

技术文件

完成时间：2011-12-18

# 工程实践与科技创新[3A] 设计报告

项目名称：**DC-DC** 开关电源模块及其控制系统

设计小组编号：**017**

设计小组名单：付旭初(组长)、林文景、吴儒俊

姓名	班级	学号	具体负责的工作	联系方式
付旭初	F0903011	5090309331	电路焊接、调试 报告撰写	15216712218
林文景	F0903011	5090309343	单片机程序设计	
吴儒俊	F0903010	5090309318	电路焊接、调试	

---

上海交通大学 电子信息与电气工程学院

地 址：东川路 800 号

邮 编：200240

### 摘 要:

本报告介绍了 DC-DC 开关电源的设计。此开关电源系统通过 AVR ATmega16 单片机控制 TL494 芯片, 实现电压 20~30V 输入到 5~10V 输出的调节。其中包含了四大模块: 降压型 DC-DC 开关电源, 单片机控制系统, 电压控制部分, 电压检测部分等。可实现单片机的开环、闭环控制。

本报告包含了此电源开关的工作原理、设计思路, 以及以上提到的四大模块的设计指标。同时附有笔者及同组人员在设计过程中遇到的问题以及心得体会。

### 关键词:

DC-DC 开关电源, AVR ATmega16 单片机, TL494 芯片, 电压控制, 电压检测, 开环控制, 闭环控制

### ABSTRACT

This report introduces the design of a DC-DC switching power supply. This switching power system uses an AVR ATmega16 microcontroller to control a TL494 chip which can realize a 20~30 voltage input to a 5~10 voltage output. This system contains four main parts, the Buck DC-DC switching power supply, the microcontroller system, the voltage control and the voltage detection, which can realize open-loop control and close-loop control.

This report contains the working principle, the design method and the design index of this switching power. In addition, the report contains some problems that we met during the experiment and our experience.

---

上海交通大学 电子信息与电气工程学院

地 址: 东川路 800 号

邮 编: 200240

## **KEYWORDS**

DC-DC switching power supply, AVR ATmega16 microcontroller, TL494 chip, voltage control, voltage detection, open-loop control, close-loop control

---

上海交通大学 电子信息与电气工程学院

地 址：东川路 800 号

邮 编：200240

# 目 录

<b>1. 概述.....</b>	<b>4</b>
1.1 编写说明.....	4
1.2 名词定义.....	4
1.3 硬件开发环境.....	4
1.4 软件开发环境.....	4
1.5 缩略语.....	5
<b>2. 系统总述.....</b>	<b>6</b>
2.1 系统组成.....	6
2.2 系统的主要功能.....	6
2.2.1 DC-DC 开关稳压电源模块.....	6
2.2.2 电压控制模块.....	7
2.2.3 电压检测模块.....	7
2.2.4 单片机小系统模块.....	8
<b>3. DC-DC 开关稳压电源模块的设计.....</b>	<b>9</b>
3.1 主要功能和设计指标.....	9
3.1.1 主要功能.....	9
3.1.2 设计指标.....	9
3.2 设计原理.....	9
3.3 电路主要部分设计及参数说明.....	10
3.3.1 TL494 介绍.....	10
3.3.2 参数设计.....	11
3.3.3 参数列表.....	12
3.4 专项讨论.....	13
<b>4. 电压控制模块的硬件设计.....</b>	<b>15</b>
4.1 主要功能和设计指标.....	15
4.1.1 主要功能.....	15
4.1.2 设计指标.....	15

4.2 设计原理.....	15
4.3 电路主要部分设计及参数说明.....	16
4.3.1 TL431 介绍及相关电路.....	16
4.3.2 CD4011B 介绍及相关电路.....	17
4.3.3 低通滤波部分.....	18
4.3.4 4n25 介绍及相关电路.....	19
4.3.5 参数列表.....	20
<b>5. 电压测量模块的硬件设计.....</b>	<b>22</b>
5.1 主要功能和设计指标.....	22
5.1.1 主要功能.....	22
5.1.2 设计指标.....	22
5.2 设计原理.....	22
5.3 电路主要部分设计及参数说明.....	23
5.3.1 电压反馈子模块.....	23
5.3.2 基准电压子模块.....	23
<b>6. 单片机小系统及软件设计.....</b>	<b>25</b>
<sup>6.1</sup> 单片机小系统 <sup>[3]</sup> .....	25
6.1.1 I/O 端口.....	25
6.1.2 16 位定时/计数器.....	25
6.1.3 A/D 转换.....	26
6.1.4 7 段数码管及按键.....	26
6.2 资源分配.....	26
6.3 软件设计.....	27
6.3.1 软件结构及功能.....	27
6.3.2 主函数.....	28
6.3.3 开环程序.....	28
6.3.4 闭环程序.....	28
<b>7. 致谢.....</b>	<b>30</b>

<b>8. 参考文献.....</b>	<b>31</b>
<b>9. 附录 A 系统操作说明书.....</b>	<b>32</b>
<b>10. 附录 B 测试和分析.....</b>	<b>34</b>
<sup>10.1</sup> 测试项目和方法 <sup>[1]</sup> .....	34
10.2 测试的资源.....	35
10.2.1 测试资源.....	35
10.2.2 测试环境.....	35
10.3 测试结果及分析.....	36
<b>11. 附录 C 课程学习心得和意见建议.....</b>	<b>38</b>
<b>12. 附录 D 软件程序清单.....</b>	<b>39</b>

# 1. 概述

## 1.1 编写说明

本报告为上海交通大学电子信息与电气工程学院工程实践与科技创新 3A 课程 DC-DC 开关电源系统的设计报告。本报告主要内容包括 DC-DC 开关电源模块、电压控制模块硬件、电压检测模块硬件、单片机小系统软件等部分的设计过程、调试过程以及使用说明。本报告的编写主要为了记录此开关电源的各项技术指标等以便使用者参考，并为今后系统的改进提供参考。本报告适于电力电子相关专业的人员阅读。

## 1.2 名词定义

开关电源：通过控制三级管开通和关断的时间比率，维持稳定输出电压的一种电源。

单片机系统：以 AVR ATMage16 为核心以及一些外围电路组成的控制电路系统。

开环控制：不将控制的结果反馈回来影响当前控制。

闭环控制：被控的输出以一定方式返回到作为控制的输入端，并对输入端施加控制影响。

纹波：叠加在直流稳定量上的交流分量。

占空比：在一串脉冲周期序列中，正脉冲的持续时间与脉冲总周期的比值。

A/D 转换：即模数转换，就是把模拟信号转换成数字信号。

## 1.3 硬件开发环境

表 1.1 硬件开发环境

硬件开发工具	运行环境
直流稳压电源	~220V
数字万用表	
示波器	~220V
函数发生器	~220V
电源引线，示波器探头，开路线	
电烙铁、焊锡、小刀等	

## 1.4 软件开发环境



表格 1.2 软件开发环境

软件开发工具	运行环境
AVR Studio4	WindowsXP
ISPLAY	WindowsXP
Microsoft Excal	WindowsXP
Origin 7.5	Windows 7

## 1.5 缩略语

DC (Direct Current ): 直流电

ADC (Analog-to-Digital Converter ): 模数变换器

PWM (Pulse Width Modulation ): 脉冲宽度调制

## 2. 系统总述

### 2.1 系统组成

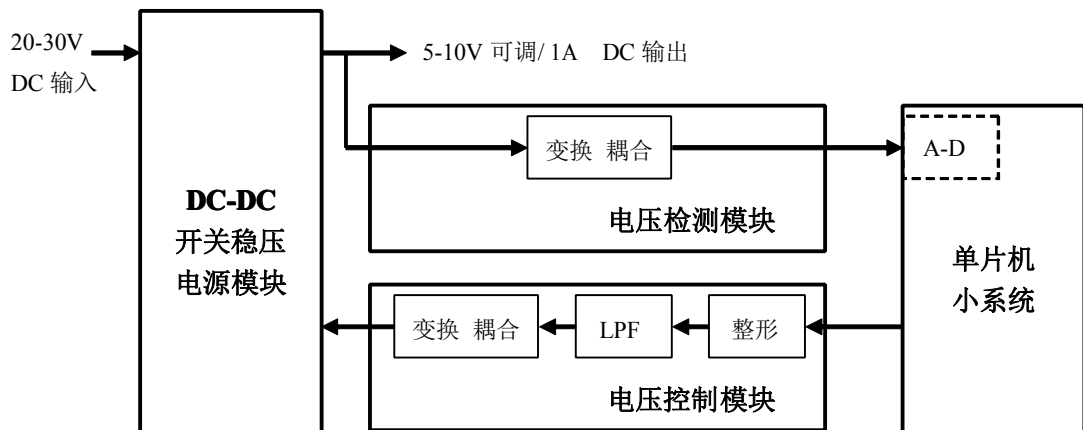


图 2.1 系统组成图<sup>[1]</sup>

上图为该 DC-DC 开关稳压电源的系统组成，其中包括了 DC-DC 开关稳压电源模块、单片机小系统模块、电压检测模块和电压控制模块。对于输入的 20~30V 直流电压，系统通过闭环控制检测输出信号，应用单片机小系统发出控制信号调节输出，使输出稳定在 5~10V 间某一期望的值。

### 2.2 系统的主要功能

此系统可以实现 20~30V DC 输入到 5~10V DC 输出的降压调节，输出电压的分辨率为 0.1，用户可根据需要选择开环或者闭环的工作方式。在单片机系统数码管处可检测当前输出电压值，精度为 0.1。开关稳压电源模块实现基本降压调压，电压检测模块通过对输出电压的检测，经过 AD 转换反馈给单片机系统，单片机系统通过改变控制信号的占空比，经过整形、滤波控制稳压电源的输出，达到稳压效果。

#### 2.2.1 DC-DC 开关稳压电源模块

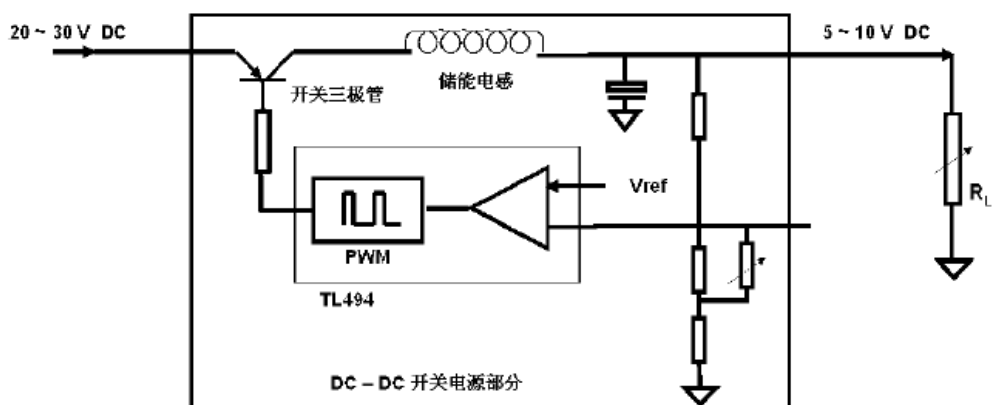


图 2.2.1 DC-DC 开关电源模块设计结构图<sup>[1]</sup>

上图为 DC-DC 开关电源模块设计结构图，系统输入信号为 20~30V 直流电压，以 TL494 为核心输出 PWM 波控制开关三极管的通断，使输出达到 5~10V 直流电压的降压调节。

### 2.2.2 电压控制模块

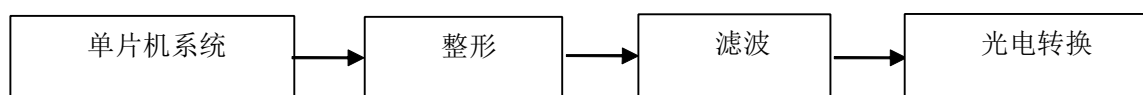


图 2.2.2 电压控制模块设计结构图

此模块中，单片机系统输出的方波信号不够稳定，经过 TL431 整形得到 4V 稳定的方波信号。方波信号经过低通滤波器，得到稳定的接近直流的电压，通过转换为光信号输入至光耦，实现对输出的控制。应用光耦主要是由于电压控制模块与开关电源模块所用的电压值不同，达到电器隔离效果。

### 2.2.3 电压检测模块

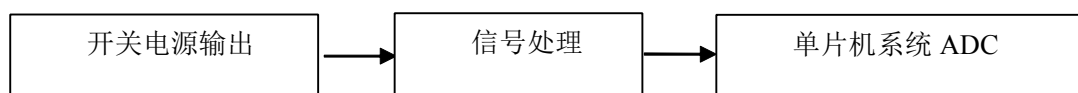


图 2.2.3 电压检测模块设计结构图

开关电源的输出电压经过分压输入单片机 ADC 端口，与基准电压的 ADC 比较得到差值，即为输出与期望的差值，作为闭环部分的反馈信号输入至单片机系统。

#### 2.2.4 单片机小系统模块

单片机小系统包含了电压开环、闭环的控制和面向用户的操作、显示界面。用户可根据按键选择开环控制或闭环控制方式，设置期望输出电压值，显示当前输出电压等操作，占空比的相对值和电压值会在数码管上显示，直观的反映给使用人员。

### 3. DC-DC 开关稳压电源模块的设计

#### 3.1 主要功能和设计指标

##### 3.1.1 主要功能

DC-DC 开关稳压电源模块可将输入的 20~30V 直流电压调整为 5~10V 可调的直流输出电压。用户可用变阻器实现调压，也可接入外围的其他三个模块的电路实现闭环或开环的调压。

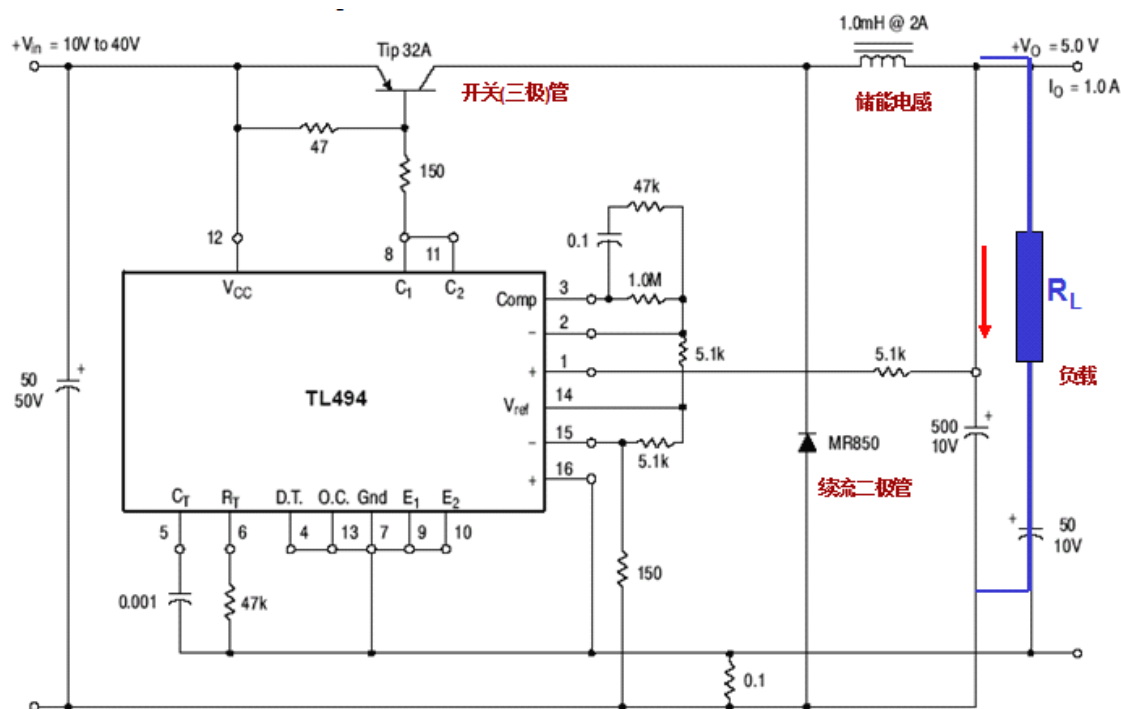
##### 3.1.2 设计指标

表 3.1 DC-DC 开关稳压电源设计指标<sup>[2]</sup>

输入电压	20~30V
输出电压	5~10V
额定输出电流	1A
限流值	1.1A
电压调整率	0.5%
电流调整率	1%
输出电压纹波	$\leq 100\text{mV}$ (p-p)
效率	$\geq 65\%$
截至电流值	1.2A

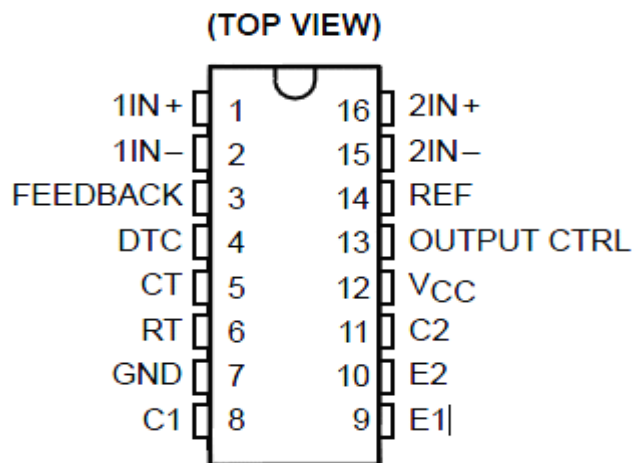
#### 3.2 设计原理

此模块通过 TL494 产生一定频率 PWM 波控制开关三极管的通断来保持一相对稳定的输入功率达到调压的效果，通过三极管后的储能电感不断地储存能量和释放能量达到稳压的效果。下图为设计电路。

图 3.1 DC-DC 开关稳压电源模块设计电路<sup>[1]</sup>

### 3.3 电路主要部分设计及参数说明

### 3.3.1 TL494 介绍

图 3.2 TL494 管脚图<sup>[3]</sup>

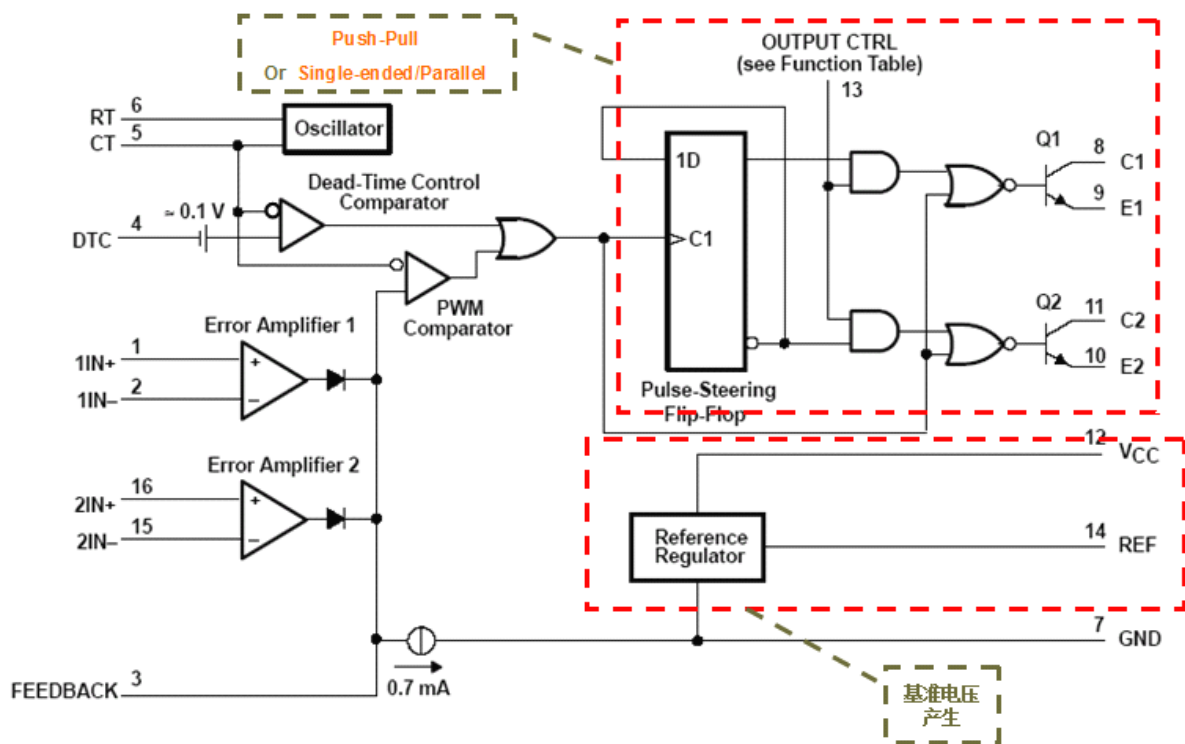


图 3.3 TL494 工作原理图<sup>[1]</sup>

TL494 管脚定义及内部电路结构如图。基准电压经分压后与输出电压采样接入误差放大器 1，经过差分放大电路后接入 PWM 比较器进行比较，可实现 PWM 波的占空比，从而控制开关三极管的通断时间比，达到调压效果。

### 3.3.2 参数设计

#### 3.3.2.1 开关频率

TL494 的 5 号和 6 号管脚可设置开关频率，取  $C_3=1000\text{pF}$ ， $R_7=5.1\text{k}\Omega$ 。则开关频率为  $f=1/(C_3 \times R_7)=196\text{kHz}$ 。虽然超过了老师推荐的频率不大于  $150\text{kHz}$ ，但在后续试验中电源电流并不大，效率也在 65% 以上。稍微提高开关频率有利于纹波的减小。

#### 3.3.2.2 开关三极管电路

通过调节电阻  $R_1$  及  $R_2$ ，可改变开关三极管的导通深度，控制纹波的大小，取  $R_1=340\Omega$ ， $R_2=100\Omega$ 。

#### 3.3.2.3 输出电压采样及差分放大电路

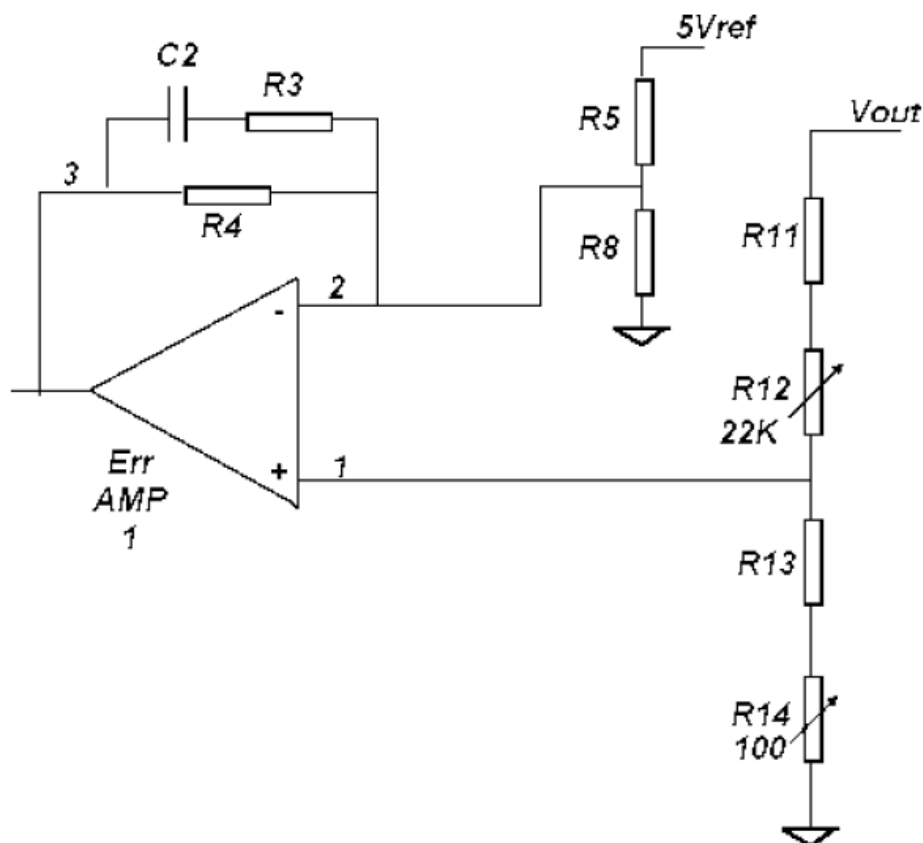


图 3.4 电压采样及差分放大电路<sup>[1]</sup>

此部分电路中，参考电压经分压与输出电压采样做比较，输入至 TL494，控制 PWM 波的占空比。在参考电压分压网络中选择  $R_5=R_8=5.1k\Omega$ ，即差分放大器反向输入端输入 2.5V 参考电压。输出为 5~10V 调压，因此  $(R_{13}+R_{14})/(R_{11}+R_{12}+R_{13}+R_{14})$  应可以取到 [0.25, 0.5]，取  $R_{11}=R_{13}=5.1k\Omega$ 。此时，当  $R_{12}$  从 0 调到 22K 时，上述比值范围为 [0.188, 1.02]，满足要求。 $C_2$  和  $R_3$  用于防治高频自激，取  $C_2=0.1\mu F$ ， $R_3=51k\Omega$ ， $R_4=1M\Omega$  <sup>[2]</sup>。

#### 3.3.2.4 储能电感

储能电感可使输出电压连续，影响输出电压波形，合理选取电感值可控制纹波的大小和输出波形。老师推荐为 1mH 左右，经仪器检测，所绕电感  $L=1.04mH$ 。

#### 3.3.3 参数列表



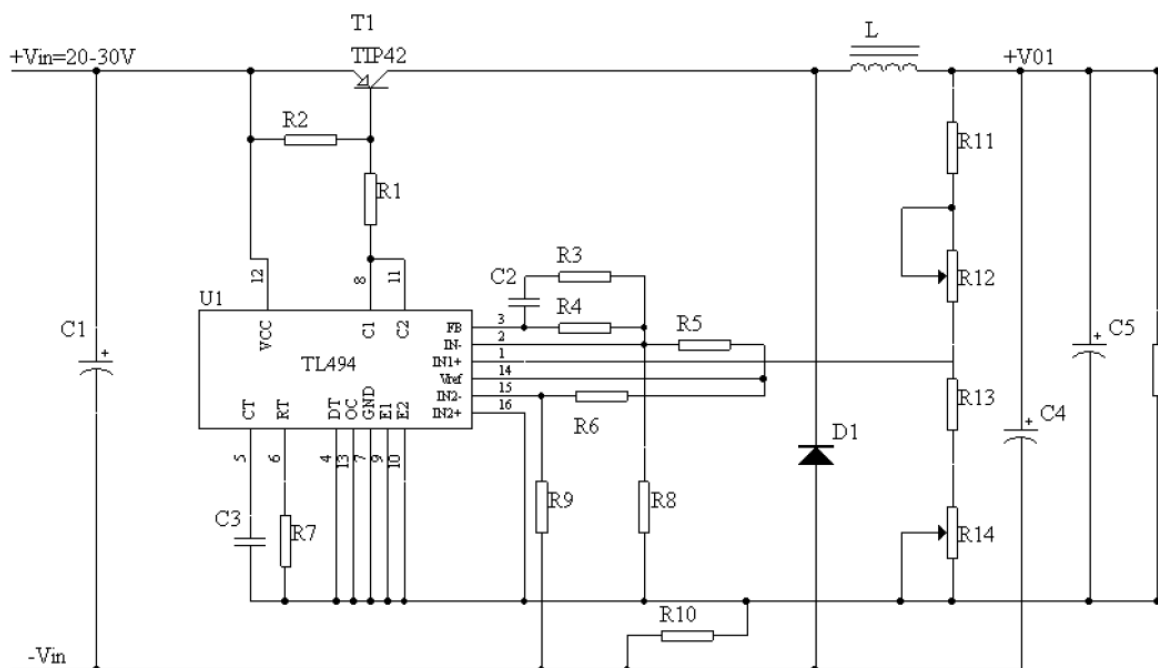


图 3.5 DC-DC 开关稳压电源元件连接标号图<sup>[2]</sup>

表 3.2 DC-DC 开关稳压电源参数列表

元件	参数	元件	参数	元件	参数
C <sub>1</sub>	100 μF	R <sub>3</sub>	51k Ω	R <sub>10</sub>	0.1 Ω
C <sub>2</sub>	0.1 μF	R <sub>4</sub>	1M Ω	R <sub>11</sub>	5.1k Ω
C <sub>3</sub>	1000pF	R <sub>5</sub>	5.1k Ω	R <sub>12</sub>	0~22k Ω
C <sub>4</sub>	100 μF	R <sub>6</sub>	5.1k Ω	R <sub>13</sub>	5.1k Ω
C <sub>5</sub>	470 μF	R <sub>7</sub>	5.1k Ω	R <sub>14</sub>	0~100 Ω
R <sub>1</sub>	340 Ω	R <sub>8</sub>	5.1k Ω	L	1.04mL
R <sub>2</sub>	100 Ω	R <sub>9</sub>	100 Ω		

### 3.4 专项讨论

在这部分电路焊接完成后，我们进行了测试，效率已达标（ $\geq 65\%$ ），但纹波峰峰值远远大于 100mV，并且波形出现尖帽状，初步判断为电感感值过大，出现磁饱和现象。通过在电感中加一层纸片，减小感值。重新测量后，纹波小于 100mV，达到要求。在此，我们没有选择提高电源频率来改善输出电压纹波，这是由于初始设置的电源开关频率为 196kHz，已经超出老师推荐的频率范

围，继续提高频率虽然可以抑制纹波，但效果并不理想，反而使得电流增大，效率降低，开关三极管也有被烧坏的危险。

此外，抑制纹波的方法还有减小开关饱和导通深度、增大滤波电容  $C_5$  等方法，在此我们没有尝试。

## 4. 电压控制模块的硬件设计

### 4.1 主要功能和设计指标

#### 4.1.1 主要功能

本模块可对单片机传至 DC-DC 开关电源模块的信号进行处理，包括整形、滤波和光电转换。

#### 4.1.2 设计指标

实现 5~10V 调压，分辨率为 0.1V，并且误差在  $\pm 0.05V$  内。

### 4.2 设计原理

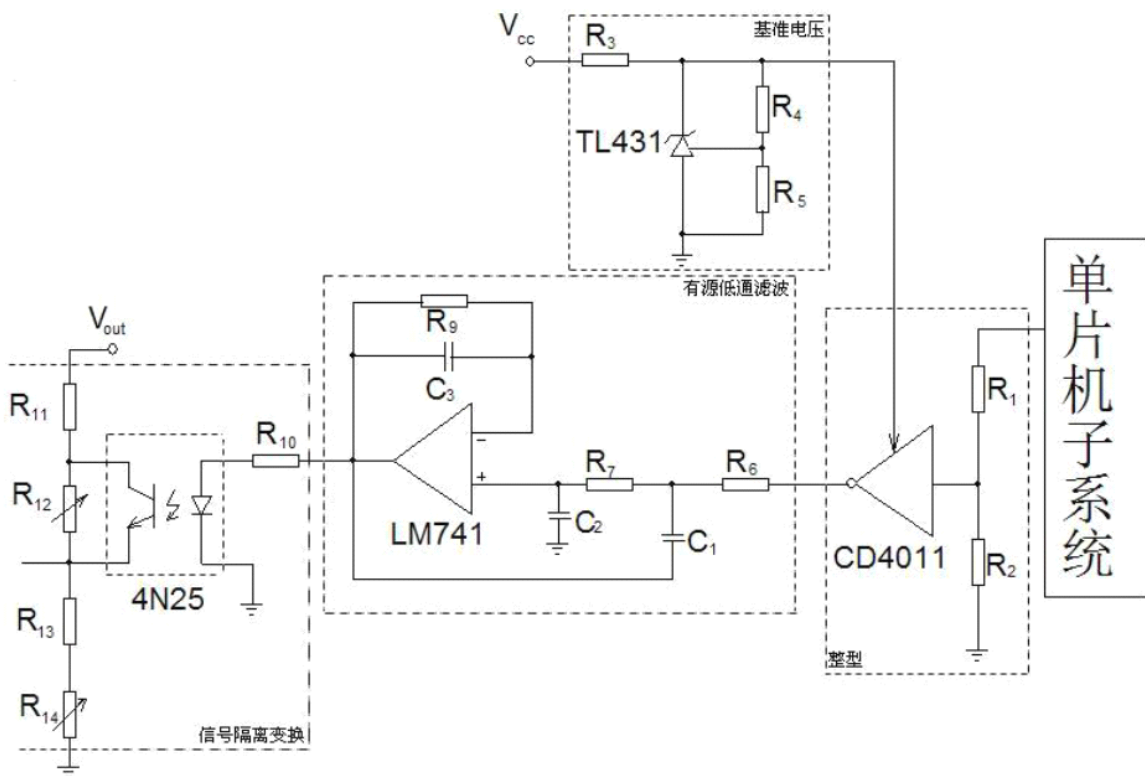


图 4.1 电压控制模块设计电路图<sup>[2]</sup>

此模块包含四个子模块，分别为整形部分、基准电压部分、有源低通滤波部分和信号隔离变换部分。按控制信号的流通方向，首先经过整形电路，将 5V 不稳定的方波信号转化成 4V 稳定的方波信号；然后经过低通滤波，使信号几乎只有直流部分；最后通过光耦进行光电信号转换实现控制。

### 4.3 电路主要部分设计及参数说明

#### 4.3.1 TL431 介绍及相关电路

TL431 是一个有良好的热稳定性能的三端可调分流基准源。它的输出电压用两个电阻就可以任意地设置到从 $V_{\text{ref}}$  (2.5V) 到36V 范围内的任何值。该器件的典型动态阻抗为 $0.2\Omega$ 。<sup>[3]</sup>

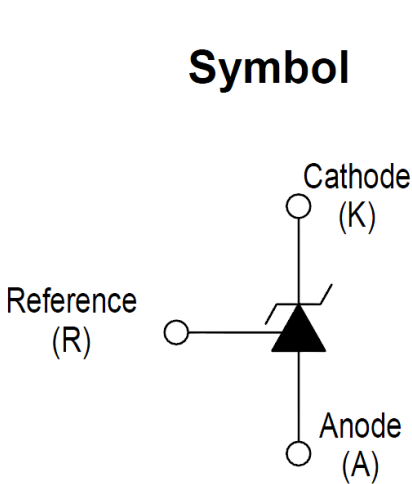


图 4.2 TL431 管脚图<sup>[3]</sup>

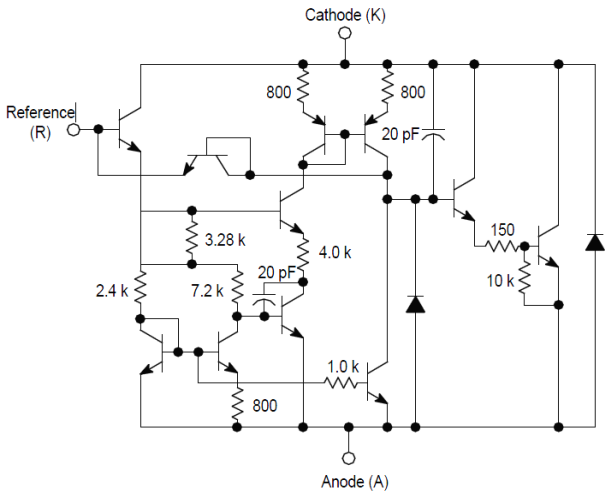


图 4.3 TL431 内部结构图

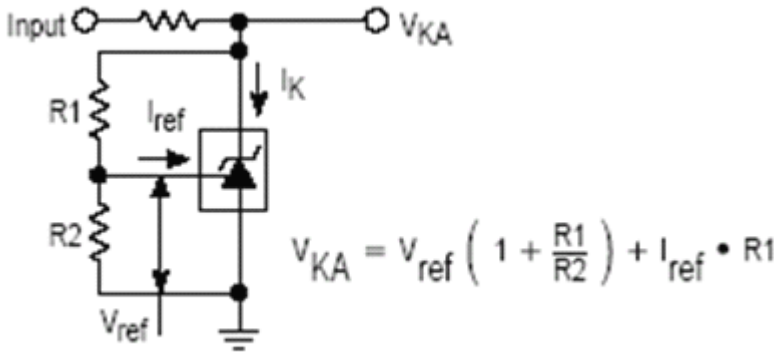


图 4.4 TL431 外围设计电路图<sup>[3]</sup>

此电路可实现 3.75V 稳定电压输出。TL431 在 R 端输出稳定的参考电压 2.5V，取  $R_1=10\text{k}\Omega$ ， $R_2=20\text{k}\Omega$ 。 $R_3$ （即 input 端串联的电阻）不宜过大，取  $100\Omega$ 。

（图 4.4 中电阻  $R_1$  和  $R_2$  分别对应设计电路中的  $R_4$  和  $R_5$ ）

### 4.3.2 CD4011B 介绍及相关电路

CD4011 是一块包含 4 个或非门的有源逻辑器件,工作时,高电平时输出  $V_{DD}$  电压 4V,低电平时输出  $V_{SS}$  电压-5V。本实验用到了管脚 1、2、3、7、14。将单片机系统传输过来的控制信号接入 1、2 号管脚则可输出反向的 4V 稳压输出。

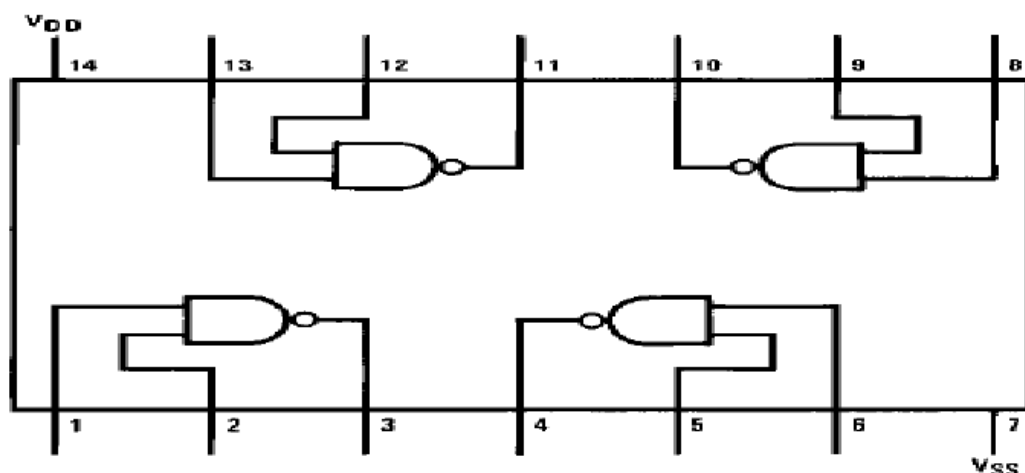


图 4.5 CD4011B 管脚和内部结构图<sup>[2]</sup>

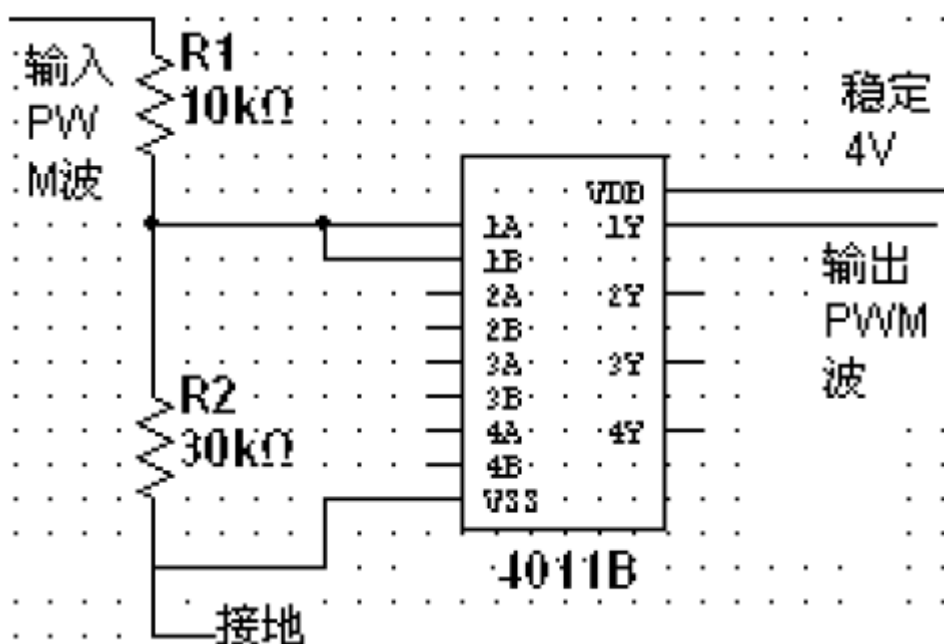


图 4.6 CD4011B 外围电路设计<sup>[2]</sup>

此电路将输入 PWM 波经分压接入 CD4011B 的一个或非门,取  $R_1=10k\Omega$ ,  $R_2=30k\Omega$ 。输入 4011B 的电压为输入的  $3/4$ , 即 3.75V, 符合规定。

### 4.3.3 低通滤波部分

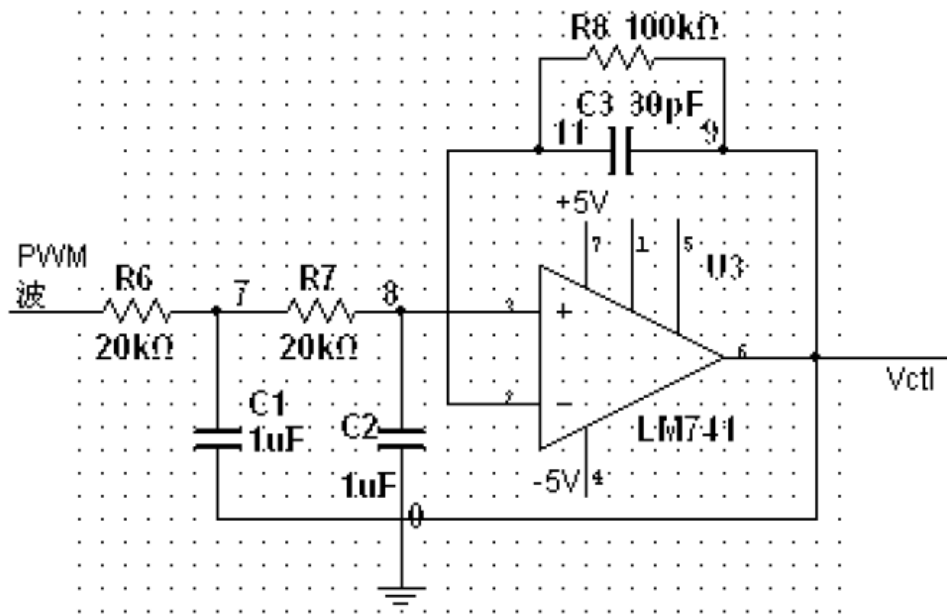


图 4.7 有源滤波设计电路<sup>[2]</sup>

此部分选择有源滤波，其优点为输出阻抗小、有放大作用。将滤波部分设计成二阶滤波，取  $R_6=R_7=20\text{k}\Omega$ ， $C_1=C_2=1\mu\text{F}$ ，得到通频带为 0~50Hz。

单独对此低通滤波部分做幅频特性：

表 4.1 低通滤波器幅频特性

频率 (Hz)	电压 (V)	频率 (Hz)	电压 (V)
1	2.80	10	1.00
2	2.20	11	0.90
3	2.00	12	0.80
4	1.88	13	0.77
5	1.80	14	0.64
6	1.70	15	0.58
7	1.42	16	0.53
8	1.29	17	0.48
9	1.13		

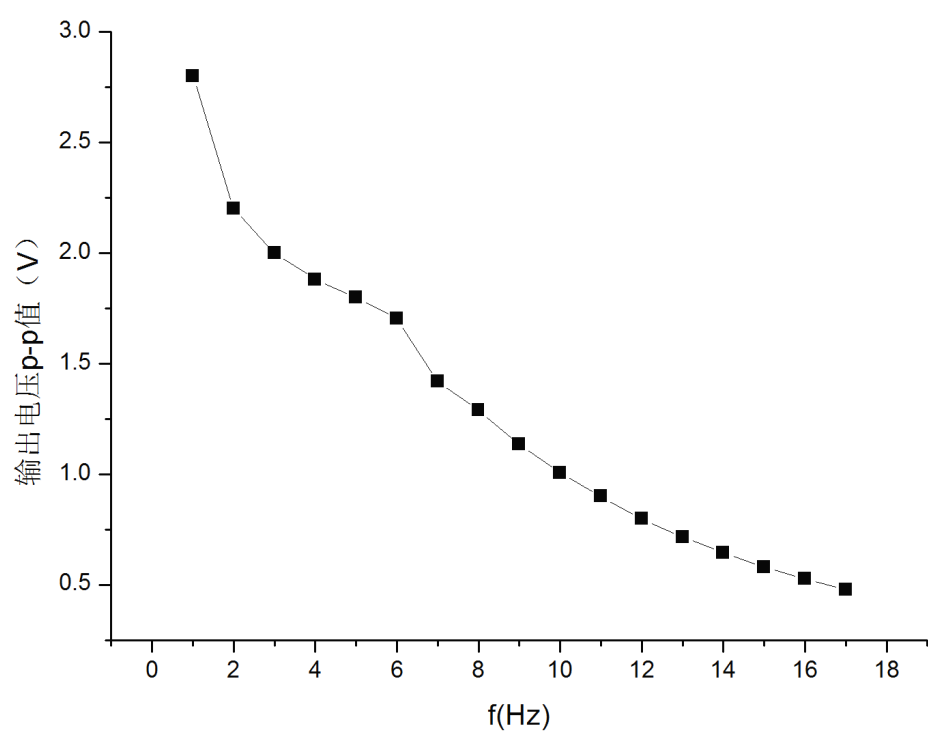


图 4.8 低通滤波器幅频特性曲线

#### 4.3.4 4n25 介绍及相关电路

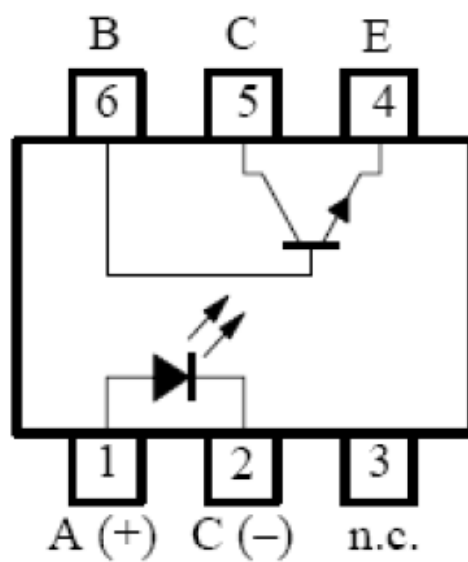


图 4.9 4n25 内部结构图<sup>[3]</sup>

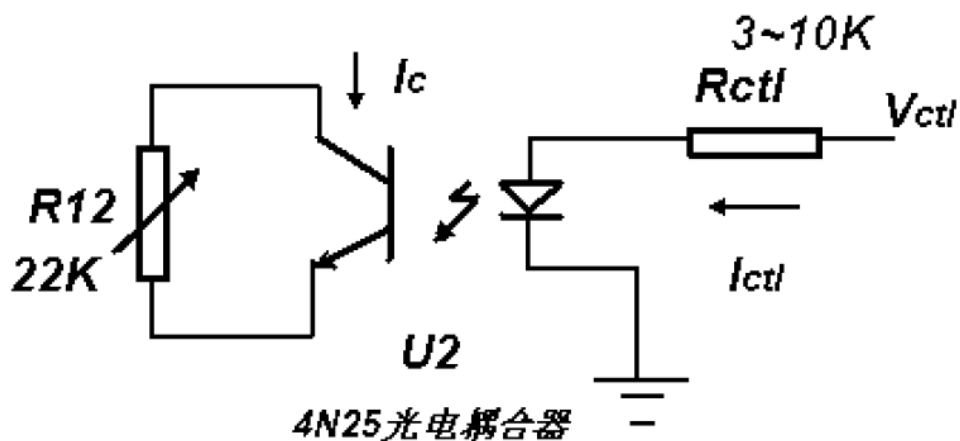


图 4.10 4n25 外围电路设计<sup>[1]</sup>

4n25 实际上是一个用光电耦合的三极管，用电流来控制电流。

Rctl的选择：Rctl如果选的太小，4n25元件中的二极管电流过大，容易烧坏；如果Rctl太大，DC-DC开关稳压电源模块的输出的可调范围达不到5~10V。取Rctl=3.9KΩ。

#### 4.3.5 参数列表

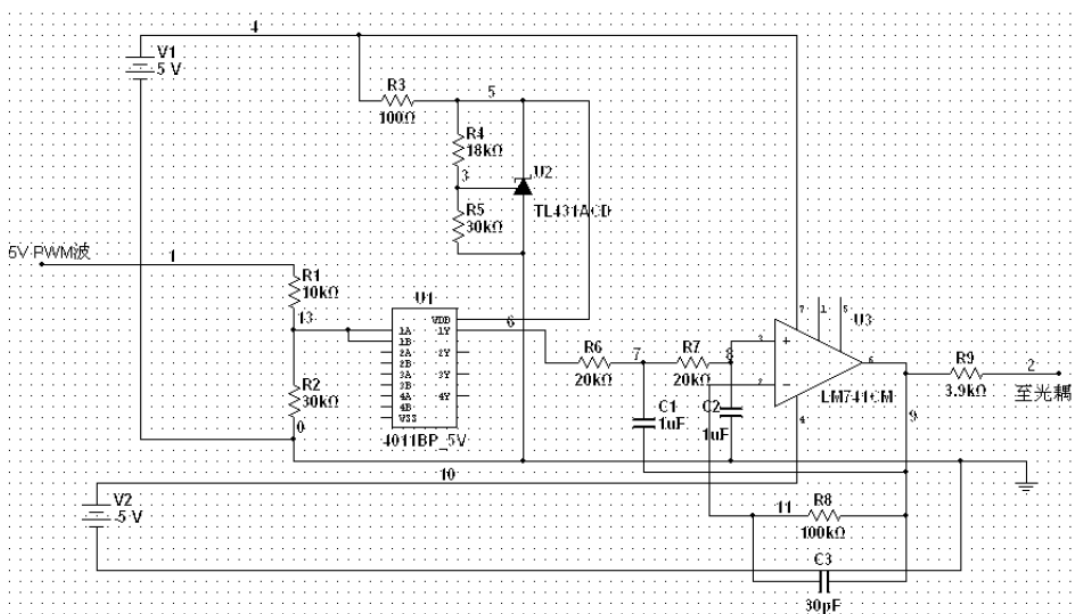


图 4.11 电压控制模块元件标号图<sup>[2]</sup>



表 4.2 电压控制模块元件参数

元件	参数	元件	参数	元件	参数
R <sub>1</sub>	10k Ω	R <sub>5</sub>	30k Ω	R <sub>9</sub>	3.9k Ω
R <sub>2</sub>	30k Ω	R <sub>6</sub>	20k Ω	C <sub>1</sub>	1 μ F
R <sub>3</sub>	100 Ω	R <sub>7</sub>	20k Ω	C <sub>2</sub>	1 μ F
R <sub>4</sub>	18k Ω	R <sub>8</sub>	100k Ω	C <sub>3</sub>	30pF

## 5. 电压测量模块的硬件设计

### 5.1 主要功能和设计指标

#### 5.1.1 主要功能

此模块可产生一个稳定的 3.75V 参考电压，以及对 DC-DC 开关电源输出的反馈电压信号变换，两者分别送入单片机的两个 ADC 转换端口。

#### 5.1.2 设计指标

输入电压 5~10V

AD 转换精度  $\pm 0.02V$

### 5.2 设计原理

通过对反馈电压和参考电压的处理，分别输入到单片机系统 ATmega16 的两个 A/D 转换口，进行差分输入，可以用足 10 位的 ADC 编码。

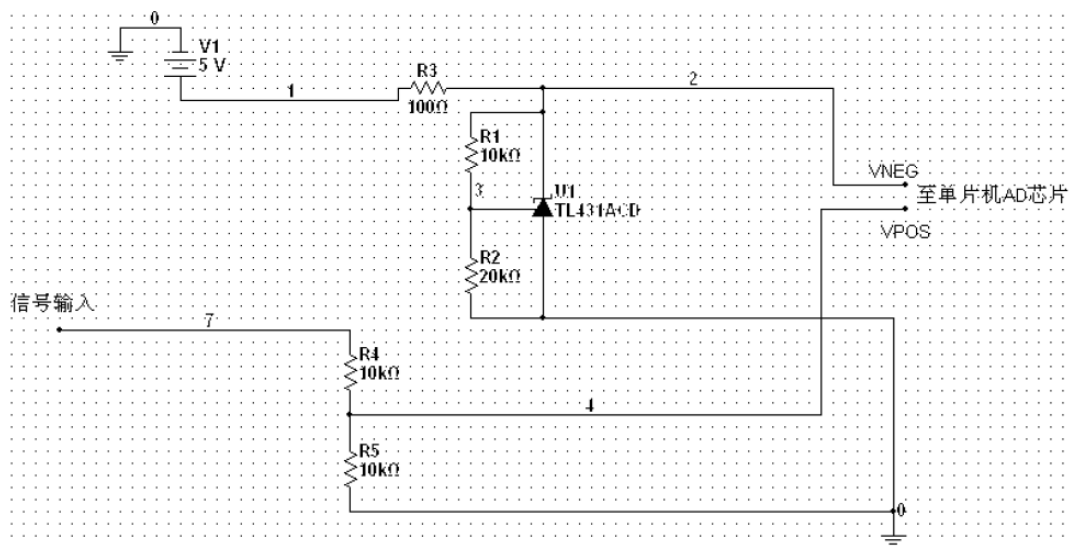


图 5.1 电压测量模块的电路设计<sup>[2]</sup>

### 5.3 电路主要部分设计及参数说明

表 5.1 电压测量模块参数列表

元件	参数
$R_1$	$10\text{k}\Omega$
$R_2$	$20\text{k}\Omega$
$R_3$	$100\Omega$
$R_4$	$10\text{k}\Omega$
$R_5$	$10\text{k}\Omega$

#### 5.3.1 电压反馈子模块

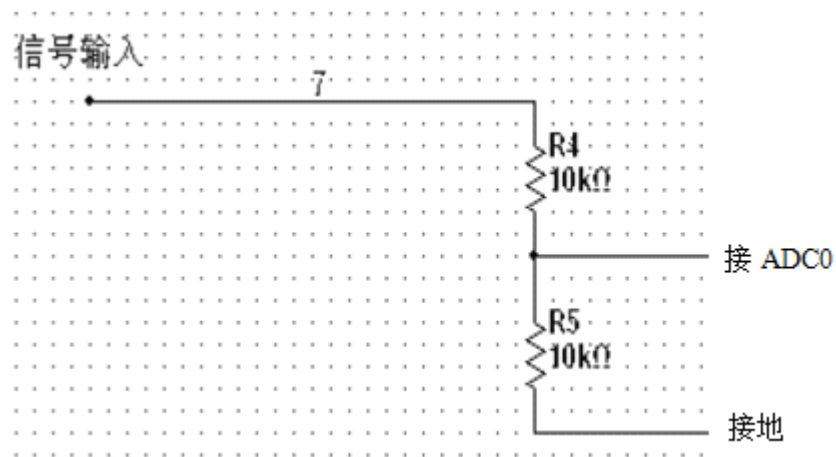


图 5.2 电压反馈子模块电路设计

输入信号为  $5\sim 10\text{V}$ ，取  $R_4=R_5=10\text{k}\Omega$ ，得到输出至单片机 ADC0 端口的电压范围为  $2.5\sim 5\text{V}$ 。

#### 5.3.2 基准电压子模块

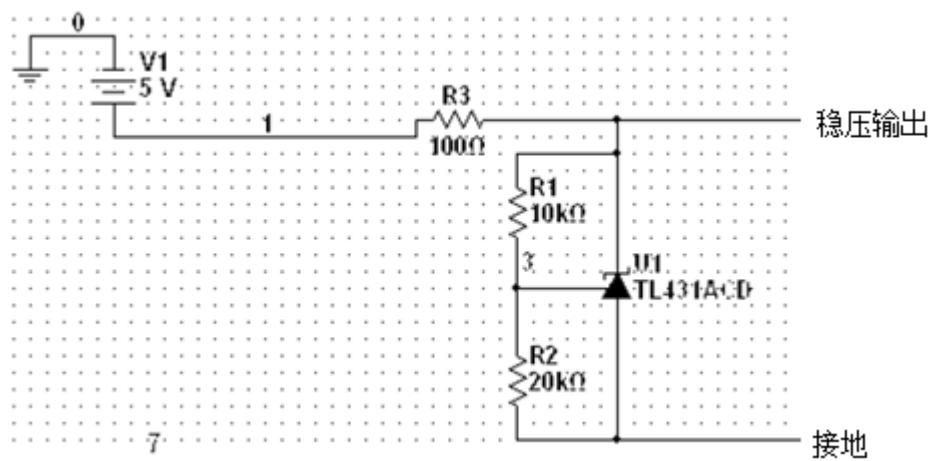


图 5.4 基准电压子模块电路设计

此电路与电压控制模块的参考电压电路设计思路相同，在此不再赘述。得到稳压输出 3.75V。

## 6. 单片机小系统及软件设计

### 6.1 单片机小系统<sup>[9]</sup>

本实验使用 AVR ATmega16 单片机，这款单片机具有高性能、低功耗的特点。ATmega16 有 16KB 的程序存储空间，1KB 内部 SRAM，512B 内置 EEPROM。外部共有 32 个 GPIO，一路 USART，一路主从 SPI，一路 I<sup>2</sup>C，两个 8 位定时器，一个 16 位定时器，4 通道 PWM 输出，8 路 10 位 AD 输入。本报告就所用到的资源做简单讲解。

#### 6.1.1 I/O 端口

AVR ATmega16 单片机共有 4 个 8 位可编程 I/O 端口，每个端口都有三个 I/O 存储器地址：数据寄存器-PORTx、数据方向寄存器-DDRx 和端口输入引脚-PINx。数据寄存器和数据方向寄存器为读/写寄存器，端口输入引脚为只读寄存器。。每个端口可用作独立的输入输出引脚来使用，部分端口还能作为单片机其他模块（如 A/D 模块、计数器）的输入输出引脚使用。

#### 6.1.2 16 位定时/计数器

AVR ATmega16 单片机拥有两个具有独立预分频器和比较器功能的 8 位定时器/计数器和一个具有预分频器、比较功能和捕捉功能的 16 位定时器/计数器。每个定时器/计数器都可以工作在多种模式，实验中用到了普通模式和快速 PWM 模式这两种。

##### 1. 普通模式

以定时器/计数器 0（8 位）为例，在此模式下计数器不断累加，当达到 8 位的最大值（TOP=0xFF）后，计数值溢出，并从最小值 0x00 开始重新计数。在溢出的同时，定时器/计数器 0 的溢出中断标志位 TOV0 置位，用户可利用此标志位控制程序产生中断。在中断程序中，用户也可以定义计数器 0 的计数初值来改变中断周期。

##### 2. 快速 PWM 模式

ATmega16 单片机自带 PWM 波的产生模式，用户只需要设定定时器/计数器的寄存器值，就能快速地得到不同占空比的 PWM 波，而不再需要对中断进行繁琐的操作了。

以定时器/计数器 1（16 位）为例，在此模式下，计数器从 BOTTOM 值计数到 TOP 值，然后立即回到 BOTTOM 重新开始。对于普通的比较输出模式，输出比较引脚 OC1x（PWM 波从此引脚输出）在计数器寄存器 TCNT1 与比较匹配寄存器 OCR1x 匹配时置位，在 TOP 时清零。TOP 值可固定地设为 0x00FF、0x01FF 或 0x03FF，也可由 ICR1 定义。不同的 OCR1x 和 ICR1 组合可以产生不同频率和占空比的 PWM 波。同样，当计数器计到 TOP 或 OCR1x 时，用户可通过中断标志位产生中断。

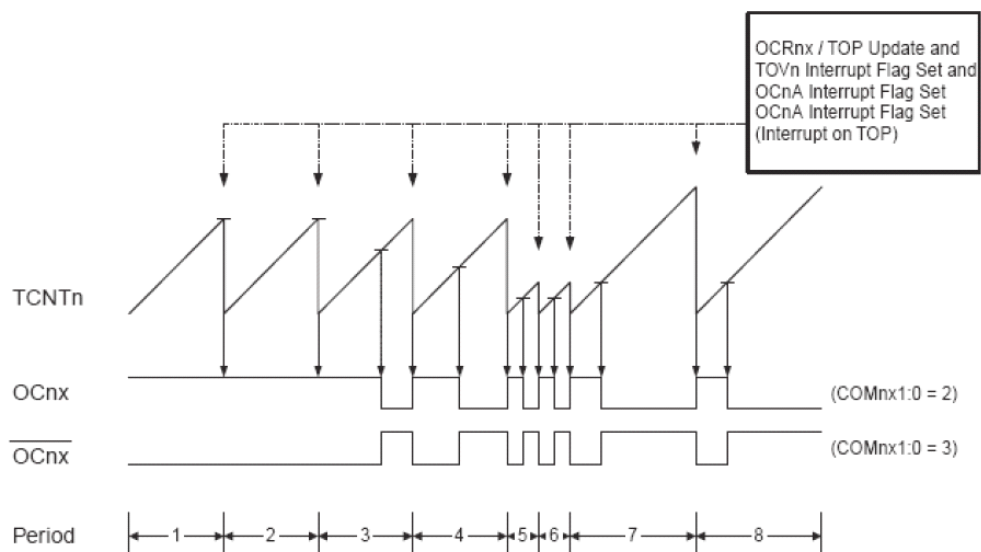


图6.1 快速PWM模式时序图<sup>[3]</sup>

### 6.1.3 A/D 转换

ATmega16单片机提供了8路复用的单端输入通道，7路差分输入通道，以及2路可选增益为10倍与200倍的差分输入通道，参考电压为2.56V。对寄存器ADCSRA写入值之后，可以启动一次模数转换，普通情况下转换时间为13个时钟周期，转换结果为十位，保存在ADCH和ADCL两个寄存器中，由程序读取。

### 6.1.4 7段数码管及按键

ATmega16还提供了4位七段数码管显示器，4个输入按键以及1个重置按键。4个按键分别连接I/O口的PC4~PC7供用户使用。

## 6.2 资源分配

1. 定时器/计数器0产生5ms中断，供程序调用中断服务子程序；
2. 定时器/计数器1产生占空比可调的PWM 信号，由I/O 端口PD4输出；
3. A/D转换采用I/O端口PA0和PA1进行差分输入，PA0接输入电压，PA1接恒定的比较电压；
4. 按键1~4分别为电压+0.1，电压-0.1，开闭环切换，电压输出期望值/电压输出实际值/相对占空比间切换；

5. 数码管显示设定电压值。

6.3 软件设计

6.3.1 软件结构及功能

单片机软件部分可以通过开环或闭环的方式对 DC-DC 开关稳压电源输出的进行控制。用户可以通过按键，自主选择开环或闭环的方式，并且设定期望的输出电压。下图为整体的软件系统流程图。

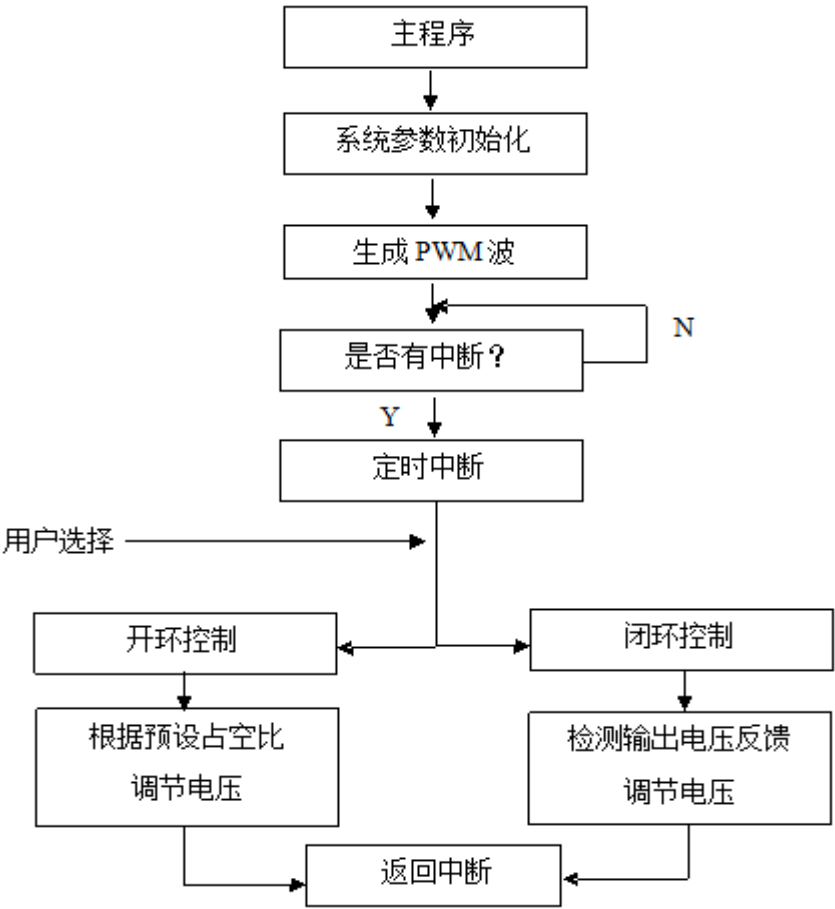


图 6.2 软件结构图

重要的全局变量：

表 6.1 重要全局变量表

类型	变量名	意义
unsigned int	duty_set	占空比数组
const	V_Tls	定时器溢出值
unsigned char	digi[4]	计数值

unsigned int	Volt	开关电源输出电压
unsigned char	output_sel	数码管和指示灯显示
unsigned char	output_8seg	数码管段显示
unsigned char	led1,led2,led3,led4	指示灯驱动
unsigned char	digi_scaner	数码管扫描驱动指针
int	keycount	按键防抖计数
unsigned char	CLC_flag	开环/闭环选择
long	SMPL_Volt	系统输出电压测量值
long	AD_value	AD 值采样
int	AD_data	ADC 存储数组
int	ADC_ptr	ADC 数组指针

### 6.3.2 主函数

程序每次运行，主程序首先对各个参数进行初始化，默认输出的控制信号为开环 5V。进入主循环后，不断扫描 7 段数码管，显示当前电压，等待中断请求。一旦用户选择了某一项功能，立即进入中断服务子程序。

### 6.3.3 开环程序

开环程序可以根据编程时预设的值进行电压的固定控制。但在实际中，往往由于环境、人为等因素，要维持某一个电压水平需要的 PWM 波占空比会稍有偏差。因此，开环控制很难做到十分精确地实现电压控制。

用户可以通过按键 1 和 2 实现电压增加或减小 0.1V。为了防止按键时的抖动，在此部分我加入了按键消抖子程序。设置每 5ms 执行一次中断，检测某一个键是否被按下。如果按下，则变量 keycount 加 1，直到按键被松开。根据预设的一个阈值，当变量 keycount 的值小于此阈值时被认为是按键抖动，操作无效。每次按键结束，都要执行 keycount 的清零。

### 6.3.4 闭环程序

闭环程序在开环程序的基础上，增加了计算系统输出电压反馈信号与期望信号的差值，并相应的微调占空比，达到较高的电压输出精度。闭环模块的加入，大大改善了开环控制所无法避免的其他因素引起的电压漂移。



## 1、系统时延及占空比调整

当用户选择闭环工作时，系统将对输出电压采样，与期望值进行对比做差，根据差值的大小，进行占空比的粗调和微调。当两个值相等时，占空比将不再改变。此过程为一个动态过程，当控制信号发出后，系统会有一个微小的作用延时，如果程序中控制信号的发出频率过高，可能当发出好多条控制指令后，系统输出端还未接收到第一条指令，这样会造成很严重的超调；相反，如果频率低，那么系统相应的时间就长，也就是达到期望值的调节时间长，也不可取。因此，要控制好调整的频率。经过多次实验测量，将每次调整时间设定为 0.5 秒。

## 2、A/D 转换

对于ATmega16单片机，对寄存器ADCSRA写入值即可启动一次A/D转换。在实际的程序中，只需执行ADCSRA=0xc6语句即可。在普通模式下，A/D转换的转换时间为13个时钟周期，时钟频率为8MHz。为了避免等待转换时间，采取了如下的方法：参考图6-3的闭环算法部分，在主程序执行初始化任务时，就启动一次A/D转换，之后每次执行中断程序时，都先读取上一次的转换结果，然后再启动下一次的A/D转换。由于中断子程序每5ms 执行一次，所以在两次中断之间，A/D转换一定可以完成，这样就省去了在启动转换后等待的时间。

在实际的操作过程中，考虑到采样时存在高斯白噪声的干扰，所以不能只对输出电压进行一次采样，而是要进行多次采样，取滑动平均。滑动平均的示意图如图6.3所示。在程序中，采用了数组ADC\_volt[50]存放A/D转换的结果，使用了指针ADC\_ptr代表当前数组中的位置，其数据结构为一循环队列：当读取一次A/D转换结果后，取代数组中最“旧”的一个数据，然后对此数组取平均值即为滑动平均。

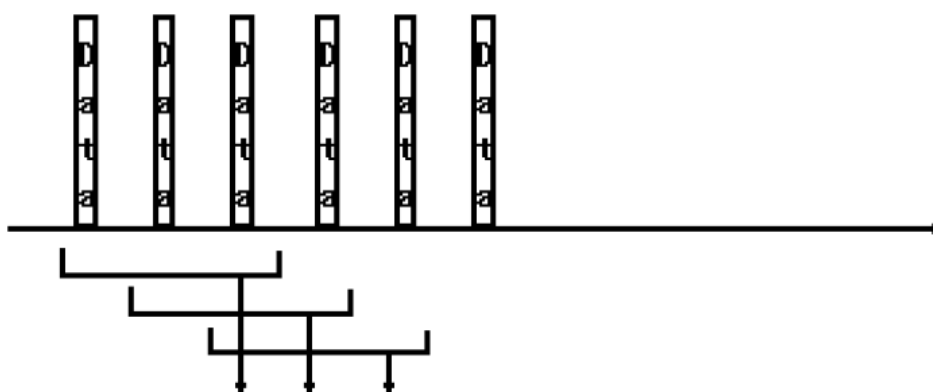


图 6.3 滑动平均示意图

## 7. 致谢

真的很感谢实验期间帮助过我们的每一位老师、助教和同学。尤其要感谢一位助教，那次我不小心将 30V 带电的电源线碰到了 ATmega16 的某一管脚，一瞬间芯片就烧掉了。买新的单片机时，我向助教询问有关烧熔丝的方法，虽然他对这方面不是很了解，但还是查阅书籍、上网搜索，历时半小时之久，终于烧好了！非常感谢！

依然要感谢自始至终为我们讲授知识的袁焱老师，带领我们从零开始，一步步走向成功！

最后还要感谢学长学姐们，因为你们在前面铺平了道路，我们才能走得如促顺利！

## 8. 参考文献

- [1] 上海交大电子工程系. 科技创新 5 讲义[EB/OL].<ftp://202.120.39.248>.
- [2] 上海交大电子工程系. 科技创新[5]-第 109 组（组长侯英杰）设计报告.
- [3] 上海交大电子工程系. 科技创新 5 部分器件资料[EB/OL].<ftp://202.120.39.248>.

## 9. 附录 A 系统操作说明书

### 1、按键说明：

KEY0：此按键可增加电压 0.1V。

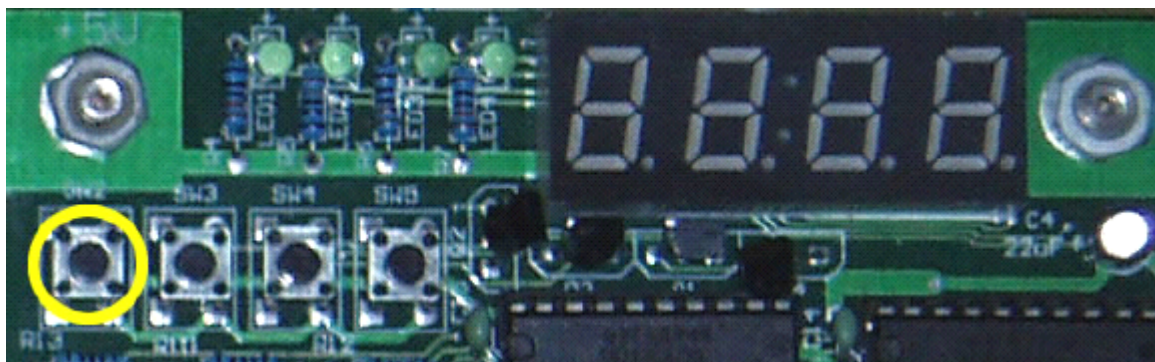


图 9.1 KEY0 按键示意图<sup>[2]</sup>

KEY1：此按键可减小电压 0.1V

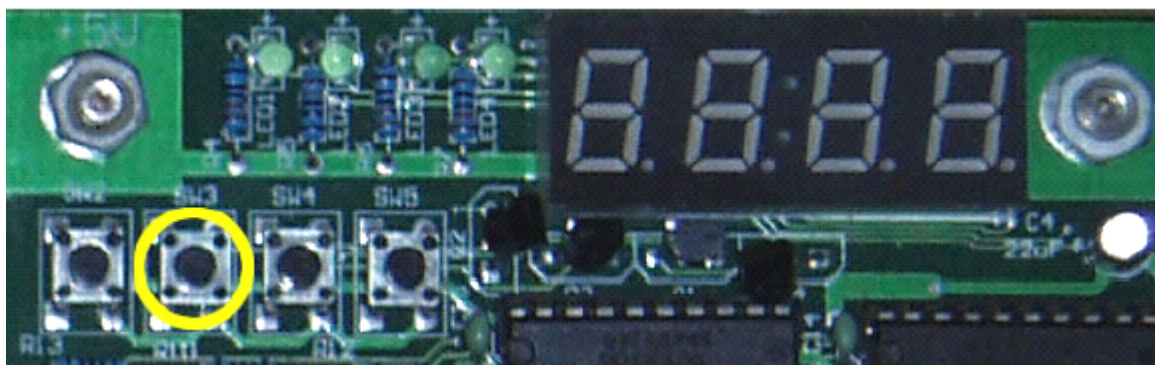


图 9.2 KEY1 按键示意图<sup>[2]</sup>

KEY2：此按键可选择开环或闭环模式

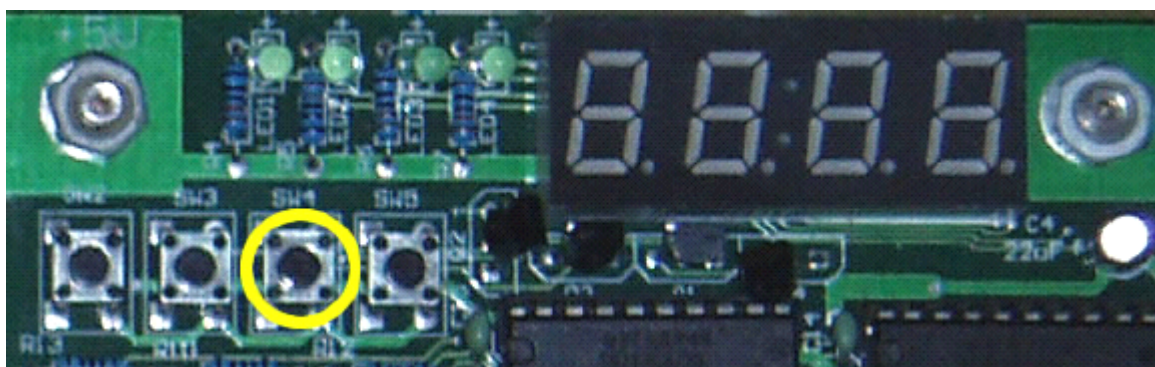


图 9.3 KEY2 按键示意图<sup>[2]</sup>

KEY3: 显示切换（包括期望电压，实测电压，占空比）

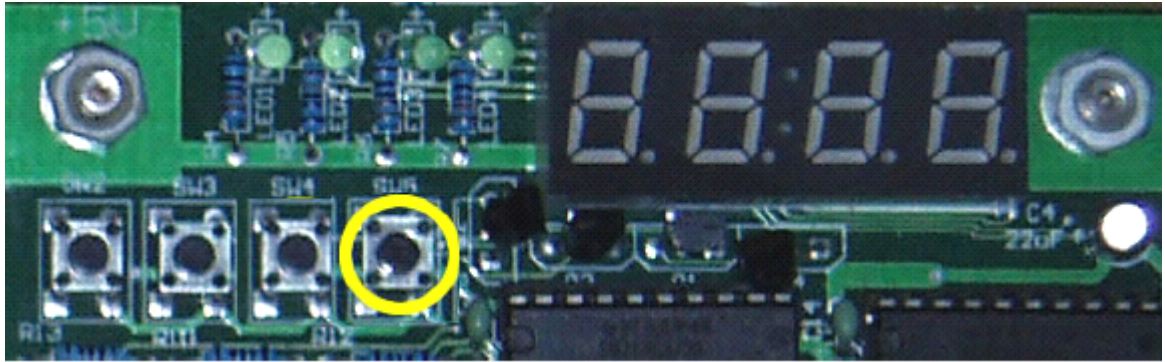


图 9.4 KEY3 按键示意图<sup>[2]</sup>

# 10. 附录 B 测试和分析

## 10.1 测试项目和方法<sup>[1]</sup>

### 1. DC-DC 模块独立测试

测试目的：检验开关电源电路的功能和达标情况。

表 10.1 DC-DC 模块独立测试和评分

项 目		评分说明	测试方法或条件
输出电压可调范围（5分）	最低输出 ≤ 5.0V；最高输出 ≥ 10.0V	上下限之一不达标，扣 2 分；上下限均每不达标，扣 5 分	输入 30.0V±0.1V 人工调整相应电位器
输出纹波（5 分）	小 于 等 于 100mV	超过限值扣 2 分；每高过限值 20mV，多扣 1 分	输入 30.0V±0.1V；输出 10.0V±0.1V；TDS 系列 60 或 100MHz 带宽示波器，探头 X10，Y 向 20 或 50mV 档
效率（5 分）	大于 65%	低于限值扣 2 分；每低过限值 5 个百分点，多扣 1 分	输入 30.0V±0.1V；输出 5.0V±0.1V；测量输入电压、输入电流、输出电压，负载以 10Ω计。
电压调整率（5 分）	小于 1%	超过限值扣 2 分；每高过限值 0.1 个百分点，多扣 1 分	输出 10.0V±0.1V；输入 20. 0V±0.1V 到 30.0V±0.1V；
工艺（5 分）		满分要求：在各项指标良好的前提下，焊点匀称可靠，元件装列整齐，电感绕制做工良好	主观评价
特色记录（指标特别优异，或其他自创设计）		可自行提出，由老师验证并记录。	
注[1]：输出电压过大波动等异常情况，酌情扣分。对各种必须记录的状况，测试时不打分，事后综合评价。			

### 2. 电压测控

测试目的：检验电路的功能和达标情况。

表 10.2 电压测控功能的测试和评分

项 目		评分说明	测试方法或条件
开环控制精度（10 分）	输出=设点电压 $1^{[\text{注}2]} \pm 0.05\text{V}$	超过限值扣 2 分；每高过限值 $0.05\text{V}$ ，多扣 1 分	断开电压测量回路； 输入 $30.0\text{V} \pm 0.1\text{V}$ ； 开始测试时，任意指定 3 个电压值分别作为设点电压
	输出=设点电压 $2^{[\text{注}2]} \pm 0.05\text{V}$	超过限值扣 2 分；每高过限值 $0.05\text{V}$ ，多扣 1 分	
	输出=设点电压 $3^{[\text{注}2]} \pm 0.05\text{V}$	超过限值扣 2 分；每高过限值 $0.05\text{V}$ ，多扣 1 分	

项 目		评分说明	测试方法或条件
闭环控制 绝对精度（10分）	输出=设点电压 $A^{[注2]} \pm 0.05V$ (实际输出记作 V1)	超过限值扣 2 分；每高 过限值 0.05V，多扣 1 分	接入电压测量回路； 输入 30.0V±0.1V；  开始测试时，任意指定 3 个电压分别作为设点电压； 记录实际输出的 3 个电压 值。
	输出=设点电压 $B^{[注2]} \pm 0.05V$ (实际输出记作 V2)	超过限值扣 2 分；每高 过限值 0.05V，多扣 1 分	
	输出=设点电压 $C^{[注2]} \pm 0.05V$ (实际输出记作 V3)	超过限值扣 2 分；每高 过限值 0.05V，多扣 1 分	
电压自动 调整的精度 （10分）	设点电压 A 输出= $V1 \pm 0.05V$	超过限值扣 2 分；每高 过限值 0.05V，多扣 1 分	调偏 R14，重做指定电压 的测试； 调偏后系统输出应在 30 秒 内达到稳定。
	设点电压 B 输出= $V2 \pm 0.05V$	超过限值扣 2 分；每高 过限值 0.05V，多扣 1 分	
	设点电压 C 输出= $V3 \pm 0.05V$	超过限值扣 2 分；每高 过限值 0.05V，多扣 1 分	
工艺（5分）		满分要求：在各项指标 良好的前提下，焊点匀称可 靠，元件装列整齐，走线清 晰不杂乱	主观评价
特色记录（指标特别优异，或其他自创设计）		可自行提出，由老师验证并记录。	
注[1]：输出电压过大波动等异常情况，酌情扣分。对各种必须记录的状况，测试时不打分，事后综合评价。 注[2]：由老师当场指定			

3. 其它拓展功能或研究的验证  
 测试时只记录不评分，总评时酌情加分（最高 15 分）。  
 请学生自拟测试流程和记录表。

## 10.2 测试的资源

### 10.2.1 测试资源

- 1、万用表
- 2、直流稳压电源
- 3、电源线及示波器探头
- 4、示波器

### 10.2.2 测试环境

直流稳压源  $\pm 5V$  和  $20 \sim 30V$  输出，示波器探头  $\times 10$

### 10.3 测试结果及分析

DC-DC 模块独立测试			
测试条件		项 目	记 录
输入电压	输出电压		
30.0V±0.1V	——	(5分) 输出电压可调范围 5.0V 至 10.0V	4.5~10.1V
20.0V±0.1V 变化到 30.0V±0.1V	设定 10.0V	(5分) 电压调整率 电压变化应小于 0.05V, 记录输出电压	9.90~9.94V
30.0V±0.1V	10.0V±0.1V	(5分) 输出纹波小于等于 100mV	≤98mV
30.0V±0.1V	5.0V±0.1V	(5分) 效率大于等于 65%, 输入电流不大于 0.13A, 记录输入电流	30V/0.12A
——	——	(5分) 工艺	
		其他情况记录 (如选测限流项目等)	单独测量 DC-DC 部分不工作, 需接入电压控制部分。
电压测控 (测试前给予准备时间, 允许进行修改程序等操作)			
测试条件		项 目	记 录
输入电压	输出电压设点		
30.0V±0.1V 电路为开环控制状态	( 5.6 )	(10分) 开环控制下输出电压绝对偏差, 记录输出电压	5.51V
	( 7.7 )		7.63V
	( 9.8 )		9.79V
30.0V±0.1V 电路为闭环控制状态	V1= ( 5.6 )	(10分) 闭环控制下输出电压绝对偏差, 老师指定 3 个电压, 记录输出电压	5.57V
	V2= ( 7.7 )		7.68V
	V3= ( 9.8 )		9.81V
30.0V±0.1V 电路为闭环控制状态, R14 调偏	V1	(10分) 闭环控制下输出电压相对偏差, 重做 3 个指定电压的测试, 记录输出电压	5.56V
	V2		7.66V
	V3		9.78V
——	——	(5分) 工艺	
		其他情况记录	
其他拓展功能 (测试前给予准备时间, 学生自拟测试项目, 此页不够可自行附页)			



测试条件	项 目	情况记录

最终的检测结果并不很理想，纹波和效率均达标，但开环和闭环控制很多点不达标，开环的控制精度并不高，每次开机、开机时间长短都会带来偏差；闭环控制的值比期望的值普遍偏小，是由于单片机软件部分的某些参数没有设置好。DC-DC 开关稳压电源不能独立工作可能的原因有开关三极管烧坏、电流过载保护。测试时接入控制部分电路可以工作，说明开关三极管没有烧坏；单独调节开关电源，将直流稳压源过载电流设置到 1.2A 依然不能正常工作。至今还未找出问题所在。

## 11. 附录 C 课程学习心得和意见建议

心得体会：

1.做事要稳重，不能急躁。有时候焊电路焊错了，或者电子元件比较多，内心难免会急躁。忙中出乱，可能会引起更加大的错误，甚至直接使实验失败。所以做事，特别是需要仔细和耐心的事情，一定要安下心来，全身心投入到学习或者实验中去。

2.失败是在所难免的，比如说第一块板子由于实验失误烧掉了，当时内心十分懊恼。但是后来，重新买了一块电路板焊接，实验时确保电流和电压，防止电流或者电压过大，终于使线路板成功实现，并且示数比较准确。我想，即使失败了，我们也不需要着急难过，重新来过，吸取上一次的教训，一定可以在后面的尝试中获得成功。

3.电路板的焊接让我想到了生活，每一处细微的环节有可能决定着最终是否会成功。元件焊错一个就可能整个电路失效，成品报废。生活也是如此，只有抓住细节，努力做好每一个细小的环节，才能构造起伟大的成功。

4.同学之间的相互帮助相互答疑，使得遇到的许多问题都得到了解决的罢办法。所以，同学或者团队之间的合作是极其重要的。一个人的力量是微小的，但是大家一起努力，就可以克服前进过程中的许多困难和问题，最终获得成功。

从本次科创实验中，我不仅品尝到成功的喜悦，也获得了许多人生的思考。从繁杂的书本中脱身而出的我，感受到一种别样的乐趣，不只是实验带给我们的乐趣，也是实验带给我们的思考和财富。

建议：

有些实验仪器如示波器已经损坏，导致了同一个开关电源在不同示波器上测出的纹波大不相同，希望能尽快修理，以备后续实验的正常进行。

## 12. 附录 D 软件程序清单

```
#include <avr/io.h>
#include <avr/interrupt.h>

//-----常量定义: 定时器-----
#define V_T1s 200 // 1s 软件定时器溢出值, 200 个 5ms
//-----

//-----变量定义: 定时器-----
unsigned char clock1s; // 1s 软件定时器计数
unsigned int test_counter; // 测试用计数器
//-----

//-----变量定义: 数码管显示-----
unsigned char output_sel; // 数码管位和指示灯显示数据变量
unsigned char output_8seg; // 数码管段显示数据变量
unsigned char volatile clock1s_flag; // 1s 软件定时器溢出标志
unsigned char led1,led2,led3,led4; // 指示灯驱动信号输出缓存
unsigned char digi_scanner; // 数码管扫描驱动指针
unsigned char volatile digi[4]; // 测试用计数值十进制表示
unsigned char KEY1,KEY2,KEY3,KEY4; // 4 个按键变量
unsigned char PWM_OUT=0;
//-----

//-----常量定义: 闭环采样-----
#define v_keycount 4 // 按键采样相同判定个数
#define AD_SMPL_NUM 50 // 电压采样个数
#define CLC_DELAY 150 // 闭环控制延迟因子
//-----

//-----变量定义: 按键状态标志和计数器-----
unsigned char key1_flag=0;
unsigned char key2_flag=0;
unsigned char key3_flag=0;
unsigned char key4_flag=0;
unsigned char key1_count=0;
unsigned char key2_count=0;
unsigned char key3_count=0;
unsigned char key4_count=0;
//-----

//-----变量定义: 按键当前状态和前一状态-----
int cur_sample1=0;
int pre_sample1=0;
int cur_sample2=0;
int pre_sample2=0;
int cur_sample3=0;
int pre_sample3=0;
int cur_sample4=0;
int pre_sample4=0;
//-----

//-----变量定义: 定时器与电压-----
unsigned int Volt; // 系统输出电压
unsigned int hilv_time; // 定时器 1 重装值变量
unsigned int lolv_time;
unsigned int hilv_th;
```

```

unsigned int lolv_th;
int state;
unsigned int Volt=500; /*任务变量赋初值,占空比 25.0%,电压 5.0V*/
long Prop=2478; //单片机输出信号占空比
long SMPL_Volt; /*系统输出测量电压*/
unsigned char setting_flag=0; /*设定和显示状态标志*/
unsigned char refresh_flag=1;
unsigned char change_flag=1;

/*开环查表*/
unsigned int Duty_Set[51]={
2520, //5.0~5.4
2570,
2610,
2660,
2710, //5.5~5.9
2760,
2810,
2860,
2910,
2960, //6.0~6.4
3010,
3050,
3090,
3130,
3170, //6.5~6.9
3210,
3250,
3290,
3330,
3370, //7.0~7.4
3420,
3470,
3520,
3570,
3620, //7.5~7.9
3670,
3720,
3770,
3820,
3870, //8.0~8.4
3920,
3970,
4020,
4075,
4130, //8.5~8.9
4185,
4240,
4295,
4350,
4405, //9.0~9.4
4460,
4510,
4570,
4620,
4680, //9.5~10.0
4745,
4810,
4875,
4940,
5005,
5070};

```

```

unsigned char i,j; /*临时贮存用变量*/

//-----变量定义: 模数转换部分与开闭环 -----
int AD_data[AD_SMPL_NUM];
long AD_value=0;
unsigned char volatile AD_flag;
unsigned char AD_ptr=0;
unsigned char AD_pflag=1;
unsigned char CLC_flag=0; /* 开\闭环控制变量 */
unsigned char DelayCounter=0;
//-----

/*****
//-----函数定义-----

// 函数名: 7 段数码显示译码
// 参 数:
// DATA: 需要显示的数字或符号;
// 返回值: 7 段译码结果 ( D7~0 = PGFEDCBA )
unsigned char NUMTOSEG7(unsigned char DATA)
{
    unsigned char AA;
    switch (DATA)
    {
        case 0: AA=0xc0;break; /* '0' */
        case 1: AA=0xf9;break; /* '1'and'i' */
        case 2: AA=0xa4;break; /* '2' */
        case 3: AA=0xb0;break; /* '3' */
        case 4: AA=0x99;break; /* '4' */
        case 5: AA=0x92;break; /* '5'and's' */
        case 6: AA=0x82;break; /* '6' */
        case 7: AA=0xf8;break; /* '7' */
        case 8: AA=0x80;break; /* '8' */
        case 9: AA=0x90;break; /* '9' */
        case 10: AA=0x88;break; /* 'A'and'R' */
        case 11: AA=0x83;break; /* 'B' */
        case 12: AA=0xc6;break; /* 'C' */
        case 13: AA=0xa1;break; /* 'D' */
        case 14: AA=0x86;break; /* 'E' */
        case 15: AA=0x8e;break; /* 'F' */
        case 16: AA=0x40;break; /* '0.' */
        case 17: AA=0x79;break; /* '1.' */
        case 18: AA=0x24;break; /* '2.' */
        case 19: AA=0x30;break; /* '3.' */
        case 20: AA=0x19;break; /* '4.' */
        case 21: AA=0x12;break; /* '5.' */
        case 22: AA=0x02;break; /* '6.' */
        case 23: AA=0x78;break; /* '7.' */
        case 24: AA=0x00;break; /* '8.' */
        case 25: AA=0x10;break; /* '9.' */
        case 26: AA=0xc8;break; /* 'n' */
        case 27: AA=0xce;break; /* 'r' */
        case 28: AA=0x87;break; /* 't' */
        case 29: AA=0xc1;break; /* 'u' */
        case 30: AA=0x8c;break; /* 'p' */
        case '!': AA=0xbf;break; /* 破折号, 此处原误为 0xdf, 系 04 级王资凯同学指正 */
        case '_': AA=0xf7;break; /* 下划线 */
        case ':': AA=0xff;break; /* 消隐 */
    }
}

```

```

        default: AA=0xff;
    }
    return(AA);
}

// 函数名: 数码管显示
void display_led(unsigned char seg,unsigned char sel)
{
    unsigned char i;

    //先将 sel 数据送 74hc595
    PORTA &= ~(1<<PA7); // PA7=0; rclk=0
    for (i=0;i<8;i++)
    {
        if((sel & 0x80) == 0) PORTA &= ~(1<<PA5); //PA5=0, 最高位送 U2 SER 端
        else PORTA |= (1<<PA5); // PA5=1

        PORTA &= ~(1<<PA6); //PA6=0
        PORTA |= (1<<PA6); //PA6=1 srclk=1, 产生移位时钟信号
        sel <<= 1; //sel 左移一位
    }

    //再将 seg 数据送 74hc595
    for (i=0;i<8;i++)
    {
        if((seg & 0x80) == 0) PORTA &= ~(1<<PA5); //PA5=0, 最高位送 U2 SER 端
        else PORTA |= (1<<PA5); //PA5=1

        PORTA &= ~(1<<PA6); //PA6=0
        PORTA |= (1<<PA6); //PA6=1 srclk=1, 产生移位时钟信号
        seg <<= 1; //seg 左移一位
    }

    PORTA |= (1<<PA7); // PA7=1; rclk=1
    PORTA &= ~(1<<PA7); // PA7=0; rclk=0, 产生锁存输出信号
}

// 函数名: 端口初始化
void port_init(void)
{
    PORTA = 0x00; DDRA = 0xE0; //PA 口 PA7、PA6、PA5 为输出
    PORTB = 0x00; DDRB = 0xFF;
    PORTC = 0xf0; DDRC = 0x00; //PC 口 为输入, PC 口 PC7、PC6、PC5、PC4 接上拉电阻
    PORTD = 0x00; DDRD = 0x00;
}

// 定时器 1: 初始化-预分频 256
// WGM: Normal
// 期望频率: 200Hz
// 实际频率: 200.321Hz (0.2%)
void timer0_init(void)
{
    TCCR0 = 0x00; //stop
    TCNT0 = 0x64; //set count
    OCR0 = 0x9C; //set compare
    TCCR0 = 0x04; //start timer
}

//函数名: 定时器 1 初始化
void timer1_init(void)
{

```

```

//PD5 PD4 已分别与 OC1A OC1B 关联在一起了
DDRD |= (1<<PD4)|(1<<PD5); //OC1A、OC1B 设置输出
TCCR1B = 0x00;
OCR1A = 0x809;
OCR1B = 0x14E;
ICR1 = 0x2710;
TCCR1A = 0xb2; //11010010; 比较匹配时置位 OC1A,清零 OC1B; 在 TOP 时清零 OC1A,置位 OC1B.计数上
限 ICR1, 快速 PWM 模式 14
TCCR1B = 0x19; //00011001;时钟选择无分频 clkio/1
}

//函数名: 定时器 2 初始化
void timer2_init(void)
{
    TCCR2 = 0x02;
    TCNT2 = 0x00;
    OCR2 = 0xFF;
}

//函数名: 定时器 0 5ms 溢出中断处理函数
ISR(TIMER0_OVF_vect)
{
    TCNT0 = 0x64; //reload counter value
    // 1 秒钟软定时器计数
    if(++clock1s>=V_T1s)
    {
        clock1s_flag = 1; //当 1 秒到时, 溢出标志置 1
        clock1s = 0;
    }

    /** 检测到按键被按下 (0) 时, 相应的指示灯亮 (0)
    KEY1 = PINC&(1<<PC4);
    KEY2 = PINC&(1<<PC5);
    KEY3 = PINC&(1<<PC6);
    KEY4 = PINC&(1<<PC7);

    //----检测 SW1-----
    if (KEY1 == 0) led1 = 0; //PC4,C 口第 4 位
    else led1 = 1;

    //----检测 SW2-----
    if (KEY2==0) led2 = 0; //PC5,C 口第 5 位
    else led2 = 1;

    //----检测 SW3-----
    if (KEY3==0) led3 = 0; //PC6,C 口第 6 位
    else led3 = 1;

    //----检测 SW4-----
    if (KEY4==0) led4 = 0; //PC7,C 口第 7 位
    else led4 = 1;

    // 数码管扫描驱动指针值从 1 到 4 重复变换, 每 5ms 间隔对一个数码管进行驱动, 20ms 一个轮回
    if(++digi_scaner>=5) digi_scaner = 1;
    output_sel=led1*16+led2*32+led3*64+led4*128; //四个发光管送高四位

    switch (digi_scaner)
    {
        case 1: // 取第一个数码管显示数据
            output_sel += 1;
            output_8seg = NUMTOSEG7(digi[0]);

```

```

        break;
    case 2: // 取第二个数码管显示数据
        output_sel += 2;
        output_8seg = NUMTOSEG7(digi[1]);
        break;
    case 3: // 取第三个数码管显示数据
        output_sel += 4;
        output_8seg = NUMTOSEG7(digi[2]);
        break;
    case 4: // 取第四个数码管显示数据
        output_sel += 8;
        output_8seg = NUMTOSEG7(digi[3]);
        break;
}

//---对按键一进行消抖-----
cur_sample1 = KEY1; /* 将按键一的当前状态读取到 cur_sample1 */
if(pre_sample1 != cur_sample1) key1_count++;
else key1_count = 0; /* 如果按键的状态发生了改变，则开始计数，反之清零 */

if(key1_count >= v_keycount) /* 如果按键在改变的状态持续达到 v_keycount 的次数执行以下片断 */
{
    key1_count = 0; /* 给出按键被按下的 flag，计数器归零，默认状态置为当前状态 */
    if(cur_sample1 == 0) key1_flag = 1;
    pre_sample1 = cur_sample1;
}

/* 对按键二进行消抖，原理同按键一，见按键一的注释 */
cur_sample2 = KEY2;
if(pre_sample2 != cur_sample2) key2_count++;
else key2_count = 0;
if(key2_count >= v_keycount)
{
    key2_count = 0;
    if(cur_sample2 == 0) key2_flag = 1;
    pre_sample2 = cur_sample2;
}

/* 对按键三进行消抖，原理同按键一，见按键一的注释 */
cur_sample3 = KEY3;
if(pre_sample3 != cur_sample3) key3_count++;
else key3_count = 0;
if(key3_count >= v_keycount)
{
    key3_count = 0;
    if(cur_sample3 == 0) key3_flag = 1;
    pre_sample3 = cur_sample3;
}

/* 对按键四进行消抖，原理同按键一，见按键一的注释 */
cur_sample4 = KEY4;
if(pre_sample4 != cur_sample4) key4_count++;
else key4_count = 0;
if(key4_count >= v_keycount)
{
    key4_count = 0;
    if(cur_sample4 == 0) key4_flag = 1;
    pre_sample4 = cur_sample4;
}

/* 按键消抖程序结束 */
/* 处理 AD 转换的任务 */

```



```

    ADCSRA = 0xc6; /* 启动一次 AD 转换*/
    AD_flag=0;
    display_led(output_8seg,output_sel); //串转并输出
}

//函数名：定时器 2 中断处理函数
ISR(TIMER2_COMP_vect)
{
    if(state==0)
    {
        PWM_OUT=1;
        OCR2 = hlv_th;
        TCNT2 = 0X00;
        state=1;
    }
    else {
        PWM_OUT=0;
        OCR2 = lolv_th;
        TCNT2 = 0X00;
        state=0;
    }

    if(PWM_OUT ==0) PORTB = 0x00;
    else PORTB = 0xff;
}

//call this routine to initialize all peripherals
void init_devices(void)
{
    cli(); //disable all interrupts
    port_init();
    timer0_init();
    timer1_init();
    timer2_init();
    ADMUX = 0xD0;
    ADCSRA = 0xC6;
    MCUCR = 0x00;
    GICR = 0x00;
    TIMSK = 0x81; //timer interrupt sources
    sei(); //re-enable interrupts
    //all peripherals are now initialized
}

//-----主函数-----
int main()
{
    init_devices( );
    // 主循环，本例中，在 T0 中断服务程序未被执行的空余时间里，处理机在以下程序中不断循环
    while(1)
    {

        //按键服务程序
        if(key4_flag) //按 4 号键可以进入不同设置状态——设置状态标志: 0 为电压, 1 为占空比
        {
            if(setting_flag++>=2)
            {
                setting_flag=0;
            }
            refresh_flag=1;
            key4_flag=0;
        }
    }
}

```

闭——CLOSE

if(key3\_flag) //按键三：开环、闭环控制切换,同时在数码管上显示相应的标识，开——OPEN，

```
{
    if(!setting_flag)
    {
        CLC_flag=!CLC_flag;
        if(!CLC_flag)
        {
            digi[0] = 0;
            digi[1] = 30;
            digi[2] = 14;
            digi[3] = 26;
        }
        if(CLC_flag)
        {
            digi[0] = 12;
            digi[1] = 1;
            digi[2] = 5;
            digi[3] = 13;
        }
    }
    key3_flag = 0;
}

if(key1_flag) //在设置状态下，1号按键的作用是增加
{
    if(!setting_flag)
    {
        if(Volt<1000) Volt+=10;
        Prop=Duty_Set[(Volt-500)/10]; //计算占空比
    }
    refresh_flag=1;
    change_flag=1;
    key1_flag=0;
}

if(key2_flag) //在设置状态下，2号按键的作用是减少
{
    if(!setting_flag)
    {
        if(Volt>500) Volt-=10;
        Prop=Duty_Set[(Volt-500)/10]; //计算占空比
    }
    refresh_flag=1;
    change_flag=1;
    key2_flag=0;
}

if(CLC_flag) //闭环程序段
{
    if(!AD_flag)
    {
        AD_data[AD_ptr] = (unsigned long)ADCL + (unsigned long)ADCH * 256 ;
        if(AD_data[AD_ptr] > 511) AD_data[AD_ptr] -= 1024;
        AD_ptr++;
        if(AD_ptr == AD_SMPL_NUM) AD_ptr = 0;
        AD_pflag++;
        AD_flag = 1;
        DelayCounter++;
    }

    if(AD_pflag >= AD_SMPL_NUM) /*计算平均值*/
    {
```

```

        AD_pflag -= 1;
        AD_value = 0;
        for(i=0;i<AD_SMPL_NUM;i++)
        {
            AD_value += AD_data[i];
        }
        AD_value = AD_value/AD_SMPL_NUM-11;
    }

    SMPL_Volt = 740 - AD_value*970/1000;
    if(Volt>900) SMPL_Volt += 1;
    if(DelayCounter == CLC_DELAY)/*延时一定时间之后计算当前测量电压值*/
    {
        DelayCounter=0; /*重置延时计数器*/
        /*闭环占空比调整部分*/
        if((SMPL_Volt-Volt)>2) /*当电压差大于 0.02V 时进行调整*/
        {
            /*分段调整，根据差值的大小动态改变占空比，实现 1 秒之内的调整
            速度*/

            if((SMPL_Volt-Volt)>100) Prop-=100;
            else if((SMPL_Volt-Volt)>20) Prop-=30;
            else Prop-=2;

            change_flag=1;
            if(setting_flag) refresh_flag=1;
        }
        else if((Volt-SMPL_Volt)>2)
        {
            if((Volt-SMPL_Volt)>100) Prop+=100;
            else if((Volt-SMPL_Volt)>20) Prop+=30;
            else Prop+=2;
            change_flag=1;
            if(setting_flag) refresh_flag=1;
        }
    }
}

/*****
显示数据和重装值刷新程序
*****/
if(change_flag) /*定时器 1 重装值计算*/
{
    hlv_th = 300*Prop/10000;
    lolv_th = 300-hlv_th;
    OCR1A = Prop;
    change_flag=0;
}
if(refresh_flag) /*数码管显示刷新*/
{
    if(!setting_flag)
    { /*刷新: 0 为电压, 1 为占空比*/
        digi[0] = 29;
        digi[1] = (Volt/1000)?1:'';
        digi[2] = Volt/100%10+16;
        digi[3] = Volt/10%10;
    }
    else if(setting_flag==1){
        digi[0] = 30;
        digi[1] = Prop/1000%10;
        digi[2] = Prop/100%10+16;
        digi[3] = Prop/10%10;
    }
}

```

```

    }
    else if(setting_flag==2){
        digi[0] = '';
        digi[1] = (SMPL_Volt/1000)?1:'';
        digi[2] = SMPL_Volt/100%10+16;
        digi[3] = SMPL_Volt/10%10;
    }
    refresh_flag=0;
}
}
}

```