Interrupt Priorities

The 8051 offers two levels of interrupt priority: high and low. By using interrupt priorities you may assign higher priority to certain interrupt conditions. For example, you may have enabled Timer 1 Interrupt which is automatically called every time Timer 1 overflows. Additionally, you may have enabled the Serial Interrupt which is called every time a character is received via the serial port. However, you may consider that receiving a character is much more important than the timer interrupt. In this case, if Timer 1 Interrupt is already executing you may wish that the serial interrupt itself interrupts the Timer 1 Interrupt. When the serial interrupt is complete, search control passes back to Timer 1 Interrupt and finally back to the main program. You may accomplish this by assigning a high priority to the Serial Interrupt and a low priority to the Timer 1 Interrupt.

What Happens When an Interrupt Occurs?

When an interrupt is triggered, the following actions are taken automatically by the microcontroller:

- The current Program Counter is saved on the stack, low-byte first.
- Interrupts of the same and lower priority are blocked.
- In the case of Timer and External interrupts, the corresponding interrupt flag is set.
- Program execution transfers to the corresponding interrupt handler vector address.
- The Interrupt Handler Routine executes. Take special note of the third step: If theinterrupt being handled is a Timer or External interrupt, the microcontroller automatically clearsthe interrupt flag before passing search control to your interrupt handler routine.

What Happens When an Interrupt Ends?

An interrupt ends when your program executes the RETI instruction. When the RETI instruction is executed the following actions are taken by the microcontroller:

- Two bytes are popped off the stack into the Program Counter to restore normal program execution.
- Interrupt status is restored to its pre-interrupt status.

LCD (LIQUID CRYSTAL DISPLAY) INTERFACE

LCDs can display numbers, characters, and graphics. To produce a proper display,the information has to be periodically refreshed. This can be done by the CPU or internally by the LCD device itself. Incorporating a refreshing controller into the LCD, relieves the CPU of this task and hence many LCDs have built-in controllers. These controllers also facilitate flexible programming for characters and graphics. Table 5.1 shows the pin description of an LCD. from Optrex.

Pin no.	Symbol	External connection	Function
1	Vss	Power supply	Signal ground for LCM
2	V_{DD}		Power supply for logic for LCM
3	V_0		Contrast adjust
4	RS	MPU	Register select signal
5	R/W	MPU	Read/write select signal
6	E	MPU	Operation (data read/write) enable signal
7~10	DB0~DB3	MPU	Four low order bi-directional three-state data bus lines. Used for data transfer between the MPU and the LCM. These four are not used during 4-bit operation.
11~14	DB4~DB7	MPU	Four high order bi-directional three-state data bus lines. Used for data transfer between the MPU

Table 5.4.1 Pin description of LCD

- Vss and VDD provide +5v and ground, V0 is used for controlling LCD contrast.
- If RS=0, the instruction command register is selected, allowing the user to send acommand such as clear display, cursor at home, etc.
- If RS=1 the data register is selected, allowing the user to send data to be displayedon the LCD.
- R/W input allows the user to Read/ Write the information to the LCD.
- The enable pin is used by the LCD to latch information presented to its data pins.
- The 8-bit data pins are used to send information to LCD.

LCD COMMAND CODES

The LCD's internal controller can accept several commands and modify thedisplay accordingly. These commands would be things like:

✓Clear screen

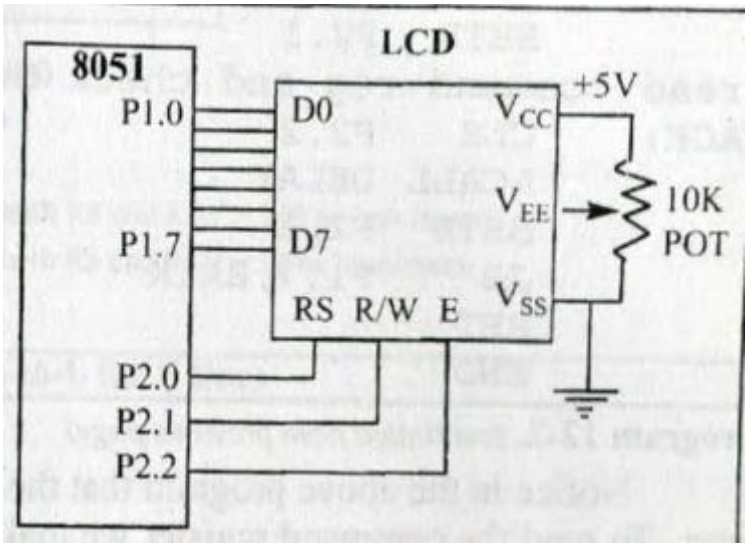
✓Return home

✓Decrement/Increment cursor

After writing to the LCD, it takes some time for it to complete its internal operations. During this time, it will not accept any new commands or data. Figure 5.4.1 shows the command codes of LCD and Figure 5.4.2 shows the LCD interfacing. We need to inserta time delay between any two commands or data sent to LCD.

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 Command Codes



LCD Connections to 8051

KEYBOARD INTERFACING WITH 8051

4X 4 KEYBOARD

Figure 5.4.31 shows a 4 x4 matrix connected to two ports.

- The rows are connected to an output port(Port 1) and the columns are connected toan input port. (Port 2)
- If no key has been pressed, reading the input port will yield 1s for all columns sincethey are all connected to high (Vcc).
- If all the rows are grounded and a key is pressed, one of the columns will have 0since the key pressed provides the path to ground.
- It is the function of the microcontroller to scan the keyboard continuously to detectand identify the key pressed.

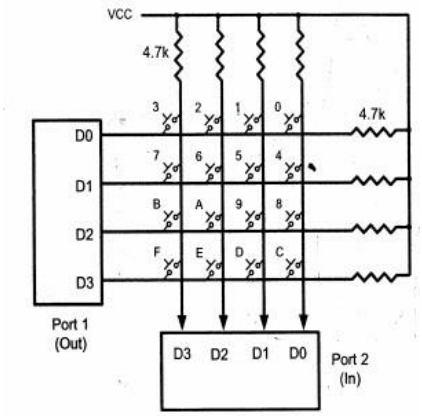


Figure 5.4.3 Matrix Keyboard Connections to Ports

[Source: "The 8051Microcontroller and Embedded Systems: Using Assembly and C" by Mohamed Ali Mazidi,Janice Gillispie Mazidi, Rolin McKinlay]