**交通灯控制系统**



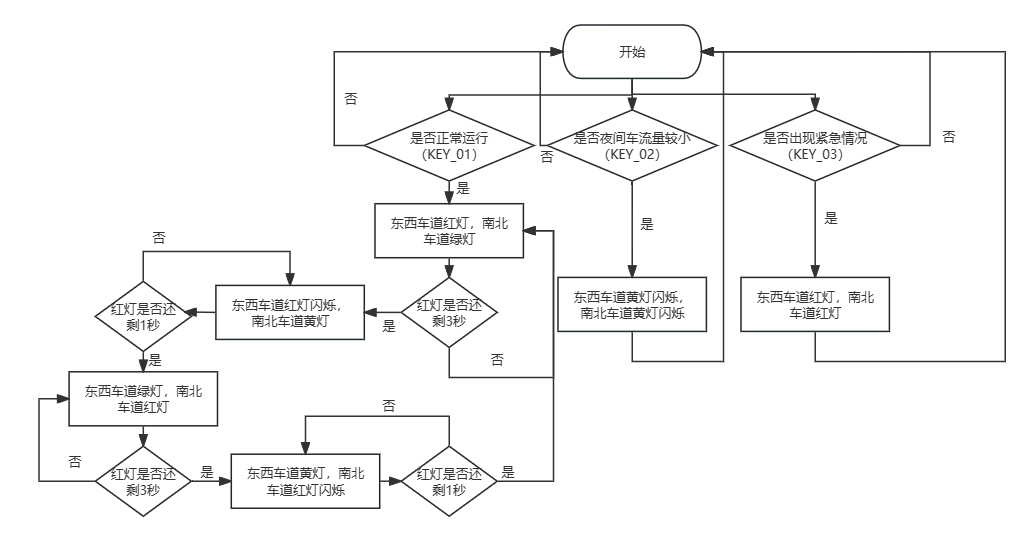
1. 实验目的
2. 通过已经学习的verilog编程经验，利用T20F256 FPGA开发板现有外设及功能进行课程设计。
3. 掌握FPGA开发环境，熟练使用Verilog描述语言编写数字电路设计，包括逻辑设计输入、引脚分配及锁定、调试、下载flash等。
4. 培养学生逻辑思维和问题解决能力，通过自行思考达到最终实验效果，并提出问题、解决问题。
5. 实验仪器

笔记本电脑、efinity开发环境、T20F256 FPGA开发板、zading-2.7驱动、开发板电源（金陵声宝电源）、ELITES-232DL下载线，Verilog编程语言等

1. 实验内容

交通灯控制系统功能如下：

1. 正常运行状态：东西车道红灯、南北车道绿灯→红灯还剩3秒时，东西车道红灯闪烁、南北车道黄灯→东西车道绿灯、南北车道红灯，如此反复；
2. 夜间车流量较小状态：东西车道黄灯闪烁、南北车道黄灯闪烁，提示减速慢行；
3. 出现紧急情况状态：东西车道红灯、南北车道红灯；
4. 程序设计流程



1. 定义输入输出变量、数码管、led灯、rgb三色灯及按键信号

module trafficlight(

input clk,

input rst\_n,

input emg,

input night,

output reg[2:0] rgb\_led,

output reg[7:0] led,

output shift\_clk,

output latch\_clk,

output seg\_data

);

//

wire [7:0] data1;

wire [7:0] data2;

reg [7:0] odata1;

reg [7:0] odata2;

wire [7:0] data3;

wire [7:0] data4;

reg [7:0] odata3;

reg [7:0] odata4;

reg [31:0] cnt;

reg [7:0] light\_dongxi;

reg [7:0] light\_nanbei;

reg [3:0] state;

always@(posedge clk)

1. 出现紧急情况状态

begin

if (!emg) //emergency

begin

light\_dongxi <= 8'd99;

light\_nanbei<=8'd99;

odata1 <= data1;

odata2 <= data2;

odata3 <= data3;

odata4 <= data4;

state<=3'd4;

End

1. 夜间车流量较小状态

else if(!night)//night

begin

light\_dongxi <= 8'd88;

light\_nanbei<=8'd88;

odata1 <= data1;

odata2 <= data2;

odata3 <= data3;

odata4 <= data4;

state<=3'd5;

end

else

begin

if(!rst\_n)

begin

state <= 3'd0;

light\_dongxi <= 8'd23;

light\_nanbei<=8'd20;

//led<=8'b1111\_1110;

End

1. 正常运行状态

else if(cnt==32'd50000000)

begin

cnt <= 1'b0;

case (state)

4'd0://dongxi red nanbei green

begin

if(light\_nanbei==8'h1)

begin

state <= 4'd1;

light\_dongxi<=8'd20;

light\_dongxi<=light\_dongxi-1'b1;

light\_nanbei<=8'd3;

end

else

begin

light\_dongxi <= light\_dongxi - 1'b1;

light\_nanbei<=light\_nanbei-1'b1;

rgb\_led<=3'b100;

led<=8'b0111\_1111;

end

end

4'd1://dongxi red nanbei yellow

begin

if(light\_dongxi==8'h3)

begin

state <= 4'd7;

light\_dongxi <= light\_dongxi - 1'b1;

light\_nanbei<=light\_nanbei-1'b1;

led<=8'b1110\_1111;

end

else if(light\_dongxi==8'h1)

begin

state<=4'd2;

light\_dongxi <= 8'd20;

light\_nanbei<=8'd23;

rgb\_led<=3'b100;

led<=8'b1110\_1111;

end

else

begin

light\_dongxi <= light\_dongxi - 1'b1;

light\_nanbei<=light\_nanbei-1'b1;

rgb\_led<=3'b100;

led<=8'b1110\_1111;

end

end

4'd2://dongxi green nanbei red

begin

if(light\_dongxi==8'h1)

begin

state <= 4'd3;

light\_dongxi <= 8'd3;

light\_nanbei<=light\_nanbei-1'b1;

end

else

begin

light\_dongxi <= light\_dongxi - 1'b1;

light\_nanbei<=light\_nanbei-1'b1;

rgb\_led<=3'b010;

led<=8'b1111\_1110;

end

end

4'd3://dongxi yellow nanbei red

begin

if(light\_nanbei==8'h3)

begin

state<=4'd8;

light\_dongxi <= light\_dongxi - 1'b1;

light\_nanbei<=light\_nanbei-1'b1;

rgb\_led<=3'b110;

end

else if(light\_nanbei==8'h1)

begin

state <= 4'd0;

light\_dongxi <= 8'd23;

light\_nanbei<=8'd20;

rgb\_led<=3'b110;

led<=8'b1111\_1110;

end

else

begin

light\_dongxi <= light\_dongxi - 1'b1;

light\_nanbei<=light\_nanbei-1'b1;

rgb\_led<=3'b110;

led<=8'b1111\_1110;

end

end

4'd4://dongxi red nanbei red

begin

if(light\_dongxi==8'h1)

begin

state<=4'd0;

light\_dongxi <= 8'd23;

light\_nanbei<=8'd20;

end

else

begin

light\_dongxi <= light\_dongxi - 1'b1;

light\_nanbei<=light\_nanbei-1'b1;

rgb\_led<=3'b100;

led<=8'b1111\_1110;

end

end

4'd5://dongxi yellow nanbei yellow

begin

rgb\_led<=3'b110;

led<=8'b1110\_1111;

state<=4'd6;

end

4'd6://dongxi none nanbei none

begin

rgb\_led<=3'b000;

led<=8'b1111\_1111;

state<=4'd5;

end

4'd7://dongxi none nanbei yellow

begin

light\_dongxi<=8'd1;

light\_nanbei=8'd1;

state<=4'd1;

rgb\_led<=3'b000;

led<=8'b1110\_1111;

end

4'd8://dongxi yellow nanbei none

begin

state<=4'd3;

light\_nanbei=8'd1;

light\_dongxi<=8'd1;

rgb\_led<=3'b110;

led<=8'b1111\_1111;

end

endcase

odata1<=data1;

odata2<=data2;

odata3 <= data3;

odata4 <= data4;

end

else

cnt <= cnt + 1'b1;

end

End

1. 南北车道状态

binary\_to\_bcd u\_binary\_to\_bcd\_1(

.binary\_input (light\_nanbei),

.bcd\_low (data1),

.bcd\_high (data2)

);

1. 东西车道状态

binary\_to\_bcd u\_binary\_to\_bcd\_2(

.binary\_input (light\_dongxi),

.bcd\_low (data3),

.bcd\_high (data4)

);、

1. 数码管输出信号

hc595\_display\_input u\_hc595\_display ////Call hc595\_display\_input to display numbers

(

.clk (clk),

.rst\_n (rst\_n),

//To be displayed

.data1 (odata1),

.data2 (odata2),

.data3 (odata3),

.data4 (odata4),

.data5 (8'hff),

.data6 (8'heb),

.data7 (8'hff),

.data8 (8'hb7),

//control signals of hc595

.shcp (shift\_clk), //shift clk

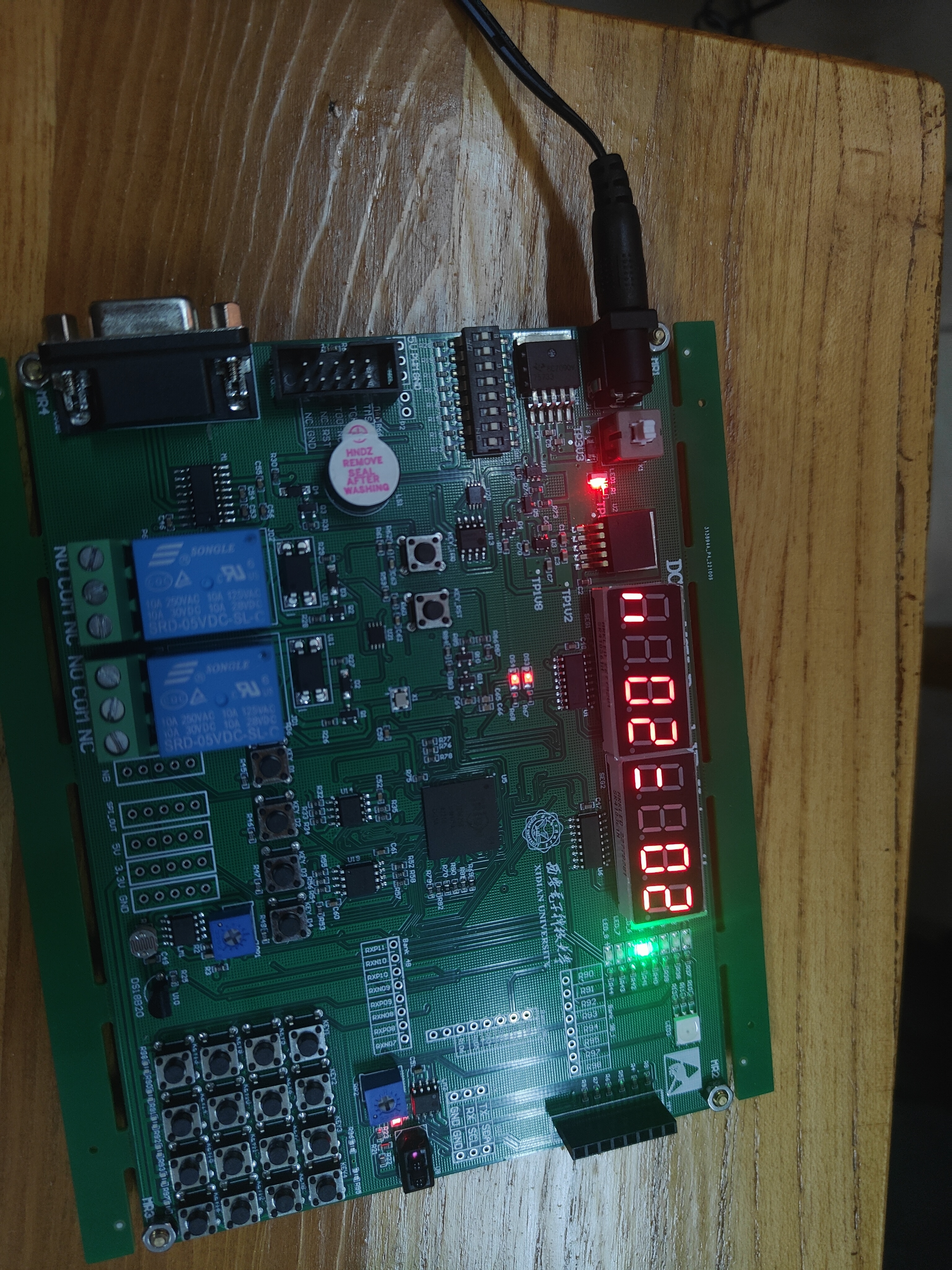
.stcp (latch\_clk), //latch clk

.seg\_data (seg\_data)

);

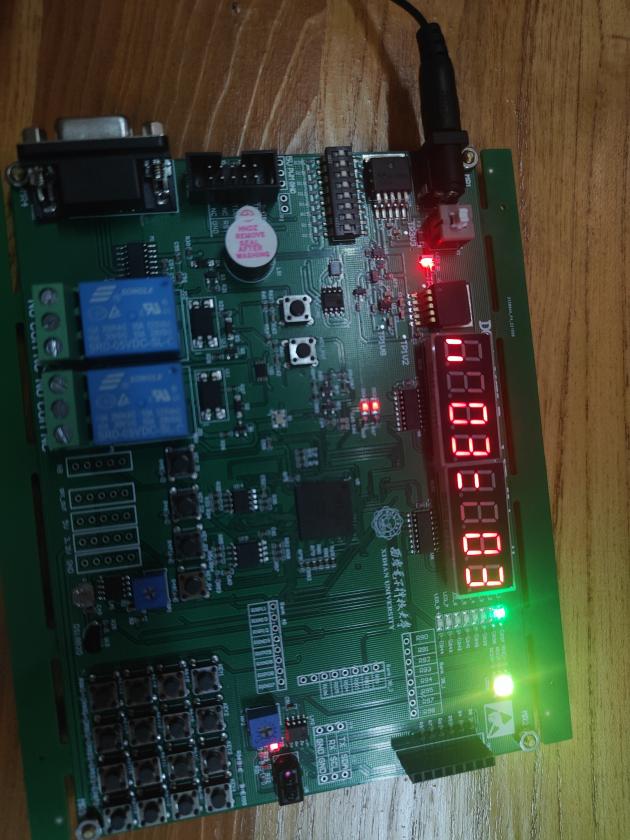
endmodule

1. 程序运行结果
2. 正常运行状态

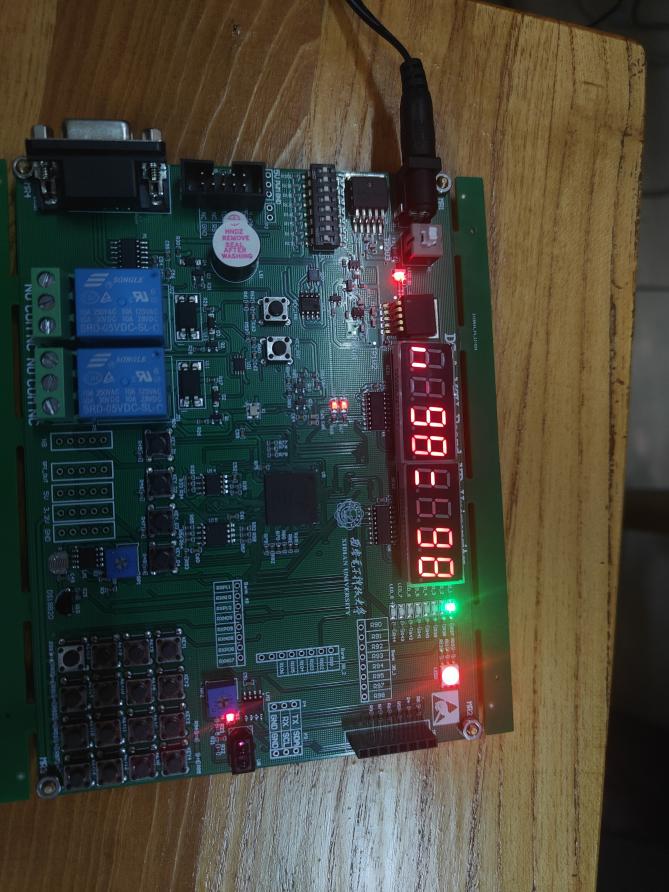




1. 夜间车流量较小状态



1. 出现紧急情况状态



1. 运行视频



1. 心得体会

通过制作基于FPGA的信号灯，我深入理解了FPGA的工作原理、硬件设计和编程方法，并提高了我的问题解决能力和实践技能。这个过程充满了挑战和机遇，每一次的成功都是一次宝贵的经验。通过选择合适的FPGA芯片和外设，并设计满足需求的硬件系统，我学会了硬件设计的基本原理。使用FPGA的硬件描述语言编写逻辑电路，并利用高级设计方法进行优化，我掌握了Verilog编程的基本技能。在仿真和测试过程中，我发现了可能存在的错误或问题，并修复了它们。最后，通过系统集成和文档编写，我提高了我的文档编写能力。通过这次实验，我不仅学会了如何制作基于FPGA的信号灯，还深入理解了数字电路设计的基本原理。我希望能够将所学的知识应用于实践，制作更复杂、功能更多的实验器件。同时，我也意识到不断学习和实践的重要性，随着技术的不断发展，我需要不断更新我的知识和技能。此外，在未来的实验中，我会考虑可扩展性和可维护性，以便更容易地添加新功能或进行修改。同时，我也会注意安全性，遵循所有的安全规范，确保项目的安全性和稳定性。总之，这次实验为我今后的学习和实践打下了坚实的基础。