

```

/*****
  FileName: timer.cpp

  Author: 胡 贵

  Version : V1.0

  Date: 2008.1.2

  Description:
    system function for timer manager
    1.set timer
    2.kill timer

  History:

  <author>   <time>   <version >   <desc>

*****/

#include "includes.h"
#include "global.h"

CManagerTimer::CManagerTimer()
{
    UINT8 i;

    for(i=0;i<MAX_TIMER_NO/8;i++)
    {
        m_TimerEnable[i]=0;
    }

    HardTimerInit(SYSTEM_RHYTHM_MS);
}

UINT8 CManagerTimer::HardTimerInit(UINT8 ms)
{
    m_TCNT0=0xFF-(UINT8)(ms*1.0/(1000*(1.0/(SYSTEM_FREQUENCY_HZ*1.0/TIMERO_PRESCALER)))); //85//
    TCNT0=m_TCNT0;
    TCCR0|=0x07; // 1024 PRESCALER
    return 1;
}

void CManagerTimer::HardTimerStart(void)
{
    TCCR0|=0x07; // 1024 PRESCALER
    TIMSK|=0x01; // OVERFLOW INTERRUPT ENABLE
}

void CManagerTimer::HardTimerClose(void)
{
    TCCR0&=0xf8;
    TIMSK&=(~0x01); // OVERFLOW INTERRUPT DISABLE
    TIFR|=0x01; // CLEAR OV FLAG
}

UINT8 CManagerTimer::SetTimer(UINT8 id,UINT8 prio,UINT32 ms)
{
    if(id>=MAX_TIMER_NO)
    {
        return 0;
    }
    else
    {
        if(id>=64)
        {
            return 0;
        }
        else
    }
}

```

```

{
    if((ms/ SYSTEM_RHYTHM_MS)>65535)
    {
        return 0;
    }
    else
    {
        if(m_TimerEnable[id/ 8]&(0x01<<(id%8)))
        {
            return 0;
        }
        else
        {
            m_TimerCount[id]=(ms/ SYSTEM_RHYTHM_MS);
            m_TimerCountInit[id]=m_TimerCount[id];
            m_TimerIsrTaskPrio[id]=prio;
            cli();
            m_TimerEnable[id/ 8]|=(0x01<<(id%8));
            sei();
            //SystemDebug("Timer set id is:%d    prio is:%d\n",(UINT16)id,(UINT16)prio);
            return 1;
        }
    }
} ? end else ?
} ? end else ?
} ? end SetTimer ?

UINT8 CManagerTimer::KillTimer(UINT8 id)
{
    if(id>=MAX_TIMER_NO)
    {
        return 0;
    }
    else
    {
        cli();
        m_TimerEnable[id/ 8]&=~(0x01<<(id%8));
        sei();
        return 1;
    }
}

void CManagerTimer::TimerDecrease(void)
{
    UINT8 i;
    // SystemDebug("---\n");
    for(i=0;i<MAX_TIMER_NO;i++)
    {
        if(m_TimerEnable[i/ 8]&(0x01<<(i%8)))
        {
            if(!(--m_TimerCount[i]))
            {
                m_TimerCount[i]=m_TimerCountInit[i];
                ObjTaskManager.TaskRdy( m_TimerIsrTaskPrio[i]);
                // WriteLog("PRIO:%d ",m_TimerIsrTaskPrio[i]);
            }
        }
    }
}

SIGNAL(SIG_OVERFLOW)
{
    TCNT0=TIMER0.m_TCNT0;
    PORTA=0xFF;
    TIMER0.TimerDecrease();
}

```