

Lab 7 : 室温计

3120104198

张瑞祥

实验目的

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。

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

树莓派板一块；
5V/1A电源一个；
microUSB线一根；
USB-TTL串口线一根（FT232RL芯片或PL2303芯片）。
以下为自备（可选）器材：
PC(Windows/Mac OS/Linux)一台；
以太网线一根（可能还需要路由器等）。

软件

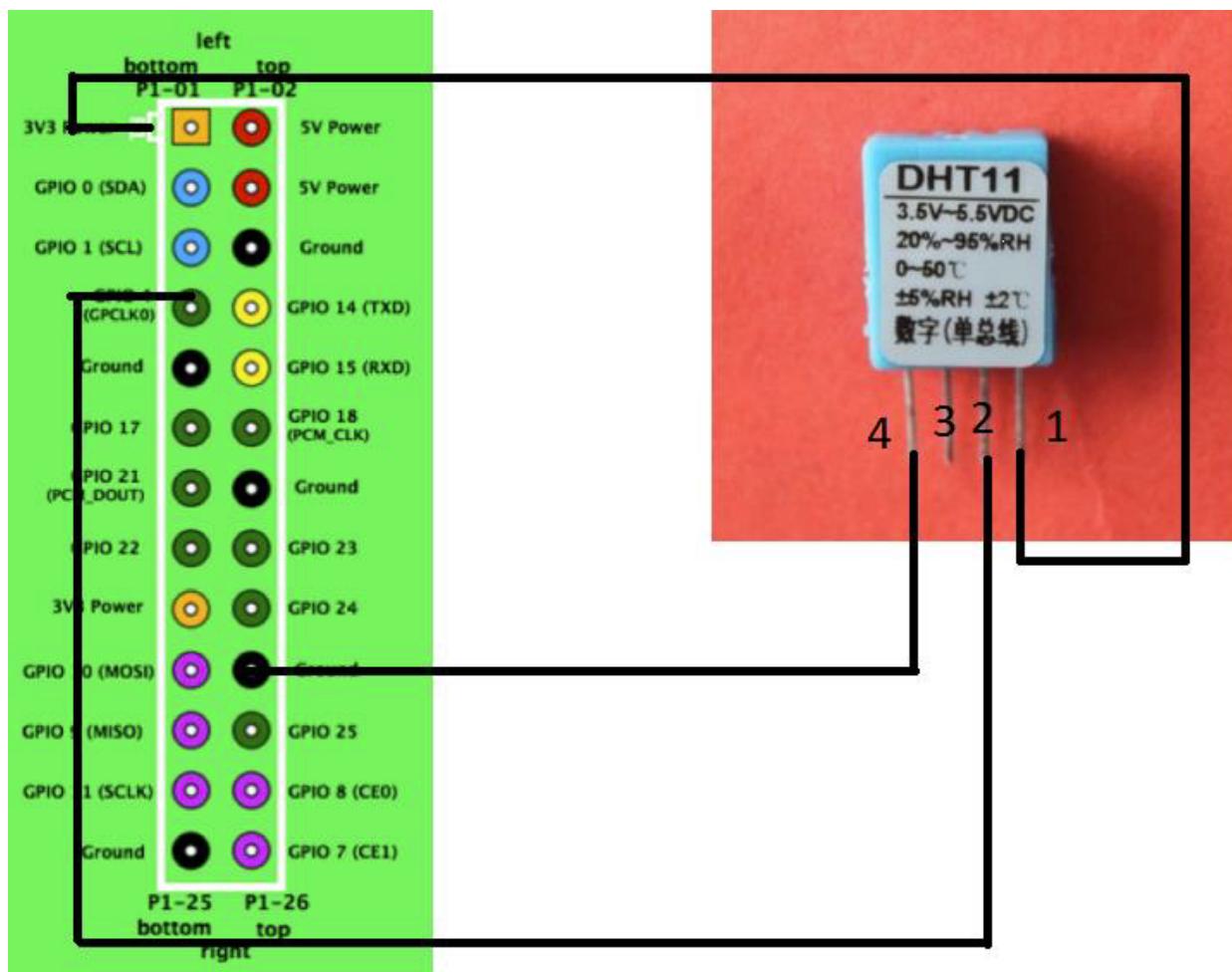
PC上的USB-TTL串口线配套的驱动程序；
PC上的串口终端软件，如minicom、picocom、putty等；
PC上的SSH软件，如putty等。
交叉编译软件

实验步骤

1. 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

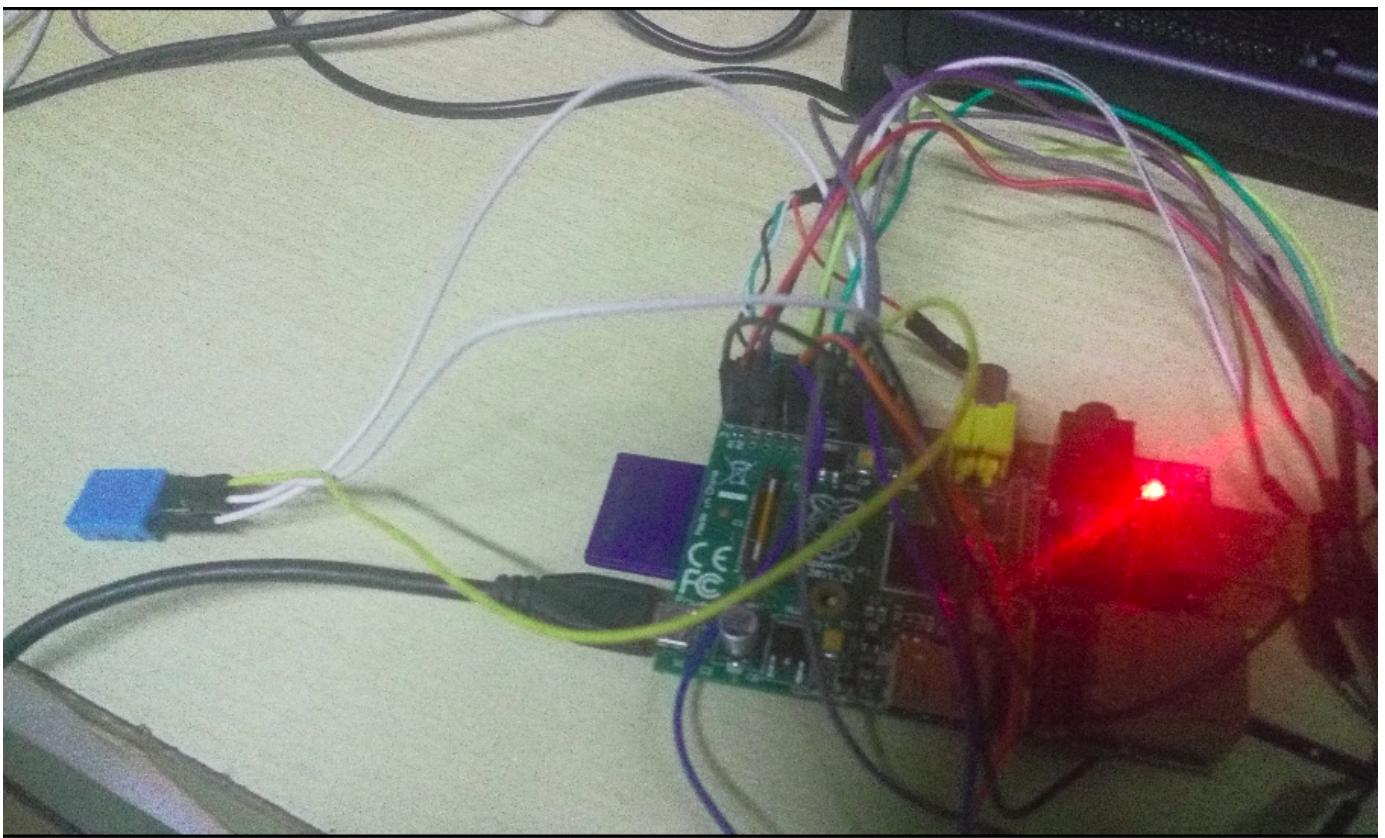
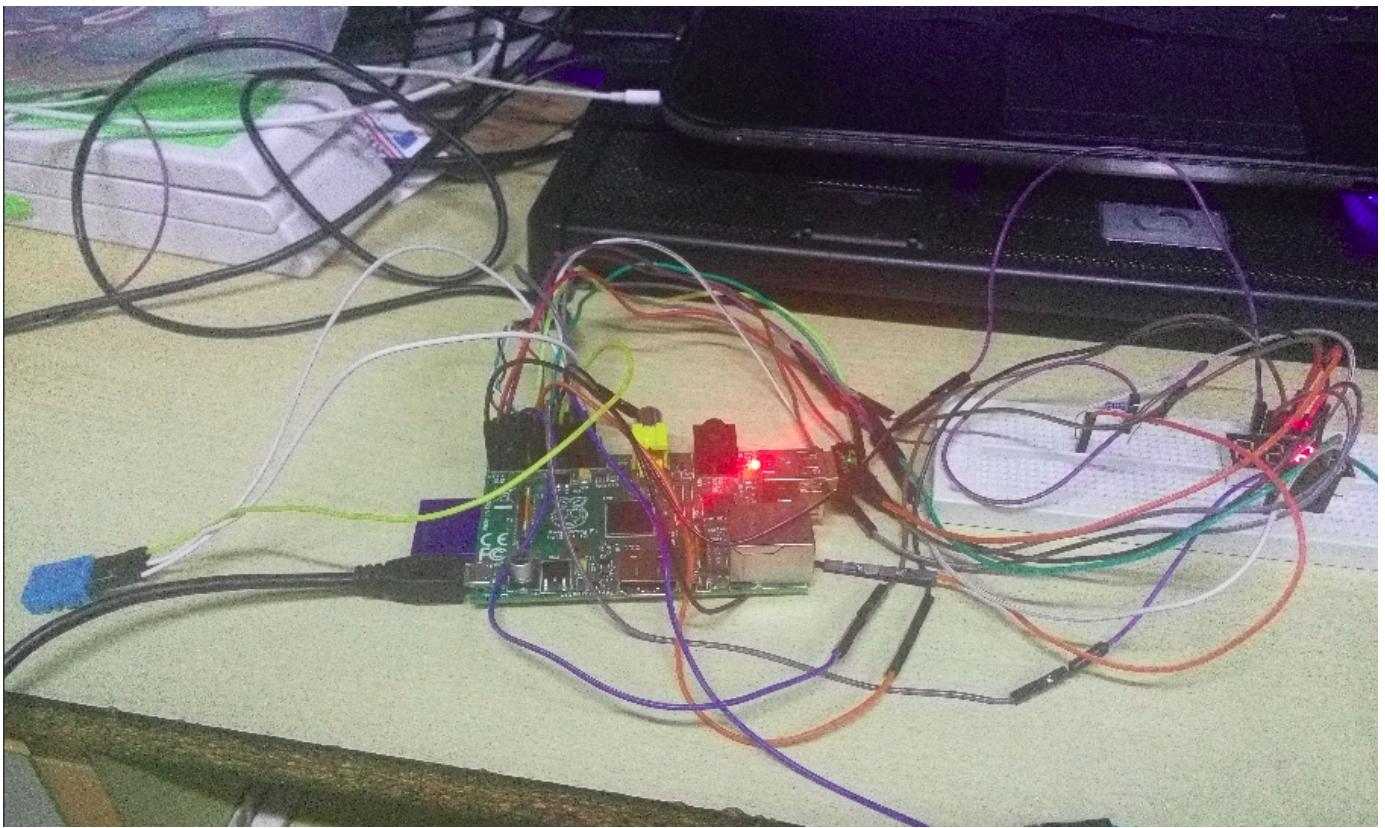
这次实验的连线可以分为两部分，一部分是dht11相关的连线，一部分是七段数码管相关的连线

dht11相关的连线参考下图所示(具体引脚有重新定义)



七段数码管的连线与之前lab6的相同

实物连接图如下所示



2.uCOS系统的移植

首先从网站上拿到原始的代码之后,修改makefile

1. 修改交叉编译工具环境
2. 修改include文件路径
3. 修改文件名问题
4. 修改汇编代码语法错误问题

最终的makefile如下所示

```
ARMGNU ?= /home/sodabeta/toolchain/raspbian-toolchain-gcc-4.7.2-
linux64/bin/arm-linux-gnueabihf

INCLUDEPATH ?= "./h"

COPS = -Wall -O2 -nostdlib -nostartfiles -ffreestanding -
mcpu=arm1176jzf-s -I $(INCLUDEPATH) -mfloating-abi=softfp
# COPS = -Wall -O2 -nostartfiles -ffreestanding -mcpu=arm1176jzf-s -I
$(INCLUDEPATH) -mfloating-abi=softfp
gcc : kernel.img

OBJS = build/startup.o

OBJS += build/uart.o
OBJS += build/timer.o
OBJS += build/interrupts.o

OBJS += build/OS_Cpu_a.o
OBJS += build/OS_Cpu_c.o

OBJS += build/uicos_ii.o

OBJS += build/main.o
OBJS += build/userApp.o

OBJS += lib/libc.a
OBJS += lib/libgcc.a
# OBJS += lib/libwiringPi.a

clean :
    rm -f build/*.o
    rm -f *.bin
    rm -f *.hex
    rm -f *.elf
    rm -f *.list
    rm -f *.img
    rm -f build/*.bc

build/%.o : port/%.s
    gcc $(COPS) -D__ASSEMBLY__ -c -o $@ $<

build/%.o : init/%.s
    gcc $(COPS) -D__ASSEMBLY__ -c -o $@ $<

build/%.o : port/%.c
```

```

gcc $(COPS) -c -o $@ $<

build/%.o : bsp/%.c
gcc $(COPS) -c -o $@ $<

build/%.o : usrApp/%.c
gcc $(COPS) -c -o $@ $<

build/ucos_iio.o : ucos/ucos_iio.c
gcc $(COPS) -c ucos/ucos_iio.c -o build/ucos_iio.o

kernel.img : raspberrypi.ld $(OBJS)
ld $(OBJS) -T raspberrypi.ld -o ucos_bcm2835.elf
# gcc $(OBJS) -T raspberrypi.ld -o ucos_bcm2835.elf
objdump -D ucos_bcm2835.elf > ucos_bcm2835.list
objcopy ucos_bcm2835.elf -O ihex ucos_bcm2835.hex
objcopy ucos_bcm2835.elf -O binary ucos_bcm2835.bin
objcopy ucos_bcm2835.elf -O binary kernel.img

```

make之后得到 kernel.img ,替换pi /boot/下的文件重新启动,就可以看到我们的两个任务userapp1 和userapp2在运行了

3.测试、实现uC/OS II对GPIO的访问

下面就是测试直接通过寄存器来实现uC/OS II对GPIO的访问,这里以测试gpio口编号为16的led灯的闪烁为例进行测试,修改 userapp.c 代码如下所示

```

#include "uart.h"
#include "ucos/includes.h"
extern void PUT32 (unsigned int, unsigned int);
extern unsigned int GET32 (unsigned int);
#define GPSET0 0x2020001C
#define GPCLR0 0x20200028
void userApp2(void * args)
{
    while(1)
    {
        PUT32(GPSET0,1<<16);
        uart_string("in userApp2");
        OSTimeDly(300);
    }
}
void userApp1(void * args)
{
    while(1)
    {
        PUT32(GPCLR0,1<<16);

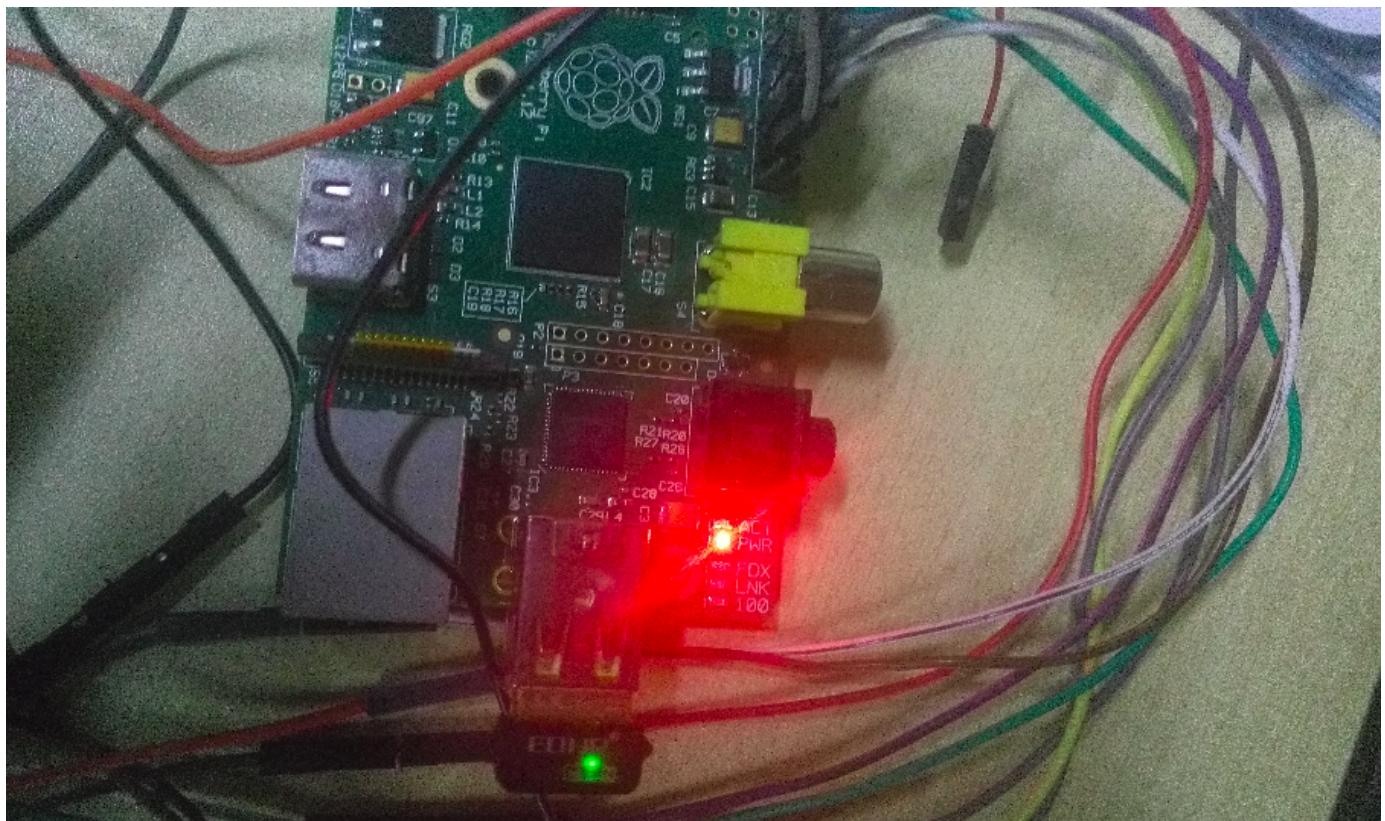
```

```

        uart_string("in userApp1");
        OSTimeDly(300);
    }
}

```

重新make之后启动,就可以看到led灯开始闪烁了



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

首先重新修改ucos的makefile,然后把 wiringPi 库静态编译成 libwiringPi.a ,然后用两个任务分别负责获取当前的温度和湿度,另一个任务负责七段数码管的显示

获取温度和湿度的主要代码如下所示

```

uint8 crc;
uint8 i;
pinMode(pinNumber,OUTPUT);
digitalWrite(pinNumber, 0);
delay(25);
digitalWrite(pinNumber, 1);
pinMode(pinNumber, INPUT);
pullUpDnControl(pinNumber,PUD_UP);
delayMicroseconds(27);
if(digitalRead(pinNumber)==0)
{
    while(!digitalRead(pinNumber));
    for(i=0;i<32;i++)

```

```

{
  while(digitalRead(pinNumber));
  while(!digitalRead(pinNumber));
  delayMicroseconds(HIGH_TIME);
  databuf*=2;
  if(digitalRead(pinNumber)==1)
  {
    databuf++;
  }
}
for(i=0;i<8;i++)
{
  while(digitalRead(pinNumber));
  while(!digitalRead(pinNumber));
  delayMicroseconds(HIGH_TIME);
  crc*=2;
  if(digitalRead(pinNumber)==1)
  {
    crc++;
  }
}

```

七段数码管显示的主要代码如下所示

```

int digits[10][8]={
  {0,0,0,0,1,1,0,0},
  {0,0,1,1,1,1,1,1},
  {0,1,0,0,0,1,0,1},
  {0,0,1,0,0,1,0,1},
  {0,0,1,1,0,1,1,0},
  {1,0,1,0,0,1,0,0},
  {1,0,0,0,0,1,0,0},
  {0,0,1,1,1,1,0,1},
  {0,0,0,0,0,1,0,0},
  {0,0,1,0,0,1,0,0}
};

void dow(const char* str){
  int d[]={13,12,8,9};
  rep(i,4){
    rep(j,4)
      digitalWrite(d[j],i==j);
    rep(j,8)
      digitalWrite(j,digits[str[i]-'0'][j]);
    delay(5);
  }
}

```

实验结果视频: [vedio](#)

