==

Generate a square wave of 100 Hz at pin P1.0 of 8051 using timer

==

```
#include <reg51.h>                // include 8051 register file
sbit pin = P1^0;                  // decleare a variable type sbit for P1.0
main()
{
      P1 = 0x00;                  // clear port
      TMOD = 0x09;                  // initialize timer 0 as 16 bit timer
loop:TL0 = 0xAF;                    // load valur 15535 = 3CAFh so after
      TH0 = 0x3C;                   // 50000 counts timer 0 will be overflow
       pin = 1;                   // send high logic to P1.0
       TR0 = 1;                   // start timer
       while(TF0 == 0)  {}        // wait for first overflow for 50 ms

       TL0 = 0xAF;                  // again reload count
       TH0 = 0x3C;
       pin = 0;                   // now send 0 to P1.0
      while(TF0 == 0)  {}         // wait for 50 ms again
  goto loop;                      // continue with the loop
}
```

==

Timer 1 Mode 2(8b auto reload) ISR Example Program

==

This example program shows how to configure timer/counter 1 as an
8-bit timer. An interrupt service routine (ISR) is invoked each
time the timer overflows (goes from 0xFF to 0x00). Inside the
ISR, a variable called overflow_count is incremented by one.

To test this program...

1. Start the debugger.
2. Set a breakpoint on the 'overflow_count++' line in the ISR.
3. Run the program.

Each time the interrupt routine is invoked, the debugger will stop
running the program. Position the cursor over 'overflow_count' to
see its current value.

```
#include <reg52.h>
#include <stdio.h>

/*------------------------------------------------
Timer 1 Interrupt Service Routine.

Set a breakpoint on 'overflow_count++' and run the
program in the debugger.  You will see this line
executes every 100 clock cycles (or 10,000 Hz).

So, overflow_count is actually a 1/10,000 sec
timer.
------------------------------------------------*/
static unsigned long overflow_count = 0;

void timer1_ISR (void) interrupt 3
{
overflow_count++;   /* Increment the overflow count */
}

/*------------------------------------------------
MAIN C function
------------------------------------------------*/
void main (void)
{
/*--------------------------------------
Set Timer1 for 8-bit timer with reload
(mode 2). The timer counts to 255,
overflows, is reloaded with 156, and
generates an interrupt.

Set the Timer1 Run control bit.
--------------------------------------*/
TMOD = (TMOD & 0x0F) | 0x20;  /* Set Mode (8-bit timer with reload) */
TH1 = 256 - 100;             /* Reload TL1 to count 100 clocks */
TL1 = TH1;
ET1 = 1;                     /* Enable Timer 1 Interrupts */
TR1 = 1;                     /* Start Timer 1 Running */
EA = 1;                      /* Global Interrupt Enable */

/*--------------------------------------
Do Nothing.  Actually, the timer 1
interrupt will occur every 100 clocks.
Since the oscillator runs at 12 MHz,
the interrupt will happen every 10 KHz.
--------------------------------------*/
while (1)
  {
  }
}
```

```
====================================================
Counter 0 Example Program
====================================================

/*This example program shows how to configure timer/counter 0 as a 16-bit counter.  Each time port
3.4 goes low, the 16-bit counter is incremented by 1.

To test this program...

1. Start the debugger.
2. Run the program.
3. Click the 'Toggle Port 3.4' pin to generate external event.

You will see the value of the counter increase each time you click on the ‘Port 3.4' pin*/

#include <reg52.h>
#include <stdio.h>

/*----------------------------------------------
MAIN C function
----------------------------------------------*/
void main (void)
{
/*------------------------------------
Set Timer0 for 16-bit counter mode.
Set the Timer0 Run control bit.
------------------------------------*/
TMOD = (TMOD & 0xF0) | 0x05;
TR0 = 1;

/*------------------------------------
Output the value of the counter.  As you
toggle P3.4, the timer/counter 0 value
will increase.
------------------------------------*/
while (1)
  {
            //display the count value
            P1 = TH0; //MSB of timer 0
            P2 = TL0; //LSB of timer 0

  }
}
```

==

Counter 1 Example Program (to count the external event on P3.5 continuously in 8b auto reload mode

==

Pseudocode

```
//configure counter 1 in mode 2 (8b auto reload mode)
//initialize TH1 to 0
//initialize P3.5 (Timer 1 input pin) as input pin
//Run the counter 1
//display the count in TL1 on IO port till counter 1 overflows
//on overflow clear overflow flag TF1 and timer run flag TR1  and run the counter again
```