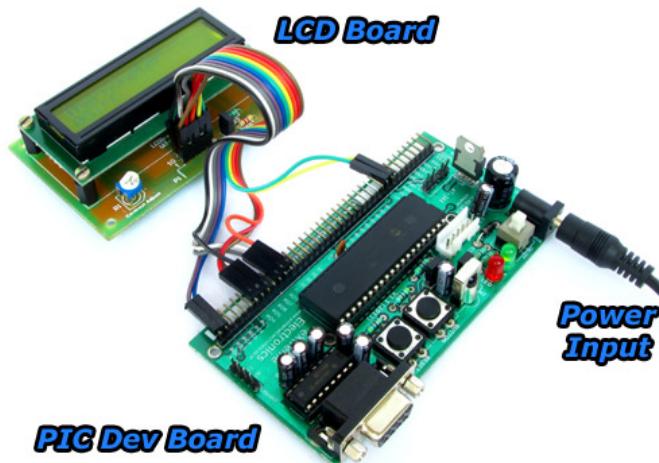


**Fig. USB Port Based PIC Programmer****Fig. Software Interface**

## Hardware

Hardware required to run this demo is described separately in this article.

- [LCD Module Interface with PIC16F877A](#)

**LCD Board Connection**

**Fig. LCD Demo**

## Downloads

- Complete MPLAB X Project for Basic LCD Demo.
- HEX File Ready to burn for Basic LCD Demo.
- HD44780 based alphanumeric lcd driver library.

By Avinash Gupta  
[www.avinashgupta.com](http://www.avinashgupta.com)

Tags: [Hd44780](#), [Lcd](#), [MPLABX](#), [Pic](#), [Pic16f877a](#), [XC8](#)

### 8 Responses To “LCD Library For PIC – Setup On MPLAB X IDE”



**Shivdas Says:**

June 25, 2013 at 8:47 pm

Thanks for lcd display module information. It is required for me.

[REPLY](#)



**Pcb\_nl Says:**

November 13, 2013 at 3:39 pm

Thanks!

This was very helpful.

Could you tell me what custom\_char.h is for?

If i use LCDWriteInt() the compiler says there is not enough memory. “could

not find space (64 bytes) for variable \_\_cgram”

i deleted a few lines of the content of \_\_cgram[] and then it compiled.

btw

using pic16f1823 and not so many program space

Program space used 2E7h (743) of 800h words (36.3%)

Data space used 3Eh (62) of 80h bytes (48.4%)

EEPROM space used 0h (0) of 100h bytes (0.0%)

Configuration bits used 2h (2) of 2h words (100.0%)

ID Location space used 0h (0) of 4h bytes (0.0%)

[REPLY](#)



**Avinash Says:**  
November 13, 2013 at 5:29 pm

@pcb\_nl,

**\_\_cgram[] is used for storing custom characters face.**

REPLY



**R Says:**  
February 13, 2014 at 2:45 am

Your libraries hint that they can only be used for 16F, but XC8 is for all 8-bit MCU's, 10F-12F-16F-18F. Would this library work with the 18F series as well?

I hope to use it for 18F4550/4553.

REPLY



**Nitin Joshi Says:**  
April 23, 2014 at 5:31 am

Sir,

Thanks for the reply.

I want to know is there any HD44780 LCD library files available for PIC33. as i am working on dsPIC33FJ16GS504 which is Microchip IC

REPLY



**Gustavo Says:**  
June 10, 2014 at 7:50 am

Hello, I followed your tutorial and work fine, but I need to move the LCD's connections to PORTB and use a 4x16 LCD.

I changed on lcd\_hd44780\_pic16.h all references to PORTB. Then I builded the project without errors but the simulation still working on PORTD. To check the build I commented:

```
//LCD Data Port
//Port PD0-PD3 are connected to D4-D7
// #define LCD_DATA_B //Port PD0-PD3 are connected to D4-D7
// #define LCD_DATA_POS 0
```

After that I builded the project again and surprise the project builded without errors. I think the defines of the ports are in other place, because when I commented those lines the compiler should give some errors like symbol undefined.

Can you help me please?

Sorry my English, I'm Spanish users from Argentina.

REPLY



**Ray Says:**  
June 12, 2014 at 3:38 am

Greetings first.

Was forced to move away from "Pic BasicPro" and to use a C type compiler. What a great step by step explanation and great introduction to the "X" environment.

Thank kindly for the great information.

REPLY



**Povorai Gabor Says:**  
July 23, 2014 at 9:17 pm

Hi! I'm having trouble of setting up the external oscillator of a pic18f4520 to 20MHz crystal, I know the #pragma define OSC=HS must be applied but it's not working (for the same LCD display) I was able to write some text with the internal oscillator though, but with bad display quality.

REPLY

## Trackbacks/Pingbacks

[LCD Module Interface With PIC16F877A |eXtreme Electronics](#)

[Making A Thermometer With PIC16F877A |eXtreme Electronics](#)

## Leave a Reply

 Name (Required) E-Mail (Required) Website (Optional)

[Submit Comment](#)

[Notify Me Of Followup Comments Via E-Mail](#)

[« LCD Module Interface with PIC16F877A](#)

[Making A Thermometer with PIC16F877A »](#)

Copyright © 2014. All Rights Reserved.