**智能硬件设计基础上机报告(五)**

学院（系）： 求实书院（两人/单人每组）

姓 名： 学号：

姓 名： 学号：

成 绩：

**一、实验目的和要求**

1 外部传感器的信息读取交互

2 定时器的定时原理问题

3 外部RTC的数据写入和读取问题

**二、实验原理和内容**

1 分析timer demo的执行过程

2 构建一个timer的过程

社交网站的手机截图

AI 生成的内容可能不正确。

更多关于定时器的内容，请回顾“P04-智能硬件的时间外设”中的内容

3 构建一个RTC的过程

(1) INS5699的“.h”文件内容如下所示：

#include "pinctrl.h"

#include "common\_def.h"

#include "soc\_osal.h"

#include "gpio.h"

#include "hal\_gpio.h"

#include "app\_init.h"

#define INS5699S\_ADDR 0x64

#define INS5699S\_REG\_SEC 0x00

#define INS5699S\_REG\_MIN 0x01

#define INS5699S\_REG\_HOUR 0x02

#define INS5699S\_REG\_WEEK 0x03

#define INS5699S\_REG\_DAY 0x04

#define INS5699S\_REG\_MONTH 0x05

#define INS5699S\_REG\_YEAR 0x06

#define INS5699S\_REG\_CONTROL 0X0F

typedef struct {

uint8\_t sec;

uint8\_t min;

uint8\_t hour;

uint8\_t week;

uint8\_t day;

uint8\_t month;

uint8\_t year;

} ins5699s\_time;

ins5699s\_time ins5699s\_GetTime(void);

void ins5699s\_init(void);

void ins5699s\_SetTime(ins5699s\_time time);

(2) INS5699的“.c”文件的部分内容如下所示：

#include "pinctrl.h"

#include "common\_def.h"

#include "soc\_osal.h"

#include "gpio.h"

#include "hal\_gpio.h"

#include "app\_init.h"

#include "ins5699s.h"

#include "i2c.h"

#define I2C\_MASTER\_BUS\_ID 1

void ins5699s\_SendREG(uint8\_t reg, uint8\_t reg\_data){

uint8\_t buffer[] = {reg, reg\_data};

i2c\_data\_t data = {0};

data.send\_buf = buffer;

data.send\_len = sizeof(buffer);

errcode\_t ret = uapi\_i2c\_master\_write(I2C\_MASTER\_BUS\_ID,

INS5699S\_ADDR >> 1, &data);

if (ret != 0) {

printf("INS5699S:I2cWriteREG(%02X) failed, %0X!\n", reg, ret);

return ;

}

}

uint8\_t ins5699s\_ReadREG(uint8\_t reg){

uint8\_t send\_buffer[] = {reg};

uint8\_t read\_buffer[1] = {0};

i2c\_data\_t data = {0};

data.send\_buf = send\_buffer;

data.send\_len = sizeof(send\_buffer);

data.receive\_buf = read\_buffer;

data.receive\_len = 1;

errcode\_t ret = uapi\_i2c\_master\_writeread(I2C\_MASTER\_BUS\_ID,

INS5699S\_ADDR >> 1, &data);

if (ret != 0) {

printf("INS5699S:I2cReadREG(%02X) failed, %0X!\n", reg, ret);

return 0;

}

return data.receive\_buf[0];

}

void ins5699s\_init(void){

osal\_msleep(100);

osal\_printk("INS5699S Init SUCC!\r\n");

}

**三、实验步骤与操作方法**

1 构建一个timer的过程

该Timer的基本要求如下：

(1) 定时器的基本计时周期是1秒钟；

(2) 定时器的中断回调函数的周期是1秒钟，且在回调函数中，涉及到重新开启定时器等基本操作；

(3) 定时器实现时钟功能，能够显示24小时制的时、分、秒；

(4) 可以考虑创建一个新的工程完成上述功能，具体过程会在课程中讲述。

2 INS5699S实验

完成INS5699的两个关键函数，分别是：

(1) ins5699s\_time ins5699s\_GetTime(void);

(2) void ins5699s\_SetTime(ins5699s\_time time);

**四、实验结果与分析**

1 请分析timer demo的运行结果

（请分析系统创建的定时器1、2、3和4的运行方式，另外，timer\_entry功能无需分析）

2 构建一个timer的完整代码

（清打印列出所有的关键源码）

3 INS5699S实验的两个关键函数代码及其分析过程

**五、讨论和建议**