```
/*************************************************************
    FileName: task_manager.cpp

    Author:胡  贵

    Version :V1.0

    Date: 2008.1.2

    Description:
        some system functions for task manager.
        imitate the UCOS- II.So some of the code is cut from there.
        In this task manager:
        1.Assume that every task cost a very short time slice,
          no long time delay function called in the task.
          If a time delay function have to be called in this task,
          we just have to seperate the large task to small task which
          cost very short time.In that case,the TCB or PCB (this is necessarry in the OS)
          is not needed.After all,here the Task Manager writen by me is only a
          task manager,not a OS.In some samll(just SMALL) application,that's enough and simple
          and reduce the RAM cost.
        2.The task schedule  base on the priority of each task.
          The Task Manager select the highest priotity task which is ready,
          then run it,after runing done, make it sleep.It doesn't record the times that the
          task need to be run,it's just one time running ,until the next time when this task is ready.
        3.Time Tick:20ms


    History:

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

*************************************************************/


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

UINT8 const  TaskUnMapTbl[256] = {
   0, 0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0,     /* 0x00 to 0x0F                             */
   4, 0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0,     /* 0x10 to 0x1F                             */
   5, 0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0,     /* 0x20 to 0x2F                             */
   4, 0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0,     /* 0x30 to 0x3F                             */
   6, 0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0,     /* 0x40 to 0x4F                             */
   4, 0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0,     /* 0x50 to 0x5F                             */
   5, 0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0,     /* 0x60 to 0x6F                             */
   4, 0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0,     /* 0x70 to 0x7F                             */
   7, 0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0,     /* 0x80 to 0x8F                             */
   4, 0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0,     /* 0x90 to 0x9F                             */
   5, 0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0,     /* 0xA0 to 0xAF                             */
   4, 0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0,     /* 0xB0 to 0xBF                             */
   6, 0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0,     /* 0xC0 to 0xCF                             */
   4, 0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0,     /* 0xD0 to 0xDF                             */
   5, 0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0,     /* 0xE0 to 0xEF                             */
   4, 0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0      /* 0xF0 to 0xFF                             */
};

UINT8 const  TaskMapTbl[8]   = {0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80};

CManagerTask::CManagerTask()
{
    TaskInit();
}


UINT8 CManagerTask::TaskInit(void)
{
    UINT8 i;
```


PDF 文件使用 "pdfFactory Pro" 试用版本创建 www.fineprint.cn

```
    m_TaskRdyGrp=0;
    for(i=0;i<(MAX_TASK_NO/8);i++)
    {
        m_TaskRdyTbl[i]=0;
        m_pTask[i]=0;
    }

    TaskRdy(63);//idle task is ready ,so be sure a idle task is created first.
    return 1;
}


//y    = OSUnMapTbl[OSRdyGrp];

//x    = OSUnMapTbl[OSRdyTbl[y]];

//prio = (y << 3) + x;


void CManagerTask::TaskRun(void)
{
    UINT8 y,x,prio;

    while(1)
    {
        cli();
        y=TaskUnMapTbl[m_TaskRdyGrp];
        x=TaskUnMapTbl[m_TaskRdyTbl[y]];
        sei();
        prio=(y<<3)+x;
        (*m_pTask[prio])();
        if(prio! =63)
        {
            //PORTA=0XFF;
            TaskSleep(prio);
        }
        else
        {
            //PORTA=0X00;
        }
    }
} ? end TaskRun ?

UINT8 CManagerTask::TaskCreat(void (*pTask)(void),UINT8 prio)
{
    if(m_pTask[prio]==0)
    {
        m_pTask[prio]=pTask;
        SystemDebug("Task create ok! - - - prio is:%d\n",(UINT16)prio);
        return 1;
    }
    else
    {
        SystemDebug("Task create failed! - - - prio is:%d\n",(UINT16)prio);
        return 0;
    }
}


void CManagerTask::TaskRdy(UINT8 prio)
{
//  SystemDebug("Task ready! - - - prio is:%d\n",(UINT16)prio);
    cli();
    m_TaskRdyGrp| =TaskMapTbl[prio>>3];
    m_TaskRdyTbl[prio>>3]| =TaskMapTbl[prio&0x07];
    sei();
}
```


PDF 文件使用 "pdfFactory Pro" 试用版本创建 www.fineprint.cn

```
void CManagerTask::TaskSleep(UINT8 prio)
{
	cli();
	if ((m_TaskRdyTbl[prio >> 3] &= ~TaskMapTbl[prio & 0x07]) == 0)
		m_TaskRdyGrp &= ~TaskMapTbl[prio >> 3];
	sei();
}
```


PDF 文件使用 "pdfFactory Pro" 试用版本创建 www.fineprint.cn

PDF 文件使用 "pdfFactory Pro" 试用版本创建 www.fineprint.cn