29/02/2020

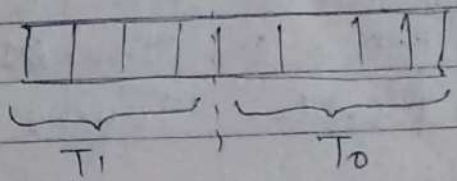
Chapter-03

Timer / Counter & Serial Port Programming

* **TIMERS:** 2, 16 bit timers T0 & T1
  used in: • generating delays
  counters → • to count the external func.

T0 & T1 can be used as timers / counters. It depends on TMOD.

TMOD: 8 bit



- lower nibble is used to configure T0
- upper nibble is used to configure T1.

T1				T0			
Gate	C/T	M1	M0	Gate	C/T	M1	M0

- Gate: 1 = Hardware way of start/stop of timers; 0 = software way of start/stop of timers.
- C/T: 0 = timer; 1 = counter
- M1 M0: 0 0 - 13 bit Timer; 0 1 - 16 bit; 1 0 - 8 bit Autoreload; 1 1 -

- T0 [TH0][TL0]
- T1 [TH1][TL1]

- T0 will act as counter if C/T = 1 & as timer if C/T = 0.

- Same applications for higher 4 bits of TMOD configuring T1.

default gate bit = 0 in our case (∵ we are using software way of start/stop timer

Gate = 0 // Software way of start/stop timer

```
SETB TR0 // Start T0
CLR  TR0 // Stop T0
```

Ques Find the values of TMOD to operate as timers in the following modes

1). Mode 1 Timer 1 → 10h
2). Mode 2 Timer 0 → 12h
3). Mode 1 Timer 01 → 12h
4). Mode 0 Timer 1 → 00h

1). Mode 1 Timer 1

0	0	0	1	0	0	0	0

TMOD = 10h

2). Mode 2 Timer 0

0	0	0	0	0	1	0

TMOD = 12h

3). Mode 1 Timer 0

0	0	0	0	0	0	0	1

TMOD = 01h

4). Mode 0 Timer 1

0	0	0	0	0	0	0	0

TMOD = 00h

* To get delay of 2ms:

(XTAL)
- Crystal Freq = 11.0592 MHz
- 1 Machine cycle has 12 pulse.

∴ Cycle Freq = $\frac{11.0592}{12}$

= 921.6 KHz

• cycle period = $\frac{1}{921.6\text{kHz}}$ = 1.085 μs

How many such cycles of 1.085 μs should elapse to get 2ms.

∴ 1.085 μs × n = 2ms.

$$n = \frac{2 \times 10^{-3}}{1.085 \times 10^{-6}}$$

$$n \approx 1843$$

We require 1843 cycles to get 2ms delay.

To

TH0	TL0
00	00

→ FFFF

⇓

decimal = 65535

∴ 65535 − 1843 + 1

= 63,693 (decimal)

= F8Cdh (Hex)

∴ TH0 = F8
TL0 = cd

→ to get 2ms delay.

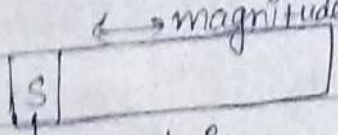
★ 8051 C data types:

1). Unsigned char (8-bit data) Range 0 - 255
000 ↲ ↳ FFFH

Eg unsigned char val;
val = 0x10;

• only +ve values can be assigned.

2). Signed char — 8 bit.



- 1 sign bit &
  7 bit magnitude
- Range −128 to +127
  (−80H) to (7FH)

3). unsigned Int
- Range 0 – 65535 (2^{16})
  (000) to (FFFFh)
- 16-bit
- used only to store large data.
  for smaller values → go for 'char'.
  To avoid memory wastage.

4). Signed Int
- Range −32623 to +32624
- 16-bit → 1 bit for sign
  → 15 bit for magnitude

NOTE:
char val ; // default it is signed data type.
Thus, for +ve no. we should be in habit of writing unsigned int/char.

5). SFR
- 8 bit

E.g: Sfr P0 = 0x80

6). Sbit — Sfr's bit declaration
- 1 bit
  ↳ [to use specific bit of any SFR.]
  ↳ [all those SFR's which are bit addressable].

E.g
```
Sbit mybit = P0^4.
mybit = 1;
```
∴ P0's 4th bit is made as '1'.

only bit addressable SFR's can be accessed using sbit.

7). BIT:

- for all bit addressable location: (20H to 2FH)
  (RAM access)

Eg BIT mybit ; // use mem. from 20H to 2FH
// explicitly, we can't assign.

ques: ① write an embedded C program to send the values -4 to 4 to P0.

② write a 8051 C program to turn on bit P1.5 for 50,000 time.

Soln: ①

```
#include <reg51.h>
main()
{ signed char i;

  for (i=-4 ; i<5 ;i++)
  {   P0 = i;
  }
}
```

②

```
#include <reg51.h>
sbit mybit=P1^5;
main()
{ unsigned int i;
  sbit mybit = P1^5;
  for (i=0; i<50000 ;i++)
      mybit = 1;
}
```

SFR, sbit decl^n should be done befor main.

9

table).

↳ bit addressable register.

* TCON Register : 88h → Add.

TF1	TR1	TF0	TR0				

Timer bit — Interrupt bits (used while configuring interrupts).

- TR0 = 1 // Start T0
- TF0 (Timer Overflow flag) : used to check if the timer T0 has reached the max^m limit i.e. FFFFh and rolls back again; then it sets.

T0

TH0	TL0

0000 → 0001 → . . . → FFFF → 0 ⟲ : TF0 = 1

* In C -	* In ALP -
TR0 = 1 // Start timer	SETB TR0 // Start timer
TR0 = 0 // Stop timer	CLR TR0 // Stop timer

* Mode 1 Programming :-

Steps :-

1. 16 bit values should be loaded into T0/T1. [based on delay required]
2. Start the timer.
3. Keep monitoring timer overflow flag.
4. Stop timer (once, it reaches to max^m limit).
5. If we want to start timer again, then reload timer registers T0/T1; Go back to step 2.

To get delay of 2ms :-

- XTAL = 11.0592 MHz (If not given)
- Crystal freq = $\frac{11.0592\ MHz}{12}$ = 921.6KHz.
- Cycle period = $\frac{1}{f} = \frac{1}{921.6\ KHz}$ = 1.085μs.

- Desired delay = 2ms.

∴ 2ms = n x 1.085μs

$n = \frac{2 \times 10^{-3}}{1.085 \times 10^{-6}}$

∴ n ≈ 1843

To : | THO | TLO |

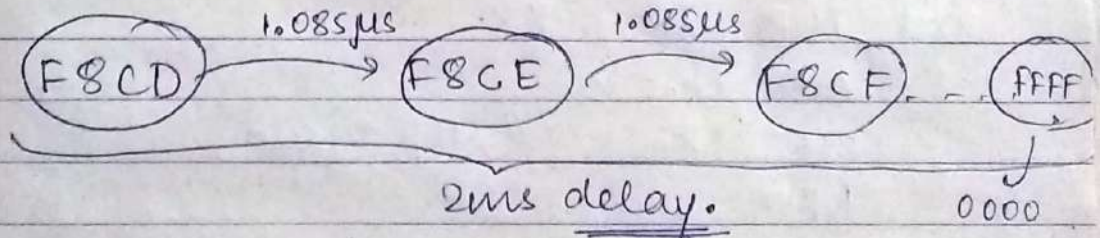
∴ FFFF ⟶ 65535

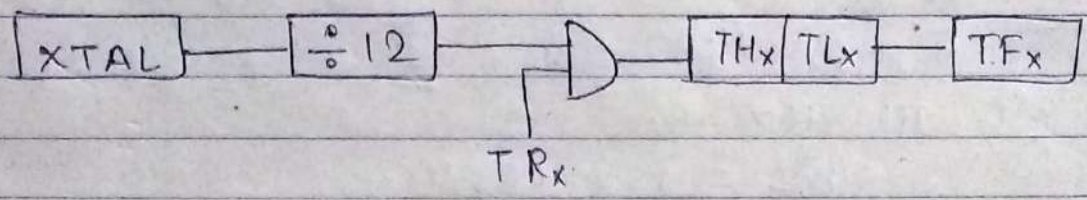
= 65535 − 1843 + 1

= 63693

= F8CD H

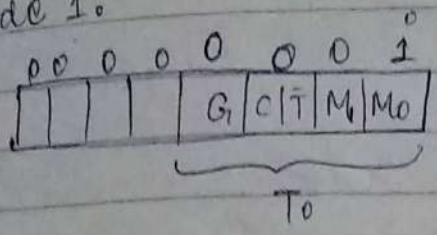
∴ TLO = CD THO = F8.





Steps to programming

Ques Write an ALP to toggle P2.5 for every 2ms of delay. Use timer 0 in mode 1.




```
    MOV  TMOD, #01H
Repeat: MOV  TH0, #0F8H
    MOV  TL0, #0CDH
```

M-1

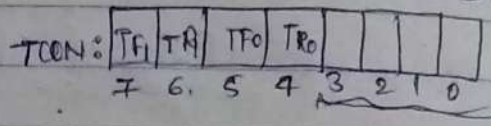
```
    SETB  P2.5        // P2.5 = 1
    Acall delay
    CLR   P2.5
    Acall delay
    SJMP  Repeat
```

M-2

```
RPT: CPL  P2.5
     Acall delay
     SJMP RPT
```

```
MOV TH0, #0F8H
MOV TL0, #0CDH
```

00: For 2nd time TH0 & TL0 is set to 00 & 00

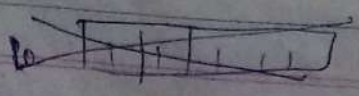



```
    org 100h
delay: SETB TR0   or  SETB TCON.4
XX: JNB  TF0, XX
    CLR  TF0
    CLR  TR0      // Stop timer
    RET
```

Ques Write an ALP to toggle P0 for every 5 ms. Use timer 1 in mode 1. Write all steps....

$$5ms = n \times 1.085\mu s$$

$$n = \frac{5 \times 10^{-3}}{1.085 \times 10^{-6}} \approx 4608$$



time elapse = 65535 − 4608 + 1 = 60928$_D$ = EE00H

```
MOV TMOD, #10H
load: MOV TH1, #0EEH
      MOV TL1, #00H
repeat: Acall load
      MOV P0, #0FFH
      Acall delay
      MOV P0, #00H.
      Acall load
      Acall delay
      SJMP Repeat
org 100H
Delay: SETB TR1.  // SETB TCON.6
XX:   JNB TF1, XX
      CLR TR1      // Stop timer
      CLR TF1
      RET          // Return.
```

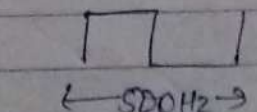
TMOD: | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
T1 | T0
1 | 0

Another method :-

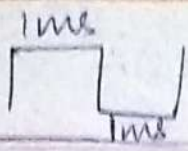
```
RPT: XRL 90h, #0FFh
     Acall delay
     SJMP RPT
```

Ques Write an ALP to generate a square wave of ~~1kHz, 50%~~ of 500Hz on pin P2.4. Use timer 0 in mode 1.

Square wave :-



∴ time (total pulse width) = $\left(\frac{1}{500}\right)$ = 2×10^{-3} = 2ms
↳ pulse width



$\therefore$ delay = $\frac{1}{2}$ (pulse width)

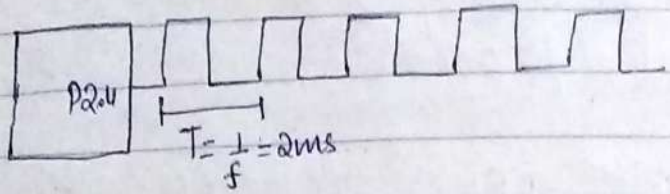
$\therefore$ delay = $\frac{1}{2} \times 2ms = 1ms$

$\therefore$ $1 \times 10^{-3} = n \times 1.085 \times 10^{-6}$

$n \approx 922$

$\therefore$ 65535 − 922 + 1 = 64614$_D$ = FC66$_H$

~~Otes: write~~




```
ALP:  MOV  TMOD ,#01H      // To in mode 1

back: CPL  P2.4
      Acall delay
      SJMP back

Org 100h
delay: MOV  TL0, #66H
       MOV  TH0, #FCH.
       SETB TR0
xx :   JNB  TF0, xx
       CLR  TR0
       CLR  TF0
       RET
```

1 km = 1000 m

C program:

```
#include <reg51.h>

sbit mybit = P2^4;        // mybit = P2.4
void delay_TOM1();

main()
{ while(1)
  {mybit = ~mybit;
   delay_TOM1();
  }
}

void delay_TOM1()
{
    TMOD = 0x01;
    TL0 = 0x66;
    TH0 = 0xFC;
    TR0 = 1;
    while (TF0 == 0);
    TF0 = 0;
    TR0 = 0;
}
```

Ques: Start → with frequency f = 22MHz generate a square wave of 100KHz on pin P2.3. Use timer 1 in mode 1.

Soln: Square wave freq = 100KHz

$\therefore T = \frac{1}{100 \times 10^{+3}} = 1 \times 10^{-5} \times \frac{10}{10}$

$= 10 \times 10^{-6}$

$= 10 \mu sec.$

$\therefore delay = \frac{1}{2} \times 10 = \underline{5 \mu s}$

Now, XTAL freq = 22MHz

$\therefore \quad \frac{22 \times 10^6}{12} = 1.833 \times 10^6$

$= 1.833 \text{ MHz}$

$\text{period} = \frac{1}{1.833 \times 10^6} = 5.4545 \times 10^{-7}$

$= 0.5454 \times 10^{-6}$

$= 0.5454 \mu s$.

$\therefore \quad 5\mu s = n \times 0.5454 \mu s$

$n = 9.16$

$n \approx 9.$

$65535 - 9 + 1 = 65527_d = FFF7_H.$

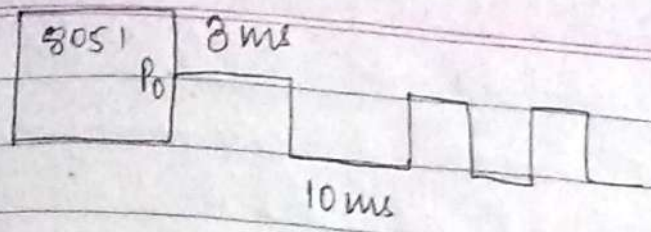
ALP

```
MOV TMOD , #10H

back: CPL P2.3
      Acall delay
      SJMP back

org 100h
delay: MOV TL1, #0F7H
       MOV TH1, #0FFH
       SETB TR1
XX :   JNB TF1, XX
       CLR TR1
       CLR TF1
       RET
```

Ques: Generate the following waveform on all the ~~means~~ ~~of~~ pins of Port 0 (P0).

1). 

XTAL = 22MHz

Use timer 0 in mode 1. WAL k embedded C code.

For delay of 8ms:

$8 \times 10^{-3} = n \times 0.545 \times 10^{-6}$

$\therefore n = 5500.55$

≈ 5500

$\therefore 65535 - 5500 + 1$

$= 60036_d = EA84_H$

For delay of 10ms:- (B853)

$10 \times 10^{-3} = n \times 0.5454 \times 10^{-6}$

$n \approx 18335$

$\therefore 65535 - 18335 + 1$

$= 47201$

$= B861_H$

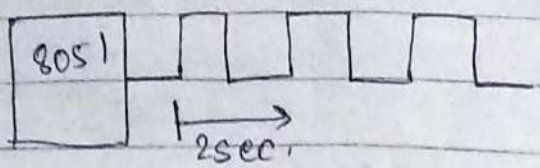
If taken 03 digit then $EA7F_H$ & $B853_H$.
Take upto 3 decimal places (0.545).

```
ALP  MOV TMOD, #01H  → Acall delay
back: XRL P0, #0FFH
      Acall delay1 → XRL P0, #0FFH
      SJMP back

org 100h
delay: MOV TL0, #84H
       MOV TH0, #0EAH
       SETB TR0
XX:    JNB TF0, XX
       CLR TR0
       CLR TF0
       RET

delay1: MOV TL0, #61H
        MOV TH0, #B8H
        SETB TR0
XX:     JNB TF0, XX
        CLR TR0
        CLR TF0
        RET
```

2). Assuming XTAL = 22MHz. Write a prog. to general pulse ~~rate~~ width of 2sec ~~period~~ on pin P2.4. Use T1 mode 1.



delay = 1sec.

$\therefore$ $1sec = n \times 0.545 \times 10^{-6}$

$n = 185185$

$\therefore$ ~~65535 - 185185 + 1 =~~

M-1 $n > 65535$.

$\therefore$ max^{m} delay $= 65535 \times 0.545 = 35.71$ ms

Now, $1sec = n \times 35.71 \times 10^{-3}$

$n = 28$

↳ If we iterate max^{m} delay for 28 times then we can get the req. delay of 1 sec.

0000 to FFFF

M-2 Find for 1 ms & iterate it for 1000 times.

or 10 ms → 100 times } no restriction

For delay of 1 ms :-

$1 \times 10^{-3} = n \times 0.545 \times 10^{-6}$

$n \approx 1835$

$\therefore 65535 - 1835 + 1 = 63701 = F8D5_H$

(P2) For entire P_2

N-1

```
MOV TMOD, #10H

MOV TH1, #00H
MOV TL1, #00H
SETB TR1        MOV R0, #00H // For 28 times
MOV R0, #28H                 ↳ we have to get delay.
back1: ACALL delay
       DJNZ R0, back1
       MOV P2, #0FFH
```

```
bk2: ACALL delay.
     DJNZ R0, bk2:
```

```
delay: SETB TR1
back: JNB TF1, back
      CLR TF1
      CLR TR1
      RET
```

---

*Only for P2.4: Soln

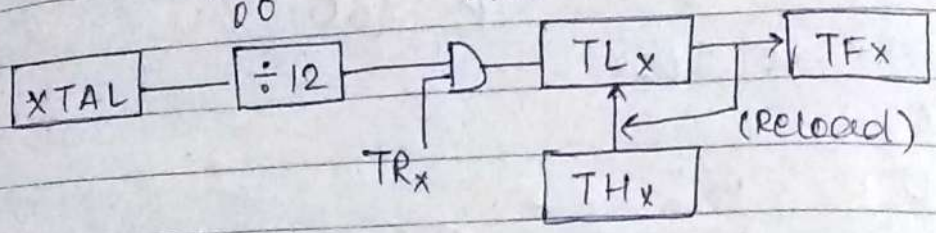
```
MOV TMOD, #10H
MOV TH1, #00H
MOV TL1, #00H
RPT: CPL P2.4
back1: MOV R0, #28
       ACALL delay
       DJNZ R0, back1
       SJMP RPT

delay: SETB TR1
back: JNB TF1, back
      CLR TR1
      CLR TF1
      RET
```

* TIMER'S mode 2 programming.

(8 bit Autoreload mode)

[00 to FF] 00 to 255 : 'H is 8 bit.



- The timer values are auto reloaded.
  - give initial count to Timers.
  - Copy of that will be given to THx & TLx.
  - When reaches max^m, 8+ep (TFx) then autoreload.

1. Configure TMOD for choosing T0/T1.
2. load initial values in THx.
3. Start timer
4. Keep monitoring TFx (Timer Overflow flag)
5. Stop timer

Ques Write an ALP & embedded C program to toggle only P1.5 continuously every 250 μs of delay. Use timer 0 in mode 2.

250 μs XTAL = 11.0592 MHz

TMOD	T1				T0			
	Gate	C/T	M1	M0	Gate	C/T	M1	M0
	0	0	0	0	0	0	1	0

10

x 0

* Calculations:

XTAL = 11.0592 MHz

∴ Timer's Frequency = $\frac{11.0592\ MHz}{12}$ = 921.6 KHz

Timer's period = $\frac{1}{f} = \frac{1}{921.6\ KHz} = 1.085 \times 10^{-6}$ Hz

Here, desired delay = 25 μs

∴ desired delay = n x timer's period

$\frac{25 \times 10^{-6}}{1.085 \times 10^{-6}} = n \approx 23$ i.e. we require 23 pulses to get delay of 25 μs.

∴ 255 - 23 + 1 = $233_d = E9_H$ -> TH

* ALP:

```
        MOV TMOD, #02H
        MOV TH0, #0E9H
Return: CPL P2.5
        Acall delay
        CPL P2.5
        SJMP Return

delay:  SETB TR0
bk:     JNB TF0, bk.
        CLR TR0
        CLR TF0
        RET
```

* Embedded C:

```
#include <reg51.h>
void TOM2_delay();
sbit mybit = P1^5;

main()
{ TMOD = 0x02;
  TH0 = 0x0E9;
  while(1)
  { mybit = ~mybit;
    TOM2_delay(); }
}

void TOM2_delay()
{ TR0 = 0x01;    //=1
  while(TF0==0);
  TR0 = 0x00;    //=0
  TF0 = 0x00;    //=0
}
```

Ques: Write an ALP to generate the square wave of frequency 500 Hz. on pin P2.4. using timer 0 in mode 2. Assuming XTAL = 22MHz.

Calculation: XTAL = 22MHz

∴ timer's frequency = $\frac{22MHz}{12}$ = 1.833 MHz

∴ timer's period = 0.545×10^{-6} sec

Now, desired pulse width = $\frac{1}{500} = 2 \times 10^{-3}$

∴ desired delay = $\frac{1}{2} \times 2 \times 10^{-3} = 1ms$

∴ $1 \times 10^{-3} = n \times 0.545 \times 10^{-6}$

$n = 1834.8$

≈ 1835 > FF or 255

∴ M-2 We try to iterate for 100μs & repeat it for 10 times to get 1ms.

$2 \times 100\mu s = 1ms \Rightarrow x = \frac{1 \times 10^{-3}}{100 \times 10^{-6}} = 10$

OR

M-1 max delay :- Time = $255 \times 0.545\mu s$

$= 138.9\mu s$

$138.9\mu \times N = 1ms$

$N = 7.19$

$\approx 7.$

$dd = n \times 0.545\mu s$ $\quad \times 100\mu s = 1ms$

$x = \frac{1ms}{100\mu} = \frac{1 \times 10^{-3}}{100 \times 10^{-6}} = \frac{10^3}{100} = 10$

for(i=0; i<10; i++) delay();

Now from M-2 :-

$100\mu s = n \times 0.545\mu s$

$n = 183.4 \Rightarrow n \approx 183$

$\therefore 255 - 183 + 1 = 73 = 49$

ALP

```
    MOV   TMOD, #02H
    MOV   TH0, #49H
RPT: MOV  R0, #10
    CPL   P2.4
BK: ACALL delay
    DJNZ  R0, BK
    SJMP  RPT

delay: SETB TR0
XX:  JNB  TF0, XX
     CLR  TR0
     CLR  TF0
     RET
```

Embedded C :-

```
#include <reg51.h>
sbit mybit = P2^4;
void delay();
main()
{
  TMOD = 0x02;
  TH0 = 0x49;
  R0 = 0x10;
  while(1)
  { R0 = 0x10;
    mybit = ~mybit;
    for(i=0; i<10; i++)
      delay();
  }
}
```

```
void delay()
{  TR0 = 01;
   while (TF0==0);
   TF0 = 00;
   TR0 = 00;
}
```

1ms

100μs → 10

$100\mu s \times x = 1ms$

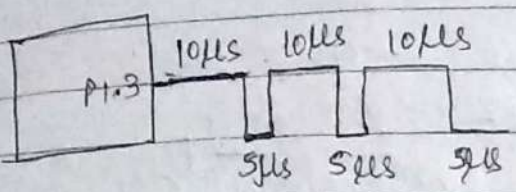
$x = \frac{1 \times 10^{-3}}{100 \times 10^{-6}}$

$= 10$

(TH1) TL1

Ques: Generate the following waveform on P1.3. Use T1 in mode 2.

XTAL = 22MHz



Timer's freq = $\frac{22\,MHz}{12} = 1.833 \times 10^{-6}\,Hz$

Timer's period = $\frac{1}{1.833 \times 10^{-6}} = 0.545\,\mu s$

For 10μs:

desired delay = 10μs

$\therefore 10 \times 10^{-6} = n \times 0.545 \times 10^{-6}$

$n = 18.34$

≈ 18

$\therefore 255 - 18 + 1 = 238_d = EE_H$.

For 5μs: desired delay = 5μs

$\therefore 5 \times 10^{-6} = n \times 0.545 \times 10^{-6}$

$n = 9.17$

≈ 9.

$\therefore 255 - 9 + 1 = 247_d = F7_H$.

We can do for only 5μs & call it twice for 10μs delay.

```
MOV TMOD, #20H

delay:
CPL P1.3
delay: 1    1.15 μs
CPL P1.3
```

ALP:

```
MOV   TMOD , #20H
MOV   TH1 , #0F7H

BK:   SETB  P1.3 ;
      ACALL delay ;
      ACALL delay ;
      CLR   P1.3 ;
      ACALL delay ;
      SETB  P1.3 ;
      SJMP  BK ;

delay: SETB  TR1 ;
XX:    JNB   TF1, XX ;
       CLR   TR1 ;
       CLR   TF1 ;
       RPT ;
```