

# 华侨大学工学院

## 实验报告

课程名称： 嵌入式系统实验

实验项目名称： 实验 5-UC/OS 多任务实验

学 院： 工学院

专业班级： 物联网工程 1 班

姓 名： 李昊唐

学 号： 1995131017

指导教师： 黄德天

2022 年 6 月 13 日

# 预 习 报 告

---

## 一、 实验目的

在现有的关于 UC/OS 的蜂鸣器实验中，再创建 2 个任务，分别用来控制 4 个 LED 和步进电机，具体如下：

控制 4 个 LED 来对 0-15 进行十六进制编码；

控制步进电机顺时针旋转。

## 二、 实验仪器

Arm Developer Suite 1.2

## 三、 实验原理

main.c

```
#include "config.h"
```

```
#define Task0StkLengh 64 // 定义用户任务0的堆栈长度
#define Task1StkLengh 64 // 定义用户任务1的堆栈长度
#define Task2StkLengh 64 // 定义用户任务2的堆栈长度
#define Task3StkLengh 64 // 定义用户任务3的堆栈长度
```

```
OS_STK Task0Stk[Task0StkLengh]; // 定义用户任务0的堆栈
OS_STK Task1Stk[Task1StkLengh]; // 定义用户任务1的堆栈
OS_STK Task2Stk[Task2StkLengh]; // 定义用户任务2的堆栈
OS_STK Task3Stk[Task3StkLengh]; // 定义用户任务3的堆栈
```

```
// 定义独立按键KEY1的输入口
```

```
#define KEY_CON (1<<4) /* GPF4口 */
```

```
// 定义蜂鸣器控制口
```

```
#define BEEP (1<<10) /* GPH10口 */
```

```
#define BEEP_MASK (~BEEP)
```

# 预 习 报 告

---

## 四、 实验内容及步骤

```
// 定义LED控制口（输出高电平时点亮LED）
#define LED1_CON      (1<<11)      /* GPE11口 */
#define LED2_CON      (1<<12)      /* GPE12口 */
#define LED3_CON      (1<<4)       /* GPH4口  */
#define LED4_CON      (1<<6)       /* GPH6口  */

// 步进电机控制口线及操作宏函数定义
#define MOTOA      (1<<5)          /* GPC5  */
#define MOTOB      (1<<6)          /* GPC6  */
#define MOTOC      (1<<7)          /* GPC7  */
#define MOTOD      (1<<0)          /* GPC0  */

#define GPIOSET(PIN)  rGPCDAT = rGPCDAT | PIN
/* 设置PIN输出1, PIN为MOTOA--MOTOD */
#define GPIOCLR(PIN)  rGPCDAT = rGPCDAT & (~PIN)
/* 设置PIN输出0, PIN为MOTOA--MOTOD */

// 按键状态
uint8 KeyState = FALSE;
uint8 i = 0;

void Task0(void *pdata);          // Task0 任务0
void Task1(void *pdata);          // Task1 任务1
void Task2(void *pdata);          // Task2 任务2
void Task3(void *pdata);          // Task3 任务3
void RunBeep(void);

int main(void) {
    OSInit();
    OSTaskCreate(Task0, (void *) 0,
&Task0Stk[Task0StkLengh - 1], 2);
    OSTaskCreate(Task1, (void *) 0,
&Task1Stk[Task1StkLengh - 1], 3);
    OSTaskCreate(Task2, (void *) 0,
&Task2Stk[Task2StkLengh - 1], 4);
    OSTaskCreate(Task3, (void *) 0,
&Task3Stk[Task3StkLengh - 1], 5);
    OSStart();
    return 0;
}
```

# 实验报告

## 五、实验原始数据

```
}
```

```
void Task0(void *pdata) {
    pdata = pdata;
    TargetInit();

    // 初始化I/O
    rGPFCON = (rGPFCON & (~(0x03 << 8)));
    // rGPFCON[9:8] = 00b, 设置GPF4为GPIO输入模式
    rGPHCON = (rGPHCON & (~(0x03 << 20))) | (0x01 << 20);
    // rGPHCON[21:20] = 01b, 设置GPH10为GPIO输出模式
    // 步进电机控制口设置
    rGPCCON = (rGPCCON & (~0x0000FC03)) | (0x00005401);
    // GPC0、GPC5--7口设置为输出
    rGPCUP = rGPCUP | 0x00E1;
    // 禁止GPC0、GPC5--7口的上拉电阻
    rGPCDAT = rGPCDAT & (~0x00E1);
    // 设置GPC0、GPC5--7口输出低电平
    // 初始化I/O
    rGPECON = (rGPECON & (~(0x0F << 22))) | (0x05 << 22);
    // rGPECON[25:22] = 0101b, 设置GPE11、GPE12为GPIO输出模式
    rGPHCON = (rGPHCON & (~(0x33 << 8))) | (0x11 << 8);
    // rGPHCON[13:8] = 01xx01b, 设置GPH4、GPH6为GPIO输出模式
    while (1) {
        if (rGPFDAT & KEY_CON) KeyState = FALSE;
        // 读取GPF口线上的电平, 判断GPF4是否为高电平
        else {
            OSTimeDly(OS_TICKS_PER_SEC / 10);
            if (rGPFDAT & KEY_CON) KeyState = FALSE;
            // 防抖动
            else KeyState = TRUE;
        }
        OSTimeDly(OS_TICKS_PER_SEC / 10);
    }
}
```

```
void Task1(void *pdata) {
```

指导老师签名: \_\_\_\_\_

时 间: \_\_\_\_\_

# 实验报告

---

## 六、 数据处理

```
pdata = pdata;

while (1) {
    if (TRUE == KeyState) { ; // 如果有按键按下, 则蜂鸣
        rGPHDAT = rGPHDAT & BEEP_MASK; // BEEP = 0
        OSTimeDly(OS_TICKS_PER_SEC / 10);
        rGPHDAT = rGPHDAT | BEEP; // BEEP = 1
        OSTimeDly(OS_TICKS_PER_SEC / 10);
    }
    OSTimeDly(OS_TICKS_PER_SEC / 10);
}

void Task2(void *pdata) {
    pdata = pdata;
    while (1) {
        if (i > 15) i = 0;

        i = i & 0x0000000F; // 参数过滤

        // 控制LED4、LED3显示(d3、d2位)
        if (i & 0x08) rGPHDAT = rGPHDAT | (0x01 << 6);
        else rGPHDAT = rGPHDAT & ~(0x01 << 6);
        if (i & 0x04) rGPHDAT = rGPHDAT | (0x01 << 4);
        else rGPHDAT = rGPHDAT & ~(0x01 << 4);
        // 控制LED2、LED1显示(d1、d0位)
        rGPEDAT = (rGPEDAT & ~(0x03 << 11)) | ((i &
0x03) << 11);
        i++;
        OSTimeDly(OS_TICKS_PER_SEC / 10);
    }
}

void Task3(void *pdata) {
    pdata = pdata;

    while (1) {
```

# 实验报告

## 七、实验结论及分析讨论

```
// AB相有效
GPIOSET(MOTOA);
GPIOSET(MOTOB);
OSTimeDly(OS_TICKS_PER_SEC / 10);
GPIOCLR(MOTOA);
GPIOCLR(MOTOB);

// BC相有效
GPIOSET(MOTOB);
GPIOSET(MOTOC);
OSTimeDly(OS_TICKS_PER_SEC / 10);
GPIOCLR(MOTOB);
GPIOCLR(MOTOC);

// CD相有效
GPIOSET(MOTOC);
GPIOSET(MOTOD);
OSTimeDly(OS_TICKS_PER_SEC / 10);
GPIOCLR(MOTOC);
GPIOCLR(MOTOD);

// DA相有效
GPIOSET(MOTOD);
GPIOSET(MOTOA);
OSTimeDly(OS_TICKS_PER_SEC / 10);
GPIOCLR(MOTOD);
GPIOCLR(MOTOA);
}
}
```

预习报告成绩	实验报告成绩	实验操作成绩	总成绩