```
/*
    file name : KeyBoardInterrupt.c
    purpose: Demonstration how to use timer1 for two purposes, one for serial port and other
             for general purpose timer/ counter. User can press a key to disable serial port
             and use timer1 for other counting. When a keyboard key is press, timer1 will
             return to it serial port mode. That is when no entry, timer1 can free for other
             usage. When entry is coming, press any key to enable the serial input.
*/

#include <reg51.h>
#include <stdio.h>

bit serialMode;             //determine whether buffer key is press or not

// com port with 9600 baud with crystal 11.0592MHz.
void init_uart(void)
{
    ES = 0;                 //disable serial port interrupt
    SCON  = 0x50;           //serial port mode, enable reception
    TMOD |= 0x20;           //Timer1 set up in Mode2 - 8 bits auto-reload timer
    TL1=0xE0;
    TH1=0xFD;
    TR1   = 1;
    TI    = 1;              //enable transmit interrupt
    RI    = 0;              //disable receive interrupt
    serialMode = 1;
}

// check if there have any key input. If yes then turn timer1 to serial port mode
bit isSerialMode(void)
{
    if(RI==1)
    {
        serialMode = 1;
        RI=0;
        init_uart();
        printf("\nnow serial port enable again!");
    }
    return serialMode;
}

// 16bit auto reload timer2
void set_Timer1(int count)
{
    TR1 = 0;
    TF1 = 0;
    TMOD &=0x0F;        // Clear only Timer 1 control bits
    TMOD |=0x10;
    TH1=(65536-count)/256;  // count 2304 machine cycle
    TL1=(65536-count)%256;  // 1 machine cycle = 12/11.0598M = 1.085us
    serialMode = 0;
}

// turn on/ off the timer1
void turnOnTimer1(void) {   ET1=1;  TR1=1;  }
void turnOffTimer1(void){   ET1=0;  TR1=0;  }

// enable serial port intterupt
void en_Serial_ISR(void)
{
    TI = 0;     //Transmit interrupt
    RI = 0;     //receive interrupt
    ES = 1;     //enable serial port interrupt
}

void main(void)
{
    char cmd;
    init_uart();
    printf("\nKeyboard Interrupt");
    printf("\nPress '1' to disable serial port, and timer1 free for other purpose");
    while(1)
    {
        if(isSerialMode())
        {
            printf("\n[? : ]");
            scanf("%c",&cmd);
            switch(cmd)
            {
               case '1':
                         printf("\nnow serial port will disable");
                         printf("\nPress any key to enable the serial port");
                         set_Timer1(2000);             //attempt to use timer1 for other purpose
                         en_Serial_ISR();              //enable serial port interrupt
                         turnOnTimer1();               //start timer1
                         break;
            }
        }
    }
}
```