Assignment 007: Lab 7：室温计

毛晨炀 3120102120

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。

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 1 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）；

软件

- 编译软件；
- Fritzing。

实验要求

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。

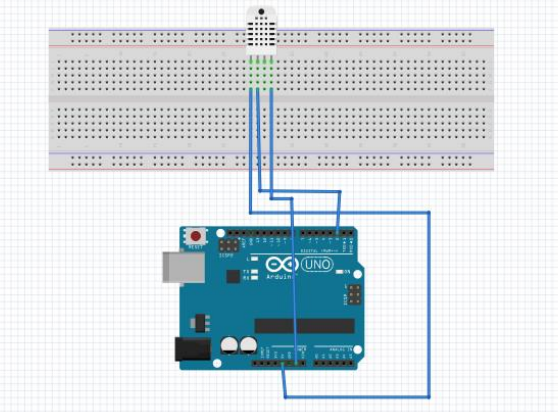
实验报告要求

1. 画出你所实际实施的连接示意图；
2. 给出所用的器材的列表；
3. 用 Fritzing 画出外部设备的连线图，附实物照片；
4. 描述所做的实验步骤，给出各步操作的命令和结果；
5. 给出代码并解释；
6. 将所做作品拍摄视频上传到优酷，给出优酷的视频网址；
7. 说明其他所做的扩展内容的情况。

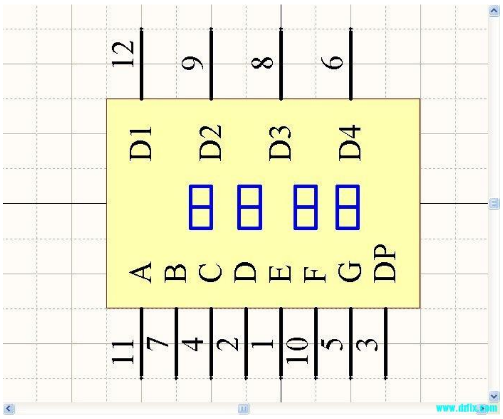
实验步骤:

1. 设计输出方案，画连线示意图；

DHT11 连接图：

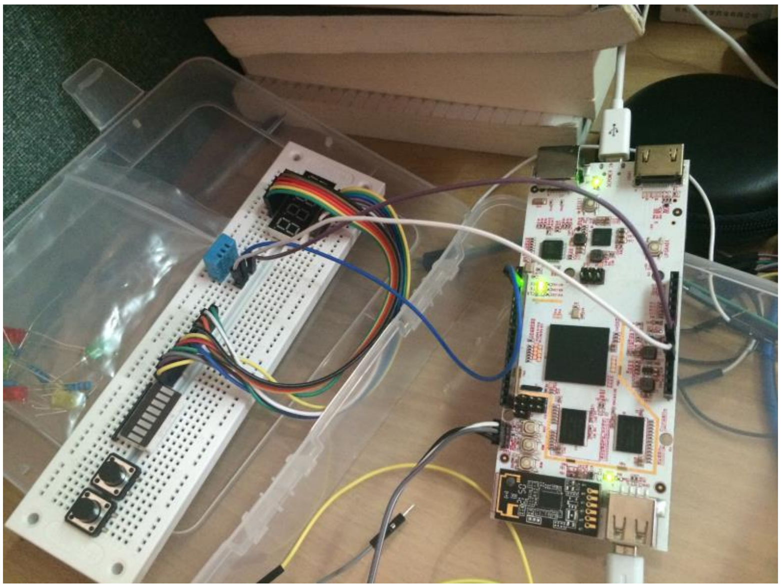


数码管原理图：

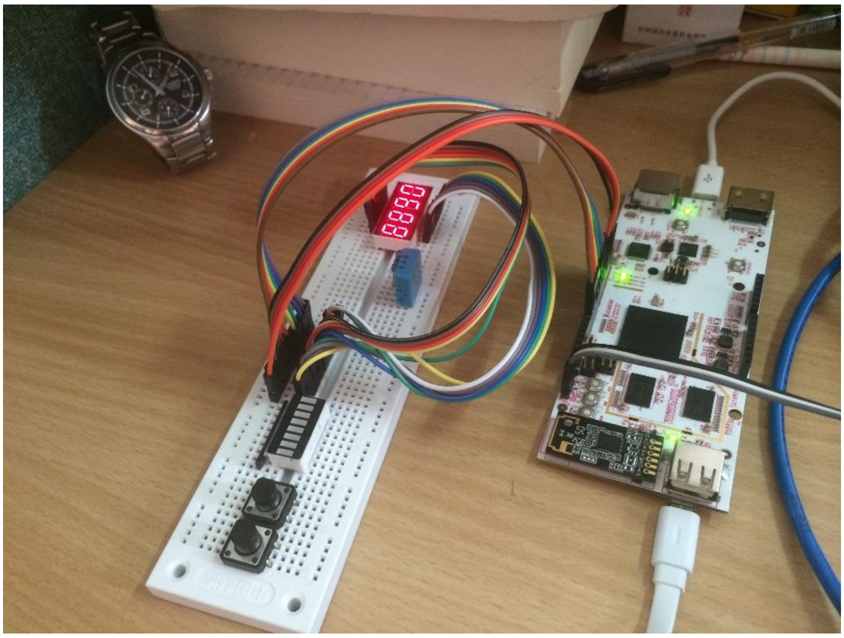


2. 在面包板上连线，完成外部电路；

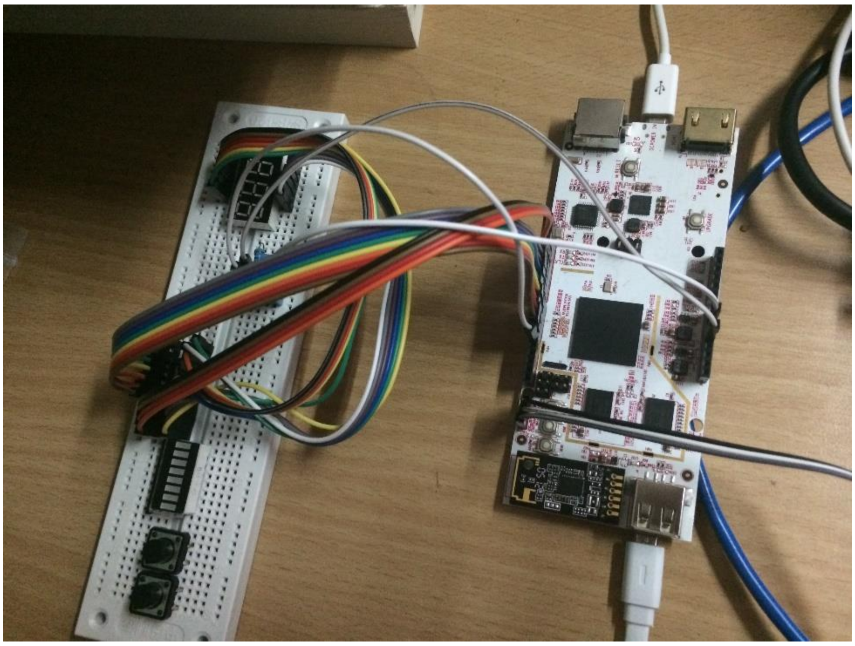
实际 DHT11 连接图：



数码管连接图：



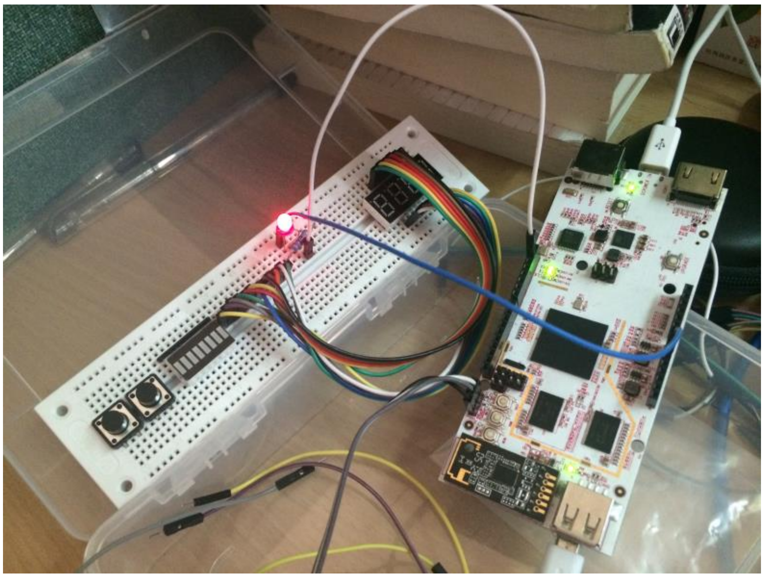
数码管 DHT11 连接图：



3. 编写 C/C++程序，测试程序和电路；

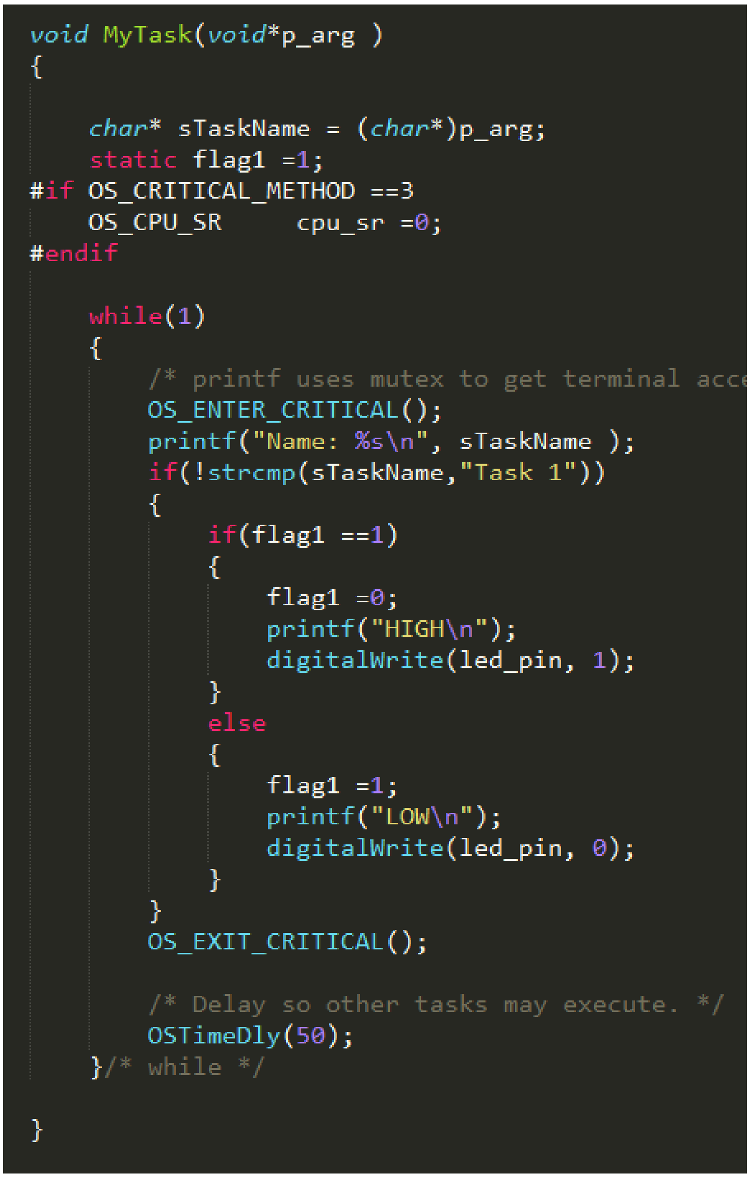
3.1. 测试、实现 uC/OS II 对 GPIO 的访问；

   a) 参考网络教程——在 pcDuino 上学习 μC/OS II：http://www.oschina.net/question/1425530_144141
   b) 使用 LED 灯进行测试，闪烁效果如下：



c)　关键代码如下：

使用 digitalwrite 进行 IO 的 LOW、HIGH 控制。



```
void MyTask(void*p_arg )
{

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

    while(1)
    {
        /* printf uses mutex to get terminal acce
        OS_ENTER_CRITICAL();
        printf("Name: %s\n", sTaskName );
        if(!strcmp(sTaskName,"Task 1"))
        {
            if(flag1 ==1)
            {
                flag1 =0;
                printf("HIGH\n");
                digitalWrite(led_pin, 1);
            }
            else
            {
                flag1 =1;
                printf("LOW\n");
                digitalWrite(led_pin, 0);
            }
        }
        OS_EXIT_CRITICAL();

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

}
```

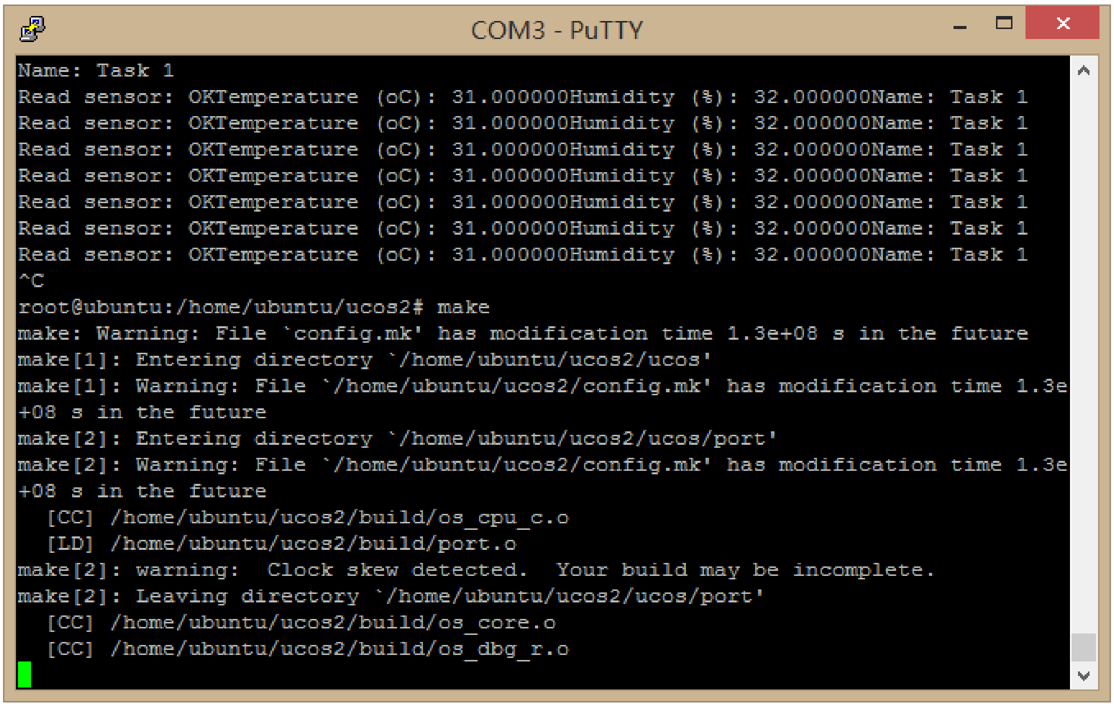
d) 完整代码请参考附件 ledcs.c

3.2. 实现 DHT-11 数据的读；

a) 改写 DHT-11 的库，使得该库适合 C 运行；

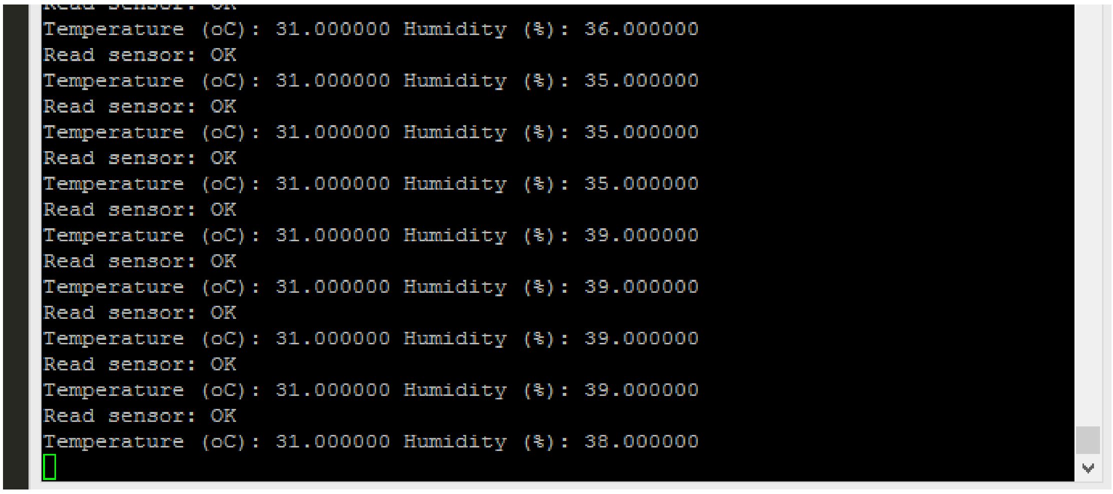
```
int f_read(int pin)
{
    // BUFFER TO RECEIVE
    uint8_t bits[5];
    uint8_t cnt = 7;
    uint8_t idx = 0;
    int i=0;
    // ACKNOWLEDGE or TIMEOUT
    unsigned int loopCnt = TIMEOUTLIMIT;
    unsigned long t ;
    uint8_t sum ;
    // EMPTY BUFFER
    for (i=0; i< 5; i++) bits[i] = 0;
    //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

e)　重新编译 ucosII，生成可运行文件：ucos-sample



```
Name: Task 1
Read sensor: OKTemperature (oC): 31.000000Humidity (%): 32.000000Name: Task 1
Read sensor: OKTemperature (oC): 31.000000Humidity (%): 32.000000Name: Task 1
Read sensor: OKTemperature (oC): 31.000000Humidity (%): 32.000000Name: Task 1
Read sensor: OKTemperature (oC): 31.000000Humidity (%): 32.000000Name: Task 1
Read sensor: OKTemperature (oC): 31.000000Humidity (%): 32.000000Name: Task 1
Read sensor: OKTemperature (oC): 31.000000Humidity (%): 32.000000Name: Task 1
Read sensor: OKTemperature (oC): 31.000000Humidity (%): 32.000000Name: Task 1
^C
root@ubuntu:/home/ubuntu/ucos2# make
make: Warning: File `config.mk' has modification time 1.3e+08 s in the future
make[1]: Entering directory `/home/ubuntu/ucos2/ucos'
make[1]: Warning: File `/home/ubuntu/ucos2/config.mk' has modification time 1.3e
+08 s in the future
make[2]: Entering directory `/home/ubuntu/ucos2/ucos/port'
make[2]: Warning: File `/home/ubuntu/ucos2/config.mk' has modification time 1.3e
+08 s in the future
  [CC] /home/ubuntu/ucos2/build/os_cpu_c.o
  [LD] /home/ubuntu/ucos2/build/port.o
make[2]: warning:  Clock skew detected.  Your build may be incomplete.
make[2]: Leaving directory `/home/ubuntu/ucos2/ucos/port'
  [CC] /home/ubuntu/ucos2/build/os_core.o
  [CC] /home/ubuntu/ucos2/build/os_dbg_r.o
```

f)　运行进行程序测试：



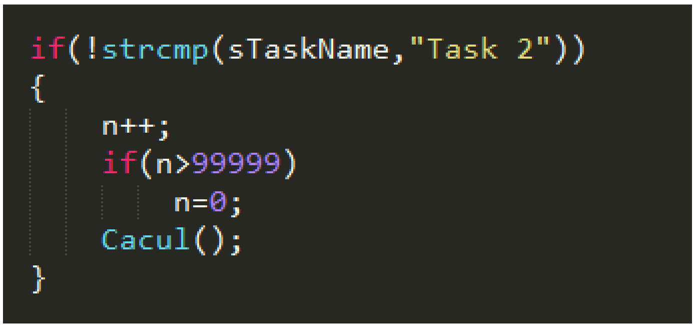
```
Read sensor: OK
Temperature (oC): 31.000000 Humidity (%): 36.000000
Read sensor: OK
Temperature (oC): 31.000000 Humidity (%): 35.000000
Read sensor: OK
Temperature (oC): 31.000000 Humidity (%): 35.000000
Read sensor: OK
Temperature (oC): 31.000000 Humidity (%): 35.000000
Read sensor: OK
Temperature (oC): 31.000000 Humidity (%): 39.000000
Read sensor: OK
Temperature (oC): 31.000000 Humidity (%): 39.000000
Read sensor: OK
Temperature (oC): 31.000000 Humidity (%): 39.000000
Read sensor: OK
Temperature (oC): 31.000000 Humidity (%): 39.000000
Read sensor: OK
Temperature (oC): 31.000000 Humidity (%): 38.000000
```

g) 完整代码请参考 DHT11.c

3.3. 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；

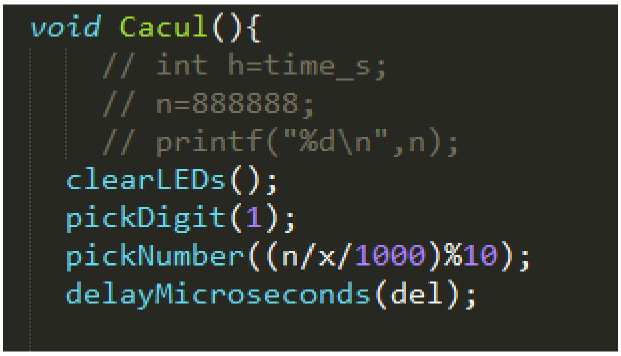
a) 关键代码展示：

首先定义为 Task 2，n 用来累加，由于累加过快，采用除以 10 的方式使得变化速度能够接受：



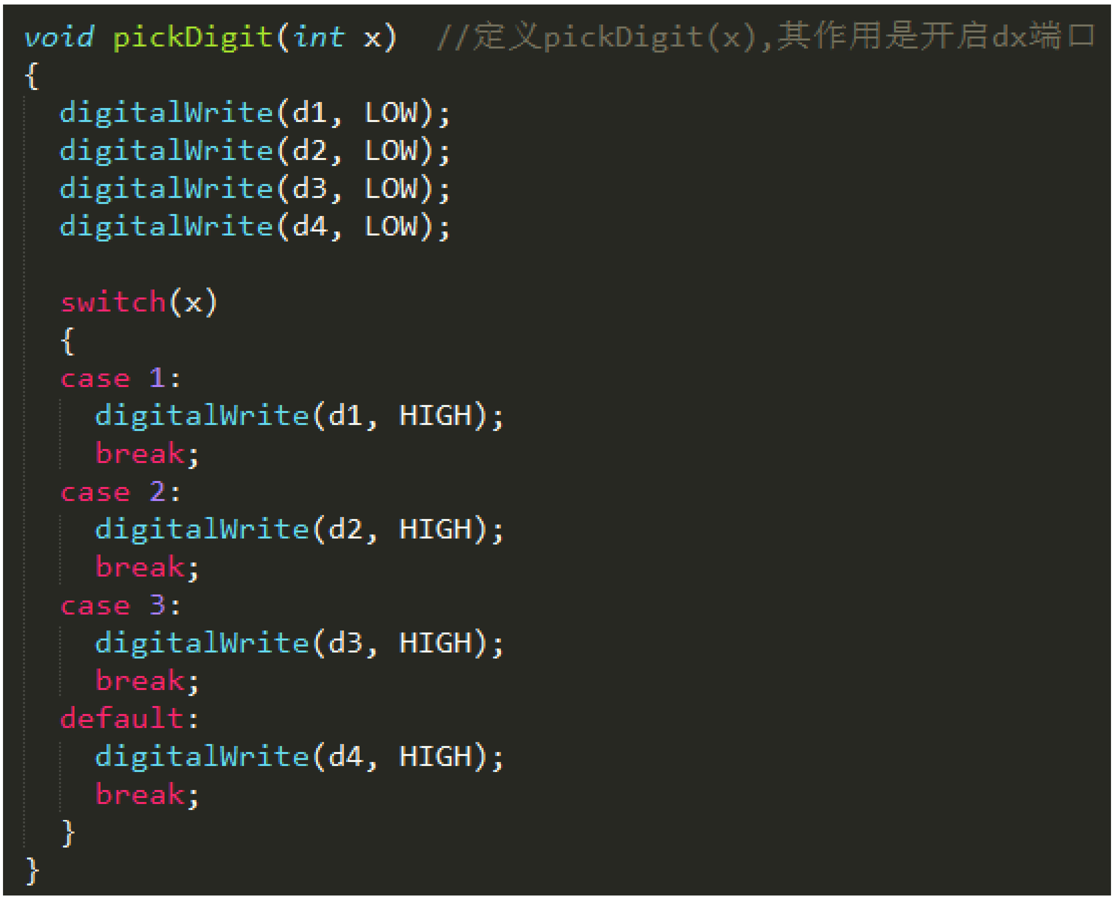
```
if(!strcmp(sTaskName,"Task 2"))
{
    n++;
    if(n>99999)
        n=0;
    Cacul();
}
```

b) 数码管显示：



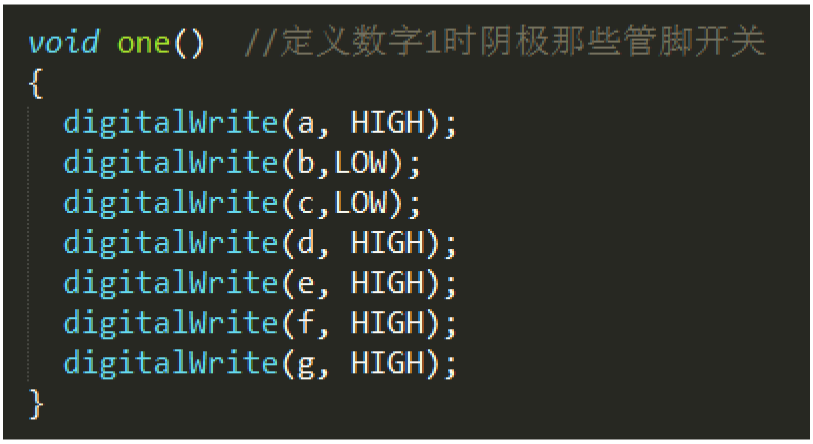
```
void Cacul(){
    // int h=time_s;
    // n=888888;
    // printf("%d\n",n);
  clearLEDs();
  pickDigit(1);
  pickNumber((n/x/1000)%10);
  delayMicroseconds(del);
```

c) 分时复用显示数码管：



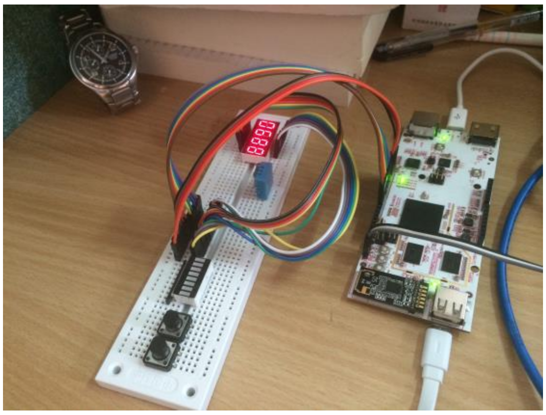
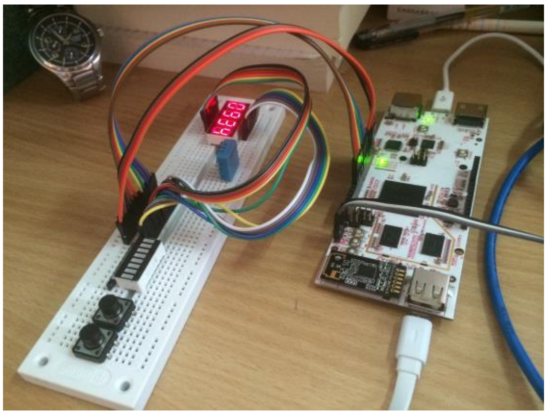
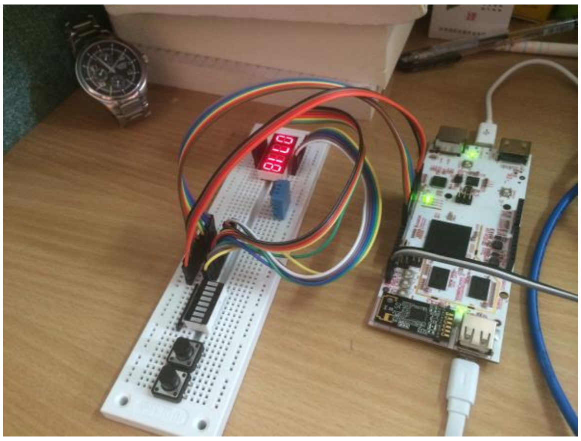
```
void pickDigit(int x)  //定义pickDigit(x),其作用是开启dx端口
{
  digitalWrite(d1, LOW);
  digitalWrite(d2, LOW);
  digitalWrite(d3, LOW);
  digitalWrite(d4, LOW);

  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;
  }
}
```

d) 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);
}
```

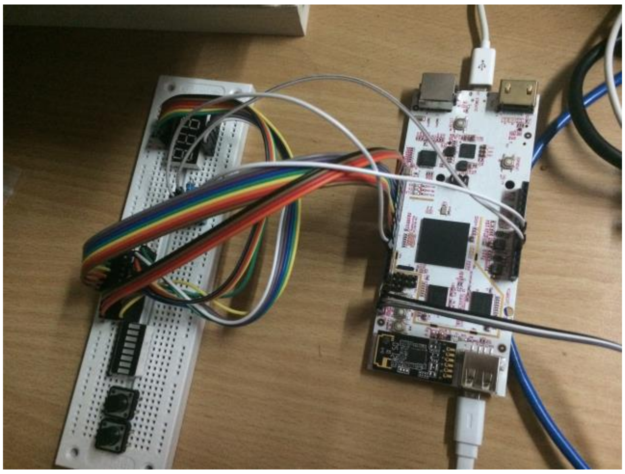
e) 实际运行展示：

f)　完整代码参见附件。

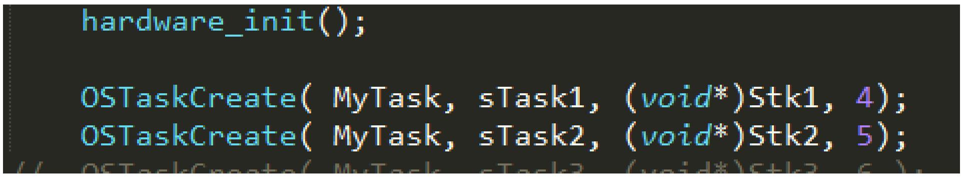
3.4.　用两个 uc/OS II 任务，一个定时读 DHT-11 数据，一个轮流驱动数码管，一秒一次显示当 前温度和湿度。注意处理 好两个任务之间的数据共享。

a)　实际连接图：



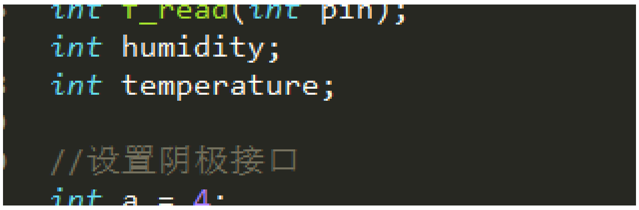
b)　关键代码：

创建两个任务，能够并行运行：



```
    hardware_init();

    OSTaskCreate( MyTask, sTask1, (void*)Stk1, 4);
    OSTaskCreate( MyTask, sTask2, (void*)Stk2, 5);
//  OSTaskCreate( MyTask, sTask3, (void*)Stk3, 6 );
```

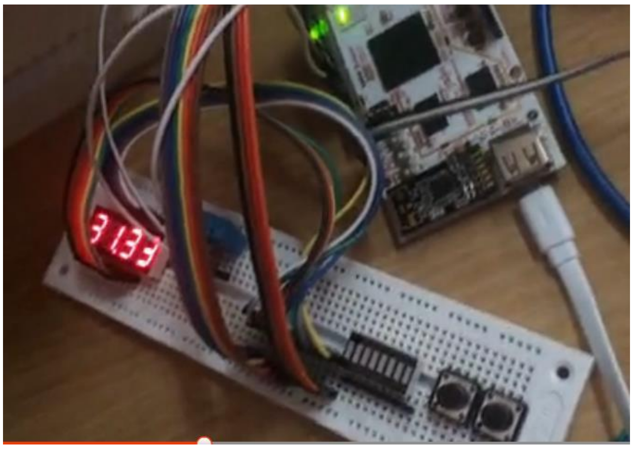
设置温度湿度为全局变量，实现数据共享：



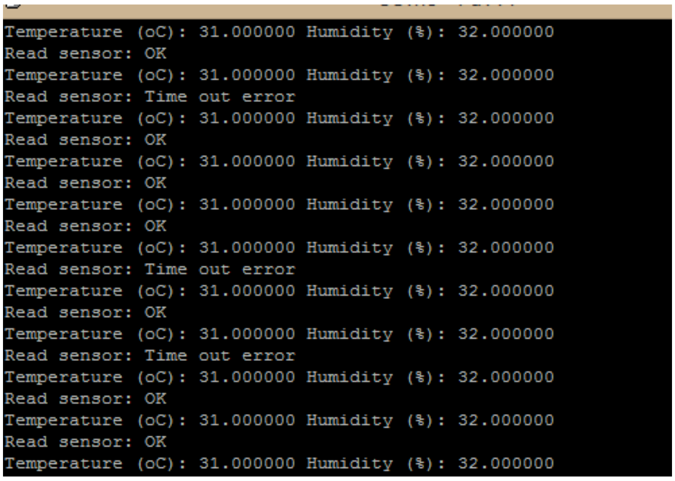
```
int f_read(int pin);
int humidity;
int temperature;

//设置阴极接口
int a = 4;
```

c)　运行效果：

PCDuino 上：



PC 端：



```
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
```

d)　完整代码请看附件。

4.　拍摄的视频优酷地址：

PCduino 室温计　ucos 应用

http://v.youku.com/v_show/id_XNzIzOTAzODgw.html

PCduino 室温计　ucos 应用　文字清晰版

http://v.youku.com/v_show/id_XNzIzOTA2MDIw.html