Interfacing an HD44780 Character LCD to a MicroConverter®

By Eamon Neary

INTRODUCTION

There is a vast array of LCD displays available. Fortunately, a majority of them comply with the HD44780U standard. This standard refers to the LCD controller chip that accepts data from the MicroConverter and communicates with the LCD screen.

HD44780 standard LCD screens are available in numerous formats, the most popular of which are the 16 × 2 and 20 × 2 formats. The various commands to control the basic functions of the LCD are outlined in this application note.

INTERFACING AN HD44780 LCD

The data bus that connects the HD44780 to the MicroConverter can be eight bits or four bits wide; this document discusses the 8-bit data bus. In addition to the data bus, three control lines are needed, requiring a total of 11 pins to interface the LCD to the MicroConverter.

The eight data lines that form the data bus are referred to as DB0 to DB7.

The three control lines are referred to as EN, RS, and RW:

EN is the enable line. This line is used to indicate the start of a transmission of a data byte to the LCD controller. To indicate the start of transmission, this line is brought high. When transmission is complete, the EN line is brought low.

RS is the register select line. This line indicates to the LCD controller whether the data byte is to be treated as a command or as text data to be displayed on the screen. If the RS line is high, the data byte is treated as text to be displayed. If the RS line is low, the data byte is treated as a command.

RW is the read/write line. When this line is low, the information on the data bus is written to the LCD controller. If this line is high, the LCD controller can be read to check the status of the LCD.

As shown in Figure 1, the eight data lines are connected to Port 0 of the MicroConverter; external pullups are required on Port 0 (not shown in diagram). The three control lines are connected to Ports 2.6, 2.7, and 3.6.

CONFIGURING THE LCD SCREEN

To display text to the LCD screen, it first must be configured. The LCD controller is told what sort of LCD screen is being used as well as the data bus format and the font.

The various commands available are listed in detail in the Hitachi HD44780 data sheet.

In the following example, the LCD is configured to use an 8-bit data bus and to display in 5 × 10 dot character font. This configuration is achieved by sending 38H to the LCD controller.

Three functions are used:

SEND_CMD, CHKBUSY, and PULSE_E

In assembly code, this interface is defined using the following:

```
SEND_CMD:   MOV   CHAR,R0     ; SAVES R0 TO LOCATION CHAR
            CALL  CHKBUSY     ; CHECKS BUSY FLAG
            MOV   R0,CHAR     ; PULLS CONTENTS OF R0
            CLR   LCD_WR      ; CLEARS R/W LINE
            CLR   LCD_RS      ; CLEARS RS LINE
            CALL  PULSE_E     ; CLOCKS IN DATA INTO LCD
            RET

CHKBUSY:    MOV   R0,#0FFH  ; SAVES R0 TO LOCATION CHAR
            MOV   LCD_DATA,R0  ; SETS UP PORT 0 AS I/P'S
            CLR   LCD_RS      ; CLEARS RS LINE
            SETB  LCD_WR      ; SETS R/W LINE
            NOP
            NOP
```

Figure 1. Connecting an HD44780 LCD to a MicroConverter

REV. 0 (uC014)
AN-645

BUSYREAD: SETB LCD_EN ; CLOCKS E HIGH
NOP
NOP
NOP
MOV ACC,LCD_DATA ; READS PORT 0
NOP
NOP
NOP
CLR LCD_EN ; CLOCKS E LOW
NOP
NOP
JB ACC.7,BUSYREAD ; IF BUSY, LOOP
RET

PULSE_E: NOP
NOP
SETB LCD_EN ; CLOCKS E HIGH
NOP
NOP
NOP
NOP
NOP
NOP
CLR LCD_EN ; CLOCKS E LOW
NOP
NOP
RET

The SEND_CMD function sends the command byte that is contained in R0 to the MicroConverter.

The CHKBUSY function checks the busy status of the LCD controller and waits for the instruction to be fully executed.

The PULSE_E function clocks in the data to the LCD controller.

Thus, the code to initialize the LCD display is as follows:

; Display initialization
MOV R0,#038H ; Function set
CALL SEND_CMD ;
MOV R0,#06H ; Entry set
CALL SEND_CMD ;
MOV R0,#02H ; Home
CALL SEND_CMD ;
MOV R0,#0CH ; Display on, cursor, etc.
CALL SEND_CMD ;

The additional commands are used to position the cursor to the start of the first row of the LCD. With the LCD initialized, text can be written to the screen.

WRITING TEXT TO THE LCD DISPLAY

In the sample program, the following text is written to the LCD screen:

"ANALOG DEVICES
LCD EXAMPLE CODE"

To do this, the string is defined in code memory:

TEXT0: DB ' ANALOG DEVICES '; welcome message
DB 099h
DB 'LCD EXAMPLE CODE '
DB 099h

The 099h is used by the code to decide to print the rest of the string on the next line.

Three additional functions are required to print the text:

BOTHLINES: PUSH PSW ; =>Save Carry
ACALL LINE1
INC DPTR
ACALL LINE2
FIN:
PUSH PSW
RET

LINE1: MOV R0,#80h ; CURSOR ADDRESS=
CALL SEND_CMD ; beginning of first line
CALL WRITE ; Write the full line
RET

LINE2: MOV R0,#0C0h ; set CURSOR ADDRESS
CALL SEND_CMD
CALL WRITE ; Write the full line
RET

WRITE:
write_loop: MOV A,R1
MOVC A,@A+DPTR
CJNE A,#99h,write_cont ; stop; writing; condition
RET

write_cont: MOV R0,A
CALL send_char ; send one ASCII character
INC DPTR ; next character
JMP write_loop

SEND_CHAR: MOV CHAR,R0 ; SAVES R0 TO LOCATION CHAR
CALL CHKBUSY ; CHECKS BUSY FLAG
MOV R0,CHAR ; PULLS CONTENTS OF R0
; FROM CHAR
MOV LCD_DATA,R0 ; SENDS TO PORT 0
CLR LCD_RS ; SETS RS LINE HIGH FOR DATA
CALL PULSE_E ; CLOCKS DATA INTO LCD
RET

The BOTHLINES function prints the text across the two lines.

The LINE1 and LINE2 functions move the cursor position.

The WRITE and SEND_CHAR functions write the text one character at a time on the LCD screen.

Thus, the code to write the text on the LCD screen is:

MOV DPTR,#TEXT0 ; Welcome message stored
; in code memory
LCALL BOTHLINES
SAMPLE PROGRAMS
Accompanying this application note is assembly code containing the LCD functions previously used in this program.

Two sample programs are available that use the LCD interface described in this document. The first program is an LCD temperature monitor. This program uses the on-chip temperature sensor of the ADuC834/ADuC824/ADuC816 and displays the die temperature on the LCD screen. A program flowchart is shown in Figure 2. The second program implements a frequency measurement using the Timer 2 input pin on the ADuC834/ADuC824/ADuC816, although the code can easily be ported to other MicroConverter products. This program is described in greater detail in the AN-644 (uC013) application note.

Figure 2. Flowchart for Temperature Monitor Routine