

浙江大学实验报告

课程名称：____ 嵌入式开发 ____ 指导老师：____ 翁凯 ____ 成绩：____
实验名称：____ 室温计 ____ 实验类型：____ 基础实验 ____

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

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；
- $360\ \Omega$ $1/8W$ 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

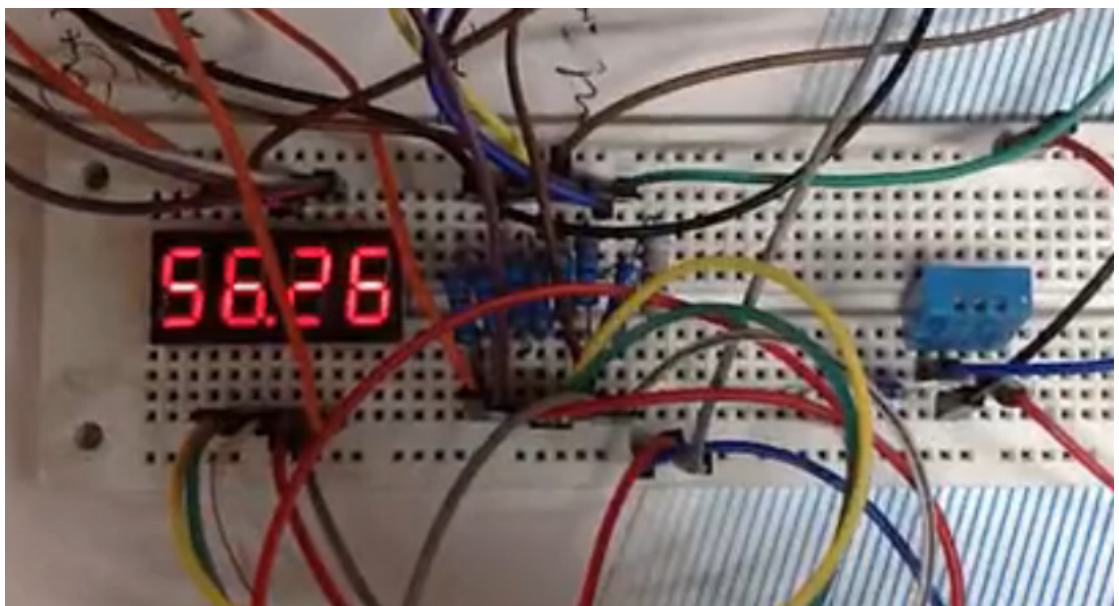
- 编译软件；
- Fritzing。

实验步骤

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；
- wrtnode 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；
 - $360\ \Omega$ $1/8W$ 电阻 2 颗；
 - DHT-11 温湿度传感器 1 个；
 - 面包线若干。
-
- PC (Windows) 一台；
 - USB-TTL 串口线一根（PL2303 芯片）；
 - 以太网线一根；
 - 1602 LCD（带配套的 5k 微调电阻）。

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；

首先从 github 上下载 uc0s-ii-for-pcduino 的代码，我们在此基础上做 wrtnode 的移植：

通过文件共享放到 wrtnode 中。

```
[zhenyuxiaoge@localhost uc0s-ii-for-pcduino-master]$ ls
src  arduino  build  config.mk  Makefile  README.md  uc0s  uc0s_2.86_original
[zhenyuxiaoge@localhost uc0s-ii-for-pcduino-master]$
```

修改 ./arduino/makefile 文件和 ./makefile 文件，用
mipsel-openwrt-linux-xxx 工具链作交叉编译。

```
#
# Assume standalone toolchain
#
#CC = mipsel-openwrt-linux-gcc
#CXX = mipsel-openwrt-linux-g++
#AR = mipsel-openwrt-linux-ld

DIR=$(shell pwd)

INCLUDES = \
    -I$(DIR) \
    -I$(DIR)/hardware \
    -I$(DIR)/hardware/arduino \
    -I$(DIR)/hardware/arduino/cores \
    -I$(DIR)/hardware/arduino/cores/arduino \
    -I$(DIR)/hardware/arduino/variants \
    -I$(DIR)/hardware/arduino/variants/sunxi \
    -I$(DIR)/libraries \
    -I$(DIR)/libraries/SPI \
    -I$(DIR)/libraries/Wire \
    -I$(DIR)/libraries/PN532_SPI

CFLAGS = -fPIC
-- INSERT --
```

12,21-28 Top

返回上层目录 make

```

[zhenyuxiaoge@localhost ucos-ii-for-pcDuino-master]$ cd arduino/
[zhenyuxiaoge@localhost arduino]$ ls
hardware  libraries  Makefile  sample
[zhenyuxiaoge@localhost arduino]$ vim Makefile
[zhenyuxiaoge@localhost arduino]$ cd ..
[zhenyuxiaoge@localhost ucos-ii-for-pcDuino-master]$ make
make[1]: Entering directory `/home/zhenyuxiaoge/EMLAB/ucos-ii-for-pcDuino-master/ucos-ii-for-pcDuino-master/ucos'
make[2]: Entering directory `/home/zhenyuxiaoge/EMLAB/ucos-ii-for-pcDuino-master/ucos-ii-for-pcDuino-master/ucos/port'
  [CC] /home/zhenyuxiaoge/EMLAB/ucos-ii-for-pcDuino-master/ucos-ii-for-pcDuino-master/build/os_cpu_c.o
  [LD] /home/zhenyuxiaoge/EMLAB/ucos-ii-for-pcDuino-master/ucos-ii-for-pcDuino-master/build/port.o
make[2]: Leaving directory `/home/zhenyuxiaoge/EMLAB/ucos-ii-for-pcDuino-master/ucos-ii-for-pcDuino-master/ucos/port'
  [CC] /home/zhenyuxiaoge/EMLAB/ucos-ii-for-pcDuino-master/ucos-ii-for-pcDuino-master/build/os_core.o
  [CC] /home/zhenyuxiaoge/EMLAB/ucos-ii-for-pcDuino-master/ucos-ii-for-pcDuino-master/build/os_dbg_r.o
  [CC] /home/zhenyuxiaoge/EMLAB/ucos-ii-for-pcDuino-master/ucos-ii-for-pcDuino-master/build/os_flag.o
  [CC] /home/zhenyuxiaoge/EMLAB/ucos-ii-for-pcDuino-master/ucos-ii-for-pcDuino-master/build/os_mbox.o

```

参考了 <http://www.sunny-song.com/blog/> 宋博同学的 blog（感谢）。

DHT11 读取温度湿度，根据它的 datasheet 来编程。

<http://wenku.baidu.com/view/1955cc70a417866fb84a8e7b.html>

代码放到 ./app/sample.c 中，编译成功后，会出现 ./ucos_sample 文件，拷贝该文件和相关的动态链接库到板卡上，即可运行 sample 程序。

5. 给出代码并解释；

```

/*
*****

```

```
*****

*                                     sample.c

*

* Description:      This sample program uses the ucos linux port to
start 5 simple tasks.

*

* Author: Philip Mitchell

*

*****
*****

*/

#include <stdio.h>

#include <stdlib.h>

#include "ucos_ii.h"

#include <core.h>
```

```
#include <string.h>

void showDigit(uint8_t* digits);

uint8_t led_pins_slice[] = {41, 21, 0, 18}; //gpio 2 can not use,
cause reset

uint8_t led_pins_segment[] = {43, 72, 37, 19, 17, 40, 42, 44};

int32_t led_pins_slice_fd[4];

int32_t led_pins_segment_fd[8];

uint8_t data_pin = 20;

uint8_t data[5];

uint8_t dataAvailable = 0;

void hardware_init()

{

//set slice pins
```

```
uint8_t i = 0;

for (i = 0; i < 4; ++i)

{

    gpio_export(led_pins_slice[i]);

    gpio_set_dir(led_pins_slice[i], OUTPUT);

    gpio_set_value(led_pins_slice[i], LOW);

    led_pins_slice_fd[i] = gpio_fd_open(led_pins_slice[i]);

}


//set segment pins

for (i = 0; i < 8; ++i)

{
```



```
gpio_export(led_pins_segment[i]);

gpio_set_dir(led_pins_segment[i], OUTPUT);

gpio_set_value(led_pins_segment[i], HIGH);

led_pins_segment_fd[i] = gpio_fd_open(led_pins_segment[i]);

}

//set data pin

gpio_export(data_pin);

gpio_set_dir(data_pin, OUTPUT);

gpio_set_value(data_pin, HIGH);

}
```

```
/* Function common to all tasks */
```

```
void controlled(void *p_arg){
```

```
#if OS_CRITICAL_METHOD == 3
```

```
OS_CPU_SR      cpu_sr = 0;
```

```
#endif
```

```
uint8_t i;
```

```
while(1){
```

```
//OS_ENTER_CRITICAL();
```

```
uint8_t values[] = {0,0,0,0};
```

```
if(dataAvailable){
```

```
values[0] = data[2] / 10;
```

```
values[1] = data[2] % 10;
```

```
values[2] = data[0] / 10;
```

```
values[3] = data[0] % 10;
```

```
}
```

```
showDigit(values);
```

```
//OS_EXIT_CRITICAL();
```

```
}
```

```
}
```

```
void showDigit(uint8_t* digits){
```

```
//slice between 0 ~ 3
```

```
//digit between 0 ~ 9
```

```
uint8_t matrix[] = {0xD7, 0x14, 0xCD, 0x5D, 0x1E,  
  
0x5B, 0xDB, 0x15, 0xDF, 0x5F};  
  
uint8_t i;  
  
uint8_t j;  
  
for(i = 0; i < 4; i++){  
  
    //close led  
  
    for(j = 0; j < 4; j++){  
  
        gpio_set_value(led_pins_slice[j], LOW);  
  
    }  
  
    //set segment  
  
    for(j = 0; j < 8; j++){  
  
        uint8_t value = !(matrix[digits[i]] & (0x80 >> j));
```

```
gpio_set_value(led_pins_segment[j], value);
```

```
}
```

```
//chose slice
```

```
for(j = 0; j < 4; j++){
```

```
if(i == j){
```

```
gpio_set_value(led_pins_slice[j], HIGH);
```

```
}
```

```
else{
```

```
gpio_set_value(led_pins_slice[j], LOW);
```

```
}
```

```
}
```

```
OSTimeDly(1);
```

```
}
```

```
}
```

```
void readData( void *p_arg )
```

```
{
```

```
char* sTaskName = (char*)p_arg;
```

```
uint8_t bit;//bit value
```

```
uint8_t isError = 0;
```

```
uint32_t t;
```

```
uint8_t bitIndex, byteIndex;
```

```
uint8_t byte;
```

```
uint32_t i;
```

```
uint32_t timeCnt;
```

```
//      static flag1 = 1;
```

```
#if OS_CRITICAL_METHOD == 3
```

```
OS_CPU_SR      cpu_sr = 0;
```

```
#endif
```

```
//wait 1s for DHT11 to warm up - OK
```

```
OSTimeDly(201);
```

```
int fd = gpio_fd_open(data_pin);
```

```
while(1)

{

    isError = 0;

    gpio_set_dir(data_pin, OUTPUT);

    gpio_set_value(data_pin, HIGH);

    OSTimeDly(201);

    OS_ENTER_CRITICAL();

    //send start signal to DHT11

    gpio_set_value(data_pin, LOW);

    delay(20); //send low signal longer than 18ms, average 29ms - OK

    gpio_set_dir(data_pin, INPUT); //140us
```



```
lseek(fd, 0, SEEK_SET);
```

```
read(fd, &bit, 1);
```

```
//get ack bit
```

```
timeCnt = 10000;
```

```
while(bit != LOW){
```

```
//gpio_get_value(data_pin, &bit);
```

```
lseek(fd, 0, SEEK_SET);
```

```
read(fd, &bit, 1);
```

```
bit -= '0';
```

```
if(-timeCnt == 0){// 200us

isError = 1;

printf("didn\'t receive LOW ack from DHT11.\n");

break;

}

}

if(isError == 1){

continue;

}

//receive ack from DHT11

// t = micros();

timeCnt = 10000;
```

```
while(bit != HIGH){

//gpio_get_value(data_pin, &bit);

lseek(fd, 0, SEEK_SET);

read(fd, &bit, 1);

bit -= '0';

if(-timeCnt == 0){// 2000us

isError = 1;

printf("didn\'t receive HIGH ack from DHT11.\n");

break;

}

}

if(isError == 1){

continue;

}

timeCnt = 10000;
```

```
while(bit != LOW){

    lseek(fd, 0, SEEK_SET);

    read(fd, &bit, 1);

    bit -= '0';

    if(-timeCnt == 0){// 2000us

        isError = 1;

        printf("didn\'t receive LOW gap from DHT11.\n");

        break;

    }

}

if(isError == 1){
```

```
OS_EXIT_CRITICAL();

continue;

}

//omit some codes here...

//...

OS_EXIT_CRITICAL();

}

}

int main (void)

{

/* pthreads allocates its own memory for task stacks. This UCOS linux
port needs a minimum stack size

in order to pass the function information within the port. */
```

```
hardware_init();

INT8U Stk1[ OSMinStkSize() ];

INT8U Stk2[ OSMinStkSize() ];

INT8U Stk3[ OSMinStkSize() ];

INT8U Stk4[ OSMinStkSize() ];

INT8U Stk5[ OSMinStkSize() ];

char sTask1[] = "Task 1" ;

char sTask2[] = "Task 2" ;

char sTask3[] = "Task 3" ;

char sTask4[] = "Task 4" ;

//      char sTask5[] = "Task 5" ;

OSInit();

//delay 1s for DHT11 to warm up.

OSTaskCreate( readData, sTask1, (void*)Stk1, 0 );

OSTaskCreate( controlled, sTask2, (void*)Stk2, 5 );

//      OSTaskCreate( MyTask, sTask3, (void*)Stk3, 6 );

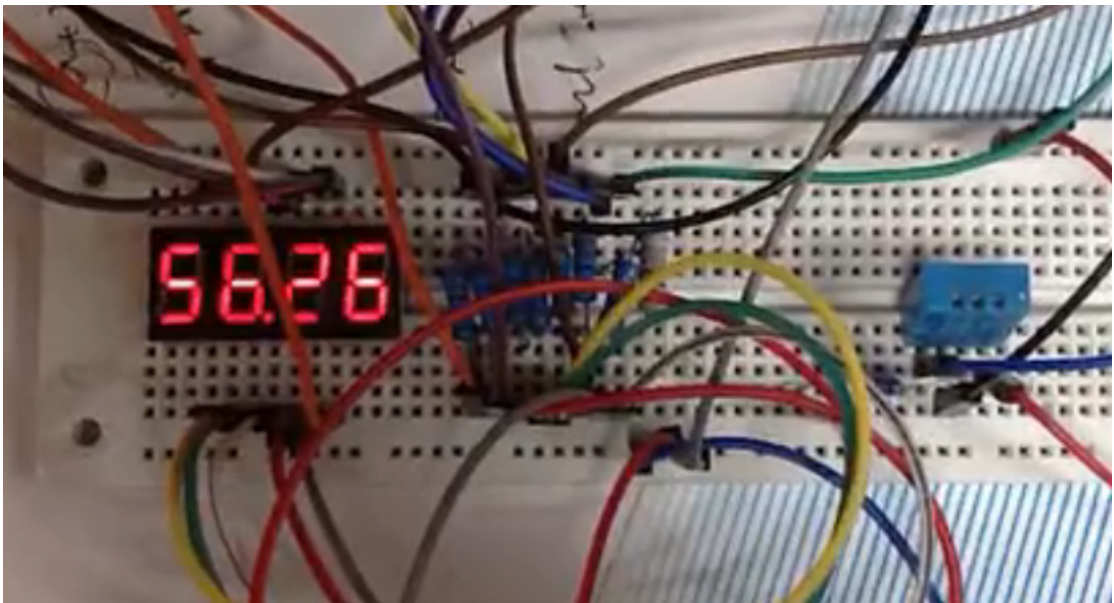
//      OSTaskCreate( MyTask, sTask4, (void*)Stk4, 7 );

//      OSTaskCreate( MyTask, sTask5, (void*)Stk5, 8 );

OSStart();
```

```
return 0;  
  
}
```

6. 将所做作品拍摄视频上传到优酷，给出优酷的视频网址；



(湿度 56，温度 26)

http://v.youku.com/v_show/id_XMTI3MTg4OTgwNA==.html?from=y1.7-1.2