8051_ INTERRUPTS PROGRAMMING

* There are 6 interrupts in 8051.

1. Reset
2. Interrupt for Timer 0
3. Interrupt for Timer 1
4. External hardware interrupt 0 (INT0)
5. External hardware interrupt 1 (INT1)
6. Serial communication interrupt (RI & TI)

Interrupt Vector Table for 8051

Interrupt	ROM location	Pin No.
Reset	0000_H	9
INT0	0003_H	P3.2 (12)
INT1	0013_H	P3.3 (13)
Timer 0 interrupt (TF0)	$000B_H$	–
Timer 1 interrupt (TF1)	$001B_H$	–
Serial COM interrupt (RI & TI)	0023_H	–

Enabling and Disabling an interrupt:

IE register (Interrupt Enable Register) is responsible for enabling and disabling the interrupts.

D7							D0
EA	–	ET2	ES	ET1	EX1	ET0	EX0

EA = 0, disables all interrupts.

EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit

ET2 :- Enables or disables timer 2 overflow interrupt (only applicable for 8952)

ES – Enables or disables the serial port interrupt.

ET1 – Enables or disables timer 1 overflow interrupt.

EX1 – Enables or disables External interrupt 1.

ET0 – Enables or disables timer 0 overflow interrupt.

EX0 – Enables or disables External interrupt 0.

Programming External Hardware Interrupts :-

To make INT0, INT1 edge-triggered interrupts, bit of TCON register must be programmed.

D_7							D_0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

IT0, IT1 : Interrupt 0, 1 type control bits.

If 0, low-level triggered interrupt

If 1, edge-triggered interrupt.

IE0, IE1 : Interrupt-in-service flags. (Edge flag)

These bits are set by CPU when an external interrupt edge is detected. Cleared when interrupt is processed.

Interrupt Priority in 8051 :-

Interrupt Priority upon reset :

Highest Priority

↓

Lowest Priority

External interrupt 0 – INT0

Timer interrupt 0 – TF0

External interrupt 1 – INT1

Timer interrupt 1 – TF1

Serial Communication – RI & TI

6. Setting Interrupt Priority with the IP register:

We can alter the sequence of the default priority table by means of IP register.

IP (Interrupt Priority Register)

D7 ... D0

–	–	PT2	PS	PT1	PX1	PT0	PX0

When any of the above bit is set high, the corresponding interrupt will be given the highest priority.

When two or more bits are set to high, interrupts are serviced according to the sequence of the table.

Eg: a) Program the IP register to assign the highest priority to INT1.

Solution:
MOV IP, # 00000100B (or)
MOV IP, # 04H (or)
SETB IP.2

Eg.b) Show the order of interrupt priority if INT0, INT1 & TF0 are activated at the same time.

INT1
INT0
TF0

Eg. c) Show the instruction to enable the serial interrupt, timer 0 interrupt, & ext. h/w int 1.

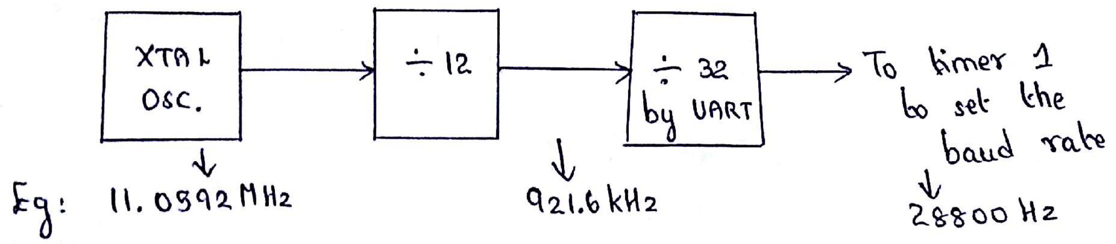
MOV IE, # 10010110B (or)
MOV IE, # 96H

Eg. d) Disable all the interrupts with a single instruction

CLR IE.7

8051 Serial Communication Programming

* For serial data communication, the byte of data must be converted to serial bits using shift register, then it can be transmitted over a single data line.
* **Baud Rate** is the number of bits transmitted per second.
* The commonly used PC Baud rates are 1200, 2400, 4800, 9600, 19200.
* Different baud rates can be achieved in 8051 with the help of timer 1 in mode 2.



* To get the desired baud rate, TH1 must be loaded with the following values:

Baud Rate	TH1 (Hex)	TH1 (Decimal)
9600	FD	−3
4800	FA	−6
2400	F4	−12
1200	E8	−24

Serial Data Buffer (SBUF) Register:

* SBUF is an 8-bit register.
* The data to be transmitted serially via TxD line, must be placed in SBUF.
* The serial data received via RxD line, will be placed in SBUF.

To access SBUF:

Eg.
```
MOV SBUF, A
MOV A, SBUF
MOV SBUF, #'D'
```

Serial Port Control Register (SCON):-

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0, SM1 :

SM0	SM1	
0	0	- Serial Mode 0
0	1	- Serial Mode 1 (8-bit data, 1 stop bit, 1 start bit)
1	0	- Serial Mode 2
1	1	- Serial Mode 3

SM2 : Enables the multiprocessor capability of 8051 (Make it 0)

REN : Receive Enable

TB8 : Transmitted 9th bit

RB8 : Received 9th bit

TI : Transmit Interrupt Flag

After finishing the transfer of 8-bit character, TI flag is raised high.

RI : After receiving a 8-bit character, RI flag is raised high.

Programming the 8051 to transfer data serially:

1. TMOD register is loaded with 20_H, indicating the use of timer 1 in mode 2 to set the baud rate.
2. TH1 is loaded with appropriate value to set the baud rate.
3. SCON register is loaded with 50_H indicating serial mode 1.
4. Timer 1 has to be started. (SETB TR1)
5. TI is cleared. (CLR TI)
6. The character byte has to written into SBUF.
7. TI flag bit is monitored (JNB TI, label)
8. To transfer the next character, repeat from step 5.

Eg: Write a program for 8051 to transfer letter "A" serially at 4800 baud continuously.

```
        MOV   TMOD, #20H
        MOV   TH1, #FAH
        MOV   SCON, #50H
        SETB  TR1
AGAIN:  MOV   SBUF, #"A"
HERE:   JNB   TI, HERE
        CLR   TI
        SJMP  AGAIN
```

Programming the 8051 to receive data serially:

1. TMOD register is loaded with 20_H, indicating the use of timer 1 in mode 2 to set the baud rate.
2. TH1 is loaded with appropriate value to set the baud rate.

3. SCON register is loaded with 50_H indicating serial mode 1.
4. Timer 1 has to be started. (SETB TR1)
5. RI is cleared. (CLR RI)
6. Monitor RI flag bit continuously.
7. When RI flag is raised, SBUF has the received byte.
8. To receive the next character, go to step 5.

Eg: Program the 8051 to receive bytes of data serially and put them in P1. Set the baud rate at 4800, 8-bit data and 1 stop-bit.

```
        MOV    TMOD, #20H
        MOV    TH1, #FAH
        MOV    SCON, #50H
        SETB   TR1
HERE:   JNB    RI, HERE
        MOV    A, SBUF
        MOV    P1, A
        CLR    RI
        SJMP   HERE
```

Doubling the baud rate in 8051:

Baud rate can be doubled by setting the SMOD bit of PCON register.

PCON (Power Control Register)

D7	D6	D5	D4	D3	D2	D1	D0
SMOD	-	-	-	GF1	GF0	PD	IDL

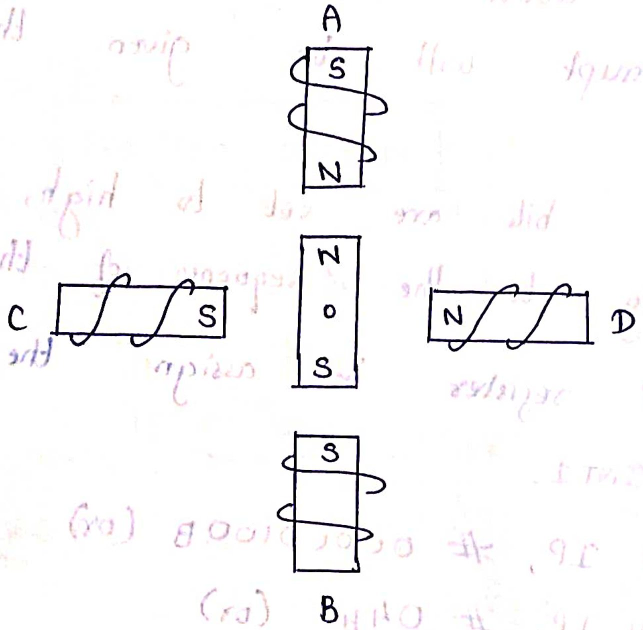
If SMOD = 1, baud rate will be doubled.

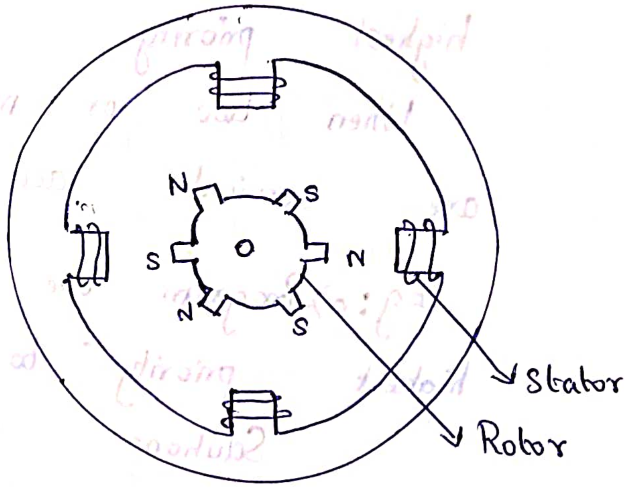
Eg:

TH1	SMOD = 0	SMOD = 1
FD_H	9600	19,200
FA_H	4800	9,600

STEPPER MOTOR INTERFACING

* Stepper Motor is a widely used device that translates electrical pulses into mechanical movement.
* Stepper Motor is used in applications such as disk drives, dot matrix printers, robotics etc. for position control.
* stepper motor has a permanent magnet rotor surrounded by a stator.





* Most motors have 4 stator windings referred as a 4-phase stepper motor.
* When a sequence of power is applied to stator, rotor will rotate.

Normal 4-step sequence:

Step #	Winding A	B	C	D
1	1	0	0	1
2	1	1	0	0
3	0	1	1	0
4	0	0	1	1

Clockwise

Counter-clockwise

Step Angle: It is defined as the minimum degree of rotation associated with a single step.

Steps per Revolution: It is the total no. of steps needed to rotate one complete rotation or 360 degrees.

$$\text{No. of steps per revolution} = \frac{360^\circ}{\text{step angle}}$$

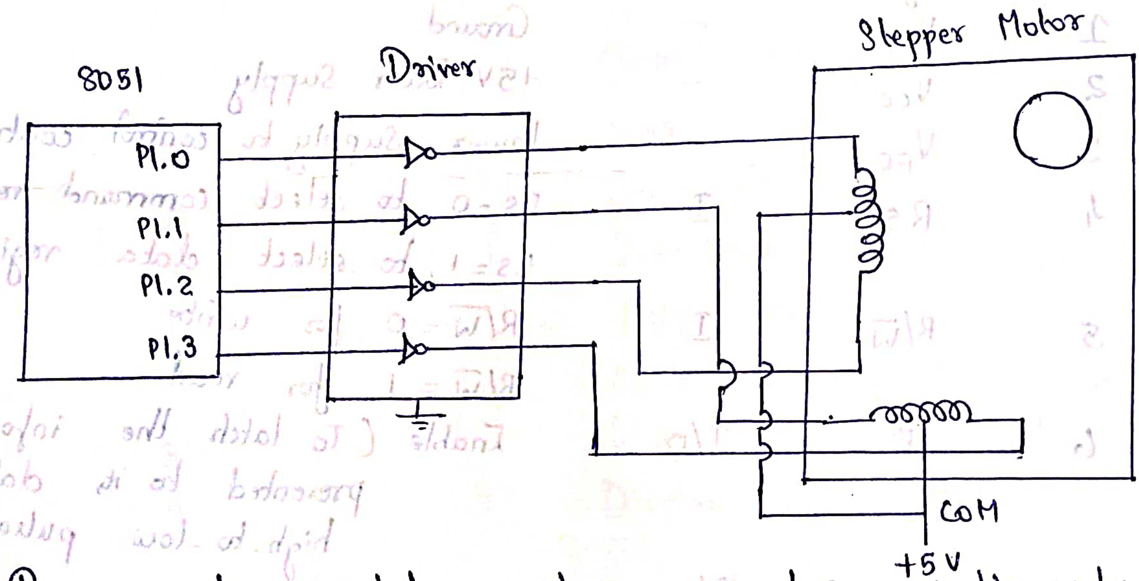
Eg: If Step angle = 2°,

Steps Per Revolution = 180

$$\text{Steps Per Second} = \frac{\text{RPM} \times \text{steps per revolution}}{60}$$

where RPM is revolutions per minute.

8051 interfacing with stepper motor:



Program to rotate stepper motor continuously:

```
START: MOV   R0, #04
       MOV   DPTR, #TABLE
NEXT:  MOVX  A, @DPTR
       MOV   P1, A
       ACALL DELAY
       INC   DPTR
       DJNZ  R0, NEXT
       SJMP  START
```

```
TABLE: DB   09   0C   06   03

DELAY:  MOV  R2, # 64H
L1:     MOV  R3, # FFH
L2:     DJNZ R3, L2
        DJNZ R2, L1
        RET
```

LCD Interfacing

* LCD (Liquid Crystal Display) has the ability to display numbers, characters and graphics.

LCD pin discription:

Pin	Symbol	I/O	Description
1	V_{SS}	–	Ground
2	V_{CC}	–	+5V Power Supply
3	V_{EE}	–	Power Supply to control contrast
4	RS	I	RS = 0, to select command register RS = 1, to select data register
5	$R/\overline{W}$	I	$R/\overline{W} = 0$ for write $R/\overline{W} = 1$ for read
6	E	I/O	Enable (To latch the information presented to its data pins, high-to-low pulse)
7	DB0	I/O	8-bit data bus (Bidirectional)
8	DB1	I/O	
9	DB2	I/O	
10	DB3	I/O	
11	DB4	I/O	
12	DB5	I/O	
13	DB6	I/O	
14	DB7	I/O	

Programming 8051 Timers

* The 8051 has 2 timers/counters. (Timer 0 & Timer 1).
* They can be used either as timers to generate a time delay or as counters to count events happening outside the microcontroller.

Basic Registers of the timers:

The 16-bit register of Timer 0 (T0) and Timer 1 (T1) can be accessed as two separate 8-bit registers : Low byte (TL) and High byte (TH).

Timer 0 register:

TH0								TL0							
D_{15}	D_{14}	D_{13}	D_{12}	D_{11}	D_{10}	D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

Timer 1 register:

TH1								TL1							
D_{15}	D_{14}	D_{13}	D_{12}	D_{11}	D_{10}	D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

TMOD (Timer Mode) Register:

MSB ... LSB

Timer 1				Timer 0			
GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0

M1, M0 : Mode bits :

M_1	M_0	Mode	Operating Mode
0	0	0	13-bit timer
0	1	1	16-bit timer
1	0	2	8-bit timer

C/T : If 0, used as a timer. (Clock source is the crystal frequency of 8051.)

If 1, used as a counter. (Clock source is the pulse outside the 8051). (Pin P3.4 and Pin P3.5)

GATE : If GATE = 0, software instructions start and stop the timerss. (SETB and CLR).

If GATE = 1, hardware means start and stop the timerss. (Pin P3.2 and P3.3)

TCON (Timer Control Register) :-

MSB							LSB
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1 : Timer 1 Overflow flag : This is set when Timer 1 overflows.

TF0 : Timer 0 Overflow flag : This flag is set when Timer 0 overflows.

TR1 : Timer 1 Run : This flag has to be set to start the timer1. (SETB TR1)

TR0 : Timer 0 Run : This flag has to be set to start the timer 0. (SETB TR0)

IE1 : External interrupt 1 edge flag.

IE0 : External interrupt 0 edge flag.

IT1 : Interrupt 1 type control bit.

IT0 : Interrupt 0 type control bit.

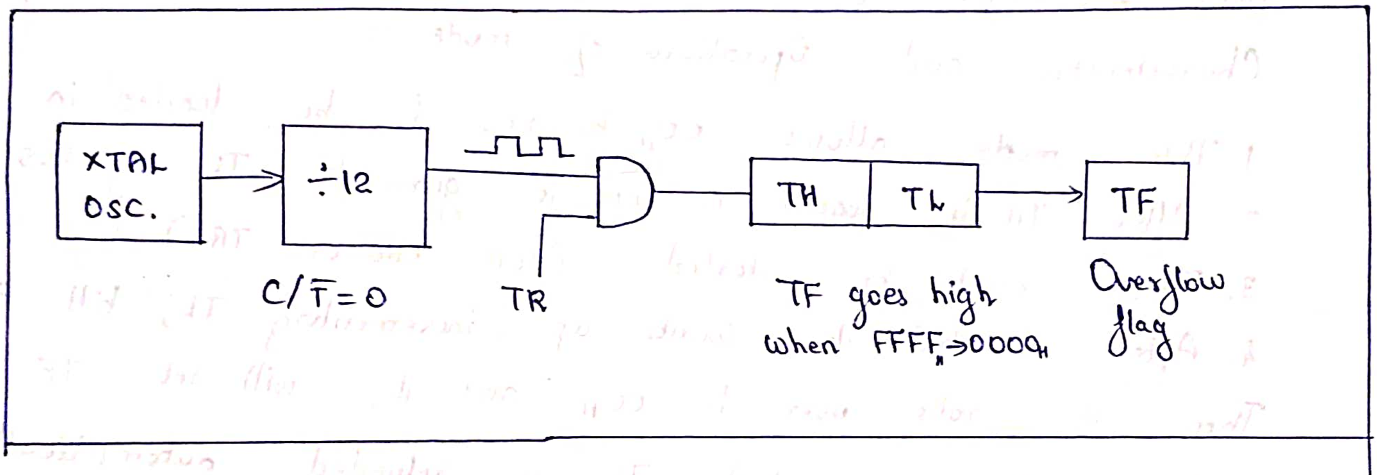
Mode

Ch

Mode 1 Programming :-

Characteristics and Operations of mode 1: (16-bit timer mode)

1. This mode allows 0000_H to $FFFF_H$ to be loaded in TL & TH.
2. After loading TL & TH, timer must be started by SETB TR0 or TR1.
3. After it is started, it starts to count-up. It counts up till $FFFF_H$. Then it rolls over to 0000_H and it will set the flag bit, TF, high.
4. In order to repeat the process, TH & TL must be reloaded with the original value and TF must be reset to 0.



Eg: Write an 8051 ALP to create a square wave of 50% duty cycle on the P1.5 bit. Use Timer 0 to generate the time delay.

```
        MOV   TMOD, #01H    ; Select Timer 0 in mode_1
HERE:   MOV   TL0, # F2H    ; Load count in TL & TH
        MOV   TH0, # FFH
        CPL   P1.5          ; Complement P1.5
        ACALL DELAY
        SJMP  HERE          ; Load TL & TH again
```

Delay using timer 0:

```
DELAY :   SETB   TR0          ; Start the Timer 0
AGAIN :   JNB    TF0, AGAIN   ; Monitor TF until it rolls
          CLR    TR0          ; Stop Timer 0
          CLR    TF0          ; Clear TF
          RET
```

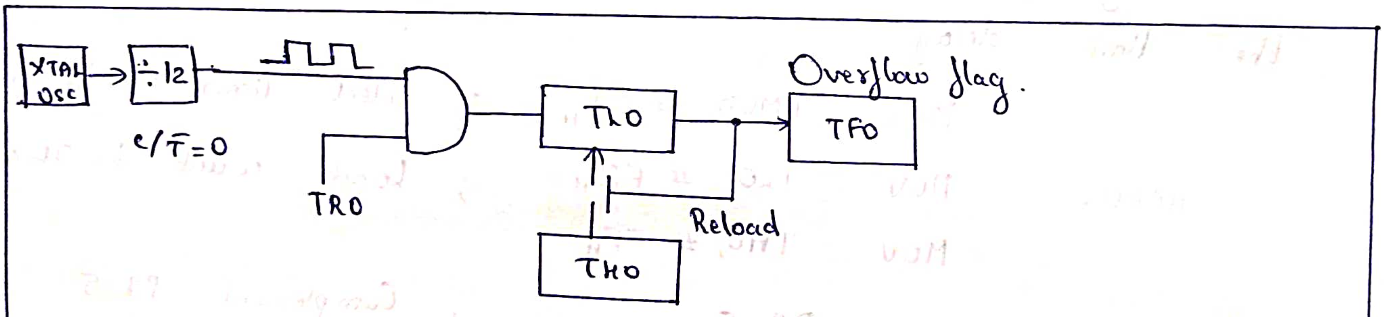
Mode 0 Programming: (13_bit timer)

Mode 0 is exactly like Mode 1, except that it is a 13_bit timer. This mode allows 0000_H to $1FFF_H$ to be loaded in TL & TH.

Mode 2 Programming: (8_bit Timer)

Characteristics and Operations of mode 2:

1. This mode allows 00_H to FF_H to be loaded in TH.
2. After TH is loaded, a copy is given to TL by 8051.
3. Timer must be started. (SETB TR0 or TR1)
4. After started, it counts up, incrementing TL, till FF_H. Then it rolls over to 00_H and it will set TF high.
5. When TF goes high, TL is reloaded automatically with the value kept in TH. To repeat the process, clear TF.



TF0 goes high when FF→0

The main application of Mode 2 is setting the baud rate in serial communication.

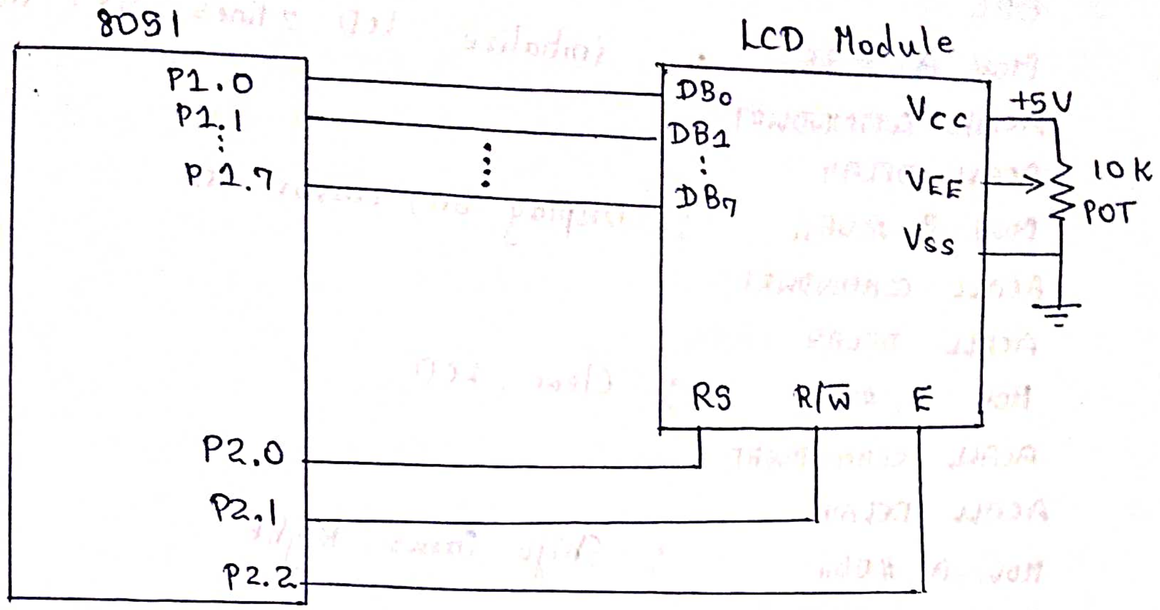
LCD Interfacing

* LCD (Liquid Crystal Display) has the ability to display numbers, characters and graphics.

LCD pin discriptions:

Pin	Symbol	I/O	Description
1	V_{SS}	–	Ground
2	V_{CC}	–	+5V Power Supply
3	V_{EE}	–	Power Supply to control contrast
4	RS	I	RS = 0, to select command register RS = 1, to select data register
5	$R/\overline{W}$	I	$R/\overline{W} = 0$ for write $R/\overline{W} = 1$ for read
6	E	I/O	Enable (To latch the information presented to its data pins, high-to-low pulse)
7	DB0	I/O	8-bit data bus (Bidirectional)
8	DB1	I/O	
9	DB2	I/O	
10	DB3	I/O	
11	DB4	I/O	
12	DB5	I/O	
13	DB6	I/O	
14	DB7	I/O	

LCD connection with 8051:



LCD Command Codes:

CODE (HEX)		Command to LCD Instruction Register
1	–	Clear display screen
2	–	Return home
4	–	Decrement cursor
6	–	Increment Cursor
5	–	Shift display right
7	–	Shift display left
8	–	Display off, cursor off
9	–	Display off, cursor on
⋮		
80	–	Force cursor to beginning of 1st line
C0	–	Force cursor to beginning of 2nd line
38	–	2 lines & 5x7 matrix

Cursor address:

16x2 LCD: 80 81 82 83 84 85 86 through 8F

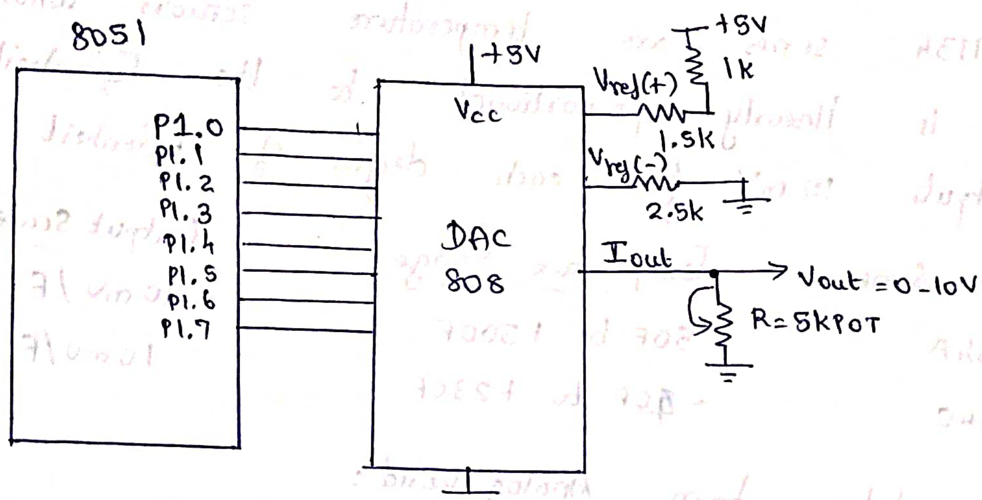
C0 C1 C2 C3 C4 C5 C6 through CF

Program to display 'NO' using LCD :-

```
        ORG
        MOV A, #38H        ; initialize LCD 2 lines, 5x7 matrix
        ACALL COMMNDWRT
        ACALL DELAY
        MOV A, #0EH        ; Display on, cursor on
        ACALL COMMNDWRT
        ACALL DELAY
        MOV A, #01H        ; Clear LCD
        ACALL COMMNDWRT
        ACALL DELAY
        MOV A, #06H        ; Shift Cursor Right
        ACALL COMMNDWRT
        ACALL DELAY
        MOV A, #84H        ; Cursor at line 1, position 4.
        ACALL COMMNDWRT
        ACALL DELAY
        MOV A, #'N'        ; Display letter 'N'
        ACALL COMMNDWRT DATAWRT
        ACALL DELAY
        MOV A, #'O'        ; Display letter 'O'
        ACALL DATAWRT
        ACALL DELAY
AGAIN:  SJMP AGAIN
COMMNDWRT:  MOV P1, A
            CLR P2.0  -> RS = 0
            CLR P2.1  -> R/W = 0
            SETB P2.2 -> E = 1 } High_to_Low
            CLR P2.2  -> E = 0 }
            RET
DATAWRT:    MOV P1, A
            SETB P2.0 -> RS = 1
            CLR P2.1  -> R/W = 0
            SETB P2.2
            CLR P2.2
            RET
DELAY:      MOV R3, #50H
HERE2:      MOV R4, #255D
HERE:       DJNZ R4, HERE
            DJNZ R3, HERE2
            RET
```

Interfacing DAC (Digital to Analog Converter)

DAC is a device used to convert digital pulses to analog signals.

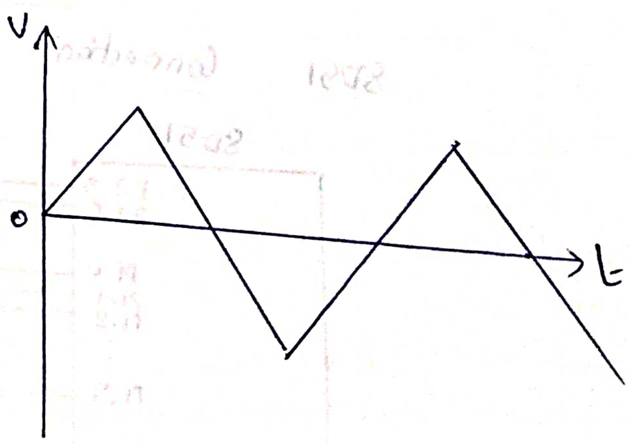


Program to generate a stair-step ramp:

```
        CLR   A
AGAIN:  MOV   P1, A
        INC   A
        ACALL DELAY
        SJMP  AGAIN
DELAY:  MOV   R4, #FFH
HERE:   DJNZ  R4, HERE
        RET
```

Program to generate a triangular wave form:

```
        MOV   A, #00H
LOOP1:  MOV   P1, A
        INC   A
        CJNE  A, #FFH, LOOP1
LOOP2:  MOV   P1, A
        DEC   A
        CJNE  A, #00H, LOOP2
        SJMP  LOOP1
```



INTERFACING SENSORS TO 8051

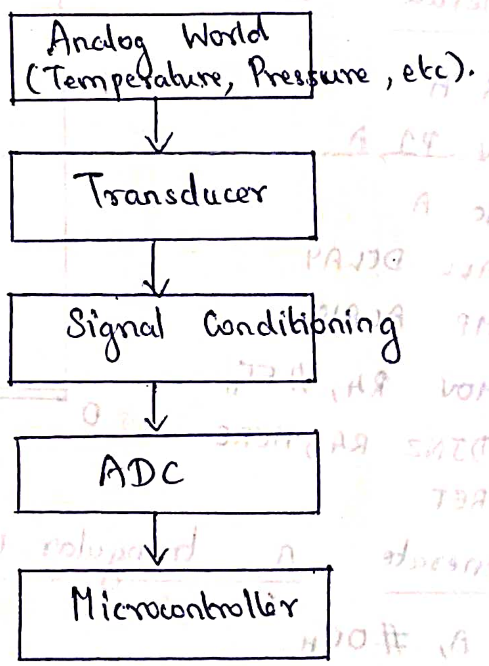
Transducers convert physical data such as temperature, light and speed to electrical signals.

LM34 and LM35 Temperature sensors:

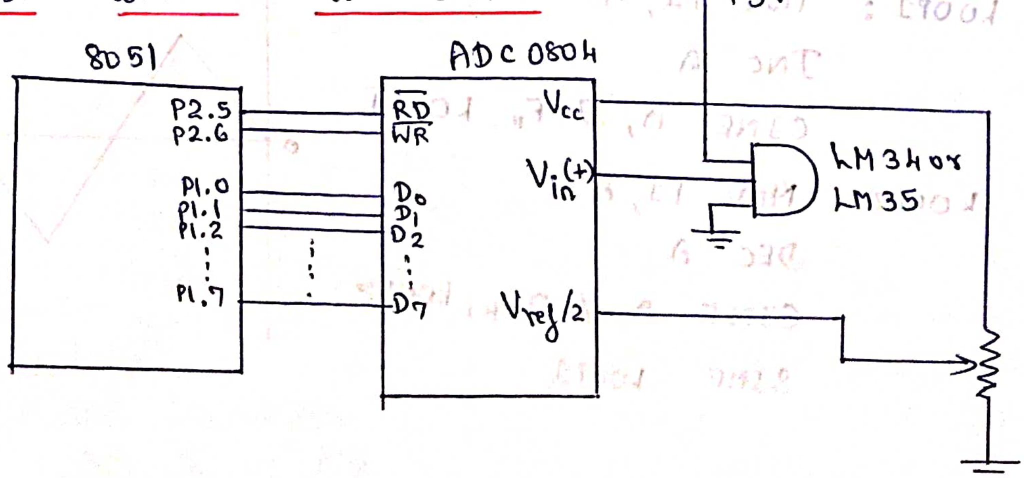
LM34 series are temperature sensors whose output voltage is linearly proportional to the Farenheit temperature. It outputs 10 mV for each degree of Farenheit temperature.

LM34 Series	Temperature Range	Output Scale
LM34A	−50F to +300F	10 mV / F
LM34C	−40F to +230F	10 mV/F

Getting data from Analog World:



8051 Connection to Sensor:



PREPARED BY   VERIFIED BY   16/9/19 APPROVED BY