

系统实验报告

专 业： 通 信 工 程
学 生 姓 名： 曹 莲
学 号： 1342401011
指 导 教 师： 陈 蓉

一、设计任务

1、根据 PCB 板及电路工作原理，绘制电路原理图，理解个芯片功能。

2、正确焊接电路板

3、编写程序，实现以下设计任务：

➤ 时钟（计时器）

- a) 状态 1：一只数码管 0-9 计数，一只按键控制开始、结束；
- b) 状态 2：二只数码管 0-59 计数，一只按键控制开始、结束；
- c) 状态 3：三只数码管低二位 0-59 计数后进位给高位 0-9 计数，一只按键控制开始、结束；
- d) 状态 4：四只数码管低二位 0-59 计数后给高二位 0-59 计数（即分秒计时），一只按键控制开始、结束；

➤ 交通灯

- e) 状态 1：仅亮灯，数码管不工作；

按下键，红黄绿三色灯交替闪亮：红（30 秒）→绿（25 秒）→黄（闪烁 5 秒）→红。

- f) 状态 2：灯和数码管相结合，模拟十字路口的交通灯。

在以上功能的基础上数码管倒计时显示时间，如下图所示。

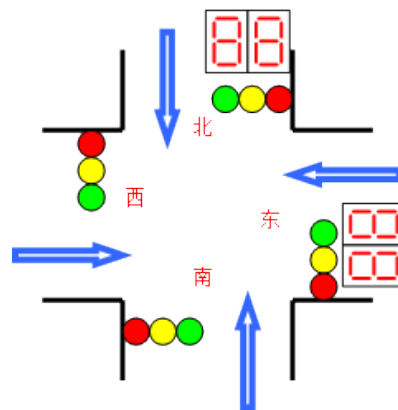


图 1. 十字路口交通灯示意图

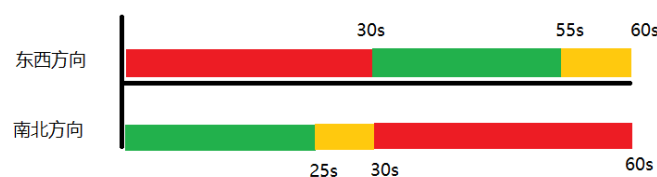


图 2. 交通灯时序图

4、烧录，调试程序，直到通过。

二、 设计方案

(A) 稳压源部分将输入的直流变压器 9V 转换为稳定的 5V 直流电压，不用接整流桥，将电路板上相应引脚用导线连通，具体接法如下图：

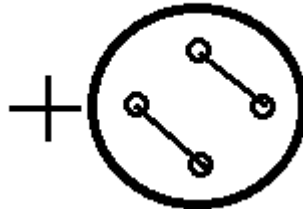


图 3. 电路板整流桥接法

(B) 作为单片机的输入部分 (P1.0~P1.3) 起控制作用；

(C) 作为单片机的输出部分，P0.0~P0.3 经由 7 段译码器 (74LS47) 控制数码管的显示结果为 0~9，P0.4~P0.7 为数码管的选通控制口，提供数码管工作电源 (公共端)，P2.7 控制数码管的小数点是否点亮；

(D) 发光二极管：输出部分，由 P2.0~P2.5 口控制，高电平灯亮；

(E) 单片机 AT89S51：低功耗，高性能的 CMOS 8 位单片机。

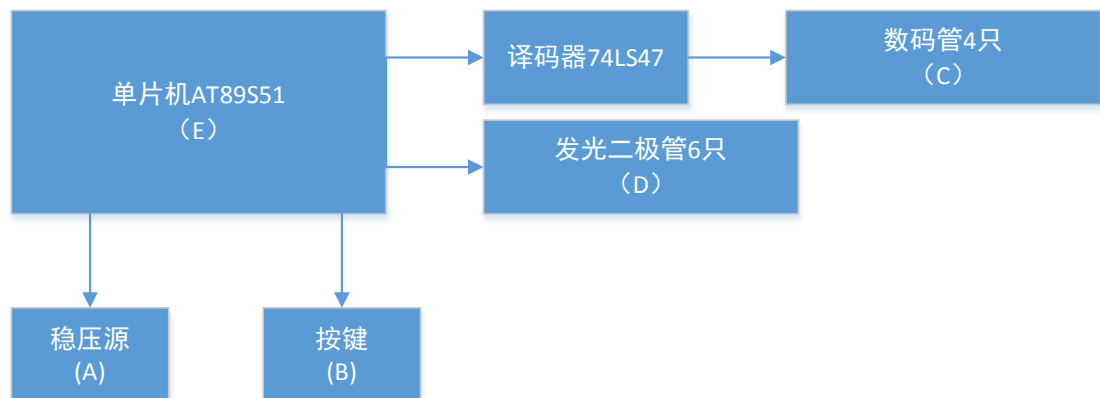


图 4. 系统原理图

三、 设计器件

1、 AT89S51

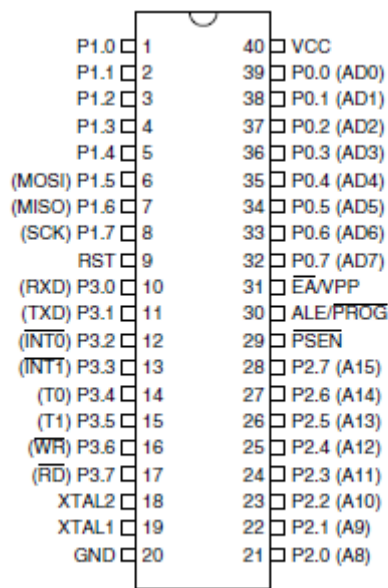


图 5. AT89S51 引脚图

VCC: ATAT89S51 电源正端输入，接+5V。

GND: 电源地端。

XTAL1: 单芯片系统时钟的反向放大器输入端。

XTAL2: 系统时钟的反向放大器输出端，一般在设计上只要在 XTAL1 和 XTAL2 上接上一只石英振荡晶体系统就可以动作了，此外可以在两个引脚与地之间加入一个 20PF 的小电容，可以使系统更稳定，避免噪声干扰而死机。

RST: AT89S51 的重置引脚，高电平动作，当要对晶片重置时，只要对此引脚电平提升至高电平并保持两个机器周期以上的时间，AT89S51 便能完成系统重置的各项动作，使得内部特殊功能寄存器之内容均被设成已知状态，并且至地址 0000H 处开始读入程序代码而执行程序。

EA/VPP: “EA”为英文“External Access”的缩写，表示存取外部程序代码之意，低电平动作，也就是说当此引脚接低电平后，系统会取用外部的程序代码（存于外部 EPROM 中）来执行程序。

ALE/PROG: ALE 是英文“Address Latch Enable”的缩写，表示地址锁存器启用信号。AT89S51 可以利用这个引脚来触发外部的 8 位锁存器（如 74LS373），将端口 0 的地址总线（A0~A7）锁进锁存器中，因为 ATAT89S51 是以多工的方式送出地址及数据。平时在程序执行时 ALE 引脚的输出频率约是系统工作频率的 1/6，因此可以用来驱动其他周边晶片的时基输入。此外在烧录 8751 程序代码时，此引脚会被当成程序规划的特殊功能来使用。

PSEN: 此为“Program Store Enable”的缩写，其意为程序储存启用，当 8051 被设成为读取外部程序代码工作模式时（EA=0），会送出此信号以便取得程序代码，通常这支脚是接到 EPROM 的 OE 脚。AT89S51 可以利用 PSEN 及 RD 引脚分别启用存在外部的 RAM 与 EPROM，使得数据存储器与程序存储器可以合并在一起而共用 64K 的定址范围。

PORT0 (P0.0~P0.7) : 端口 0 是一个 8 位宽的开路电极 (Open Drain) 双向输出输入端口，共有 8 个位，P0.0 表示位 0，P0.1 表示位 1，依此类推。其他三个 I/O 端口 (P1、P2、P3) 则不具有此电路组态，而是内部有一提升电路，P0 在当作 I/O 用时可以推动 8 个 LS 的 TTL 负载。如果当 EA 引脚为低电平时（即取用外部程序代码或数据存储器），P0 就以多工方式提供地址总线 (A0~A7) 及数据总线 (D0~D7)。设计者必须外加一个锁存器将端口 0 送出的地址锁住成为 A0~A7，再配合端口 2 所送出的 A8~A15 合成一组完整的 16 位地址总线，而定位地址到 64K 的外部存储器空间。

PORT2 (P2.0~P2.7) : 端口 2 是具有内部提升电路的双向 I/O 端口，每一个引脚可以推动 4 个 LS 的 TTL 负载，若将端口 2 的输出设为高电平时，此端口便能当成输入端口来使用。P2 除了当作一般 I/O 端口使用外，若是在 AT89S51 扩充外接程序存储器或数据存储器时，也提供地址总线的高字节 A8~A15，这个时候 P2 便不能当作 I/O 来使用了。

PORT1 (P1.0~P1.7) : 端口 1 也是具有内部提升电路的双向 I/O 端口，其输出缓冲器可以推动 4 个 LS TTL 负载，同样地，若将端口 1 的输出设为高电平，便是由此端口来输入数据。如果是使用 8052 或是 8032 的话，P1.0 又当作定时器 2 的外部脉冲输入脚，而 P1.1 可以有 T2EX 功能，可以做外部中断输入的触发引脚。

PORT3 (P3.0~P3.7) : 端口 3 也具有内部提升电路的双向 I/O 端口，其输出缓冲器可以推动 4 个 TTL 负载，同时还多工具有其他的额外特殊功能，包括串行通信、外部中断控制、计时计数控制及外部数据存储器内容的读取或写入控制等功能。其引脚分配如下：

P3.0: RXD, 串行通信输入。

P3.1: TXD, 串行通信输出。 P3.2: INT0, 外部中断 0 输入。

P3.3: INT1, 外部中断 1 输入。 P3.4: T0, 计时计数器 0 输入。

P3.5: T1, 计时计数器 1 输入。 P3.6: WR: 外部数据存储器的写入信号。

P3.7: RD, 外部数据存储器的读取信号。

2、74LS47

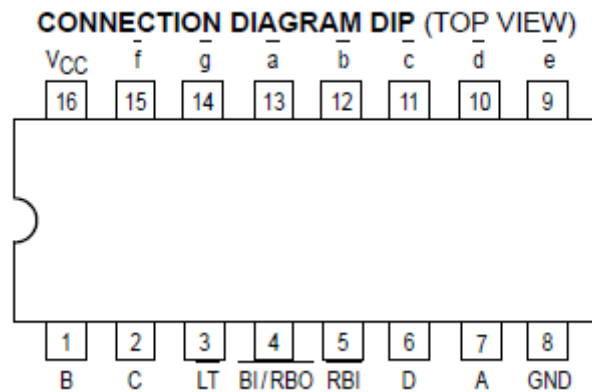


图 6. 74LS47 引脚图

74LS47 是 BCD-7 段数码管译码器/驱动器，74LS47 的功能用于将 BCD 码转化成数码块中的数字, 通过它解码，可以直接把数字转换为数码管的显示数字。

ABCD 接输入 BCD 码，74LS47 输出 a, b, c, d, e, f, g 为低电平有效。

其他引脚功能：

(1)**LT**：试灯输入，是为了检查数码管各段是否能正常发光而设置的。当 LT=0 时，无论输入 A3，A2，A1，A0 为何种状态，译码器输出均为低电平，也就是七段将全亮，若驱动的数码管正常，是显示 8。

(2)**BI**：灭灯输入，是为控制多位数码显示的灭灯所设置的。当 BI=0 时，不论 LT 和输入 A3，A2，A1，A0 为何种状态，译码器输出均为高电平，使共阳极数码管熄灭。

(3)**RBI**：灭零输入，它是为使不希望显示的 0 熄灭而设定的。当对每一位 A3=A2=A1=A0=0 时，本应显示 0，但是在 RBI=0 作用下，使译码器输出全为高电平。其结果和加入灭灯信号的结果一样，将 0 熄灭。

(4)**RBO**：灭零输出，它和灭灯输入 BI 共用一端，两者配合使用，可以实现多位数码显示的灭零控制。

3、LM7805

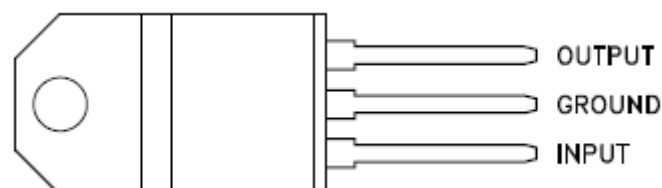


图 7. LM7805 引脚图

INPUT：电压输入引脚，输入电压最大值为 35V

GROUND：接地

OUTPUT：5V 稳压输出脚

- 4、晶振 11.0592MHz
- 5、三极管 9013 共 10 个
- 6、四角扁平按键 5 个
- 7、共阳极数码管 4 个
- 8、发光二极管：红灯 3 个，黄灯 2 个，绿灯 2 个
- 9、排阻 10K 共三个，150 欧姆电阻 8 个，1K 电阻 8 个
- 10、0.1uF 电容一个，20pF 电容 2 个，22uF、470uF 电解电容各一个

四、 设计原理图

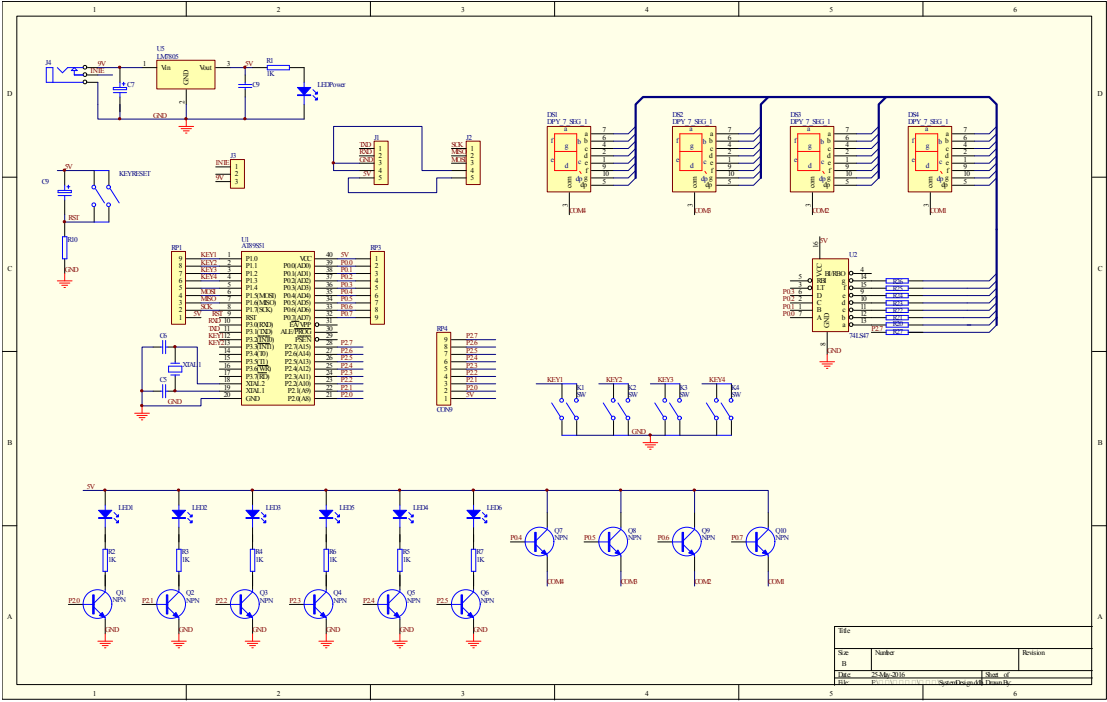


图 8. 设计原理图

五、 软件设计流程图

- 1、分秒计时器流程图

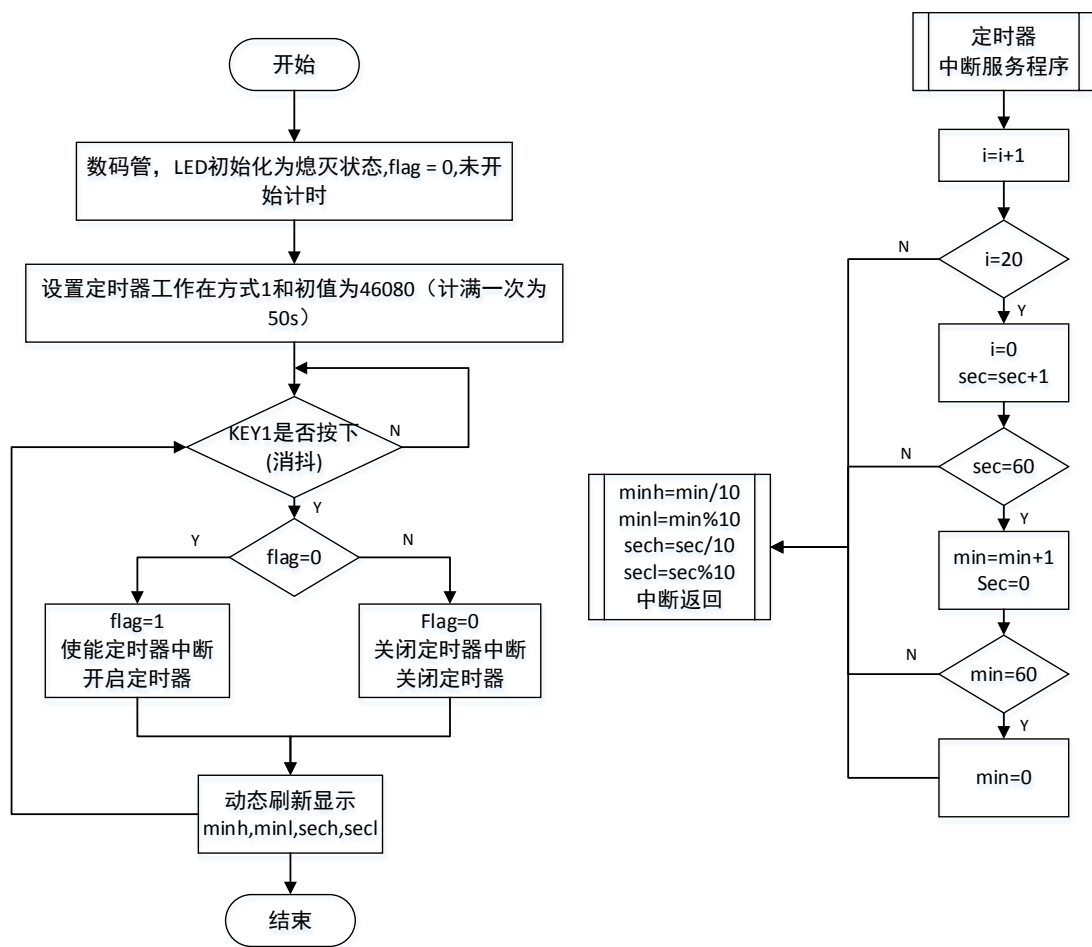


图 9. 分秒计时器流程图

2、交通灯流程图

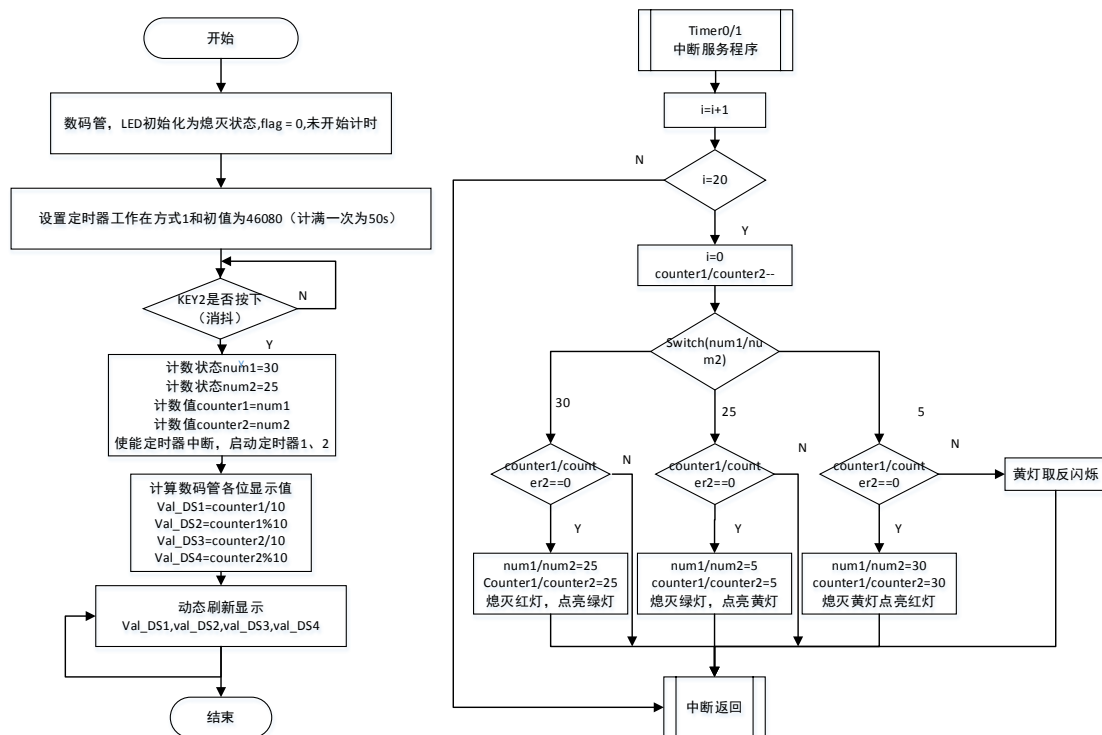


图 10. 交通灯流程图

六、 源程序代码

(见附录)

七、 实现功能

- 1、 各状态功能通过单独下载程序，均可顺利实现
- 2、 总程序可以实现按下 KEY4 开启交通灯，按下 KEY2 关闭交通灯，按下 KEY3 开启分秒计时器，按下 KEY1 关闭分秒计时器。

八、 调试问题

- 1、 问题：按键扫描时，按键不稳定，有时有用有时失灵
解决方法：修改按键扫描防抖时间，20ms 时，按键检测效果较好。防抖等待时间过短，无法起到防抖效果，可能会过于灵敏；防抖时间过长时，可能出现检测不到按键按下的情况，按键过于迟钝。
- 2、 问题：数码管小数点持续点亮
解决方法：小数点一直亮说明 dp 引脚持续输入低电平，在初始化时，应使 P2.7 输出高电平，且在控制 LED 灯时也不应改变 P2.7 位。
- 3、 问题：数码管显示会闪动，视觉效果较差
解决方法：数码管动态显示频率太低，应加大动态刷新频率，减小各数码管显示的时间间隔。
- 4、 问题：按键按下没有反应
解决方法：在读取 51 单片机 I/O 口电平时，应先向相应的 I/O 口写高电平，否则无法正确读取 I/O 口结果。
- 5、 问题：在烧录程序时，偶尔会出现 SUPERPRO 软件警告引脚接触错误或损坏
解决方法：开始担心是否是芯片烧坏了，后来发现是因为芯片与烧录器接触不良，没有压紧，才导致管脚无法读取。

九、 心得小结

这次系统实验历时两周的时间，在这实验过程里我巩固了从焊接电路板到调试的专业知识，逻辑思维和动手能力都得到了很大的提高。

实验开始第一件工作是理解整个电路的原理，按原理及 PCB 绘制原理图，我是先用 Protel 99 SE 软件绘制原理图，然后再誊写到 A4 纸上，虽然麻烦了

一点，但是软件绘制也带来了许多方便，有助于我更好的理解整个系统，同时也锻炼了软件绘图的能力，有助于今后的电子设计学习。

在焊接好电路板后，我首先测试了整块电路板有无焊接错误的地方，用万用表测试了所有的 VCC 和地，以及一些重要的引脚，上电后测量 LM7805 的输出脚是否为 5V，确保无误后，开始编写程序。

本次系统实验主要有两个题目，分秒计时器和交通灯，总体来说，分秒计时器较为简单，重点在于掌握数码管动态刷新的原理和秒和分的进位。而交通灯这一题之前觉得也不困难，但是实际写程序时，才发现需要考虑很多方面的因素。首先，交通灯的红黄绿三个灯色显示的时间各不相同，且黄灯需要闪烁，所以需要针对不同灯色做不同计时处理，其次，两个路口灯的状态是不相同的，而且时间上存在重叠，所以设计选用两个定时器中断，以对两个路口的红绿灯分别控制。为了更清晰地理解交通灯的工作原理和方式，我绘制了交通灯的时序图，这对我编写程序有极大的帮助。经过循序渐进的编写程序，烧录，观察结果，修改，最终顺利完成了所有程序，实现了要求的功能。

通过这次系统实验，对以前学过的知识进行了巩固，加深了理解，提高了应用的能力，而且提高了我们的发现、分析、解决问题的能力，同时提高了我对专业的认识及兴趣，对于我们工科生来说，对以后就业很有帮助。

最后，尤其要感谢陈蓉和谢门喜老师的指导和帮助，我们才得以顺利完成这次系统试验。