综合设计实验报告

一、实验目的

(一) 掌握 μCOS-II 中任务管理的基本方法，熟练使用 μCOS-II 任务管理基本系统调用。
(二) 掌握 uC/OS-II 操作系统下使用信号量解决任务之间的同步互斥问题。
(三) 掌握嵌入式实时操作系统 μC/OS 中时间中断的使用情况。
(四) 掌握 uC/OS-II 操作系统下任务间通讯的方法。
(五) 加强综合处理问题的能力。

二、实验内容

结合内容 1 和内容 2，修改程序，创建三个任务，两个按键任务一个响应任务。有键按下即发送消息邮箱。任一消息均控制响应任务运行。两个按键任务中任意一个按键按下，则发送一个消息邮箱给响应任务。若按下是数字（0-9），则在串口打印输出，并在数码管显示；如为非数字（其他），则分别对应 LED 灯的闪烁、步进电机的正转，反转，停止、加速和减速。

三、实验步骤

(一) 参考 Eg_kbd.apj，Eg_stepper.apj，Eg_timer.apj，了解如何通过按键控制 LED 的闪烁和对步进电机的控制。
(二) 参考 Eg3.aws 了解如何通过邮箱通信控制两个任务之间的通信。
(三) 修改程序，使一个任务中能响应按键，使数字能在串口打印；并能控制 LED 灯的闪烁和对步进电机的控制；同时发数据到邮箱。另一个任务通过 OSMboxPend()函数等待一个邮箱中的消息，如果邮箱中没有可用的消息，调用任务就被挂起，直到邮箱中有了消息或者等待超时即执行 User_SEG_Blink()把按键值在数码管上显示出来。

四、实验核心代码（如是在已有程序上的修改，直接写出修改部分程序）。

开头定义：

```
int seg=0;
OS_EVENT *Mbox1;
```

在 void User_SEG_Blink(void)中：

```
*((unsigned char *)0x10000004) = seg_value[seg];
```

在 void TaskLED(void *Id)中：

```
char  Msg[100];
INT8U      err;
int        nCount = 0;
……
for (;;) {
      INT8U    ch; //相应按键的值
      ch = Key_GetKey();
      if(ch == 0)
      continue;
      switch( ch )
      {
      case '1':
            uHALr_printf("\r1");
            seg=1;
            sprintf(Msg, "TaskSEG %d", nCount++); //发送数据到邮箱
            OSMboxPost(Mbox1, Msg);
            break;
      ……
      case 'A':
            uHALr_printf("\rA\n");
            User_LED_Blink();
            seg=10;
            sprintf(Msg, "TaskSEG %d", nCount++);
            OSMboxPost(Mbox1, Msg);
            break;
       case 'C':   // 加速
            uHALr_printf("\r 加速");
            DRVStepperSpeedUp();
```

```
            seg=12;
            sprintf(Msg, "TaskSEG %d", nCount++);
            OSMboxPost(Mbox1, Msg);
        break;
        case 'D':   // 减速
            uHALr_printf("\r 减速");
            DRVStepperSpeedDown();
            seg=13;
            sprintf(Msg, "TaskSEG %d", nCount++);
            OSMboxPost(Mbox1, Msg);
        break;
        case 'E': // 正反转控制
            if(direct                                   ==
STEP_MOTOR_CLOCKWISE)
            {
            uHALr_printf("\r 正转");
            direct                                      =
STEP_MOTOR_ANTICLOCKWISE;
            seg=14;
            sprintf(Msg, "TaskSEG %d", nCount++);
            OSMboxPost(Mbox1, Msg);
            }
            else
            {
            uHALr_printf("\r 反转");
            direct = STEP_MOTOR_CLOCKWISE;
            seg=14;
            sprintf(Msg, "TaskSEG %d", nCount++);
            OSMboxPost(Mbox1, Msg);
            }
            DRVStepperSetDirect(direct);
        break;
        case 'F': //使能控制
            if(benable == STEP_MOTOR_ENABLE)
            {
            uHALr_printf("\r 停止");
            benable = STEP_MOTOR_DISABLE;
            seg=15;
            sprintf(Msg, "TaskSEG %d", nCount++);
            OSMboxPost(Mbox1, Msg);
            }
            else
            {
            uHALr_printf("\r 启动");
            benable = STEP_MOTOR_ENABLE;
            seg=15;
            sprintf(Msg, "TaskSEG %d", nCount++);
            OSMboxPost(Mbox1, Msg);
            }
            DRVStepperControl(benable);
        break;

void TaskSEG(void *Id)
{
char  *Msg;
INT8U      err;
for (;;) {
        Msg = (char *)OSMboxPend(Mbox1, 0, &err);
        uHALr_printf("Task2() called\n")；
        OSSchedLock();
          sprintf(print_buf,  "Task%c()  turned\n",  *(char
*)Id);
          uHALr_printf(print_buf);
          User_SEG_Blink();
          OSSchedUnlock();
          OSTimeDly(10)；
        }
}

void Main(void)
Mbox1 = OSMboxCreate((void *)0);
```

五、实验结果及分析

一个任务中能响应按键，使数字能在串口打印并把按键对应值赋给 seg；通过 case 语句控制 LED 灯的闪烁和对步进电机的控制；同时发数据到邮箱。另一个任务通过 OSMboxPend()函数等待一个邮箱中的消息，如果邮箱中没有可用的消息，调用任务就被挂起，直到邮箱中有了消息或者等待超时即执行 User_SEG_Blink()通过在另一个任务里的赋值的 seg 查阅数码管显示数组把按键值在数码管上显示出来。