

图书分类号: _____
密 级: _____

徐州工程学院

毕业设计(论文)

基于单片机的静电防护在线监控系统设计与实现
ON LINE MONITORING SYSTEM DESGIN
AND IMPLEMENTATION OF ELECTROSTATIC
PROTECTION BASED ON SCM

学生姓名	徐 容
学院名称	信电工程学院
学 号	20120503201
班 级	12 电 气 2
专业名称	电气工程及其自动化
指导教师	潘晓博

2016 年 6 月 3 日

徐州工程学院学位论文原创性声明

本人郑重声明： 所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用或参考的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标注。

本人完全意识到本声明的法律结果由本人承担。

论文作者签名：_____ 日期：_____年____月____日

徐州工程学院学位论文版权协议书

本人完全了解徐州工程学院关于收集、保存、使用学位论文的规定，即：本校学生在学习期间所完成的学位论文的知识产权归徐州工程学院所拥有。徐州工程学院有权保留并相国家有关部门或机构送交学位论文的纸本复印件和电子文档拷贝，允许论文被查阅和借阅。徐州工程学院可以公布学位论文的全部或部分内容，可以将本学位论文的全部或部分内容提交至各类数据库进行发布和检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

论文作者签名：_____ 导师签名：_____

日期：_____年____月____日 日期：_____年____月____日

摘要

摩擦起电和人体静电是电子工业中的两大危害，常常造成电子电器产品运行不稳定，甚至损坏。我们的身上和周围就带有很高的静电电压，对于一些敏感仪器来讲，这个电压可能会是致命的危害。所以我们要能及时的监测到静电的产生并及时的消除静电，以减少其对精密设备的危害。该系统利用单片机 AT89S52 实现对静电手环和用电设备的在线监控，能够及时知道用电设备防护静电的情况，进行可视化监控管理，通过蜂鸣器和 LED 灯的闪烁来提醒工作人员，防静电更加安全可靠。

关键词 单片机；AT89S52；静电释放

Abstract

Friction static electricity and the human body is in the electronic industry, the two major hazards, often resulting in electrical and electronic products, unstable operation, even damage. Our body and surrounding it with a high electrostatic voltage, for some sensitive instruments, the voltage may would be fatal. So we should timely monitoring to the electrostatic and timely elimination of static electricity, in order to reduce the harm of precision equipment. The system use AT89S52 SCM implementation of electrostatic bracelet and electrical equipment on-line monitoring can timely know electric equipment for the protection of the static, visual monitoring, through the buzzer and LED flashing lights to To remind the staff, anti-static more safe and reliable.

Keywords SCM AT89S52 Electrostatic discharge

目 录

1 绪论.....	1
1.1 选题背景.....	1
1.2 设计要求及要解决的主要问题.....	1
1.3 发展趋势.....	1
2 系统总体设计方案.....	3
2.1 系统设计主要内容.....	3
2.2 系统设计要求.....	3
3 设计方案.....	5
3.1 系统设计原理.....	5
3.2 器件选择.....	5
3.2.1 单片机的选择.....	5
3.2.2 串行接口的选择.....	5
3.2.3 蜂鸣器的选择.....	6
3.2.4 电源适配器的选择.....	6
3.2.5 监控电路芯片的选择.....	6
3.3 硬件介绍.....	7
3.3.1 电容器.....	7
3.3.2 瞬态抑制二极管.....	8
3.3.3 拨码开关.....	8
4 电路设计.....	10
4.1 电源电路设计.....	10
4.2 被检查设备连入系统电路设计.....	10
4.3 静电手环连入系统电路设计.....	13
4.3.1 AT89S52 单片机介绍.....	15
4.3.2 AT89S52 振荡电路.....	16
4.3.3 AT89S52 监控电路.....	17
4.3.4 AT89S52 与蜂鸣器连接电路.....	18
4.3.5 AT89S52 与 LED 灯连接电路.....	19
5 软件设计.....	21
5.1 单片机的主要片脚接口.....	21
5.2 单片机程序控制流程图.....	22
5.2.1 被检验设备程序流程图.....	22

5.2.2 静电手环监测流程图.....	23
5.3 系统主程序.....	24
6 硬件测试.....	26
结论.....	29
致谢.....	30
参考文献.....	31

1 绪论

1.1 选题背景

摩擦电能在任何两种物体间产生, 和我们的生活有着密不可分的联系。在气候干燥的时候如果你穿着塑料底鞋字干净的地毯或地板上长时间行走后, 伸手去抓金属门柄时会有一个小火花跳到你的手上。人们使用的越来越多的塑料制品, 化纤衣物也都是摩擦电的栖息地。但这些还只是生活上的小事, 在工业生产中静电的危害就更加明显了。在煤矿矿井中摩擦电释放出的火花会引起瓦斯爆炸, 给人们的生命财产造成了巨大的损失。在手术台上静电火花引起麻醉剂乙醚爆炸使无数的病人死在手术台上, 狂风卷起的沙石网网携带大量的摩擦电荷, 中断无线电通信, 有时还会引起铁路、航空等自动信号系统的信号错误, 造成严重事故。石油工业中静电的危害也是非常大的, 石油产品及其蒸汽是危险的易燃易爆品, 300V 以上的火花放电就会造成含有汽油或汽油蒸汽的空气燃烧爆炸。而在石油生产和石油储运中几乎处处都有静电。在电子行业中, 由于静电放电产生的击穿会使集成电路、电容等被击穿破坏而不能工作。特别是在一些精密仪器中, 静电的破坏是无穷的, 所以我们要知道如何检测和防护静电。

静电灾害的防护从本质上来说就是使得电荷不产生和电荷不蓄积的问题, 但是这在实际中是几乎不可能实现的。最基本的方法就是减少静电电荷的产生。一般来说静电灾害中电荷产生的原因多半是摩擦起电, 这种摩擦起电的电荷与作为电荷蓄积对象的物质的电导率有一定的关系。若该物质的电导率提高则电荷的产生就会减少, 预防方法就是使物质性质发生变化。其中包括对表面进行处理, 改变带电体周围导电环境, 采用接地技术, 尽可能减少接触面积或降低速度等。采用接地技术消除静电时需注意接地时必须真正可靠的接地, 且静电接地不能直接接地, 必须通过一定的阻值的电阻后才能接入大地。

1.2 设计要求

快速、稳定、直观、可控的监测静电的产生并且对工作人员的不正当操作或其他原因产生的静电能够快速的做出判断并且提醒工作人员。此系统将静电这个微观世界的东西实现了宏观的控制。从多方面进行实时连续的监控。做到有记录, 让生产各环节有根据可询。

解决的主要问题是监控设备地垫、静电手环; 接地不良可报错; 按照不同类型实现报错。

1.3 发展趋势

人类在静电危害面前并不是束手无策的, 用良导体把容易产生静电的部位接地是一种简单易行而又安全可靠的方法。这种方法在石油储运中应用的最多, 输油管每隔 100 米接地一次, 油罐轮胎是绝缘的, 油罐汽车后面拖一根拖地的金属链条。湿度调节也是控制静电的另外一种方法, 在其他条件允许的情况下适当的增加空气湿度。在集成电路的制造、检验、组装工作中, 为了使静电防护工作精益求精, 人们还采用了在地板上、工作台上铺

设铝箔、铝板、导电橡胶、导电薄膜；工作人员佩戴静电手环、戴导电橡皮手套等许多措施，随着静电防护研究的不断发展，静电危害是一定可以监测并克服的。

2 系统总体设计内容

2.1 系统设计主要内容

本系统采用电源、稳压管、单片机 AT89S52、通过运算放大器 LM324N、监控电路 MAX813L、蜂鸣器报警装置、静电手环以及 LED 灯完成系统硬件设计。电源输出的 9V 电压经过稳压管变成稳定的 5V 电压供给整个电路。为了防止 AT89S52 单片机程序跑飞或死机,提高单片机的可靠性,给它加上一个监控电路,及时保存数据或重新开始运行。通过静电手环和被检查设备发出外部输入信号,经过运算放大器 LM324N,输入到作为主机的 AT89S52 单片机,通过软件编程的程序,做出判断,静电手环和被检查设备是否能够正常运行,结果再经 LED 灯和蜂鸣器作为输出显示。当被检查正常及作业人员静电手环正确佩戴时,系统亮绿灯。当被检查设备不正常或者作业人员没有正确佩戴静电手环时,LED 亮红灯且蜂鸣器发出警报。我们通过单片机比较被检查设备电阻与静电输出电阻比较,来判断被检查设备是否受到静电作用。根据 ESD 国际标准对防护静电的电阻的范围要求和被检查的设备,我们选择 500 欧姆为被检查设备的输出电阻上限值,而静电手环是通过皮肤导电,人体的等效电阻为 1000 到 5000 欧,静电手环保护电阻为 10^6 欧,与人体并联后的等效电阻约为 1000 到 5000 欧姆,为了确保人身安全,我们选择最大值 5000 欧姆为的是否有静电的电阻上限值。这两个电阻比较值只是一个估计值,我们通过可调电阻实现,其大小以后可以根据被检查的不同设备进行更改。系统框图见图 2-1。

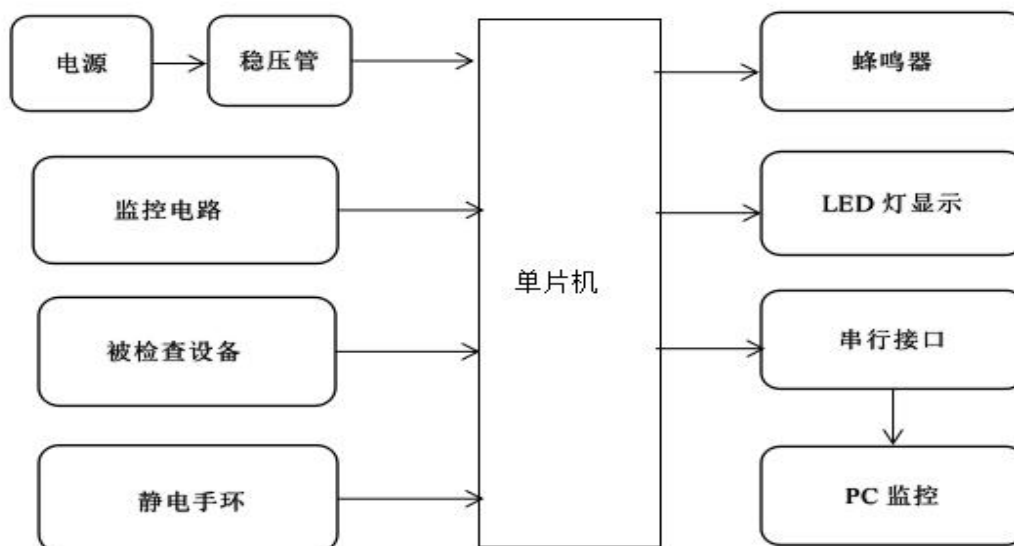


图 2-1 系统框图

2.2 系统设计要求

基于单片机的静电防护在线监控的设计要求如下:

- 1.当工作人员正确佩戴静电手环时,与静电手环对应的 LED 绿灯亮,蜂鸣器不响。
- 2.当工作人员未正确配带静电手环导致手环与人体间的电阻值超上限值 5000 欧姆时,与静电手环对应的 LED 红灯亮,同时蜂鸣器响。

- 3.当被检查设备正常时，与被检查设备对应的 LED 绿灯亮，蜂鸣器不响。
- 4.当被检查设备电阻超过设定上限值 500 欧姆时，与被检查设备对应的 LED 红灯亮，同时蜂鸣器响。
- 5.留一组黄色 LED 灯备用，当哪个灯泡出现故障时更改一下 LED 灯就好，还能达到预期监测的目的。

3 设计方案

3.1 系统设计原理

根据电工原理,电压源的输出阻抗与外部电路的负载输入阻抗相等时,外部负载获得最大的功率。

设电源电压为 E , 内阻为 r , 外部电阻为 R 。则外部电阻的功率见式(3.1)。

$$P = \left[\frac{E}{R+r} \right]^2 \times R \quad \text{式(3.1)}$$

当 $R=r$ 时, P 取最大值, 即外部负载获得最大功率。

静电的输出阻抗很高, 所以当外部电子元件的输入阻抗高时易受到静电作用。高到与静电输出阻抗相等时受到静电作用达到最大。我们通过单片机比较被检查设备电阻与静电输出电阻比较, 来判断被检查设备是否受到静电作用。

3.2 器件选择

3.2.1 单片机的选择

我们常用的单片机有 51 系列和 52 系列, 而 52 系列中以 AT89S52 单片机最具有代表性, AT89S52 与 AT89S51 相比片内程序存储器由 4 个 KB 增加到 8KB, 内存上大大增加了可写入程序, AT89S52 单片机还比 AT89S51 的片内 RAM 由 128 个字节增加到 256 个字节; 同时 AT89S52 单片机的片内定时器也由 AT89S51 的 2 个定时器增加到 3 个, 多增加了一个 T2 定时器。从而 AT89S52 的性能更加完善, 应用更加灵活, 而且价格和 51 相差不多, 具有较高的性价比。所以这里我们选择 AT89S52 作为我们该系统的核心。

3.2.2 串行接口的选择

经过发展串行通信接口标准目前已经有几种。RS-232、RS-485 与 RS-422 都是串行数据接口标准, RS-232 最早发布作为工业标准, 来保证兼容性。而 RS-422 是在 RS-232 基础上进行了改进, 克服了速率低、通讯距离短等的缺点。RS-422 传输速率达到 10Mb/s, 传输距离达到 4000 英尺, 且一条总线上最多允许连接 10 个接收器。所以我们选择 RS-422 作为我们的串行接口。连接器引脚见图 3-1。

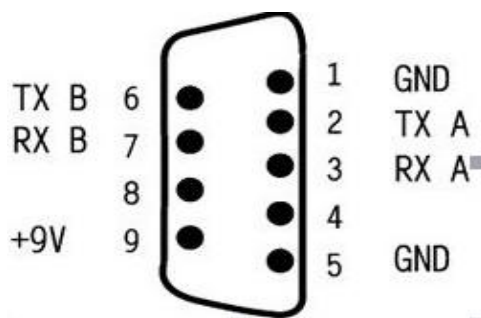


图 3-1 RS-422 连接引脚图

3.2.3 蜂鸣器的选择

蜂鸣器作为一种电子讯响器，广泛应用在电子产品中做发声器件。

蜂鸣器按结构不同可分为压电式蜂鸣器和电磁式蜂鸣器。压电式蜂鸣器的工作原理是过压电陶瓷的压电效应来带动金属片的震动，从而发出响声。电磁式蜂鸣器的工作原理是利用电磁原理，在通电的情况下将金属振动膜吸下，不通电时金属振动膜回弹。振荡器产生的电流通过电磁线圈从而产生磁场，振动膜在电磁线圈和磁铁的共同作用下周期性的振动发声。电磁式蜂鸣器用 1.5V 的电源电压就能发出 85 分贝以上的音压，而压电式蜂鸣器需要较高的电压才能产生足够的额音压，一般为 9V 以上。而我们的电源电压一般不到 9V，所以这里选择电磁式蜂鸣器。

蜂鸣器又可分为有源蜂鸣器和无源蜂鸣器。有源蜂鸣器在额定直流电源下就可以连续发声，而无源蜂鸣器需要外接音频输出电路，其内部没有驱动电路。为了简化电路所以这里我们选择有源蜂鸣器。综上所述，我们选择的蜂鸣器为有源电磁式蜂鸣器，蜂鸣器的型号为 TMB12A05。实物见图 3-2。



图 3-2 蜂鸣器实物图

3.2.4 电源适配器的选择

该系统采用的核心芯片为 AT89S52 单片机，其供电电压为 5V，我们选择稍微大于 5V 的电源适配器，将常用的 220V 交流电转换成直流电。最终选择为 9V 的电源适配器。

3.2.5 稳压管的选择

稳压管工作在反相击穿区，当反相电流有一个比较大的变化量时稳压管两端电压的变化量却很小，具有稳压的作用，因而广泛应用于稳压电源电路之中。根据单片机 AT89S52 的供电电压为 5V，我们选用最大输入电压为 35V，输出电压为 4.75~5.25V 的 L7805CV 稳压管。L7805CV 组成的稳压电源所需的其他外围元件很少，电路内部还有过流、过热的保护电路，使用起来可靠、方便，而且价格低廉。

3.2.6 监控电路芯片的选择

单片机以其体积小、价格低廉等独特优点，在各方面获得了广泛的应用。与此同时，

人们对单片机应用系统的可靠性的要求也越来越高。影响单片机可靠性的因素是多方面的，如系统的电子元器件本身的可靠性、系统各部分之间的相互耦合等。所以为了提高单片机的可靠性，我们给他外接一个监控电路。为了简化电路设计我们直接选择一个芯片来达到提高单片机可靠性的目的。常用的监控电路芯片为 MAX813L。可广泛应用于计算机、自动化设备、智能设备、控制器等的监控电路。

3.3 硬件介绍

3.3.1 电容器

常见的电容器可分为电解电容和瓷片电容。电解电容阳极为金属箔，阴极为导电材料和电解质，电介质是正极紧贴金属的氧化膜。点解电容分正负极。瓷片电容是在陶瓷表面涂一层金属薄膜，经过高温后作为电极而成的电容。通常作用在高稳定振荡回路中，常用来消除高频干扰。

电解电容的特性：

- 1.单位体积的电容量比较大，一般为 $1\text{--}33000\mu\text{F}$ 。比其他类型的电容大到几十倍甚至几百倍。
- 2.额定容量比较大，很容易就达到几万伏甚至几伏。
- 3.价格相对较低，组成的材料都是相对比较常见的，也可以大规模生产。
- 4.容量误差较大，耐高温性较差。

电解电容的接入方式：

电解电容器通常在电源电路、低频或中频电路中一般作用于电源滤波、信号耦合、时间常数设定和隔直流等。一般不用于交流电源电路，在直流电源电路中作用为滤波电容时，阳极应与电源电压的正极相连，阴极与电源电压的负极相连，不能反接，否则会损坏电容。一般电解电容的侧面有“-”是负极、“+”是正极，如果没有标明正负极，则长脚为正极，短脚为负极。电解电容实物见图 3-3。



图 3-3 电解电容实物图

瓷片电容的特性：

高压瓷片电容是以陶瓷材料为介质的圆板电容器。高压瓷片电容具有稳定性好，绝缘性好，耐高压的优点，但同时也有易碎且容量比较小，故障率也比较高的缺点。因此适用于高压旁路和耦合电路中。瓷片电容实物见图 3-4。



图 3-4 瓷片电容的实物图

瓷片电容的电容表示法:

三位数字的电容表示法也叫电容量的数码表示法,三位数字中的前两位为标称容量的有效数字,第三位数字为前两位的有效数字,即有效数字后满的零的个数。单位都为 pF。例如:104 表示标称电容容量为 100000pF。

也有一种特殊情况,当第三个数字为 9 时表示是用有效数字乘上十的负一次方来表示电容容量的大小。例如:119 表示标称电容容量为 1.1pF。

3.3.2 瞬态抑制二极管

瞬态抑制二极管 TVS 的电路符号与普通的二极管一样,外形也与普通的二极管差不多,但是当 TVS 管两端经受瞬间的高能量冲击时,能以极高的速度使其电阻突然降低,同时吸收一个大电流,将其两端的电压钳固在一个预定的数值上,从而保证后面的元件免收瞬态高能量的冲击而损坏。实物见图 3-5。



图 3-5 瞬态抑制二极管实物图

3.3.3 拨码开关

拨码开关也叫拨动开关,是一款用来操作控制的地址开关,采用的原理是 0/1 的二进制编码。拨码开关的实质也就是一款能用手拨动的微型开关,所以也通常称为指拨开关。根据位数来说正常是都是 1-10 位、12 位等 11 个位数。每一个键对应的背面上下各有两个引脚,拨至 ON 一侧,这下面两个引脚接通;反之则断开。这四个键是独立的,相互没有关联。此类元件多用于二进制编码。广泛使用于数据处理、通信、遥控和防盗自动警铃系统。拨码开关的实物见图 3-13。本系统中我们利用拨码开关来对地址进行编码,使每一个相同的电路板都有唯一的编码,也可以用来选择连入电路的被检查设备。拨码开关拨到下

面对应的单片机的引脚置位高电平，选通该地址位，才能接入单片机进行电阻值的比较。



图 3-13 拨码开关实物图

4 电路设计

4.1 电源电路设计

根据单片机 AT89S52 的正常供电电压是 5V，我们选择了对应的电源适配器和稳压管 L7805CV，三脚稳压管中间的管脚接地。电源通过整流二极管的整流作用和电解电容的滤波作用，经过抗干扰电路和 2 个稳压管将 9V 电源电压转化成适合该系统使用的 5V 供电电压。将一路电压分成两路稳定的电源输入到电路中。电路图见图 4-1。

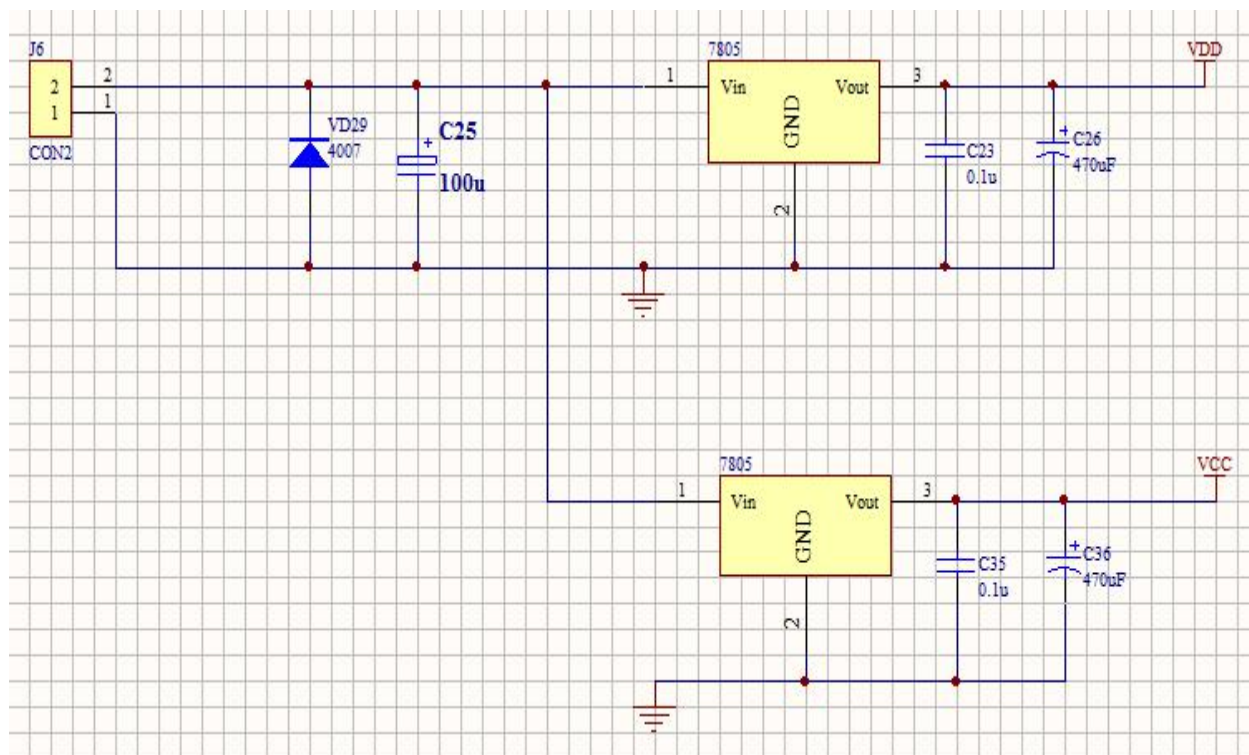


图 4-1 电源电路

4.2 被检查设备连入系统电路设计

针对被检查设备不确定的问题，我们不能确定选择什么样的接口使设备连入电路中，安装连线器可以通过将设备的正负极电线插入卡口中使电流流通，达到监测不同设备的静电防护问题。连线器实物图见图 4-2。

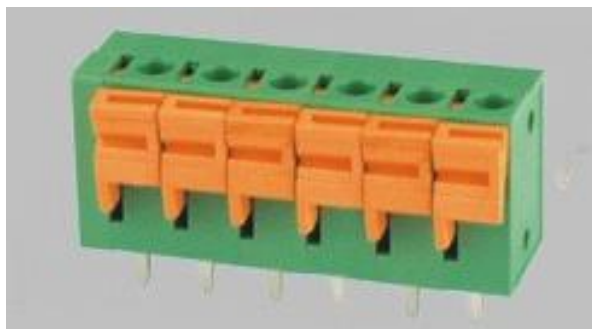


图 4-2 连线器实物图

按可以同时检查两个设备的标准,我们选择的是三个卡口的连线器。中间的卡口用来接负极,两边的卡口用来接两个被检查设备的正极。连线器通过常用的运算放大器 LM324N 将输入信号传给单片机, LM324 引脚图见图 4-3。

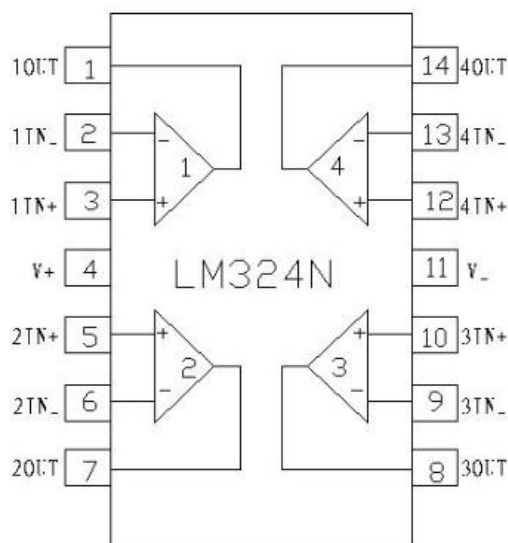


图 4-3 LM324N 引脚图

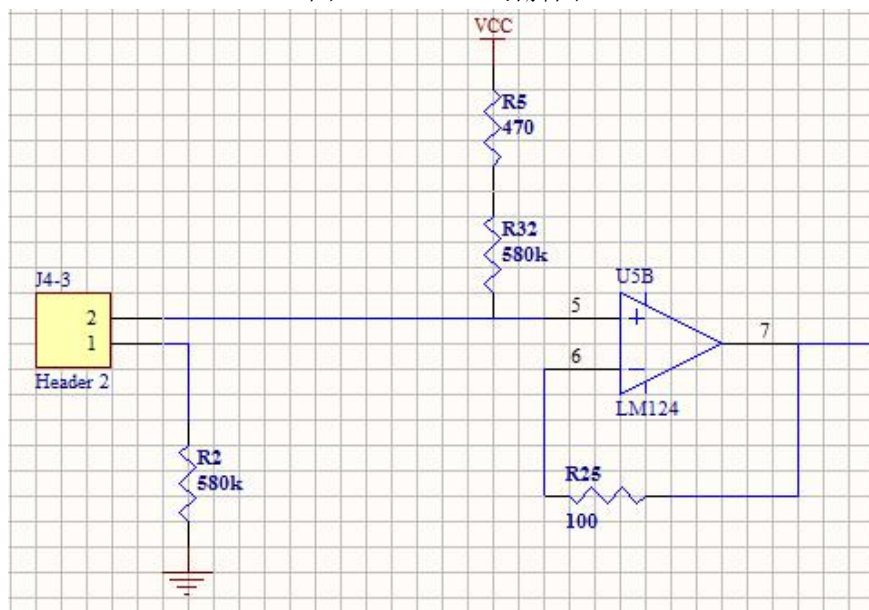


图 4-4 检查设备连入电路

LM324N 内部含有 4 个运算放大器,我们需要的运算放大器比较多,用 LM324N 更简单方便,同时也简化电路板的设计和排版。将设备从连线器接进电路,5V 电源 Vcc 通过电阻 R5 分压后给运算放大器提供一个基准电压即图中 A 点电压,被检查设备的电阻值变大或变小时对应的 A 点电压相应的变大或变小。R25 为运放的反馈电阻,用来抑制零点漂移的。检测电路输入电阻为 $(R5+R32)/R2$ 约为 $290K\Omega$,为了减少检测电路和被检查设备之间的相互影响。一般根据经验或 LM324 使用手册选取阻值为 100 欧姆的电阻。R2 用来释放静电,防止过大的静电破坏设备。运算放大器 U5B 的输入电阻无穷大所以反相端输入电流为零, R25 上面并没有电压降,所以同相、反相与输出端电压相等,运放的电压放大倍

数为 1。即图中 A 点电压就为运放输出电压，当被检查设备的电阻为 470 欧姆时，A 点的电压为 $1/2V_{cc}$ 。当被检查设备较小时 A 点的电压将会降低，运放输出的电压也会降低。当被检查设备的等效电阻大于 470 欧姆时，A 点的电压将大于 $1/2V_{cc}$ 。当被检查设备的等效电阻小于 470 欧姆时，A 点的电压将小于 $1/2V_{cc}$ 。电路图见图 4-4。

电源 V_{cc} 经过电阻 R4、R3 组成电压分压器，R3 的阻值调节为 200 欧姆，给运放提供一个基准电压即图中 B 点电压为 $1/2V_{cc}$ ，电容 C3 用来消除高频干扰，提高抗干扰能力。通过调节可调电阻 R3 就可以改变该运放的基准电压，运放 U5D 和电阻 R19 的作用和上图中的运放 U5B、R25 一致，则图中 B 点的电压即为运放的输出电压。所以可调电阻 R3 的电阻值越大对应的运放 U5D 的输出电压也就越大。电路见图 4-5。

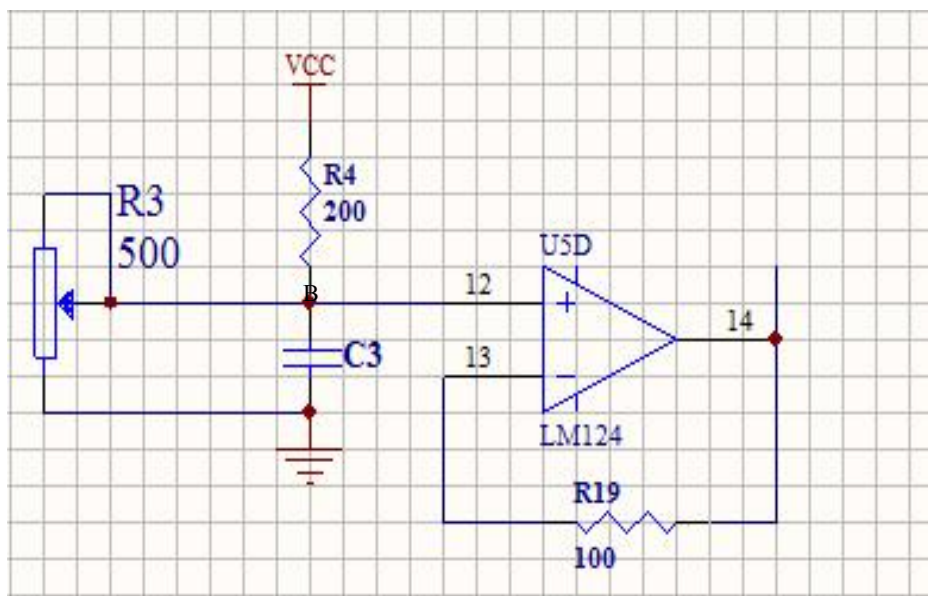


图 4-5 电阻比较值连入电路

将上面两个图中的运放 U5B、U5D 的输出端连接到第三个运算放大器 U5A 上面，即图 4-4 的输出端和图 4-5 的输出端各自串联一个限流电阻后作为第三个运放的同相输入端和反相输入端，即引脚 7 作为第三个运放的同相输入端引脚 3，引脚 14 作为第三个运放的反相输入端引脚 2，该运放放大器没有反馈电阻作为比较器使用。运放的输出端经过一个限流电阻后将信号传输给单片机引脚 P10。P10 的输出值就是 U5B 和 U5D 输出端的电压比较值。因 U5B 和 U5D 运放的功能一致放大倍数也一样，所以 P10 的值也就是这两个运放输入端对应的 A、B 点的电压的比较值，当被检查设备的电阻大于 470 欧姆时，A 点的电压会高于 $1/2V_{cc}$ ，则 U5B 的输出电压大于 $1/2V_{cc}$ ，而 U5D 的输出电压为 $1/2V_{cc}$ ，经过比较器 U5A 后同相输入电压大于反相输入电压则输出高电压，对应的单片机 P10 接口得到高电平。所以当被检查设备的电阻低于 470 欧姆的时候，单片机输出为低电平。低电平表示被检查设备正常，电路图见图 4-6。连接的第二个被检查设备和第一个设备一样连入电路中即可，输出到单片机的引脚 P27。LM324 中一共有 4 个运算放大器，不用的运放可以悬空，等于接高电位避免出现电压抖动影响其他运放的使用。因为电阻会有误差，所以我

们要留一定的裕量,所以当被检查设备的电阻低于 500 欧姆的时候,单片机输出为低电平,被检查设备正常。

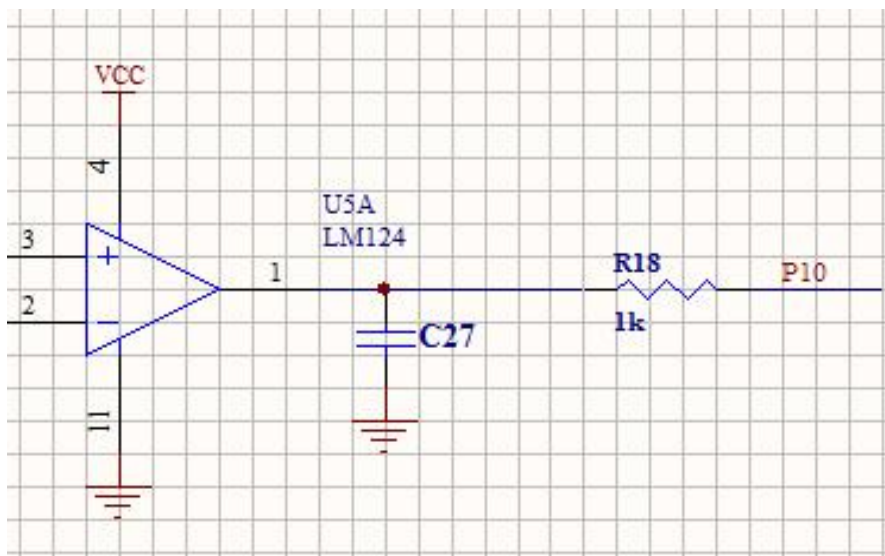


图 4-6 电阻比较值与被检查设备连入电路

4.3 静电手环连入系统电路设计

静电手环是由导电松紧带、活动按扣、弹簧 PU 线、保护电阻及插头或鳄鱼夹组成的,是一种用于释放人体所存留的静电以起到保护人体作用的小型设备。利用手环使操作工人接地。手环能在静电损坏电子产品之前,驱散人体所带静电。通过腕带上的塑料盒内的一个 1 兆欧电阻以及导线将静电通过防静电接地线引入大地,可以避免瞬间大电流的对元件的冲击,避免器件损坏,然而,串连该电阻主要是处于人身安全考虑。

此系统可以接入两个静电手环通过接口 OP1 和 OP2,我们以接入一个静电手环的接口 OP1 为例。两个二极管并联放在运算放大器的输入端,将输入信号幅度钳制在-0.7 到+5.7V 之间。为了提高瞬态响应速度,电阻 R11 两端并联电容 C6,使输入信号电压超前。二极管 V5 与 VD7 用途一致,为消除干扰,加入滤波电容 C4、C5,让电路更加稳定可靠。OP1 为静电手环连入电路的接口,Vcc 通过电阻 R14 通过 R15 分压后,给手环提供一个基准电位。

因为采集测量对象输出内阻较大(人体电阻约为 $10K\Omega$ 到 $50K\Omega$),将 OP1 的信号和 P20 的信号并联后作为运算放大器的同相输入端,反相输入端与输出端连在一起提高驱动能力,运算放大器虚段功能,正相输入和反相输入端电位相等,而且输出端和反相输入端相连,所以三个电位相等。同时,因为运放的输入电阻极大,所以对前级电路影响非常小。另外加上运放的输出电阻极小,因此运放常常连接组成一个电压跟随器,作为小信号大内阻的电路的输出驱动电路。LM324 运放的其他两个引脚接入电源的正负端。即图中 C 点电压就为运放输出电压,电路图见图 4-7。

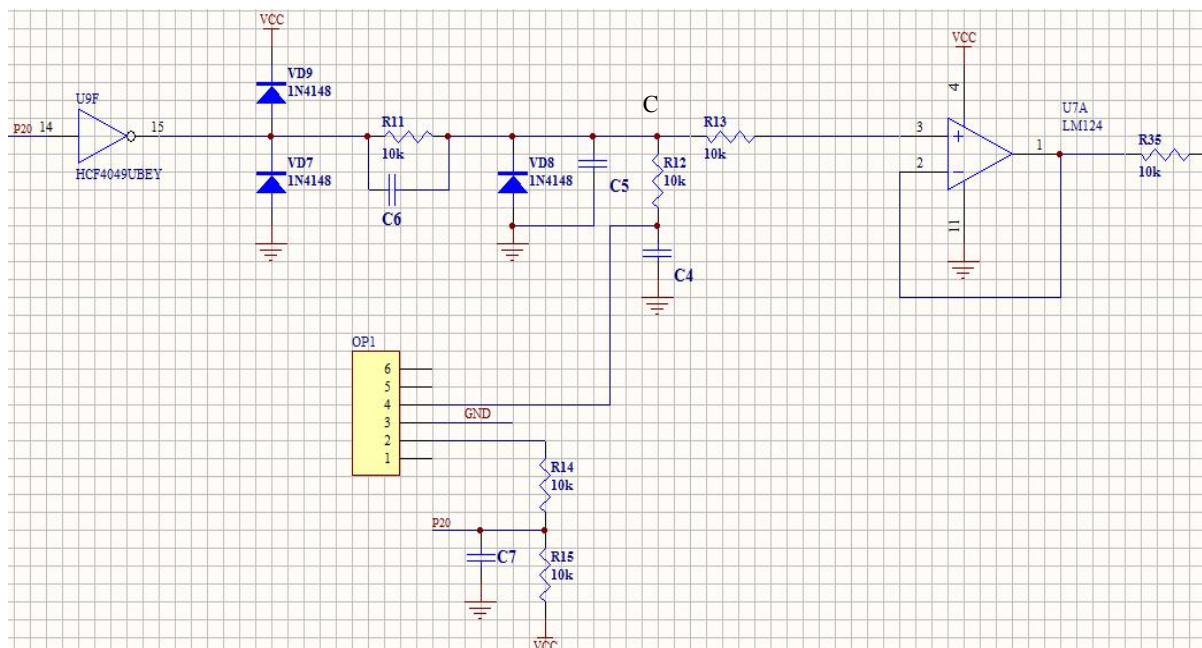


图 4-7 静电手环接入电路

电源 Vcc 经过电阻 R4、R7 组成电压分压器，给运放提供一个基准电压即图中 D 点电压。电容 C28 用来消除高频干扰，提高抗干扰能力。通过调节可调电阻 R4 就可以改变该运放的基准电压，运放 U7C 和电阻 R20 的作用和上面所提到的运放 U5B、R25 一致，则图中 D 点的电压即为运放的输出电压。所以可调电阻 R4 的电阻值越大对应的运放 U5D 的输出电压也就越大。当 R4 的电阻为 2000 欧姆时，运放 U7D 的输出电压为 3.6V，此时若静电手环的电阻为 5000 欧姆时，运放 U7A 输出电压也为 3.6V。电路图见图 4-8。

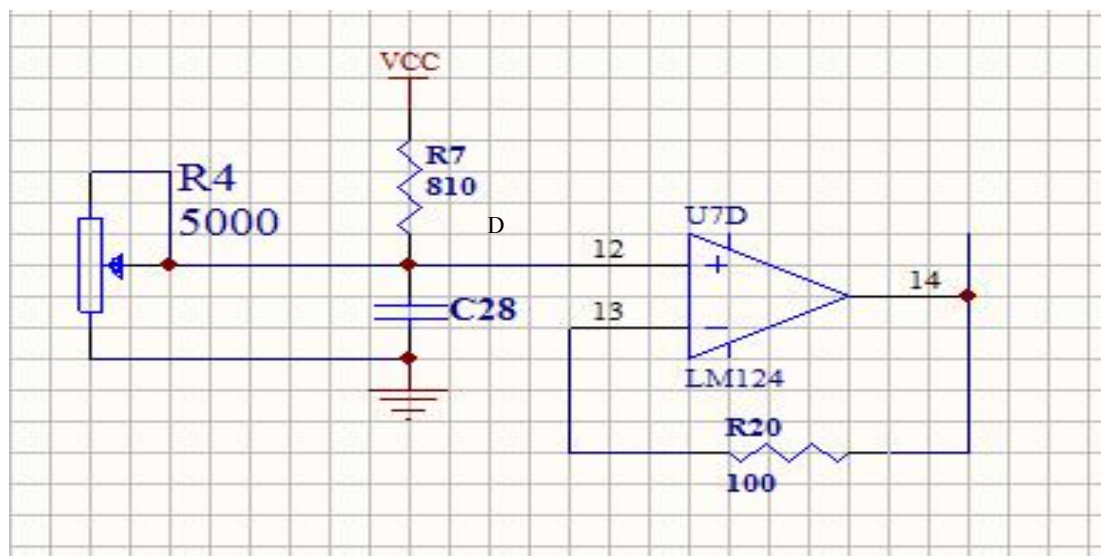


图 4-8 运算放大电路

将上面两个图中的运放 U7C、U7A 的输出端连接到第三个运算放大器 U7D 上面，即引脚 1 作为第三个运放的同相输入端引脚 12，引脚 8 作为第三个运放的反相输入端引脚 13，该运放放大器没有反馈电阻作为比较器使用。运放的输出端经过一个限流电阻后将信号传输给单片机引脚 P22。P22 的输入值就是 U7C 和 U7A 输出端的电压比较结果。因 U7C

和 U5D 对电压无放大, 所以 P22 的值也就是这两个运放输入端对应的 C、D 点的电压的比较值。当静电手环的电阻大于 5000 欧姆时, 运放 U7A 的输出电压也就大于 3.6V, 也就是大于 U7D 的输出端电压, 经过比较器 U7D 后同相输入端的电压大于反相输入端的电压, 输出端为高电平, 静电手环不正常, 只有当静电手环电阻小于 5000 欧姆时, 输出端为低电平, 静电手环正常, 电路图见图 4-6。连接的第二个静电手环和第一个手环一样连入电路中即可, 输出到单片机的引脚 P22。LM324 中一共有 4 个运算放大器, 不用的运放输入接地, 输出悬空, 避免出现电压抖动影响其他运放的使用。

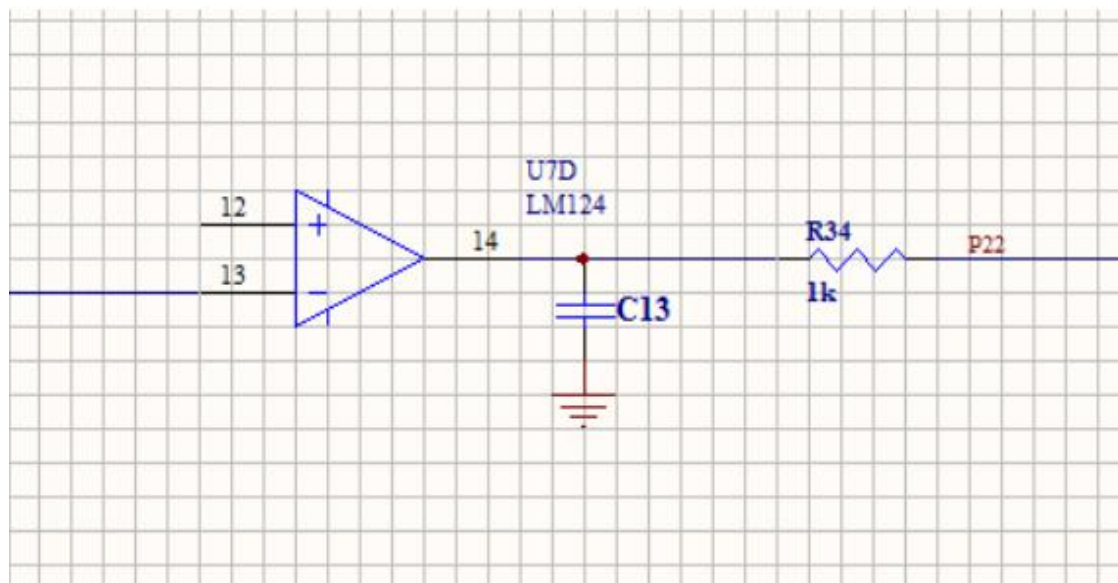


图 4-9 电阻比较值接入电路

4.3.1 AT89S52 单片机介绍

AT89S52 是低功耗, 高性能, 采用 CMOS 工艺的 8 位单片机。其片内具有 8KB 的可在线编程的 Flash 存储器。该单片机与工业标准型 80C51 单片机指令和引脚完全兼容, 片内的 Flash 存储器可在线重新编程, 或使用通用的非易失性存储器编程器; 通用的 8 位 CPU 与在线可编程 Flash 集成在一个芯片上, 从而使 AT89S52 的性能更加完善, 应用更加灵活; 具有较高的性能价格比, 使其在嵌入式控制系统中有着广泛的应用。AT89S52 的引脚排列见图 4-10。

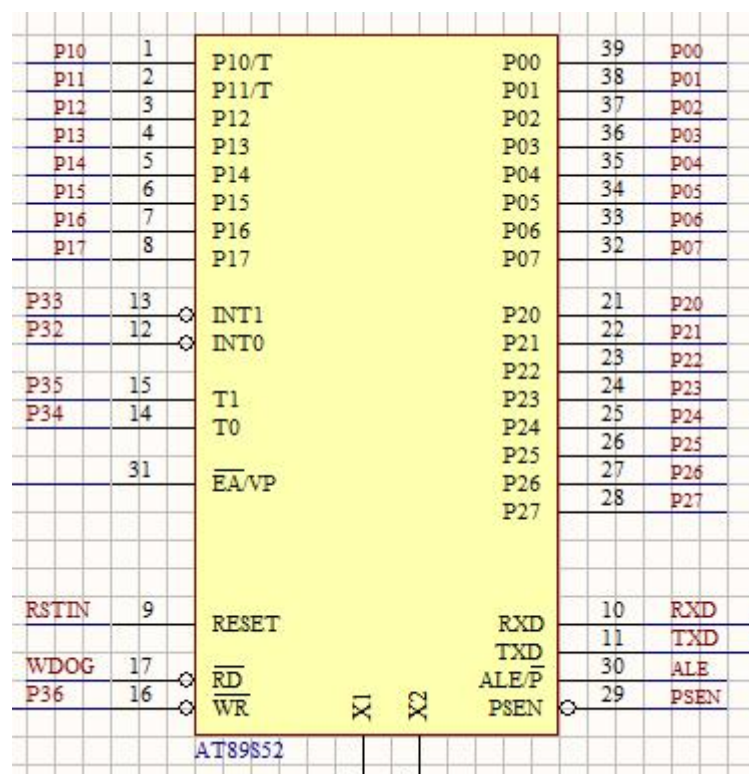


图 4-10 AT89S52 引脚图

1.RST: RST 是复位输入端, 振荡器工作时 RST 引脚出现两个机器周期以上的高电平时可使单片机复位。

2.ALE/ $\overline{\text{PROG}}$: ALE/ $\overline{\text{PROG}}$ 是地址允许锁存/编程脉冲输入, 当访问外部数据存储器或程序存储器时, ALE 输出脉冲来锁存地址的低 8 位字节。一般情况下 ALE 以时钟振荡频率的 1/6 输出脉冲信号, 因此它可对外输出时钟或用来定时。

3. $\overline{\text{PSEN}}$: $\overline{\text{PSEN}}$ 是外部程序存储器的读选通信号, 低电平有效。当 AT89C52 由外部程序存储器读取指令或数据时每个机器周期两次 PSEN 有效, 即输出两个脉冲。在访问外部数据存储器时无效。

4. $\overline{\text{EA}}/\text{V}_{\text{PP}}$: $\overline{\text{EA}}/\text{V}_{\text{PP}}$ 为外部访问允许端。想要让 CPU 只访问外部程序存储器 $\overline{\text{EA}}$ 端必须保持低电平, 地址为 0000H~FFFFH。若 $\overline{\text{EA}}$ 接高电平则执行内部程序存储器中的指令。

5.XTAL1: XTAL1 是片内振荡器反相放大器及内部时钟发生器的输入端。

6.XTAL2: XTAL2 是片内振荡器反相放大器的输出端。

4.3.2 AT89S52 振荡电路

单片机 AT89S52 的内部有一个反相放大器, 单片机芯片的引脚 XTAL1、XTAL2 (即单片机的 19、18 引脚) 分别为方相相放大器的输入端和输出端。在这两个端口之间连接一个石英晶体或陶瓷谐振器。若使用石英晶体, C1 和 C2 的取值范围为 22~33pF, 本电路图中选取了 22pF 的石英晶体。在本设计中为了软件方面便于调试串口的波特率, 选用了 11.592MHZ 的晶振。这便构成了一个振荡电路, 其具体连接电路见图 4-11。振荡电路产生振荡电流, 为系统提供基本的时钟信号。

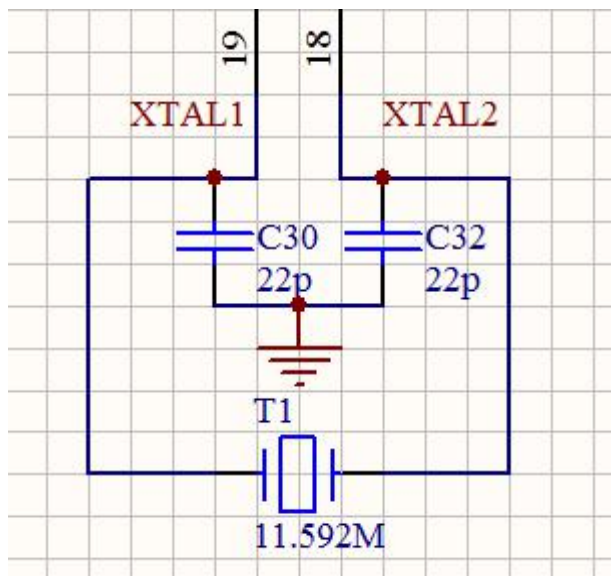


图 4-11 AT89S52 振荡电路

4.3.3 AT89S52 监控电路

MAX813L 常用在计算机等控制器的监控电路中单片机在执行指令的时候会受到干扰的冲击，使指令出错，导致程序“跑飞”或进入死循环。为了使程序自动恢复重新工作，采用硬件看门狗技术。若程序发生故障时，所有的运行状态全部停止，只有 RAM 和 SFR 中的数据被保存下来。当电源发生故障时保存数据且断开外围设备用电电源，降低整个系统的功耗。当电源恢复正常后由看门狗产生复位信号，使单片机重新进入正常工作状态。对单片机 AT89S52 进行监控。引脚图见图 4-12。

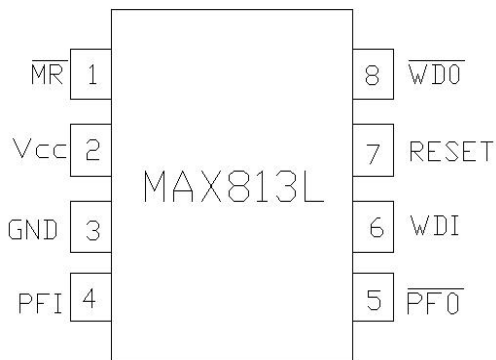


图 4-12 MAX813L 引脚图

MAX813L 引脚功能如下：

1. $\overline{\text{MR}}$ ：手动复位输入，低电平有效；
2. PFI：电源故障输入；
3. $\overline{\text{PFO}}$ ：电源故障输出；
4. WDI：看门狗输入；
5. RESET：复位输出；
6. $\overline{\text{WDO}}$ ：看门狗输出。

MAX813L 功能如下:

1.上电、掉电状态下复位功能,引脚 7 连接到 AT89S52 的 RSTIN 端口,就不需要单片机再外接复位电路;

2.MAX813L 的 WDI 引脚还有看门狗输出功能,连接到 AT89S52 的 WDOG 端口。如果在 1.6s 内 WDI 端口没有脉冲输入,则 \overline{WDO} 输出个低电平信号;

3.内有一个 1.25V 掉电告警门限检测器;

4.手动复位输入, \overline{MR} 端为手动复位控制端,低电平有效。 \overline{MR} 输出一个低电平则 RST 产生一个复位输出,控制单片机的复位;

5.GND 和 V_{CC} 引脚连接到电源的正负极。

MAX813L 的引脚 1 和引脚 8 与相连,RESET 脚接单片机的复位脚即 AT89S52 的 RST 引脚;WDI 与单片机看门狗相连,不断输出脉冲,如果因为某种原因导致单片机进入死循环,则单片机无脉冲输出,于是 1.6s 后 \overline{WDO} 输出个低电平信号,该低电平加到 \overline{MR} 端,使 MAX813L 的 RST 产生复位输出,则单片机有效复位,摆脱死循环。监控电路见图 4-13。

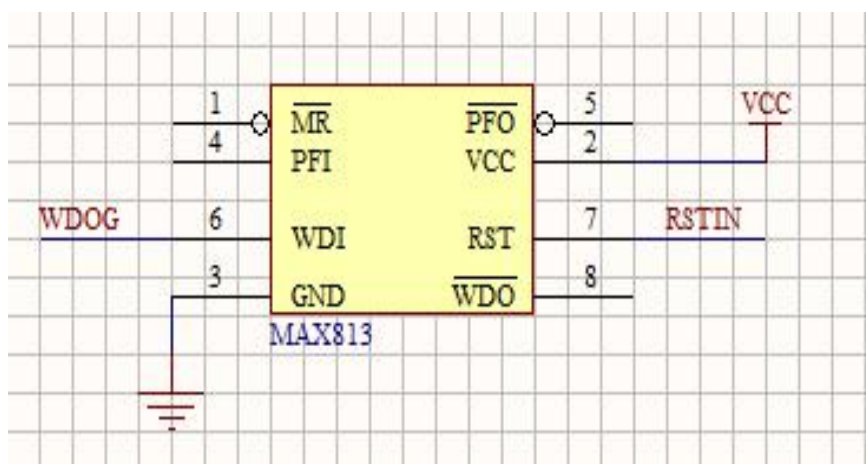


图 4-13 AT89S52 监控电路

4.3.4 AT89S52 与蜂鸣器连接电路

这里选用的是有源蜂鸣器并且是电磁式,额定电压为 5V 工作电压为 4~7V。蜂鸣器的一端通过电阻 R52 连到电源上,R52 用一个小阻值的电阻来限流和防止蜂鸣器短路时破坏电流,也可以不用。蜂鸣器的另外一端连到了 \overline{WR} 端,当单片机的 P36 输出低电平时蜂鸣器发出声音。电容 C33 起到滤波的作用。单片机的驱动电流很小所以电源经过电阻 R52 后再连入单片机接口。电路连接图见图 4-14。

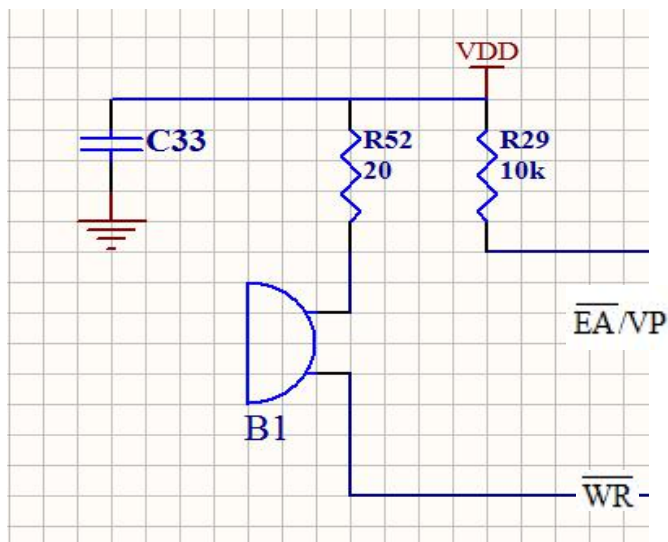


图 4-14 蜂鸣器连接电路图

4.3.5 AT89S52 与 LED 灯连接电路

将 10 个发光二极管的负极依次连到 AT89S52 的 10 个端口，低电平有效。所有二极管上面接一个上拉电阻，通过上拉电阻来增大电流，提高 VDD 端驱动能力。当单片机端口输出低电平时，对应的 LED 灯亮，提示工作人员被检查设备和静电手环情况。电路图见图 4-15。

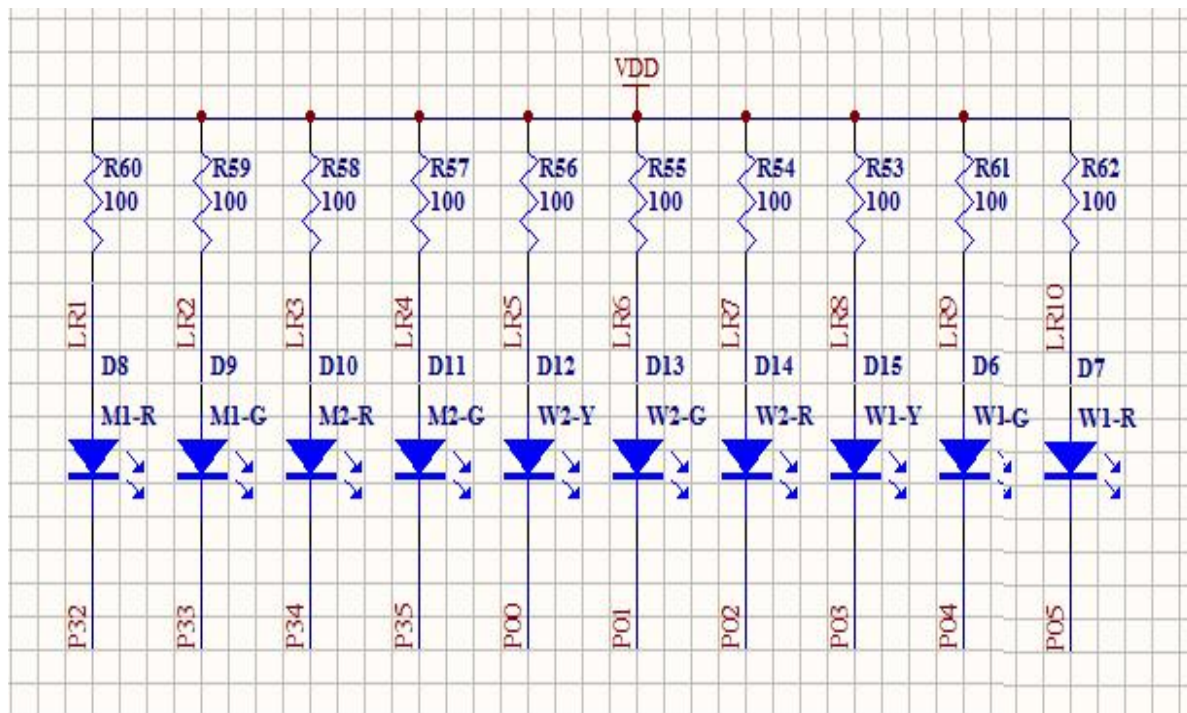


图 4-15 AT89S52 与 LED 灯连接电路图

单片机端口 P00 对应备用的 LED 亮黄灯，P01 对应静电手环 2 的 LED 亮绿灯，P02 对应静电手环 2 的 LED 亮红灯，P03 对应备用的 LED 亮黄灯，P04 对应静电手环 1 的 LED 亮绿灯，P05 对应静电手环 1 的 LED 亮红灯，P32 对应设备 1 的 LED 亮绿灯，P33 对应设备 1 的 LED 亮红灯，P34 对应设备 2 的 LED 亮绿灯，P35 对应设备 1 的 LED 亮红灯。

当静电手环正常时，对应的 LED 绿灯亮。当静电手环电阻超上限时，对应的 LED 红灯亮。当被检查设备正常时，对应的 LED 绿灯亮，当被检查设备超上限时，对应的 LED 红灯亮。

5 软件设计

5.1 单片机的主要片脚接口

AT89S52 内部有 40 个引脚，部分引脚功能端见表 5-1。

表 5-1 单片机主要片脚接口

序号	单片机片脚	所接功能端
1	$\overline{\text{EA}}/\text{V}_{\text{PP}}$	接高电平，执行内部程序
2	XTAL1、XTAL2	接晶振电路
3	P10	被检查设备 1 的检测输入口
4	P27	被检查设备 2 的检测输入口
5	RXD、TXD	接收发器
6	VCC、GND	接电源电路
7	WDOG	接入看门狗电路
8	$\overline{\text{RD}}$ 、RESET	接入 MAX813L 监控电路
9	$\overline{\text{WR}}$	连接蜂鸣器
10	P12~P16	拨码开关，电路板编号
11	P11	拨码开关，选择被检查设备 1
12	P25	拨码开关，选择被检查设备 2
12	P23	手环 2 的屏蔽信号
13	P20	手环 1 的屏蔽信号
14	P22	静电手环 1 的检测输入口
15	P26	静电手环 2 的检测输入口
16	P35	被检查设备 2 亮绿灯
17	P34	被检查设备 2 亮红灯
18	P33	被检查设备 1 亮绿灯
19	P32	被检查设备 1 亮红灯
20	P01	静电手环 2 亮绿灯
21	P02	静电手环 2 亮红灯
22	P03	备用 LED 黄灯
23	P00	备用 LED 黄灯
24	P04	静电手环 1 亮绿灯
25	P05	静电手环 1 亮红灯

5.2 单片机程序控制流程图

5.2.1 被检验设备程序流程图

系统开始执行后先判断一下被检查设备对应的按键是否按下，确认按下后再通过单片机的 P10 和 P27 的高低电平来比较被检查设备电阻值接入电阻是否小于固定电阻值 500 欧姆，从而来确定设备是否正常，设备正常时的电阻比较值应该小于 500 欧姆即与单片机的 P10 引脚输出为低电平，此时与被检查设备对应的 LED 绿灯亮，当被检查设备超过预定电阻 500 欧姆即与单片机相连的引脚输出为高电平，对应的 LED 红灯亮，同时蜂鸣器对应的单片机引脚 P36 为低电平，蜂鸣器发出声音。提醒工作人员注意。被检查设备流程图见图 5-1。

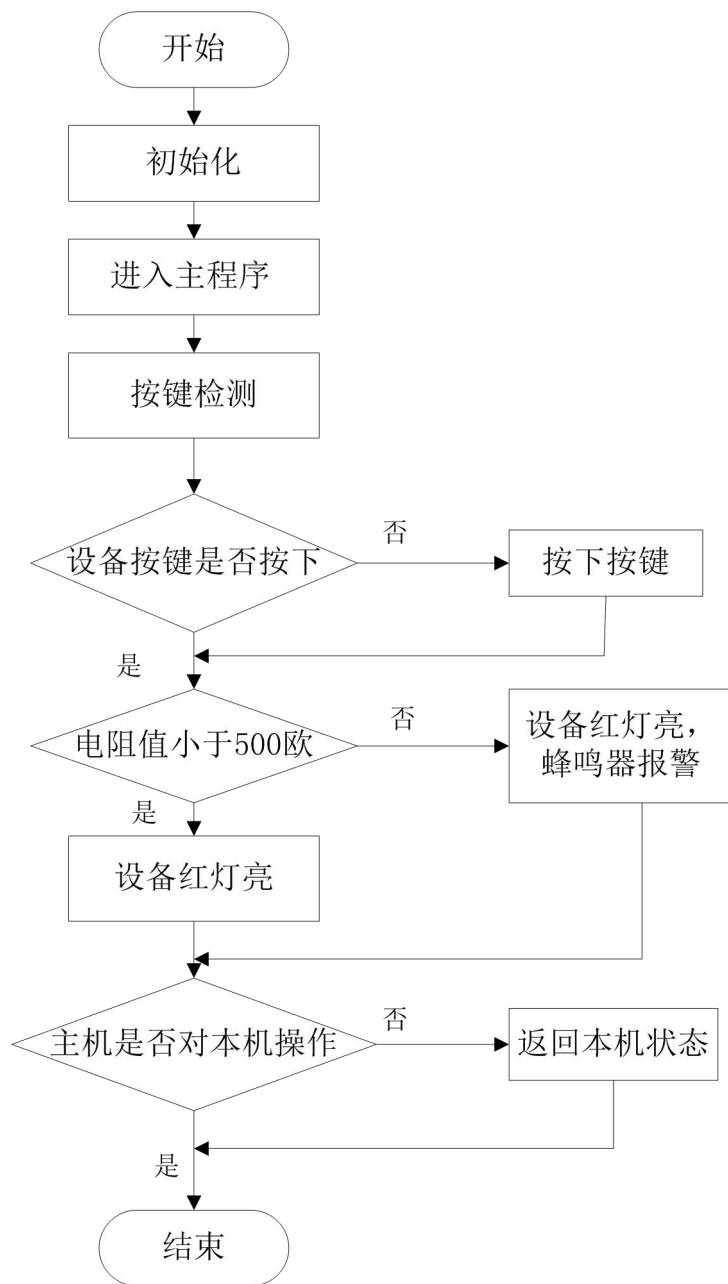


图 5-1 被检查设备流程图

5.2.2 静电手环监测流程图

系统开始执行后先确认一下是否屏蔽了两个静电手环对应的接口, 确认没有屏蔽后再通过单片机的 P22 和 P26 的高低电平来比较静电手环电阻值接入电阻是否小于固定电阻值 5000 欧姆, 从而来确定设备是否正常, 手环正常时的电阻比较值应该小于 500 欧姆即与单片机的 P22 引脚输出为低电平, 此时与静电手环对应的 LED 绿灯亮, 当静电手环超过预定电阻 500 欧姆即与单片机相连的引脚 P22 输出为高电平, 对应的 LED 红灯亮, 同时蜂鸣器对应的单片机引脚 P36 为低电平, 同时蜂鸣器发出声音。被检查设备流程图见图 5-2。

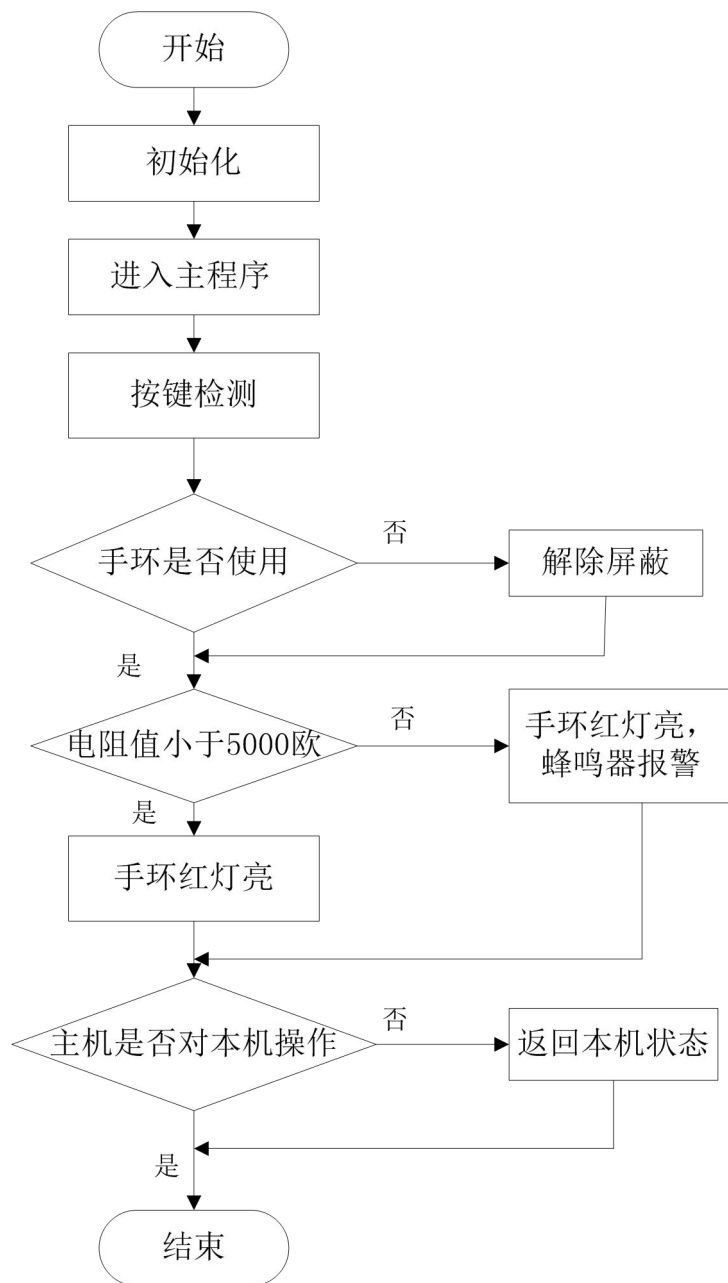


图 5-2 静电手环检测流程图

5.3 系统主程序

系统主程序如下所示。

```
void main()
{
    UART_CMD();                //串口初始化
    inti();                    //定时器 0 初始化
    while(1)
    {
        scan();                //按键扫描
        uarts();
        if((hand_1==0)&&(hand_2==0)&&(unit_1==0)&&(unit_2==0))
        {
            TR0=0;ET0=0;EA=0;spk=0;
        } //如果手环 1、2 设备 1、2 都正常，就关闭定时器 0 关闭蜂鸣器
        if(key1==1)             //手环是否使用标志
        {
            if(hand_1==1)        //正常输出为 0，超上限输出为 1
            {
                hand_1r=1;
                hand_1g=0;        //电阻超上限，LED 红灯亮
                TR0=1;
                ET0=1;            // 启动定时器 0，蜂鸣器响
            }
            else {hand_1g=1; hand_1r=0;}
        }
        if(key2==1)             //手环 2 是否使用标志 /
        {
            if(hand_2==1)        //正常输出为 0，超上限输出为 1
            {
                hand_2r=1;        //电阻超上限，LED 红灯亮
                hand_2g=0;
                TR0=1;
            }
        }
    }
}
```

```

        ET0=1;                                //启动定时器 0, 蜂鸣器响

    }
    else {hand_2g=1;hand_2r=0;}
}
if(key3==1)                                  //被检查设备 1 使用标志
{
    if(unit_1==1)                            //正常输出为 0, 超上限输出为 1
    {
        unit_1r=1;                          //电阻超上限, LED 红灯亮
        unit_1g=0;
        TR0=1;
        ET0=1;                              //启动定时器 0, 蜂鸣器响

    }
    else {unit_1g=1; unit_1r=0;}
    key3=0;                                  //清除标志
}
if(key4==1)                                  // 被检查设备 2 使用标志
{
    if(unit_2==1)                            // 正常输出为 0, 超上限输出为 1
    {
        unit_2r=1;                          //电阻超上限, LED 红灯亮
        unit_2g=0;
        TR0=1;                              //启动定时器 0, 蜂鸣器响
        ET0=1;

    }
    else {unit_2g=1;unit_2r=0;}
    key4=0;                                  //清除标志
}
}

```

6 硬件测试

系统是否正常运行由这 10 个 LED 灯的亮灭和蜂鸣器来表现,上面一行的 LED 灯表示静电手环 1 和被检查设备 1,第二排的 LED 灯表示静电手环 2 和被检查设备 2,前面两排的红灯和绿灯表示静电手环,后面两排的红灯和绿灯表示被检查设备。中间的 2 个黄色 LED 灯留着备用。当其他 LED 发生故障时,改变程序中接口的定义就能将黄色的 LED 灯重新定义。系统没有连接电源之前所有的小灯泡都不亮,蜂鸣器也不响。系统图见图 6-1。我们以下的检测以各连接一个静电手环和一个被检查设备为准。



图 6-1 系统未运行时 LED 灯

当设备正常且工作人员正确佩戴静电手环时,静电手环电阻比较值和被检查设备电阻比较值都没有超过上限,单片机端口输出为低电平。根据程序判断得到静电手环对应的第一排的 LED 绿灯亮,且被检查设备对应的第四排的绿灯也亮,定时器 0 没有启动蜂鸣器不响,见图 6-2。

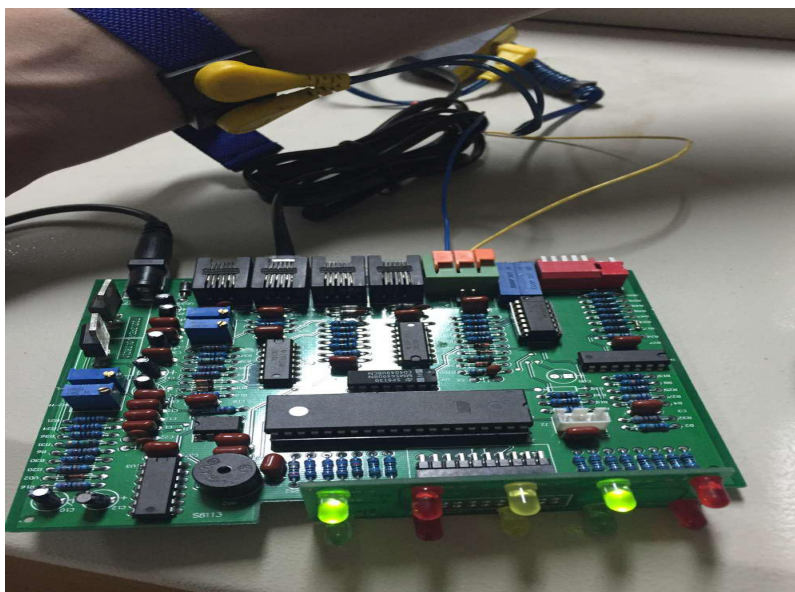


图 6-2 系统正常运行时 LED 灯

设备正常但工作人员没有佩戴静电手环时，被检查设备电阻比较值没有超过上限但静电手环电阻比较值超过上限，手环对应的单片机端口输出为高电平，根据程序判断得到静电手环对应的第二排的 LED 红灯亮。被检查设备对应的单片机端口输出为低电平，根据程序判断得到被检查设备对应的第四排 LED 绿灯亮，定时器 0 启动蜂鸣器响，见图 6-3。

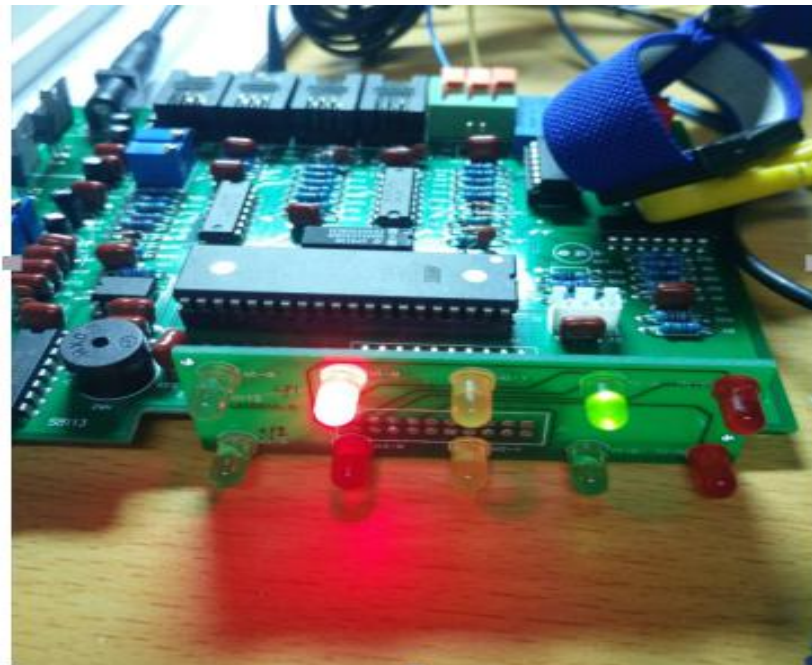


图 6-3 系统故障时 LED 灯

设备不正常但工作人员佩戴静电手环时，被检查设备电阻比较值超过上限但静电手环电阻比较值没有超过上限，手环对应的单片机端口输出为低电平，根据程序判断得到静电手环对应的第一排的 LED 绿灯亮。被检查设备对应的单片机端口输出为高电平，根据程序判断得到被检查设备对应的第五排 LED 红灯亮，定时器 0 启动蜂鸣器响，见图 6-4。

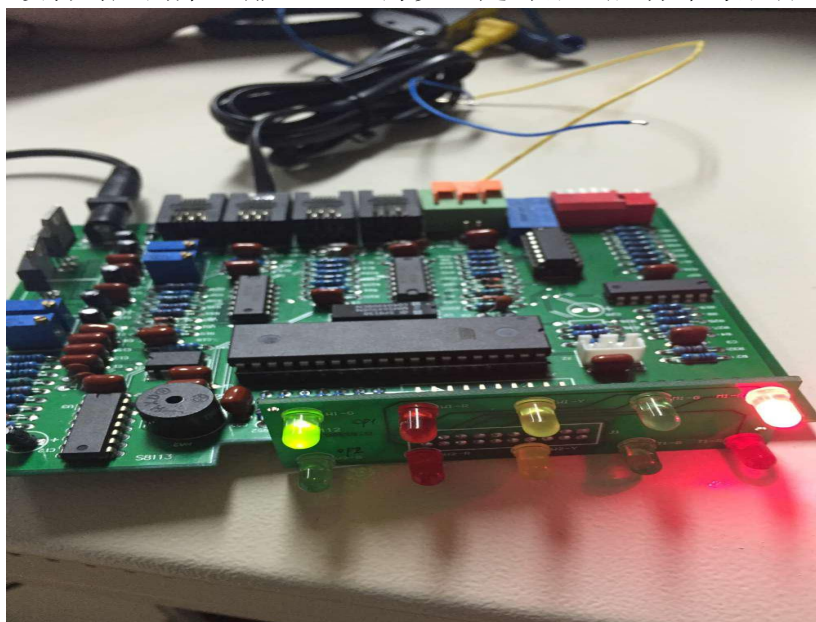


图 6-4 系统故障时 LED 灯

设备没有正常接入且工作人员没有佩戴静电手环时，被检查设备电阻比较值超过上限

但静电手环电阻比较值超过上限，手环对应的单片机端口输出为高电平，根据程序判断得到静电手环对应的第一排的 LED 红灯亮。被检查设备对应的单片机端口输出为高电平，根据程序判断得到被检查设备对应的第五排 LED 红灯亮，定时器 0 启动蜂鸣器响，见图 6-5。

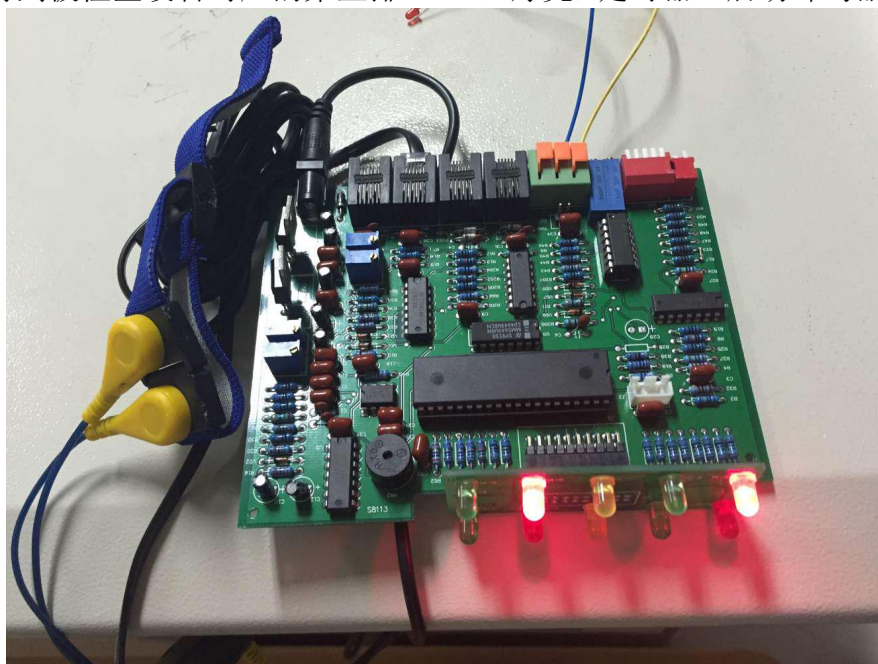


图 6-5 系统故障时 LED 灯

结论

通过单片机 AT89S52、运算放大器、LED 灯、蜂鸣器等组成一个简单的静电防护监控系统。使用该系统实现 LED 显示防静电情况，进行可视化监控管理，防静电更可靠。当工作人员静电泄放出现异常时，联网报警系统自动亮灯并发出蜂鸣声报警。从听觉上来监控，约束员工规范配带。本产品实现连续实时监测操作人员的静电手腕带佩戴情况、使用的仪器设备防静电情况等。使整个车间实现统一监控，出现异常状况，自动发出报警。确保 ESD 状况获得的正常。

但是虽然系统留出了连接主机的接口，方便日后进行联网管理，暂时由于其他原因未能将其连入主机。只能通过 LED 和蜂鸣器进行报警，系统不太完善。

致谢

本论文是通过潘晓博老师辛勤的指导才得以完成，所以首先要感谢给与我们巨大帮助的老师，同时也要感谢学校给与了我们学生这个平台锻炼自己，为我们步入社会打下坚实的基础。也感谢我实习的公司对我的巨大信任让我有机会参与到这个项目中来，感谢公司领导对我的耐心指导和教诲。经过 2 个多月的努力，基本上完成了设计的要求。

在论文工作中，感谢指导老师给与我巨大的帮助，细心的讲解和耐心的指导让我能够按时间内完成工作。在此，对论文的指导老师、支持与帮助我的同学和同事和在四年中给予我教导的老师表示最衷心的感谢。

参考文献

- [1] 石东海.单片机数据通信技术[M].西安:电子科技大学出版社,2002.
- [2] 程昱.Protel DXP 电路设计[M].北京:清华大学出版社,2003.
- [3] 董传岱.数字电子技术基础[M].杭州:中国石油大学出版社,2009.
- [4] 吴宗汉.基础静电学[M].北京:北京大学出版社,2010.
- [5] 白洪欢.汇编语言程序设计[M].北京:科学出版社,2003.
- [6] 袁亚飞.电子工业静电防护技术与管理[M].北京:中国宇航出版社,2013.
- [7] 吴宗汉.基础静电学[M].北京:北京大学出版社,2010.
- [8] 张平,葛金印.AT89S52 单片机基础项目教程[M].北京:北京理工大学出版社,2012.
- [9] 孙玉才.ATMEL 新型 AT89S52 系列单片机及其应用[M].北京:清华大学出版社,2005.
- [10] 姜志海,黄育清,刘连鑫.单片机原理及应用[M].北京:电子工业出版社,2013.
- [11] 李广弟,朱月秀.单片机基础[M].北京:北京航空航天大学出版社,2007.
- [12] 韩雁,董树荣.集成电路 ESD 防护设计理论、方法与实践[M].北京:科学出版社,2014.
- [13] 刘志伟,陈继芝.ESD 设计与综合[M].北京:机械工业出版社,2013.

附录

系统程序如下所示。

```
#include <reg52.h>
#define uchar unsigned char
#define uint unsigned int
bit key1,key2;          //手环是否使用标志
bit key3,key4;          //设备检测按键标志
int dog;
uchar log;              程序进行时向主机发送本机状态
sbit hand_2g=P0^1;      // 2#手环绿色指示灯
sbit hand_2r=P0^2;      // 2#手环红色指示
sbit hand_1g=P0^4;      // 1#手环绿色指示灯
sbit hand_1r=P0^5;      // 1#手环红色指示灯
sbit unit_2g=P3^5;      // 2#设备绿色指示灯
sbit unit_2r=P3^4;      // 2#设备红色指示灯
sbit unit_1g=P3^3;      // 1#设备绿色指示灯
sbit unit_1r=P3^2;      // 1#设备红色指示灯
sbit unit_1=P2^7;       // 1#设备检测输入口
sbit unit_2=P1^0;       // 2#设备检测输入口
sbit spk = P3^6;        // 蜂鸣器
sbit scan_3=P1^1;       // 1#设备检测按键
sbit scan_4=P2^5;       // 2#设备检测按键
sbit hand_1=P2^2;       // 1#静电手环检测端口
sbit hand_2=P2^6;       // 2#静电手环检测端口
sbit sw_1=P2^0;         //屏蔽手环 1 报警
sbit sw_2=P2^3         //屏蔽手环 2 报警
sbit bm1=P1^2;          //拨码开关
sbit bm2=P1^3;
sbit bm3=P1^4;
sbit bm4=P1^5;
sbit bm5=P1^6;
sbit wdog=P1^7;         //看门狗信号
sbit cont=P1^7;         //491 发送接收控制位
uchar bm;              //存储本地编码
```

```

uchar tab_bm[3];
uchar code  table0[]={"本机工作正常  "};
uchar code  table1[]={"1#手环正常使用"} ;
uchar code  table2[]={"2#手环正常使用"} ;
uchar code  table3[]={"1#手环异常使用"} ;
uchar code  table4[]={"2#手环异常使用"} ;
uchar code  table5[]={"本机 1#手环闲置"} ;
uchar code  table6[]={"本机 2#手环闲置"} ;
uchar code  table7[]={"本机 1#设备正常"};
uchar code  table8[]={"本机 2#设备正常"} ;
uchar code  table9[]={"本机 1#设备异常"};
uchar code  table10[]={"本机 2#设备异常"} ;
void delay(unsigned char ms);    //延时函数
void inti() ;                    //初始化函数
void scan();                     //按键处理函数
void UART_CMD() ;               //串口初始化
void uarts();                   // 串口通讯
void delay(unsigned char ms)    // 延时子程序,以 1ms 为一个单位
{
    unsigned char i;
    while(ms--)
    {
        for(i = 0; i < 120; i++);
    }
}
i=P1&0x7c;                      //读取本地编码 屏蔽 P1 口的最高位和最低两位
delay(20);
if(i==(P1&0x7c))
{
    i&=0x7c;
    i=i>>2;
    bm=i;                        //本地编码保存在 bm 变量中
}
if(scan_3==0)                   //检测 1#设备按键
{

```

```
        delay(20);
        if(scan_3==0)                                // 消除抖动

                key3=0;                                //设备 1 的拨码开关为 0
        }
        if(scan_4==0)                                //检测 2#设备按键
        {
                delay(20);
                if(scan_4==0)
                        key4=0;
        }
}

void inti()                                           // 定时器 0 中断初始化程序
{
        TMOD = 0x01;                                //定时器工作方式 1
        TH0 = (65536-500)/256;                       // 装入定时器初值
        TL0 = (65536-500)%256;
        EA=1;                                         //中断使能
}

void timer0() interrupt 1                            //定时器 0 中断服务程序
{
        TH0 = (65536-500)/256;
        TL0 = (65536-500)%256;
        spk = ~spk;                                  //驱动蜂鸣器
        dog++;
        if(dog==1000){wdog=1;wdog=0;dog=0;}        ////给定看门狗脉冲信号
}

void main()
{

        UART_CMD();                                  //串口初始化
        inti();                                       // 定时器 0 初始化, 工作方式 1 定时器
        while(1)
        {
                scan();                                ////按键扫描
```



```
    uarts();
    if((hand_1==0)&&(hand_2==0)&&(unit_1==0)&&(unit_2==0))
    {
        TR0=0;ET0=0;EA=0;spk=0;
    }

    if(key1==1)
    {
        if(hand_1==1)
        {
            hand_1r=1;
            hand_1g=0;
            TR0=1;
            ET0=1;
            TR0=1 为启动定时器 0， ET0=1 定时器 0 中断使能

        }
        else {hand_1g=1; hand_1r=0;}
    }
    else sw_1=1;
    if(key2==1) //判断 2#手环是否使用
    {
        if(hand_2==1)
        {
            hand_2r=1;
            hand_2g=0;
            TR0=1;
            ET0=1;

        }
        else {hand_2g=1;hand_2r=0;}
    }
    else sw_2=1;

    if(key3==1)
    {
```

```

        if(unit_1==1)
        {
            unit_1r=1;
            unit_1g=0;
            TR0=1;
            ET0=1;

        }
        else {unit_1g=1; unit_1r=0;}
        key3=0;                                //清除检测标志
    }
    if(key4==1)                                //检测 2#设备
    {
        if(unit_2==1)
        {
            unit_2r=1;
            unit_2g=0;
            TR0=1;
            ET0=1;

        }
        else {unit_2g=1;unit_2r=0;}
        key4=0;                                //清除检测标志
    }
}

```

串口通讯初始化

```

void UART_CMD()                                //串口初始化
{
    SCON = 0X50;    //串口工作方式 3，
    SCON = 0XD0;    // 串口工作方式 1，允许串口接收
    EA=1;           //中断允许开
    ES=1;           //允许串口中断开
    TMOD &= 0x0F;
}

```

```

TMOD |= 0X20;    //定时器 1 为工作方式 2
TL1=0xfd;        //9600 波特率
TH1=0xfd;
TR1=1;
cont=1;          //接收状态
}

void sio_int() interrupt 4 using 2          //串口接收
{
    uchar z,zd;
    ES=0;                                //关串口中断
    if(RI==1)                            //判断接收中断标志位是否为 1
    {
        z=SBUF;                          //移出串口接收寄存器
        if((log==0)&&z>(0x29)&&(z<0x40))    //仅接收文本发送的数字
        {
            {tab_bm[zd]=z; zd++;          //将接收字符送入数组 tab_bm
              if(zd>3 ) {zd=0;log=1;}      //接收满三个字符，将标志 log 置 1
            }
        }
    }

    RI=0;                                //数据已接收，清除接收中断标志
}

ES=1;                                    //开串口中断
}

void UARTSendByte(unsigned char* str)      // 串口发送字符串
{
    int i = 0,j=600;
    while(str[i]!=0)
    {
        SBUF = str[i];                  //字符串逐个送入串口发送寄存器
        i++;
        while(TI==0)                    //等待数据发送完毕
            {j--;if(j==0)TI=1;}          //最多等待 600 个指令周期，防止陷入死循环
        TI=0;                            //清除发送中断标志
        if( i>15) break;                 //发完 14 个字符，停止发送
    }
}

```

```

}

void trans(unsigned char dat)           //串口发送单个字符
{
    SBUF=dat;                           //将字符送入串口发送寄存器
    while(!TI);                         // 等待发送
    TI=0;                               //清除发送中断标志
}

void uarts()                           ///向上位机发送当前工况
{
    int j,i;

    if(log==1)                          //如果接收到数据标志为 1，进入处理程序
    {

        for(i=0;i++;i<2)
        {
            j=tab_bm[i]+j*10;           //将字符转化为数值
        }
        if(j==991)key1=1;               //如果主机发送 991，就屏蔽手环 1
        if(j==992) key3=1;              //如果主机发送 992，就屏蔽手环 2

        if(j==bm)                      //判断是否发送给本机信号
        {
            cont=0;                    //置 max491 为发送状态
            if((hand_1==0)&&(hand_2==0)&&(unit_1==0)&&(unit_2==0))
                UARTSendByte(table0); //如果没有任何异常，发送“本机工作正常”
            if(hand_1==0)                //如果 1#手环没有警报
                UARTSendByte(table1);   //发送 1#手环正常
            if(hand_2==0)                //如果 2#手环没有警报
                UARTSendByte(table2);   //发送 2#手环正常，
            if(hand_1==1)                //如果 1#手环有警报
                UARTSendByte(table3);   //发送 1#手环异常
            if(hand_2==1)                //如果 2#手环有警报
                UARTSendByte(table4);   //发送 2#手环异常
            if(key1==1)                  // 如果 1#手环闲置，发送手环未使用
                UARTSendByte(table5);
        }
    }
}

```

```
if(key2==1)                                // 如果 2#手环闲置, 发送手环未使用
    UARTSendByte(table6);
if(unit_1==0)                              // 如果 1#设备没有警报
    UARTSendByte(table7);                  // 发送 1#设备正常
if(unit_2==0)                              // 如果 2#设备没有警报
    UARTSendByte(table8);                  // 发送 2#设备正常
if(unit_1==1)                              // 如果 1#设备有警报
    UARTSendByte(table9);                  //发送 1#设备异常
if(unit_2==1)                              //如果 2#设备有警报
    UARTSendByte(table10);                 //发送 2#设备异常
}
if(j==0x99)trans(bm);
    log=0;                                //清除接收到数据的标志
    cont=1;                                //改 max491 为接收状态
}
```