Computer Interface for Liquid Crystal Display (LCD)

M. Rabiee
Eastern Kentucky University

Abstract:

Mono-Color Liquid Crystal Displays (LCD) are often used in portable instruments, calculators, and digital watches. Color LCD displays are used with Lap-Top computers and small color televisions. In this paper we will first explain the construction of LCD displays. Then, we will design and build a digital interface circuit to connect a LCD display to the computer. Finally, we will discuss how to write a program to drive a particular LCD display.

Introduction:

Liquid Crystal Displays (LCDs) use very low power. Therefore, they are often used in battery-powered instruments. LCDs however do not omit light. They simply change the reflection of available light. Liquid Crystal Displays are created by placing a thin (approximately 10 mm) layer of liquid crystal fluid between two glass plates. A transparent electrically conductive film is placed on the rear glass plate. Transparent sections of the conductive film in the shape of characters are coated on the front glass. Voltage is applied between the LCD segment and the backplane. This produces an electric field region under the segment. The electric field changes the transmission of the light through the region under the segment film.

There are two types of commonly available LCDs, the Dynamic Scattering, and the Field Effect. In the Dynamic scattering LCD Display type, the modules under the fields are scrambled. This produces light characters on a dark background. In the Field-Effect type, light is absorbed in the presence of an electric field. This produces dark character on a silver-gray background. Figure 1 illustrates the basic construction of a LCD display.

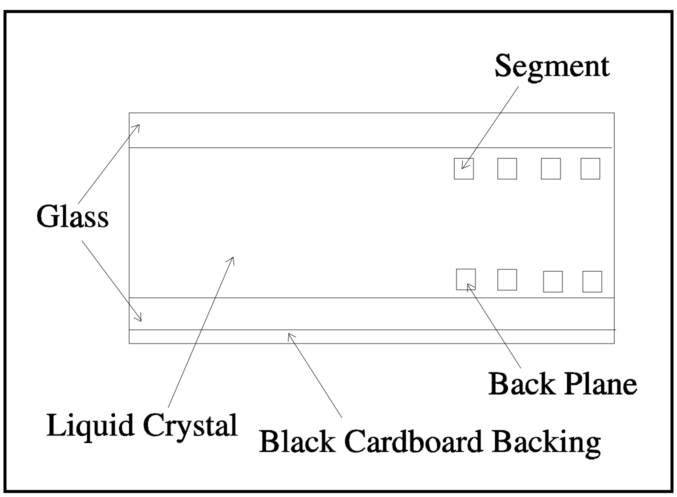

Figure 1 A Liquid Crystal Display

In this paper we will describe the use of the Hitachi 2570 Mono-Color Dot Matrix Liquid Crystal Module. This module consists of a 16-character x 1 line LCD, a built-in Large Scale Integrated (LSI) circuit controller chip (HD44780), and two Large Scale Integrated (LSI) circuit LCD display drivers. Figure 2 shows the schematic diagram of a HD2570 LCM [1].

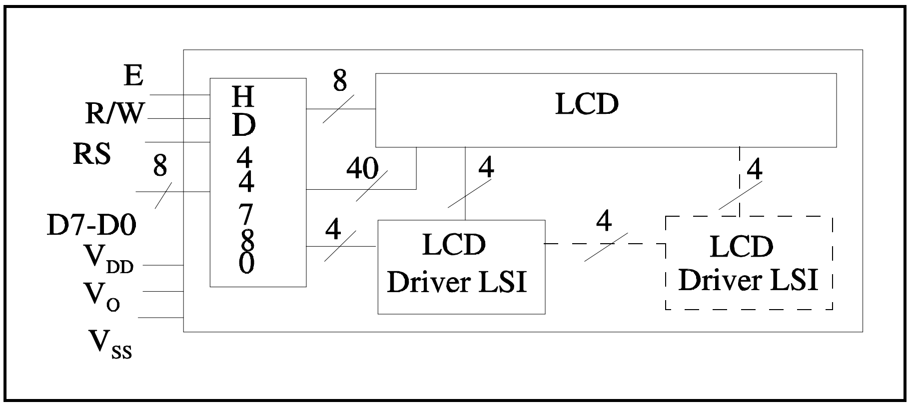

Figure 2 Driver Circuit Block Diagram

Figure 3 shows the computer interface system block diagram. We will use the 8255 Programmable Peripheral Interface (PPI) chip to connect the LCD module to the PC-Interface card that resides in an IBM compatible personal computer.

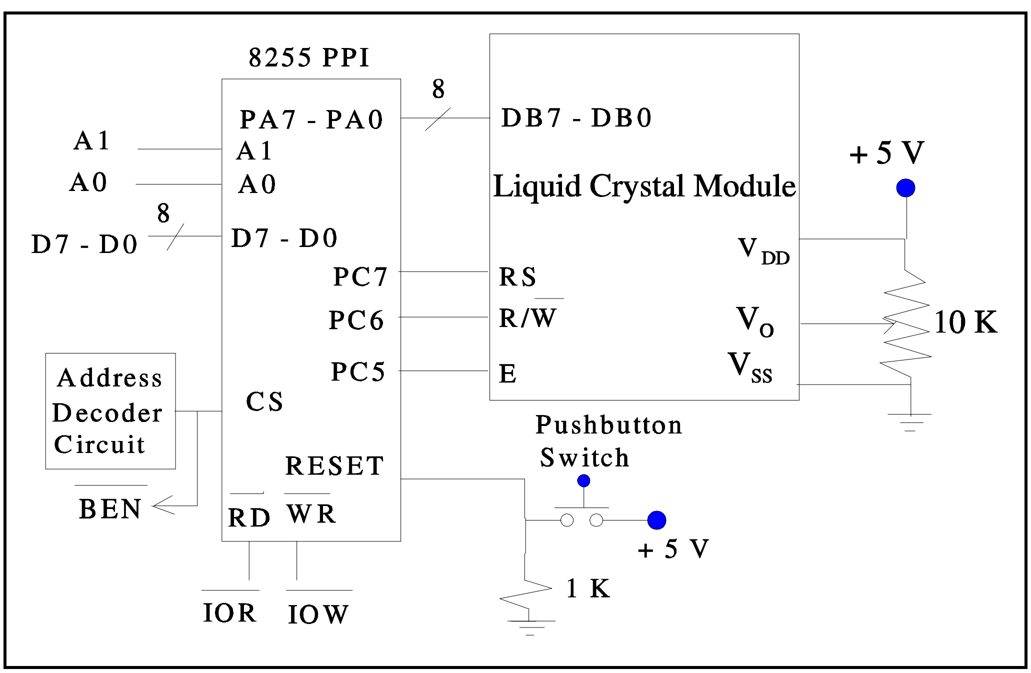

Figure 3 System Block Diagram

The PC-Interface card can be purchased through the Electronic Industries Association (EIA) [2]. Figure 4 illustrates the pinout diagram of the PC-Interface card. This card can be placed in a 32-pin Industry Standard Association (ISA) port of an IBM compatible personal computer.

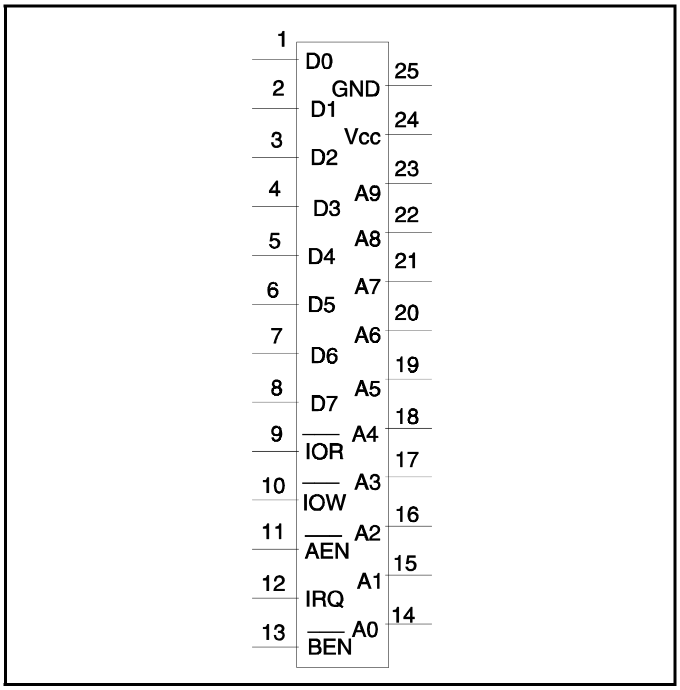

Figure 4 Pinout Diagram for the PC Interface Card

As illustrated in figure 3, we will use an 8255 Programmable Peripheral Interface (PPI) chip to drive the Liquid Crystal Module (LCM). Figure 5 shows the pinout diagram of an 8255-PPI chip [3].

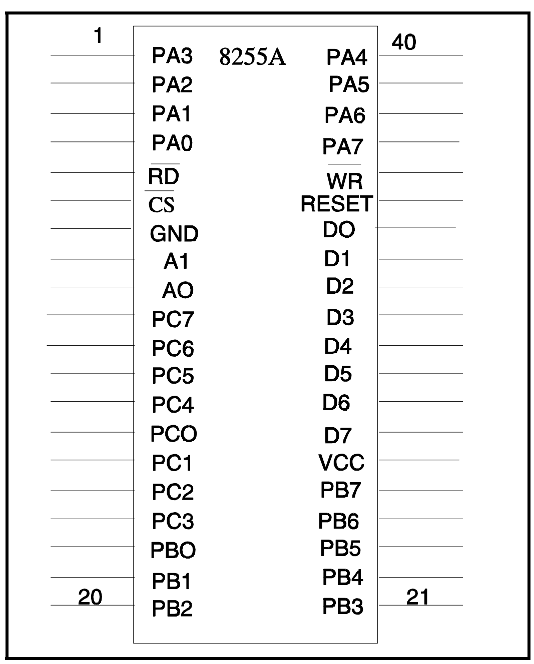

Figure 5 The 8255 PPI pinout diagram

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	HEX Address Range
0	0	0	0	0	0	1	1	0	0	0	X	X	↕	↕	↕	0300h TO 03FFh

Figure 6 Address decoding map

Address Decoder Circuit:

On most personal computers the address range 300-to-3FF hex is available for external use. Figure 6 shows the address map of a decoder circuit that will generate the chip-enable signals for this address range.

Using Transistor-Transistor Logic (TTL) gates, we will build the decoder circuit displayed in Figure 7**.** We will connect the output of the decoder circuit to the Bus Enable (BEN) pin of the PC interfacing card, and the chip-select pin of an 8255 Programmable Peripheral Interface (PPI) chip.

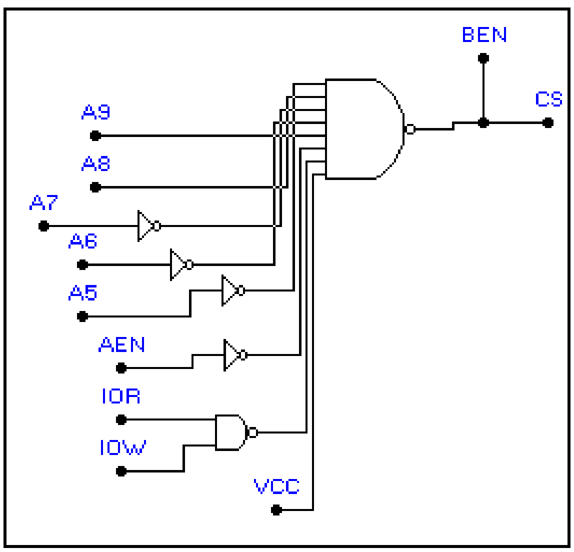

Figure 7 The Address Decoder Circuit

System Software:

The 8255 PPI chip has four internal registers. The signal on the chip-select (CS) pin will turn the chip on. The address lines A0 and A1 will select one of the four internal registers at one time. Table 1 contains the address assignments for ports A, B, C, and the Control Register (CR).

Port/register Number	**A1**	**A0**	**Address in Hex Number**	**Address in decimal**
Port A	0	0	300 h	768 decimal
Port B	0	1	301 h	769 decimal
Port C	1	0	302 h	770 decimal
Control Register (CR)	1	1	303 h	771 decimal

Table 1 Address Assignments for Internal Registers of the 8255 PPI

The 8255 PPI chip can operate in three different modes [3]. The control word in the control register sets the 8255 chip in mode zero (0), one (1), or two (2). Mode zero is simple input/output for ports A, B, and C. Mode one is input/output with partial hand shacking for ports A and B. Mode two is bidirectional input/output for port A only. Table 2 contains the control word that will configure the 8255 PPI to operate in mode zero. All three ports A, B, and C are set to be output ports operating in mode zero.

D7	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	
1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**= 80H**

Table 2 The 8255 PPI Control Word for Mode Zero Output Ports

Table 3 contains the internal pin connections of the HD44780 LCD module [4]. We will configure the LCD display for an 8-bit long data interface [5]. Note that for an instruction code input the RS line must be low, while for data input the RS line must be high. Also, in order to transfer the ASCII (American Standard Code for Information Interchange) data or instruction code, we must hold the E-line high for at least 5 milliseconds and then let it drop low. At the trailing edge of the pulse on E-line, the octal latch will input the data into the LCD driver (see figure 2).

Pin No.	Symbol	Level	Function
1	V_{SS}	—	0 V
2	V_{DD}	—	+5 V
3	V_{O}	—	—
4	RS	H/L	L: Instruction Code Input; H: Data Input
5	R/W	H/L	H: Data Read from LCD; L: Data Write to LCD
6	E	H, H/L	Enable Signal
7 - 14	DB0 - DB7	H/L	Data Bus Line

Table 3 The Hitachi HD44780 LCD Display Internal Pin Connection

Figure 8 shows the flow chart of the program. We will use the DOS Function Call INT 21 with an entry signal of AH= 01 to read the characters from the computer keyboard [6]. The characters will be displayed on the computer screen, and the LCD display, simultaneously. When sixteen characters are displayed on the LCD display, an instruction code will clear the LCD display and move the cursor to the left-hand side of the LCD display. We can now display more characters on the LCD display.

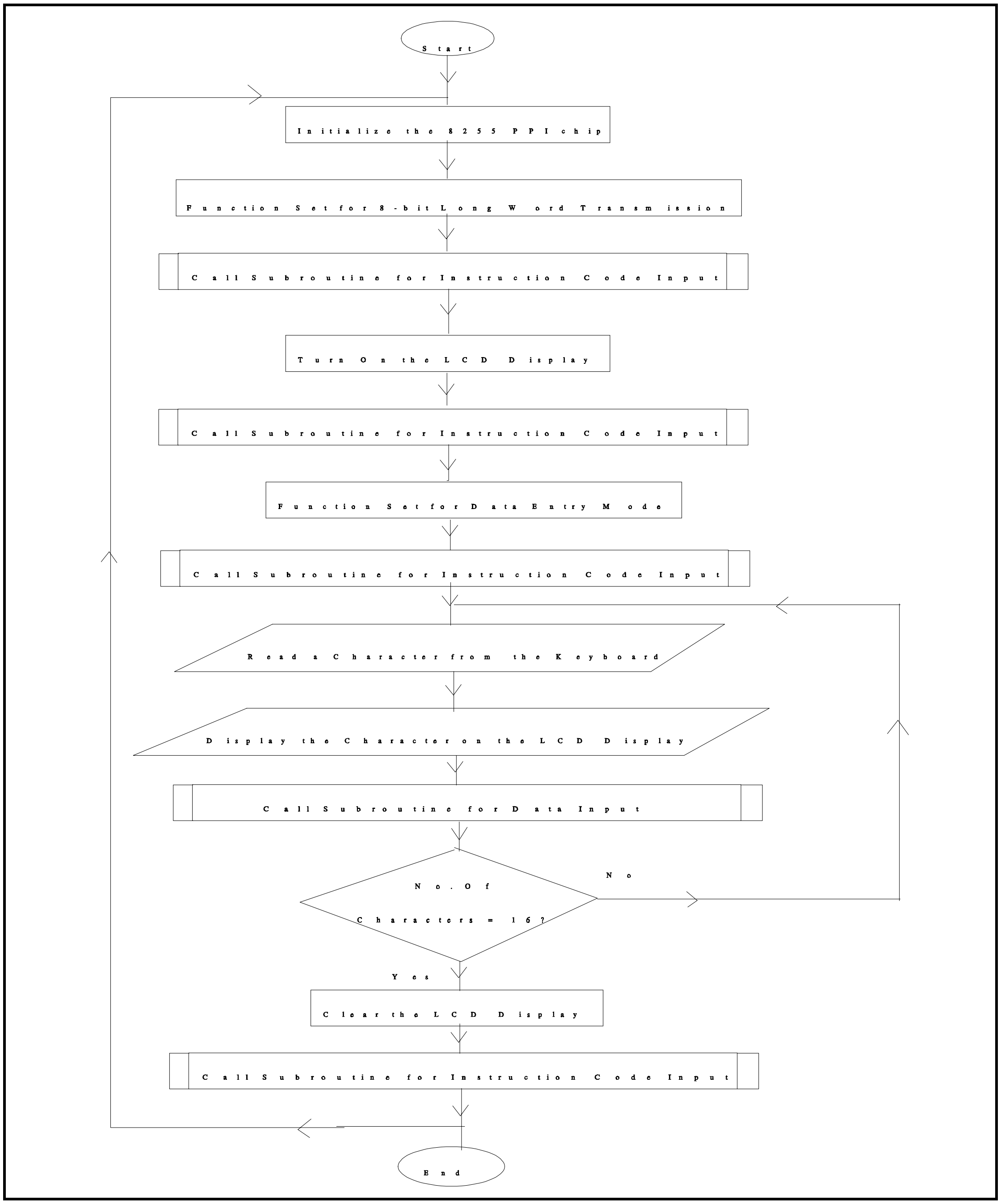

Figure 8 Program Flow Chart

There are two subroutines in the program. The subroutine that starts at the address CS:200 is used every time we want to send an instruction code. This subroutine will cause the RS and R/W lines to drop low. Then the E-line is held high for 5 milliseconds, and then dropped low.

The subroutine that starts at address CS:300 is used when we send ASCII data to the LCD module. In this subroutine, the RS line is held high, while the R/W line is held low. Then, similar to the subroutine starting at address CS:200, we will pulse the E-line.

Appendix A contains the assembly language program with comments on each instruction line.

Conclusion:

The market demand for portable electronic devices has created a challenge for the digital electronic industry. To meet this challenge, we not only need to experiment with the construction of batteries that can hold more electric energy, but also the more efficient use of low power Integrated Circuit (IC) chips, and displays. Lab experiments similar to the one described in this paper will spark an interest in the classroom for design, construction, and use of efficient electronic circuits.

Appendix:

; The Main Program

Address	Op- Code	Operand	Comment
1E08:0100	MOV	DX,0303	; Initialize the 8255 PPI chip to set port A, B,
1E08:0103	MOV	AL,80	; and C as output operating in mode zero.
1E08:0105	OUT	DX,AL	;
1E08:0106	MOV	DX,0300	; Function set to operate the LCD display for 8-bit
1E08:0109	MOV	AL,30	; long word.
1E08:010B	OUT	DX,AL	;
1E08:010C	CALL	0200	; Go to the subroutine for Instruction Code Input.
1E08:010F	MOV	AL,0E	; Function set to turn the LCD display on.
1E08:0111	MOV	DX,0300	;
1E08:0114	OUT	DX,AL	;
1E08:0115	CALL	0200	; Go to the subroutine for Instruction Code Input.
1E08:0118	MOV	AL,06	; Function set for Data Entry Mode.
1E08:011A	MOV	DX,0300	;
1E08:011D	OUT	DX,AL	;
1E08:011E	CALL	0200	; Go to the subroutine for Instruction Code Input.
1E08:0121	MOV	BL,00	; Set the counter to zero.
1E08:0123	MOV	AH,01	; Read a character from the keyboard.
1E08:0125	INT	21	;
1E08:0127	MOV	DX,0300	; Point to the LSI controller in the LCD module.
1E08:012A	OUT	DX,AL	; Display the character on the LCD display.
1E08:012B	CALL	0300	; Go to the subroutine for Data Input.
1E08:012E	INC	BL	; Increment the counter.
1E08:0130	CMP	BL,10	; Check to see if we have displayed 16 characters?
1E08:0133	JNB	0137	; If yes, then clear the LCD display.
1E08:0135	JMP	0123	; If no, then go to read more characters.
1E08:0137	MOV	AL,01	; Clear the LCD display.

```
1E08:0139    MOV     DX,0300    ;
1E08:013C    OUT     DX,AL      ;
1E08:013D    CALL    0200       ;  Go to the subroutine for Instruction Code Input.
1E08:0140    JMP     0100       ;  Go to the top of the program.
1E08:0142    INT     20         ;  End.
```

; Subroutine for Instruction Code Input

Address	Op- Code	Operand	Comment
1E08:0200	MOV	AL,20	; Set the Enable (E) line. Clear the R/W line for
1E08:0202	MOV	DX,0302	; writing code to the LCD. Clear the RS line to
1E08:0205	OUT	DX,AL	; operate in the Instruction Code Input mode.
1E08:0206	MOV	CX,0A32	; Wait for five milli seconds (5 ms).
1E08:0209	NOP		;
1E08:020A	NOP		;
1E08:020B	NOP		;
1E08:020C	NOP		;
1E08:020D	NOP		;
1E08:020E	LOOP	0209	;
1E08:0210	MOV	AL,00	; Drop the enable (E) line to release the data from
1E08:0212	MOV	DX,0302	; the LSI controller onto the LCD display.
1E08:0215	OUT	DX,AL	;
1E08:0216	RET		; Return from the subroutine.

; Subroutine for Data Input

Address	Op- Code	Operand	Comment
1E08:0300	MOV	AL,A0	; Set the Enable (E) line. Clear the R/W line for
1E08:0302	MOV	DX,0302	; writing code to the LCD. Set the RS line to
1E08:0305	OUT	DX,AL	; operate in the Data Input mode.
1E08:0306	MOV	CX,0A32	; Wait for five milli seconds (5 ms).
1E08:0309	NOP		;
1E08:030A	NOP		;
1E08:030B	NOP		;
1E08:030C	NOP		;
1E08:030D	NOP		;
1E08:030E	LOOP	0309	;
1E08:0310	MOV	AL,80	; Drop the enable (E) line to release the data from
1E08:0312	MOV	DX,0302	; the LSI controller onto the LCD display.
1E08:0315	OUT	DX,AL	;
1E08:0317	RET		; Return from the subroutine.

References:

1. Hitachi LCD Display Reference Manual, Section 6, HD44780, Page 240.

2. Electronic Industries Association, Consumer Electronic Group, Washington, D.C.

3. Intel Peripheral Components, Intel Literature Sales, Mt. Prospect IL, 1993.

4. Hitachi LCD Display Reference Manual, Section 5, H2570, Page 140.

5. Hitachi LCD Display Reference Manual, Section 6, HD44780, Pages 253 and 254.

6. IBM Technical Reference Manual, Chapter 5 (DOS Interrupts and Function Calls).

Massoud Rabiee received his Ph.D. in Electrical Engineering, from University of Kentucky, in 1987. He is presently an associate professor at Eastern Kentucky University. Dr. Rabiee is a registered professional Engineer in the State of Kentucky, and a member of IEEE, ASEE, and NAIT.