

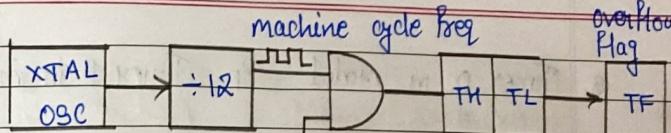
* TMOD

if gate = 1; h/w, INTx → INTO → Timer 0
 if gate = 0; g/w, TRx → INTI → Timer 1
 ← TRO → Timer 0
 ← TRI → Timer 1.

Fox M, & Mo:

M1	M0	Timer
0	0	mode 0 (13-bit)
0	1	mode 1 (16-bit)
1	0	mode 2 (18-bit)
1	1	mode 3 (split timer mode).

* Timer MODE1 (16 bit) :



$$11.059 \Omega \quad C/F = 0 \quad TR = 1 \text{ (start)}$$

Machine cycle =
18 clock pulses.

TF goes
high

$$FFFF = 0$$

* || Configure

1. Timer 0 in mode 1.

$$TMOP = \text{oxoI}$$

2. Timer 1 in mode 2

Ans	7				8			
	G ₁	C/T	M ₁	M ₀	G ₁	C/T	M ₁	M ₀
	0	0	1	0	0	0	0	0

TMOD = 0x20

3. Timer 0 in mode1 & Timer 1 in mode2.

Ans.	T	G	C/F	M ₁	M ₀	G	C/F	M ₁	M ₀
	7	0	0	1	0	0	0	0	1

$$TMOD = 0x21.$$

* Note :

Whenever a single timer is configured in a particular mode by default the other unused timer will be configured in mode0.

- * Find the timers clock Frequency and its time period for various 8051 based systems with the following crystal frequencies:

$$1. 11.0592 \text{ MHz} = XTAL F_{\text{req}}$$

Calculate m/c freq & time period.

$$\text{Ans. M/c cycle Frequency} = \frac{\text{crystal freq}}{12}$$

$$= 11.0592$$

$$= 921.6 \text{ KHz.}$$

$$\text{Time period} = \frac{1}{f} = 1$$

m/c cycle freq

$$= \frac{1}{1}$$

$$= 921.6 \text{ KHz}$$

$$= 1.086 \mu\text{sec.}$$

$$XTAL F_{\text{req}} = 22 \text{ MHz.}$$

Calculate m/c Frequency & Time period.

$$\text{Ans. m/c cycle frequency} = \frac{\text{crystal freq}}{12}$$

$$= 22$$

$$= 12$$

$$= 1.833 \text{ KHz}$$

$$\text{Time Period} = \frac{1}{22}$$

m/c cycle Frequency

$$XXYY-ZZZZ = 11100000$$

$$1.833$$

$$XXYY-ZZZZ = 0.5455 \mu\text{sec.}$$

$$00A2-ZZZZ = XXYY$$

* Questions

- * Generate 25 ms time delay using timer 1 in mode 1. Assume XTAL = 11.0592 MHz. count to be loaded

Ans. Required delay = $[(FFFF - YYXX) + 1] \times \text{time period}$

$\begin{array}{c} \uparrow \\ TH \\ \downarrow \\ TL \end{array}$ $\begin{array}{c} \rightarrow \text{roll back} \\ \downarrow \end{array}$

Time period = $\frac{1}{\text{m/c cycle freq.}} = \frac{1}{(11.0592 \times 12)} = 1.08549$

$$\therefore 25 = [(FFFF - YYXX) + 1] \times 1.085$$

$$\frac{25}{1.085} = [(FFFF - YYXX) + 1]$$

$$25041 - 1 = FFFF - YYXX$$

$$25040 = FFFF - YYXX$$

$$\therefore YYXX = FFFF - 5A00 \\ = A5FF$$

- * Obtain the maximum possible delay using timer 0 in mode 1. Assume XTAL = 11.0592 MHz.

Ans. Required delay = $[(FFFF - YYXX) + 1] \times 1.085$

\therefore To get max delay put YYXX as 0000

$$\begin{aligned} \therefore \text{Required delay} &= [(FFFF - 0000) + 1] \times 1.085 \\ &= (65535 + 1) \times 1.085 \mu\text{sec.} \\ &= 71106.56 \mu\text{sec} \\ &= 71.1 \text{ msec} \end{aligned}$$

- * Obtain the values to be loaded into TH & TL register to generate a delay of 10ms.

Ans. Required delay = $[(FFFF - YYXX) + 1] \times \text{time period}$

$$\frac{10 \times 10^{-3}}{1.085 \times 10^6}$$

$$10 = [(FFFF - YYXX) + 1] \times 1.085 \mu\text{sec.}$$

$$\begin{aligned} (9217 - 1) &= FFFF - YYXX \\ YYXX &= FFFF - 2400 \\ &= DBEE. (OBFF). \\ \text{TH TL.} & \end{aligned}$$

decimal to hexa.

mode 1 - 16 bit mode.

mode 2 - 8 bit mode.

classmate

Date

Page

- * Calculate the delay required for the given count 5000 using timers in mode 1.

Ans. Required delay = $[(FFFF - 44XX) + 1] \times \text{time period}$

$$= [(FFFF - 5000) + 1] \times 1.085 \mu\text{sec}$$

$$= (A3FF + 1) \times 1.085 \mu\text{sec}$$

$$= (41983 + 1) \times 1.085$$

$$= \frac{45552.64}{1000} \mu\text{sec}$$

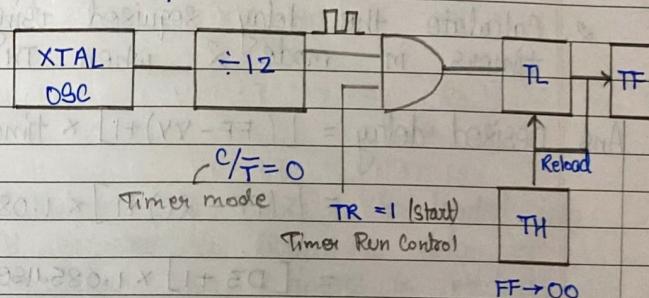
$$= 46.5 \mu\text{sec.}$$

- * Note :

Formula to calculate delay in decimal :

$$\text{delay} = [65536 - NNNN] \times \text{time period.}$$

- * Mode 2 : 8-bit mode
(Auto Reload)



- * Problems :

1. Calculate the max. delay obtained, using timers in mode 2.

Ans. Desired delay = $[(FF - 44) + 1] \times \text{time period.}$

To obtain max. delay put 44 as 00.

$$\begin{aligned} \text{Desired delay} &= [(FF - 00) + 1] \times 1.085 \mu\text{sec.} \\ &= (255 + 1) \times 1.085 \mu\text{sec} \end{aligned}$$

$$= 0.27 \text{ msec.}$$

2. Calculate the delay required using timers in mode 2 when $TH = 2C$.

Ans. Desired delay = $[(FF - YY) + 1] \times \text{time period}$

$$= [(FF - 2C) + 1] \times 1.085 \mu\text{sec.}$$

$$= [D3 + 1] \times 1.085 \mu\text{sec}$$

$$= (211 + 1) \times 1.085 \mu\text{sec}$$

$$= 0.23 \text{ msec.}$$

3. Calculate the count to be loaded into TH register to obtain a delay of 50 msec. Use timers in mode 2. Assume $XTAL = 11.0592 \text{ MHz}$

Ans. Desired delay = $[(FF - YY) + 1] \times \text{time period}$.

$$50 \frac{\mu\text{sec}}{\mu\text{sec}} = [(FF - YY) + 1] \times 1.085 \mu\text{sec}$$

$$\frac{50}{1.085 \mu\text{sec}} = (FF - YY + 1)$$

$$16 - 1 = FF - YY$$

$$YY = FF - 45.$$

$$= FF - 8D$$

$$= D2.$$

$$\therefore TH = 0XD2.$$

4. Calculate the count to be loaded into TH register to obtain a delay of 500 μsec using timers in mode 2. Assume $XTAL = 11.0592 \text{ MHz}$

Ans.

$$500 \mu\text{sec} = [(FF - YY) + 1] \times 1.085 \mu\text{sec}$$

$$16 - 1 = FF - YY$$

$$YY = FF = 460$$

$$= FF - 1CC$$

$$= FFFFFFF33$$

$$\frac{500}{1.085} = 460.82$$

$$> 0.27 \text{ msec.}$$

[Not possible to generate 500 μsec delay using mode 2].

5. Same abv problem use delay of 3 μsec .

$$FF - 276$$

$$\rightarrow FF - 2 \rightarrow FD.$$

Ans. Desired delay = $[(FF - YY) + 1] \times \text{time period}$.

$$3\text{usec} = [(FF - YY) + 1] \times 1.085\text{usec.}$$

$$\left(\frac{3}{1.085}\right) - 1 = FF - YY.$$

$$1.7649 = FF - YY$$

$$\therefore YY = FF - 0R \\ = FD.$$

- * Write an 8051 C program to toggle all the bits of port 1 continuously with some delay in between. Use Timer 0 in mode 1 to generate the delay.

```
#include <8051.h>
```

```
void TODelay(void);
```

```
void main(void)
```

```
{
```

```
P1 = 0x00;
```

```
TODelay();
```

Timer 0 in mode 1

has TH & TL always.

as mode 1 is 16 bit mode.

(FFFF - YYXX).

TH = 0x^{PP};

TODelay();

}

void TODelay()

TMOD = 0x01 ; // timer 0 in mode 1.

TLO = 0xFE ;

TH0 = 0XB0 ; // delay size unknown

TR0 = 1 ; // start the timer

while (TFO == 0) ; // monitor timer overflow flag.

TR0 = 0 ;

TFO = 0 ;

}

- * Write an 8051 C program to toggle only bit P1₅ continuously every 50msec, use timer 0 in mode 1.

```
#include <8051.h>
```

```
void TODelay(void);
```

```
gbit mybit = P1^5;
```

Write an 8051 C program to toggle
only bit P2.7 continuously with
200msec delay use T1 in mode 1.

classmate
Date _____
Page _____

```
void main (void)
{
    while (1)
    {
        mybit = ~mybit; // Using NOT operator
        T0Delay ();
    }
}
```

```
void T0Delay (void)
{
    TMOD = 0x01; // Timer 0 in mode 1.
    TH0 = 0x4B; // Count for 50msec delay
    TL0 = 0xFE;
    TR0 = 1; // Start the timer.
    while (TF0 == 0);
    TR0 = 0;
    TF0 = 0;
}
```

* Write an 8051 C program to toggle
all bits of P2X continuously every
25msec use timer 1 in mode 1.

classmate
Date _____
Page _____

71.2 8msec max delay
↓ so divide 500.

* Write an 8051 C program to toggle
bits of P2X continuously every 500msec,
use timer 1 in mode 1.

```
#include <8051.h>
void TIM1Delay (void);
void main (void)
{
    unsigned char i;
    while (1)
}
```

```
P2 = 0x00;
TIM1Delay ();
P2 = 0xFF;
TIM1Delay ();
for (i = 0; i < 10; i++)
```

```
}
```

void TIM1Delay (void)

```
TMOD = 0x10; // Timer 1 in mode 1.
TH1 = 0x4B; // 50msec delay.
TR1 = 1; // T1 = 0XF;
while (TF1 == 0);
TR1 = 0;
```

$TRI = 0 ;$
 $TFI = 0 ;$

}

* Write an 8051 C program to toggle
only T1S continuously every 250 msec,
use timer 0 in mode 2 to generate
the delay.

Hns. Delay required = 250 msec

$$\therefore \text{Delay} = [(FF - YY) + 1] \times 1.085 \mu\text{sec}$$

$$250 \text{ msec} = [(FF - YY) + 1] \text{ max delay}$$

$$1.085 \mu\text{sec} \quad 250 \times 40 \quad 71.1 \mu\text{sec}$$

choose 25usec delay $\times 10000 = 250 \text{ msec delay}$

$$250 \times 10000 = 2500000 \mu\text{sec}$$

$$25 \mu\text{sec} = [(FF - YY) + 1]$$

$$1.085 \mu\text{sec}$$

$$28 - 1 = FF - YY$$

$$\therefore YY = FF - 16 (\text{in hexa}).$$

$$= E9.$$

$$\therefore TH0 = 0xE9.$$

#include <8051.h>

```
void TOM2Delay(void);
void mybit = P1^5;
void main()
{

```

```
unsigned char x, y;
while(1)
{

```

```
    mybit = ~mybit;
    for (x = 0; x < 250; x++)

```

```
        for (y = 0; y < 400; y++) ; // For loop to
        // From & Delay();
    }
}
```

repeat the
delay 10000
times.

```
void TOM2Delay(void)
{

```

$TMOD = 0x02 ; // \text{Timer 0 in mode 2.}$

$TH0 = 0xE9 ;$

$TR0 = 1 ; // \text{Start the timer.}$

$\text{while } (TF0 == 0) ;$

$TR0 = 0 ;$

$TF0 = 0 ;$

}

* Write an 8051 C program to toggle only bit 2.7 continuously with 200 msec delay. Use T1 in mode 1.

$$\frac{250 \times 40}{200 \mu\text{sec} \times (0000)} = 200 \mu\text{sec}$$

```
#include <8051.h>
```

```
void delay(void);
sbit mybit = P2^7;
void main()
```

```
{
```

```
    unsigned char x, y;
```

```
    while(1)
```

```
{
```

```
    mybit = ~mybit;
```

```
    for(x=0; x<250; x++)
```

```
        for(y=0; y<40; y++)
```

```
            delay();
```

$$\frac{20}{1.085} - 1 = FFFF - YYYE$$

```
.... YYYE = FFEE
```

```
TH1 = FF
```

```
TI1 = FE
```

```
TR1 = 0
```

```
TMOD = 0x10
```

```
TH1 = 0xFF
```

```
TI1 = 0xFE
```

```
TR1 = 1
```

```
void delay()
```

```
{
```

```
    TMOD = 0x10;
```

```
    TH1 = 0xFF;
```

```
    TI1 = 0xFE;
```

```
    TR1 = 1;
```

* Write an 8051 C program to generate a frequency of 2500Hz on pin 2.7, use timer 1 mode 2 to create the delay.

Ans.

Pin \rightarrow 2.7

Freq \rightarrow 2500Hz.

$$\text{W.R.T. : } t = \frac{1}{f} = \frac{1}{2500} = 400 \mu\text{sec}$$

$$T = T_{ON} + T_{OFF}$$

$$T_{ON} = T_{OFF} = 400 = 200 \mu\text{sec}$$

$$\text{Delay} = [(FF - YY) + 1] \times 1.085 \mu\text{sec}$$

$$\frac{200}{1.085} = (FF - YY) + 1$$

$$18U - 1 = FF - YY$$

$$YY = 255 - 188$$

$$= 48 (\text{in hexadecimal}).$$

$$\therefore TH = 0x48 \text{ for } 200 \mu\text{sec} \text{ delay.}$$

Date _____
Page _____

```
#include <8051.h>
void TIM2Delay(void);
sbit mybit = P2^7;

void main(void)
{
    while(1)
    {
        mybit = ~mybit; // toggle P2.7
        TIM2Delay();
    }
}
```

```
void TIM2Delay()
{
    TMOD = 0x20; // Configure T1 in mode 2.
    TH1 = 0x48; // count for 200musec delay
    TR1 = 1;
    while (TF1 == 0); // monitor TF1
    TR1 = 0; // stop the timer.
    TF1 = 0; // clear TF1.
}
```

* A switch is connected to Pin P1.2, write an 8051 C program to monitor switch & create the following frequencies on Pin P1.7 when SW = 0, 500Hz needs to be generated & when SW = 1, 750Hz needs to be generated use Timer 0 in mode 1 for both of them.

Ans. SW → P1.2 P1.7 → square wave.

SW = 0, 500Hz
SW = 1, 750Hz
in times 0 in mode 1.

$$T = \frac{1}{F} = \frac{1}{500} = 2\text{ms} (0.002 \times 1000).$$

$$\therefore T_{ON} = T_{OFF} = \frac{2\text{ms}}{2} = 1\text{ms}.$$

$$T = \frac{1}{F} = \frac{1}{750} = 1.33\text{ms}$$

$$\therefore T_{ON} = T_{OFF} = \frac{1.33}{2} = 0.66\text{ms}.$$

```

#include <seg51.h>
sbit gw = P1^2;
sbit out = P1^7;
void T0M1Delay (unsigned char);
void main (void)
{
    gw = 1; // Configure gw as input.
    while(1)
    {
        out = ~out; // toggling / generating square
                     // wave on P1.7
        if (gw == 0)
            T0M1Delay (0);
        else
            T0M1Delay (1);
    }
}
void T0M1Delay (unsigned char c)
{
    TMOD = 0x01; // Timer 0 in mode 1.
    if (c == 0)
    {
}

```

TH0 = 0xFC; // generate square
 TLO = 0xB6; wave of 500 Hz
 } freq on P1.7
 else
 {
 TLO = 0x9A; // generate square
 TH0 = 0xFD; wave of 750 Hz
 } freq on P1.7.
 TRO = 1; // Start timer
 while (TFO == 0); // Monitor Flag.
 TRO = 0; // Stop timer
 TFO = 0; // clear Flag

* Programming Timers as Counters:

P3 Bit	Function	Pin
P3.0	RxD	10
P3.1	TxD	11
P3.2	INT0	12
P3.3	INT1	13
P3.4	TO	14
P3.5	T1	15
P3.6	WR	16
P3.7	RD	17

* Assume that 1Hz external clock is being fed into pin T1 (P3.5). Write a C program for counter 1 in mode 2 to count up and display the state of the TL count on P1. Start the count at 00.

```
#include <8051.h>
gbit T1 = P3^5; // T1 is P3^5 no need to declare
void main (void)
{
    T1 = 1; // make T1 as input
    TMOD = 0x60; // Timer 1 as counter 1 in mode 2
    TH1 = 0; // start counting from zero.
    while(1)
    {
        do
        {
            TR1 = 1; // start counter
            P1 = T1; // send count on P1
            if(TF1 == 0);
            {
                TR1 = 0; // stop the counter
                TF1 = 0; // clear flag
            }
        }
    }
}
```

* Assume that 1Hz external clock is being fed into pin T0 (P3.4). Write a C program for counter 0 in mode 1 to count the pulses and display the state of TH0 and TL0 registers on P2 & P1 respectively.

```
#include <8051.h>
void main (void)
{
    TO = 1; // make TO as input
    TMOD = 0x05; // configure TO as counter 0 in mode 1
    TH0 = 0;
    TL0 = 0; // clear TH & TL
    while(1)
    {
        do
        {
            TR0 = 1; // start counter
            P1 = TL0; // send TL value on P1
            P2 = TH0; // " " TH " " P2.
            if(TFO == 0);
        }
    }
}
```

$TRO = 0$; //stop counter

$TF0 = 0$; //clear flag.

* Serial data transmission

Note:

Imp * Timer 1 is used in mode 2 to generate the baud rate for serial data transmission.

* With XTAL = 11.0592 MHz Find the TH1 value needed to have the following baud rates:

- 9600.
- 2400.
- 1800.

Ans: $XTAL = 11.0592 \text{ MHz}$

Machine cycle freq = $11.0592 = 921.6 \text{ kHz}$

Now: $= \frac{921.6}{32} \text{ by UART}$
 $= 28800 \text{ Hz}$

Now to get 1st case [9600]

$$\text{we have: } \frac{28800}{3} = 9600$$

Now convert 3 into 2's compliment:

$$\begin{array}{r}
 & 0000\ 0011 \\
 1's & 1111\ 1100 \\
 + & \\
 2's & \hline 1111\ 1101
 \end{array} = 0xFD.$$

∴ To generate a baud rate of 9600, FD is loaded into TH1.

ii. To get 2nd case [2400]

$$\text{we have: } \frac{28800}{2400} = 12.$$

Now convert 12 into 2's compliment

$$\begin{array}{r}
 & 0000\ 1100 \\
 1's & 1111\ 0011 \\
 + & \\
 2's & \hline 1111\ 0100
 \end{array} = 0xF4.$$

∴ F4 must be loaded into TH1 for 2400.

iii To get 3rd case : [1200] :
 we have : $\frac{28800}{1200} = 24$

Now convert all into A's compliment.

∴ ES must be loaded into TMI Pos 1200.

* Note :
Start bit is always low.
Stop bit/bits are always high.

* Note :
There are two lines on 8051 meant
for serial data transmission / reception
these are TXD & RXD supply.

Pin 3.0 Pos TXD. RXD

Pin 3-1 Pos R&D: TAD

* | gcon :

It is 8 bit register used to program the start bit, stop bit and data bits of data framing, among other things.

SMO SMI SMZ REN TB8 RB8 TI RT

SMD	SCON.7	Serial Port Mode Specifier.
SMI	SCON.6	Serial " " "
SM2	SCON.5	Used for multiprocess comm.
REN	SCON.4	Set/Cleared by software to enable
TB8	SCON.3	Not Used disable serial
RBS	SCON.2	Not Used
TI	SCON.1	Transmit interrupt flag. Set by HW at the begin of the stop bit model and cleared by SW.

RI 8 CON. 0 Receive interrupt flag set by HW at the begin of the stop bit mode 1 and cleared by SW.

Note : Always make SMA , $TB8$ & $RB8 = 0$.

* G_{M0}, G_{M1}:

G _{M0}	G _{M1}	Serial mode 0
0	0	Serial mode 0
0	1	Serial mode 1
1	0	" " 2
1	1	" " 3

Always serial mode 1 is used. It has 8 bit data, 1 stop bit & 1 start bit.

* G_{M2}: This enables the multiprocessing capability of 8051.

* REN (Receive enable)

It is bit addressable register:

→ When it is high, it allows 8051 to receive data on RxD pin.

→ If low it is disabled.

* TI (transmit interrupt)

When 8051 finishes the transfer of 8 bit character

→ It raises TI Flag to indicate that it is

→ Ready to transfer another byte.
TI bit is raised at the beginning of the stop bit.

* RI (receive interrupt)

When 8051 receives data serially via RxD, it gets rid of the start & stop bit and places the byte in SBUF register.

→ It raises the RI Flag bit to indicate that a byte has been received and should be picked up before it is lost.

→ RI is raised halfway through the stop bit.

* Syntax is : [transfer character serially]:

TMOD = 0x80 ; // configure Timer1 in mode 2.

TH1 = 0xFD ; // 9600 baud rate.

SCON = 0x50 ; // selects mode1 start bit

RI = 1 ; // start timer1.

TI = 0 ; // clear TI.

SBUF = Byte of data to be transmitted;

while (TI == 0) ; // monitors TI Flag.

* Write a C program for 8051 to transfer the letter ox character 'A' serially at 4800 baud rate continuously. Use 8bit data & one stop bit.

```
#include <8051.h>
void main (void)
{
```

```
    TMOD = 0x20; //Configure T1 in M2 to set
                  //baud rate.
```

```
    TH1 = 0xFA; //baud rate = 4800.
```

```
    SCON = 0x50; //Serial M1 & REN enabled.
```

```
    TR1 = 1; //Start timer
```

```
    while(1)
```

```
{
```

```
    SBUF = 'A'; //place value in SBUF
```

```
    while (TI == 0); //Monitor transmit flag
```

```
    TI = 0; //clear T1 after the completion
            //of transmission.
```

```
}
```

* Same prev problem statement.
characters 'GIT' and baud rate : 9600.

```
#include <8051.h>
```

```
void serialTx (unsigned char);
void main (void)
{
```

```
    TMOD = 0x20;
```

```
    TH1 = 0xFD;
```

```
    SCON = 0x50;
```

```
    TR1 = 1;
```

```
    while(1)
```

```
{
```

```
    SerialTx ('G');
```

```
    SerialTx ('I');
```

```
    SerialTx ('T');
```

```
{
```

```
{
```

```
SerialTx ( unsigned char x )
```

```
{
```

```
    SBUF = x; //load with char to be transmitted
```

```
    while (TI == 0); //monitor TI.
```

```
    TI = 0;
```

```
{
```

20 in (hex) \rightarrow binary
space is obtained.

classmate

Date _____

Page _____

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

```
#include <8051.h>
void main (void)
```

{

TMOD = 0x20;

TH1 = 0xFA;

SCON = 0x50;

TR1 = 1; //start timer

unsigned char mybyte;

while (1)

{

while (RI == 0); //wait to receive

mybyte = SBUF; //write SBUF content
to mybyte.

P1 = mybyte; //place the byte on P1.

RI = 0; //clear

}

SMOD = 0 (Normal baud)
SMOD = 1 (baud $\times 2$)
PCON = Power Control

classmate

Date _____

Page _____

* Doubling Baud Rate:

There are two ways to increase the baud rate of data transfer:

1. To use a high frequency crystal.
2. To change a bit in the PCON register.

PCON register is an 8-bit register:

When 8051 is powered up, SMOD is zero we can set it to high by software and thereby double the baud rate.

(not used by us)

SMOD	-	-	-	GFI	GFO	PD	TDL
------	---	---	---	-----	-----	----	-----

SMOD=1

57600 Hz

XTAL
Oscillator

$\rightarrow \frac{1}{12}$ m/c cycle freq
921.6 kHz

$\frac{1}{32}$

28800 Hz

To timer
1 to set
the baud
rate

TH1 (Decimal)	(Hex)	SMOD=0	SMOD=1
-3	F0	9600	19200
-6	FA	4800	9600
-12	F4	2400	4800
-24	F8	1200	2400

Special function registers:

1. SCON → selects the mode, RE=1 *classmate*
2. SBUF → holds byte for data *RE=0*
3. PCON → SMOD = 0 ; normal
SMOD = 1 ; ×2

* Find the TH1 value (in both decimal & hex) to set the baud rate to each of the following

- a. 9600
- b. 4800
if SMOD = 1. Assume XTAL = 11.0592MHz.

Ans.

a. 9600 baud rate, SMOD = 1.
(×1000)

$$\text{Now: } \text{XTAL} = 11.0592 = 921.6 \text{ kHz.}$$

$$\therefore \text{XTAL} = \frac{921.6}{12} = 7600 \text{ Hz.}$$

$$\text{Now: } \frac{57600}{9600} = 6$$

Now 2's compliment of 6:

$$\begin{array}{r} 000000110 \\ 111111001 \\ + \quad \quad \quad 1 \\ \hline 00000011010 \end{array}$$

$$1111\ 1010 \Rightarrow 0xFA.$$

b. 4800 baud rate, SMOD = 1.

$$\begin{aligned} \text{Now: } \text{XTAL} &= \frac{11.0592}{12} = 921.6 \text{ kHz.} \\ &= \frac{921.6}{16} = 57600 \text{ Hz.} \end{aligned}$$

$$\text{Now: } \frac{57600}{4800} = 12.$$

decimal value = -12

Now 2's compliment of 12:

$$\begin{array}{r} 00001100 \\ 11111001 \\ + \quad \quad \quad 1 \\ \hline 111110100 \end{array}$$

$$1111\ 0100 \Rightarrow 0xF4.$$

* Write an 8051 C program to send the two messages "Normal Speed" and "High speed" to the serial port. Assume that SW is connected to pin P2.0, monitors its status and set the baud as follows:

SW = 0, 8800 baud rate (0xFF)

SW = 1, 56K baud rate

Assume XTAL = 11.0592 MHz for both cases.

```
#include <8051.h>
```

```
sbit SW = P2^0; // input switch
```

```
void main(void)
```

```
{ unsigned char mess1[] = " Normal Speed";
```

```
unsigned char mess2[] = " High Speed";
```

```
TMO = 0x20; // Timer 1 in mode 2
```

```
TH1 = 0xFF; // Count for 8800 baud
```

```
SCON = 0x50; // Serial mode 1, REN=1
```

```
TR1 = 1; // Start timer.
```

```
unsigned char z; -
```

```
if (SW == 0)
```

```
{
```

```
for (z = 0; z < 12; z++)
```

PCON is not bit addressable

MSB of PCON is used to double the baud rate.

```
gBUF = mess1[z]; // place char in gBUF
```

```
while (TI == 0); // wait for transmission
```

```
TI = 0; // clear TI
```

```
}
```

```
else
```

```
{ (0000 0000)
```

```
PCON = PCON & 0x80; // SMOD=1
```

```
for (z = 0; z = 10; z++)
```

```
{
```

```
gBUF = mess2[z]; // place char in gBUF
```

```
while (TI == 0);
```

```
TI = 0;
```

```
}
```

```
{
```

* LCD Interfacing :

LCD Pin Descriptions :

Pin	Symbol	I/O	Description
1		-	
2		-	
3		-	
4	RS	I	
5	R/W	I	
6	E	I/O	Enable
7	DB0	I/O	The 8 bit data bus
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 interfacing diagram :

- * LCD Command codes :
Code(Mex) Command
 1 clear display screen
 2
 3
 4
 5
 6
 7
 8
 A
 C
 E
 F
 80
 CO
 38

* Write an 8051 C program to send the letters 'M', 'D' and 'E' to the LCD.
 Use the busy Flag method
`#include <8051.h>`
 If P1 is holding LCD data = 10
 gbit xs = P2^7 ;

(Lab program can also be written instead of this)

gbit $\text{XW} = \text{P2}^{\text{A}6}$;

gbit $\text{EN} = \text{P2}^{\text{A}6}$;

gbit $\text{busy} = \text{P1}^{\text{A}7}$;

void main(void)

{

$\text{lcdcmd}(0x38)$; //use 2 lines 5x7 matrix.

$\text{lcdcmd}(0x0E)$; //display ON cursor ON blinking.

$\text{lcdcmd}(0x01)$; //clear display screen.

$\text{lcdcmd}(0x06)$; //shift cursor to right

$\text{lcdcmd}(0x80)$; //line 1 position 1.

$\text{lcddata}('m')$; //send char M.

$\text{lcddata}('O')$;

$\text{lcddata}('E')$;

}

void $\text{lcdecmd}(\text{unsigned char value})$

$\text{lcdready}()$; //check the busy flag.

$\text{P1} = \text{value}$; //put the value/char on P1 pins
(receive the cmd)

$\text{xs} = 0$; //select cmd register

$\text{XW} = 0$; //write to lcd.

$\text{en} = 1$;

$\text{MDelay}(1)$; //LCD is slow device hence delay is used

$\text{en} = 0$;
return;

void $\text{lcddata}(\text{unsigned char value})$

{

$\text{lcdready}()$; //check busy flag.

$\text{P1} = \text{value}$; //place value on P1.

$\text{xs} = 1$; //select data register.

$\text{XW} = 0$;

$\text{en} = 1$;

$\text{MDelay}(1)$;

$\text{en} = 0$;

return;

}

void $\text{lcdready}()$

{

$\text{busy} = 1$; //make P1.7 as input.

$\text{xs} = 0$; //select Cmd register.

$\text{XW} = 1$; //read the value.

while ($\text{busy} == 1$)

{

$\text{en} = 0$;

MSDelay(1);

en = 1;

{

{

void MSdelay(...)

{

{