浙江大学

本科实验报告

课程名称: 嵌入式系统

姓　　名: 王睿

学　　院: 计算机学院

专　　业: 计算机科学与技术

学　　号: 3120000305

指导老师: 蔡铭

2015 年 6 月 20 日

浙江大学实验报告

课程名称：嵌入式系统　　实验类型：综合实验

实验项目名称：Lab7：室温计

学生姓名：王睿　　专业：计算机科学与技术　　学号：3120000305

同组学生姓名：龚源、朱稼乐　　指导老师：蔡铭

实验地点：曹西 501　　实验日期：2015 年 6 月 20 日

一、实验目的与要求

1. 学习 uC/OS II 的应用程序编写；
2. 理解如何直接操纵 GPIO，体会与 Linux 的不同；
3. 学习单总线设备的访问方式；
4. 学习 7 段数码管的时分复用驱动方式。

二、实验内容与原理

这个实验的目的是理解 uC/OS II 的任务调度方式，编写 uC/OS II 的应用程序，通过寄存器直接操纵 GPIO 来驱动外部设备。

三、主要仪器设备

硬件

- 树莓派一块；
- 5V/1A 电源一个；
- microUSB 线一根；
- 面包板一块；
- 两位 7 段数码管（共阳）一颗；
- 360Ω 1/8W 电阻 2 颗；
- DHT-11 温湿度传感器 1 个；
- 面包线若干。

软件

- 交叉编译软件

四、操作方法与步骤

1. 设计输出方案，画连线示意图；
2. 在面包板上连线，完成外部电路；
3. 编写 C/C++ 程序，测试程序和电路；
   (a) 测试、实现 uC/OS II 对 GPIO 的访问；
   (b) 实现 DHT-11 数据的读；
   (c) 实现以时分复用方式在四位 7 段数码管上依次显示 0000-9999 的数字；
   (d) 用两个 uc/OS II 任务，一个定时读 DHT-11 数据，一个轮流驱动数码管，一秒一次显示当前温度和湿度。注意处理好两个任务之间的数据共享。

五、实验结果与分析

实现代码如下：

```
//const.h

#define BIT0 22
#define BIT1 23
#define BIT2 24
#define BIT3 25

#define DIGITA 4
#define DIGITB 2
#define DIGITC 3
#define DIGITD 8
#define DIGITE 7
#define DIGITF 10
#define DIGITG 9

#define DHT11PIN 11

#define INPUT 0x0
#define OUTPUT 0x1

#define HIGH 1
#define LOW 0

```

```
#define GPFSEL0 0x20200000

typedef unsigned int ui;
```

```
//main.c

#include "uart.h"
#include "regs.h"
#include "timer.h"
#include "ucos/includes.h"
#include "const.h"

OS_STK  userAppTaskStk1[1000];
OS_STK  userAppTaskStk2[1000];
OS_STK  displayLEDTaskStk[1000];

extern void userApp1(void *);
extern void userApp2(void *);
extern void displayLED(void *);
void init();
extern void PUT32 ( unsigned int, unsigned int );
extern ui GET32 ( ui );

void main()
{
        uart_init();

        InitInterruptController();

        DisableInterrupts();

        timer_init();

        OSInit();

        init();

        OSTaskCreate(userApp1, (void *) 0, &userAppTaskStk1[1000-1],5);

```

```
        OSTaskCreate(userApp2, (void *) 0, &userAppTaskStk2[1000-1],6);

        OSStart();

        while(1);
}

void pinMode(int pin, int st) {
        ui x, offset, gpfsel, val;
        x = pin / 10;
        offset = pin % 10;
        gpfsel = GPFSEL0 + x * 4;
        val = GET32(gpfsel);
        val &= ~ (0x7 << offset * 3);
        val |= st << offset * 3;
        PUT32(gpfsel, val);
}


void init() {
        pinMode (BIT0, OUTPUT) ;
    pinMode (BIT1, OUTPUT) ;
    pinMode (BIT2, OUTPUT) ;
    pinMode (BIT3, OUTPUT) ;

    pinMode (DIGITA, OUTPUT) ;
    pinMode (DIGITB, OUTPUT) ;
    pinMode (DIGITC, OUTPUT) ;
    pinMode (DIGITD, OUTPUT) ;
    pinMode (DIGITE, OUTPUT) ;
    pinMode (DIGITF, OUTPUT) ;
    pinMode (DIGITG, OUTPUT) ;

        pinMode (16, OUTPUT) ;
        pinMode (27, OUTPUT) ;
}
```

```
//userApp.c

```

```c
#include "uart.h"
#include "ucos/includes.h"
#include "const.h"

extern void PUT32 ( unsigned int, unsigned int );
extern ui GET32 ( unsigned int );
extern void pinMode(int, int);
void digitalWrite(int pin, int v);
ui digitalRead(int pin);
int dht11_read_val();

int dht11_val[5]={0,0,0,0,0};

#define GPSET0      0x2020001C
#define GPCLR0      0x20200028
void light(int bit, int val);

void userApp2(void * args) {
        int count = 4;
        while(1) {
                if (count >= 4) {
                        digitalWrite(16, HIGH);
                        digitalWrite(27, HIGH);
                        count = 0;
                }
                dht11_read_val();
                uart_string("in userApp2");
                OSTimeDly(1000);
                count++;
        }
}

void userApp1(void * args) {
        int count = 0;
        while(1) {
                if (count >= 6) {
                        digitalWrite(16, LOW);
                        digitalWrite(27, LOW);
                        count = 2;
```

```
                }
                light(0, dht11_val[0]/10);
                OSTimeDly(1);
                light(1, dht11_val[0]%10);
                OSTimeDly(1);
                light(2, dht11_val[2]/10);
                OSTimeDly(1);
                light(3, dht11_val[2]%10);
                OSTimeDly(1);
                uart_string("in userApp1");
                count++;
        }
}


char digit[10][8] =  //数字显示定义
    {
        {0,0,0,0,0,0,1,1}, //0
        {1,0,0,1,1,1,1,1}, //1
        {0,0,1,0,0,1,0,1}, //2
        {0,0,0,0,1,1,0,1}, //3
        {1,0,0,1,1,0,0,1}, //4
        {0,1,0,0,1,0,0,1}, //5
        {0,1,0,0,0,0,0,1}, //6
        {0,0,0,1,1,1,1,1}, //7
        {0,0,0,0,0,0,0,1}, //8
        {0,0,0,0,1,0,0,1}  //9
    };

void light(int bit, int val){
        if(bit == 0)
                digitalWrite(BIT0,1);
        else
                digitalWrite(BIT0,0);
        if(bit == 1)
                digitalWrite(BIT1,1);
        else
                digitalWrite(BIT1,0);
        if(bit == 2)
```

```
                digitalWrite(BIT2,1);
        else
                digitalWrite(BIT2,0);
        if(bit == 3)
                digitalWrite(BIT3,1);
        else
                digitalWrite(BIT3,0);

        digitalWrite(DIGITA, digit[val][0]);
        digitalWrite(DIGITB, digit[val][1]);
        digitalWrite(DIGITC, digit[val][2]);
        digitalWrite(DIGITD, digit[val][3]);
        digitalWrite(DIGITE, digit[val][4]);
        digitalWrite(DIGITF, digit[val][5]);
        digitalWrite(DIGITG, digit[val][6]);

}

void digitalWrite(int pin, int v) {
        if (v) PUT32(GPSET0, 1<<pin);
        else   PUT32(GPCLR0, 1<<pin);
}

ui digitalRead(int pin) {
        ui res = GET32(GPSET0);
        if (res & (1 << pin))
                return 1;
        else
                return 0;
}

typedef unsigned char uint8_t;

int dht11_read_val(){
    ui lststate=HIGH;          //last state
    ui counter=0;
    ui j=0,i;

    //host send start signal
```

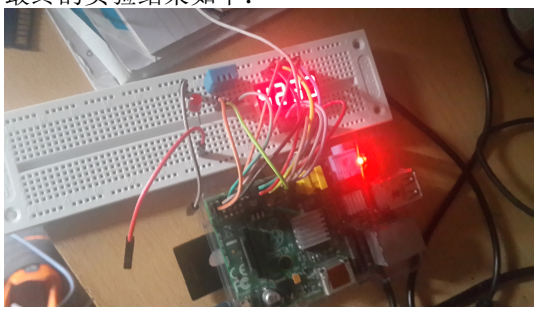
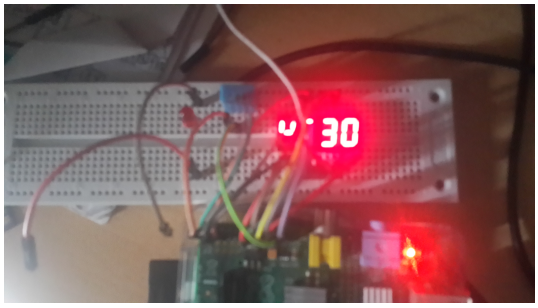
```
    pinMode(DHT11PIN,OUTPUT);        //set pin to output
    digitalWrite(DHT11PIN,LOW);      //set to low at least 18ms
        OSTimeDly(18);
    digitalWrite(DHT11PIN,HIGH);     //set to high 20-40us
        OSTimeDly(40);

    //start recieve dht response
    pinMode(DHT11PIN,INPUT);         //set pin to input
    for(i=0;i<MAX_TIME;i++)
    {
        counter=0;
        while(digitalRead(DHT11PIN)==lststate){      //read pin state to see if dht responsed. if dht always high for 255 + 1 times, break this while circle
            counter++;
                        OSTimeDly(1);
            if(counter==255)
                break;
        }
        lststate=digitalRead(DHT11PIN);              //read current state and store as last state.
        if(counter==255)                             //if dht always high for 255 + 1 times, break this for circle
            break;
        // top 3 transistions are ignored, maybe aim to wait for dht finish response signal
        if((i>=4)&&(i%2==0)){
            dht11_val[j/8]<<=1;                      //write 1 bit to 0 by moving left (auto add 0)
            if(counter>16)                           //long mean 1
                dht11_val[j/8]|=1;                   //write 1 bit to 1
            j++;
        }
    }
    // verify checksum and print the verified data
    if((j>=40)&&(dht11_val[4]==((dht11_val[0]+dht11_val[1]+dht11_val[2]+dht11_val[3])& 0xFF))){
        uart_string("RH:");
        hexstrings(dht11_val[0]);
```

```
        uart_string("TEMP:");
        hexstrings(dht11_val[2]);

        return 1;
    }
    else
        return 0;
}
```

最终的实验结果如下：

如上图所示，左侧两个数字为湿度，右侧为温度。

六、讨论与心得

这次实验颇为复杂，由于对 μcOS 如何使用串口并不了解，所以在寻找相关资料上花费了较多的时间。并且由于之前没有使用过 dht11 和数码管，在如何使用 gpio 控制这些器件上也花费了较多的精力。但最终结果还算可以接受，并且熟悉了 μcOS 的特性。