Interfacing LCD with 8051

LCD INTERFACING

LCD Pin Descriptions

Pin Descriptions for LCD

Pin	Symbol	I/O	Descriptions
1	VSS	--	Ground
2	VCC	--	+5V power supply
3	VEE	--	Power supply to control contrast
4	RS	I	RS=0 to select command register, RS=1 to select data register
5	R/W	I	R/W=0 for write, R/W=1 for read
6	E	I/O	Enable
7	DB0	I/O	The 8-bit data bus
8	DB1	I/O	The 8-bit data bus
9	DB2	I/O	The 8-bit data bus
10	DB3	I/O	The 8-bit data bus
11	DB4	I/O	The 8-bit data bus
12	DB5	I/O	The 8-bit data bus
13	DB6	I/O	The 8-bit data bus
14	DB7	I/O	The 8-bit data bus

used by the LCD to latch information presented to its data bus

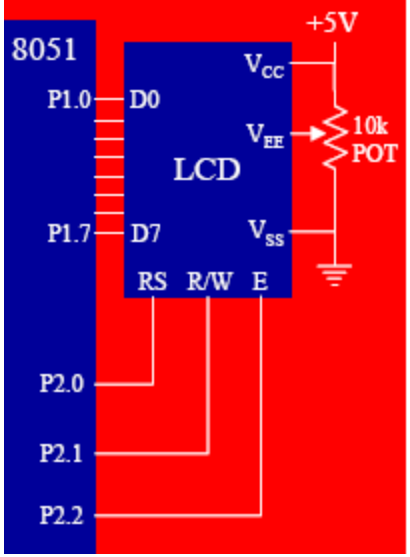
- Send displayed information or instruction command codes to the LCD
- Read the contents of the LCD's internal registers

LCD Command Codes

Code (Hex)	Command to LCD Instruction Register
1	Clear display screen
2	Return home
4	Decrement cursor (shift cursor to left)
6	Increment cursor (shift cursor to right)
5	Shift display right
7	Shift display left
8	Display off, cursor off
A	Display off, cursor on
C	Display on, cursor off
E	Display on, cursor blinking
F	Display on, cursor blinking
10	Shift cursor position to left
14	Shift cursor position to right
18	Shift the entire display to the left
1C	Shift the entire display to the right
80	Force cursor to beginning to 1st line
C0	Force cursor to beginning to 2nd line
38	2 lines and 5x7 matrix

LCD INTERFACING

Sending Data/ Commands to LCDs w/ Time Delay

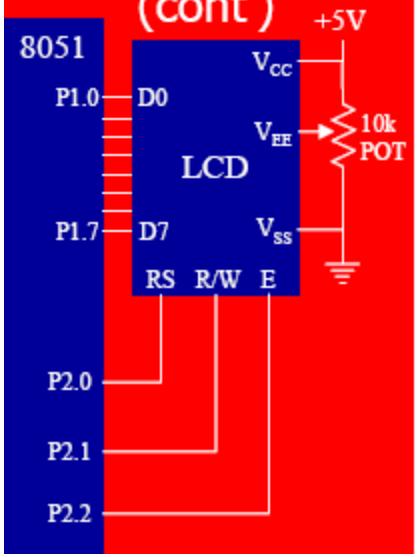


To send any of the commands to the LCD, make pin RS=0. For data, make RS=1. Then send a high-to-low pulse to the E pin to enable the internal latch of the LCD. This is shown in the code below.

```
;calls a time delay before sending next data/command
;P1.0-P1.7 are connected to LCD data pins D0-D7
;P2.0 is connected to RS pin of LCD
;P2.1 is connected to R/W pin of LCD
;P2.2 is connected to E pin of LCD
        ORG   0H
        MOV   A,#38H  ;INIT. 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 some time
        MOV   A,#06H  ;shift cursor right
        ACALL COMNWRT ;call command subroutine
        ACALL DELAY   ;give LCD some time
        MOV   A,#84H  ;cursor at line 1, pos. 4
        ACALL COMNWRT ;call command subroutine
        ACALL DELAY   ;give LCD some time
.....
```

LCD INTERFACING

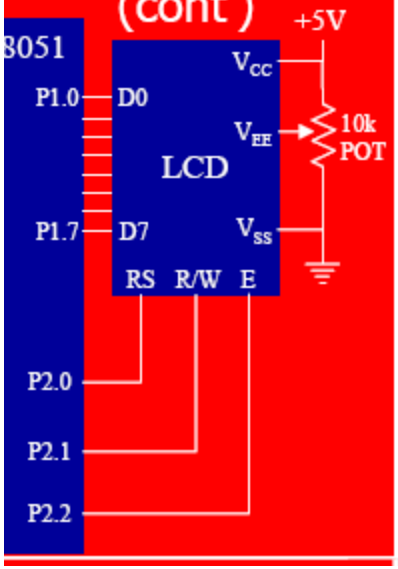
Sending Data/ Commands to LCDs w/ Time Delay (cont')




```
.....
        MOV   A,#'N'    ;display letter N
        ACALL DATAWRT   ;call display subroutine
        ACALL DELAY     ;give LCD some time
        MOV   A,#'O'     ;display letter O
        ACALL DATAWRT   ;call display subroutine
AGAIN:  SJMP  AGAIN     ;stay here
COMNWRT:                ;send command to LCD
        MOV   P1,A      ;copy reg A to port 1
        CLR   P2.0      ;RS=0 for command
        CLR   P2.1      ;R/W=0 for write
        SETB  P2.2      ;E=1 for high pulse
        ACALL DELAY     ;give LCD some time
        CLR   P2.2      ;E=0 for H-to-L pulse
        RET
DATAWRT:                ;write data to LCD
        MOV   P1,A      ;copy reg A to port 1
        SETB  P2.0      ;RS=1 for data
        CLR   P2.1      ;R/W=0 for write
        SETB  P2.2      ;E=1 for high pulse
        ACALL DELAY     ;give LCD some time
        CLR   P2.2      ;E=0 for H-to-L pulse
        RET
DELAY:  MOV   R3,#50    ;50 or higher for fast CPUs
HERE2:  MOV   R4,#255   ;R4 = 255
HERE:   DJNZ  R4,HERE   ;stay until R4 becomes 0
        DJNZ  R3,HERE2
        RET
        END
```

LCD INTERFACING

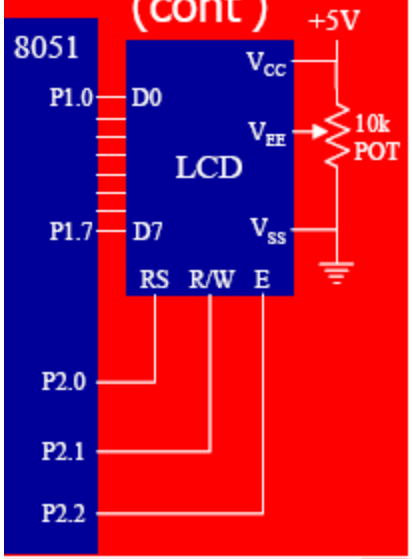
Sending Data/ Commands to LCDs w/ Time Delay (cont')




```
;Check busy flag before sending data, command to LCD
;p1=data pin
;P2.0 connected to RS pin
;P2.1 connected to R/W pin
;P2.2 connected to E pin
      ORG    0H
      MOV    A,#38H        ;init. LCD 2 lines ,5x7 matrix
      ACALL  COMMAND       ;issue command
      MOV    A,#0EH        ;LCD on, cursor on
      ACALL  COMMAND       ;issue command
      MOV    A,#01H        ;clear LCD command
      ACALL  COMMAND       ;issue command
      MOV    A,#06H        ;shift cursor right
      ACALL  COMMAND       ;issue command
      MOV    A,#86H        ;cursor: line 1, pos. 6
      ACALL  COMMAND       ;command subroutine
      MOV    A,#'N'        ;display letter N
      ACALL  DATA_DISPLAY
      MOV    A,#'O'        ;display letter O
      ACALL  DATA_DISPLAY
HERE:SJMP   HERE           ;STAY HERE
.....
```

LCD INTERFACING

Sending Codes and Data to LCDs w/ Busy Flag (cont')




```
.....
COMMAND:
      ACALL READY          ;is LCD ready?
      MOV   P1,A           ;issue command code
      CLR   P2.0           ;RS=0 for command
      CLR   P2.1           ;R/W=0 to write to LCD
      SETB  P2.2           ;E=1 for H-to-L pulse
      CLR   P2.2           ;E=0,latch in
      RET
DATA_DISPLAY:
      ACALL READY          ;is LCD ready?
      MOV   P1,A           ;issue data
      SETB  P2.0           ;RS=1 for data
      CLR   P2.1           ;R/W =0 to write to LCD
      SETB  P2.2           ;E=1 for H-to-L pulse
      CLR   P2.2           ;E=0 latch in
      RET
READY:
      SETB  P1.7           ;make P1.7 input port
      CLR   P2.0           ;RS=0 access command reg
      SETB  P2.1           ;R/W=1 read command reg
;read command reg and check busy flag
BACK:SETB  P2.2            ;E=1 for H-to-L pulse
      CLR   P2.2           ;E=0 H-to-L pulse
      JB    P1.7,BACK      ;stay until busy flag=0
      RET
      END
```

To read the command register, we make R/W=1, RS=0, and a H-to-L pulse for the E pin.

If bit 7 (busy flag) is high, the LCD is busy and no information should be issued to it.

The data pins of an LCD are connected to P1. The information is latched into the LCD whenever its Enable pin goes from high to low. Write an 8051 C program to send "The Earth is but One Country" to this LCD.

Solution:

```
#include <reg51.h>
#define LCDData P1   //LCDData declaration
sbit En=P2^0;        //the enable pin

void main(void)
  {
    unsigned char message[]
              ="The Earth is but One Country";
    unsigned char z;
    for (z=0;z<28;z++) //send 28 characters
       {
         LCDData=message[z];
         En=1;     //a high-
         En=0;     //-to-low pulse to latch data
       }
  }
```