

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 4-定时器控制步进电机实验

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 6 日

预 习 报 告

一、 实验目的

使用 S3C2410A 的定时器 0 实现 3 秒的定时并产生中断，每产生一次中断来控制 LED 灯和步进电机动作，具体如下：

二、 实验仪器

Arm Developer Suite 1.2

三、 实验原理

奇数次进入定时器中断时，点亮 LED1 关闭 LED2，并控制步进电机顺时针旋转 180 度；

偶数次进入定时器中断时，关闭 LED1 点亮 LED2，并控制步进电机逆时针旋转 180 度。

预 习 报 告

四、 实验内容及步骤

main.c

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

```
// 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 */
```

```
// 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 */
```

```
extern int flag = 1;
```

```
void DelayNS(uint32 dly) {
```

```
    uint32 i;
```

```
    for(; dly>0; dly--) for(i=0; i<50000; i++);
```

```
}
```

```
void AB(uint8 dly) { // AB相有效
```

```
    GPIOSET(MOTOA);
```

```
    GPIOSET(MOTOB);
```

```
    DelayNS(dly);
```

```
    GPIOCLR(MOTOA);
```

```
    GPIOCLR(MOTOB);
```

```
}
```

```
void BC(uint8 dly) { // BC相有效
```

```
    GPIOSET(MOTOB);
```

```
    GPIOSET(MOTOC);
```

```
    DelayNS(dly);
```

实验报告

五、实验原始数据

```
    GPIOCLR(MOTOB);
    GPIOCLR(MOTOC);
}

void CD(uint8 dly) {    // CD相有效
    GPIOSET(MOTOC);
    GPIOSET(MOTOD);
    DelayNS(dly);
    GPIOCLR(MOTOC);
    GPIOCLR(MOTOD);
}

void DA(uint8 dly) {    // DA相有效
    GPIOSET(MOTOD);
    GPIOSET(MOTOA);
    DelayNS(dly);
    GPIOCLR(MOTOD);
    GPIOCLR(MOTOA);
}

void MOTO_Mode2(int dly) {
    if (dly < 0){
        dly = - dly;
        AB(dly);
        BC(dly);
        CD(dly);
        DA(dly);
        AB(dly);
        BC(dly);
        CD(dly);
        DA(dly);
        AB(dly);
        BC(dly);
    } else {
        AB(dly);
        DA(dly);
    }
}
```

指导老师签名: _____

时 间: _____

实验报告

六、 数据处理

```
        CD(dly);
        BC(dly);
        AB(dly);
        DA(dly);
        CD(dly);
        BC(dly);
        AB(dly);
        DA(dly);
    }
}

void IRQ_Time0(void) {
    // 取反蜂鸣器控制I/O口的状态
    if(flag == 0) {
        rGPEDAT = rGPEDAT & ~(0x01<<11);
        rGPEDAT = rGPEDAT | (0x01<<12);
        MOTO_Mode2(-1);    // 控制步进电机逆转
        flag = 1;
    } else {
        rGPEDAT = rGPEDAT | (0x01<<11);
        rGPEDAT = rGPEDAT & ~(0x01<<12);
        MOTO_Mode2(1);    // 控制步进电机正转
        flag = 0;
    }

    // 清除中断标志
    rSRCPND = 1<<10;
    rINTPND = rINTPND;
}

int main(void) {
    // 初始化I/O
    rGPHCON = (rGPHCON & ~(0x33<<8)) | (0x11<<8);
    // rGPHCON[13:8] = 01xx01b, 设置GPH4、GPH6为GPIO输出模式
    rGPHDAT = rGPHDAT & ~(0x05<<4);
    rGPECON = (rGPECON & ~(0x0F<<22)) | (0x05<<22);
    // rGPECON[25:22] = 0101b, 设置GPE11、GPE12为GPIO输出模式

    // 步进电机控制口设置
```

实验报告

七、实验结论及分析讨论

```
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

// 设置中断服务程序
VICVectAddr[10] = (uint32) IRQ_Time0;

// 设置中断控制器
rPRIORITY = 0x00000000;    // 使用默认的固定的优先级
rINTMOD = 0x00000000;    // 所有中断均为IRQ中断
rINTMSK = ~(1<<10);    // 打开TIMER0中断允许

// 定时器设置
// Fclk=200MHz, 时钟分频配置为1:2:4, 即Pclk=50MHz。
rTCFG0 = 250;    // 预分频器0设置为250, 取得200KHz
rTCFG1 = 3;    // TIMER0再取1/16分频, 取得12.5KHz
rTCMPB0 = 0x0000;    // 设置定时器为0
rTCNTB0 = 3 * 12500;    // 定时0.5秒
rTCON = (1<<1);    // 更新定时器数据
rTCON = (1<<0)|(1<<3)    // 启动定时器

IRQEnable();    // 使能IRQ中断(CPSR)

while(1);

return 0;
}
```

预习报告成绩	实验报告成绩	实验操作成绩	总成绩