HMI Peripheral board with LadderWORK

OBJECTIVE: MAKING YOUR HMI BOARD WORK WITH LADDERWORK®

ASSUMPTION:

- ✓ *Board with LCD and minimum 8K ram is ready*
- ✓ *Database File editor*
- ✓ *USASM51 and USLIB for assembling and creating library respectively*

PREPARATION:

- ✓ Note down the connections for LCD

WRITTEN BY: AJAY BHARGAV

DATED: *FEB 10TH, 2007*

REVISION CODE: 1.0

This tutorial will provide you the basic information on How to make your HMI peripherals (e.g. LCD, LED Display, 7-Seg LED etc) to work with your LadderWORK® software.

Note1: In this tutorial we are using single character based LCD Display, so is not referred with a terminal ID.

Note2: DBF editor you are going to use must support DBF III plus Files.

Step1: Understand Hook functions

LadderWORK provide certain hook functions which are to be bind with user defined functions to make user defined functions work with LadderWORK. As for now we are using LCD display, so the hooks used for the LCD purpose are as follows:

- __update
- __locate
- __put_asciiz_code
- __put_asciiz_data
- __putchar

The information regarding the above hook functions and all other hook functions can be found in ***LadderWORK help*** under the following section

Advanced Technical Information » Generic 8051 Board Adapting » Panel & Keyboard handling functions (HMI & MMI)

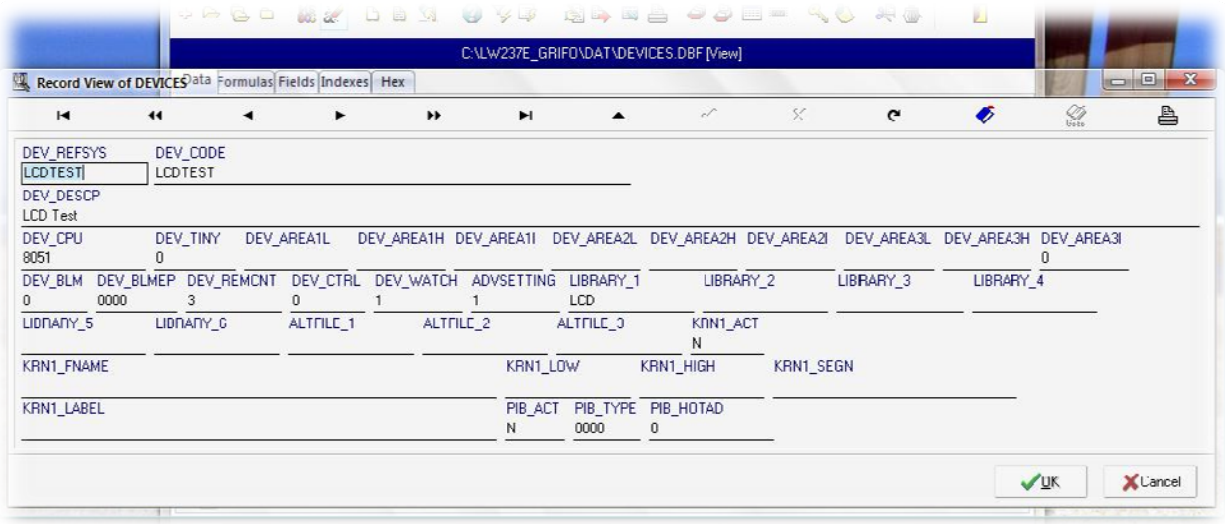
Now you have the information regarding the function which are to bind with the above mentioned hook function lets move to step #2.

Note3: We will define a new LCDtest board for better explanation

Step 2: Adding your test Board in LadderWORK Device list

To add your LCD test board we will make use of DBF editor and edit the “devices.dbf” database file in DAT folder.

We are not considering any other port or peripheral except the LCD.



As shown in the figure above, LIBRARY_1 is defined as LCD that means, we will define LCD library which will be going to used in this tutorial and will be explained in the further steps.

Now Assuming that you know further steps regarding how to add your board which is already explained in the previous tutorial (*Making your own PLC work with LadderWORK*) we will skip to the library creation step, Providing you the help regarding the HDW file.

As we are not taking care of any other peripheral or port, so HDW file will be having a simple SOC and EOC routines as shown.

```
##-------------------------------------------------------------------------
## Module      : LCDTEST.hdw
## Subject     : LadderWORK® Evaluation Board
## Update      : 10.2.2007
## Company     : MicroSHADOW Research (uS)
## Author      : Ajay Bhargav
##
##
##-------------------------------------------------------------------------




        ##
        ## RESOURCES
        ##
resources:
endresources

        ##
        ## FUNCTIONS
        ##
functions


##-------------------------------------------------------------------------
## LWMAIN
##-------------------------------------------------------------------------
function:[LWMAIN/LCDTEST,Class](void)
        INCLUDE "KERNEL.INC"
        INCLUDE "SFR8051.INC"
        cseg
endfunction

##-------------------------------------------------------------------------
## OPENCYCLE
##-------------------------------------------------------------------------
function:[SOC/LCDTEST,Class](void)
        cseg
        ##-----------------------------------------------------------------------
endfunction
```

```
##-------------------------------------------------------------------------
## CLOSECYCLE
##-------------------------------------------------------------------------
function:[EOC/LCDTEST,Class](void)
        cseg
endfunction


## POSTINIT ( Module PostInit )
function:[POSTINIT/LCDTEST,Class](empty)
endfunction;


## SHUTDOWN ( Module ShutDown )
function:[SHUTDOWN/LCDTEST,Class](empty)
endfunction;

end;

##-----------------
## LCDTEST.hdw
##-----------------
```

Step 3: Writing Library for your LCD

As already said, we need to write routines to run our LCD. The basic functions or routines needed are as follows

- LCD Initialization
- LCD Display Character
- LCD Command
- LCD Busy

LCD INITIALIZATION: This routine will initialize the LCD display sending few set of commands to the LCD. We are assuming here in tutorial that the LCD used is a 2 lines and 16 character based LCD, with 5x8 dots display.

LCD DISPLAY CHARACTER: This routine will display a single character on the LCD panel.

LCD COMMAND: This routine will send commands to the LCD.

LCD BUSY: This routine is used to check if the LCD is ready to accept next command or data.

After defining these basic functions we will hook them to the LadderWORK APIs, hence our library will now work along with the LadderWORK software.

Sample code used for the LCD Library is given below.

```
; Starting of code
include "sfr8051.inc"
include "kernel.inc"
include "hmimmi.inc"
;-------------------------------------------------------------------------
; Publics
;-------------------------------------------------------------------------
public __scrclr
public __terminal_init

;-------------------------------------------------------------------------
; Hooks
;-------------------------------------------------------------------------
public __putchar
public __put_asciiz_code
public __put_asciiz_data
public __update
public __locate
```

```
; Display mapping
;
; P1.0-P1.7 .. Data
; P3.5 ..... RS
; P3.4 ..... RW
; P3.3 ..... E
;
;
;
;
;-------------------------------------------------------------------------
; START OF LCD ROUTINES
;-------------------------------------------------------------------------
LCD_RS              EQU p3.5
LCD_RW              EQU p3.4
LCD_E               EQU p3.3
LCD_DATA            EQU P1
LCD_BUSY_FLAG       EQU p1.7
__lcd_wait_busy:
                    setb LCD_BUSY_FLAG
                    clr LCD_RS
                    setb LCD_RW

__WaitBusy:
                    clr LCD_E
                    setb LCD_E
                    jb LCD_BUSY_FLAG,__WaitBusy
                    clr LCD_E
                    clr LCD_RS
                    clr LCD_RW
                    ret
          ;-------------------------------------------------------------------------
          ; LCD Command
          ;-------------------------------------------------------------------------

__lcd_write_command:
                    lcall __lcd_wait_busy
                    ;-------------------------------------------------------------------
                    ; Place data
                    ;-------------------------------------------------------------------
                    mov LCD_DATA,a
```

```
                ;---------------------------------------------------------------------
                ; Write
                ;---------------------------------------------------------------------
                clr LCD_RW
                ;---------------------------------------------------------------------
                ; Select command register
                ;---------------------------------------------------------------------
                clr LCD_RS
                ;---------------------------------------------------------------------
                ; Setup Time
                ;---------------------------------------------------------------------
                nop
                ;---------------------------------------------------------------------
                ; E Pulse
                ;---------------------------------------------------------------------
                setb LCD_E
                nop
                clr LCD_E
                clr LCD_RS
                clr LCD_RW
                ret


        ;-------------------------------------------------------------------------
        ; LCD Data
        ;-------------------------------------------------------------------------

__lcd_write_data:
                lcall __lcd_wait_busy
                ;---------------------------------------------------------------------
                ; Place data
                ;---------------------------------------------------------------------
                mov LCD_DATA,a
                ;---------------------------------------------------------------------
                ; Select data register
                ;---------------------------------------------------------------------
                setb LCD_RS
                ;---------------------------------------------------------------------
                ; Write
                ;---------------------------------------------------------------------
                clr LCD_RW
```

```
                ;---------------------------------------------------------------
                ; Setup Time
                ;---------------------------------------------------------------
                nop
                ;---------------------------------------------------------------
                ; E Pulse
                ;---------------------------------------------------------------
                setb LCD_E
                nop
                clr LCD_E
                clr LCD_RS
                clr LCD_RW
                ret



        ;-------------------------------------------------------------------------
        ; __lcd_locate
        ;
        ;    ON ENTRY :
        ;                            R7 = X Coord
        ;                            R6 = Y Coord
        ;-------------------------------------------------------------------------
__lcd_locate:
                mov a,r6
                lcall __map
                add a,r7
                ;-----------------------------------------
                ; This prevent invalid commands o display
                ; Address setting is [1][A][A][A][A][A][A][A]
                ; No other command can be detected by module
                ; if the bit#7 is one
                ;-----------------------------------------
                orl a,#80h
                lcall __lcd_write_command
                ret
__map:
                inc a
                movc a,@a+pc
                ret
                db 080h
                db 0c0h
```

```
                db 094h
                db 0d4h
                ret

        ;---------------------------------------------------------------------
        ; LCD Initialization
        ;---------------------------------------------------------------------

__lcd_init:
                mov     a,#38H
                lcall __lcd_write_command
                mov a,#0cH
                lcall __lcd_write_command
                mov a,#01H
                lcall __lcd_write_command
                mov a,#06H
                lcall __lcd_write_command
                mov a,#80H
                lcall __lcd_write_command
                ret



        ;---------------------------------------------------------------------
        ; Clear LCD
        ;---------------------------------------------------------------------

__scrclr:
                mov     a,#01H
                lcall __lcd_write_command
                ret


__terminal_init:

                lcall __lcd_init
                ret
```

```
;------------------------------------------------------------------------
; Hooks
;------------------------------------------------------------------------

__update:
        ret

__locate:
        lcall __lcd_locate
        ret

__put_asciiz_code:
        clr a
        movc a,@a+dptr
        jz __eos_code
        lcall __lcd_write_data
        inc dptr
        sjmp __put_asciiz_code

__eos_code:
        ret

__put_asciiz_data:
        movx a,@dptr
        jz __eos_data
        lcall __lcd_write_data
        inc dptr
        sjmp __put_asciiz_data
__eos_data:
        ret



__putchar:
        lcall __lcd_write_data
        ret
```

Assembling:

To assemble our code we are using USASM51 kit. To make our work simple we can write a bat file and with required files to be created. So an example bat file is given below.

Note4: To create a bat file open notepad, put list of instructions and save the file as "lcd.bat"

"C:\usm51kit\USASM51.EXE" -L -oobj hmi.so1

"C:\usm51kit\USLIB.EXE" -a'HMI.Uo1' LCDl.LIB

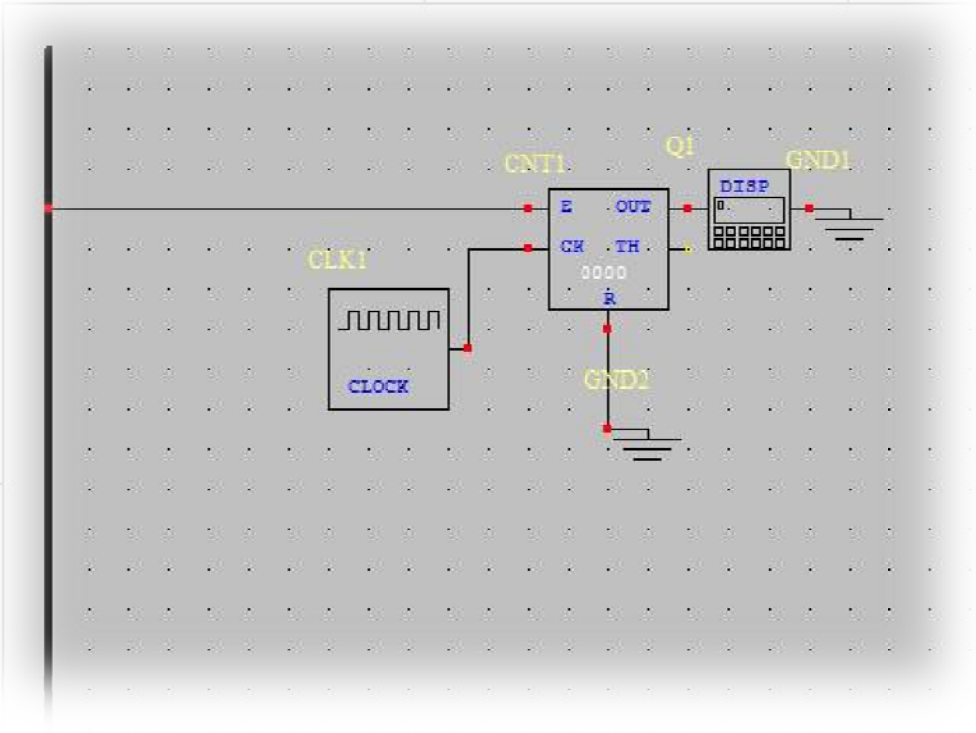
Now simply run the bat file and you will get the library file with name "LCDL.LIB". Copy this library file into your LadderWORK "lib" folder.

*Note5: As we will use external data memory so if you are creating library with your own name then append the library name with letter "L" at the end as in above case we used "LCD**L**"*

Now we need to test our LCD.

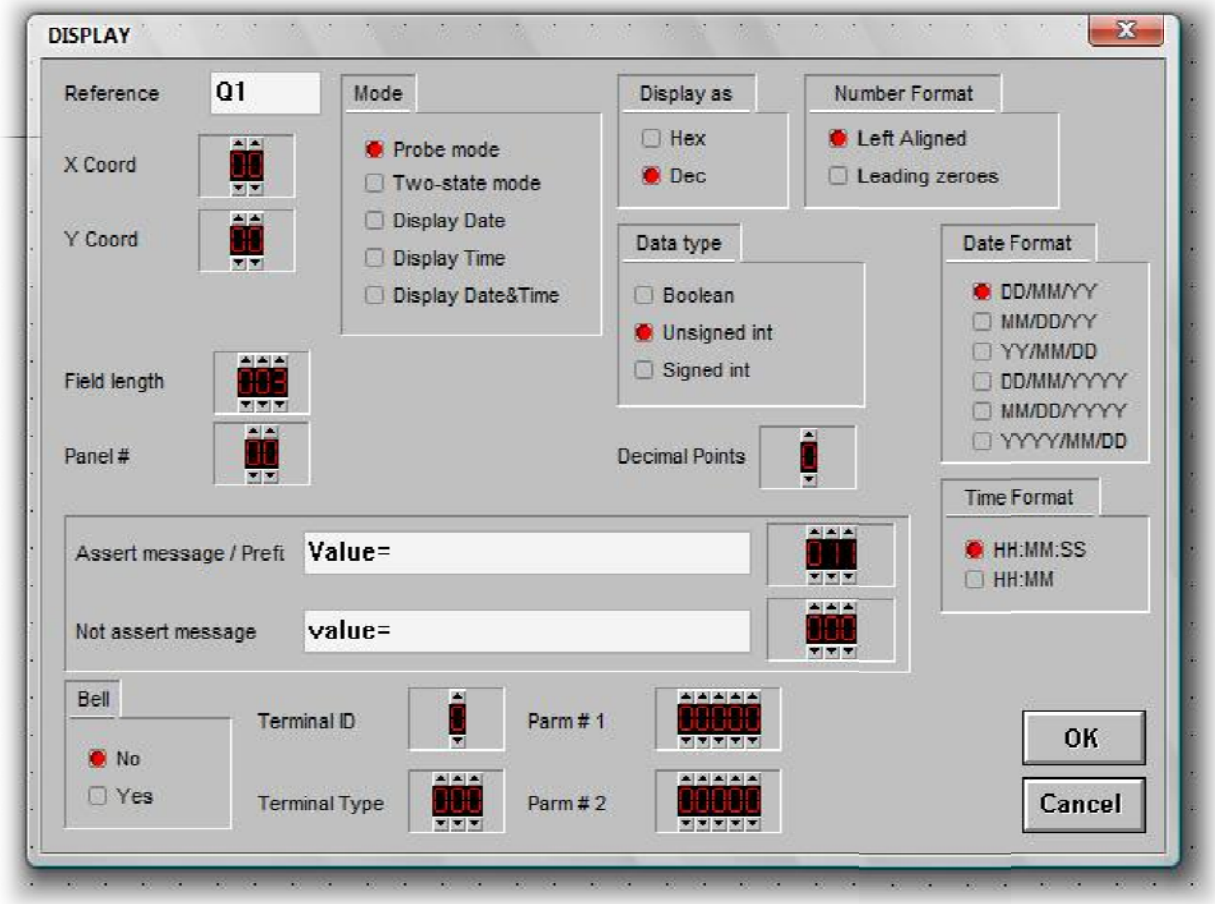
Step 4: Using LCD in LadderWORK

Now we will create a simple project to test the working of LCD.



Now set the Counter with the maximum count you want and set the clock frequency to required value. For instance we are using 1Hz of frequency and up counter from 1 to 9.

Display setup is made as shown in the diagram below.



Mode: Probe

Display As: Dec (decimal)

Number Format: Left Aligned

Data type: Boolean

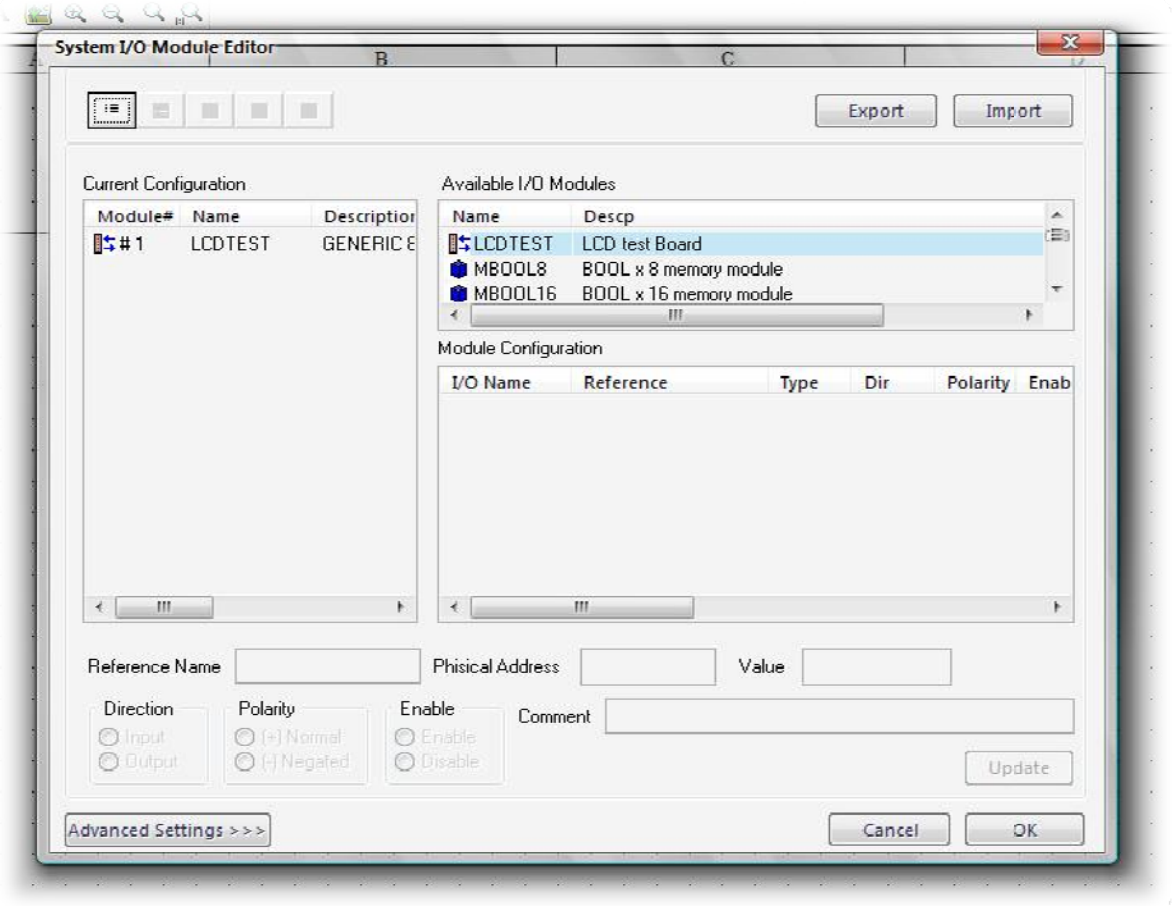
X & Y Coordinate: 0

Field Length: 3

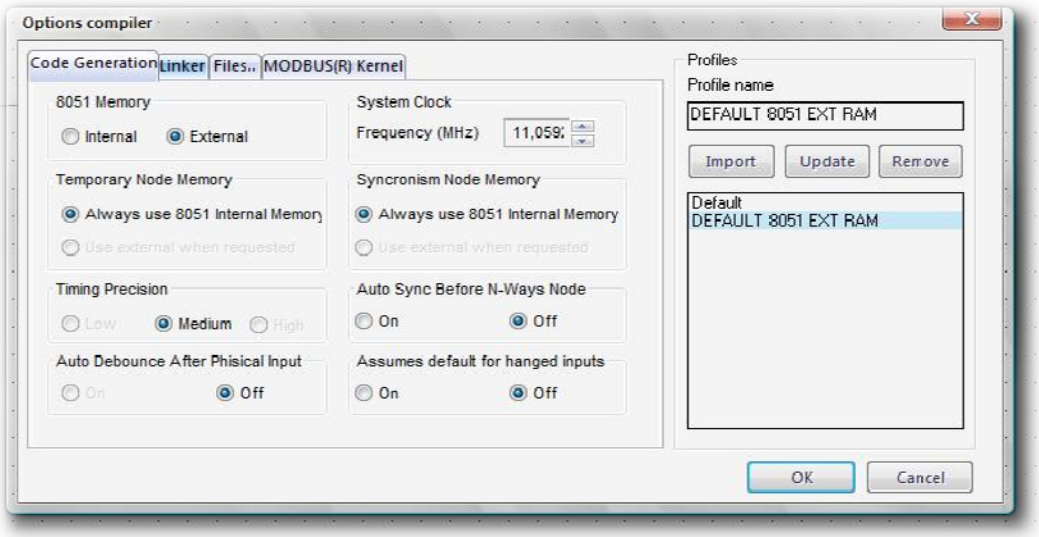
Assert and N-Assert message: Value= (you can change it to whatever you want)

Now press ok! To exit the display setup menu.

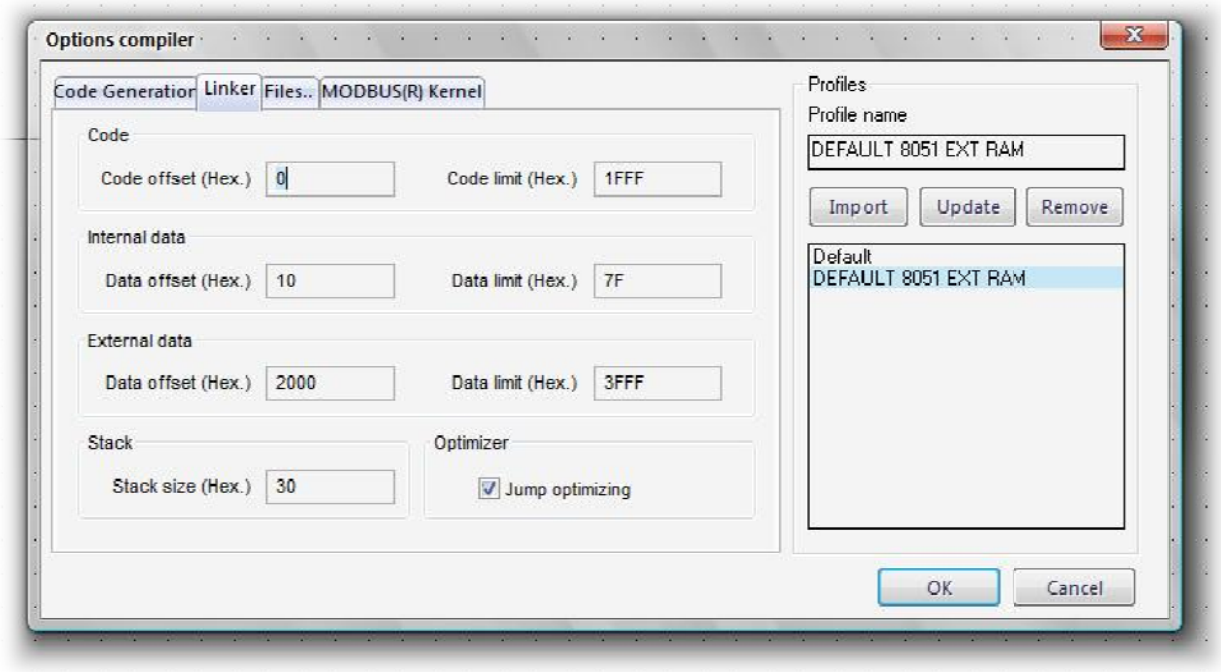
Now we need to edit the system edit configuration, as shown below.



After adding the LCD module, we need to edit the compiler configuration (compiler configuration can change according to your board).



Now click on linker tab to set the code space and data space for External RAM.



Note6: Please DO NOT FORGET TO INCREASE THE STACK SIZE to 30 or higher value.

Now we are done. Simply compiler the project. Burn the Hex file on your controller and Check the LCD. You will see count going from 1 to 9 as…

“Value= 1”

“Value= 2”

“Value= 3”

“Value= 4”

“Value= 5”

“Value= 6”

“Value= 7”

“Value= 8”

“Value= 9”

I hope this tutorial provided you enough information on what is to be done to make your HMI peripheral work with LadderWORK.