Assignment 007: Lab 7：室温计

3120102153　傅益芳

7 室温计

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。

配合课程

第七次：RTOS

实验目的

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

实验器材

硬件

pcDuino v2 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。

以下为自备（可选）器材：

PC（Windows/Mac OS/Linux）一台；
USB-TTL 串口线一根（FT232RL 芯片或 PL2303 芯片）；
以太网线一根（可能还需要路由器等）；
1602 LCD（带配套的 5k 微调电阻）。

软件

编译软件；
Fritzing。

实验步骤

设计输出方案，画连线示意图；

- 在面包板上连线，完成外部电路；
- 编写 C/C++程序，测试程序和电路；

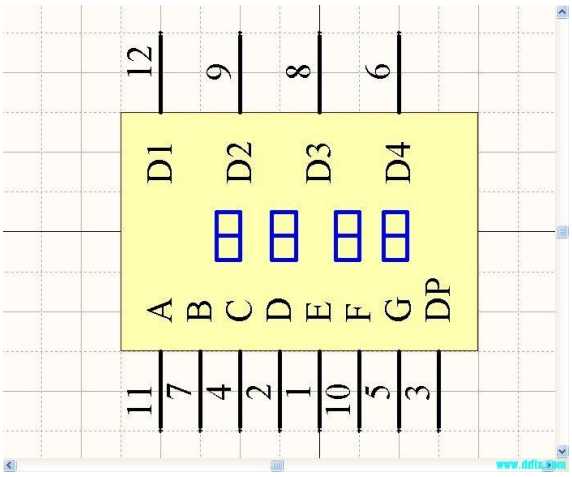
a. 测试、实现 uC/OS II 对 GPIO 的访问；
b. 实现 DHT-11 数据的读；
c. 实现以时分复用方式在四位 7 段数码管上依次显示 0000-9999 的数字；
d. 用两个 uc/OS II 任务，一个定时读 DHT-11 数据，一个轮流驱动数码管，一秒一次显示当 前温度和湿度。注意处理 好两个任务之间的数据共享。

自选扩展内容

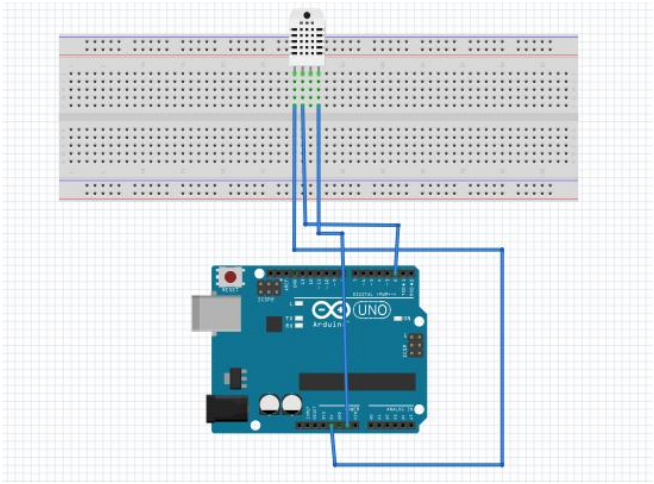
在 LCD 上显示。

实验步骤

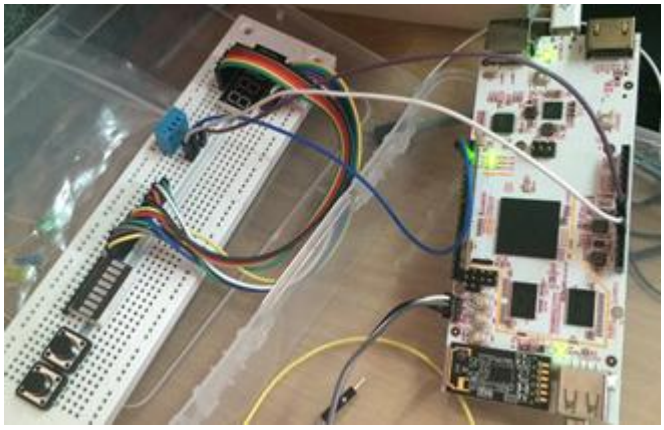
1、 数码管原理图



2、DHT11 连接图 ：

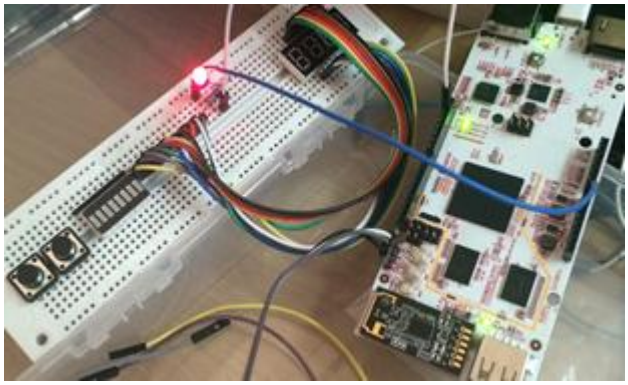


3、实际连接图



4、测试、实现 uC/OS II 对 GPIO 的访问
参考教程 http://www.oschina.net/question/1425530_144141

使用 LED 灯进行测试，闪烁效果如下



5、使用 digitalwrite 进行 IO 的 LOW、HIGH 控制

```
    void MyTask(void*p_arg )
    {

        char* sTaskName = (char*)p_arg;
        static flag1 =1;
        int a=-1;
    #if OS_CRITICAL_METHOD ==3
        OS_CPU_SR    cpu_sr =0;
    #endif

        while(1)
        {
            /* printf uses mutex to get terminal access,
therefore must enter critical section */
            OS_ENTER_CRITICAL();
            printf("Name: %s\n", sTaskName );
            if(!strcmp(sTaskName,"Task 1"))
```

```
            {
                if(flag1 ==1)
                {
                    flag1 =0;
                    printf("HIGH\n");
                    digitalWrite(led_pin, HIGH);
                    a=digitalRead(btn_pin);
                    printf("a:%d\n",a);
                }
                else
                {
                    flag1 =1;
                    printf("LOW\n");
                    digitalWrite(led_pin, LOW);
                    a=digitalRead(btn_pin);
                    printf("a:%d\n",a);
                }
            }
            OS_EXIT_CRITICAL();

            /* Delay so other tasks may execute. */
            OSTimeDly(50);
        }/* while */

}
```

6、完整代码见 cs.c

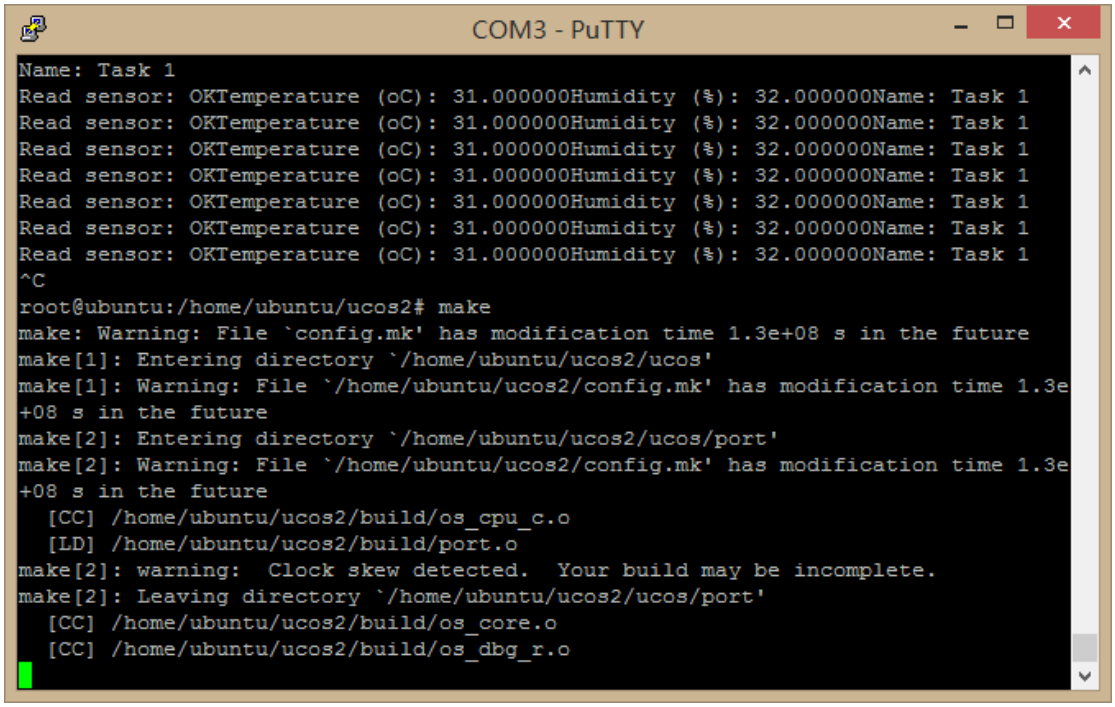
7、实现 DHT-11 数据的读；

改写 DHT-11 的库，使得该库适合 C 运行；

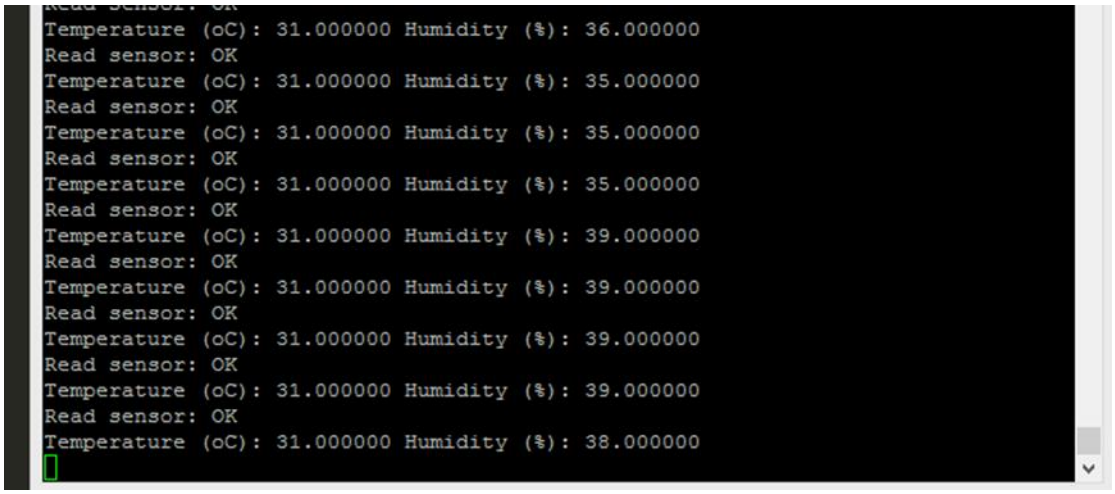
```
int f_read(int pin)
{
    // BUFFER TO RECEIVE
    uint8_t bits[5];
    uint8_t cnt = 7;
    uint8_t idx = 0;
    int i=0;
    // ACKNOWLEDGE or TIMEOUT
    unsigned int loopCnt = TIMEOUTLIMIT;
    unsigned long t ;
    uint8_t sum ;
    // EMPTY BUFFER
    for (i=0; i< 5; i++) bits[i] = 0;
    // REQUEST SAMPLE
    pinMode(pin, OUTPUT);
```

```
    digitalWrite(pin, LOW);
    delay(12);
    digitalWrite(pin, HIGH);
    delayMicroseconds(40);
    pinMode(pin, INPUT);
    // printf("1\n");
    loopCnt = TIMEOUTLIMIT;
    while(digitalRead(pin) == LOW)
        if (loopCnt-- == 0) return DHTLIB_ERROR_TIMEOUT;
    // printf("2\n");
    loopCnt = TIMEOUTLIMIT;
    while(digitalRead(pin) == HIGH)
        if (loopCnt-- == 0) return DHTLIB_ERROR_TIMEOUT;
    // READ OUTPUT - 40 BITS => 5 BYTES or TIMEOUT
    for (i=0; i<40; i++)
    {
        // printf("%d-1\n",i);
        loopCnt = TIMEOUTLIMIT;
        while(digitalRead(pin) == LOW)
            if (loopCnt-- == 0) return DHTLIB_ERROR_TIMEOUT;
        t = micros();
        // printf("%d-2\n",i);
        loopCnt = TIMEOUTLIMIT;
        while(digitalRead(pin) == HIGH)
            if (loopCnt-- == 0) return DHTLIB_ERROR_TIMEOUT;
        // printf("%d-3\n",i);
        if ((micros() - t) > 40) bits[idx] |= (1 << cnt);
        if (cnt == 0)   // next byte?
        {
            cnt = 7;    // restart at MSB
            idx++;      // next byte!
        }
        else cnt--;
    }
    humidity    = bits[0];
    temperature = bits[2];
    sum = bits[0] + bits[2];
    if (bits[4] != sum) return DHTLIB_ERROR_CHECKSUM;
    return DHTLIB_OK;
}
```

重新编译 ucosII，生成可运行文件：ucos-sample



运行进行程序测试



完整代码请参考 DHT11.c

8、 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

```
if(!strcmp(sTaskName,"Task 2"))
      {
          n++;
          if(n>99999)
              n=0;
          Cacul();
      }
```

9、 数码管显示

```
    void Cacul(){
        // int h=time_s;
        // n=888888;
```

```
      // printf("%d\n",n);
    clearLEDs();
    pickDigit(1);
    pickNumber((n/x/1000)%10);
    delayMicroseconds(del);

    clearLEDs();
    pickDigit(2);
    pickNumber((n/x/100)%10);
    delayMicroseconds(del);

    clearLEDs();
    pickDigit(3);
    // dispDec(3);
    pickNumber((n/x/10)%10);
    delayMicroseconds(del);

    clearLEDs();
    pickDigit(4);
    pickNumber(n/x%10);
    delayMicroseconds(del);
}
```

10、 分时复用显示数码管

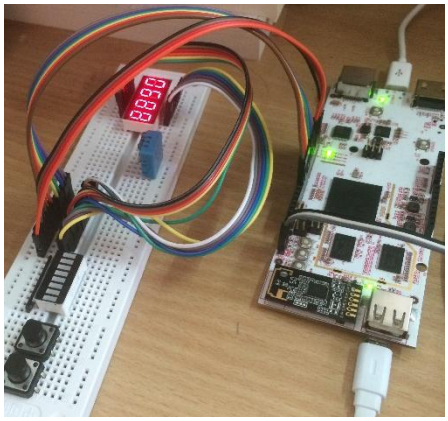
```
  void pickDigit(int x)  //定义pickDigit(x),其作用是开
启dx端口
  {
    digitalWrite(d1, LOW);
    digitalWrite(d2, LOW);
    digitalWrite(d3, LOW);
    digitalWrite(d4, LOW);

    switch(x)
    {
    case 1:
      digitalWrite(d1, HIGH);
      break;
    case 2:
      digitalWrite(d2, HIGH);
      break;
    case 3:
      digitalWrite(d3, HIGH);
```

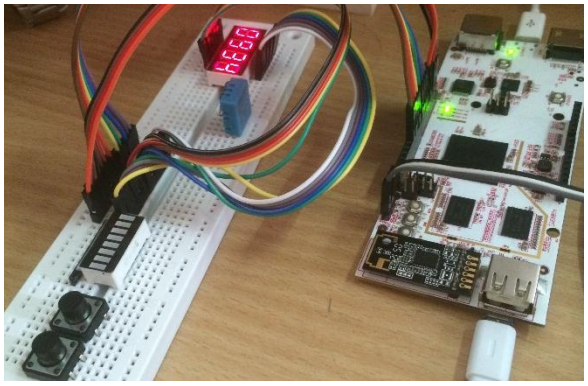
```
        break;
      default:
        digitalWrite(d4, HIGH);
        break;
      }
    }
```

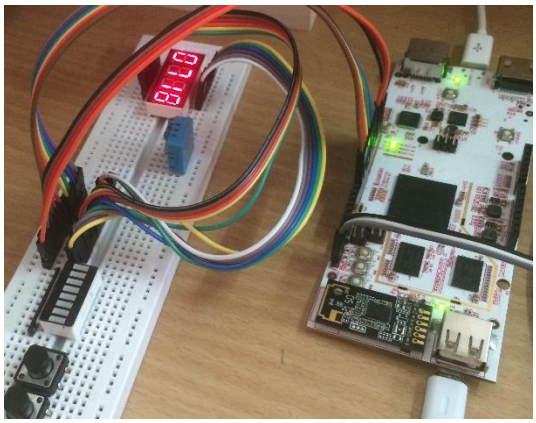
11、 八段数码管代码

```
    void one()  //定义数字 1 时阴极那些管脚开关
    {
      digitalWrite(a, HIGH);
      digitalWrite(b,LOW);
      digitalWrite(c,LOW);
      digitalWrite(d, HIGH);
      digitalWrite(e, HIGH);
      digitalWrite(f, HIGH);
      digitalWrite(g, HIGH);
}
```

12、 实际运行展示

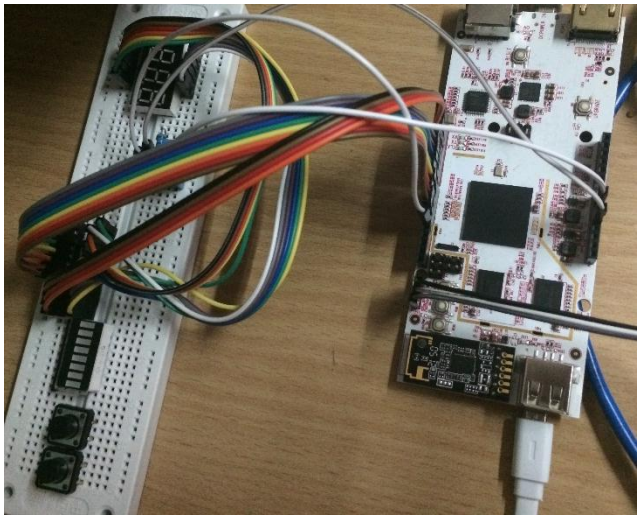






13、 用两个 uc/OS II 任务，一个定时读 DHT-11 数据，一个轮流驱动数码管，一秒一次显示当 前温度和湿度。

实际连接图



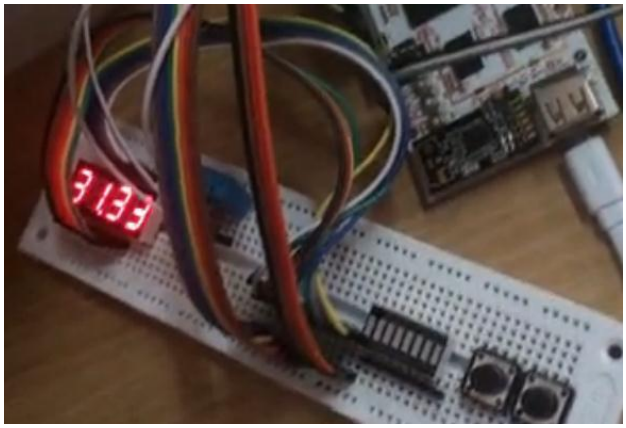
14、 创建两个任务，能够并行运行

```
    hardware_init();

    OSTaskCreate( MyTask, sTask1, (void*)Stk1,4);
```

设置温度湿度为全局变量，实现数据共享

```
int humidity;
int temperature;
```

15、 运行效果



```
Temperature (oC): 31.000000 Humidity (%): 32.000000
Read sensor: OK
Temperature (oC): 31.000000 Humidity (%): 32.000000
Read sensor: Time out error
Temperature (oC): 31.000000 Humidity (%): 32.000000
Read sensor: OK
Temperature (oC): 31.000000 Humidity (%): 32.000000
Read sensor: OK
Temperature (oC): 31.000000 Humidity (%): 32.000000
Read sensor: OK
Temperature (oC): 31.000000 Humidity (%): 32.000000
Read sensor: Time out error
Temperature (oC): 31.000000 Humidity (%): 32.000000
Read sensor: OK
Temperature (oC): 31.000000 Humidity (%): 32.000000
Read sensor: Time out error
Temperature (oC): 31.000000 Humidity (%): 32.000000
Read sensor: OK
Temperature (oC): 31.000000 Humidity (%): 32.000000
Read sensor: OK
Temperature (oC): 31.000000 Humidity (%): 32.000000
```

16、 视频

http://v.youku.com/v_show/id_XMTI2NjE4NzY5Ng==.html