

Assignment 007: Lab 7: 室温计

一、实验目的

- 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、硬件

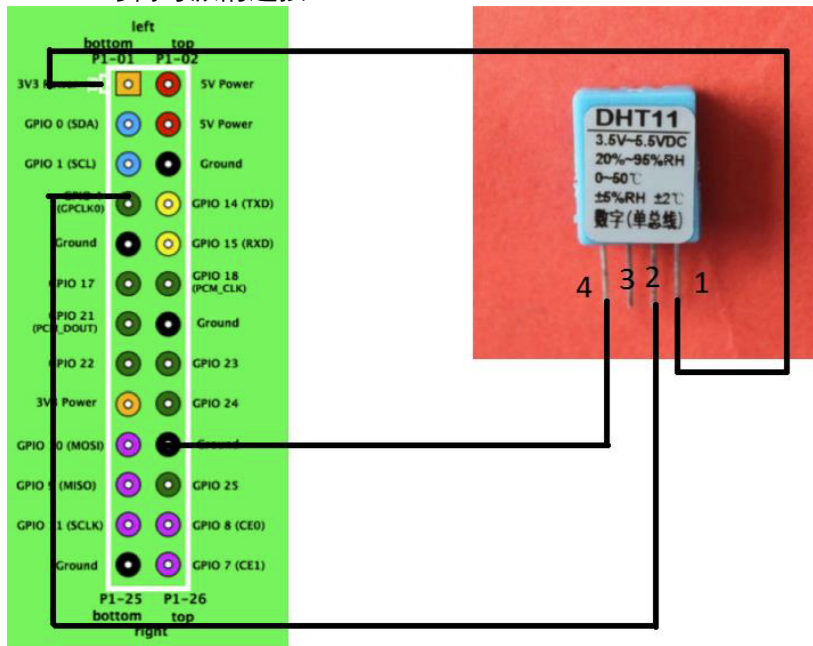
- Raspberry板一块；
- 5V/1A电源一个；
- microUSB线一根；
- 面包板一块；
- 两位7段数码管（共阳）一颗；
- 360Ω 1/8W电阻2颗；
- DHT-11 温湿度传感器1个；
- 面包线若干。

三、软件

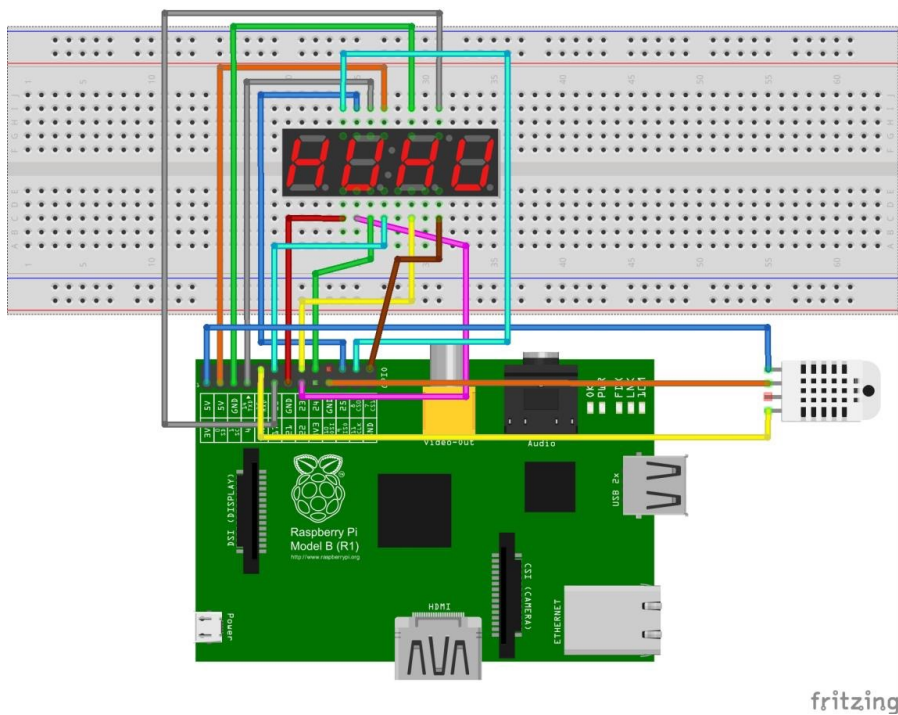
- 编译软件；
- Fritzing。

四、实验步骤

1 dht11与树莓派的连接



2 电路连接



3 ucos调用程序

需要对GPIO接口直接操作，需要定义如下变量：

```
#define BCM2708_PERI_BASE 0x20000000
#define GPIO_PADS
#define CLOCK_BASE
#define GPIO_BASE
#define GPIO_TIMER
#define GPIO_PWM
(BCM2708_PERI_BASE + 0x00100000)
(BCM2708_PERI_BASE + 0x00101000)
(BCM2708_PERI_BASE + 0x00200000)
(BCM2708_PERI_BASE + 0x00000000)
(BCM2708_PERI_BASE + 0x0020C000)
```

4 初始化借口代码

```
void init(){
    int i = 0;
    unsigned int base = GET32(GPIO_BASE);
    set_output(0, &base);
    set_output(1, &base);
    set_output(2, &base);
    set_output(3, &base);
    set_output(4, &base);
    set_output(5, &base);
    set_output(6, &base);
    set_output(7, &base);
    set_output(8, &base);
    set_output(9, &base);
    set_output(10, &base);
    set_output(11, &base);
    set_input(DHT, &base);
}

void set_output(unsigned int p, unsigned int *base){
    p %= 10;
```

```

    *base = ((0xffffffff - (111<<(p*3)))&(*base))|(1<<(p*3));
}
void set_input(unsigned int p, unsigned int *base){
    p %= 10;
    *base = ((0xffffffff - (111<<(p*3)))&(*base));
}

```

5 显示程序代码如下

```

void led(){
    led_one(data, 0);
    led_one(data/10, 1);
    led_one(data/100, 2);
    led_one(data/1000, 3);
}
void led_one(int data, int count){
    int i;
    data %= 10;
    PUT32(GPIO_SET, 1<<pin[count]);
    for(i=0;i<8;i++){
        if((~digit[num])&seg[i]){
            PUT32(GPIO_CLR, i<<pin[i+4]);
            OSTimeDlyHMSM(0,0,0,1);
            PUT32(GPIO_SET, 1<<pin[i+4]);
        }
    }
    PUT32(GPIO_CLR, 1<<pin[row]);
}

```

将上述 DHT11 的程序改为直接操控 GPIO 的模式,之后打开 uc0s 的 usrApp 文件夹下的 userApp.c 文件,将显示和读取分别粘贴到 userApp1 和 userApp2 程序中,然后重新编译运行。

6 实验结果

左边为湿度, 右边为温度。

