BECE204P-Microprocessors & Microcontrollers Lab

LAB-11

INTERRUPT PROGRAMMING & LCD INTERFACING WITH 8051

INTERRUPT

- Interrupt is an event that temporarily suspends the main program, passes the control to a subroutine program and resumes the main program flow where it had left off.

- The 8051 microcontroller can recognize six different events that cause the main program to interrupt from the normal execution.

Interrupt	Flag	Interrupt vector address
Reset	-	0000H
INT0 (P3.2)	IE0	0003H
Timer 0	TF0	000BH
INT1 (P3.3)	IE1	0013H
Timer 1	TF1	001BH
Serial	TI/RI	0023H

- Vector Address: This is the address where the controller jumps after the interrupt to serve the ISR (interrupt service routine).

INTERRUPT

- **Reset:** It is the highest priority interrupt, upon reset 8051 microcontroller start executing code from 0x0000 address.
- **Timer Interrupts:** Whenever timer overflows, timer overflow flags (TF0/TF1) are set.
- **External Interrupt:** Pin 12 (P3.2) and pin 13 (P3.3) of the 8051, designated as INT0 and INT1, are used as external hardware interrupts
- **Serial interrupt(TI/SI):** In serial communication,
  - The last bit (stop bit) of a byte is transmitted, the TI serial interrupt flag is set
  - When the last bit (stop bit) of the receiving data byte is received, the RI flag set
- **Interrupt Enable (IE) Register:**
  - It is a bit addressable register responsible for enabling and disabling the interrupt.
  - In IE, bit corresponding to 1 activates the interrupt and 0 disables the interrupt.
  - The corresponding bit in this register enables particular interrupt

INTERRUPT

INTERRUPT ENABLE (IE) REGISTER

EA	--	ET2	ES	ET1	EX1	ET0	EX0

BIT	NAME	DESCRIPTION
7	EA	Enable All must be set to 1 in order activate each interrupt given in the register
6	--	Reserved for future use
5	ET2	Enable/Disable Timer 2 overflow interrupt (for 8952)
4	ES	Enable/Disable Serial port interrupt
3	ET1	Enable/Disable Timer 1 overflow interrupt
2	EX1	Enable/Disable eXternal interrupt 1
1	ET0	Enable/Disable Timer 1 overflow interrupt
0	EX0	Enable/Disable eXternal interrupt 0

LAB TASK-1

Write a program using interrupts that continuously get 8-bit data from P0 and sends it to P1 while simultaneously creating a square wave of 200 µs period on pin P2.1. Use timer 0 to create the square wave. Assume that XTAL = 11.0592 MHz.

Delay calculation: Use timer 0 in mode 2 (auto reload). TH0 = 100/1.085 us = 92 =A4H

```
;--upon wake-up go to main, avoid using memory allocated to Interrupt Vector Table
            ORG 0000H
            LJMP MAIN                   ;by-pass interrupt vector table


;--ISR for timer 0 to generate square wave
            ORG 000BH                   ;Timer 0 interrupt vector table
            CPL P2.1                    ;toggle P2.1 pin
            RETI                        ;return from ISR
```

LAB TASK-1

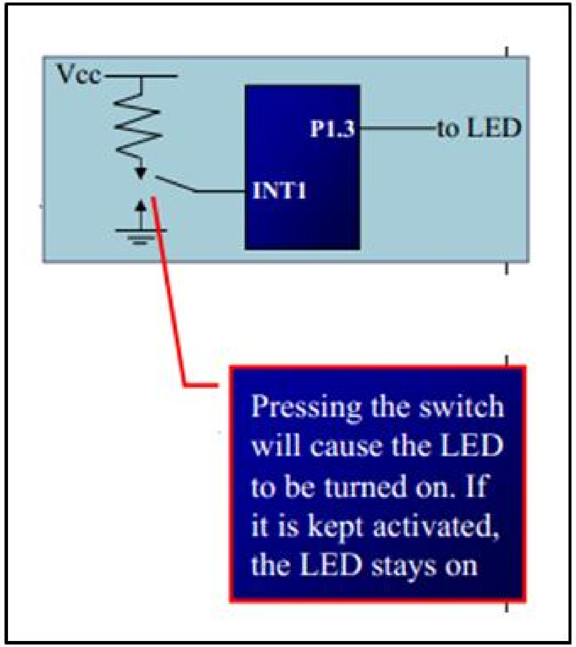
```
;--The main program for initialization
            ORG 0030H            ;after vector table space
MAIN:       MOV P0,#0FFH         ;make P0 an input port
            MOV P1,#00H          ;make P0 an output port
            CLR P2.1             ;make P2.1 as output pin
            MOV TMOD,#02H        ;Timer 0, mode 2
            MOV TH0,#0A4H        ;TH0=A4H for -92
            MOV IE,#82H          ;IE=10000010(bin) enable Timer 0
            SETB TR0             ;Start Timer 0
BACK:       MOV A,P0             ;get data from P0
            MOV P1,A             ;issue it to P1
            SJMP BACK            ;keep doing it loop unless interrupted by TF0
            END
```

LAB TASK-2

Assume that the INT1 pin is connected to a switch that is normally high. Whenever it goes low, it should turn on an LED. The LED is connected to P1.3 and is normally off. As long as the switch is pressed low, the LED should stay on. Simultaneously perform a toggle operation in P1.5 with the delay of 500ms.

```
        ORG 0000H
        LJMP MAIN

//ISR for INT1
        ORG 0013H
        SETB P1.3
        RETI
```



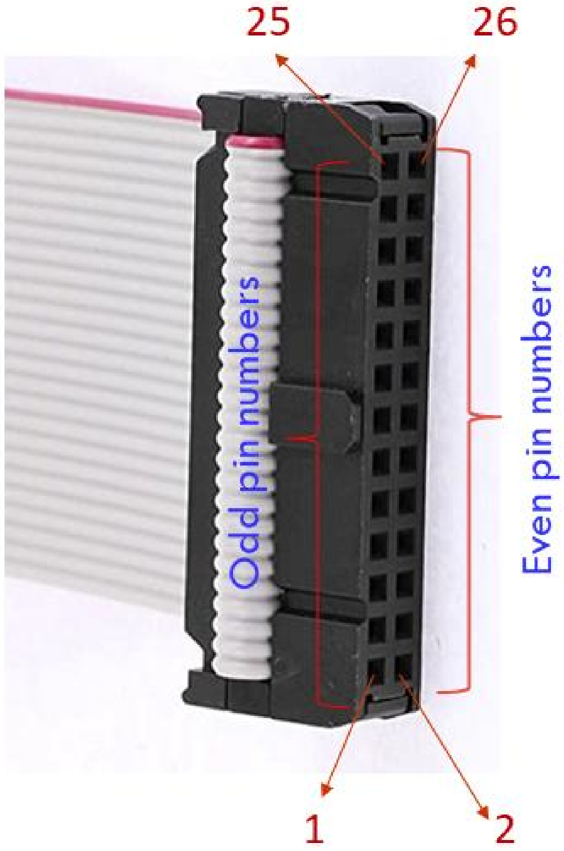
Pressing the switch will cause the LED to be turned on. If it is kept activated, the LED stays on

LAB TASK-2

```
        ORG 30H
MAIN:   SETB P3.3
        CLR P1.3
        CLR P1.5
        MOV IE,#10000100B
HERE:   CLR P1.3
        SETB P1.5
        ACALL DELAY
        CLR P1.5
        ACALL DELAY
        SJMP HERE
```

```
//Delay of 500ms
DELAY:  MOV R2,#04H        ;LOAD R2 WITH 07 HEX
HERE3:  MOV R1,#0FFH       ;LOAD R1 WITH 0FF HEX
HERE2:  MOV R0,#0FFH       ;LOAD R2 WITH 0FF HEX
HERE1:  DJNZ R0,HERE1      ;DECREMENT R0
        DJNZ R1,HERE2      ;DECREMENT R1
        DJNZ R2,HERE3      ;DECREMENT R2
        RET                ;RETURN
        END
```

ESA 8051 MICROCONTROLLER KIT PIN DETAILS



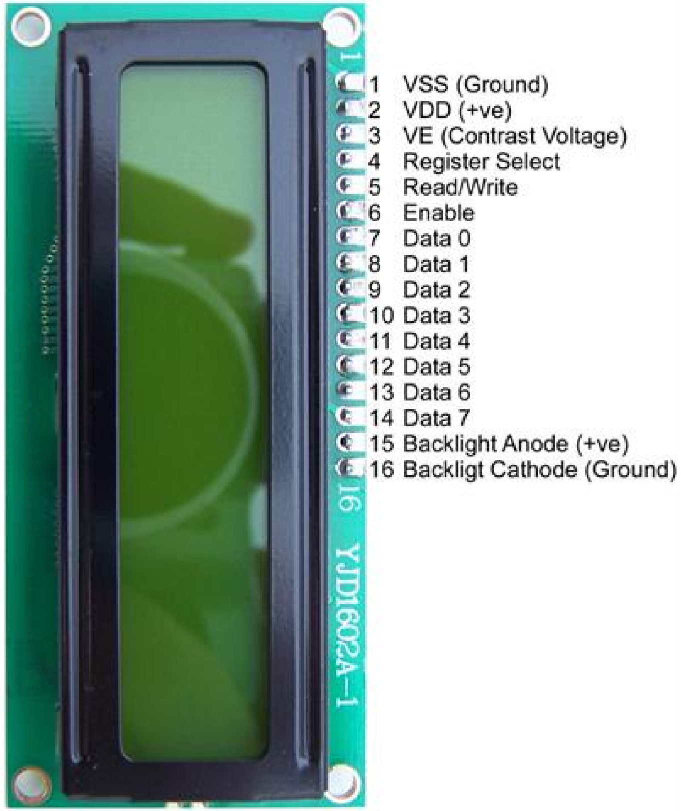
J7: 26-Pin Male Connector for Ports P0, P1 and P2

Pin No. on J7	PORT Line	Pin No. on J7	PORT Line
1	P2.4	2	P2.5
3	P2.2	4	P2.3
5	P2.0	6	P2.1
7	P1.6	8	P1.7
9	P1.4	10	P1.5
11	P1.2	12	P1.3
13	P1.0	14	P1.1
15	P0.6	16	P0.7
17	P0.4	18	P0.5
19	P0.2	20	P0.3
21	P0.0	22	P0.1
23	P2.6	24	P2.7
25	Vcc	26	Gnd

LIQUID CRYSTAL DISPLAY (LCD)

- Display units are the most important output devices in many electronics products and LCD is one of the most used display unit in many applications.
- LCD is composed of liquid crystal particles which do not emit light on their own instead they are illuminated by a backlight hence they need an external light source to work.
- In 16x2 LCD, 2 represents number of lines(row) and 16 represents number of characters (column) displayed in each line.
- Each character in LCD is displayed in a matrix of 5x7 pixels.
- It supports all the ASCII characters and is basically used for displaying the alpha numeric characters.
- It also provides the provision to display the custom characters by creating the pattern.

LIQUID CRYSTAL DISPLAY (LCD)



16x2 LCD

PIN NO.	NAME	FUNCTION
1	VSS	Ground pin
2	VCC	Power supply pin of 5V
3	VEE	Used for adjusting the contrast commonly attached to the potentiometer.
4	RS	RS is the register select pin used to write display data to the LCD (characters), this pin has to be high when writing the data to the LCD. During the initializing sequence and other commands this pin should low.
5	R/W	Reading and writing data to the LCD for reading the data R/W pin should be high (R/W=1) to write the data to LCD R/W pin should be low (R/W=0)
6	E	Enable pin is for starting or enabling the module. A high to low pulse of about 450ns pulse is given to this pin.
7	DB0	DB0-DB7 Data pins for giving data (normal data like numbers Characters or command data) which is meant to be displayed
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	
15	LED+	Back light of the LCD which should be connected to Vcc
16	LED-	Back light of LCD which should be connected to ground.

LIQUID CRYSTAL DISPLAY (LCD)

- The 16X2 LCD has two built in registers namely data register and command register.

- Command Register - stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing, clearing the screen, setting the cursor position, controlling display etc.

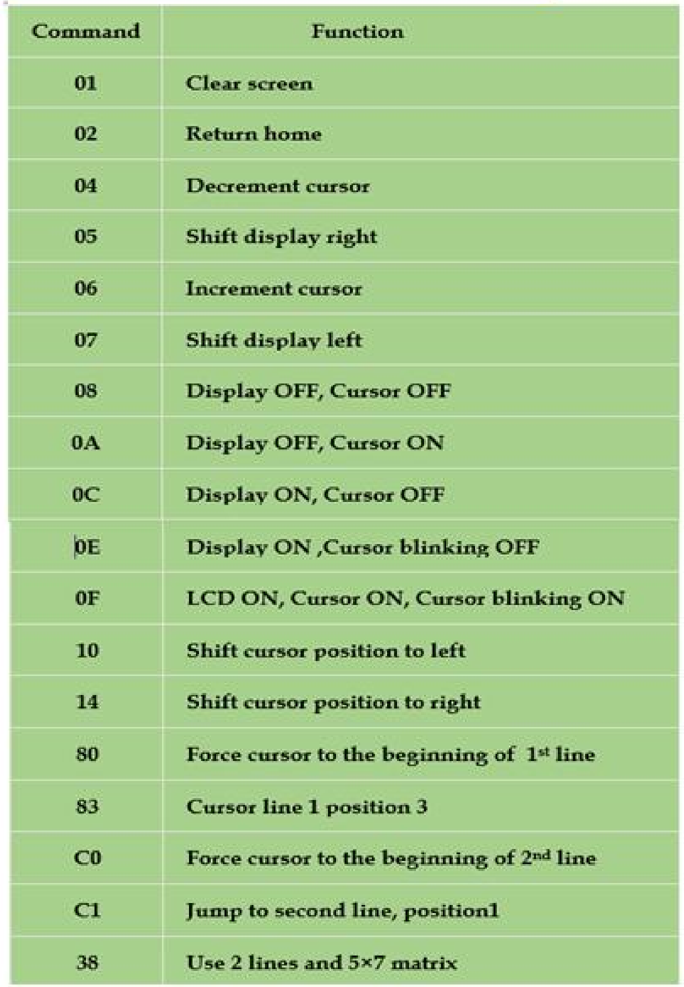
- Data Register - stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

- For programming LCD follow these steps:

  STEP1: Initialization of LCD.

  STEP2: Sending command to LCD.

  STEP3: Writing the data to LCD.



Command	Function
01	Clear screen
02	Return home
04	Decrement cursor
05	Shift display right
06	Increment cursor
07	Shift display left
08	Display OFF, Cursor OFF
0A	Display OFF, Cursor ON
0C	Display ON, Cursor OFF
0E	Display ON ,Cursor blinking OFF
0F	LCD ON, Cursor ON, Cursor blinking ON
10	Shift cursor position to left
14	Shift cursor position to right
80	Force cursor to the beginning of 1st line
83	Cursor line 1 position 3
C0	Force cursor to the beginning of 2nd line
C1	Jump to second line, position1
38	Use 2 lines and 5×7 matrix

LIQUID CRYSTAL DISPLAY (LCD)

Step1: LCD initialization (common for almost all LCD applications)

1. Send 38H to the 8 bit data line for initialization
2. Send 0FH for making LCD ON, cursor ON and cursor blinking ON.
3. Send 06H for incrementing cursor position.
4. Send 01H for clearing the display and return the cursor.

Step2: Sending command to LCD

1. Send the command data to command register
2. Make R/W low.
3. Make RS=0 if data byte is a command
4. Pulse E from high to low with some delay.
5. Repeat above steps for sending another command.

Step3: Writing the data to LCD

1. Place data byte on the data register.
2. Make R/W low.
3. make RS=1 if the data byte is a data to be displayed.
4. Pulse E from high to low with some delay.
5. Repeat above steps for sending another data.

LAB TASK-3

Write an 8051 assembly language program to display the message "your reg. no" (example:21BEC1001) on LCD display using DPTR. Assume P2.0-P2.7=D0-D7, P3.7=RS, P3.6=R/W, P3.5=E.

```
                ORG 0030H
                MOV DPTR, #MYCOM
C1:             CLR A
                MOVC A,@A+DPTR
                ACALL COMNWRT
                ACALL DELAY
                INC DPTR
                JZ SEND_DAT
                SJMP C1
```

LAB TASK-3

```
SEND_DAT:   MOV DPTR, #MYDATA
D1:         CLR A
            MOVC A,@A+DPTR
            ACALL DATAWRT
            ACALL DELAY
            INC DPTR
            JZ AGAIN
            SJMP D1
AGAIN:      SJMP AGAIN
COMNWRT:    MOV P2, A             ; send command to LCD by coping A to P1
            CLR P3.7              ; RS=0 for command
            CLR P3.6              ; R/W=0 for write
            SETB P3.5             ; E=1  for high pulse
            ACALL DELAY           ; give LCD some time
            CLR P3.5              ; E=0 for H-to-L pulse
            RET
```

LAB TASK-3

```
DATAWRT:    MOV P2, A           ; write data to LCD by coping A into P1
            SETB P3.7           ; RS=1 for data
            CLR P3.6            ; R/W=0 for write
            SETB P3.5           ; E=1 for high pulse
            ACALL DELAY         ; give LCD some time
            CLR P3.5            ; E=0 for H-to-L pulse
            RET
DELAY:      MOV R3, #250        ; 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
ORG 300H
MYCOM:      DB 38H, 0EH, 01, 06, 84H, 0         ; commands and null
MYDATA:     DB "21BEC1001", 0
END
```