

# 上海电力大学

## DSP 原理与应用实验报告



实验名称: 数码管显示控制

专业班级: 集成电路设计与集成系统

姓 名: \_\_\_\_\_

学 号: \_\_\_\_\_

2024 年 1 月 6 日

## 一、实验内容

通过 DSP 的通用输入输出多路复合器 GPIO 来控制开发板上数码管的显示。

## 二、实验内容

(注: 实验板上的数码管是共阳数码管)

### 1. GPIO 的寄存器

对于 DSP 输入/输出引脚的操作, 都是通过对寄存器的设置来实现的。例如, 选择某个引脚是作外设功能引脚还是作通用数字 I/O 口; 当引脚作为通用数字 I/O 口时, 是作输入还是作输出; 如何使其输出高电平或者低电平; 如何使其引脚电平翻转; 如何知道引脚上的电平是高或者是低, 这些都是通过对 GPIO 寄存器的操作来实现的。

GPIO 的寄存器分成了两大类: 一类是控制寄存器, 主要由功能选择控制寄存器 GPxMUX, 方向控制寄存器 GPxDIR, 输入限定控制寄存器 GPxQUAL 组成, 其中 x 代表 A、B、D、E、F 或者是 G; 另一类是数据寄存器, 主要由数据寄存器 GPxDAT、置位寄存器 GPxSET、清除寄存器 GPxCLEAR 和取反寄存器 GPxTOGGLE 组成, 如图一所示。

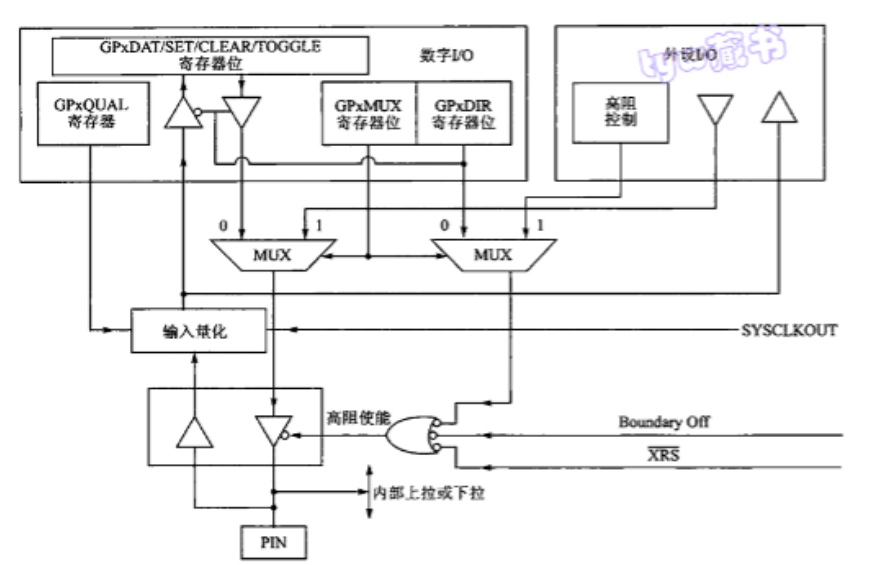


图 1: GPIO 多路功能复用的原理

### 2. 实验板的 LED 连接图

电路原理图如图 2 所示, 七段数码管由 A-F 控制数字显示, DP 为小数点显示, 连接至 2812 的 PB0—PB6, DIG1—DIG6 控制数码管的开关, DIG1—DIG6 由三极管开关电路实现高低电平转换, 2812 的 PB8—PB13 负责控制开关电路。

|      |    |                  |
|------|----|------------------|
| DPX  | 45 | PWM7/IOB0        |
| CX   | 46 | PWM8/IOB1        |
| DX   | 47 | PWM9/IOB2        |
| EX   | 48 | PWM10/IOB3       |
| GX   | 49 | PWM11/IOB4       |
| BX   | 50 | PWM12/IOB5       |
| AX   | 53 | T3PWM_T3CMP/IOB6 |
| FX   | 55 | T4PWM_T4CMP/IOB7 |
| DY1  | 57 | CAP4_QEP3/IOB8   |
| DY2  | 59 | CAP5_QEP5/IOB9   |
| DY3  | 60 | CAP6_QEPI2/IOB10 |
| DY4  | 71 | TDIRB/IOB11      |
| DY5  | 72 | TCKINB/IOB12     |
| DY6  | 61 | C4TRIP/IOB13     |
| SEL5 | 62 | C5TRIP/IOB14     |
| SEL6 | 63 | C6TRIP/IOB15     |

图 2: 2812 数码管连接图

### 三、实验步骤

#### 1. 编写代码

下面是需要在给出代码上进行修改的部分。

Macro\_definition.h

```

1  #ifndef MACRO_DEFINITION_H_
2  #define MACRO_DEFINITION_H_
3  #include "DSP281x_Device.h"
4
5  #define LED1_ON  GpioDataRegs.GPFCLEAR.bit.GPIOF11=1
6  #define LED1_OFF GpioDataRegs.GPFSET.bit.GPIOF11=1
7  #define LED2_ON  GpioDataRegs.GPFCLEAR.bit.GPIOF8=1
8  #define LED2_OFF GpioDataRegs.GPFSET.bit.GPIOF8=1
9  #define LED3_ON  GpioDataRegs.GPFCLEAR.bit.GPIOF10=1
10 #define LED3_OFF GpioDataRegs.GPFSET.bit.GPIOF10=1
11 #define LED4_ON  GpioDataRegs.GPFCLEAR.bit.GPIOF9=1
12 #define LED4_OFF GpioDataRegs.GPFSET.bit.GPIOF9=1
13 #define LEDSD_1_ON GpioMuxRegs.GPBMUX.all = 0x0000
14
15 #define DY1_ON  GpioDataRegs.GPBCLEAR.bit.GPIOB8=1
16 #define DY2_ON  GpioDataRegs.GPBCLEAR.bit.GPIOB9=1
17 #define DY3_ON  GpioDataRegs.GPBCLEAR.bit.GPIOB10=1
18 #define DY4_ON  GpioDataRegs.GPBCLEAR.bit.GPIOB11=1

```

```
19 #define DY5_ON  GpioDataRegs.GPBCLEAR.bit.GPIOB12=1
20 #define DY6_ON  GpioDataRegs.GPBCLEAR.bit.GPIOB13=1
21
22 #define DY1_OFF GpioDataRegs.GPBSET.bit.GPIOB8=1
23 #define DY2_OFF GpioDataRegs.GPBSET.bit.GPIOB9=1
24 #define DY3_OFF GpioDataRegs.GPBSET.bit.GPIOB10=1
25 #define DY4_OFF GpioDataRegs.GPBSET.bit.GPIOB11=1
26 #define DY5_OFF GpioDataRegs.GPBSET.bit.GPIOB12=1
27 #define DY6_OFF GpioDataRegs.GPBSET.bit.GPIOB13=1
28
29 #define DPX_ON GpioDataRegs.GPBCLEAR.bit.GPIOB0=1 // 小数点亮暗
30 #define CX_ON  GpioDataRegs.GPBCLEAR.bit.GPIOB1=1
31 #define DX_ON  GpioDataRegs.GPBCLEAR.bit.GPIOB2=1
32 #define EX_ON  GpioDataRegs.GPBCLEAR.bit.GPIOB3=1
33 #define GX_ON  GpioDataRegs.GPBCLEAR.bit.GPIOB4=1
34 #define BX_ON  GpioDataRegs.GPBCLEAR.bit.GPIOB5=1
35 #define AX_ON  GpioDataRegs.GPBCLEAR.bit.GPIOB6=1
36 #define FX_ON  GpioDataRegs.GPBCLEAR.bit.GPIOB7=1
37
38 #define DPX_OFF GpioDataRegs.GPBSET.bit.GPIOB0=1
39 #define CX_OFF  GpioDataRegs.GPBSET.bit.GPIOB1=1
40 #define DX_OFF  GpioDataRegs.GPBSET.bit.GPIOB2=1
41 #define EX_OFF  GpioDataRegs.GPBSET.bit.GPIOB3=1
42 #define GX_OFF  GpioDataRegs.GPBSET.bit.GPIOB4=1
43 #define BX_OFF  GpioDataRegs.GPBSET.bit.GPIOB5=1
44 #define AX_OFF  GpioDataRegs.GPBSET.bit.GPIOB6=1
45 #define FX_OFF  GpioDataRegs.GPBSET.bit.GPIOB7=1
46
47 #endif /* MACRO_DEFINITION_H_ */
```

num.h, 这是 7 段数码管显示的代码, 控制一下亮暗就行, 需要补充 0-9 数字的显示。

```
1 #ifndef SHOW_NUM_H_
2 #define SHOW_NUM_H_
3
4 #include "Macro_definition.h"
5
6 void show_num0(void)
7 {
8     AX_ON;
9     BX_ON;
10    CX_ON;
11    DX_ON;
12    EX_ON;
13    FX_ON;
14    GX_OFF;
15 }
16
17 void show_num1(void)
18 {
19     AX_OFF;
20     BX_ON;
21     CX_ON;
```

```
22     DX_OFF;
23     EX_OFF;
24     FX_OFF;
25     GX_OFF;
26 }
27
28 void show_num2(void)
29 {
30     AX_ON;
31     BX_ON;
32     CX_OFF;
33     DX_ON;
34     EX_ON;
35     FX_OFF;
36     GX_ON;
37 }
38
39 void show_num3(void)
40 {
41     AX_ON;
42     BX_ON;
43     CX_ON;
44     DX_ON;
45     EX_OFF;
46     FX_OFF;
47     GX_ON;
48 }
49
50 void show_num4(void)
51 {
52     AX_OFF;
53     BX_ON;
54     CX_ON;
55     DX_OFF;
56     EX_OFF;
57     FX_ON;
58     GX_ON;
59 }
60
61 void show_num5(void)
62 {
63     AX_ON;
64     BX_OFF;
65     CX_ON;
66     DX_ON;
67     EX_OFF;
68     FX_ON;
69     GX_ON;
70 }
71
72 void show_num6(void)
73 {
74     AX_ON;
75     BX_OFF;
```

```
76     CX_ON;
77     DX_ON;
78     EX_ON;
79     FX_ON;
80     GX_ON;
81 }
82
83 void show_num7(void)
84 {
85     AX_ON;
86     BX_ON;
87     CX_ON;
88     DX_OFF;
89     EX_OFF;
90     FX_OFF;
91     GX_OFF;
92 }
93
94 void show_num8(void)
95 {
96     AX_ON;
97     BX_ON;
98     CX_ON;
99     DX_ON;
100    EX_ON;
101    FX_ON;
102    GX_ON;
103 }
104
105 void show_num9(void)
106 {
107     AX_ON;
108     BX_ON;
109     CX_ON;
110     DX_ON;
111     EX_OFF;
112     FX_ON;
113     GX_ON;
114 }
115
116
117 void show_num_init(void)
118 {
119     DPX_OFF;
120     CX_OFF;
121     DX_OFF;
122     EX_OFF;
123     GX_OFF;
124     BX_OFF;
125     AX_OFF;
126     FX_OFF;
127 }
128
129 void order_open(void)
```

```
130 {
131     DY1_ON;
132     DY1_OFF;
133     DY2_ON;
134     DY2_OFF;
135     DY3_ON;
136     DY3_OFF;
137     DY4_ON;
138     DY4_OFF;
139     DY5_ON;
140     DY5_OFF;
141     DY6_ON;
142     DY6_OFF;
143 }
144
145
146 #endif /* SHOW_NUM_H_ */
```

2. 编译调试

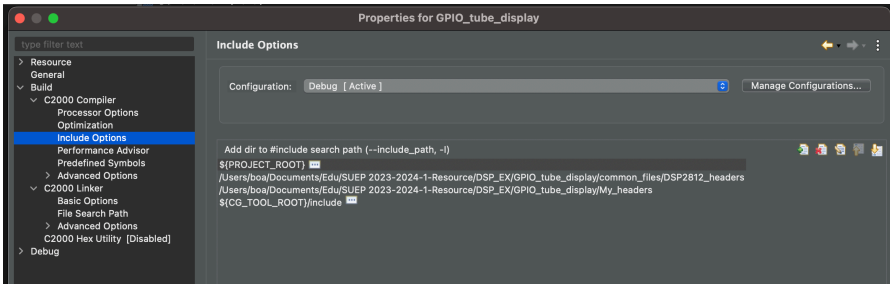


图 3: edit include option

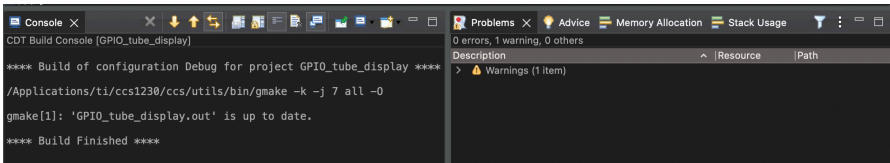


图 4: build

3. 下载运行、实验现象

首先，需要添加头文件。

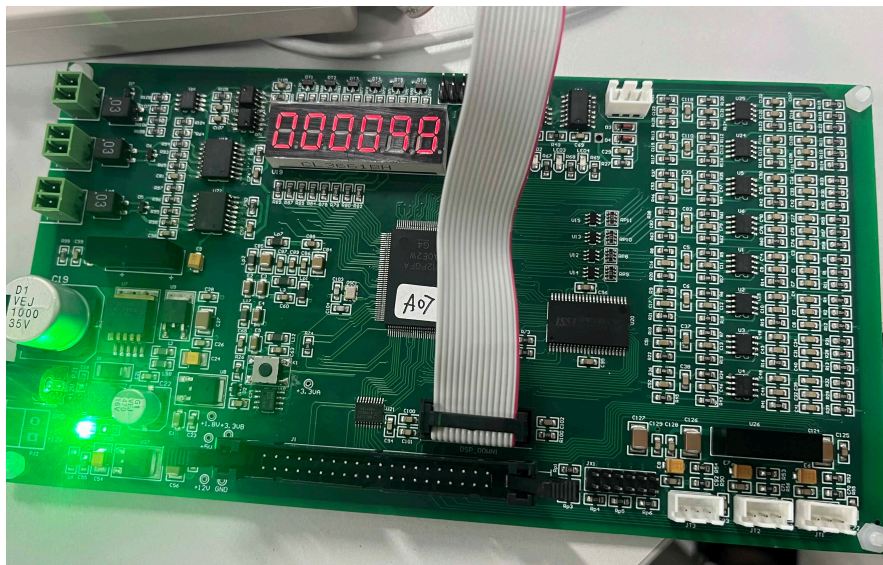


图 5: 运行现象

#### 四、实验小结