

中文摘要

摘要：电网接地电阻值是发电场、变电站等地方的接地系统最主要的技术指标。接地电阻值是否正常直接影响到电力系统能否安全运行。但是由于接地电阻的设计值与实际值常常存在一定误差，所以接地电阻的测量成为电力工作者长期研究的问题之一。为了解决这个问题，研究者提出了多种测试方法并根据其中一些方法研制出了测试设备。当前，研制出能够简便、准确地测量接地网的真实接地电阻值的测试仪是各个电力公司迫切希望并共同努力的方向。

本文首先阐述了电力系统接地电阻测试的意义及其发展背景，列举了不同历史时期出现的各种接地电阻测试理论、方法及相关测试装置。

其次，本文分析了异频测量法和四极法布线方式。利用这两种方法可以有效地降低工频干扰和引线干扰的影响，使测量更加准确。

再次，也是本论文的主要创新之处，在于将嵌入式技术及嵌入式操作系统引入到仪器的开发中来，研制出新型的基于 ARM 芯片和 μ C/OS II 操作系统技术的测试仪。嵌入式系统的引入，主要作用是提高了系统的可靠性，增强了系统功能，并且大大减小了仪器的重量和体积，从而降低了劳动强度提高了工作效率，使得员工到现场测试时携带与操作都更加便捷。另外，也为以后的功能扩展奠定了基础。

本测量装置针对测量方法及功能需求，硬件采用了基于 ARM 芯片和 DSP 芯片的双核心系统。论文的第三章对系统的硬件结构作了介绍，并对主要的芯片和外围设备进行了说明。两块核心芯片各自独立运行在两块不同的电路板上，两者通过双口 RAM 进行数据交换。

本论文的主要工作在软件的设计与实现上。包括第四章 μ C/OS II 操作系统的移植与裁剪工作，和五、六章应用软件的设计与实现工作。其中应用软件的开发是采用功能模块化的设计思路。第五章为 ARM 部分，主要用来实现控制功能，主程序流程按照功能模块的运行顺序设计。第六章为 DSP 部分，用来实现数据处理功能，程序完成了对数据的各种操作功能。

最后，描述了模拟实验的过程，并对测量数据进行了分析，初步验证了测试及各项功能的可行性。结论部分对研究工作做了总结并讨论了仪器尚待解决的问题和改进意见。

关键词：接地电阻；ARM；DSP；嵌入式系统； μ C/OS II

ABSTRACT

ABSTRACT: grid grounding resistance value is the most important technical parameter of the grounding system in power plant、transformer substation and some other places. Whether right the Ground Resistance value is directly impact on whether safely work the power system can. However, because there are some errors between the design value of ground resistance and the actual value of it, the measurement of grounding resistance become into one of the issues which electricity workers long-term research in. To solve this problem, the researchers gave a variety of testing methods and base on some of these methods developed a variety of measurement equipment. At present, developing an equipment which can easily and accurately measure the true value of the grid grounding resistance have become the goal that companies are eager to achieve and work hard for.

Firstly this paper describe the meaning of grounding resistance measure and development background, enumerate the various ground resistance testing theories, methods and related measurement equipment in different historical periods.

Secondly, this paper analyze the different frequency measurement and four pole way of wiring which can reduce the interference.

Thirdly, according to the measuring method and functional requirements, the measuring equipment adopt dual-core system that base on ARM chip and DSP chip. They take through the dual-port RAM for data exchange. Chapter III in paper, the hardware structure of the system are described, and the main chip and peripheral equipments are introduced.

The focus of this paper is the design of software equipment. Transplant μ C/OS II embedded real-time operating system on the ARM core chip. Chapter V for the ARM part, main program are designed in accordance with the function modular. Chapter VI as part of DSP, firstly analyze the data-processing algorithm, followed with a detailed description of the various functions of the implementation process.

KEYWORDS: Grounding resistance; ARM; DSP; Embedded system; μ C/OS II

学位论文版权使用授权书

本学位论文作者完全了解北京交通大学有关保留、使用学位论文的规定。特授权北京交通大学可以将学位论文的全部或部分内容编入有关数据库进行检索，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。

(保密的学位论文在解密后适用本授权说明)

学位论文作者签名: 贾巍


导师签名: 陈春

签字日期: 2008年6月11日

签字日期: 2008年6月11日

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京交通大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：  签字日期： 2008 年 6 月 11 日

致谢

本论文的工作是在我的导师陈后金教授的悉心指导下完成的，陈后金教授严谨的治学态度和科学的工作方法给了我极大的帮助和影响。两年来，陈后金教授不仅在学业上给我以精心指导，同时还在思想上、生活上我以无微不至的关怀，在思想道德情操和为人处事上给我树立了光辉的榜样。从课题的选择到项目的最终完成，陈后金教授都始终给予我细心的指导和不懈的支持。在此谨向陈后金老师致以诚挚的感谢和崇高的敬意。

科学院的陶炜工程师和谷瑞英工程师指导我们完成了科学院的实习工作，在学习上和生活中都给予了我很大的关心和帮助，在此向陶炜工程师表示衷心的感谢。

电路与系统实验室的陈后金、侯建军、杜普选、刘颖、路勇、郝晓丽、薛健老师在我学习期间给予我的热心帮助，对我在实验室的学习和科研给予了极大的支持，他们孜孜不倦、尽心尽力的敬业精神时刻影响着我，在此表示衷心的感谢。

在实验室工作及撰写论文期间，李志林、张颖等同学对我论文的研究工作给予了热情帮助，在此向他们表达我的感激之情。

另外也感谢我的父母家人，他们的理解和支持使我能够在学校专心完成我的学业。

最后祝所有给予我帮助的亲人、老师、朋友和同学一切顺利、幸福健康。

1 绪论

1.1 本课题的研究对象及其意义

接地电阻是接地体或自然接地体的对地电阻和接地线电阻的总和，其数值等于接地装置对地电压与通过接地体流入地中电流的比值，它与土壤的特性及接地极的几何尺寸等因素有关。接地电阻值是衡量接地系统的有效性、安全性以及鉴定接地系统是否符合规程要求的重要指标。^[1]

接地电网的目的是为了在电力系统正常工作时，保证中性点电位不发生偏移；故障时，保证人身和设备的安全。接地电网，特别是大型发电站的主接地电网在保证电力设备的安全工作和人身安全方面起着重要作用。由于接地装置的缺陷而造成电力设备损坏，就酿成过重大事故，造成十分严重的经济损失，此种沉痛教训已有多次。例如：华中电网曾因接地装置的缺陷，发生几起高压窜入二次系统及操作系统，造成继电保护失效、变电站主要设备损坏、供电全停的重大事故。东北电网也曾发生类似的事故，500kV 董家变电站在调试过程中，因主接地电网电位升高，使继电保护误动；魏家变电站 220kV 侧接地故障，高压窜入电缆沟，使操作系统失灵，造成全站停电的事故等。由于接地网缺陷而酿成重大事故的教训，已引起人们对发电站接地网的高度重视。东北电业管理局、湖北电力局曾下达文件《关于认真做好接地网检查工作的通知》，指示局直属单位应迅速行动起来，彻底检查接地网及接地引下线，不合格的要列入计划尽早改善。

一般地网合格与否的主要判据是其接地电阻值。由于土壤电阻率的不均匀或土壤电阻率测量值的不准确，特别是季节系数和利用系数的取值，以及计算公式中的系数取值等的影响，使电网的接地电阻的设计值只能起到参考作用。施工完成后，还必须通过实测来确定其真实值。目前，设计接地系统时，经常采用等值电阻率来等效实际复杂的土壤结构，进行地网接地电阻的估算，其结果往往是设计值与实际值相差较大，在这种情况下更有必要对接地系统的接地电阻进行测量。为了检查运行中接地网的状况，有关技术规程也规定了其测试周期。接地网接地电阻的测量，是接地网验收和运行中检查其合格与否的重要手段，也是检验接地网在电力系统发生故障时，能否发挥其保证安全作用的重要措施，已成为保障电力系统安全运行的一项重要工作。

电网接地电阻测试仪是测量接地电阻的常用仪表，也是电气安全检查与接地工程竣工验收不可缺少的工具，九十年代开始，计算机技术有了飞速发展，接地

电阻测试仪也采用了大量的微处理机技术,其测量功能与精度是一般传统的测试仪器所不能比拟的。目前先进的接地电阻测试仪能满足所有接地测量要求。除了应用在发电厂、变电站等电力场所的测试外,接地电阻测试仪还可广泛应用于电力电信系统,建筑大楼,机场,铁路,油槽,避雷装置,高压铁塔等地方的接地电阻测试。

近年来随着嵌入式技术飞速发展,嵌入式控制器的应用几乎无处不在;移动电话、家用电器、汽车等等全都应用了嵌入式技术。嵌入控制器因其体积小、可靠性高、功能强、灵活方便等许多优点,其应用已深入到工业、农业、教育、国防、科研以及日常生活等各个领域,对各行各业的技术改造、产品更新换代、加速自动化进程、提高生产率等方面起到了极其重要的推动作用。

由于传统的电网接地电阻测量方式测量起来很麻烦,误差较大而且较不稳定,员工在进行测量时劳动强度非常大。所以将先进的嵌入式技术引入到接地电阻的测量中来,以提高了测试信号的精确度,增强测试系统的稳定性与可靠性,并且能够很大程度上减小测试装置的体积,方便了员工的测量,大大降低了测试的劳动强度。目前,这种新型的基于嵌入式系统的接地电阻测试仪的研发已成为国内外电力系统的研究员们极为感兴趣的一个新课题。

1.2 本课题的研究背景及现状分析

根据接地电阻的定义,最初人们对接地电阻的测量是使用伏安两点法(即用电流表、电压表测定),试验非常原始,就是简单的在距离地网较远的位置布置一个电极,直接将辅助电极与地网两端电压和电流的比值作为地网接地电阻值。在测定电阻时须先估计电流的大小,选出适当截面的绝缘导线,在预备试验时可利用可变电阻 R 调整电流,当正式测定时,则将可变电阻短路,由安培计和伏特计测量所得的数值可以算出接地电阻值。伏安法测量接地电阻有明显不足之处:麻烦、繁琐、工作量大,而且没有考虑引入辅助电极对地面电位分布的影响。另外受外界干扰影响极大,在强电压区域内有时简直无法测量。从工程使用效果来看,这种方法测量误差很大,测量可靠性和重复性都非常差。目前在发、变电站地网的测量中基本不使用这种方法。

随着电力技术的不断发展,对接地电阻的理解和分析也愈加深入,从六十年代开始,电位降测量理论逐渐得到了广泛的认可。时至今日,ANSI/IEEE 标准仍然推荐使用电位降法来测量接地电阻,也就是布置电流极和电压极两个辅助电极,通过不断改变电压极位置测量得到接地体与电压极之间的电位降曲线,分析测量曲线,得到接地电阻的测量值。^[2, 3, 4]但由于电位降测量法需要反复多次测量,工

作量很大，而电位降曲线的绘制相对也比较困难，很不利于现场操作。因此，国内外研究人员在电位降理论上发展了一系列简化测量理论，其中最常采用的是由电位降法衍生出的三极电位补偿法，包括我国电力部门过去以及目前仍大量使用的 0.618 法和 30 度夹角法^[4]。

为了消除干扰影响，研究人员在三极补偿法基础上发展出了四极法、倒相法、大电流法等测量方法。四极法^[5]即在电位降法的基础上再增加一个电极，在被测电极附近地中插入一个辅助电压极的测量方法。倒相法即通过改变电流极性进行两次测量来抑制干扰。大电流法即测试电流不小于 30A，通过注入大测试电流提高信噪比，增加测量准确性。从消除电流、电压引线间互感影响的角度来看，四极法是一种比较理想的方法，能有效地消除电压测量线上的互感影响，并且通过倒相消除地中干扰电流的影响，从而得到真实的接地电阻值。应用四极法测量接地电阻也存在一些问题，例如测量时间受到线路停电时间的限制；对于大型地网，现场测量实现起来比较困难^[6]；在土壤电阻率比较高的变电站很难达到所需的测量电流等。而且四极法和倒相法都需要分阶段测量多个电压值和多个电流值，而实际上，地中干扰信号大多数情况下是一个随机量，两次或多次测量，很难保证干扰信号的同一性，这必然影响到最后的测量值。而大电流法的使用必然导致测量电源装置的庞大和笨重，现场测量工作时使用极为不便^[7]。

为了更好的剔除干扰信号，提高接地电阻测量的准确性和可重复性，七十年代以后，国内外研究人员相继提出了基于白噪声的接地电阻测量法^[8]，基于功率谱和高阶谱的接地电阻测量方法^[9]，相位补偿法、变频法、异频法和小电流法等测量方法。其中异频法显示出突出的优越性，成为近年来接地电阻测量方法研究的主要内容之一。

测量理论的发展也促进了测量仪器的发展。根据使用的仪器不同，又产生了许多不同的测量方法，如电流表—功率表法、电桥法、电位计法等。五六十年代苏联的 E 型摇表采用补偿原理和电位计结构，携带方便，电源是手摇发电机，因此工作量比伏安法简单，取代了伏安法。但受人为因素影响大，精度低。

七十年代国产接地电阻仪问世，ZC 系列无论在结构、体积、重量、测量范围、分度值、准确性，都要胜于 E 型摇表。这种测量仪属电位计型。用磁电系检流计作指零仪，仪表备有机械整流器或相敏整流器，以便将交流转化为检流计所需的直流电源，并可消除地中工频杂散电流对测量的影响，此外在电压输出线的回路中还串联了一个电容以隔断直流。手摇发电机输出电流的频率为 105~120 Hz，而且电源容量小，不能提供大的测量电流。由于手摇发电机的关系，精度也不高。许多实践表明^[10-11]，测量值随干扰电流值的加大而明显增大，即抗干扰能力低。

八十年代数字接地电阻仪的投入使用给接地电阻测试带来了生机，一些西方

国家相继采用变频方法进行接地电阻的测量研究^[12]。虽然采用变频方法测量接地电阻在测试的接线方法同 ZC 系列没什么两样,但是由于注入的异频(频率接近工频但异与工频)电流只有几安培,因而对电流极引线的线径没有特殊要求,只需要能通过几安培的小线径导线即可。由于注入的电流的频率异与工频但接近与工频,因而仪器的抗干扰能力强、测量精度高。而且由于注入的电流很小,测量地网接地电阻的时候还不需要断开保护设备,提高了安全系数。异频接地电阻测量仪是接地电阻测量仪器的重要发展方向之一^[13]。

九十年代钳口式接地电阻仪的诞生打破了传统式测试方法。钳式接地电阻测试的最大特点是不必辅助地棒和断开接地引线,只要钳住接地线或接地棒就能测出其接地电阻,可进行在线监测。具有快速测试、操作简单等优点。钳口式接地电阻测量仪电源为干电池。主要分为单钳式(如 CA6411)和双钳式(如 GEOX 型)。测试原理是一样的,都有两个线圈,双钳口式两线圈是分开的,单钳口式把两个线圈合并在一个钳口上。单钳口式地阻仪主要用于检查在地面以上相连的多电极接地网络,通过环路地阻查询各接地电阻测量;双钳口接地电阻仪测量范围和精度均有所提高。但由于钳口法测量采用电磁感应原理,易受干扰,测量误差比较大,不能满足高精度测量要求,接地电阻在小于 0.7 欧以下,无法分辨,不适于大型接地系统。

近年来最新引进的意大利 HT 公司 234 数显精密接地电阻仪结合了传统伏安法测量的特点与钳口法新技术原理,再运用先进的计算机控制技术而成为当代首屈一指的智能型接地电阻测量仪。具有精度高,功能齐全,操作简便的特点,目前在国内邮电、电力、航空等行业都进行了配置^[14]。但是,由于这种方法测得的接地电阻包括接地体电阻和辅助极电阻,因而只有在辅助接地极的接地电阻和接地极的接地电阻相比小的多的时候,测得的结果才可以作为接地极的接地电阻^[15]。

1.3 本课题的主要工作

本课题是针对电网接地电阻测试仪器的研发。参考了传统的电网接地电阻的测试技术,运用了异频法测量原理和四极回路法布线方法相结合的测量方式,大大提高了抗工频干扰和引线干扰的能力。并且采用数字信号分析和数字显示的方法,提高了系统的可靠性和稳定性。本系统可以在 40—60Hz 范围内测量接地电阻、电抗以及土壤电阻率的值。

本课题的重点工作在仪器软件的设计上。为了实现多任务运行的功能,本系统又在基于 ARM 内核的芯片上移植了 μ C/OS II 嵌入式实时操作系统,用来对多任务进行控制管理以及对资源进行合理分配。应用软件的开发是采用功能模块化

的设计思路。

具体来说，本论文完成了以下工作内容：

(1) 系统硬件的选择与介绍

本仪器采用了双处理器构建了整个系统，选择了 PHILIPS 公司生产的基于 32 位 ARM 核心的 LPC2214 芯片为控制核心，美国 TI 公司生产的 TMS320C32 芯片为数据处理核心，建立电阻测试系统。外围主要设备包括 A/D 转换电路采集数据，LCD 液晶显示屏显示数据，串口与上位机进行数据传输。

(2) μ C/OS II 的移植和裁剪

μ C/OS II 作为一个开放源码的实时操作系统，针对具体的 LPC2214 芯片，必须按照硬件接口的要求做特定的移植工作。并且根据特定的功能进行系统的裁剪，以便更有效的利用 CPU 和内存资源。

(3) 接地电阻测试的软件设计与实现

主要包括两个方面，首先是 ARM 方面基于嵌入式操作系统的主程序及功能函数的设计与代码实现工作。其次是 DSP 数据处理方面，包括算法的选择与分析，用汇编语言实现数据处理的各功能。

(4) 实验工作

首先在实验室进行各个功能模块及整个程序功能实现的调试工作，再利用专用的接地电阻测试模拟实验箱进行模拟实验，并对实验数据进行了分析，获得了满足要求的接地电阻值，初步验证了测试的可行性。

2 本课题选用的接地电阻测试方法

接地电阻测试技术已经发展了半个世纪，现在已经提出的测试理论与测试方法种类繁多，经过对各种测试技术的比较与利弊分析，最终本课题选择了异频法测量技术与四极法布线方式相结合的方法。

测量电网接地电阻之所以困难，主要是因为测量过程中存在严重的干扰。干扰主要指的是工频地电流干扰和引线间干扰两种干扰。由于工频干扰主要是由 50Hz 的工频造成，消除工频地电流干扰最直接的方法是停电测量。但现场难以实现，目前普遍采用加大测量电流法以提高信噪比来减小测量误差。但是增大电流后，引线间的干扰也会随着增大，消除干扰的效果并不好。^[16, 17]

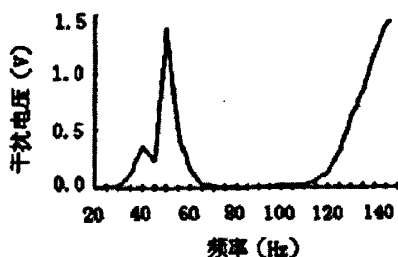


图 2.1 地网干扰电压的频率

Fig. 2.1 frequency of grid interference

voltage

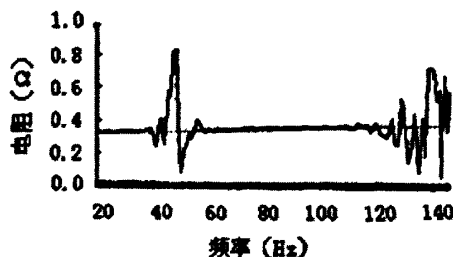


图 2.2 变频测量接地电阻值

Fig. 2.2 different frequency grounding

resistance value

图 2.1^[18]是未向地网注入电流时，地网与电压极之间测量到的干扰信号，可以发现 50Hz 和 150Hz 附近幅度最大，在 35Hz 以下和 60-120Hz 之间趋近于零，即干扰信号的主要能量集中在工频及其奇次谐波处，其他频带的干扰较小。图 2.2 中实线是地网干扰信号对接地电阻测量结果的影响，可见在 50Hz、150Hz 附近的区域干扰电压影响较大。可以设想，如果避开工频及其奇次谐波频率处的电流，在其它的频率上进行接地电阻的测量，干扰必将大大降低。

近年提出的异频法^[19]就是用非 50Hz 试验电源将干扰电流从测量用的信号电流中分离出来，使之不进入测量系统，消除其所致测量误差，因此测试电流不需太大就可提高测量精度，由于避免了大电流测量，可大大减小设备重量和试验人员的劳动强度。

2.1 异频法测量原理

如图 2.3，G 为接地装置，P，C 分别为电压极、电流极，变频电源 S 向地网

注入电流 I ，电流互感器变一次电流为二次电流 i ，测量电网电压为 u ，不计干扰电压时接地电阻 $R_G = u/i$ 。采样时中性点入地的零序电流等产生的干扰信号与有用信号叠加在一起被采集。

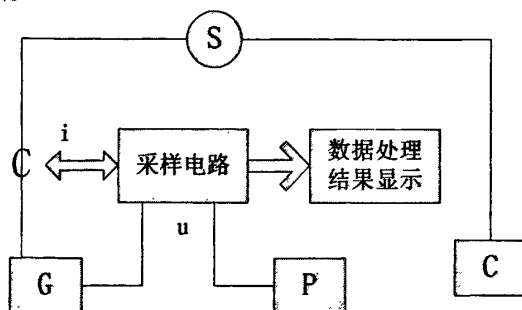


图 2.3 异频法测量原理图

Fig. 2.3 Different frequency measurement schematics

这种情况下，只有采用相位鉴别法或频率鉴别法去除干扰信号，但接地测量时，干扰信号频谱很宽，相位难以确定。而有用信号频率是已知的，所以频率识别和提取有用信号更加可行。因此，需将有用信号频率与干扰信号频率拉开一定的距离，所以需要选用变频电源作为测试信号源，可在一定频带宽度内变化，再由高精度选频滤波器过滤测量信号，去除干扰信号。

需要注意的是：频率并不能错开工频太多，往高几百赫兹，往低十几赫兹，都不允许，因为接地网电阻从根本上讲是土壤的电阻，而土壤电阻率与大地电流频率之间存在着较为复杂的关系，土壤电阻会随着频率的不同产生很大的变化。测量接地网电阻是确定工频下的电阻，为排除干扰，选择变频应尽量接近于工频。国际大电网会议 36.04.01(变电站接地)工作组(1982 年)指出，测量电流的频率与工频频率的偏差应不大于 10Hz(即测量频率应在 40-60Hz 之间)，偏差太大将产生显著误差。

异频法测量的可行性分析：

地网测试结果包括接地电阻 R 和电抗 X ；在工频 $50 \pm 10\text{Hz}$ 频率范围内，接地阻抗中的电阻分量变化很小，而电抗与频率成正比例关系。设在 $50 - \Delta f$ 和 $50 + \Delta f$ 两个频率下测量的电阻值分别为 R_1 和 R_2 ，电抗值分别为 X_1 和 X_2 ，接地阻抗为 Z_1 和 Z_2 ，则接地网接的电阻为：

$$R_0 = \frac{(R_1 + R_2)}{2} \quad (2.1)$$

在频率 50Hz 下接地网电抗为：

$$X_0 = \frac{(X_1 + X_2)}{2} \quad (2.2)$$

在频率 50Hz 下接地网接阻抗为：

$$Z_0 = \sqrt{R_0^2 + X_0^2} \quad (2.3)$$

式(2.1)和(2.2)虽然形式相同，但含义不同。式(2.1)是同一个量两次测量的平均值，而式(2.2)的含义是线性插值。

理论分析表明^[20]，在与工频对称的两个频率 $50 - \Delta f$ 和 $50 + \Delta f$ 下分别测量阻抗值，取其平均值，与在工频 50Hz 下的接地阻抗相比较，误差很小。

2.2 接地网接地电阻的测量方法

根据接地电阻的定义，当测量电流 I 注入被测地网，向无穷远处流散时，地网相对于无穷远处的电压与测试电流的比值即地网的接地电阻。然而，辅助电压极 P 不可能布置在无穷远处，无法测量地网相对于无穷远处的电位，因此定义中的电路是无法实现的，需要在有限远处布置一个辅助电流极 C ，与地网形成回路。测量的基本原理就是在较短的电流极引线的情况下，根据具体的地网形状和结构以及具体的土壤结构参数，通过计算找到电压极的正确位置。在计算求得电压极的正确位置后，在进行接地电阻的测量时，可采用异频测量法避开工频干扰，测量过程只需注入较小的电流。

实际测量中，把频率设为 45Hz 和 55Hz 来测量接地电阻。四极法布线方式如图 2.4^[21]

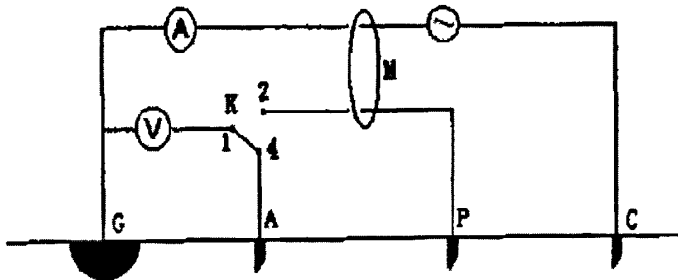


图 2.4 四极法布线图

Fig. 2.4 four poles wiring

所谓四极法布线方式，就是用四个测量电极，图中 G 为接地极，最远处的 C 为电流极，较远处的 P 为电压极，在被测电极 G 附近地中再插入一个辅助电压极 A 。当在接地极 G 和电流极 C 间施加电流 I 时，可以测出 G 、 P 、 A 各点间的电位差 U_{GP} 、 U_{GA} 和 U_{PA} 。

采用四极法布线方式，避免了需要在多点进行测量所造成的不便和误差，使得测量更加的简单，测量值更加精确。

3 系统硬件结构的介绍

3.1 嵌入式系统介绍

3.1.1 嵌入式系统的发展

根据 IEEE（国际电机工程师协会）的定义，嵌入式系统是“控制、监视或者辅助装置、机器和设备运行的装置”（原文为的 devices used to control, monitor, or assist the operation of equipment, machinery or plants）。这主要是从应用上加以定义的，从中可以看出嵌入式系统是软件和硬件的综合体，还可以涵盖机械等附属装置。

按照历史性、本质性、普遍性要求，嵌入式系统应定义为：“嵌入到对象体系中的专用计算机系统”。“嵌入性”、“专用性”与“计算机系统”是嵌入式系统的三个基本要素。

与通用型计算机系统相比，嵌入式系统面向特定应用，可根据需要灵活定制，具有功耗低、可靠性高、功能强大、性价比高、实时性强、支持实时性占用空间小、效率高的优点。

综观嵌入式技术的发展，大致经历了以下 4 个阶段。

第一阶段是以单芯片为核心的可编程控制器形式的系统，同时具有与监测、伺服、指示设备相配合的功能。这种系统大部分应用于一些专业性极强的工业控制系统中，一般没有操作系统的支持，通过汇编语言编程对系统进行直接控制，运行结束后清除内存。这一阶段系统的主要特点是：系统结构和功能都相对单一，处理效率较低，存储容量较小，几乎没有用户接口。由于这种嵌入式系统使用简便、价格很低，以前在国内工业领域应用较为普遍，但是已经远远不能适应高效的、需要大容量存储介质的现代化工业控制和新兴的信息家电等领域的需求。

第二阶段是以嵌入式 CPU 为基础、以简单操作系统为核心的嵌入式系统。这一阶段系统的主要特点是：CPU 种类繁多，通用性比较弱；系统开销小，效率高；操作系统具有一定的兼容性和扩展性；应用软件较专业，用户界面不够友好；系统主要用来控制系统负载以及监控应用程序运行。

第三阶段是以嵌入式操作系统为标志的嵌入式系统。这一阶段系统的主要特点是：嵌入式操作系统能运行于各种不同类型的微处理器上，兼容性好；操作系统内核精小、效率高，并且具有高度的模块化和扩展性；具备文件和目录管理、

设备支持、多任务、网络支持、图形窗口以及用户界面等功能；具有大量的应用程序接口（API），开发应用程序简单；嵌入式应用软件丰富。

第四阶段是以基于 Internet 为标志的嵌入式系统，这是一个正在迅速发展的阶段。目前大多数嵌入式系统还孤立于 Internet 之外，但随着 Internet 的发展以及 Internet 技术与信息家电、工业控制技术等结合日益密切，嵌入式设备与 Internet 的结合将代表着嵌入式技术的真正未来。

现在，以计算机和软件为核心的数字化技术取得了迅猛发展，不仅广泛渗透到社会经济、军事、交通、通信等相关行业，而且深入到家电、娱乐、艺术、社会文化等各个领域，掀起了一场数字化技术革命。多媒体技术与 Internet 的应用迅速普及，消费电子(Consumptive electron)，算机(Computer)，通信(Communication)，3C 一体化趋势日趋明显，嵌入式技术再度成为一个研究热点。

3.1.2 嵌入式系统的结构与特点

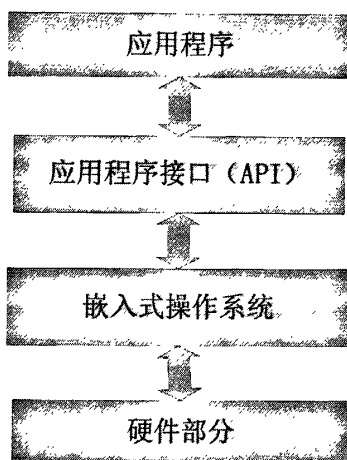


图 3.1 嵌入式系统结构框图

Fig. 3.1 Embedded System structure diagram

从图 3.1 可以看出，嵌入式系统的结构主要包括硬件电路部分和软件部分，其中软件部分又包括嵌入式操作系统部分、API 和应用程序部分。

一般的嵌入式系统从硬件上都包含以下几个部分：嵌入式微控制器、存储系统、外围电路和电源部分。如图所示：

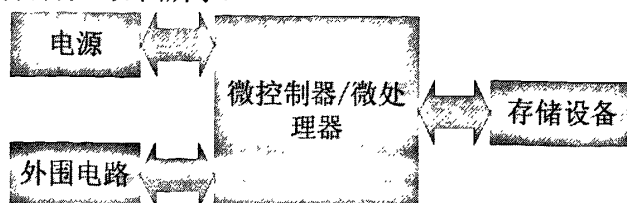


图 3.2 嵌入式系统硬件结构框图

Fig. 3.2 Embedded System hardware structure diagram

嵌入式处理核心可以分成下面几类：嵌入式微处理器(Embedded Micro-processor Unit, EMPU)，嵌入式微控制器(Microcontroller Unit, MCU)，嵌入式 DSP 处理器(Embedded Digital Signal Processor, EDSP)，嵌入式片上系统(System On Chip, SOC)。本课题主要用到了 ARM 控制器和 DSP 处理器。下面重点介绍下 ARM 控制器和 DSP 处理器芯片的特性及应用。

嵌入式微控制器又称单片机，顾名思义，就是将整个计算机系统集成到一块芯片中。嵌入式微控制器一般以某一种微处理器内核为核心，芯片内部集成 ROM/EPROM、RAM、总线、总线逻辑、定时/计数器、WatchDog、I/O、串行口、脉宽调制输出、A/D、D/A、Flash RAM、EEPROM 等各种必要功能和外设。为适应不同的应用需求，一般一个系列的单片机具有多种衍生产品，每种衍生产品的处理器内核都是一样的，不同的是存储器和外设的配置及封装。这样可以使单片机最大限度地和应用需求相匹配，功能不多不少，从而减少功耗和成本。和嵌入式微处理器相比，微控制器的最大特点是单片化，体积大大减小，从而使功耗和成本下降、可靠性提高^[22]。

目前常用的嵌入式微控制器体系结构有 ARM、PowerPC、X86、MIPS 等等，这些体系的微控制器均基于 RISC 结构。其中 ARM 体系结构在过去的十多年中已成为世界上最受欢迎的 32 微处理器体系结构，市场份额占有所有微控制器的 75%以上。例如飞利浦公司的基于 ARM 的 ARM7TDMI 系列芯片作为微控制器核心在国内自动控制等领域得到了广泛的应用。

DSP 处理器芯片，也称数字信号处理器，是一种具有特殊结构的微处理器。DSP 芯片的内部采用程序和数据分开的哈佛结构，具有专门的硬件乘法器，广泛采用流水线操作，提供特殊的 DSP 指令，可以用来快速的实现各种数字信号处理算法。

根据数字信号处理的要求，DSP 芯片一般具有如下的一些主要特点^[23]：

- 在一个指令周期内可完成一次乘法和一次加法。
- 程序和数据空间分开，可以同时访问指令和数据。
- 片内具有快速 RAM，通常可通过独立的数据总线在两块中同时访问。
- 具有低开销或无开销循环及跳转的硬件支持。
- 快速的中断处理和硬件 I/O 支持。
- 具有在单周期内操作的多个硬件地址产生器。
- 可以并行执行多个操作。
- 支持流水线操作，使取指、译码和执行等操作可以重叠执行。

与通用微处理器相比，DSP 芯片的其他通用功能相对较弱些。

目前世界上生产 DSP 芯片的主要厂家有 TI、AD、Motorola、NEC 等，其中最成功当数美国德州仪器公司 (TI)。如今 TI 公司的一系列 DSP 产品已经成为当今世界上最有影响的 DSP 芯片。

3.2 系统硬件结构与设备选择

本测试仪器主要采用两块 CUP 分别完成数据处理功能和控制功能，数据处理所用的 DSP 芯片为 TI 公司生产的 TMS320LF2407 芯片，控制所用的控制器为 Philips 公司生产的基于 ARM 内核的 32 位 LPC2214 微处理器。仪器的硬件结构框图如图 3.3 所示。

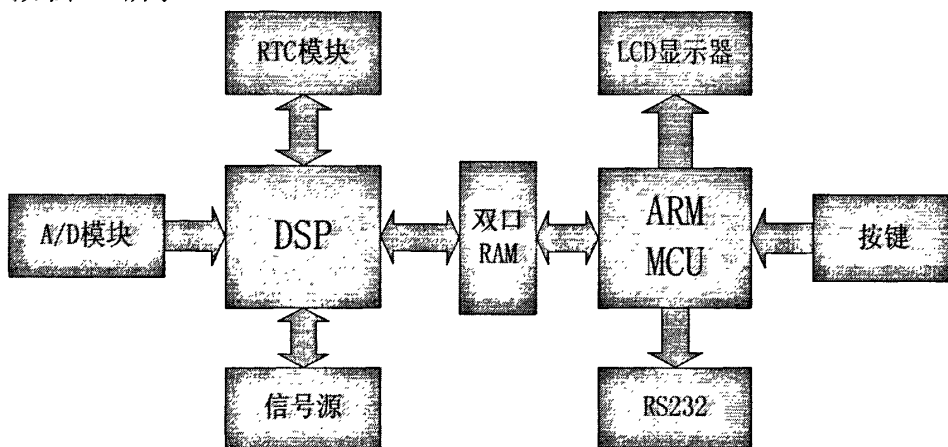


图 3.3 系统硬件结构框图

Fig. 3.3 System hardware structure diagram

3.2.1 主控制器的选择

本仪器选用的是 ARM7TDMI-S 处理器，它是一款通用的 32 位微处理器，具有高性能和低功耗的特点。ARM 结构是基于精简指令集计算机(RISC)原理而设计的。指令集和相关的译码机制比复杂指令集计算机要简单得多。这样使用一个小的、廉价的处理器核就可实现很高的指令吞吐量和实时的中断响应。

由于使用了流水线技术，处理和存储系统的所有部分都可连续工作。通常在执行一条指令的同时对下一条指令进行译码，并将第三条指令从存储器中取出。

ARM7TDMI-S 处理器使用了一个被称为 THUMB 的独特的结构化策略，它非常适用于那些对存储器有限制或者需要较高代码密度的大批量产品的应用。关于

THUMB 一个关键的概念是“超精简指令集”。ARM7TDMI-S 处理器基本上具有两个指令集：

标准 32 位 ARM 指令集

16 位 THUMB 指令集

THUMB 指令集的 16 位指令长度使其可以达到标准 ARM 代码两倍的密度，却仍然保持 ARM 的大多数性能上的优势，这些优势是使用 16 位寄存器的 16 位处理器所不具有的。这是因为 THUMB 代码和 ARM 代码一样，在相同的 32 位寄存器上进行操作。

THUMB 代码仅为 ARM 代码规模的 65%，但其性能却相当于连接到 16 位存储器系统的相同 ARM 处理器性能的 160%。

嵌入式系统的核心是嵌入式微处理器，本系统采用的是 Philips 公司的 ARM 芯片 LPC2214，LPC2114 基于一个支持实时仿真和跟踪的 16/32 位 ARM7TDMI-S CPU，并带有 128/256 k 字节(kB)嵌入的高速 Flash 存储器。128 位宽度的存储器接口和独特的加速结构使 32 位代码能够在最大时钟速率下运行。对代码规模有严格控制的应用可使用 16 位 Thumb 模式将代码规模降低超过 30%，而性能的损失却很小。

由于 LPC2114 非常小的 64 脚封装、极低的功耗、多个 32 位定时器、4 路 10 位 ADC、PWM 输出、46 个 GPIO 以及多达 9 个外部中断使它们特别适用于工业控制、医疗系统、访问控制和电子收款机 (POS)。由于内置了宽范围的串行通信接口，它们也非常适合于通信网关、协议转换器、嵌入式软件调制解调器以及其它各种类型的应用。

此款芯片的主要特性如下：

- 16/32 位 ARM7TDMI-S 核，超小 LQFP64 封装。
- 16 kB 片内静态 RAM。
- 128/256 kB 片内 Flash 程序存储器，128 位宽度接口/加速器可实现高达 60 MHz 工作频率。
- 通过片内 boot 装载程序实现在系统编程 (ISP) 和在应用编程 (IAP)。512 字节行编程时间为 1ms。单扇区或整片擦除时间为 400ms。
- Embedded ICE 可实现断点和观察点。当使用片内 RealMonitor 软件对前台任务进行调试时，中断服务程序可继续运行。
- 嵌入式跟踪宏单元 (ETM) 支持对执行代码进行无干扰的高速实时跟踪
- 4 路 10 位 A/D 转换器，转换时间低至 2.44 μ S。
- 2 个 32 位定时器 (带 4 路捕获和 4 路比较通道)、PWM 单元 (6 路输出)、实时时钟和看门狗。

- 多个串行接口，包括 2 个 16C550 工业标准 UART、高速 I2C 接口（400 kbit/s）和 2 个 SPI 接口。
- 通过片内锁相环（PLL）可实现最大为 60MHz 的 CPU 操作频率。
- 向量中断控制器。可配置优先级和向量地址。多达 46 个通用 I/O 口（可承受 5V 电压），9 个边沿或电平触发的外部中断引脚。
- 片内晶振频率范围：1~30 MHz。
- 2 个低功耗模式：空闲和掉电。
 - 通过外部中断将处理器从掉电模式中唤醒。
 - 可通过个别使能/禁止外部功能来优化功耗。
- 双电源
 - CPU 操作电压范围：1.65~1.95 V(1.8 V \pm 0.15 V)
 - I/O 操作电压范围：3.0~3.6 V(3.0 V \pm 10%)，可承受 5V 电压。

3.2.2 数据处理器的选择

本仪器选用的 DSP 芯片是美国 TI 公司生产的 TMS320C32，此款芯片具有强大的数据处理功能，尤其在电力系统中有着广泛的用途。

TMS320C32 是 TMS320 系列 DSP(数字信号处理器)的第三代产。它在保证 TMS320C30 基本性能不降低的基础上做了一些简化及扩充，价格比 TMS320C30 低得多，因而在实时数据采集与信号处理系统中得到广泛应用。

TMS320C32DSP 的特点：^[24]

- 超高速的处理能力：60MFLOPS，30MPIS；
- 512*32 位单周期双获取片上 RAM；
- 64*32 位指令缓存；
- 32 位指令和数据，24 位地址；
- 16/32 位定点运算，32/40 位浮点运算；
- 8 个扩展精度辅助寄存器；
- 增强外部存储器接口；
- 支持 8/16/32 外部数据访问；
- 支持 16/32 位外部程序运行；
- 一个外部引脚 PRGW，配置外部存储器宽度 16 位或 32 位；
- 两组外部存储器选通信号(STRB0, STRB1)，一组 I/O 选通信号(IOSTRB)，简化接口逻辑；每组选通信号各有一个独立的总线控制寄存器，控制它所访问的外部存储器的宽度，数据位数和软件插入等待的个数；

- 单周期累加和乘法并行指令；
- 块重复指令；
- 四级外部中断；
- TMS320C32 存储器映射；

TMS320C32CPU 为 C3XCPU 的增强版,增强部分包括:可变宽度存储器接口、更快的指令周期、节电模式、可重定位的向量表、外部中断可由沿或电平触发;

采用 TMS320C32 芯片作为数据处理芯片,主要考虑仪器在进行数据处理时需要大量使用数字滤波技术,运算量大,一般 CPU 难以满足其要求。但 TMS320C32 芯片的控制引脚较少,难以满足液晶显示等方面的控制要求,因此本文选择了一块 LPC2214 微处理器作为控制核心,两者通过双口 RAM 交换数据。

3.2.3 其它设备的选择

3.2.3.1 A/D 采样电路

AD7678 是一款 18 位,100kSPS,电荷分配的完全差分逐次逼近型模数转换器。它由单一的 5V 电源供电,内含一个高速 18 位采样 ADC,一个内部转换时钟,一个内部参考缓存,纠错电路以及串行和并行系统接口。是一款速度快,功耗低,精度高的逐次逼近结构的模数转换器。AD7678 的线性度和动态范围类似于或优于很多的 Δ - Σ 型 ADC。由于逐次逼近结构易于多路技术和可用低电源等优点,所以它在应用上相比一般的 Δ - Σ 型的 ADC 有很多优势。AD7678 逐次逼近型 ADC 由于不存在传输延迟,所以是用于多通道技术的理想 ADC 器件。AD7678 可以由 5V 的单电源供电,同时可以与 5V 或 3V 的数字逻辑电平兼容。它的 48 脚 LQFP 封装和 48 脚 LFCSP 封装既节省了空间又可以提供串行和并行两种接口,且其配置更加灵活,AD7678 与 AD7674, AD7676, AD7679 引脚兼容。工作温度为 $-40\sim+85^{\circ}\text{C}$ 。

3.2.3.2 LCD 液晶显示屏

本仪器采用的是 240128A 液晶显示器模块。240128A 是一种点阵 LCD 模块,它能显示 $240*128$ 的点阵。它可用于显示字符,图形和字符-图形,包括了多种功能。

这一模块有一个 8 位的平行数据总线和通过 MPU 接口读、写的控制线。它可以直接连接在 TMPZ-80 上。它有能控制一个外部显示 RAM 的 128 个字的字库 ROM。能很容易地分配字符,图形,或外部字符字库。

3.2.3.3 双口 RAM

双口 RAM 是常见的共享式多端口存储器,双口 RAM 最大的特点是存储数据

共享。一个存储器配备两套独立的地址、数据和控制线，允许两个独立的 CPU 或控制器同时异步地访问存储单元。既然数据共享，就必须存在访问仲裁控制。内部仲裁逻辑控制提供以下功能：对同一地址单元访问的时序控制；存储单元数据块的访问权限分配；信令交换逻辑（例如中断信号）等。

3.2.3.4 RTC 芯片

本仪器采用的是 DS1742W-100 型号的 RTC 芯片，主要利用其时钟/日历功能为仪器提供时钟，并利用其 NVRAM 作为保存历史记录存储器。

该 DS1742W-100 芯片是一款全功能的时钟/日历芯片，它具有实时时钟/日历（已解决千年虫问题）的功能，和 2k* 8bit 的非易失性静态 RAM。实时时钟 (RTC) 的信息和控制位定位在 RAM 的最高八位地址上。实时时钟寄存器包含世纪、年、月、日、小时、分钟和秒的数据，这些数据以 BCD 码的形式 24 小时保存在寄存器里。RTC 芯片还能够自动更正每个月包含的不同日期数和闰年的日期。

4 操作系统的选择与移植

4.1 嵌入式操作系统的介绍

嵌入式操作系统 EOS (Embedded Operating System) 是一种用途广泛的系统软件, 过去它主要应用于工业控制和国防系统领域。EOS 负责嵌入系统的全部软、硬件资源的分配、调度工作, 控制协调并发活动; 它必须体现其所在系统的特征, 能够通过装卸某些模块来达到系统所要求的功能。目前, 已推出一些应用比较成功的 EOS 产品系列。随着 Internet 技术的发展、信息家电的普及应用及 EOS 的微型化和专业化, EOS 开始从单一的弱功能向高专业化的强功能方向发展。嵌入式操作系统在系统实时高效性、硬件的相关依赖性、软件固态化以及应用的专用性等方面具有较为突出的特点。EOS 是相对于一般操作系统而言的, 它除具备了一般操作系统最基本的功能, 如任务调度、同步机制、中断处理、文件功能等外, 还有以下特点:

- 可装卸性。开放性、可伸缩性的体系结构。
- 强实时性。EOS 实时性一般较强, 可用于各种设备控制当中。
- 统一的接口。提供各种设备驱动接口。
- 操作方便、简单、提供友好的图形 GUI, 追求易学易用。
- 提供强大的网络功能, 支持 TCP/IP 协议及其它协议, 提供 TCP/UDP/IP/PPP 协议支持及统一的 MAC 访问层接口, 为各种移动计算设备预留接口。
- 强稳定性, 弱交互性。嵌入式系统一旦开始运行就不需要用户过多的干预, 这就要负责系统管理的 EOS 具有较强的稳定性。嵌入式操作系统的用户接口一般不提供操作命令, 它通过系统调用命令向用户程序提供服务。
- 固化代码。在嵌入系统中, 嵌入式操作系统和应用软件被固化在嵌入式系统计算机的 ROM 中。辅助存储器在嵌入式系统中很少使用, 因此, 嵌入式操作系统的文件管理功能应该能够很容易地拆卸, 而用各种内存文件系统。
- 更好的硬件适应性, 也就是良好的移植性。

国际上用于信息电器的嵌入式操作系统有 40 种左右。现在, 市场上非常流行的 EOS 产品, 包括 3Com 公司下属子公司的 Palm OS, 全球占有份额达 50%, Microsoft 公司的 Windows CE 占 29%。在美国市场, Palm OS 更以 80% 的占有率远超 Windows CE。开放源代码的 Linux 很适于做信息家电的开发。

常见的嵌入式系统有：Linux、uClinux、WinCE、PalmOS、Symbian、eCos、uCOS-II、VxWorks、pSOS、Nucleus、ThreadX、Rtems、QNX、INTEGRITY、OSE、C Executive^[25]。

与其它类型的操作系统相比，嵌入式操作系统具有以下一些特点。

- 体积小。嵌入式系统有别于一般的计算机处理系统，它不具备像硬盘那样大容量的存储介质，而大多使用闪存（Flash Memory）作为存储介质。这就要求嵌入式操作系统只能运行在有限的内存中，不能使用虚拟内存，中断的使用也受到限制。因此，嵌入式操作系统必须结构紧凑，体积微小。
- 实时性。大多数嵌入式系统都是实时系统，而且多是强实时多任务系统，要求相应的嵌入式操作系统也必须是实时操作系统(RTOS)。实时操作系统作为操作系统的一个重要分支已成为研究的一个热点，主要探讨实时多任务调度算法和可调度性、死锁解除等问题。
- 特殊的开发调试环境。提供完整的集成开发环境是每一个嵌入式系统开发人员所期待的。一个完整的嵌入式系统的集成开发环境一般需要提供的工具是编译/连接器、内核调试/跟踪器和集成图形界面开发平台。其中的集成图形界面开发平台包括编辑器、调试器、软件仿真器和监视器等。

4.2 μ C/OS II 操作系统

本课题选用了源码公开且便于移植的 μ C/OS II 操作系统。

μ C/OS II 操作系统是一个源码公开的嵌入式实时操作系统，由 Jean.Labrosse 先生开发。操作系统的源代码是公开的，如果用于非商业目的，可以免费使用，而且操作系统的注释非常详细，便于学习。绝大部分 μ C/OS II 操作系统的源代码使用 ANSI C 编写，与处理器硬件相关的部分代码用汇编语言编写。用汇编语言编写的部分代码只与移植有关，已经压缩到最低限度。因此通过修改汇编语言代码和其中小部分 C 语言代码就可以把 μ C/OS II 移植到很多处理器上。

μ C/OS II 主要有以下特点^[26]：

- 公开源代码 μ C/OS II 全部的源代码都是公开的。
- 多任务 μ C/OS II 可以管理 64 个任务。
- 可移植性 μ C/OS II 的源代码绝大部分是使用移植性很强的 ANSI C 编写，与微处理硬件相关的部分是使用汇编语言编写。汇编语言写的部分已经压缩到最低的限度，以使 μ C/OS II 便于移植到其它处理器上。目前， μ C/OS II 已经被移植到多种不同架构的微处理器上。
- 占先式内核 μ C/OS II 是完全可剥夺型实时内核，总是运行就绪条件下优

优先级最高的任务。

- **可确定性** 绝大多数 μ C/OS II 的函数调用和服务的执行时间具有确定性，也就是说，用户总能知道 μ C/OS II 的函数调用与服务执行了多长时间。
- **中断管理** 中断可以使正在执行的任务暂时挂起，如果优先级更高的任务被中断唤醒，则高优先级的任务在中断嵌套全部退出后立即执行，中断嵌套层数可达 255 层。
- **任务堆栈** 每个任务有自己单独的任务堆栈。
- **系统服务** 提供多种系统服务如信箱、信号量、定时器相关功能等。
- **可固化** 只要具备适合的软硬件工具，就可以将 μ C/OS II 嵌入式到产品中作为产品的一部分。
- **可剪裁** μ C/OS II 使用条件编译实现可裁减，用户程序可以只编译自己需要的功能，而不编译不需要的功能，以减少 μ C/OS II 对代码空间和数据空间的占用。
- **稳定性与可靠性** μ C/OS II 是基于 μ C/OS 的， μ C/OS 自 1992 年以来有数百个商业应用。 μ C/OS II 与 μ C/OS 的内核一样的，只是提供了更多的功能。

另外，2000 年 7 月， μ C/OS II 在一个航空项目中得到了美国联邦航空管理局对商用飞机的、符合 RTCADO-178B 标准的认证。这一结论表明，该操作系统的质量得到了认证，可以在任何应用中使用。

由上述特点可知， μ C/OS II 是一款源代码公开、具有很好的移植性，高稳定性和可靠性，抢占式多任务的 RTOS。它已经应用于数码相机、医疗仪器、音响设备、发动机控制、网络接入设备、ATM 机、工业机器人等嵌入式实时系统。很多高等院校将 μ C/OS II 用于实时操作系统教学。它是非常有生命力和潜力的操作系统。

基于以上这些原因，本课题也选择了 μ C/OS II 操作系统作为软件开发平台。

4.3 μ C/OS II 操作系统的移植

由于 μ C/OS II 已经充分考虑了可移植性，所以它的移植工作相对来说很简单。它所需要移植的代码集中在 OS_CPU.h, OS_CPU_C.c 和 OS_CPU.asm 中。这三个文件包含与处理器相关的代码，具体需要移植的内容如表 4.1 所示^[15]：

表 4.1 μ C/OS II 需要移植的代码

Tab. 4.1 μ C/OS II codes need to transplant

移植内容	类型	所属文件	描述
BOOLEAN, INT8U, INT8S, INT16U, INT16S, INT32S, FP32, FP64	数据类型	OS_CPU.h	与编译器无关的数据类型
OS_STK	数据类型	OS_CPU.h	堆栈的数据类型
OS_ENTER_CRITICAL(), OS_EXIT_CRITICAL()	宏	OS_CPU.h	开关中断的代码
OS_STK_GROWTH	常量	OS_CPU.h	定义堆栈的增长方向
OS_TASK_SW	函数	OS_CPU.h	任务切换时执行的代码
OSTaskStkInit()	函数	OS_CPU_C.c	任务堆栈初始化函数
OSInitHookBegin() OSInitHookEnd() OSTaskCreateHook() OSTaskDelHook(), OSTaskSwHook(), OSTaskStateHook() OSTCBInitHook(), OSTimeTickHook() OSTaskIdleHook()	函数	OS_CPU_C.c	μ C/OS II 执行某些操作时调用的用户函数, 一般为空
OSStartHighRdy()	函数	OS_CPU_A.asm	进入多任务环境时运行优先级最高的任务
OSIntCtxSw()	函数	OS_CPU_A.asm	中断退出时的任务切换函数
OSTickISR ()	中断服务程序	OS_CPU_A.asm	时钟节拍中断服务程序

具体到本系统, μ C/OS II 需要移植到 LPC2214 上。在移植过程中, 主要的过程及代码参考了广州周立功的 ARM 嵌入式系统基础教程。下面具体说明移植过程所做的工作:

(1) 定义与编译器无关的数据类型

因为不同的微处理器有不同的字长, 所以 μ C/OS II 的移植包括了一系列的数据类型定义。C 语言中的 short, long, int 等数据类型与处理器类型相关, 不能在 μ C/OS II 中采用, 而代之以移植性强的整型数据类型。另外, 必须通过定义 OS_STK 来将任务堆栈的数据类型告诉操作系统, 这些数据类型的代码如下:

```
typedef unsigned char  BOOLEAN;      /* 布尔变量*/
typedef unsigned char  INT8U;        /* 无符号 8 位整型变量*/
typedef signed char    INT8S;        /* 有符号 8 位整型变量*/
```

```

typedef unsigned short INT16U;      /* 无符号 16 位整型变量*/
typedef signed short INT16S;        /* 有符号 16 位整型变量*/
typedef unsigned int INT32U;        /* 无符号 32 位整型变*/
typedef signed int INT32S;          /* 有符号 32 位整型变*/
typedef float FP32;                 /* 单精度浮点数 (32 位长度) */
typedef double FP64;                /* 双精度浮点数 (64 位长度) */
typedef INT32U OS_STK;              /* 堆栈是 32 位宽度*/

```

(2) OS_STK_GROWTH

μ C/OS II 在任务相关的函数处理时经常会用到堆栈的顶端地址或底端地址，因此必须明确堆栈的生长方式。这通过 OS_STK_GROWTH 指定，为 0 表示堆栈从下往上长，为 1 表示从上往下长，ADS 的 C 语言编译器只支持从上往下长，因此定义如下：

```
#define OS_STK_GROWTH 1 /* 堆栈是从上往下长的*/
```

(3) 使用软中断 SWI 实现系统调用

ARM 处理器共有 7 种模式，用户任务可以使用两种处理器模式：用户模式和系统模式。一般的应用程序都是工作在用户模式的，而系统模式属于特权模式，在此模式下可以访问所有的系统资源。在这里要实现的几个函数都是属于操作系统的底层接口函数，都应该工作在系统模式下，在 ADS 中通过关键字 `_swi` 就可以实现这一目的。

`_swi` 相当于一个中断指令，但它不是由外部中断源产生的，所以它也被叫做软中断。用它声明一个不存在的函数，同时指定功能号，函数像一般函数一样，也有返回值和参数，这样用户程序可以像调用一般程序一样调用系统函数。但实际上，系统函数和一般函数不一样，它只有声明没有具体实现，它是在 `swi` 异常中断处理程序中根据中断号来实现具体的功能的。下面列举几个函数的声明如下：

```

_swsi(0x00) void OS_TASK_SW(void);      /*任务切换函数*/
_swsi(0x01) void _OSStartHighRdy(void); /*运行优先级最高的任务*/
_swsi(0x02) void OS_ENTER_CRITICAL(void); /*关中断 */
_swsi(0x03) void OS_EXIT_CRITICAL(void); /*开中断 */

```

在中断处理函数 `SWI_Exception` 中，对不同的功能号做出了相应的处理，此处不一一列举。

(4) 实现堆栈的初始化函数

`OSTaskStkInit()` 要实现任务堆栈结构的初始化，必须明确本系统的堆栈结构，根据查阅的系统的堆栈结构的情况，程序代码及说明如下：

```
OS_STK *OSTaskStkInit (void (*task) (void *pd), void *pdata, OS_STK *ptos,
```

INT16U opt)/*参数说明: Pd: 指向任务代码的指针; Pdata: 指向一个数据区,用于向建立的任务传递参数; Ptos: 指向任务堆栈栈顶的指针; Opt: 存放与任务相关的操作信息;

```

{
    OS_STK *stk;
    opt = opt;          /* `opt` 没有使用 作用是避免编译器警告*/
    stk = pto;         /* 获取堆栈指针 */
                        /* 建立任务环境, ADS1.2 使用满递减堆栈*/
    *stk = (OS_STK) task;    /* pc */
    *--stk = (OS_STK) task;  /* lr */
    *--stk = 0;             /* r12 */
    *--stk = 0;             /* r11 */
    .....
    *--stk = 0;             /* r1 */
    *--stk = (unsigned int) pdata; /* r0,第一个参数使用 R0 传递 */
    *--stk = (USER_USING_MODE|0x00);/* spsr, 允许 IRQ, FIQ 中断*/
    *--stk = 0;             /* 关中断计数器 OsEnterSum;*/
    return (stk);
}

```

(5)Hook 接口函数的实现

μ C/OS II 预留了许多接口函数, 以允许用户扩展系统的功能。在本移植中, 所有的 Hook 函数都为空函数, 没有特殊实现。

(6)用汇编语言实现任务切换和时钟节拍中断服务程序

在 OS_CPU_A.ASM 中用汇编语言实现了 OSIntCtxSw 和 __OSStartHighRdy, OSIntCtxSw 在中断退出时被调用以执行任务切换功能, __OSStartHighRdy 在 μ C/OS II 启动时用来运行第一个任务。相关代码不在此列出。

4.4 μ C/OS II 操作系统的功能裁剪

系统移植完成后, 由于本仪器中使用的 DSP 和 ARM 双内核, 信号处理与存储功能都在 DSP 中实现, 而 ARM 上操作系统的主要用途是实现多任务管理和系统资源分配, 液晶驱动等功能。许多其它的功能没有用到, 所以需要对所使用的 μ C/OS II 操作系统进行功能裁剪, 关闭没有用到的功能函数, 以减少系统内存的负荷, 增强 CPU 的利用率。

μ C/OS II 的系统功能函数的配置可以在头文件 `os_cfg.h` 中实现:

创建任务最大数: 用于定义用户程序中可以使用最多任务数。它的值不能大于 62。因为 μ C/OS II 保留了 2 个系统任务(空闲任务和统计任务)。本次课题建立了 5 个任务,但如果把创建任务最大数设定为 7,则增加任务同时也要修改最大任务数值,这样不利于日后扩展,如果设定的任务数远远大于当前任务数的值,则会大大浪费内存空间。综合考虑以上因素,本次最大任务数设定值为 11。代码如下:

```
#define OS_MAX_TASKS      11
```

最低优先级: 用于设定系统中要使用的最低优先级。设定该项可以节省 μ C/OS II 所需要的 RAM 空间。它的值为一个小于 63 的数值。由于系统中保留了 2 个优先级自用: `OS_LOWEST_PRIO` 和 `OS_LOWEST_PRIO 1`。分别分配给空闲任务和统计任务。所以它必须大于系统中的实际任务数。为便于日后扩展,设置为 20。代码如下:

```
#define OS_LOWEST_PRIO    20
```

空闲任务堆栈的容量: 用于设定 μ C/OS II 中空闲任务堆栈的容量。这里堆栈容量单位不是字节,而是 `OS_STK` 设定的 32 位(字)。空闲任务堆栈的容量取决于所使用的处理器,以及预期的最大中断嵌套数。代码如下

```
#define OS_TASK_IDLE_STK_SIZE  512
```

每秒节拍数: 用于设定调用 `OSTimeTickO` 函数的频率。一般设定的节拍数为 10~200Hz。本次设定为 200。

```
#define OS_TICKS_PER_SEC      200
```

CPU 句柄: 用于设定是否在 `OS_CPU_C.C` 中声明对外接口函数。因为移植中设定了这些接口函数,所以该项设置为 1。代码如下:

```
#define OS_CPU_HOOKS_EN      1
```

参数检测: 用于设定是否希望大多数 μ C/OS II 中的函数执行参数检查的功能。它会尽量减少 μ C/OS II 所需要的代码量和处理时间。若必须将代码量降至最低,可将它设置为 0。但是本次设置参考一般的设计,允许参数检查。设置为 1。代码如下:

```
#define OS_ARG_CHK_EN        1
```

事件标志设置: `OS_FLAG_EN` 用于控制是否使用事件标志函数及其相关函数。由于本次任务中没有使用到事件标志,所以关闭此函数。只需要把事件标志的开关量 `OS_FLAG_EN` 的值设定为 0,而其它与消息邮箱有关的功能函数的宏定义可以任意设置为 1 或 0 都不会起作用。代码如下:

```
#define OS_FLAG_EN          0
```

消息邮箱设置: OS_MBOX_En 用来控制是否使用 μ C/OS II 中的消息邮箱。本次任务中没有用到消息邮箱功能, 所以关闭此函数。只需要把消息邮箱的开关量 OS_MBOX_EN 的值设定为 0, 而其它与消息邮箱有关的功能函数的宏定义可以任意设置为 1 或 0 都不会起作用。代码设置如下:

```
#define OS_MBOX_EN 0
```

内存管理: OS_MEM_QUERY_EN 用于系统内存查询, 由于本任务中使用的是 DSP 芯片上的 RAM 以及外接了 CF 卡进行存储, 并没有用到 ARM 的存储, 所以关闭此函数。只需要把内存管理的开关量 OS_MEM_EN 的值设定为 0, 而其它与消息邮箱有关的功能函数的宏定义可以任意设置为 1 或 0 都不会起作用。代码如下:

```
#define OS_MEM_EN 0
```

互斥型信号量: OS_MUTEX_EN 控制是否使用互斥型信号量函数及相关的数据结构, 本次任务中没有用到互斥型信号量, 所以关闭此函数。只需要把互斥型信号量的开关量 OS_MUTEX_EN 的值设定为 0, 而其它与消息邮箱有关的功能函数的宏定义可以任意设置为 1 或 0 都不会起作用。代码如下:

```
#define OS_MUTEX_EN 0
```

消息队列: 由于本次任务用到了消息队列及其相关函数, 所以需要将消息队列 OS_Q_EN 的开关量函数值设置为 1, 并把相应的消息队列函数都设置为有效。代码如下:

```
#define OS_Q_EN 1
#define OS_Q_ACCEPT_EN 1
#define OS_Q_DEL_EN 1
#define OS_Q_FLUSH_EN 1
#define OS_Q_POST_EN 1
#define OS_Q_POST_FRONT_EN 1
#define OS_Q_POST_OPT_EN 1
#define OS_Q_QUERY_EN 1
```

信号量: OS_SEM_EN 控制是否使用信号量及其相关的函数。本次任务中虽然没有用到信号量, 但是考虑到本仪器在下一不的升级过程中需要实现远程控制及远程数据传输功能, 需要用到网口传输代替串口传输, 在网口传输中使用 TCP/IP 协议时需要用到信号量功能, 所以在此将信号量开关函数设置为 1, 并将其相关功能函数设置为有效, 以方便下一步仪器的升级。代码如下:

```
#define OS_SEM_EN 1
#define OS_SEM_ACCEPT_EN 1
```

```
#define OS_SEM_DEL_EN      1
#define OS_SEM_QUERY_EN   1
```

任务管理：包括了六个与任务管理相关的函数。

OS_TASK_CHANGE_PRIO_EN 控制是否需要改变任务的优先级。本次任务中有可能需要改变运行任务的优先级，所以将此开关量设置为 1。

OS_TASK_CREATE_EN 控制是否需要创建任务。使用该函数创建的任务可以与 μ C/OS 创建的任务函数兼容。本次任务只使用了 OSTaskCreateExt()函数没有使用 OSTaskCreate()函数。所以将此开关量设为 0。

OS_TASK_CREATE_EXT_EN 用来控制系统中是否需要创建任务，它控制 OSTaskcreateExtO 函数，与 OSTaskCreate()函数不同，次函数为可扩展的、功能更强的新创建任务函数。本次任务中建立任务使用的都是该函数。所以将开关量设为 1。

OS_TASK_DEL_EN 控制系统中是否需要删除任务的函数。本次任务中有可能用到次功能，所以将开关量设为 1。

OS_TASK_SUSPEND_EN 用来控制系统中的任务挂起函数 OSTaskSuspend()和任务重新启动函数 OSTaskResume()。本次任务中用到了这两个功能。所以将开关量设置为 1。

OS_TASK_QUERY_EN 用来控制任务询问函数 OSTaskQuery()。本次任务中有可能用到此函数，所以将其开关量设置为 1。

代码如下：

```
#define OS_TASK_CHANGE_PRIO_EN   1
#define OS_TASK_CREATE_EN        0
#define OS_TASK_CREATE_EXT_EN    1
#define OS_TASK_DEL_EN           1
#define OS_TASK_SUSPEND_EN       1
#define OS_TASK_QUERY_EN         1
```

时钟管理函数：其中时钟延时控制开关量 OS_TIME_DLY_HMSM_EN 控制是否使用函数 OSTimeDlyHMSM()；OS_TIME_DLY_RESUME_EN 控制是否使用函数 OSTimeDlyResume()；OS_TIME_GET_SET_EN 控制是否使用函数 OSTimeGetO 和函数 OSTimeSet()。本次任务中没有使用这三个系统的时间延迟函数，而是使用了另外一个时间延迟函数 OSTimeDly。所以设定开关量为 0。代码如下：

```
#define OS_TIME_DLY_HMSM_EN      0
#define OS_TIME_DLY_RESUME_EN    0
#define OS_TIME_GET_SET_EN       0.
```

事件控制块最大数：由于系统中的每一个消息邮箱、消息队列、互斥型信号量或者信号量都要一个事件控制块。本次使用的事件控制块设置为 42 为系统默认值。代码如下：

```
#define OS_MAX_EVENTS    42
```

消息队列最大数：用于定义系统中可以创建的消息队列的最大数目。本次任务中设置为 10。代码如下：

```
#define OS_MAX_QS        10
```

对操作系统功能进行裁剪以后代码量(Code)减小了 6K。一定程度上减轻了 CPU 的负担和内存占用率，为系统扩展更多的功能提供了空间。

4.5 μ C/OS II 操作系统的多任务相关概念

本课题选择基于嵌入式操作系统开发软件，主要目的是利用操作系统对实时多任务程序运行具有强大的管理和调度功能。所以程序开发前需要对操作系统多任务管理相关的概念有所了解。

在 μ C/OS II 操作系统下开发应用程序，有许多关于任务的基本概念与程序设计密切相关，下面进行简要的说明^[4]：

任务：一个任务，也称为一个线程，是一个简单的程序，该程序可以认为 CPU 完全只属于该程序自己。实时应用程序的设计过程就包括如何把问题分割成多个任务。每个任务都是整个应用的一部分，都被赋予一定的优先级，有自己的一套 CPU 寄存器内容和栈空间。典型的是，每个任务都是一个无限的循环，都可能处在以下 5 种状态之一：休眠态、就绪态、运行态、挂起态和中断态。

任务切换：任务切换的英文是 context switch，有时也叫 CPU 寄存器内容切换。当多任务内核决定运行另外的任务时，它保存正在运行任务的当前状态，即 CPU 寄存器中的全部内容。这些内容保存在任务的当前状况保存区，也就是任务自己的栈区之中。入栈工作完成后，就把下一个将要运行的任务的当前状况从该任务的栈中重新装入 CPU 的寄存器，并开始下一个任务的运行，这一过程就叫做任务切换。

任务的优先级：每个任务都被赋以不同的优先级。最多可以管理 64 个任务，为简化设计， μ C/OS II 规定所有任务的优先级必须不同，任务的优先级同时也只唯一地标识了该任务，优先级号越低，任务的优先级越高。 μ C/OS II 总是运行进入就绪态的优先级最高的任务。

多任务：多任务的运行实际上是靠 CPU 在许多任务之间转换和调度。CPU 只有一个，轮番服务于一系列任务中的某一个。多任务使 CPU 的利用率达到最高，

并使应用程序模块化。在实时应用中，多任务化的最大特点是，开发人员可以将很复杂的应用程序层次化。

任务间通信：任务间或中断与任务间的信息传递称为任务间的通信，英文是 *intertask communication*。实现的途径有两个：通过全程变量，或发消息给另一个任务。用全程变量时，必须保证每个任务或中断服务子程序独享该变量。中断服务中保证独享的唯一办法是关中断。如果两个任务共享某变量，各任务实现独享该变量的办法可以是关中断，再开中断，或者使用信号量。

5 基于 ARM 控制器和操作系统的软件设计

软件部分分为 DSP 程序模块和基于 ARM 的嵌入式操作系统控制模块两部分。本系统将 ARM 作为主控制器，主要是想利用 ARM 强大的处理器以及嵌入式操作系统的多任务管理功能，对系统各个任务之间的资源分配上进行合理的分配和调度。而从处理器采用 DSP 芯片，主要是想利用其强大的数据处理功能对采集到的数据进行处理。所以系统中与数据相关的主要功能，如数据采样、数据处理、数据保存以及查看历史记录等，都是由 ARM 向 DSP 发出命令，而由 DSP 来完成具体的操作。而 ARM 完成的功能主要是按键扫描、液晶显示、串口通讯等功能外围设备驱动控制的功能。两者之间通过双口 RAM 进行数据通信。这样能够充分发挥各个功能模块的强项，使得仪器能够更加稳定高效的完成任务。更重要的一点是为日后仪器各个功能的升级提供了足够的空间。

软件设计主要用来实现系统应用功能。嵌入式软件的开发是基于嵌入式操作系统的，也就是由嵌入式操作系统提供系统函数的 API 接口，开发人员通过 API 来开发应用软件。开发者根据具体的应用要求及开发环境来决定采用什么样的开发语言以及开发环境。本课题使用的是 ADS 集成开发环境，用 C 语言编写的程序。

一般情况下，嵌入式系统应用软件需要在主机上开发，然后经交叉编译之后再移植到嵌入式平台上进行仿真调试运行。

5.1 软件开发环境介绍

本课题采用的是 ADS 集成开发环境和 EasyJTAG 仿真器，ADS 是 ARM 公司推出的 ARM 核微控制器集成开发工具，本课题使用的版本为 ADS1.2。

ADS1.2 使用了 CodeWarrior IDE 集成开发环境和 AXD 调试器，CodeWarrior IDE 集成了 ARM 汇编器、ARM 的 C/C++ 编译器、Thumb 的 C/C++ 编译器、ARM 连接器，包含工程管理器、代码生成接口、语法敏感(对关键字以不同颜色显示)编辑器、源文件和类浏览器等等。

AXD 调试器为 ARM 扩展调试器(即 ARM eXtended Debugger)，包括 ADW/ADU 的所有特性，支持硬件仿真和软件仿真(ARMulator)。AXD 能够装载映像文件到目标内存，具有单步、全速和断点等调试功能，可以观察变量、寄存器和内存的数据等等^[27, 28, 29]。

EasyJTAG 仿真器是广州周立功单片机发展有限公司开发的 LPC2000 系列 ARM7 微控制器的 JTAG 仿真器，支持 ADS1.2 集成开发环境，支持单步、全速及

断点等调试功能，支持下载程序到片内 FLASH 和特定型号的片外 FLASH，采用 ARM 公司提出的标准 20 脚 JTAG 仿真调试接口。其主要特点如下：

- ◇ 采用 RDI0 通讯接口，无缝嵌接 ADS1.2 和其它采用 RDI 接口的 IDE 调试环境。
- ◇ 高达 1M 速率的 JTAG 时钟驱动。
- ◇ 采用同步 Flash 刷新技术 (synFLASH)，同步下载用户代码到 Flash 中，即下即调。
- ◇ 采用同步时序控制技术 (synTIME)，仿真可靠稳定。
- ◇ 支持 32 位 ARM 指令/16 位 THUMB 指令的混合调试。
- ◇ 增加映射寄存器窗口，方便用户查看/修改寄存器数值。

5.2 控制功能的设计与实现

根据功能划分，程序总体可分为按键扫描、数据采集、液晶显示、数据存储和串口通信五大功能模块。

本系统一共开设了五个任务，从 Task0 到 Task4，优先级依次降低。在操作系统主函数中开两个任务，Task0 为界面控制任务，Task1 为按键扫描任务，在 Task0 中实现液晶显示的四个界面的逻辑关系和界面间的切换，此任务是程序中最主要的任务。同时，在 Task0 中还开设了三个子任务，实现了仪器的各种控制功能，任务的名称及功能说明如表 5.1 所示：

表 5.1 ARM 任务列表

Tab. 5.1 ARM Task list

任务号	任务名称	功能说明
Task0	界面控制任务	控制 LCD 显示的四个界面的逻辑关系和界面间的切换
Task1	按键扫描任务	扫描仪器的按键输入值
Task2	与 DSP 通信任务	通过双口 RAM 与 DSP 进行数据和命令的传输
Task3	串口通信任务	将测试结果通过串口传给上位机
Task4	标题循环任务	在 LCD 显示屏上方动态显示标题

液晶显示的四个界面分别为在线测试、查询功能、设置功能、串口通信。当系统启动后，系统自动跳入在线测试功能模块并显示在线测试界面，通过界面切

换按键，用户可以顺序循环切换四个界面。

仪器控制功能的实现是以界面为依据划分的。它们的作用分别如下：

一、测试功能：

1. 给 DSP 发送测试命令，DSP 接收到命令后开始采集数据并进行数据处理，将处理后的数据发送给 ARM 由 ARM 决定下一步的处理方式。

ARM 在发出命令后读取双口 RAM 中 DSP 的状态位，此时若 DSP 出现错误则 ARM 显示“系统错误”，当 DSP 工作正常并进入采集状态时，DSP 会将采集到的数据处理后放入双口 RAM 的数据位，并置状态位为 0x01，ARM 读取状态位为 0x01 后，从数据位接收 DSP 发送来的数据和时间并将数据显示在 LCD 显示器上。同时根据用户的按键指令进行下一步操作。

所显示的测试数据包括：测试电流 I，测试电压 U，阻抗 Z，接地电阻 R，电阻率 ρ ，频率和干扰电压 U50。在屏幕下方显示日期和时间。

2. 在线测试界面下，当用户按下保存按键时，ARM 将向 DSP 发送保存命令，DSP 接到命令后，将当前处理完的数据和当前日期保存到 RTC 的 NVRAM 中，以便日后进行查询分析。由于存储器空间有限，本仪器设置最多可保存 50 条记录，若超过 50 条，则无法保存，屏幕显示“存储区已满”。

二、查询功能：

本仪器最多可保存 50 条历史记录，在查询界面下，可查询任何一条记录值。初始值为最新保存的记录值。通过“+”“-”按键可以搜索显示上一条和下一条的历史记录值以及当时的具体日期和时间。若无记录保存，显示“无记录”，当查询记录到两端时，显示“已到第一条”或“已到最后一条”。

用户按下清除按键，ARM 扫描到键值后向 DSP 发送命令，DSP 清除当前显示的历史记录值及其日期。因为存储空间有限，仪器只能保存有限的历史记录值，所以没用的记录需要及时删除。

三、设置功能：

在设置界面时，用户可通过选项按键选择需要设置的选项，通过“+”“-”按键改变选项值。当设置完成后，按下保存按键，ARM 向 DSP 发送保存命令，并将设置完的数据以数据帧的形式发送给 DSP，这四个选项分两组保存，时间和日期一组保存，频率和间距一组保存，由 DSP 将数据保存到 RTC 的 NVRAM 中。若设置后不按保存按键，则修改的数据不保存。设置的参数如下：

1. 设置测量频率：根据国际大电网会议 36.04.01(变电站接地)工作组(1982 年)的规定，测量电流的频率与工频频率的偏差应不大于 10Hz。我国的工频为 50Hz，所以本仪器将频率设置规定应该在 40-50Hz 之间。
2. 设置间距：电极间距用来作为计算电阻率的参数，要根据实际测试中的电极间

距来设定。

3. 设置背光：通过给背光驱动位置 1 或置 0 来打开或关闭背光。
4. 设置系统时间：本系统由专用 RTC 芯片 DS1742W-100 来控制时钟，ARM 将设定的时间值传给 DSP，再由 DSP 将时间日期保存在 RTC 缓存中，包括年、月、日、时、分、秒。

四、通信功能：

通过串口将数据传给 PC 机。与上位机的数据传输是通过 RS-232 串行通信接口进行的，当进入通信界面时，仪器自动开始向串口发送数据，通信波特率设置为 9600bps。上位机利用串口通信软件接收数据。数据以数据帧格式传输，传输是单向的，上位机不向仪器发送数据。此通信方式比较简单，并未采用中断方式，当仪器跳出通信界面时传输停止。

仪器设置串口通信的目的是为了方便用户将测试出的数据直接存储在计算机里方便携带保存。

LCD 显示器显示的四个界面由下面一组图 5.1 表示

<>华电科研DRT820电网接地电阻测 在线测试 I=14.500A U=10.433V 测试中... Z=000.000Ω R=000.000Ω 频率:08Hz ρ=0000ΩM 06年03月05日 15:58:30	<>华电科研DRT820电网接地电阻测 查询记录 I=14.500A U=10.433V 测试停止 Z=000.000Ω R=000.000Ω ρ=0000ΩM U50=00.000V 08年03月05日 15:58:30
<>华电科研DRT820电网接地电阻测 参数设置 I=14.500A U=10.433V 测试停止 Z=000.000Ω 背光:关闭 R=000.000Ω 频率:55Hz ρ=0000ΩM 间距:50M U50=00.000V 08年03月05日 15:58:30	<>华电科研DRT820电网接地电阻测 串口通信 I=14.500A U=10.433V 通信中... Z=000.000Ω R=000.000Ω ρ=0000ΩM U50=00.000V 08年03月05日 15:58:30

图 5.1 LCD 显示界面组图

Fig. 5.1 a group of pictures about LCD Display Interface

5.3 程序运行流程及相关功能函数

5.3.1 主程序流程设计

整个程序代码用 C 语言实现，本工程的主程序即程序入口点就是 main 函数，

主程序流程图如图 5.2 所示:

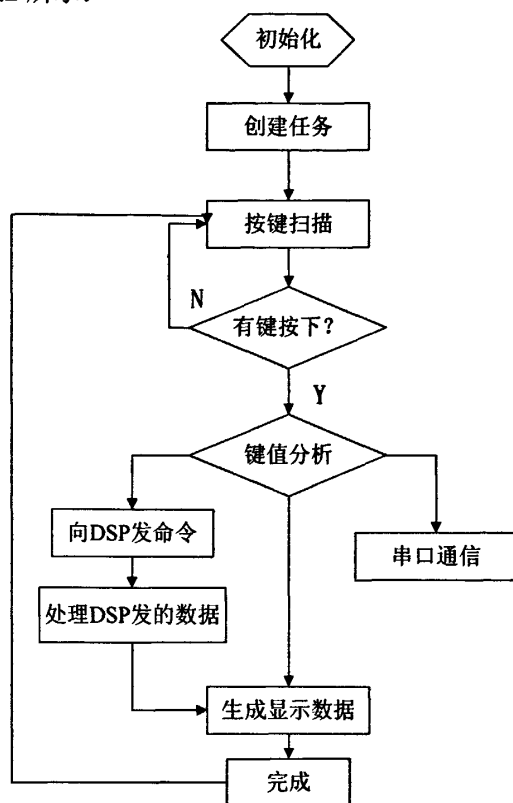


图 5.2 ARM 主程序流程图

Fig. 5.2 ARM Main program flow chart

在使用 μ C/OS II 提供的任何功能之前,首先要对系统及外围设备进行初始化。系统初始化需调用系统函数 `OSInit()`。`OSInit()`的任务是初始化 μ C/OS II 中所有的变量和数据结构,同时它会建立两个系统任务:空闲任务 `OSTaskIdle()`,它的优先级最低,该任务只是不停地给一个 32 位的整型变量加一,该数用于计算 CPU 利用率;统计任务 `OSTaskStat()`,该任务每秒运行一次,此任务负责计算 CPU 的利用率。系统运行时,只有至少生成一个任务时,才能进行多任务调度。

此外本程序中还需要对外围器件初始化,如液晶显示器调用了 `LCD_Initialize()` 初始化函数,串口初始化调用了 `UART0_Init()` 函数。

在系统初始化之后, μ C/OS II 开始调用系统函数 `OSTaskCreateExt()`,此函数用来创建任务,本程序中在主函数里创建了两个任务,即 `Task0` 和 `Task1`,分别为界面控制任务和按键扫描任务。在 `Task0` 中又开设三个子任务 `Task2`、`Task3`、`Task4`,其中主任务 `Task0` 采用的是循环轮询结构。多个任务之间通过操作系统进行资源分配,“同时”运行。

任务创建完成后,调用系统函数 `OSStart()`,将控制权交给 μ C/OS II 操作系统。

开始正式运行多任务。LPC2214 内核属于抢占式内核系统，程序运行时，正在运行的任务期待某一事件的发生时要等待，此时该任务进入等待态，系统让就绪态下优先级最高的任务进入运行态。当该事件发生了，被挂起的任务进入就绪态。事件发生的报告有可能来自另一个任务运行产生，也可能来自外部中断服务子程序。如果事件发生的报告来自另一个任务时，系统将立刻进行任务优先级检查，如果当前任务不是优先级最高的任务，将进行任务调度，运行优先级最高的就绪态任务；当事件发生的报告来自中断服务子程序时，则在子程序返回时进行任务优先级检查，并决定是否进行任务调度。此种运行方式总能保证优先级最高的就绪态任务能够抢占当前运行态任务，从而保证了最重要的任务能在确定的时间内完成。当所有任务都处在等待状态时，操作系统执行空闲任务，空闲任务 OSTaskIdle()总是处于就绪态。

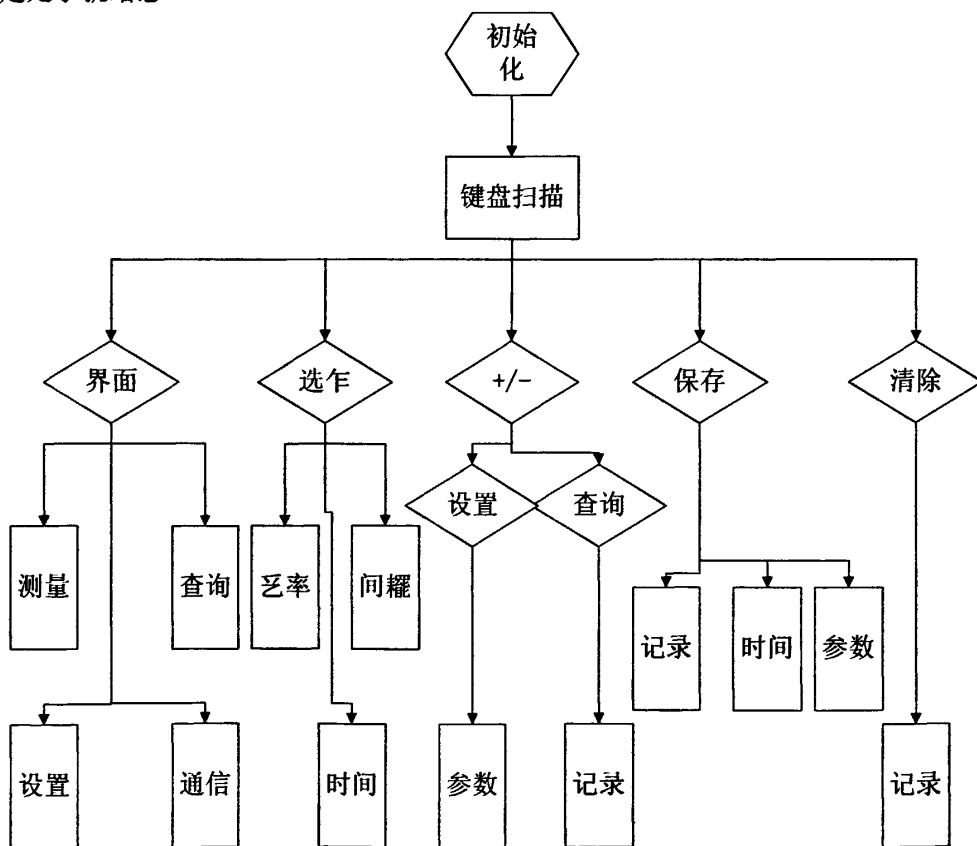


图 5.3 按键功能结构图

Fig. 5.3 key function chart

整个系统的按键功能结构图如图 5.3，仪器面板共有六个按键，用来输入用户命令，此按键设计基本能够实现测试所需要的所有功能。

在编写具体的功能任务函数之前，先列出程序编写需要用到的 ARM 芯片的地址及部分相关引脚分配表 5.2:

表 5.2 ARM 芯片资源列表

Tab. 5.2 ARM chip resources list

信号	地址	目标设备	备注
	0x40000000~0x40003FFF	片内 RAM	16KB (存储数据)
	0x00000000~0x0001FFFF	片内 FLASH	128KB (存储程序或数据)
nCS0A	0x80000000~0x80FFFFFF	SRAM	2MB(存储数据) 0x80000000~0x801FFFFF
nCS1A	0x81000000~0x81FFFFFF	DARAM	8K(与 DSP 数据交换) 0x81000000~0x81001FFF
nCS2A	0x82000000~0x82FFFFFF	液晶屏	0x82000000 为液晶屏数据 口 0x82000002 为液晶 屏命令口
P1.16A- P1.23A	P1.16~ P1.23A	按键	
LEDA	P0.22	指示灯	指示设备工作状态
P1.14A	P1.14	液晶屏	液晶屏背光开关。输出高电 平打开液晶屏背光, 输出地 电平关闭液晶屏背光

调用的相关函数主要有以下几个:

名称: void Task1(void *pdata)

```
while(1)
```

```
{
    uint32 temp;
    temp = IO1PIN & 0x00ff0000;    /* 读键值 */
    delay(100);
    if(temp == (IO1PIN & 0x00ff0000))
    {
        delay(100);
        if(temp == (IO1PIN & 0x00ff0000))
        {
            if(temp != DIKey_Image)
            {
                DIKey_Image = temp;    /* 新的按键按下 */
                DIKeyTime = 2;
            }
        }
        else
        {
            if(--DIKeyTime <= 0)
            {
                DIKey0 = DIKey_Image;    /* 滤波时间到 */
                DIKeyTime = 0;
            }
        }
    }
}
```

```

    }
}
OSTimeDly(OS_TICKS_PER_SEC / 100); /* 延时 10 毫秒 */
}

```

功 能：按键扫描任务，主要利用了防抖动技术。此任务用死循环函数不断读取八位按键值。由于电磁干扰或人为的抖动可能会影响按键值，所以当读到值时，函数采用一段时间的时延去除抖动，若时延后按键值仍保持不变，则将值存入变量 DIKey0。

以下函数由于代码较多，不再详细列举。

名 称：uint8 *data_deal (int num_bit,unsigned short dataself)

功 能：将所收到的 16 位数据转换成 10 位数据显示

入口参数：int num_bit :数据共有几位； unsigned short dataself 数据本身

出口参数：指向一个八位整型数据的地址指针

名 称：void SendByte(uint8 data)

功 能：向串口 UART0 发送字节数

入口参数：要发送的数据

名 称：void SendBuf(void)

功 能：将缓冲区的数据发送给主机(使用 FIFO)，并等待发送完毕。

5.3.2 ARM 与 DSP 的数据传输实现

ARM 与 DSP 之间的通信是通过双口 RAM 进行的。双口 RAM 存储单元为 16 位。通信包括三部分，向 DSP 发出命令，发送或接受数据和读取双口 RAM 数据位状态部分。

本系统中 ARM 为控制芯片，DSP 的主要作用为信号处理，接受 ARM 的控制，ARM 发出命令，DSP 接到后执行。

写命令地址 DARAMARMCW 0x81000000

规定命令字长为 32，在双口 RAM 中采用两个 16 位存储空间存储一个命令字。

本系统所用到的命令，如下表：

表 5.3 系统命令列表

Tab. 5.3 system command list

含 义	命令发送缓冲格式
使 DSP 开始测试	0x00100000
使 DSP 保存当前记录	0x00100001
查询当前记录	0x00200000
查询上一条记录	0x00200001
查询下一条记录	0x00200002
清除当前记录	0x00200003
保存时间设置	0x00300000
保存频率和间距设置	0x00300001

ARM 与 DSP 之间的通信采用数据帧的形式进行。

ARM 发送数据的地址为: DARAMARMDW 0x81000400

ARM 接收数据的地址为: DARAMDSPDR 0x81000200

本系统两 CPU 通信时约定了相同的数据帧格式。如表 5.4 所示, 每个数据占两个字节, 一次向 DSP 发一帧数据。

表 5.4 数据帧格式

Tab. 5.4 data frame format

1 年	8 U (V)
2 月	9 Z (Ω)
3 日	10 R (Ω)
4 时	11 F (Hz)
5 分	12 Ud(干扰电压)
6 秒	13 ρ (ΩM)(电阻率)
7 I (A)	

通过设置双口 RAM 的状态标志使 ARM 与 DSP 之间的数据传输同步, 当 DSP 向双口 RAM 中写入数据后置状态标志位为 0x01, 当 ARM 取走数据后置状态标志位为 0x02。同时, 在 DSP 每次操作完成后, ARM 根据状态位的值判断 DSP 操作是否正常, 如果出现操作错误, 则 ARM 在显示器上显示“系统错误”, 并重新向 DSP 发送操作命令。状态位的命令如表 5.5。

读取双口 RAM 状态地址: DARAMDSPSR 0x81000100

表 5.5 状态位命令及含义

Tab. 5.5 Status bit command and meaning

状态标志	含义
0x00	DSP 操作正常
0x01	双口 RAM 数据位有数据
0x02	双口 RAM 数据位无数据
0xFF	DSP 操作错误

5.3.3 LCD 显示的实现

本仪器采用的是 240128A 液晶显示器模块能显示 240*128 的点阵。它可用于显示字符，图形和字符-图形。在本课题中显示采用的是纯图形模式，为了能够显示各种汉字和字符，并且尽量少增加代码的数量，本课题中选择在程序工程中自己建立两个外部字库，分别为全角（16*16 点）和半角（8*16 点）类型字库。建立字库利用字模软件，此软件能够自动生成所需要文字的对应代码，将生成的代码针对此款 LCD 显示器的显示地址位进行调整，调整后的数据放入字库中，并标明序号。最后调用系统中的基于 LCD 驱动函数编写的写全脚字符和写半脚字符函数，就可以实现 LCD 显示器的文字和字符显示。

调用的相关函数主要有以下几个：

名称：void LCD_CRT2(uint8 plot_x,uint8 plot_y,uint8 form,uint8 data_leng,uint8 chra_num)

功能：显示字符串

入口参数：坐标 (plot_x,plot_y) 为字符串第一个字符的位置；半角或全角：form 0 为全角，1 为半角；字符串长度 data_leng，最大值为 16（全角），32（半角）；显示内容 chra_num；

名称：void LCD_CLEAR(void)

功能：清屏

名称：void jiemian(void)

功能：界面显示

名称：void biaoti(void)

功能：标题滚动显示。函数内部用字符数组存储显示的文字在字库中的标号，调用 LCD_CRT2()函数显示，每显示一次，整个数组中的字符左循环一个字符位。

名称：CCW1_PR()

功能：文字反白显示

名称: LCD_Initialize()

功能: LCM 初始化, 将 LCM 初始化为纯图形模式。

名称: void timedisplay(void)

功能: 显示时间和日期

6 基于 DSP 的软件设计及算法分析

6.1 异频测量法的算法实现

由于在信号处理时需要用到 FFT 变换和滤波技术，所以这里简要介绍下 FFT 算法的基本原理。

傅里叶变换是常用于数字信号处理的重要算法，变换的目的是将信息从时间域变换到频率域，实现傅立叶变换计算上有效的方法是快速傅里叶变换(FFT)。

离散傅里叶变换(DFT)的功能是将一段数字信号转换成其各个频率的正弦波分量。如果我们的信号可以表示成 $x[n], n=0$ 到 $N-1$ ，那么 DFT 的公式如下：

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-\frac{j2\pi}{N}nk} = \sum_{n=0}^{N-1} x[n] \cdot W_N^{nk}, n = 0, 1, 2, \dots, N-1 \quad (6.1)$$

这些傅立叶系数 $X[k]$ 所代表的信号是 k 的函数，而 k 直接和频率有正比关系，因此这些系数 $X[k]$ 通称为频谱，对于 $X[k]$ 的分析，通称为频谱分析。也可以有这些傅立叶系数 $X[k]$ 反推出原始信号 $x(n)$ ，即：

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \cdot e^{\frac{j2\pi}{N}nk} = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \cdot W_N^{-nk}, k = 0, 1, 2, \dots, N-1 \quad (6.2)$$

这里几点要说明：如果原始信号 $x[n]$ 有 N 个点，那么转换出来的信号 $X[k]$ 也有 N 个点。一般而言， $X[k]$ 是一个复数，如果原始信号 $x[n]$ 都是实数，那么 $X[k]$ 和 $X[k-1]$ 是共轭复数。如果信号是一正弦波则能量频谱只有在两点不为零，其他各点理论上应该为零。相位频谱没有一定规则，像是乱数。这是由于大部分的能量频谱很小，因此实际的相位频谱并没有任何意义。事实上当 $x[n]$ 无限长，则其频带有限，当 $x[n]$ 有限长，则其频带无限。如果不采取特别的措施很容易出现三种现象：混叠、泄漏、栅栏。

如果 $x[n]$ 的频率不是这些正弦波中的一个，那么 DFT 还是会将此信号拆解成基本正弦波的组合，因此能量频谱也就由其各个基本正弦波组成。如果所用的 DFT 系数越多，所合成的波形就越接近原来的波形。

显然，求出 N 点 $X(k)$ 需要 $N*N$ 次复数乘法及 $N*(N-1)$ 次复数加法。因为实现一次复数乘法需要四次实数乘法两次实数加法，实现一次复数加法需要两次实数加法。所以当 N 很大时计算非常大。例如， $N=1024$ ，则需要 1048567 次复数乘法，即要进行 4194304 次实数乘法运算，所需时间很长，难于“实时”进行。

1965 年 Cooley 和 Tukey 提出了快速傅立叶变换(FFT)使 N 点 DFT 的复数乘法计算量有 $N*N$ 次降到了 $N/2 * \log_2^N$ 次。以 $N=1024$ 为例，计算量降为 5120 次，仅为原来的 4.88%^[30]。

FFT 算法现在已经出现的有很多种,不过常用的算法还是基 2 时间抽取(DIT)算法。由于该算法中将时间下标按奇偶分开,故称为时间抽取算法。

在传统的测量方法中,注入接地网的电流一般是工频电流。用工频电源测量接地电阻,主要问题来自于大地中的干扰电流。它和测量用的信号电流混杂在一起。进入测量仪表。使读数产生很大的误差。因而大大限制了该测量系统在运行变电站中的实用价值。新建的变电站,虽然本站电力设备尚未投运,似乎地中干扰电流较小。可是实践证明。依然不能回避这个问题。由此可见,传统的工频测量系统难于解决地中干扰电流与信号电流的分离问题。小容量电源就不能实现运行中变电站接地电阻的正确测量问题。反之,用异频测量系统,由于测量频率偏离工频,可用小电流信号实现正确测量。

但接地网的接地电阻和频率相关,在低频范围内接地网的接地电阻随频率的增加有缓慢上升的趋势。国际大电网会议 36.04.01(变电站接地)工作组(1982 年)指出,测量电流的频率与工频频率的偏差应不大于 10Hz,偏离太大将产生显著的误差。

利用异频法进行测量的基本步骤可分为:^[31]

- 1、测量地网干扰信号的频谱,找到在 50Hz 附近,40Hz~60Hz 之间干扰比较小的一段频段。
- 2、调节测量电源的频率,使之工作在这个频段上。
- 3、通过提取电流极引线中的电流和电压极引线中的电压中对应于测量电源工作频率下的信号的幅值和相位,从而得到测量接地电阻值。

由于接地网的接地电阻和频率相关,为了实现比较准确的测量,一般在多个频率下进行测量,最终结果由插值得到

若接地网中的电流为

$$i_g = I_{g0} + \sum_{n=1}^N \sqrt{2} I_{gn} \sin(2\pi n f_g t + \varphi_{ni}) \quad (6.3)$$

其中 f_g 为工频频率; I_{gn} 为谐波电流有效值(对工频基波, $n=1$); I_{g0} 为直流分量,地网无故障电流时 $I_{g0} \approx 0$; n 为谐波次数; φ_{ni} 为谐波电流初相角。若向地网中注入异于工频(频率为 f_y)的电流

$$i_y = \sqrt{2} I_y \sin(2\pi f_y t + \varphi_{yi}) \quad (6.4)$$

并令 $w_g = 2\pi f_g$, $w_y = 2\pi f_y$, $k = w_g / w_y$, 则测量时接地网中的全电流和电压极与接地网之间的压降分别为

$$i = \sum_{n=1}^N \sqrt{2} I_{gn} \sin(nk w_y t + \varphi_{ni}) + \sqrt{2} I_y \sin(2\pi f_y t + \varphi_{yi}) \quad (6.5)$$

$$u = \sum_{n=1}^N \sqrt{2} U_{gn} \sin(nk w_y t + \varphi_{nu}) + \sqrt{2} U_y \sin(2\pi f_y t + \varphi_{yu}) \quad (6.6)$$

因此,应用异频法测量的关键之一是如何从 i 、 u 中无幅值和相位畸变地提取异频电流 $\dot{I}_y = I_y e^{j\varphi_y}$ 和电压 $\dot{U}_y = U_y e^{j\varphi_y}$, 并算出工频接地阻抗

$$Z = \dot{U}_y / \dot{I}_y = R_g + jX_g \quad (6.7)$$

由于快速傅里叶变换(FFT)存在栅栏效应和泄漏现象,直接利用快速傅里叶变换提取测量信号中的幅值存在较大的误差。文献[32]针对 FFT 的频谱泄漏和栅栏效应,对 FFT 序列进行简单改造,然后进行插值计算,对 FFT 的结果进行修正,极大地提高了计算精度。

设测量系统的频率分辨率为 Δf , 测量电源的实际工作频率为 f_0 , 由于不能完全实现整周期采样, f_0 不是 Δf 的整数倍, 设 $f_0 = (k_1 + r)\Delta f$, 其中 k_1 为整数, $0 < r < 1$ 。设离散信号 $x(n)$ 的频谱为 $X(k)$, 根据已有的频谱序列, 构造出一个新的序列 $X_1(n)$:

$$X_1(n) = X(n) - \frac{1}{2}(X(n-1) + X(n+1)) \quad (6.8)$$

这种算法其实相当于加窗 FFT 算法(汉宁窗)。这种做法不必构造窗函数,只需对原始的 FFT 序列做简单的变换,实现起来比较方便。

序列进行变换后,设幅值最大且是相邻的两条谱线 ($n = k_1$ 和 $n = k_1 + 1$) 的幅值之比为 α , 则可得到以下关系:

$$\alpha = \frac{X_1(k_1)}{X_1(k_1+1)} = \frac{2-r}{1+r} \quad (6.9)$$

$$r = \frac{2-\alpha}{1+\alpha} \quad (6.10)$$

把 r 代入 $f_0 = (k_1 + r)\Delta f$ 就可以得出测量电源的实际工作频率。并根据幅值计算公式求出信号的幅值 A_m [33]。

一般认为,在无法实现整周期采样时,FFT 所测得的相角的误差太大无应用价值,但文献[31]的研究表明对于在非整周期采样时,也可应用。文献[31]分析了用 FFT 测量周期信号相角误差的来源,对非整周期采样造成的泄漏误差进行了估计,指出当采样总时间足够长时,相角测量值与实际值基本上相差一个与非整周期数成比例的常数,因此,在测量同频率的两个信号的相角差时,主要误差恰好抵消,得到很高的测量精度。[31]

6.2 数据处理的程序设计与实现

首先需要了解 DSP 的资源和地址是如何分配的,列表如下:

表 6.1 DSP 芯片资源列表

Tab. 6.1 DSP chip resources list

信号	地址	目标设备	备注
----	----	------	----

RAMB0	0x87FE00---0x87FEFF	片内 RAM	存储数据 (256*32)
RAMB1	0x87FF00---0x87FFFF	片内 RAM	存储堆栈 (256*32)
STRB0	0x000000---0x03FFFF	FLASH	存储程序或数据 (256K*32)
STRB1	0x900000---0x93FFFF	SRAM	存储数据 (256K*32)
IOSTRB	0x810000---0x81002F	AD	48 B
	0x818000---0x819FFF	DARAM	双口 RAM,用于与 ARM 进行数据 交换 (4K*16)
	0x828000---0x8287FF	RTC	存储数据 (2K B)
DSPRDY	0x820000 Bit0	与主控 ARM 握手线	指示设备忙否
STARTC	INT1		用于触发中断/查询
		JTAG	用于程序仿真

DSP 程序主要根据 ARM 发来的命令进行定时采样、数据处理以及数据保存等功能。实现流程图如下:

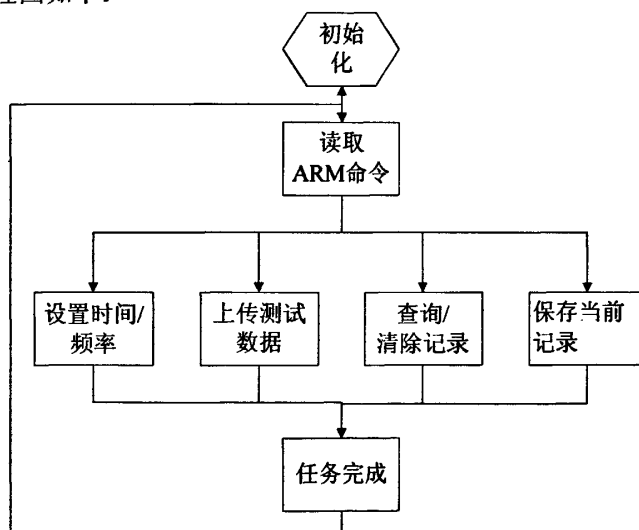


图 6.1 DSP 功能流程图

Fig. 6.1 DSP function flow chart

ARM 命令变量定义:

DARAM_C_PTR: DARAM (双口 RAM) 中的命令存储区, 16 位

CMD_PTR: DSP 中的命令存储区, 32 位

测试:

UPDATA=0x0010 0000

保存当前记录:

SAVEDATA=0x0010 0001

查询/清除记录:

INQDATA (CURRENT)=0x0020 0000 查询最新记录

INQDATA (PRE)=0x0020 0001 查询上一条记录

INQDATA (NEXT)=0x0020 0002 查询下一条记录

INQDATA (CLEAR)=0x0020 0003 清除当前记录

设置时间/频率和间距:

SET(TIME)=0x0030 0000 保存时间设置

SET(FREQUENCY)=0x0030 0001 保存频率和间距设置

状态区:

STATUS_PTR: DSP 中的状态区

DARAM_S_PTR: DARAM (双口 RAM) 中的状态存储区, 16 位

DSP 状态: 00H 操作正常;

FFH 操作错误;

DSP 接收数据区:

DSP_R_PTR: DSP 中的数据接收区, 32 位

DARAM_R_PTR: DARAM (双口 RAM) 中的数据接收区, 16 位

DSP 发送数据区:

DSP_T_PTR: DSP 中的数据发送区, 32 位

DARAM_T_PTR: DARAM (双口 RAM) 中的数据发送区, 16 位

时钟:

R_Timebuf_PTR: DSP 中的读时钟区域

S_Timebuf_PTR: DSP 中的写时钟区域

RTC_PTR: RTC 中时钟内部的存储区

DSP 与双口 RAM、ARM 之间通信的数据格式在 ARM 章已经描述过。这里不再重复。

6.2.1 测试界面程序设计

首先, 当仪器上电启动后, 自动进入到测试界面, 此时 ARM 向 DSP 发送测试命令, DSP 接到命令后开始测试。DSP 开启中断 INT0 接收来自 AD 芯片的采样信号, AD 将采集到的模拟电压量转换成数字量后, 发送给 DSP, 当采集到一定数

量的采样点后，DSP 关闭中断，再由 DSP 进行分析处理，然后由 DSP 将处理后的数据信息通过双口 RAM 发送给 ARM 处理器进行显示或传给上位机。具体过程流程图如下：

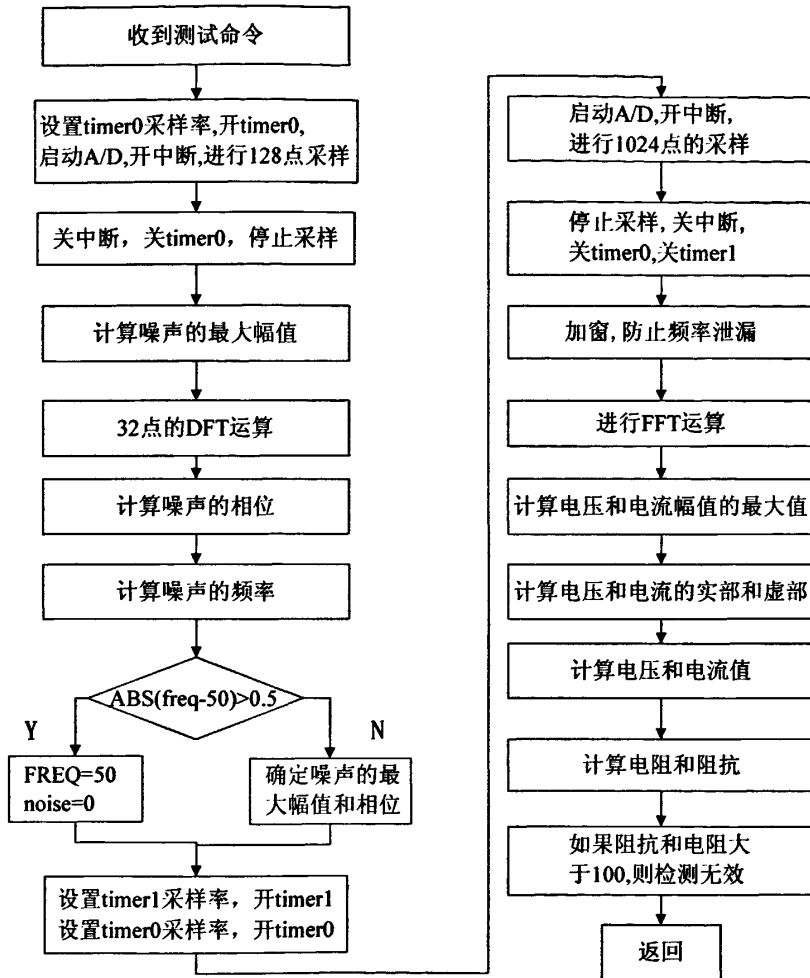


图 6.2 测试功能流程图

Fig. 6.2 measuring function flow chart

在测试状态时，如果用户点击保存按钮时，ARM 将给 DSP 发送保存命令，DSP 把当时采集到的数据保存到 RTC 的存储器中，由于 RTC 使用的是非易失性存储器，所以在系统掉电后记录仍然存在。RTC 有 2K*8bit 的存储空间，最多可以保存 50 条记录。保存过程流程图如下图 6.3：

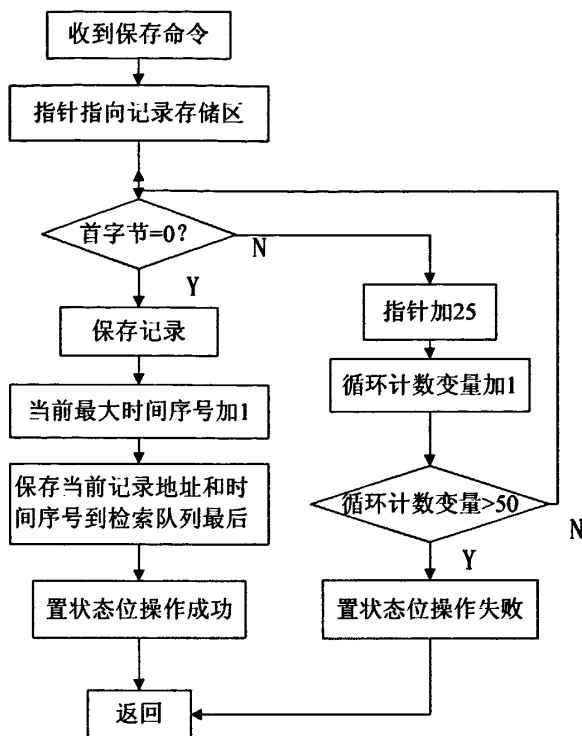


图 6.3 保存功能流程图

Fig. 6.3 save function flow chart

测试程序将测量到的和经过处理后得到的数据上传给 ARM，上传数据包括测量电压(U)、电流(I)、电阻(R)、电抗(Z)、干扰电压(Ud)、土壤电阻率(ρ)。若其中某一项数据超过规定的范围，则置标志 overflag 为 1，检测无效，但不具体告知究竟是哪一个数据超出范围。

6.2.2 查询界面程序设计

当进入查询界面时，程序自动显示最新一次保存的历史记录值及其保存时的时间和日期。若查询时出现错误，则界面显示“系统错误”，程序自动返回。

存储区变量定义：

RAM_INDEX_PTR: RTC 中的 RAM 中的检索存储区，用来存放历史记录索引数，此存储区掉电保持，8 位

RAM_RECORD_PTR: RTC 中的 RAM 中的记录存储区，用来存放历史记录，此存储区掉电保持，8 位

查询过程流程如下图 6.4

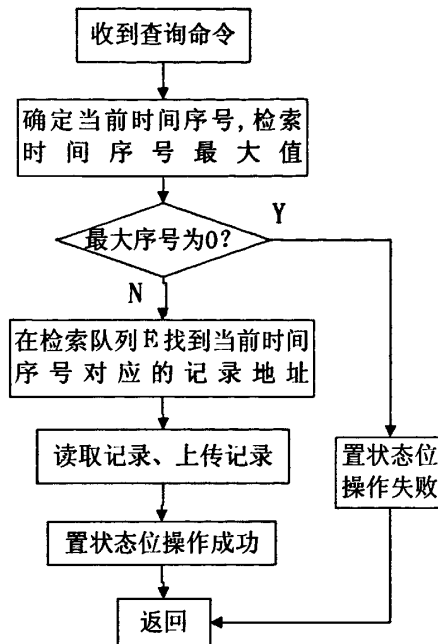


图 6.4 查询功能流程图

Fig. 6.4 research function flow chart

用户可通过控制面板上的“+”、“-”按键逐条查询存储器里所有的历史记录，加号键查询上一条记录，减号键查询下一条记录。当查询到两端的记录时，若再按加号键或减号键，界面显示“已到最后一条”或“已到第一条”。若无记录保存，显示“无记录”。

用户可通过控制面板的“清除”按键清除当前显示的历史记录信息。流程图如下：

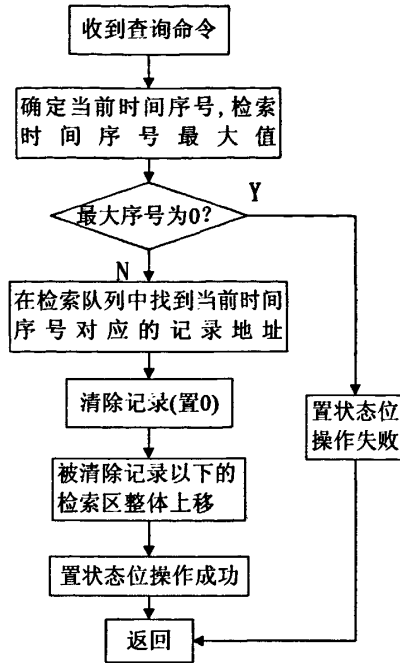


图 6.5 清除记录功能流程图

Fig. 6.5 Clear history function flow chart

6.2.3 设置界面程序设计

在设置界面可选择设置时间、日期、频率和间距。通过“选项”按键可以在这四个选项之间进行切换，当反白显示停在哪个选项上时，可以通过“+”、“-”按键设置该选项的数值。修改完选项值后，需要按“保存”键才能将修改的值保存，此项功能由 ARM 控制，而有 DSP 来完成保存操作。这四个选项分两组保存，时间和日期一起保存，频率和间距一起保存，DSP 接到 ARM 的命令执行将数据保存到 RTC 的 NVRAM 中的动作。DSP 中需要的变量定义如下：

设置时间：

RTC_YEAR: 年

RTC_MONTH: 月

RTC_DATE: 日

RTC_H: 时

RTC_M: 分

RTC_S: 秒

RTC_CTRL: 控制

RTC_OSC: 晶振

设置频率、间距:

fs: 频率

miu: 间距

数据格式:

表 6.2 参数设置数据结构

Tab. 6.2 parameters set data structure

1	年
2	月
3	日
4	时
5	分
6	秒
7	F (Hz)
8	miu (间距, M)

程序流程如下图 6.6 所示:

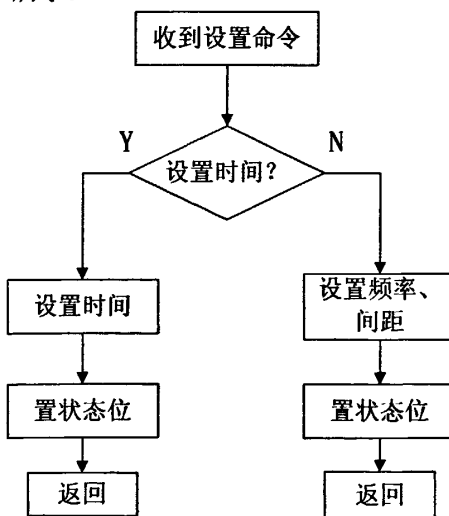


图 6.6 设置功能流程图

Fig. 6.6 setting function flow chart

7 模拟实验及结果分析

实验设备采用一个信号发生器和一个用来模拟接地电网中工频干扰的接地电阻测试模拟实验箱。此实验箱的原理图如下图 7.1。实验箱主要由一个调压器和一个隔离变压器组成。调压器有一个可以滑动的触点和一个环形的线圈，可以改变触点在环形线圈的位置来改变模拟地网电压值，电压变换范围为 0-30V。模拟接地电阻 R 为 0.2Ω 。用隔离变压器可以消除 220V 交流电压干扰。

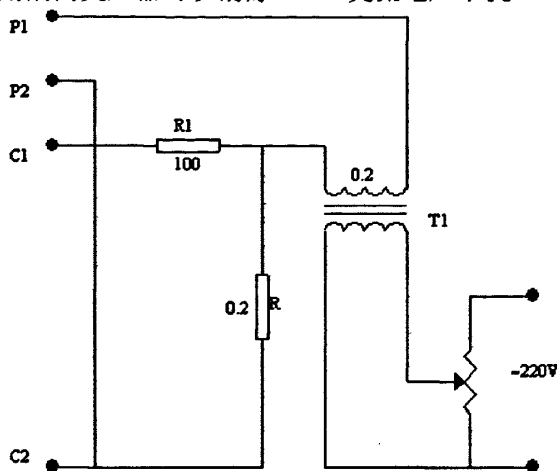


图 7.1 实验箱原理图

Fig. 7.1 Experimental box schematics

实验是在 2008 年 5 月 14 日下午三点开始进行。实验时将信号源与测试仪相联，由 DSP 电路控制测试信号频率，频率设置范围为 40-60Hz 之间。测试仪的 P1、P2 口与实验箱 P1、P2 口相联，用来向实验箱输入测试信号同时作为电流测试端。测试仪 C1、C2 口与实验箱 C1、C2 口相联作为电压测试端。实验结果如下表：

表 7.1 实验数据表

Tab. 7.1 Experimental data list

实验一	测试结果	实验二	测试结果	实验三	测试结果	实验四	测试结果
I	0.232A	I	0.233A	I	0.178A	I	0.238A
U	0.046V	U	0.046V	U	0.035V	U	15.670V
Z	0.199 Ω	Z	0.197 Ω	Z	0.199 Ω	Z	0.298 Ω
R	0.199 Ω	R	0.197 Ω	R	0.199 Ω	R	22.974 Ω
ρ	62 Ω M	ρ	62 Ω M	ρ	62 Ω M	ρ	7217 Ω M
Ud	0.164V	Ud	22.152V	Ud	21.650V	Ud	21.643V
F	55Hz	F	55Hz	F	45Hz	F	50Hz

Date	08.05.14	Date	08.05.14	Date	08.05.14	Date	08.05.14
Time	15:05:35	Time	15:18:51	Time	16:13:54	Time	16:14:23

实验一测试频率为 55Hz，干扰信号极小仅为 0.164V，测得的电阻值为 0.199 Ω ，误差仅为 0.001 Ω 。实验二测试频率为 55Hz，干扰信号为 22.152V 时，测得的电阻值为 0.197 Ω ，误差为 0.003 Ω ，能够满足测量要求。实验三测试频率为 45Hz，干扰信号为 21.650V 时测得的电阻值为 0.199 Ω ，十分接近准确值。实验四测量信号和干扰信号同为 50Hz 时，测得的电阻值为 22.974 Ω ，此时误差非常大，而且此时每次采样测得的电阻值变化非常大，从几欧姆到几百欧姆，此现象是由干扰信号与测试信号叠加所导致，测试效果与理论分析相符合。

由实验结果可以看出，本接地电阻测试仪能够很好的实现测量功能，抗干扰能力很强，而且测量误差小，精确度高，测量方法十分简捷。模拟实验的实物图和实验显示的结果图见附录。

8 总结与展望

目前随着各种大型电力设备的广泛应用，电网接地电阻测试仪已经越来越多的应用到各个领域，并且需求仍在飞速增长。而嵌入式技术更是已经深入到人们生活中的各个方面。本文将现代的研究热点嵌入式技术引入到传统的接地电阻测试当中，研制出了一种基于嵌入式系统的电阻测试设备。有效的改进了传统的电阻测试仪体积笨重、测量烦琐、操作不便的缺点。论文的主要内容包括：

1、研究和分析了目前国内外的电网接地电阻测试技术的测试理论和测试方法，掌握了电阻测试技术的发展历史，并对不同的测试原理进行了比较，具体阐述了本文采用的异频法测量原理。在此基础上构建了系统硬件平台的框架和结构。

2、介绍了嵌入式系统的发展过程和特点，并对本仪器采用的硬件结构进行了说明。本文的工作重点是软件部分的开发，包括嵌入式系统的选择与移植，ARM 方面基于 LPC2214 芯片控制程序的设计与实现，主要用来实现液晶显示、与上位机通讯以及仪器的整体控制功能。DSP 方面基于 TMS320C32 芯片信号处理程序的设计与实现，主要负责信号采集、数据处理与数据的保存。本文以减小设备的体积、增强测试的便捷性、提高系统可靠性和精确度为主要目的，设计了一款基于嵌入式系统的电网接地电阻测试仪。

3、对仪器进行调试和实验，基本上完成了系统设计的各项功能，能够测量出精确度较高的接地电阻值。

下一步的工作展望及改进意见：

1. 由于电场测试的特殊环境，可以为仪器增加网口通信功能，提供远程控制的测试功能。能够有效保障测试人员的安全。
2. 在输入信号进入 AD 采样前，可以采用一些措施（如陷波器等）对工频信号（50Hz）进行抑制，而不影响测试信号，以减少工频信号对测试的影响，提高测量精度。
3. 由于在实际测试中信号频率的小范围波动会造成频谱的泄露，可以考虑利用锁相环（PLL）技术使信号频率与采样频率同步，这样可以有效防止频谱泄露并增强系统实时性。

参考文献

- [1]刘黎,地网接地电阻短距测量方法和降阻新技术研究.[学位论文].2003
- [2]ASIC/IEEE Std80-1983, IEEE guide for measuring earth resistivity, ground impedance, and earth surface potentials of a ground system.
- [3]刘宝,低电压大电流法检测接地网技术研究.华北电力技术,1999, (2):7-8
- [4]DL/T621-1997,交流电气装置的接地.
- [5]肖如泉,何金泉.高电压试验工程.北京:清华大学出版社,2001
- [6]刘晋,陈鹏云.解读新版《接地装置特性参数测量导则》.华中电力.2006, 19(5):33-34
- [7]许聪颖,接地网接地电阻近距离测试原理及测试装置的研究.2007.
- [8]李茂堂.基于白噪音的接地电阻测量原理与误差分析.电网技术.1996,20 (10):41-43
- [9]李茂堂,刘月生,王新中.基于功率谱和高阶谱的接地电阻测量方法研究.电子测量与仪器学报.1998,12(3):7-12
- [10]曹梅月等.变频技术用于运行中大型变电站地网接地电阻的测量.供用电.1993 (2):18-20
- [11]杨兆军.异频法测量地网工频接地电阻.贵州电力技术.2001,(7):39-40
- [12]李澎森.异频法测量大型地网接地电阻的研究.高电压技术,2000,26 (6):37-39
- [13]Guangwu Wang, Wenjun Zhou. A frequency conversion measure system of grounding resistance. High Voltage Engineering, Eleventh International Symposium on,1999,2:369-371
- [14]JEAN J.Labrosse, Micro C/OS-II The Real Time Kernel Second Edition,北京航空航天大学出版社,2003
- [15]周立功等编著,ARM 与嵌入式系统基础教程,广州周立功单片机发展有限公司,2004
- [16]常美生,袁改莲.大型地网接地电阻测量的新方法.高电压技术.1995,21(2):68-70
- [17]应顺潮.接地电阻测量中引线互感误差的消除.高电压技术.1992,64(2):71-73
- [18]周文俊,文习山,汪广武.有效消除地网干扰的接地电阻变频测量法.高电压技术.1998 24(3):51-53
- [19]段绍辉.变频法测量 500kv 变电站地网接地电阻的分析.高电压技术.1993,19(2):43-46
- [20]邓南竹,周建,叶国强.用异频法测量地网参数.四川电力技术.2005,(1):44-48
- [21]解广润编.电力系统接地技术.北京:水利电力出版社,1991
- [22]戴胜军.嵌入式系统原理与应用.2007:1-18
- [23]杜普选,实时 DSP 技术及浮点处理器的应用,2006:1-16
- [24]Texas Instrument ,TMS320C3X User's Guide, 2000 :1-7
- [25]徐胤,基于 ARM 和 μ COS- II 的电力参数实时监测系统的设计, [学位论文]2005
- [26]张谦,竹利平. μ C/OS-II 实时嵌入式操作系统的实时性分析与测试.计算机工程与设计.2005,26(9):2422-2424
- [27]ARM Limited.ARM Developer Suite Version 1.2 CodeWarrior IDE Guide.2001:1.1-1.6
- [28]ARM Limited.ARM Developer Suite Version 1.2 AXD and arm Debuggers Guide. 2001 :1.1-1.14
- [29]ARM Limited.ARM Developer Suite Version 1.2 Developer Guide.2001:1.1-1.8

- [30] 孙东雷,基于 DSP 的点焊信号采集与 FFT 分析系统[学位论文]2005:29
- [31]于达仁,鲍文,徐敏强,用 FFT 测量周期信号相角差及其误差分析哈尔滨工业大学学报,1995, 27 (3): 84-87
- [32]薛蕙,杨仁刚,基于 FFT 的高精度谐波检测算法,中国电机工程学报,2002,22(12) 106-110
- [33]黄锐锋, 变电站接地网接地电阻近距离测量与短路电流分布计算. 2003:14-16

附录

模拟实验图

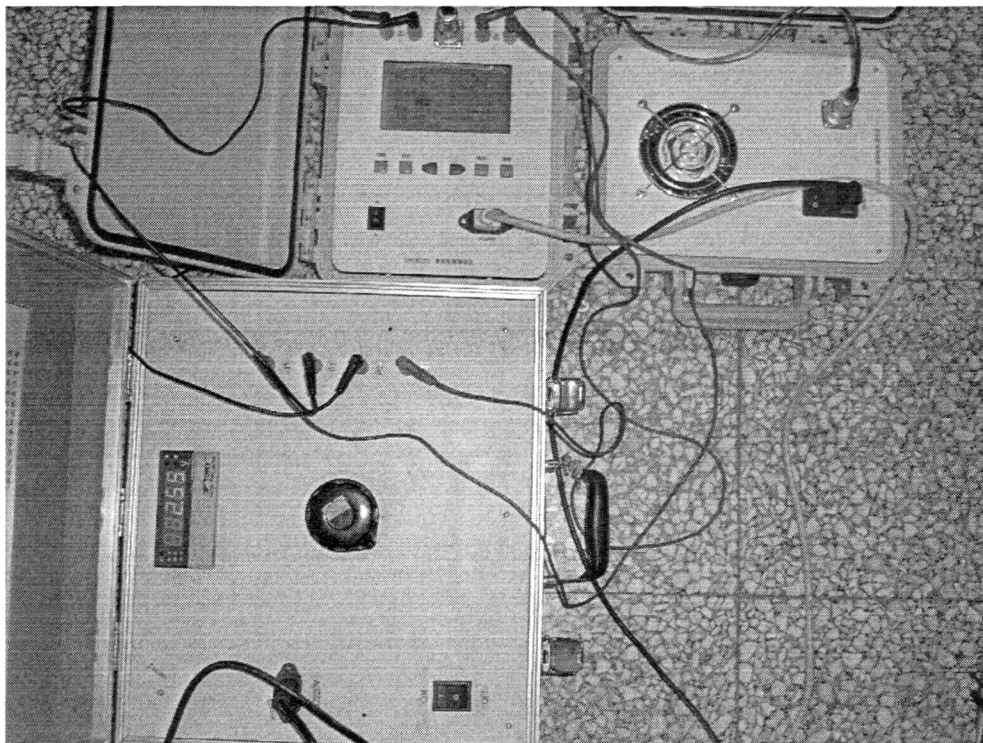


图 1

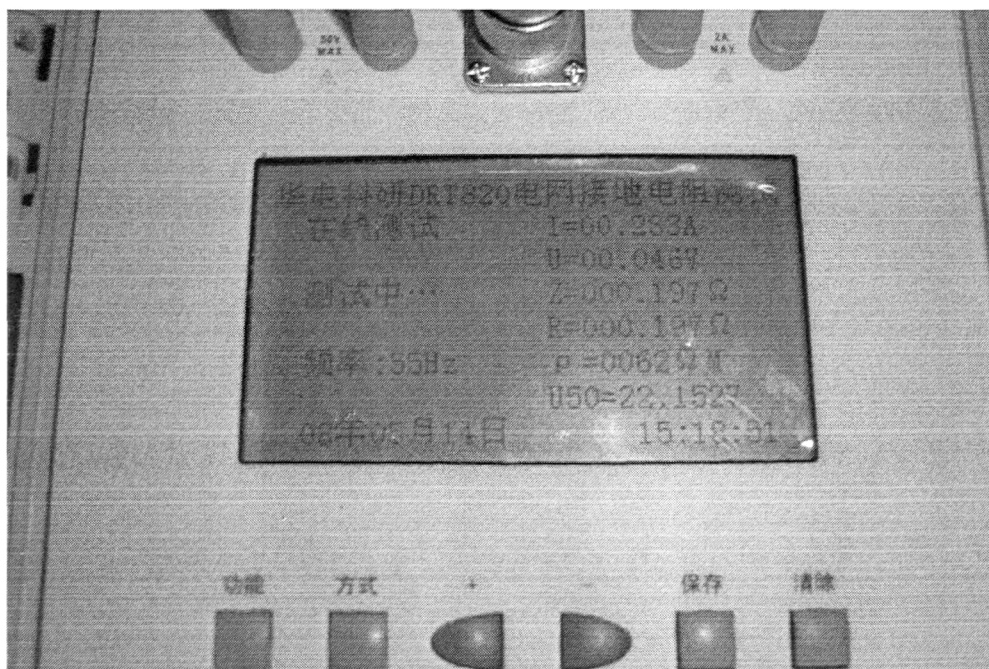


图 2

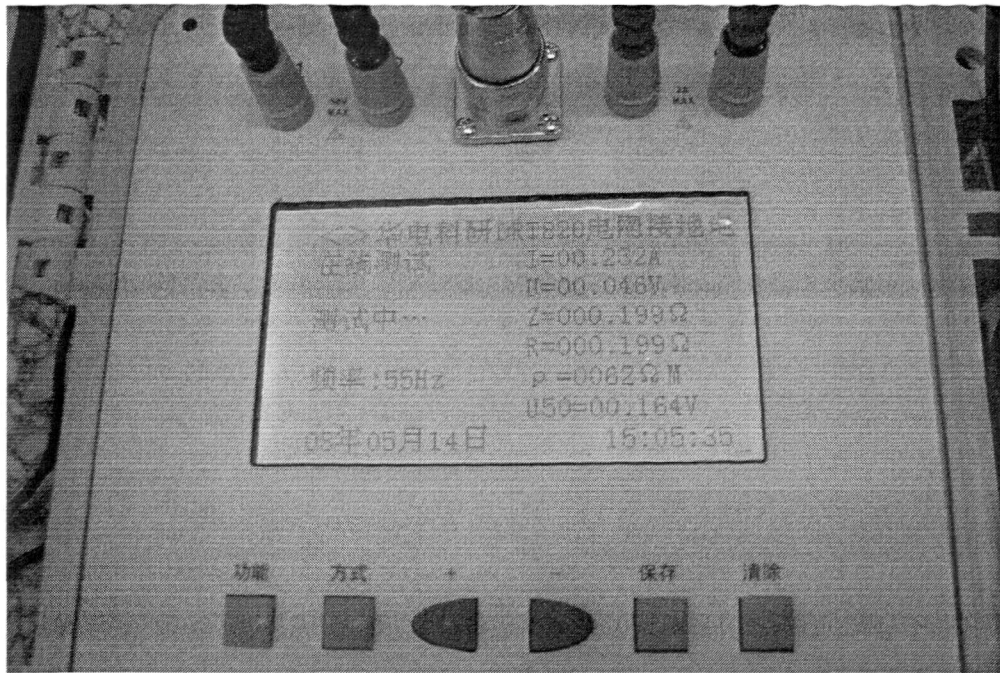


图 3

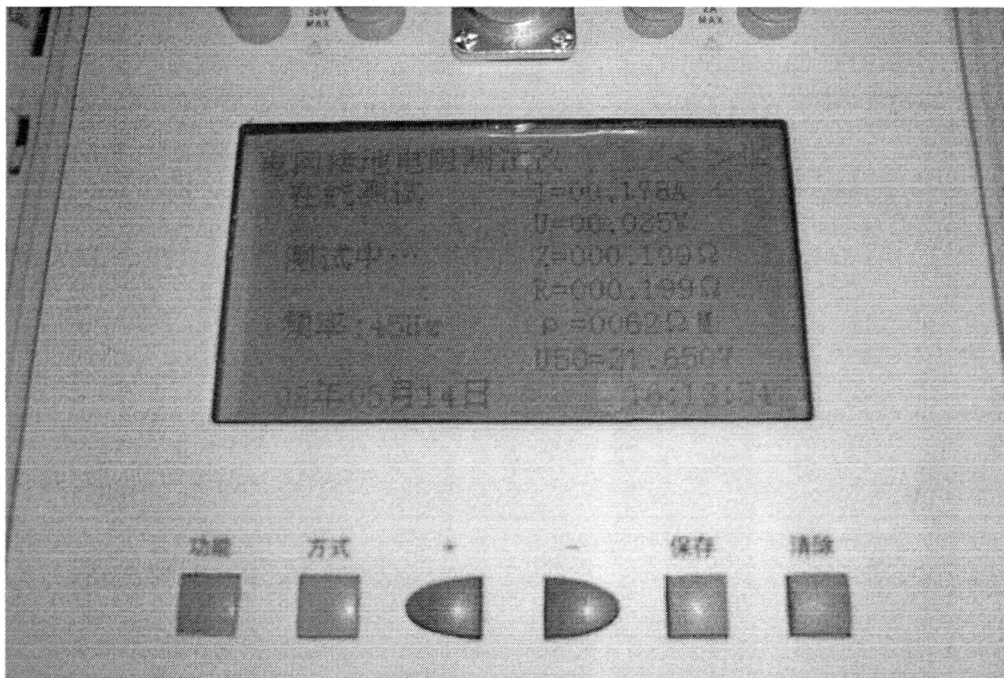


图 4

作者简历

贾淼，男，1983年2月出生，籍贯北京。2005年7月获得北京交通大学机械电子工程学院工学学士学位。2006年9月至今于北京交通大学电子信息工程学院攻读硕士学位，主要研究方向为数字信号处理、嵌入式技术应用。

攻读硕士学位期间实习经历和项目经历：

2007.5-2008.5 华北电力科学研究院实习

2006.5-2006.10 艾默生公司实习

2007.8-2008.5 参与了发电机绝缘过热监测装置 FJRIII的研发项目

2007.5-2007.9 参与了电网接地电阻测试仪的研发项目

2006.7-2006.9 充油质量测试项目