Interfacing 8051 Microcontroller with LCD Using multisim Simulator

S. B. JAGADAL and S. V. HALSE
Karnataka State Women University, Bijapur, INDIA.

(Received on: October 18, 2012)

ABSTRACT

This paper explores the interfacing techniques of 8051 microcontroller in a virtual environment with a complete example of LCD interface. This gives a real vision of device connection and program execution. We can examine a program written in C or Assembly language for LCD Display that is interfaced to the 8051 using simulator software before it is burned on a 8051. The display can be of two lines or multi-line and up to 20 characters per line; in this paper we explained the interfacing of 8051 microcontroller with LCD 2 lines, 5x7 matrix displays in a virtual environment. We programmed the 8051 to display “KSWU Bijapur” on LCD screen in the 1st line, and Electronics department” in 2nd line using LCD commands and assembly language programming.

The 8051 Microcontroller can be programmed to display any desired character string. The National Instrumentation Software Multisim V.11.0.1 does not require any 8051 board, it runs on a PC environment we can write a program to display desired Text on LCD Display along with some other programs like external Hardware interrupt, displaying count of an external event on LCD, this simulator allows user to write and debug and execute a program, and to analyze the results virtually also to examine the various register contents. It is very essential that before burning the program on actual hardware it can be tested and modified. This saves a lot of time and efforts as well as the cost of hardware testing.

Keywords: Microcontroller, LCD module, Interfacing, Virtual Environment, Multisim Simulator.
INTRODUCTION

The Multisim simulator is software simulation tools which provide an accurate simulation of digital and analog circuit operations. Multisim allows us to grasp concepts quicker and gain deeper intuition for circuits. The operating system windows XP/ Vista / 64bit Vista and Windows 7 supports fully to this Multisim software. It has been designed to help hardware designers’ gain better understanding of circuit behavior. Since the quality of simulation results is highly dependent on applied signals as well as the methods of analyzing and displaying simulation. It helps to close gap between design and test. We can interface real world signal from inside Multisim and output data to drive real world circuitry, or display simulation data in a more suitable to our needs. Using this software we can design our project before it is executed on real components. The purpose of this paper is to explore the important features of this software by giving a complete example of interfacing LCD display to 8051 microcontroller. As 8051 chip and its family is extensively used in embedded system design and in many embedded applications. And LCD display is most commonly used in embedded system and other electronic devices. Microcontrollers and LCD modules are used in various embedded applications such as copiers fax machines, laser printers industrial test equipment, network equipment: routers and storage devices. We can do many experiments or projects placing various components on workplace area of Multisim software on a PC, do connections, write the assembly language program, debug it and run the same to see the result on output devices connected. The best way to learn is to experiment, there is no need to afraid to try out complicated circuits and new features in Multisim. So beginner can easy simulate complex circuits. One of the most powerful features of is its interactive nature. It enhances visualization, and makes capture easier. We should know how our circuit works and so we can figure out if our simulation makes sense or not.

CIRCUIT DIAGRAM:

![Circuit Diagram](image)

Figure 1: schematic diagram of interfacing 8051 MCU with LCD display.
INTERFACING TECHNIQUES OF 8051

Microcontrollers can communicate with other devices such as sensors, motors. Techniques/methods have been developed over the years to best meet optimization challenges in embedded systems. Interface can be analog or digital. Analog interface is based on voltage and current monitoring whereas digital interface is based on ON/OFF monitoring. Since microcontrollers do not have built-in analog input and output, so analog interface is complicated because of the use of external ADC or DAC. Many microcontroller designs typically mix multiple interfacing methods. Microcontroller can be used to single device simplest interface for on/off monitoring. For example digital inputs/outputs in case of reading the status of buttons or switches, keypad interface, LED interface and relay interface. Many interface methods have been developed over the year to solve the complex problem of balancing circuit design criteria such as features, cost, size, power consumption, reliability, manufacturability.

LCD DISPLAY

Liquid crystal display also called LCD is a very helpful in providing user interface as well as debugging purposes. The LCD is finding widespread use replacing LEDs (seven segment LEDs). Because of the incorporation of a refreshing controller into the LCD, ability to display numbers, characters, and graphics, and also because of low price of LCD. In contrast, the LED must be refreshed by the CPU to keep displaying the data and they have limited numbers and characters to display.

The LCD requires three control lines (RS, R/W & EN) & 8(or 4) data lines. The number on data lines depends on the mode of operation. If operated in 8-bit mode then 8 data lines+3 control lines i.e. total 11 lines are required. And if operated in 4-bit mode then 4 data lines +3 control lines i.e. 7 lines are required. How do we decide which mode to use? It’s simple if you have sufficient data lines you can go for 8 bit mode & if there is a time constrain i.e. display should be faster then, we have to use 8-bit mode because basically 4-bit modes takes twice as much time as compared to 8-bit.

The LCD used here has 14 pins there are 8 data lines which is to connected to any one port of 8051 Vcc and Vss provide +5V supply. The Vee pin 3 is used for controlling LCD contrast. The actual implementation of the contrast control function varies according to the manufacturer. There are two very important register inside the LCD. the RS (register select pin) used for their selection as follows, if RS pin is=0 ,the instruction command code register is selected, allowing the user to send a command such as clear screen, cursor at home, etc. If RS=1 the data command register is selected, allowing the user to send data to be displayed on the LCD. The enable E pin is used to latch information presented to its data pins. When data is supplied to this pin, a high-to-low pulse must be applied to this pin in order for the LCD to latch in the data present at the pins. This pulse must be a minimum of 450ns wide. R/W input pin allows user to write information to the LCD or read information from it. R/W=1 when reading. R/W=0 when writing. There are two internal registers labeled the data register and the instruction register, a RAM area of
display data (DDRAM), a character generator ROM, a character generator RAM, the data register, DR, is used to temporarily store data to be written into DDRAM or CGRAM as well as temporarily store data read from DDRAM or CGRAM. Data placed in the data register is automatically written into DDRAM.

![Image of 7 segment indicator](image)

Figure 2: Shows the 7 segment indicator in the dialog box……

We also use RS=0 to check the busy flag bit to see if the LCD is ready to receive the information. The busy flag is D7 and can be read when R/W =1 and RS=0, as follows: if R/w=1, RS=0. If D7=1 the LCD is busy taking care of internal operations i.e., busy flag =1. If D7=0, the LCD is ready to receive new information. So it is necessary to check the busy flag before any data is written to the LCD. To send any commands to the LCD, make pin RS=0, for data, and make RS=1 for command, then send a high-to-low pulse to the enable pin E to enable internal latch of the LCD.

Figure 1 shows the circuit diagram which is constructed by using Multisim software on a personal computer screen i.e. the workplace area, where the required equipment’s are placed in the workplace, the

multisim simulator software allows user to lead through the circuit design flow, from schematic capture, through simulation and analysis. To launch this simulator we have to select the all programm >> national instruments >> multisim v 11.0.1 then a blank file opens, then select the file option from menu bar and select a new >> design a dialog box appears on screen of PC by a name select a component. Among various groups select MCU 8051 click ok. Then place the MCU 8051 on workplace at a desire place. Soon after placing again a dialog box appears (MCU wizard - step1 of 3). In the 1st step we have to enter the workplace name, next select the programming language option in c/assembly, then press next. In the last step we need to write the source file name i.e. if we are writing the code in assembly then we have to take the source file name as main.asm. or save the same with a desire name. We have to all place the components according to our schematic design, to do so we select place >> components to display the select a component browser as shown below.

Navigate to 8051 microcontroller and LCD one by one and click ok, these appear as “ghost” on the cursor. We can choose any combination of component’s value, for example the register value, type etc. as in fig 2 shown.

All components have pins that we use to wire them to other components or instruments. As soon as our cursor is over the a pin, Multisim knows we want to wire and the pointer changes to a crosshair.

![Figure 3: Schematic connection on PC](image-url)
To wire the circuit....

1. Click on a pin on a component to start the connection (your pointer turns into a crosshair) and move the mouse. A wire appears, attached to your cursor.
2. Click on a pin on the second component to finish the connection. Multisim automatically places the wire, which conveniently snaps to an appropriate configuration, as shown below. This feature saves a great deal of time when wired a large circuits.
3. We can also control the flow of wire by clicking on points as we move the mouse. Each click “fixes” the wire to that point.

We can finish the interfacing of LCD with 8051 microcontroller as shown in fig 2. In the workplace.

To display the text on LCD screen, write the assembly language program by selecting MCU option from main menu bar>>MCU 8051 U1>>MCU code manager, this displays a box. Double click on main.asm then press ok button. Here appear a comment; please enter your code here. Below this comment we can start writing codes as under.

CD program to display a text

Following is the assembly language program to display DEPT OF ELECTRONICS, KSWU BIJAPUR on LCD screen.

$MOD51 ; this includes 8051 definitions for the Metalink assembler

; Please insert your code here.
ORG 0000

MOV A,#38H ; initialize. LCD 2 lines, 5x7 Matrix.
ACALL COMNWRT ; Call command Subroutine
ACALL DELAY ; Give LCD some time.
MOV A, #0EH ; Display on, cursor on.
ACALL COMNWRT ; Call command Subroutine.
ACALL DELAY ; Give LCD some time.
MOV A, #01 ; Clear LCD.
ACALL COMNWRT ; Call command subroutine
; ACALL DELAY ; Give LCD sometime
MOV A, #06H ; Shift cursor right.
ACALL COMNWRT ; ACALL DELAY
BACK: MOV A, #82H ; Cursor at line 1 position 2
ACALL COMNWRT ; Call command subroutine.
; ACALL DELAY ; Give LCD some time

; // MESSAGE DISPLY
MOV A, #'K' ; Display letter K
ACALL DATAWRT ; Call Data command subroutine
MOV A, #'S' ; Display letter S
ACALL DATAWRT ; Call Data command subroutine
MOV A, #'W' ; Display letter W
ACALL DATAWRT ; Call Data command subroutine
MOV A, #'U' ; Display letter U
ACALL DATAWRT ; Call Data command subroutine
MOV A, #0 ; Leave some space
ACALL DATAWRT ; Call Data command subroutine
MOV A, #'B' ; Display letter B
ACALL DATAWRT ; Call Data command subroutine
MOV A, #'I' ; Display letter I
ACALL DATAWRT ; Call Data command subroutine
MOV A, #'J' ; Display letter J
ACALL DATAWRT ; Call Data command subroutine
MOV A, #'A' ; Display letter A
ACALL DATAWRT ; Call Data command subroutine
MOV A, #'P' ; Display letter P
ACALL DATAWRT ; Call Data command subroutine
MOV A, #'U' ; Display letter U
ACALL DATAWRT ; Call Data command subroutine
MOV A, #'R' ; Display letter R
ACALL DATAWRT

; SECOND LINE DISPLAY
MOV A, #0C0H ; Display in second line on position 0
ACALL COMNWRT ; Call Data command subroutine
MOV A, #'E' ; Display letter E
ACALL DATAWRT ; Call Data command subroutine
MOV A, #'L' ; Display letter L
ACALL DATAWRT ; Call Data command subroutine
MOV A, #'E' ; Display letter E
ACALL DATAWRT ; Call Data command subroutine
MOV A, #'C' ; Display letter C
ACALL DATAWRT ; Call Data command subroutine
MOV A, #'T' ; Display letter T
ACALL DATAWRT ; Call Data command subroutine
MOV A, #'R' ; Display letter R
ACALL DATAWRT ; Call Data command subroutine
MOV A, #'O' ; Display letter O
ACALL DATAWRT ; Call Data command subroutine
MOV A, #'N' ; Display letter N
ACALL DATAWRT ; Call Data command subroutine

MOV A, #'I'          ; Display letter I
ACALL DATAWRT        ; Call Data command subroutine

MOV A, #'C'           ; Display letter C
ACALL DATAWRT         ; Call Data command subroutine

MOV A, #'S'           ; Display letter S
ACALL DATAWRT         ; Call Data command subroutine

MOV A, # '-'          ; Display -
ACALL DATAWRT         ; Call Data command subroutine

ACALL DATAWRT         ; Call Data command subroutine
MOV A, # 'D'          ; Display letter D
ACALL DATAWRT         ; Call Data command subroutine

MOV A, # 'E'          ; Display letter E
ACALL DATAWRT         ; Call Data command subroutine

MOV A, # 'P'          ; Display letter P
ACALL DATAWRT         ; Call Data command subroutine

; MOV A, # 'M'          ; Display letter M
; ACALL DATAWRT         ; Call Data command subroutine

MOV A, # 'T'          ; Display letter T
ACALL DATAWRT         ; Call Data command subroutine

MOV A, #01            ; Clear screen
ACALL COMNWRT         ; Call Data command subroutine
SJMP BACK             ; Keep displaying these letters

; AGAIN: SJMP AGAIN
COMNWRT:

    MOV P1, A
    CLR P3.0; RS=0 FOR COMMAND WRITE
    CLR P3.1; R/W=0 FOR WRITE
    SETB P3.2; E=1 FOR HIGH PULSE
    CLR P3.2 ; E=0 FOR H-TO-L PULSE
    RET

DATAWRT:

    MOV P1, A; WRITE DATA TO LCD
    SETB P3.0; RS=1 FOR DATA
    CLR P3.1; R/W=0 FOR WRITE
    SETB P3.2; E=1 FOR HIGH PULSE
    CLR P3.2 ; E=0 FOR H-TO-L PULSE
    RET

DELAY:

    MOV R4, #1

HERE: DJNZ R4, HERE
RET
END

When the assembly program is written, debug and run the program to display the text on LCD display. To do so we have to click the green color play button this appears below the main menu bar.

RESULT AND CONCLUSION

This program displays the two lines text KSWU BIJAPUR and electronics department on LCD screen. We can use this program to display our desired messages as required. We can also program using interrupt service subroutine to display a message after an interrupt occurs.

The Multisim simulator helps us to perform and practice experiments to improve our understanding of the various electronic concepts. It is also helpful to design and program embedded system applications in our further research work. We can debug, execute verify our results before implementing any project on real hardware’s.

REFERENCES

3. Multisim guide pdf
4. Multyremotes.com/keil-interfacing-programs.htm
5. MICROCONTROLLER: Theory and Applications: By Ajay V. Deshmukh.
7. Some Assembly Language Examples www.scm.tees.ac.uk/users/u0000408/CYS/68Kexamples.htm
8. tenettech.com/ustick/?page_id=35 (Interfacing LCD to 8051 U stick. T) www.scm.tees.ac.uk/users/u0000408/CYS/68Kexamples.htm
9. The 8051 Microcontroller and embedded system by Muhammad Ali Mazidi Janice Gillispie
11. www.google.co.in. Microcontroller interfacing techniques pdf
12. Wikipedia MCS 8051 and LCD interface
13. www.slideshare.net/.../interfacing-lcd-with-8051-microcontroller...
14. www.dnatechindia.com › TUTORIAL › 8051 Tutorial