

# 嵌入式系统导论实验报告

姓名	学号	班级	电话	邮箱
陈潇	15352048	15M1	13727022947	<a href="mailto:616131435@qq.com">616131435@qq.com</a>

## 1.实验题目

1.修改程序改变按键对应的驱动灯颜色并对比，做调试分析。

2.修改NOT gate项目并作分析。

## 2.实验结果

1.

程序代码部分：

```
void PortF_Init(void){ volatile uint32_t delay;
    SYSCTL_RCGCGPIO_R |= 0x00000020; // 1) activate clock for Port F
    delay = SYSCTL_RCGCGPIO_R;        // allow time for clock to start
    GPIO_PORTF_LOCK_R = 0x4C4F434B;   // 2) unlock GPIO Port F
    GPIO_PORTF_CR_R = 0x1F;           // allow changes to PF4-0
    // only PF0 needs to be unlocked, other bits can't be locked
    GPIO_PORTF_AMSEL_R = 0x00;        // 3) disable analog on PF
    GPIO_PORTF_PCTL_R = 0x00000000;    // 4) PCTL GPIO on PF4-0
    GPIO_PORTF_DIR_R = 0x0E;          // 5) PF4,PF0 in, PF3-1 out
    GPIO_PORTF_AFSEL_R = 0x00;        // 6) disable alt funct on PF7-0
    GPIO_PORTF_PUR_R = 0x11;          // enable pull-up on PF0 and PF4
    GPIO_PORTF_DEN_R = 0x1F;          // 7) enable digital I/O on PF4-0
}

int main(void){ uint32_t status;
    PortF_Init();                     // initialize PF0 and PF4 and make them inputs
                                     // make PF3-1 out (PF3-1 built-in LEDs)
    while(1){
        status = PortF_Input();
        switch(status){              // switches are negative logic on PF0 and PF4
            case 0x01: PortF_Output(YELLOW); break; // SW1 pressed
            case 0x10: PortF_Output(SKY_BLUE); break; // SW2 pressed
            case 0x00: PortF_Output(PINK); break; // both switches pressed
            case 0x11: PortF_Output(0); break; // neither switch pressed
        }
    }
}
```

可以在注释代码中看到驱动灯所对应的颜色是以16进制存储。在main函数中首先进行端口F的初始化（代码第一段），关键部分是GPIO\_PORTF\_DIR\_R（=0x0E=01110）即设定1-3为输出灯，0和4位两个管脚是输入信号。GPIO\_PORTF\_PUR\_R（=0x11=10001）即设定两个输出管脚上拉电阻。在while反复循环中反复读取PF4和PF0的状态，根据这二者的电平状态决定GPIO\_PORTF\_DATA\_R的输出状态，因为只有第1-3位是输出位，因此该输出只会对1-3位产生影响。



以上为实际操作结果。

2.

首先，对于没有修改代码之前执行仿真操作得到的波形：



可以看到在PORTD的第三个管脚有波形，高电平的持续时间约为低电平的2倍。

修改代码：

```
LDR R1, =GPIO_PORTD_DIR_R      ; R1 = &GPIO_PORTD_DIR_R
LDR R0, [R1]                   ; R0 = [R1]
ORR R0, R0, #0x04               ; R0 = R0|0x04 (make PD2 output)
BIC R0, R0, #0x01               ; R0 = R0 & NOT(0x01) (make PD0 input)
STR R0, [R1]                   ; [R1] = R0
; 4) regular port function
LDR R1, =GPIO_PORTD_AFSEL_R    ; R1 = &GPIO_PORTD_AFSEL_R
LDR R0, [R1]                   ; R0 = [R1]
BIC R0, R0, #0x05               ; R0 = R0&~0x05 (disable alt funct on PD2,PD0)
STR R0, [R1]                   ; [R1] = R0
; 5) enable digital port
LDR R1, =GPIO_PORTD_DEN_R      ; R1 = &GPIO_PORTD_DEN_R
LDR R0, [R1]                   ; R0 = [R1]
ORR R0, R0, #0x05               ; R0 = R0|0x05 (enable digital I/O on PD2,PD0)
STR R0, [R1]                   ; [R1] = R0
```

总而言之就是将所有原本指向PORTD3的操作全部改到PD2，对应的都是位的修改，没有代码的添加或删除操作。

```
loop
    LDR R1, [R0]
    AND R1, #0x01                ; Isolate PD0
    EOR R1, #0x01                ; NOT state of PD0 read into R1
    STR R1, [R0]
    nop
    nop
    LSL R1, #2                   ; SHIFT left negated state of PD0 read into R1
```

在loop中每次的数据显示标签改到R1左移2位，即PD2。

得到结果：



要使得占空比产生变化，即在loop循环中多添加nop语句即可。



修改代码如下

```
loop
    LDR R1,[R0]
    AND R1,#0x01      ; Isolate PD0
    EOR R1,#0x01      ; NOT state of PD0 read into R1
    STR R1,[R0]
    nop
    nop
    nop
    nop
    nop
    nop
    LSL R1,#2          ; SHIFT left negated state of PD0 read into R1
    STR R1,[R0]        ; Write to PortD DATA register to update LED on PD3
    B loop             ; unconditional branch to 'loop'
```