

摘 要

电能质量监测技术是当前电力系统领域的研究热点之一。对电能质量进行监测与分析从而提高和改善电能质量具有重要的现实意义。

本文根据电能质量监测对于系统算法和实时性的特殊要求，并在综合分析了目前电能质量监测现状的基础上，分析了一种基于 ARM 和 GPS 的电能质量在线监测系统的设计方案。以 S3C44B0X 芯片为主处理器，设计嵌入式硬件平台，采用全球定位系统（GPS）授时技术对系统进行同步处理，并选用源码公开的实时操作系统 $\mu\text{C}/\text{OS-II}$ 和 TCP/IP 协议栈 LWIP 构成软件平台。最后对系统软件设计进行了详细的说明。在此基础上，实现对电能质量的在线监测。

关键词：电能质量，ARM，GPS，在线监测

ABSTRACT

The power quality monitoring technology is one of the research focuses of the present electric power system field. There is great practical significance to improve the power quality by monitoring and analyzing.

In this paper, According to the special demand of the power quality monitoring for the system algorithm and real-time, on the basis of analyzing synthetically the present situation of power quality monitoring, a design scheme of power quality on-line monitoring system based on ARM and GPS is analyzed. The S3C44B0X chip is applied as the main processor to design embedded hardware platform. The system adopts GPS which is used to the clock synchronization of system. The system chooses the real-time operate system $\mu\text{C}/\text{OS-II}$ whose source code is open and the protocol stack LWIP of TCP/IP make the software platform. At last, the software design of system is illuminated in detail. On this foundation, it realizes on-line monitoring system for the power quality.

Li Hang (Communication and Information Systems)

Directed by associate prof. Xie Zhiyuan

KEY WORDS: power quality, ARM, GPS, on-line monitoring

摘 要

电能质量监测技术是当前电力系统领域的研究热点之一。对电能质量进行监测与分析从而提高和改善电能质量具有重要的现实意义。

本文根据电能质量监测对于系统算法和实时性的特殊要求，并在综合分析了目前电能质量监测现状的基础上，分析了一种基于 ARM 和 GPS 的电能质量在线监测系统的设计方案。以 S3C44B0X 芯片为主处理器，设计嵌入式硬件平台，采用全球定位系统（GPS）授时技术对系统进行同步处理，并选用源码公开的实时操作系统 $\mu\text{C}/\text{OS-II}$ 和 TCP/IP 协议栈 LWIP 构成软件平台。最后对系统软件设计进行了详细的说明。在此基础上，实现对电能质量的在线监测。

关键词：电能质量，ARM，GPS，在线监测

ABSTRACT

The power quality monitoring technology is one of the research focuses of the present electric power system field. There is great practical significance to improve the power quality by monitoring and analyzing.

In this paper, According to the special demand of the power quality monitoring for the system algorithm and real-time, on the basis of analyzing synthetically the present situation of power quality monitoring, a design scheme of power quality on-line monitoring system based on ARM and GPS is analyzed. The S3C44B0X chip is applied as the main processor to design embedded hardware platform. The system adopts GPS which is used to the clock synchronization of system. The system chooses the real-time operate system $\mu\text{C}/\text{OS-II}$ whose source code is open and the protocol stack LWIP of TCP/IP make the software platform. At last, the software design of system is illuminated in detail. On this foundation, it realizes on-line monitoring system for the power quality.

Li Hang (Communication and Information Systems)

Directed by associate prof. Xie Zhiyuan

KEY WORDS: power quality, ARM, GPS, on-line monitoring

声 明

本人郑重声明：此处所提交的硕士学位论文《基于 GPS 的配电网电能质量在线监测系统的研究》，是本人在华北电力大学攻读硕士学位期间，在导师指导下进行的研究工作和取得的研究成果。据本人所知，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，也不包含为获得华北电力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： 李航 日 期： 2008.3.16

关于学位论文使用授权的说明

本人完全了解华北电力大学有关保留、使用学位论文的规定，即：①学校有权保留、并向有关部门送交学位论文的原件与复印件；②学校可以采用影印、缩印或其它复制手段复制并保存学位论文；③学校可允许学位论文被查阅或借阅；④学校可以学术交流为目的，复制赠送和交换学位论文；⑤同意学校可以用不同方式在不同媒体上发表、传播学位论文的全部或部分内容。

(涉密的学位论文在解密后遵守此规定)

作者签名： 李航

导师签名： 谢志远

日 期： 2008.3.16

日 期： 08.3.17

第一章 引言

1.1 课题研究的背景和意义

从上个世纪 80 年代以来,现代电力系统中用电负荷结构发生了重大变化,非线性、冲击性和不平衡性的用电负荷急增,对供电电能质量造成了严重的干扰和污染,导致生产设备的损坏和产品废品率的增加,影响正常的工业生产和电网运行,并直接造成巨大的经济损失,而伴随着计算机等高新技术的发展,大量对电能质量非常敏感的新型电力负荷迅速发展,他们对电能质量提出了更高的要求。另外,在社会主义市场经济机制下,电能作为一种商品,参与市场竞争,就要符合商品的一般特征,即保持较高的质量。因此电能质量问题逐渐受到电力部门和电力用户的共同关注。

供电质量的保证同样关系着国家的工农业生产和社会经济发展,至 2000 年底我国结合国内外的研究成果和实际情况分别制定了五个国家电能质量系列标准:GB12325-90《供电电压允许偏差》;GB12326-2000《电压允许波动和闪变》;GB/T14549-93《公用电网谐波》;GB/T15543-95《三相电压允许不平衡度》;GB/T15945-95《电力系统频率允许偏差》。2001 年我国又制定了一项新的标准:GB/T18481-2001《暂时过电压和瞬态过电压》。^[8]

从上述标准可以看出:国家电能质量系列标准已将反映电能质量的几个主要指标包括在内,这些指标和电力系统正常运行状态密切相关。衡量系统是否达到国家电能质量的标准能为电力系统的设计提供参考,通过对电能质量的监测,可以为电力部门提供电力系统运行的基本状态和性能情况。据此可以了解公用电网的电能质量水平和存在的问题,从而对公用电网的性能做出正确和全面的评估。

在庞大的、瞬变的多输入输出的电力系统中,监测和控制的参数要求实时性较强,有些采集的特征量频率变化快而且复杂,如暂态突变量、高频的故障行波等,普通的采集处理方法对多路进行采集计算时,就会显得吃力甚至难以实现。随着电子技术的迅速发展,微处理器从运算速度、性能等方面都有了极大的提高。为了适应现代电力系统的要求,将基于 ARM 的嵌入式微处理器应用到电力系统来,充分发挥其快速强大的运算和处理能力以及并行运行的能力,满足了电力系统监测的实时性和处理算法的复杂性等更高的要求。

九十年代以来,基于全球同步卫星定位系统(GPS)的高精度定时技术逐步被引入电力系统,它可以实时和全天候的为全球任意位置的接收机提供高精度的三维位置、三维速度和时间信息,其时间误差小于 $1\mu\text{s}$,对于 50Hz 的工频信号其相位误差不超过 0.018° ,电力系统利用 GPS 的精确时间定位技术,对电网内运行的电气量进

行同步采样，从而保证了数据的同时性，有助于分析电力系统故障时刻的各种状态值。使整个供电系统能及早采取一定的控制措施避免事故的发生，也可以在事故状态下紧急投入控制措施防止事故扩大，保证系统安全运行。

随着国内电网规模的扩大和发展建设，电网的结构也发生了重大变化，当前又面临着电力市场等经营模式的改变，这将加剧电网稳定特性的复杂程度。因此，建立电能质量实时监测系统是必不可少的，它可对电能质量进行综合评估，为电能商业运行提供可靠的技术数据，并为保障规范用电及监督管理，为电能企业面向市场，适应市场竞争提供强有力的手段，而这些也必将在电力系统中起着越来越重要的作用。

1.2 国内外研究现状和发展趋势

在工业发达国家，电能质量问题早已被当作电力系统面临的重要问题看待，各国均在加强有关电能质量问题的研究，已得出不少理论成果，并提出一系列综合监测控制和管理方法。

对电能质量的监督有赖于科学合理的测量方法和准确可靠的测量仪器。近年来，随着对电能质量问题的重视和研究，国内外仪器制造厂推出了多种类型的电能质量监测与分析仪器（PQMA）。大体上分为以下三类：^[25]

(1) 远程监测仪器，该类仪表被定点安装在现场，连续监测公共供电点的一些主要电能质量指标，产生实时数据，以通信的方式把数据集中上传到管理机进行分析处理；

(2) 手持式仪表，由技术人员随身携带，定期测量或是随机测量一些电能质量指标，分析功能比较简单；

(3) 便携式多功能分析仪器，此类仪表特点是智能化、多功能、信息容量大、数据处理功能强，主要适用于现场专项测试，价格较高。

由于国外对电能质量的长期重视和科学技术上的整体优势，他们研制的电能质量监测设备水平普遍较高，且种类繁多。远程监测仪表以 reliable power meter 品牌的智能型电力品质记录仪的技术水平较高，它的实时性较强，可以固定在户外，适用于恶劣的环境。手持式仪表以美国福禄克公司生产的 FLUKE 43 型电能质量分析仪最具代表。这些仪器多是采用硬件 DSP 技术对电信号进行分析处理的。

过去由于电力系统采用的同步时钟精度低和接收困难，再加上通讯技术和计算机技术水平的限制，使得同步时钟的误差偏大。美国的全球定位系统 GPS 的出现，提供了时钟精度上的保证。因此现有相量测量装置都选用 GPS 作为同步时钟。1990 年 A. G. Phadke 研制了基于 GPS 时钟的同步相量测量装置，并将其应用于两个变电站之间的连线上。1990 年，法国也研制了基于 GPS 时钟的同步相量测量装置。目

前随着电力系统规模的不断扩大,大容量、超高压、远距离输电日益增多,系统结构也日益复杂,从而电力系统的时钟同步就显得越来越重要。因此, GPS 技术在电力系统中得到了广泛应用。在电能质量监测系统中也发挥了重要作用。

随着集成电路技术和计算机技术的发展,网络技术和嵌入式实时系统技术的日益成熟和完善,电能质量监测系统正朝着在线监测、实时分析、网络化和智能化的方向发展。

1.3 本文完成的主要工作

(1) 分析了电能质量在线监测系统的研究背景、发展现状,明确了课题的研究意义和实用价值。

(2) 阐述了电能质量监测系统的基本要求,并对电能质量的各项指标进行了详细的分析,研究了各项指标的定义、测量方法,选择 FFT 作为电能质量的数学分析算法。

(3) 在系统硬件设计中,以 ARM 处理器 S3C44B0X 为中心,搭建系统硬件平台,并设计了各部分的电路。

(4) 本系统利用 GPS 的授时功能提供一个高精度的时钟信号,本文深入分析了 GPS 时间信息的提取。

(5) $\mu\text{C}/\text{OS-II}$ 作为嵌入式实时操作系统,分析研究了它的结构和特点,并说明了如何将它移植在 S3C44B0X 处理器上。此外, LWIP 作为 TCP/IP 协议栈,对它的特点进行了详细说明以及介绍了 LWIP 如何在 $\mu\text{C}/\text{OS-II}$ 上实现所做的工作。

(6) 在系统软件设计中,对其应用软件部分进行了详细说明,对数据采集处理部分、GPS 部分、网络通信传输部分、LCD 显示部分和键盘部分的软件进行了设计,给出了各部分的流程图。

第二章 电能质量监测系统的要求和实现

2.1 电能质量监测系统的要求

电能质量问题不仅仅关系到用电设备运行的可靠性和安全性，而且还关系到供电市场的规范性。电能质量问题的产生来源于供电方、用电端或雷电等自然现象，只有对电能质量进行有效的监测才会对问题的产生和影响有清楚的认识，才能为电能质量的改善、供用电双方的协调和供用电市场的规范化提供真实依据，以便采取有效的解决措施。因此要求电能质量监测仪器能够真实有效的测量电能质量的各项指标。随着电能质量监测仪器的广泛应用，电能质量数据采集、管理及分析的任务也日益显得繁重，所以人们对电能质量监测系统提出了更高的要求，即要求对整个监测系统实现网络化、智能化管理，后台管理分析软件具有强大的分析功能。

电能质量监测仪器的传统设计大多采用工业计算机配备数据采集卡来实现电能质量的数据采集和分析，主要针对稳态指标进行监测，系统的实时性难以保证，智能化和网络化水平也不高。一方面是由于人们对电能质量的认知水平还没有达到一定的高度，另一方面也受到当时技术条件的限制。

随着电能质量对国民经济的影响逐渐加大和人们对电能质量研究的逐渐深入，人们对电能质量关注的焦点已不仅仅是电压、频率和谐波等各种稳态指标，还包括影响电能质量的实时信息，如瞬时扰动和暂态谐波等，同时也要求电能质量监测系统提供更为直观的分析结果，以利于对电能质量问题做出决策，如要求系统能够进行故障识别、干扰源识别、故障预测和信息共享等。如今电子技术和网络技术的飞速发展使得这种需要的实现成为可能。为满足电力系统网络化、自动化的发展需要，电能质量监测系统正在朝着在线监测、实时分析、网络化和智能化的方向发展。

从设备的考虑角度，目前电能质量监测的发展趋势倾向于采用永久的固定设备对现场数据进行在线监测。对设备的具体要求可概括为：

- (1) 设备要能满足实时性的要求，具备对电网问题的快速捕获能力。
- (2) 在分析手段上要能够对采样数据预处理，比如小波去噪，能对数据实时分析，对瞬时和稳态干扰性时间进行跟踪和预分析，这样可以降低对存储容量的要求，同时为现场的实时监控提供更快速、更有价值的评估和决策信息。
- (3) 要具备强大的通信能力，能方便集成到企业信息管理系统中和互联网上，以便于为电能质量的深入分析、事件统计、长期评估和预测提供可靠数据。
- (4) 在功能上要具备配置灵活性，以适应电力系统的不同应用场合。
- (5) 在成本上要适合于大量安装到现场。

从系统的角度考虑，文献[10][11]都提到了智能电能质量的概念，但对此尚未

有明确的定义。文献[10]给出了智能系统的功能模型和应用领域，文献[11]给出了智能体系的基本特征，这代表了智能电能检测系统发展的最新方向。综合以上论述，可以归纳出智能系统除了具备传统的监测功能外，在功能和技术上更加强调：各种信息的提取、收集、分类和管理，分析手段的智能化，电网运行性能和条件的自动评估，为用户提供决策支持，对问题能够进行预测。

综上所述，人们对电能质量的要求不断提高，也对监测系统在功能上提出了更高的要求，同时也表明这一应用领域的研究需要多种技术的相互融合和各个领域专家的密切合作。

2.2 电能质量的测量方法

电能质量是指公用电网供到用户受电端的交流电能质量。我国根据国情，已经制定了六项电能质量国家标准，目前的研究主要集中在如下五项指标：（1）供电电压允许偏差；（2）公用电网谐波；（3）三相电压允许不平衡度；（4）电力系统频率允许偏差；（5）电压波动和闪变。对于暂时过电压和瞬态过电压，由于标准出现的较晚，该部分的研究也相对较少。

电能质量把电看成一种商品，对它的各项技术指标做出规定，以判断其是否合格。总的来说，使电能指标越接近额定值就是电能高质量的标准。下面就五项指标的测量方法进行阐述。

2.2.1 基本参数测量

电能质量标准中虽然不包括各种功率及功率因数的监测指标，但有功功率、无功功率、视在功率、功率因数以及电压、电流有效值都是电力系统的重要参数^[7]。

采用数字化方法，就要对电气量进行离散化采样，一般是进行等间隔采样，得到离散化序列 $\{u_k\}$ $\{i_k\}$ ，则电压有效值的计算公式为：

$$U = \sqrt{\frac{1}{N} \sum_{k=0}^{N-1} u_k^2} \quad (2-1)$$

式中， u_k —第 k 个时间间隔的电压采样值；

N —一个周期内的采样点数。

电流有效值的计算公式为：

$$I = \sqrt{\frac{1}{N} \sum_{k=0}^{N-1} i_k^2} \quad (2-2)$$

有功功率：

有功功率 P 的定义为

$$P = \frac{1}{T} \int_0^{T} P(t) dt = \frac{1}{T} \int_0^{T} u(t)i(t) dt \quad (2-3)$$

若每周周期采样 N 点，则上式离散化：

$$P = \frac{1}{N} \sum_0^{N-1} u_k i_k \quad (2-4)$$

视在功率：

$$\text{视在功率 } S \text{ 的定义为：} \quad S = U * I \quad (2-5)$$

U、I 为电压、电流的有效值。

三相功率为：

$$P = \frac{1}{N} \left(\sum_0^{N-1} u_{Ak} i_{Ak} + \sum_0^{N-1} u_{Bk} i_{Bk} + \sum_0^{N-1} u_{Ck} i_{Ck} \right) \quad (2-6)$$

$$S = U_A I_A + U_B I_B + U_C I_C \quad (2-7)$$

$$\text{功率因数：} \quad \cos \phi = P / S \quad (2-8)$$

2.2.2 频率的测量^[6]

电力系统的频率偏差是指电力系统正常运行条件下，系统频率的实际值与标称值（50Hz 或 60Hz，我国采用 50Hz 标准）之差。用公式表示为：

$$\delta f = f_{re} - f_N \quad (2-9)$$

电力系统频率一方面作为衡量电能质量的指标，需要加以监测，另一方面作为实施安全稳定控制的重要状态反馈量，要求能实施重构。另外，频率同步也是实现高精度电气量测量的重要措施之一。

关于频率测量的研究，国内外已经有很多的成果。

进行频率测量的算法主要有：

(1) 周期法：周期法即为零交法。通过测量信号波形相继过零点间的时间宽度来计算频率。

(2) 解析法：通过对信号观测模型进行数学交换，将待测量 f 或 Δf 表示为样本值的显函数来估计，但精度总体不高。

(3) 误差最小化原理类算法：包括最小二乘算法、最小绝对值近似法、牛顿类算法、离散卡尔曼滤波算法。

(4) DFT(FFT)类算法及其改进算法。

(5) 正交调制法。

本文采用基于简单信号观测模型的周期法测量频率信号，周期法物理概念清晰，易于实现。

电压信号普遍取单一的相电压或相电流（ $x(t)=A_m \sin(2\pi ft + \theta)$ ）经低通滤波器滤

除掉高次干扰成分后，送入到比较器得到与电压信号频率相同的方波信号输出，经可编程逻辑器件自动完成计数，从而计算出频率。

$$f = \text{计数基准频率} / \text{测频计数器} \quad (2-10)$$

信号周期 $T = 1/f$ ，将信号周期 N 等分，就可得到信号的采样间隔^{[10][17]}，从而实现采样频率对系统频率的跟踪。

2.2.3 电压偏差的测量

电压允许偏差是指电力系统电压缓慢变化时，实际电压与系统标称电压之差。通常指电压变化率小于每秒 1% 时实际电压值与系统标称电压之差。

$$\text{电压偏差}(\%) = \frac{\text{实际电压} - \text{额定电压}}{\text{额定电压}} \times 100\% \quad (2-11)$$

式中实际电压为实际测量电压，额定电压为系统标称电压 220V 或 380V。

电压变动指的是供电点电压在两个相邻、持续一定时间的电压均方根值 U_1 和 U_2 之间的差值。通常以额定电压的百分数来表示电压变动的相对百分值 ΔV ，即：

$$\Delta V = \frac{U_1 - U_2}{U_N} \times 100\% \quad (2-12)$$

在同一方向小于 30ms 的快速变化不计入电压变动。小于 30ms 的期间内，同方向的电压均方根值的变动只算作一次变动。在单位时间内电压变动的次数称为电压变动的频度。一般以 1/min 或 1/s 为单位。

2.2.4 三相不平衡度的测量^[5]

三相不平衡度的测量建立在各序分量的计算上。电力系统三相不对称时，三相电路的电压和电流的基频分量都呈现为不对称的分量。根据对称分量法可以将三相不对称的分量唯一的分解为 3 组对称的分量，即正序分量、负序分量和零序分量。以电压为例，A、B、C 三相电压的零序 U_0 、正序 U_1 、负序 U_2 的公式如下：

$$\begin{cases} \dot{U}_0 = (\dot{U}_A + \dot{U}_B + \dot{U}_C) / 3 \\ \dot{U}_1 = (\dot{U}_A + \alpha \dot{U}_B + \alpha^2 \dot{U}_C) / 3 \\ \dot{U}_2 = (\dot{U}_A + \alpha^2 \dot{U}_B + \alpha \dot{U}_C) / 3 \end{cases} \quad (2-13)$$

式中 \dot{U}_A 、 \dot{U}_B 、 \dot{U}_C 为 A、B、C 三相电压矢量， α 为旋转因子，三相电压不平衡度通常以负序分量与正序分量均方根值的百分比来表示。

$$\varepsilon = \frac{U_2}{U_1} \times 100\% \quad (2-14)$$

结合 (2-14) 式便可得到三相电压、电流的不平衡度。

不平衡度的 3 秒均方根值：

$$\varepsilon = \sqrt{\frac{1}{m} \sum_{k=1}^m (\varepsilon_k)^2} \quad (2-15)$$

式中 ε_k ：在 3 秒内第 k 次测得的不平衡度；

m ：在 3 秒内均匀间隔取值次数， $m=6$ 。

不平衡度的统计测量方式和谐波的统计测量方式相似。

采用上述方法计算较为繁琐，国际中给出了几个简化的计算公式，但都有其限定的应用条件。

$$\text{公式：} \varepsilon = \sqrt{\frac{1 - \sqrt{3 - 6L}}{1 + \sqrt{3 - 6L}}} \times 100(\%) \quad (2-16)$$

式中： $L = (a^4 + b^4 + c^4) / (a^2 + b^2 + c^2)$ ， a 、 b 、 c 为三相量。

该公式适用于没有零序分量的三相系统中。而且计算繁琐，不适用于实时运算。另有不平衡度的近似计算公式：

$$\varepsilon_u = \frac{\sqrt{3} I_2 * U_L}{10 S_k} (\%) \quad (2-17)$$

式中： I_2 —电流的负序值，A； S_k —公共连接点的短路容量，MV·A；
 U_L —线电压，KV。

该公式的前提是假设公共连接点的正序阻抗与负序阻抗相等。

2.2.5 基于复序列 FFT 的谐波测量

在谐波测量中^[3]，需要用数字方法对采集的电压电流信号进行谐波分析，针对不同类型的谐波有不同的分析方法，对于稳态谐波通常用 DFT、FFT、加窗法，此外还有 Hartley 变换 (FHT) 算法、离散 W 变换等。本文只讨论对整数次、准稳态谐波的测量，这对于离散化的周期信号而言，快速傅立叶变换 (FFT) 无疑是分析谐波的最好算法之一，利用 FFT 可以直接得到波形所含的各频谱分量。为了满足谐波分析的实时性要求，从尽量减少数据分析的运算量的角度出发，本文分析了一种基于复序列 FFT 算法来实现谐波的测量，该算法与 FFT 比较，分析同样序列长度的电压电流数据，其运算量减少了一半。

1、基于复序列的 FFT 算法推导

基于复序列的 FFT 算法是 FFT 的改进算法。当输入信号 $f(t)$ 可为周期函数或可近似地作为周期函数处理时，且满足狄利赫里条件（电力系统信号均满足），则它可被分解为一个各种频率的正弦函数序列之和，即傅立叶级数，其三角级数形式为：

$$f(t) = c_0 + \sum_{h=1}^{+\infty} a_h \cos(h\omega_1 t) + \sum_{h=1}^{+\infty} b_h \sin(h\omega_1 t) \quad (2-18)$$

$$\begin{aligned}
 \text{式中: } \omega_1 &= \frac{2\pi}{T}; & c_0 &= \frac{1}{T} \int_0^T f(t) dt, & c_h &= \sqrt{a_h^2 + b_h^2}; \\
 \omega_h &= \tan^{-1} \frac{a_h}{b_h}, & b_h &> 0; & a_h &= \frac{2}{T} \int_0^T f(t) \cos(h\omega_1 t) dt, & h=1, 2, \\
 &= \tan^{-1} \frac{a_h}{b_h} + \pi, & b_h &< 0; & b_h &= \frac{2}{T} \int_0^T f(t) \sin(h\omega_1 t) dt, & h=1, 2, \dots
 \end{aligned}$$

若设 $F_0 = c_0$, $F_h = 1/2(a_h - jb_h)$, $h=1, 2, \dots$

$$\text{则 } F_h = \frac{1}{T} \int_0^T f(t) (\cosh \omega_1 t - j \sinh \omega_1 t) dt \quad (2-19)$$

将欧拉公式 $e^{j\theta} = \cos \theta + j \sin \theta$ 代入上式可得:

$$F_h = \frac{1}{T} \int_0^T f(t) e^{-jh\omega_1 t} dt, \quad h=0, 1, 2, \dots \quad (2-20)$$

上式即为傅立叶级数的复数形式。

对信号 $f(t)$ 进行每周波均匀采样 N 个点时, 将上式进行离散化处理得:

$$F_h = \frac{1}{N} \sum_{n=0}^{N-1} f(n) e^{-jhn\frac{2\pi}{N}}, \quad h=0, 1, 2, \dots \quad (2-21)$$

$$\text{令 } W_N = e^{-j\frac{2\pi}{N}}, \quad \text{则 } F_h = \frac{1}{N} \sum_{n=0}^{N-1} f(n) W_N^{nh}, \quad h=0, 1, 2, \dots \quad (2-22)$$

上式为傅立叶级数的离散形式, 称为离散傅立叶变换 (DFT)。

若直接用离散傅立叶变换 (DFT) 进行计算, 当采样点很多时, 计算量使很大的, FFT 是 DFT 的一种快速算法, 它利用 W_N^{nk} 的对称性 $(W_N^{nk})^* = W_N^{-nk}$ 和周期性 $W_N^{nk} = W_N^{(n+1)k} = W_N^{n(k+N)}$, 将长序列的 DFT 分解为若干短序列的 DFT 进行计算。

FFT 算法基本可分为两大类即按时间抽取 (DIT) 法和按频率抽取 (DIF) 法。

以 8 点的输入序列为例, 从基 2 DIT 算法的流图中可以看出, DIT 法是按序列在时间上的次序是属于偶数还是属于奇数来分解为两个更短的子序列, 直到分解为 2 点的 DFT 为止。两点的 DFT 称为一个蝶形运算, 每个蝶形运算需要一次复数乘法和两次复数加 (减) 法。直接 DFT 算法复数乘法次数为 N^2 , 而 FFT 算法需复数乘法次数为 $(N \cdot \log_2 N) / 2$, 可见 FFT 算法运算量要小得多, 尤其是点数 N 越大时 FFT 优点越突出。

由于 DFT 或 FFT 运算都是以复数信号进行处理的, 而实际中需处理的一般都是实数信号, 直接用 FFT 对实数信号进行处理是很不经济的。在实际处理序列时, 可将两个同长度的实序列分别作为复序列的实部和虚部来进行处理, 这就叫做基于复序列的 FFT 算法。

设两个实序列 $x_1(n), x_2(n)$, 令 $y_1(n) = x_1(n) + jx_2(n), y_2(n) = x_1(n) - jx_2(n)$, 可以得

$$\text{出: } x_1(n) = \frac{1}{2}[y_1(n) + y_2(n)], x_2(n) = \frac{1}{2j}[y_1(n) - y_2(n)] \quad (2-23)$$

$y_1(n)$ 的傅立叶变换 $Y_1(k)$ 为:

$$\begin{aligned} Y_1(k) &= X_{1R}(k) + jX_{1I}(k) + j[X_{2R}(k) + jX_{2I}(k)] \\ &= X_{1R}(k) - X_{2I}(k) + j[X_{1I}(k) + X_{2R}(k)] \end{aligned} \quad (2-24)$$

其中 $X_{1R}(k)$ 、 $X_{1I}(k)$ 、 $X_{2R}(k)$ 、 $X_{2I}(k)$ 分别为 $X_1(k)$ 、 $X_2(k)$ 的实部和虚部。

$y_2(n)$ 的傅立叶变换 $Y_2(k)$ 为:

$$\begin{aligned} Y_2(k) &= X_{1R}(k) + jX_{1I}(k) - j[X_{2R}(k) + jX_{2I}(k)] \\ &= X_{1R}(k) + X_{2I}(k) + j[X_{1I}(k) - X_{2R}(k)] \end{aligned} \quad (2-25)$$

$$\text{由式(2-24)得: } Y_1^*(k) = X_{1R}(k) - X_{2I}(k) - j[X_{1I}(k) + X_{2R}(k)] \quad (2-26)$$

$$\text{则: } Y_1^*(N-k) = X_{1R}(N-k) - X_{2I}(N-k) - j[X_{1I}(N-k) + X_{2R}(N-k)] \quad (2-27)$$

根据傅立叶变换的周期性和奇偶虚实性, 可知实序列的傅立叶变换的实部为偶函数, 虚部为奇函数。可得:

$$\begin{aligned} Y_1^*(N-k) &= X_{1R}(N-k) - X_{2I}(N-k) - j[X_{1I}(N-k) + X_{2R}(N-k)] \\ &= X_{1R}(-k) - X_{2I}(-k) - j[X_{1I}(-k) + X_{2R}(-k)] \\ &= X_{1R}(k) + X_{2I}(k) + j[X_{1I}(k) - X_{2R}(k)] = Y_2(k) \end{aligned} \quad (2-28)$$

对式(2-23)进行FFT并考虑其复共扼性质, 则可得到 $x_1(n)$ 、 $x_2(n)$ 的频谱为:

$$\begin{aligned} X_1(K) &= [Y_1(K) + Y_2(K)]/2 = [Y_1(K) + Y_1^*(N-K)]/2 \\ X_2(K) &= [Y_1(K) - Y_2(K)]/2j = [Y_1(K) - Y_1^*(N-K)]/2j \end{aligned} \quad (2-29)$$

式中: Y_1^* 为 Y_1 的共扼复数, 这就是复序列FFT算法原理。

2、基于复序列FFT算法的电参量测量

设畸变电压和畸变电流含有L次谐波的形式

$$u(t) = \sum_{n=1}^L u_n(t) = \sum_{n=1}^L U_{nm} \sin(n\omega t + \alpha_n) \quad (2-30)$$

$$i(t) = \sum_{n=1}^L i_n(t) = \sum_{n=1}^L I_{nm} \sin(n\omega t + \beta_n) \quad (2-31)$$

现在若已经测量到N点电压序列 $\{u(n)\}$ 和电流序列 $\{i(n)\}$, 则可构造一个复序列:

$$x(n) = u(n) + ji(n) \quad (0 \leq n \leq N-1) \quad (2-32)$$

对于复序列 $\{x(n)\}$, 其离散傅立叶变换为:

$$X(k) = DFT[x(n)] = \frac{1}{N} \sum_{n=0}^{N-1} [x(n)e^{-j(2\pi/N)nk}] \quad (2-33)$$

由式(2-32)可得:

$$u(n) = \frac{1}{2}[x(n) + x^*(n)] \quad i(n) = \frac{1}{2j}[x(n) - x^*(n)] \quad (2-34)$$

对式 (2-34) 进行 DFT 并考虑其共扼性质, 则可得到电压、电流的频谱为:

$$U(k) = \frac{1}{2}[X(k) + X^*(N-k)] \quad I(k) = \frac{1}{2j}[X(k) - X^*(N-k)] \quad (2-35)$$

式中 $x^*(n)$ 、 $X^*(N-k)$ 为别为 $x(n)$ 、 $X(N-k)$ 的共扼复数。

设 $u(t)$ 是一个仅含有第 q 次谐波的正弦电压信号, 则由式 (2-30) 得:

$$u_q(t) = U_{qm} \sin(q\omega t + \alpha_q) \quad (2-36)$$

其相量表示为: $\dot{U}_q(t) = U_{qm} \angle \alpha_q$ (2-37)

当在一个信号周期内对 $u_q(t)$ 进行 N 点等间隔采样时, 则有:

$$u_q(n) = U_{qm} \left(\frac{2\pi qn}{N} + \alpha_q \right) \quad (2-38)$$

对于非正弦周期信号的离散傅立叶变换, 若已知最高次谐波为 L , 则在一个信号周期内的采样点数 N 应为: $N > 2L$ 。在此, 对一个仅含有第 q 次谐波的正弦电压信号而言, 应满足 $N > 2q$ 。因 q 为正整数, 故 $1 < q < N/2 - 1$ 。容易证明电压频谱 $U(q)$ 为:

$$U(q) = DFT[u_q(n)] = \frac{1}{2j} U_{qm} (\cos \alpha_q + j \sin \alpha_q) \quad (2-39)$$

其相量表示为: $\dot{U}_q = \frac{1}{2j} \bullet U_{qm} \angle \alpha_q$ (2-40)

由式 (2-37) 和 (2-40) 可得电压、电流 (推导方法同电压) 相量与其频谱的关系为: $\dot{U}_q = 2jU(q)$ $\dot{I}_q = 2jI(q)$ (2-41)

由式 (2-41) 和 (2-35) 可导出各次 ($1 < q < N/2 - 1$) 谐波电压和谐波电流的有效值和功率如下:

$$U_k = \frac{1}{\sqrt{2}} \sqrt{[X_I(N-k) - X_I(k)]^2 + [X_R(k) + X_R(N-k)]^2} \quad (2-42)$$

$$I_k = \frac{1}{\sqrt{2}} \sqrt{[X_I(N-k) + X_I(k)]^2 + [X_R(k) - X_R(N-k)]^2} \quad (2-43)$$

$$S_k = P_k + jQ_k = \frac{1}{2} \dot{U}_k \bullet \dot{I}_k = \frac{2}{N^2} U(k) \bullet I^*(k) \quad (2-44)$$

$$P_k = \frac{1}{N} [X_R(k) - X_I(N-k) + X_I(k) \bullet X_R(N-k)] \quad (2-45)$$

$$Q_k = \frac{1}{2N} [X_R^2(k) - X_I^2(k) - X_R^2(N-k) - X_I^2(N-k)] \quad (2-46)$$

式中 X_R 、 X_I 分别为 X 的实部和虚部。

根据以上结果，可得电压和电流有效值、有功功率、无功功率、功率因数分别为：

$$U = \sqrt{\sum_{n=1}^L U_n^2}、I = \sqrt{\sum_{n=1}^L I_n^2}、P = \sum_{n=1}^L P_n、Q = \sum_{n=1}^L Q_n、\cos\varphi = \frac{P}{\sqrt{P^2 + Q^2}}$$

从以上推论可知，在整个过程中只需对式（2-27）进行一次 FFT 变换，就可以完成谐波、电压电流有效值和功率的计算，与传统的 FFT 比较，对相同长度电压、电流采样序列进行谐波分析，采用基于复序列 FFT 算法可以把 FFT 分析运算量减半，这样大大提高了数据处理的效率。

2.2.6 电压波动和闪变的测量

电压波动^[21]与闪变的测量^[20]，一般是测量等值 10Hz 电压闪变值 ΔV_{10} 、短时间闪变严重程度 P_{st} 及长时间闪变严重程度 P_{lt} 。

常用的电压波动检测方法有整流检测法、有效值检测法和同步检测法。

为检测出电压波动分量，通常将电压波动看成以工频电压为载波，其电压的均方根值或峰值受到以电压波动分量作为调幅波的调制。对于任何波形的调幅波均可看作是由各种频率分量合成。下面一般性的分析仅含单一频率的调幅波对工频载波的调制。调制波解析式的一般表达式为：

$$u(t) = [U_m + v(\omega_F t)] \cos \omega_N t \quad (2-47)$$

式中： U_m —工频载波电压的幅值，V； ω_N —工频载波电压的角频率，rad/s；

v —调幅波电压，V； ω_F —调幅波电压的角频率，rad/s。

若调幅波电压为单一频率的正弦波形，则 $v(\omega_F t) = V_m \cos \omega_F t$ ，将其带入一般表达式（2-47），则有：

$$u(t) = U_m [1 + \frac{V_m}{U_m} \cos \omega_F t] \cos \omega_N t = U_m [1 + m \cos \omega_F t] \cos \omega_N t \quad (2-48)$$

式中， m 称为调制指数， $m = \frac{V_m}{U_m} = \frac{\text{调制波电压幅值}}{\text{载波电压幅值}}$ ，要求 $m < 1$ ，否则将出现包络线畸变。

按照同步检测方法，可将调幅波电压自乘求平方，得到：

$$\begin{aligned} u^2(t) &= U_m^2 (1 + 2m \cos \omega_F t + m^2 \cos^2 \omega_F t) \cos^2 \omega_N t \\ &= \frac{U_m^2}{2} (1 + \frac{m^2}{2}) + U_m^2 m \cos \omega_F t + \frac{U_m^2}{2} (1 + \frac{m^2}{2}) \cos \omega_N t - \frac{U_m^2 m^2}{4} \cos 2\omega_F t \\ &\quad + \frac{U_m^2 m^2}{8} \cos 2(\omega_N + \omega_F)t + \frac{U_m^2 m^2}{8} \cos 2(\omega_N - \omega_F)t \end{aligned}$$

$$+ \frac{U_m^2 m}{2} \cos(2\omega_N + \omega_F)t + \frac{U_m^2 m}{2} \cos(2\omega_N - \omega_F)t \quad (2-49)$$

从上式可以看出，调制电压的平方项除了有直流成分外，含有以下频率分量： ω_F 、 $2\omega_F$ 、 $2\omega_N$ 、 $2(\omega_N \pm \omega_F)$ 。

利用 0.05~35Hz 的带通滤波器滤除其中的直流分量和工频及以上频率的分量，且考虑到实际中存在的调幅波电压的倍频分量幅值远小于调幅波的幅值，可忽略不计。因此滤波后可实现解调，获得近似加权的调幅波电压。

$$v(t) \approx mU_m^2 \cos \omega_F t = U_m V_m \cos \omega_F t \quad (2-50)$$

已知相对电压变动值为 $\frac{\Delta V}{U_N}$ ，并且假定调幅波为正弦函数波形，则有：

$$m = \frac{V_m}{U_m} = \frac{\Delta V}{2\sqrt{2}U_N} = \frac{1}{2\sqrt{2}} d \quad (2-51)$$

将它带入式 (2-50)，可以得到用相对电压变动 d 参量表示的表达式：

$$v(t) \approx mU_m^2 \cos \omega_F t = 0.35U_m^2 d \quad (2-52)$$

由调幅波得到电压波动值 ΔU_f ，并由 ΔU_f 得到 10Hz 等值电压闪变的百分数：

$$\Delta U_{10} = \sqrt{\sum (\alpha_f \Delta U_{f1})^2} \quad (2-53)$$

式中， α_f 为闪变视感度系数， ΔU_{f1} 为电压调幅波中频率为 f 的正弦波分量一分钟均方根值。

2.3 小节

本章说明了电能质量监测系统在设备、功能等方面的基本要求，并且从电能质量的基本概念出发，对衡量电能质量的五项指标，包括电压偏差、频率、谐波、三相不平衡度、电压波动和闪变进行了阐述和深入探讨。文章叙述了五项指标的定义、数学表达和测量方法，在对谐波测量进行分析时，分析了一种基于复序列 FFT 的算法，该算法可使运算量减半，大大提高了数据处理的速度。

第三章 电能质量监测系统的硬件设计

3.1 基于 ARM 的嵌入式系统^[1]

从 20 世纪 70 年代初出现第一个微处理器开始,嵌入式系统的发展已经有近 30 年的历史了。随着计算机技术和产品对其它行业的广泛渗透,以应用为中心的分类方法变得更为切合实际,也就是按计算机的嵌入式应用和非嵌入式应用将其分为嵌入式计算机和通用计算机。通用计算机具有计算机的标准形式,通过装配不同的应用软件,出现并应用在社会的各个方面,其典型产品为 PC;而嵌入式计算机则是以嵌入式系统的形式隐藏在各种装置、产品和系统中。嵌入式系统将在“后 PC”时期大显身手。嵌入式系统是根据应用的要求,将操作系统和功能软件集成于计算机硬件系统之中,从而实现软件与硬件一体化的计算机系统,具有小巧、高度自动化、响应速度快等特点。

3.1.1 ARM 微处理器介绍

ARM(Advanced RISC Machines),既可以认为是一个公司的名字,也可以认为是对一类微处理器的通称,还可以认为是一种技术的名字。

1985 年 4 月 26 日,第一个 ARM 原型在美国剑桥的 Acron 计算机有限公司诞生,由美国加州 San Jose VISI 技术公司研制。20 世纪 80 年代后期,ARM 很快开发出 Acron 的台式机产品,1990 年成立了 Advanced RISC Machines Limited。20 世纪 90 年代,ARM32 位嵌入式 RISC(Reduced Instruction Set Computer)处理器扩展到世界范围,占据了低功耗、低成本和高性能的嵌入式系统应用领域的领先地位。ARM 公司使用通用的基础体系结构,以极低的成本和功耗提供了高性能、多系列的 32 位 RISC 处理器核。ARM 处理器在耗电、数据传送、数据处理速度以及带 DSP 功能等方面处于领先地位。

到目前为止,ARM 微处理器及技术的应用几乎已经深入到各个领域,包括工业控制领域、无线通讯领域、网络应用领域、消费类电子产品、成像和安全产品。除此之外,还应用到许多不同的领域,并会在将来取得更加广泛的应用。

采用 RISC 架构的 ARM 微处理器一般具有如下特点:

- 体积小、低功耗、低成本、高性能;
- 支持 Thumb(16 位)/ARM(32 位)双指令集,能很好的兼容 8 位/16 位器件;
- 大量使用寄存器,指令执行速度更快;
- 大多数数据操作都在寄存器中完成;
- 寻址方式灵活、简单,执行效率高;

- 指令长度固定。

ARM 处理器当前有 5 个产品系列：ARM7、ARM9、ARM9E、ARM10 和 SecurCore。进一步的产品来自于合作伙伴，例如 Intel Xscale 微处理器结构和 StrongARM 产品。ARM7、ARM9、ARM9E 和 ARM10 为 4 个通用处理器系列，每个系列提供一套相对独特的性能来满足不同应用领域的需求。SecurCore 系列专门为安全要求较高的应用而设计。

ARM7 系列微处理器包括如下几种类型的核：ARM7TDMI、ARM7TDMI-S、ARM720T、ARM7EJ。其中 ARM7TDMI 是目前使用最广泛的 32 位嵌入式 RISC 处理器，属低端 ARM 处理器核。

3.1.2 片内外设及存储空间

除 ARM 微处理器核以外，几乎所有的 ARM 芯片均根据各自不同的应用领域，扩展了相关功能模块，并集成在芯片中，如 USB 接口、IIS 接口、LCD 控制器、键盘接口、RTC、ADC 和 DAC、DSP 协处理器等。设计者应分析系统的需求，尽可能采用片内外围电路完成所需的功能。这样既可简化系统的设计，同时也提高了系统的可靠性。

另外，大多数 ARM 微处理器片内存储器的容量都不太大，需要用户在设计系统时外扩存储器，但也有部分芯片具有相对较大的片内存储空间。如 ATMEL 的 AT91F40162 就具有高达 2MB 的片内程序存储空间，用户在设计时可考虑选用这种类型，以简化系统的设计。

3.1.3 基于 ARM 核的 S3C44B0X 嵌入式处理器

S3C44B0X 微处理器是一款由 Samsung 公司为手持设备设计的低功耗、高集成度的基于 ARM7TDMI 的微处理器^[33]。它采用了 ARM7TDMI 内核，0.25um 工艺的 CMOS 标准宏单元和存储编译器。它的低功耗和出色的全静态设计特别适用于对成本和功耗敏感的应用。同时，S3C44B0X 还采用了一种新的总线结构，即 SAMBA II。

S3C44B0X 的杰出特性是它的 CPU 核，是由 ARM 公司设计的 16/32 位 ARM7TDMI RISC 处理器(66MHz)。ARM7TDMI 体系结构的特点是它集成了 Thumb 代码压缩器，片上的 ICE 断点调试支持和一个 32 位的硬件乘法器。

3.1.4 片上硬件资源

为了降低成本，S3C44B0X 提供了丰富的内置部件，包括：外部存储器控制器；LCD 控制器（最大支持 256 色、LCD 具有专用 DMA）；2 通道通用 DMA、2 通道外设 DMA 并具有外部请求引脚；1 通道多主 IIC-BUS 控制器；1 通道 IIS-BUS 控制

器；5个PWM定时器和1通道内部定时器；看门狗定时器；71个多路I/O端口；8个外部中断端口；8通道10位ADC；具有日历功能的RTC；具有PLL的片上时钟发生器等。

S3C44B0X通过提供全面的、通用的片上外设，大大减少了系统电路中除处理器以外的元器件配置，从而最小化系统的成本。

3.2 系统硬件总体设计

3.2.1 基于S3C44B0X的嵌入式硬件平台

本文设计的电能质量在线监测系统具有采集、存储、传输、处理和输出功能。硬件设计主要以ARM微处理器S3C44B0X为核心，根据系统的功能具体可以分为四部分：GPS时钟部分，数据输入及采集部分，ARM中央处理单元部分和网络通信传输部分。GPS时钟部分主要是提供系统校时脉冲以及国际标准时间(UTC)，这是电力系统测量中的前提条件，在完成数据的同步及获得时间信息后，方便了对后续数据的分析处理。数据输入及采集部分包括对三相电压和三相电流模拟量的输入，根据分析要求实时的对各个监测点进行数据采集。ARM中央处理单元部分是整个系统的核心，协调系统整体工作，实现同步数据的采集，并完成对采集数据的分析处理计算同时完成对GPS时间信息的提取。网络通信传输部分主要通过串口通信和以太网通信，实现对监测数据的上传。系统的总体结构框图如图3-1所示。

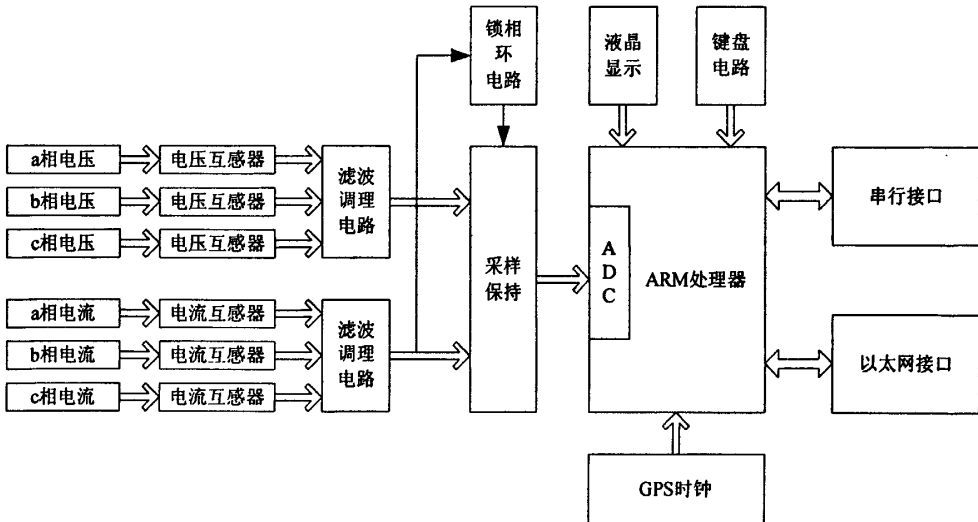


图 3-1 系统结构框图

3.2.2 系统原理

本文所设计的电能质量在线监测系统作为现场监测终端，主要安装在中、低压

配电端、大型电力用户输入端等现场，实现对电能质量的在线监测。

整个系统的工作原理是：三相电压、三相电流通过互感器采集进来，经滤波器滤波后，完成采样保持、模数转换，转换结果保存在存储器中，之后采样信号送入 ARM 处理器进行自动分类计算和数据处理、显示结果。系统对电压电流连续信号进行采样，得到的是有限长的离散时间序列，以此为基础构造周期性时间序列，使其在每个周期内都和有限长离散序列相同。取其中一个周期序列，利用快速傅立叶变换（FFT）进行谐波分析，可以求出各次谐波频率、得到各次谐波频率下的电压、电流幅值等参数，再逐项进行分析计算，可以得到有功功率、无功功率等。此外，系统通过获取 GPS 提供的高精度时钟信号，可以对电能质量参数进行同步采样，从而保证采样数据的同时性，并把实时的计算结果送到显示屏显示。通过 RS232 串口总线可以实现点对点的本地数据传输，利用以太网接口可以通过以太网实现远程数据的高速传输，并且可以通过网络实现对监测数据的分析管理。

3.3 GPS 时钟部分

GPS(全球定位系统)其主体是 24 颗由美国发射、距地球约 1.7 万 km、在 6 个轨道上运行的人造卫星。它可以实时和全天候的为全球范围内的任一物体提供三维位置(经度、纬度、高度)、三维速度和时间共七维高精度导航信息^[2]。其定位精度民用可到 20m 左右，定时精度可小于 0.5 μ s。GPS 的用户设备主要由接收机硬件和处理软件组成。用户通过用户设备接收 GPS 卫星信号，经信号处理而获得用户位置、速度等信息，最终实现利用 GPS 进行导航、定位、授时等目的。目前 GPS 主要用于定位和导航，授时的用途比较少，但它具有授时精度高，不受任何地理条件限制的优点，是一个非常理想的电力系统同步时钟。因此，对于电力系统网来讲，要实现电力系统的综合自动化，采用 GPS 授时是非常必要的。

由于世界上 GPS 应用市场的迅速扩大，GPS 接收设备的生产许可以已经向其他生产商开放。目前市场商常见的是美国和日本一些厂商生产的接收设备。虽然各个生产厂生产的 GPS 接收设备类型不尽相同，但他们都有着统一的组成部分和接口，并且所遵从的通信协议也是相同的。

近几年，微型高性能 GPS 接收机大大地扩大了它地应用领域。本系统所采用的 FURUNO 公司生产的 GN-80 型 GPS 接收机，利用最新发展的信号处理技术和高速搜索引擎技术，实现了性能上的较大改进，具有低功耗、高灵敏度以及更快的首次定位时间等特点^[4]。GN-80 的主要性能指标如下：

- 接收频率为 1575.42MHz;
- 16 通道并行接收;
- 通讯速度为 4800bps;

- 数据更新时间为 1s;
 - 秒脉冲精度为 $78\text{ns} \pm 20\text{ns}$ (1PPS 与 UTC 时间同步, 且两秒之间没有误差) 等。
- GN-80 的数据输出格式为: NMEA0183 的 ASC II 码语句。

GPS 时钟装置是整个电能质量在线监测系统的基础部分, 它为全系统提供高精度的时间基准。GPS 接收机输出信号通过解码后, 一个通道输出 1 个秒脉冲 (PPS) 信号, 其脉冲前沿与 UTC 的同步误差不超过 $1\mu\text{s}$; 另一个通道为串口输出, 在每个秒脉冲发出后, 输出包含绝对时间的卫星定位信息, 并可由显示部件显示。GN-80 的管脚定义如表 3-1 所示。

表 3-1 GN-80 管脚定义表

管脚号	信号	I/O	定义
1	TEST	I	出厂检测用控制信号, L: 正常模式, H: 检测模式
2	MODE	I	内存写入用控制信号, L: 正常模式, H: 闪存写入
3	TD	O	异步串口输出数据
4	RD	I	异步串口输入数据
5	1PPS	O	UTC同步时钟脉冲 (每秒一个脉冲)
6	RST_N	I	模块rese信号, L: reset, H: 运行
7	VBCK	I	3.3V DC后备供电
8	GND	-	地线
9	VCC	I	3.3V DC电源供电
10	VANT	I	5V DC天线前置放大器供电

GPS 接收机与 ARM 的硬件接口如图 3-2 所示。GN-80 可以方便的通过串口输入控制语句, 用来控制 GPS 接收机 GN-80 的输出。

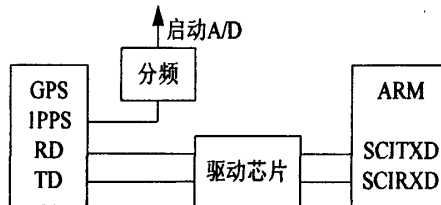


图 3-2 GPS 硬件接口电路

GN-80 的输入语句有:

- \$XXGLL: 设置经度纬度的初始值;
- \$XXGGA: 设置经度纬度的初始值;
- \$XXZDA: 设置日期时间的初始值;
- \$XXRMC: 设置初始位;
- \$PFEC, GPclr: 重新启动;
- \$PFEC, GPset: 设置输出参数;
- \$PFEC, GPSrq: 获取输出参数;

\$PFEC, GPint: 设置各条输出语句的时间间隔;

\$PFEC, GPirq: 获取各条语句输出时间间隔;

\$PFEC, GPdif: 设置 DGPS 参数;

\$PFEC, GPdrq: 获取 DGPS 参数。

GN-80 的输出语句包括搜索到的卫星的信息、定位后的经度以及纬度、日期时间等信息。共包括以下语句:

\$GPGGA: 这条语句输出的内容包括时间信息、定位的经度纬度信息等;

\$GPZDA: 这条语句包括时间信息和日期信息;

\$GPGLL: 该条语句内容为位置信息和时间信息;

\$GPGSA: 位置状态信息;

\$GPGSV: 搜索到的卫星的各项信息;

以及还有 \$GPDTM、\$GPVTG、\$GPRMC 和 \$PFEC、\$GPanc 等各条语句。除了这些信息以外, GN-80 的输出还包括一个同步秒脉冲, 其上升沿与 UTC 标准时间同步。

ARM 微处理器通过串口 RS-232 接收到数据, 从中提取时间信息。GPS 接收机 RD 引脚接收初始化信息, 包括初始位置、时间、秒脉冲、状态差分模式、NMEA 输出间隔等设置信息。1PPS 引脚输出秒脉冲信号, 经过分频启动 A/D 采样。

本系统利用 GPS 建立基于时间标签的数据同步采样, 其主要思路为: 利用 GPS 接收板发出的串口时间信息和同步秒脉冲, 在测量系统内建立整个测量系统的实时时钟, 由串口时间信息建立测量系统的年、月、日和时、分、秒, 由同步秒脉冲和计数器建立测量系统的微秒级时钟, 使得测量系统可以在任意时刻读出精度达到 1 微秒的时间, 这样在采样过程中, 不但可以获得采样点的值, 同时可以由 CPU 对该采样点打上时间标签。通过 GPS 的同步授时功能, 实现了数据采集的同步性。

3.4 数据输入及采集部分

3.4.1 结构设计原理

数据输入及采集部分硬件设计要满足经济合理、安全可靠、有足够的抗干扰能力的原则^[19]。本系统所要采集的数据是配电网中的三相电压、电流信号, 在信号输入端为电压和电流的模拟信号。首先, 待测线路上的三相交流电压和交流电流原始信号需经过变换。之后对变换后的模拟信号进行采样保持、模数转换并把转换结果送到 ARM 处理器中进行分析处理。

由于电力系统中各种干扰持续时间长, 随机性将, 为了精确测量网络中的高次谐波值, 必须保持较高的采样频率。本系统采用“采样保持器+多路开关+AD 转换器”的模式, 结构框图如图 3-3 所示, 每个采样保持器采集一路交流量输入, 这

样控制灵活方便。可以根据实际待测路数来确定采样保持的个数，易于实现装置的模块化。

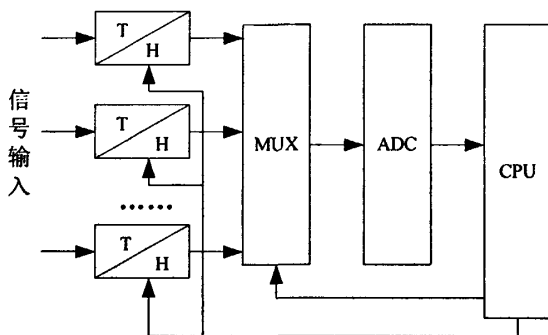


图 3-3 同步采集单元框图

3.4.2 信号提取电路

(1) 互感器的选择

三相电压、三相电流原始信号采集前需经过转换，电压信号通过电压互感器转换，电流信号通过电流互感器转换，电压互感器和电流互感器均采用电流型互感器。其原理图如图 3-4 所示。

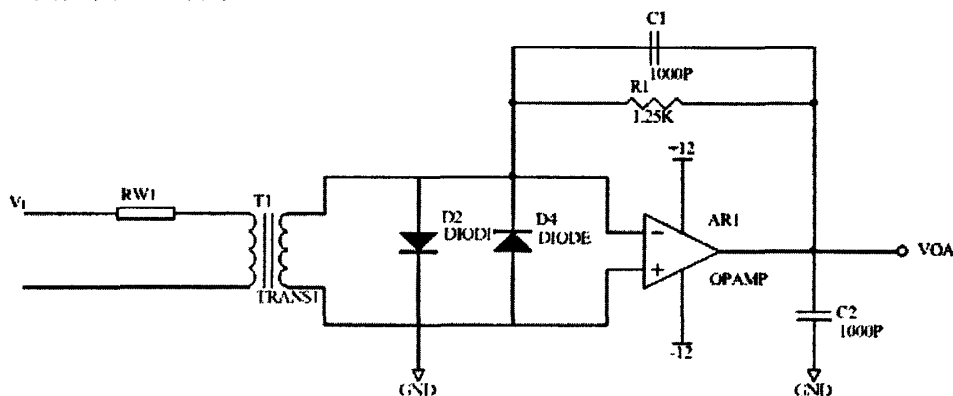


图 3-4 电压互感器电路

三路电压互感器均采用电流型电压互感器。实际上就是额定电流比为 1，一次二次电流都是毫安级的电流互感器（如 2mA/2mA）。图中 RW1 为限流电阻，调节其值可使输入电流为额定值，副边是电流/电压转换电路，通过选择不同的电阻值可以得到不同的电压输出范围。为了得到 $\pm 2.5V$ 的输出范围，由 $2.5V/2mA$ 可得电阻值为 $1.25k\Omega$ 。图中电容用来去耦和滤波。两个反并联的二极管起限幅作用，以保护后级电路。电流互感器和电压互感器相似，只是原边没有限流电阻。

(2) 抗混叠滤波器

为了防止采样信号的频谱混叠，一般都要在对模拟信号的采样和数字化之前进

行抗混叠滤波，电压和电流互感器采集进来的信号必须经过一个抗混叠滤波器，滤掉 31 次以上的谐波，抗混叠滤波电路的实质就是迫使信号通过一个有限带宽的低通滤波器，该滤波器的截止频率为奈奎斯特频率（采样频率的一半）。本系统的抗混叠滤波器采用六阶巴特沃斯低通滤波器，它是由三个二阶低通滤波器串联组成，其电路图如图 3-5 所示。

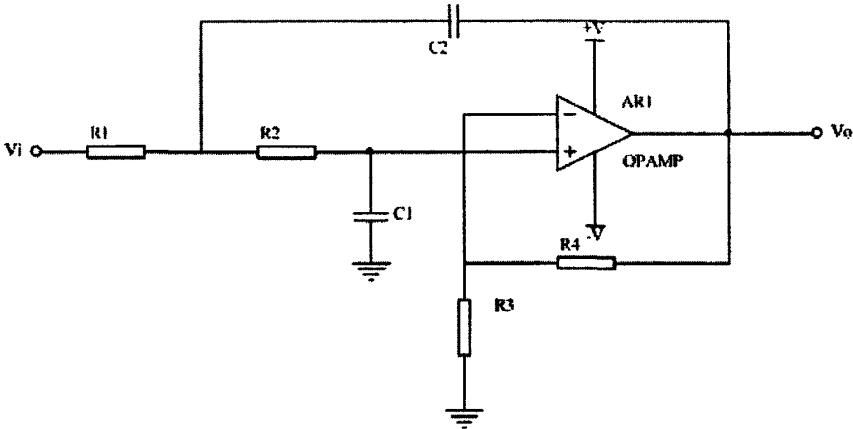


图 3-5 二阶低通滤波器电路图

其传递函数 $\frac{V_o}{V_i} = \frac{KC\omega_c^2}{S^2 + B\omega_c^2 + C\omega_c^2}$ ，其中 $C\omega_c^2 = \frac{1}{R_1 R_2 C_1 C_2}$ ，

$B\omega_c = \frac{1}{C_2} \left(\frac{1}{R_1} + \frac{1}{R_2} \right) + \frac{1}{R_2 C_1} (1-K)$ ， $K = 1 + \frac{R_4}{R_3}$ ， ω_c 为截止频率，K 为增益，B、C 为巴特沃斯滤波系数。

由前面可知需滤掉 31 次以上的谐波，但实际上只需测量 21 次以内的谐波，所以只需滤掉 21 次以上的谐波即可，则 ω_c 可取 1050Hz。由于二阶环节串联会降低截止频率，所以实际应用中 ω_c 应取得稍微大一些，在这里 ω_c 取为 1200Hz。 C_2 近似于

$10/f_c \mu F$ ，即可取为 $0.01 \mu F$ ，而 $C_1 \leq \frac{[B^2 + 4C(K-1)]C_2}{4C}$ 。整个系统的传递函数增益

为 1，巴特沃斯低通滤波器系数 B、C 由文献可查得，由此可得到表 3-2。

表 3-2 巴特沃斯低通滤波器系数表

二阶环节	B	C	$C_2(\mu F)$	$C_1(pF)$	$R_1(k)$	$R_2(k)$	R_3	R_4
1	0.517638	1.0	0.01	300	33	110	∞	0
2	1.414214	1.0	0.01	4700	15	25	∞	0
3	1.931852	1.0	0.01	6800	9	30	∞	0

3.4.3 A/D 转换及采样保持电路

本系统所选用的 ARM 处理器为 S3C44B0X,它自带 8 路复用 10 位 A/D 转换器,用它构成数据采集系统不但能够很好的满足系统的要求,而且能够降低系统的复杂性。它的内部结构中包括模拟输入多路复用器 (AMUX)、自动调零比较器 (COMP)、时钟产生器 (PSR)、10 位逐次逼近寄存器 (SAR) 和输出寄存器 (ADCDAT),如图 3-6 所示。这个 A/D 还提供了可编程选择的休眠模式,以节省功耗。它的主要特性为:精度为 10 位,微分线性误差 $\pm 1\text{LSB}$,积分线性误差 $\pm 2\text{LSB}$ (最大不超过 $\pm 3\text{LSB}$),最大转换速率为 100kb/s,输入电压范围 0~2.5V,输入带宽 0~100Hz,低功耗工作。

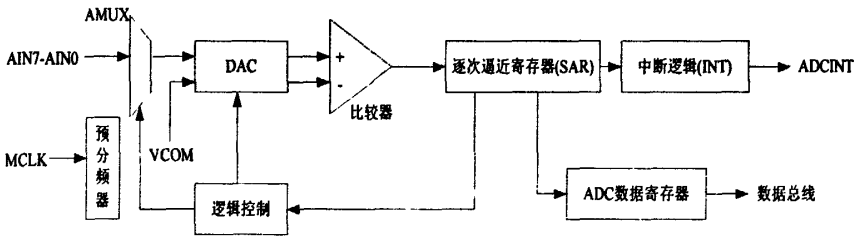


图 3-6 S3C44B0X 处理器的 ADC 结构框图

ADC 的外部引脚除了 ADCINT 引脚、10 位 ADC 数据总线引脚、8 路模拟输入引脚 (AIN7~AIN0) 及模拟公共电压引脚外,还有正向参考电压 (AREFT) 引脚和反向参考电压 (AREFB) 引脚。为了增强电压的稳定性,在 VCOM 引脚、AREFT 引脚及 AREFB 引脚对地之间必须接旁路电容(一个 $10\mu\text{F}$ 和一个 $0.1\mu\text{F}$ 的电容并联)。

S3C44B0X 自带 A/D 的电压范围是 0~2.5V,没有负电压,所以要将初次转换的信号通过平移电路进行平移,在本设计中采用基本放大电路中的加法电路,如图 3-7 所示,使 A/D 采集的信号符合要求。

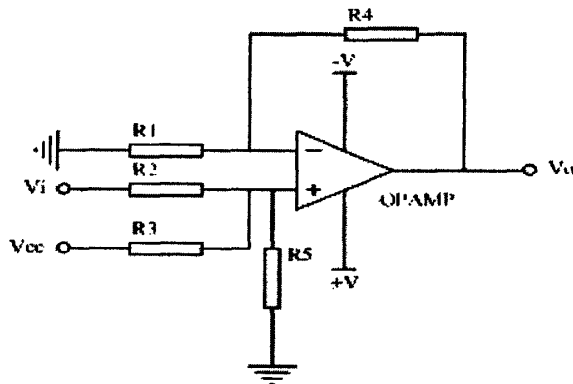


图 3-7 电压平移电路

A/D 转换时间即完成一次 A/D 转换所需要的时间,当系统的时钟频率为 64MHz,且 ADC 时钟源的预分频值为 20 时,10 位数字量的转换时间为:

$(64 \text{ MHz}/2 \times (20 + 1)) / 16 = 95.2 \text{ kHz}$ ，即 $10.5 \mu\text{s}$ 。式中，16 表示 10 位操作至少要 16 个周期。S3C44B0X 的 ADC 的输出为 10 位数字量，由于输入的满刻度电压为 2.5V，因此，ADC 能分辨出来的输入电压变化的最小值为 $2.5\text{V}/2^{10} = 2.4\text{mV}$ 。

在数据采集集中，采样保持器用来对输入 A/D 转换器的模拟信号进行采集和保持，以确保 A/D 转换的精度。要保证 A/D 转换的精度，就必须确保 A/D 转换过程中输入模拟信号的变化量不得大于 $\text{LSB}/2$ 。如果模拟信号不经过采样保持器而直接输入 A/D 转换器，那么系统允许该模拟信号的变化率就得减低。在 S3C44B0X 中 A/D 模块有 8 个模拟输入通道，通道的切换可以由内部的定时器完成。如果要进行 8 个通道连续变化的信号的转换，则必须在 8 个通道全部加采样保持器。采样保持的接口电路如图 3-8 所示，模拟输入信号为需要转换的信号；驱动控制信号可以通过编程利用 ARM 中的 timer 产生，也可以通过 I/O 口来控制；输出信号直接接到 A/D 模块中的输入通道。

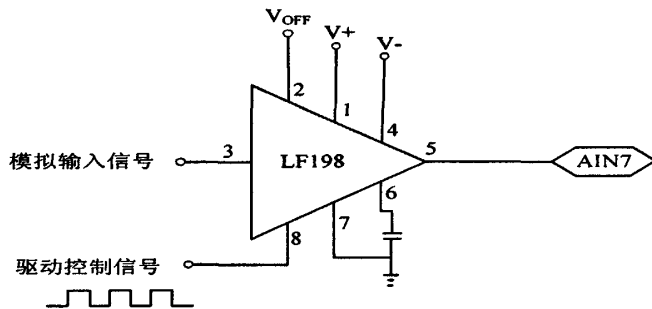


图 3-8 采样保持接口电路

3.4.4 锁相环电路^[14]

为了使采样信号同步，本系统在测量工频交流电压信号和电流信号时，利用锁相环对信号进行 128 倍频得到的脉冲去控制 ARM 芯片 S3C44B0X 对电压、电流信号进行采样，既可实现相位测量，又能保证在被测信号的频率不稳定时，始终以 128 点采集一个周期的完整信号，从而保证后面数字计算时的精度要求。结构框图如图 3-9 所示。

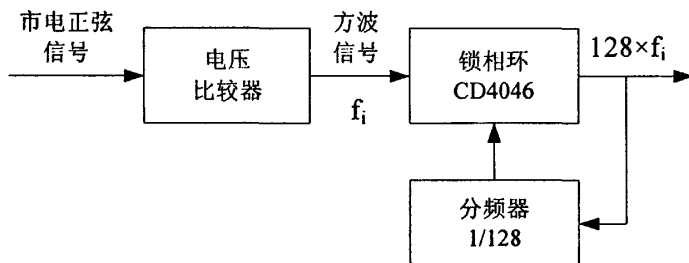


图 3-9 同步采集结构框图

在测量工频交流电压、电流信号时，必须把输入的信号变成方波信号，这就需要方波转换电路，如图 3-10 所示。

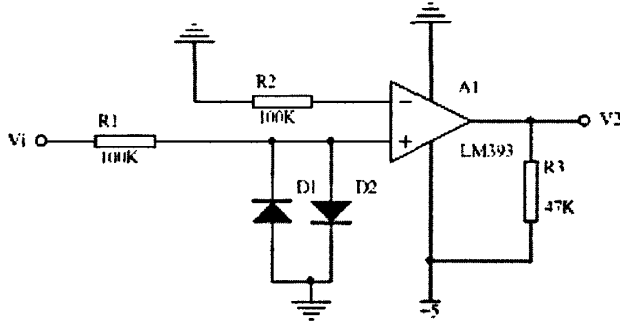


图 3-10 方波转换电路

因为要保证每个工频周波中对信号进行 128 次采样，而且要保证所采样的点数都在同一周期内，即要保证采样的第一个点是一个周期的开始，而最后一个点是周期的结束，所以设计了锁相环电路。本系统中采用 CD4046 作为锁相环，74LS393 作为分频器构成采样信号触发电路，CD4046 是通用的 CMOS 锁相环集成电路，其特点是电源电压范围宽 (3~18V)，输入阻抗高 (约 100MΩ)，动态功耗小，在中心频率 f_0 为 10kHz 下功耗仅为 600μW，属低功耗器件，电路如图 3-11 所示。

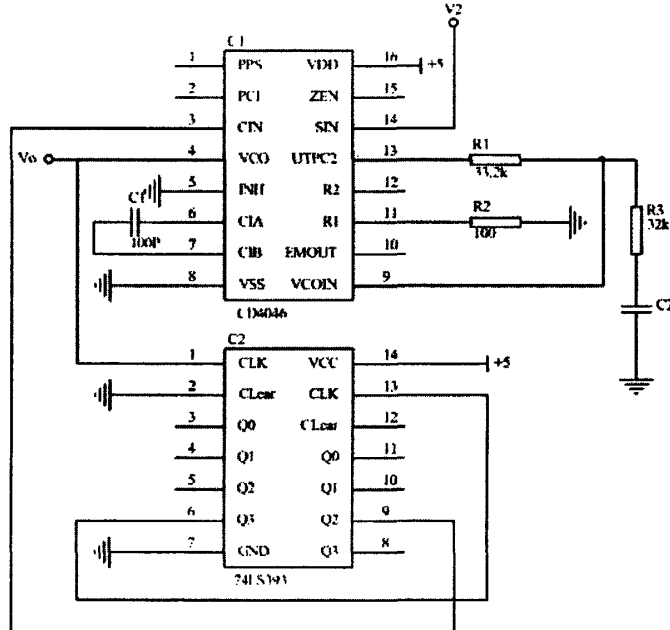


图 3-11 锁相环电路

CD4046 采用的是 RC 型压控振荡器 (VCO)，必须外接电容 C_1 和电阻 R_1 作为充放电元件。当锁相环对跟踪的输入信号的频率宽度有要求时还需要外接电阻 R_2 ，由于 CD4046 的 VCO 是一个电流控制振荡器，对定时电容 C_1 的充电电流与从 9 脚

输入的控制电压成正比，使 VCO 的振荡频率亦正比于该控制电压。当 VCO 控制电压为 0 时，其输出频率最低；当输入控制电压等于电源电压 V_{DD} 时，输出频率则线性的增大到最高输出频率。VCO 振荡频率的范围由 R_1 、 R_2 和 C_1 决定。由于它的充电和放电都由同一个电容 C_1 完成，故它的输出波形是对称方波，一般规定 CD4046 的最高频率 f_{max} 为 1.2MHz ($V_{DD}=15V$)，若 $V_{DD}<15V$ ，则 f_{max} 要降低一些。

由图 3-11 可知，经图 3-10 电路调理的方波信号 V_2 作为频率跟踪的输入信号，通过内部相位比较器于加法计数器 74LS393 的分频器作相位比较，加控制电压于内部压控振荡器 VCO 的输入端，使得压控振荡器的输出频率不断调整，最后使得输出信号 V_0 频率为输入信号 V_2 频率的 128 倍，这样获得的输出信号就可以用来实现每周期 128 点的采样控制。

本系统利用锁相环电路把信号的频率进行了 128 倍频，从而在需采集信号的一个周期中产生了 128 个脉冲，利用此脉冲信号作为 ARM 的外部中断信号，快速启动 A/D 进行转换，实现高速数据采集。这种方法实现简单，可靠性强，且简化了软件的设计。

3.5 ARM 中央处理单元

通过数据采集部分采集进来的电能质量参数经过 A/D 数字化后，电路将触发中断信号给 ARM 处理器，读入实时数据到 RAM 中，开始执行 FFT 单元，计算电网中的各种电能参数，分析电网的电能质量信息。

3.5.1 电源电路

在 S3C44B0X 上由于其内核采用 2.5V，I/O 接口采用 3.3V 供电，因此需要将通用的 5V 转换成 2.5V 和 3.3V。图 3-12 为使用 LM1117 电源转换芯片把 5V 转换成 3.3V 和 2.5V 的转换电路。

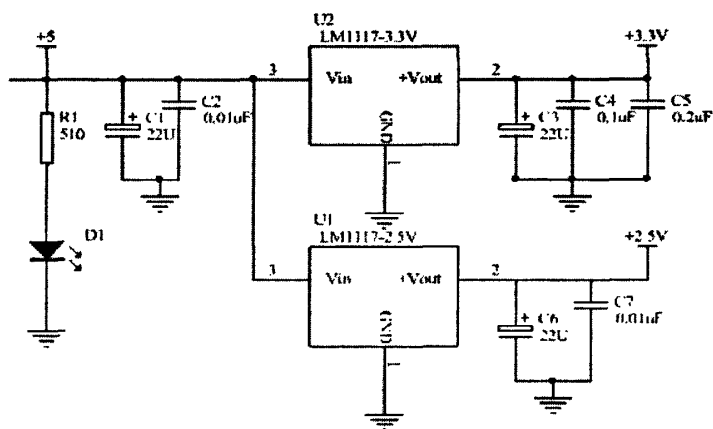


图 3-12 电源转换电路

3.5.2 LCD 显示部分

液晶显示模块在各类测量及控制仪表中由于其功耗低、寿命长、价格低、接口控制方便等优点而被广泛的应用。本系统采用 HY12864 液晶显示模块, 该显示模块为 128×64 点阵 LCD。

一块 LCD 屏显示图象, 不但需要 LCD 驱动器, 还需要有相应的 LCD 控制器。而本系统所采用的 ARM 处理器 S3C44B0X 内部已经集成了 LCD 控制器, 因此可以很方便的去控制各种类型的 LCD。LCD 控制器外部接口信号的定义及其与 LCD 模块各信号之间的对应关系如下:

VFRAME: LCD 控制器和 LCD 驱动器之间的帧同步信号, 该信号与 LCD 模块的 YD 信号相对应。

VLINE: LCD 控制器和 LCD 驱动器之间的线同步脉冲信号, 该信号与 LCD 模块的 LP 信号相对应。

VCLK: LCD 控制器和 LCD 驱动器之间的像素时钟信号, 该信号与 LCD 模块的 XCK 信号相对应。

VM: LCD 驱动器的 AC 信号。

VD7~0: LCD 像素点数据输出端口, 与 LCD 模块的 D7~0 相对应。

另外可能还需要: 采用一个 GPIO 与 LCD 模块的 DISP 信号连接(一般情况下, 高电平为开显示, 低电平为关显示); 采用一个 GPIO 作为 LCD 模块的背光开关。

3.5.3 键盘接口部分

键盘是用来设定控制参数的最基本的手段。为了用户使用方便, 本系统硬件构成成为 4×4 的行列式键盘, 这比传统的 4 个按键的键盘输入方便。用户通过键盘可用来设定控制参数, 如电压、电流上限, 补偿系数等参数。键盘的连接电路如图 3-13 所示, 行线通过一个电阻被上拉到 VCC, VCC 是一个 +5V 电压。行线与按键的一个引脚相连, 列线与按键的另一个引脚相连。平时列线被置成低电平, 没有按键被按下时, 行线保持高电平; 而有按键被按下时, 行线被拉成低电平, 这时候控制器知道有按键被按下, 但只能判断出在哪一行, 不能判断出在哪一列, 因此接下来就要进行键盘扫描, 以确定具体是哪个按键被按下。

键盘扫描的过程是将列线逐列置成低电平, 然后读取行线状态, 直到行线中出现低电平, 这时可知哪一列是低电平; 然后将行线与列线的状态装入键码寄存器, 进行按键译码, 得到按下的按键的相应编码, 这样就完成了按键扫描过程。当然一个完整的按键扫描过程还需要配合相应的键盘去抖动手段才能正确地识别按键, 不会发生重键和错误判断等情况。

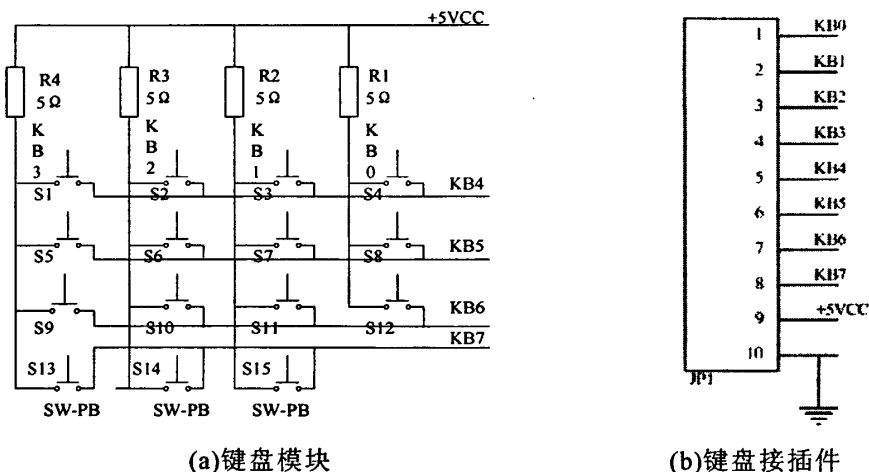


图 3-13 键盘电路原理图

3.5.4 外扩存储器电路

本系统外扩一片 512Kbit 的 EEPROM，用来存储固定数据参数。EEPROM 采用 ATMEL 公司的 AT24C512，它是 ATMEL 公司推出的具有 I2C 总线容量达 512Kbit (64×8) 的 E²PROM，该芯片的主要特性为：存储容量为 6553byte；与 100kHz、400kHz、1Mhz I2C 总线兼容；100000 次编程/擦写周期；单电源、读写电压为 1.8V~5.5V；ESD 保护电压>4kV；数据可保存 40 年；具有写保护功能，当 WP 为高电平时，进入写保护状态；CMOS 低功耗技术，最大写入电流为 3mA；128 字节页写入缓存器；自动定时的写周期；具有 8 引脚 PDIP 及 20 引脚 SOIC 封装等多种封装形式。本系统采用 8 引脚 PDIP 封装的 AT24C512，3.3V 供电，通过串口与 ARM 的串口实现互连，电路图如图 3-14 所示。

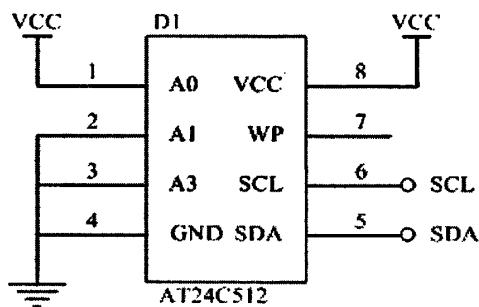


图 3-14 存储器电路图

3.6 网络通信传输部分

本系统的网络通信传输主要实现 RS232 和以太网通信两部分，采用 RS232 串口总线实现点对点的本地数据传输，利用以太网口可以通过以太网实现远程数据的

高速传输。

3.6.1 串行接口设计

S3C44B0X 只有 UART0/1 并没有集成 RS-232C 接口,因此若要与其他设备进行串行通信,通常需要采用 RS-232C 标准接口,而 S3C44B0X 的 I/O 接口电源为 3.3V,因此不能选用 5V 供电的 RS-232 接口芯片,选用 MAX3232 作为 UART 到 RS-232 电平转换芯片。利用 S3C44B0X 的 UART0 与接口电平转换芯片 MAX232 就可以实现双机或多机通信,地线相连即可。其串行接口电路如图 3-15 所示。

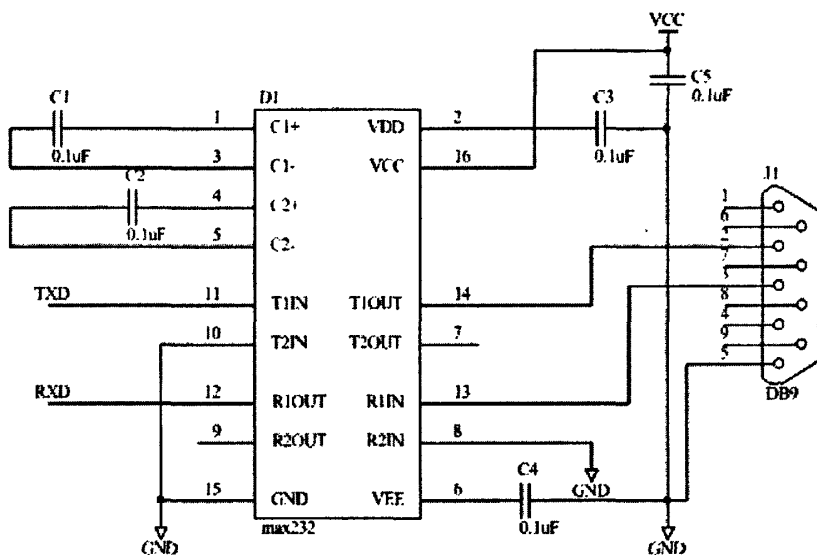


图 3-15 串口电路图

ARM 的 UART 单元提供独立的异步串行 I/O 端口,每个都可以在中断和 DMA 两种模式下工作。支持的最高波特率为 115.2kbps,每个 UART 通道包含 2 个 16 位 FIFO,分别用于接收和发送数据。TTL/CMOS 输入输出信号与 ARM 的 UART 输入/输出口对接。

3.6.2 以太网接口设计

本系统选用一款专门为第三代快速以太网连接而设计的 RTL8019AS 10M/100Mbps 兼容以太网接口芯片,将数据依据 TCP/IP 协议进行网络传输。它支持多种嵌入式处理器芯片,内置 FIFO 缓存器用于发送和接收数据。系统通过 RJ-45 连上以太网,连接电路如图 3-16 所示,网卡芯片单独不能工作,还必须有一个网络变压器在 RJ-45 接口和网卡芯片中间进行电平变换。

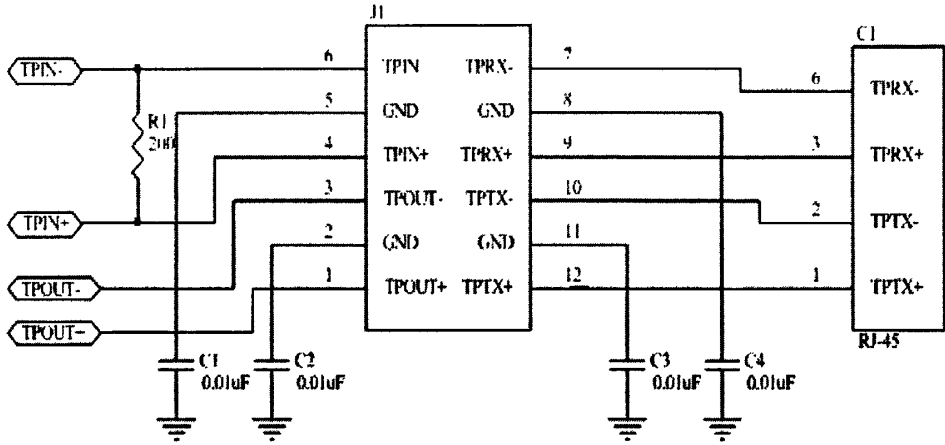


图 3-16 以太网连接电路

3.7 小节

在本章中，首先对基于 ARM 的嵌入式系统做了详细的介绍，对本系统所选用的 ARM 微处理器 S3C44B0X 的性能有了一定的了解。之后，主要介绍了以 S3C44B0X 为核心的电能质量监测系统的硬件结构，根据实现的功能共分为四部分，分别为：GPS 时钟部分，数据输入及采集部分，ARM 中央处理单元部分和网络通信传输部分。并详细介绍了每一部分实现的功能以及各部分主要电路的设计，搭建起了系统的硬件平台。

第四章 ARM 系统软件平台的建立

4.1 嵌入式实时操作系统

实时多任务操作系统（Real Time multi-tasking Operating System, RTOS）是嵌入式应用软件的基础和开发平台。RTOS 将应用分解成多个任务，简化了应用系统软件的设计。良好的多任务设计，有助于提高系统的稳定性和可靠性。基于 RTOS 开发出的程序，绝大部分代码与硬件平台无关，具有较高的可移植性。它使得应用程序的设计和扩展变得容易，无需大的改动就可以增加新的功能。通过将应用程序分割成若干独立的任务，RTOS 使得应用程序的设计过程大为简化。使用可剥夺型内核时，所有时间要求苛刻的事件都得到了尽可能快捷、有效的处理。通过有效的服务，如信号量、邮箱、队列、延时及超时等，RTOS 使资源得到更好的利用。嵌入式软件的函数化、产品化能够提高软件的可重用性，减少软件的重复开发，提高应用算法的研发效率。

在硬件平台的基础上，在多任务嵌入式系统中，对重要性各不相同的任务进行统筹兼顾的合理调度是保证每个任务及时执行的关键，单纯通过提高处理器速度是无法完成和没有效率的，必须有操作系统对各种任务进行管理和调度，嵌入式实时多任务操作系统（RTOS）特别适合这种单处理器多任务的系统。

嵌入式实时操作系统的种类繁多，大体上可分为两种：商用型和源码开放型。商用型的主要有美国 WindRiver 公司的 VxWork、pSOS；Microwave 公司的 OS-9；Lynx Real-time System 的 LynxOS 和加拿大 QNX 公司的 QNX 等。源码开放型的主要有嵌入式 Linux（软实时）和 $\mu\text{C}/\text{OS}$ ， $\mu\text{C}/\text{OS-II}$ 是 $\mu\text{C}/\text{OS}$ 的升级版本，它支持 x86、ARM、PowerPC、MIPS 等众多体系结构。本文采用的是免费的源代码开放的 $\mu\text{C}/\text{OS-II}$ 。

4.1.1 $\mu\text{C}/\text{OS-II}$ 结构与硬件平台关系

$\mu\text{C}/\text{OS-II}$ 是由 Labrosse 先生编写的嵌入式实时操作系统内核（RTOS）， $\mu\text{C}/\text{OS-II}$ 的内核文件结构与硬件平台的关系如图 4-1 所示。可以看出 $\mu\text{C}/\text{OS-II}$ 内核介于硬件平台和用户应用程序之间，通过与 CPU 相关的几个模块与硬件平台进行交互，为用户应用程序提供良好的应用程序接口（API）。

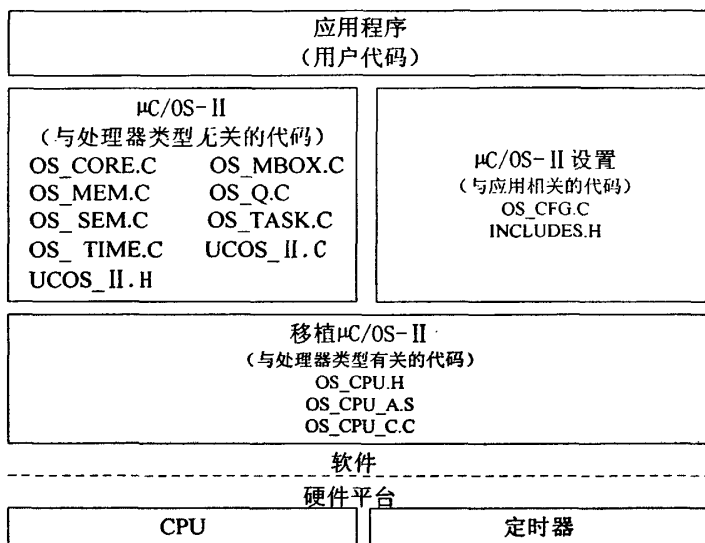


图 4-1 μC/OS-II 结构与硬件平台关系

操作系统把系统软件和硬件部分隔离开来，这样就使系统的设备操作程序与硬件设备无关，从而大大提高了系统的可移植性。从软硬件测试角度来看，软硬件的测试工作都可分别基于操作系统来完成，使得软硬件的测试工作的并行进行成为可能。

4.1.2 μC/OS-II 的主要特点^[13]

μC/OS-II 的主要特点如下：

- 源代码开放

源代码全部公开，这样使系统变得完全透明，很容易就能把操作系统移植到各个不同的硬件平台上。

- 可移植性

μC/OS-II 绝大部分源代码是用 ANSI C 写的，可移植性较强。而与微处理器硬件相关的部分是用汇编语言写的，已经压到最低限度，使其便于移植。

- 可固化

μC/OS-II 是为嵌入式应用而设计的，这就意味着，只要开发者有固化手段（C 编译、连接、下载和固化），μC/OS-II 就可以嵌入到开发者的产品中成为产品的一部分。

- 可裁减性

用户可以在应用程序中选择需要的 μC/OS-II 的功能，从而减少所需的存储空间。其可裁剪性是通过条件编译实现的。

- 占先式

μC/OS-II 完全是占先式的实时内核，它总是运行就绪条件下优先级最高的任

务。

- 多任务

$\mu\text{C}/\text{OS}-\text{II}$ 可以管理 64 个任务，其中 8 个留给系统，应用程序最多可以有 56 个任务，赋予每个任务的优先级必须是不同的。

- 可确定性

全部 $\mu\text{C}/\text{OS}-\text{II}$ 的函数调用与服务的执行时间具有可确定性，是可知的，系统服务的执行时间不依赖于应用程序任务的多少。

- 任务栈

每个任务有自己单独的栈， $\mu\text{C}/\text{OS}-\text{II}$ 允许每个任务有不同的栈空间，以便压低应用程序对 RAM 的需求。

- 系统服务

$\mu\text{C}/\text{OS}-\text{II}$ 提供很多系统服务，例如邮箱、消息队列、信号量、块大小固定的内存的申请和释放、时间相关函数等。

- 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该中断唤醒，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。

- 稳定性和可靠性

$\mu\text{C}/\text{OS}-\text{II}$ 是基于 $\mu\text{C}/\text{OS}$ 的，已经有许多成功的商业应用，具有很好的稳定性和可靠性。

4.1.3 嵌入式操作系统 $\mu\text{C}/\text{OS}-\text{II}$ 在 S3C44B0X 处理器上的移植

所谓移植，就是使一个实时内核能在其他的微处理器和微控制器上运行。因为大部分 $\mu\text{C}/\text{OS}-\text{II}$ 的代码是用 C 语言编写的，因此 $\mu\text{C}/\text{OS}-\text{II}$ 的可移植性较强。然而，仍需要用 C 语言和汇编语言编写一些与处理器硬件相关的代码。

要移植 $\mu\text{C}/\text{OS}-\text{II}$ ，处理器必须满足以下要求：

- 处理器的 C 编译器能产生可重入代码。
- 用 C 语言就可以打开和关闭中断。
- 处理器支持中断，并且能产生定时中断（通常在 10~100Hz 之间）。
- 处理器支持能够容纳一定量数据（可能是几千字节）的硬件堆栈。
- 处理器有将堆栈指针和其他 CPU 寄存器读出和存储到堆栈或内存中的指令。

Sam sung 公司的 ARM7TDMI 的嵌入式处理器 S3C44B0X 能够满足以上的要求。ARM 公司提供的 SDT2.5 开发软件中包含的 armcc 编译器能够产生可重入性代码。接下来介绍如何把 $\mu\text{C}/\text{OS}-\text{II}$ 移植到处理器 S3C44B0X 上。

对 $\mu\text{C}/\text{OS}-\text{II}$ 的移植实际上就是对与处理器有关的代码进行重写或修改，移植工

作包括以下几个内容。

- 用 #define 设置一个常量的值 (OS_CPU.H 文件中)。
- 声明 10 个数据类型 (OS_CPU.H 文件中)。
- 用 #define 声明 3 个宏 (OS_CPU.H 文件中)。
- 用 C 语言编写 6 个简单的函数 (OS_CPU_C.C 文件中)。
- 编写 4 个汇编语言函数 (OS_CPU_A.ASM 文件中)。

4.1.3.1 设置 includes.h 中与处理器及编译器相关的代码

(1) 与编译器有关的数据类型

因为不同的微处理器有不同的字长, 所以 $\mu\text{C}/\text{OS-II}$ 的移植包括一系列的类型定义以确保其可移植性。尤其是 $\mu\text{C}/\text{OS-II}$ 代码从不使用 C 的 short、int 和 long 等数据类型, 因为它们是与编译器相关的, 不可移植的。相反的, 我们定义的整形数据类型既是可移植的又是直观的。

为了方便, 虽然 $\mu\text{C}/\text{OS-II}$ 不使用浮点数据, 但我们还是定义了浮点数据类型。例如, INT16U 数据类型总是代表 16 位的无符号整数。现在 $\mu\text{C}/\text{OS-II}$ 和用户的应用程序就可以估计出声明为该数据的变量的取值范围是 0~65535。将 $\mu\text{C}/\text{OS-II}$ 移植到 32 位的处理器上也就意味着 INT16U 实际被声明为无符号短整型数据结构而不是无符号整数数据结构。但是 $\mu\text{C}/\text{OS-II}$ 处理的仍然是 INT16U。

用户必须将任务堆栈的数据类型告诉给 $\mu\text{C}/\text{OS-II}$ 。这个过程是通过为 OS_STK 声明正确的 C 数据类型来完成的。处理器上的堆栈成员是 16 位的, 所以将 OS_TSK 声明为无符号整形数据类型。所有的任务堆栈都必须用 OS_TSK 声明数据类型。

(2) OS_ENTER_CRITICAL() 和 OS_EXIT_CRITICAL()

与所有的实时内核一样, $\mu\text{C}/\text{OS-II}$ 需要先禁止中断再访问代码的临界区, 并且在访问完毕后重新允许中断。这就使得 $\mu\text{C}/\text{OS-II}$ 能够保护临界区代码免受多任务或中断服务子程序的侵扰。

通常每个处理器都会用一定的汇编指令来开关中断, C 编译器必须有一定的机制直接从 C 语言中执行这些操作。有些编译器允许在 C 源代码中插入行汇编语句, 这就使得插入处理器指令来开关中断变得比较容易, 有些其他的编译器提供语言扩展功能, 可以直接从 C 语言中开/关中断。

在 S3C44B0X 上是通过两个函数(OS_CPU_A.S)实现开关中断的。开关中断函数实现的流程如下图 4-2a、图 4-2b 所示。

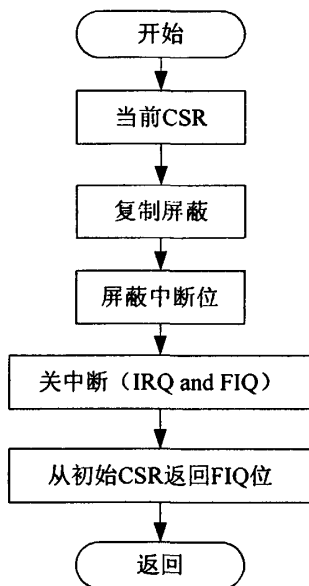


图 4-2a 关中断函数 INTS_OFF

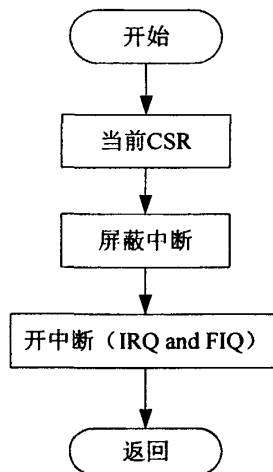


图 4-2b 开中断函数 INTS_ON

(3) OS_STK_GROWTH

绝大多数的微处理器和微控制器的堆栈是从上往下长的。但是某些处理器是用另外一种方式工作的。 $\mu\text{C}/\text{OS-II}$ 被设计成两种情况都可以处理，只要在结构常量 OS_STK_GROWTH 中指定堆栈的生长方式就可以了。

置 OS_STK_GROWTH 为 0 表示堆栈从下往上长。

置 OS_STK_GROWTH 为 1 表示堆栈从上往下长。

4.1.3.2 用 C 语言编写 6 个与操作系统相关的函数(OS_CPU_C.C)

$\mu\text{C}/\text{OS-II}$ 的移植要求用户编写以下 6 个简单的 C 函数

- OSTaskStkInit()
- OSTaskCreateHook()
- OSTaskDelHook()
- OSTaskSwHook()
- OSTaskStatHook()
- OSTaskTickHook()

(1) 任务堆栈初始化

OSTaskCreate() 和 OSTaskCreateExt() 通过调用 OSTaskStkInt() 来初始化任务的堆栈结构并返回新的堆栈指针 stk。在 ARM 体系结构下，任务堆栈空间由高至低依次将保存着 pc、lr、r12、r11、r10、...、r1、r0、CPSR 和 SPSR、图 4-3 说明了 OSTaskStkInt() 初始化后（也是新创建任务）的堆栈内容。堆栈初始化工作结束后，OSTaskStkInt() 返回新的堆栈栈顶指针，OSTaskCreate() 或 OSTaskCreateExt() 将指针保存在任务的

OS_TCB 中。

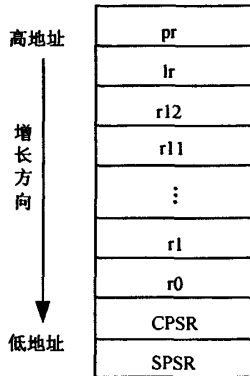


图 4-3 OSTaskStkInt() 初始化后的堆栈内容

(2) 系统 hook 函数

在这个文件里还需要实现几个操作系统规定的 hook 函数。这些函数为用户定义函数，它将在相应的操作系统调用后执行由用户定义的这些 hook 函数，执行特定的用户操作，如果没有特殊需求，则只需简单地将它们都实现为空函数即可。这些函数包括 OSTaskCreateHook()、OSTaskDelHook()、OSTaskSwHook()、OSTaskStatHook()、OSTaskTickHook()。

(3) 中断级任务切换函数

该函数由 OSIntExit 和 OSExIntExit 调用。它若在时钟中断 ISR (中断服务例程) 中发现有优先级任务等待的时钟信号到来，则需要在中断退出后并不返回被中断的任务，而是直接调度就绪的高优先级任务执行。这样做的目的主要是能够尽快地让高优先级的任务得到响应，以保证系统的实时性能。该函数通过设置一个全局变量 need_to_swap_context 标志来表示在中断服务程序中进行任务切换，然后在 OSTickISR() 中判断该变量以进行正确的动作，其函数如下：

```
Void OSIntCtxSw(void)
{
    need_to_swap_context=1
}
```

4.1.3.3 用汇编语言编写 4 个与处理器相关的函数(OS_CPU_A.ASM)

在 OS_CPU_A.ASM 文件中，μC/OS-II 的移植要求用户编写以下四个简单的汇编语言函数。

- OSStartHighRdy()
- OSCtxSw()
- OSIntCtxSw()
- OSTickISR()

OSStartHighRdy() 函数用来启动当前就绪的优先级最高的任务。OSCtxSw() 用来进行任务间的切换。OSIntExit 通过调用 OSIntCtxSw() 来从 ISR 中执行切换任务。另外, $\mu\text{C}/\text{OS-II}$ 要求用户提供一个时钟源来实现时钟节拍的功能。时钟节拍推荐为每秒钟发生 10~100 次。

经过上述这些移植工作, $\mu\text{C}/\text{OS-II}$ 就可以在 S3C44B0X 处理器系统上开始工作了。

4.2 实时操作系统下 TCP/IP 协议的实现

4.2.1 基于 $\mu\text{C}/\text{OS-II}$ 的网络平台简介

TCP/IP 是 Internet 的基本协议, 具有很强的实用性、高效性。嵌入式设备要与 Internet 直接交换信息, 就必须支持 TCP/IP 协议。

$\mu\text{C}/\text{OS-II}$ 是一个开放的 RTOS, 源代码全部公开。但是它目前的一些第三方 TCP/IP 支持都是完全商业化的, 很少给出源代码, 这就影响了 $\mu\text{C}/\text{OS-II}$ 的研究和推广。通过把开放源代码的 TCP/IP 协议栈 LWIP 移植到 $\mu\text{C}/\text{OS-II}$ 上来, 就得到了一个嵌入式网络软件平台。

4.2.2 TCP/IP 协议栈 LWIP

LWIP 是瑞士计算机科学院开发的一套用于嵌入式系统的开放源代码的 TCP/IP 协议栈。LWIP 的含义是 Light Weight (轻型) IP 协议。LWIP 可以移植到操作系统上, 也可以在没有操作系统的情况下独立运行。LWIP TCP/IP 实现的重点是在保持 TCP/IP 协议主要功能的基础上减少对 RAM 的占用。一般它只需要几十 k 的 RAM 和 40k 左右的 ROM 就可以运行, 这使 LWIP 协议栈适合在低端嵌入式系统中使用。

4.2.3 LWIP 进程模型

TCP/IP 协议栈的 process model 一般有几种方式。

(1) TCP/IP 协议每一层是一个单独进程 (链路层、IP 层、TCP 层), 这样的好处是网络协议的每一层都非常清晰, 代码的调试和理解都非常容易, 但最大的坏处是数据层跨层传递时会引起上下文切换。对于接收一个 TCP segment 要引起 3 次 context switch (从网卡驱动程序到链路层进程, 从链路层进程到 IP 进程, 从 IP 层进程到 TCP 进程)。通常对于操作系统来说, 任务切换是要浪费时间的, 过频的 context switch 是不可取的。

(2) 另外一种方式是 TCP/IP 协议栈在操作系统内核当中, 应用程序通过操作系统的系统调用(system call)和协议栈来进行通讯, 这样 TCP/IP 的协议栈就限定于特定的操作系统内核了, 如 Windows 就是这种方式。

(3)LWIP 的 process model:所有 TCP/IP 协议栈都在一个进程当中,这样 TCP/IP 协议栈就和操作系统内核分开了,而应用层程序既可以是单独的进程也可以驻留在 TCP/IP 进程中。如果应用程序是单独的进程可以通过操作系统的邮箱,消息队列等和 TCP/IP 层进行通信。如果应用层程序驻留 TCP/IP 进程中,那应用层程序就利用内部回调函数和 TCP/IP 协议栈通信。对于 $\mu\text{C}/\text{OS-II}$ 来说进程就是一个系统任务。LWIP 的进程模型如图 4-4 所示。

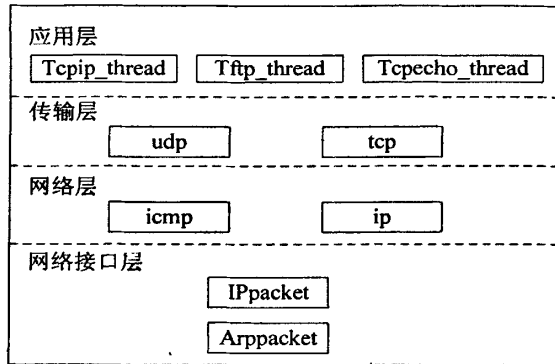


图 4-4 LWIP 进程模型

4.2.4 LWIP 在 $\mu\text{C}/\text{OS-II}$ 下的实现^[16]

LWIP 协议栈在设计时就考虑到了将来的移植问题,因此把所有与硬件、操作系统、编译器相关的部分独立出来,放在/src/arch 目录下,因此 LWIP 在 $\mu\text{C}/\text{OS-II}$ 上的实现就是修改这个目录下的文件,其它的文件一般不应该修改。

(1) 与 CPU 或编译器相关的 include 文件

/src/arch/include/arch 目录下的 cc.h, cpu.h, perf.h 有一些与 CPU 或编译器相关的定义,如数据长度,字的高低位顺序等,这应该与前面实现 $\mu\text{C}/\text{OS-II}$ 时定义的数据长度等参数是一致的。

此外,一般情况下,C 语言的结构体 struct 是 4 字节对其的,但是在处理数据包的时候,LWIP 使用的是通过结构体中不同数据的长度来读取相应的数据的。所以,一定要在定义 struct 的时候使用_packed 关键字,让编译器放弃 struct 的字节对齐。LWIP 也考虑到了这个问题,所以在它的结构体定义中有几个 PACKED_FIELD_xxx 宏。默认的时候这几个宏都是空的,可以在移植的时候添加不同的编译器所对应的_packed 关键字,在 ARM7 上的定义:

```
#define PACK_STRUCT_FIELD(x)x
#define PACK_STRUCT_STRUCT
#define PACK_STRUCT_BEGIN
#define PACK_STRUCT_END
```


(2) sys_arch 操作系统相关部分

与操作系统相关的一些结构和函数都在 sys_arch 目录下面，主要可以分为四个部分：

a) sys_sem_t 信号量

LWIP 中需要使用信号量通信，所以在 sys_arch 中应实现信号量结构体和处理函数。

```
struct sys_sem_t
sys_sem_new()      //创建一个信号量结构
sys_sem_free()    //释放一个信号量结构
sys_sem_signal()  //发送信号量
sys_sem_wait()    //请求信号量
```

由于 $\mu\text{C}/\text{OS-II}$ 已经实现了信号量 OS_EVENT 的各种操作，并且功能和 LWIP 上面几个函数的目的功能是完全一样的，所以只要把 $\mu\text{C}/\text{OS-II}$ 的函数重新包装成上面的函数就可以直接使用。

b) sys_mbox_t 消息

LWIP 使用消息队列来缓冲、传递数据报文，因此要在 sys_arch 中实现消息队列结构 sys_mbox_t 以及相应的操作函数。

```
sys_mbox_new(): 创建一个消息队列
sys_mbox_free(): 释放一个消息队列
sys_mbox_post(): 向着消息队列发送消息
sys_arch_mbox_fetch(): 从消息队列中获取消息
```

$\mu\text{C}/\text{OS-II}$ 同样实现了消息队列结构 OSQ 及其操作，但是 $\mu\text{C}/\text{OS-II}$ 没有对消息队列中的消息进行管理，因此不能直接使用，必须在 $\mu\text{C}/\text{OS-II}$ 的基础上重新实现。为了实现对消息的管理，定义了以下结构。

```
Typedef struct{
OS_EVENT* PQ;
Void* pvQEntries[MAX_QUEUE_ENTRIES];
}sys_mbox_t;
```

在以上结构中，包括 OS_EVENT 类型的队列指针 (PQ) 和队列内的消息 (pvQEntries) 两部分，对队列本身的管理利用 $\mu\text{C}/\text{OS-II}$ 自己的 OSQ 操作完成，然后使用 $\mu\text{C}/\text{OS-II}$ 中的内存管理模块实现对消息的创建、使用、删除、回收，两部分综合起来形成了 LWIP 的消息队列功能。

c) sys_arch_timeout 函数

LWIP 中每个与外界网络连接的线程都有自己的 timeout 属性，即等待超时时间。这个属性表现为每个线程都对应一个 sys_timeout 结构体队列，包括这个线程的

timeout 时间长度，以及超时后应调用的 timeout 函数。该函数会做一些释放连接，回收资源的工作。如果一个线程对应的 sys_timeout 为空(NULL)，说明该线程对连接做永久的等待。

Timeout 结构体已经由 LWIP 自己在 sys_h 中定义好了，而且对结构体队列的数据操作也由 LWIP 负责，我们所要实现的是如下函数：

```
struct sys_timeouts* syw_arch_timeouts(void)
```

这个函数的功能是返回目前正处于运行态的线程所对应的 timeout 队列指针，timeout 队列属于线程的属性，因此是 OS 相关的函数，只能由用户实现。

d) sys_thread_new 创建新线程

LWIP 可以是单线程运行，即只有一个 tcp/ip 线程 (tcpip_thread)，负责处理所有的 tcp/udp 连接，各种网络程序都通过 tcp/ip 线程与网络交互。但 LWIP 也可以多线程运行，以提高效率，降低编程复杂度，这时就需要用户实现创建新线程的函数：

```
void sys_thread_new(void(* thread)(void* arg),void* arg);
```

在 $\mu\text{C}/\text{OS-II}$ 中，没有线程(thread)的概念，只有任务(Task)。它已经提供了创建新任务的系统 API 调用 OSTaskCreate，因此，只要把 OSTaskCreate 封装一下，就可以实现 sys_thread_new。需要注意的是 LWIP 中的 thread 并没有 $\mu\text{C}/\text{OS-II}$ 中优先级的概念，实现时要由用户事先为 LWIP 中创建的线程分配好优先级。

(3) lib_arch 中库函数的实现

LWIP 协议栈中用到了 8 个外部函数，这些函数通常与用户使用的系统或编译器有关，因此，留给用户自己实现，如下：

```
16 位数据高低字节交换：u16_t htons(u16_t n)和 u16_t ntohs(u16_t n);
```

```
32 位数据大小头对调：u32_t htonl(u32_t n)和 u32_t ntohl(u32_t n);
```

```
返回字符串长度：int strlen(const char* str);
```

```
字符串比较：int strcmp(const char* str1,const char* str2,int len);
```

```
内存数据块之间的互相拷贝：void bcopy(const void* src, void* dest,int len);
```

```
内存中指定长度的数据块清零：void bzero(void* data,int n)。
```

(4) 网络设备驱动程序

开发板使用的网络芯片为 Real Tek 的 RTL8019AS 芯片，这是 ISA 10BASE-T 的以太网芯片，与 Ne2k 兼容。在 LWIP 中可以有多个网络接口，每个网路接口都对应了一个 struct netif，这个 netif 包含了相应的网络接口的属性、收发函数。LWIP 调用 netif 的方法：netif→input() 及 netif→output() 进行以太网 packet 的收、发等操作。在驱动中主要做的就是实现网络接口的收、发、初始化以及中断处理函数。驱动程序工作在 IP 协议模型的网络接口层，它提供给上层 (IP 层) 的接口函数如下：

```
网卡初始化函数：void ethernetif_init(struct netif* netif)
```

```
网卡接收函数 (从网络接口接收以太网数据包并把其中的 IP 报文向 IP 层发送，
```

在中断方式下由网卡 ISR 调用): `void ethernetif_input(struct netif* netif)`

网卡发送函数 (给 IP 层传送过来的 IP 报文加上以太网包头并通过网络接口发送): `err_t ethernetif_output (struct netif* netif, struct pbnf* p, struct ip_addr* ip addr)`

网卡中断处理函数 ISR: `void ethernetif_isr(void)`

以上的函数都可以分为协议栈本身的处理和对网络接口硬件的操作两部分, 但硬件操作是对上层屏蔽的。

4.3 小节

本章主要介绍了嵌入式操作系统 $\mu\text{C}/\text{OS-II}$ 在 S3C44B0X 上的移植以及如何用源代码开放的 TCP/IP 协议栈 LWIP 在 $\mu\text{C}/\text{OS-II}$ 上建立网络平台。 $\mu\text{C}/\text{OS-II}$ 在 S3C44B0X 上的移植主要包括以下工作: 一是修改与处理器及编译器有关的代码; 二是编写 OS_CPU_C.C 中与操作系统相关的函数; 三是编写 OS_CPU_A.ASM 中与处理器相关的函数。将协议栈 LWIP 移植到 $\mu\text{C}/\text{OS-II}$ 系统中, 使其具有 TCP/IP 的网络通信功能。主要工作包括: 修改与 CPU 或编译器相关的 include 文件与 sys_arch 操作系统相关部分, 以及 lib_arch 中库函数和网络设备驱动程序的实现。完成以上工作, 就完成了嵌入式系统软件操作平台的搭建。

第五章 系统的软件设计

软件设计是电能质量在线监测系统的重要组成部分。软件的优劣不仅关系到系统监测功能的实现，而且涉及到监测系统的可靠性和使用的方便性。从系统的软件设计的整体角度来讲，软件部分包括驱动程序的编写和应用程序的编写两部分，二者的设计息息相关。

5.1 系统驱动程序设计

设备驱动程序是操作系统内核和机器硬件之间的接口。设备驱动程序主要用于对设备进行数据的读写操作以及对设备进行配置和读取配置信息的操作，也用于使用和管理来自设备的中断。设备驱动程序为应用程序屏蔽了硬件的工作细节，这样在应用程序看来，硬件设备只是一个函数的接口。应用程序可以像操作普通接口函数一样对硬件设备进行操作。驱动程序的主要功能是管理系统硬件设备，它是操作系统的重要组成部分。设备驱动程序是内核的一部分，它完成以下的功能：

- (1) 对设备初始化和释放；
- (2) 把数据从内核传送到硬件和从硬件读取数据；
- (3) 读取应用程序传送给设备文件的数据和回送应用程序请求的数据；
- (4) 检测和处理设备出现的错误。

驱动程序的目标之一就是软件组织成一种层次结构，低层次软件用来屏蔽具体设备细节，高层次软件则为用户提供了一个简洁规范的界面。这种层次也充分体现了对上层用户的设备无关性。

从系统的结构角度来说，设备驱动的实现可以有两种方式，一种是把特定的设备驱动做成一个服务进程，当访问该项设备的进程时，通过进程间的通讯机制（如信箱、消息队列）向服务进程发出请求，由服务进程完成对设备的操作。这样无论有几个进程同时要求访问该设备，由于每个设备只有一个服务进程，并且这个服务进程每次只能接受并处理一个报文，这些访问自然就可以串行化了，而各个进程对设备操作的互斥性也就得到了保证。另一种是对设备的操作看作一组函数调用，以库函数的形式向各个进程提供设备驱动，一般而言这些函数中应该采用能保证互斥访问的措施，如信号量。

5.2 系统应用程序

本系统软件的应用程序设计主要由三部分组成，一是 ARM 处理器 S3C44B0X 通过串口 RS232 获得 GPS 接收机的输出信号，并提取时间信息；二是数据采集处理部分，指 ARM 处理器周期性的对分析站点的三相电压电流进行采集，并将采集

结果进行处理，得到相应的结果；三是网络通信传输部分，它包括串口通信和以太网通信，使数据通过 TCP/IP 协议转换，然后通过网络上传，使分析站点的主机能够接收到来自不同监测点的数据信息，方便信息的综合处理。系统的主程序流程图如图 5-1 所示。

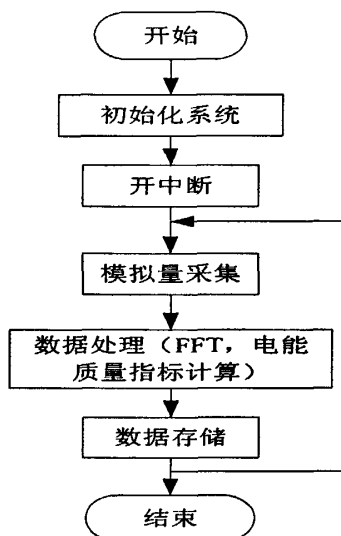


图 5-1 主程序流程图

5.2.1 GPS 接收部分^[24]

使用卫星传递标准时间信号为要求高精度时间同步系统提供了理想的手段。全球定位系统 GPS 本身不但是一个精确定位系统，而且也能传递高精度时间信息，其中用 C/A 码的时间比对精度为 $0.1\mu\text{s}$ ，与 UTC（国际标准时间）的同步精度优于 $1\mu\text{s}$ 。

接收机在任何时刻能同时接收其视野范围内的 4~8 颗卫星信号，其内部硬件电路和软件通过对接收到的信息进行解码和处理，能从中提取并输出两种时间信号，一种是同步脉冲信号，包括间隔为 1 秒的同步脉冲信号 1PPS，其脉冲前沿与 UTC 的同步误差不超过 $1\mu\text{s}$ ，以及 1PPM 和 1PPH；二是时间码信号，包括在串口输出信息中的 UTC 绝对时间（国际标准时间和日期），它是与 1PPS 脉冲前沿相对应的。

ARM 处理器 S3C44B0X 通过与 GPS 接收板 GN-80 相连的串口来发送指令控制 GN-80 的输出与接收 GN-80 的输出信息，并从接收到的数据信息中判断 GN-80 是否已经定位以及提取标准时间信息用于显示并控制采样的开始。

S3C44B0X 与 GN-80 的通信协议都遵从 GPS 接收板 GN-80 的规定，波特率为 4900，输出数据为 8 位，无奇偶校验。程序中首先按照 NMEA-0183 相关规定，向 GN-80 发送命令语句：

\$PFEC, Gpint, GGA01, ZDA00, GLL00, GSA00, GSV00, VYG00, anc00,

acc00, ast00, tst00, die00, CR, LF

这条输入语句的作用是令 GN-80 每秒通过串口只发出一条\$GPGGA 语句，去除多余的不必要的信息，在程序中预先先将这条语句的内容存放在某一位置，并在 RAM 中开辟一段数据空间保存 GGA 语句的内容。GGA 语句内容的格式如下所示：

\$GPGGA, 123456, 3444, 0000, N, 13521, 0000, E, 1, 04, 02, 00, …… ,CR LF

Field# <1> <2> <3> <4> <5><6><7> <8>……

语句所表示的含义如下：

1、UTC 时间信息

“12”：时；“34”：分；“56”：秒。

2~3、纬度

“34”：度；“44”：分；“0000”：分（小数）；“N”：北纬（S 表示南纬）；

4~5、经度

“135”：度；“21”：分；“0000”：分（小数）；“E”：东经（W 表示西经）。

6、GPS 输出是否有效

“0”表示无效（未定位）；“1”表示有效（已定位）。

ARM 通过串口 RS232 接收来自 GPS 接收板的数据，GPS 每秒输出脉宽可调的秒脉冲信号，同时输出 UTC 时间信号，当接收到秒脉冲后，立即产生外部中断信号，系统通过串口读取 UTC 时间信息。在接收时间信息完毕后，需要进行时间信息分析。根据上述 GPGGA 的语句的格式可以提取出时间信息。接收到的数据是以 ASCII 的形式存在，要进行数据转换(减去 0 的 ASCII 的值就是真正的数值)，GPS 接收的是 UTC 时间，所以在提取的时间信息上加 8 小时，即得到北京时间。处理完时间数据后，时间信息被发送到缓冲区，为下一步数据传输做准备。流程图如图 5-2 所示。

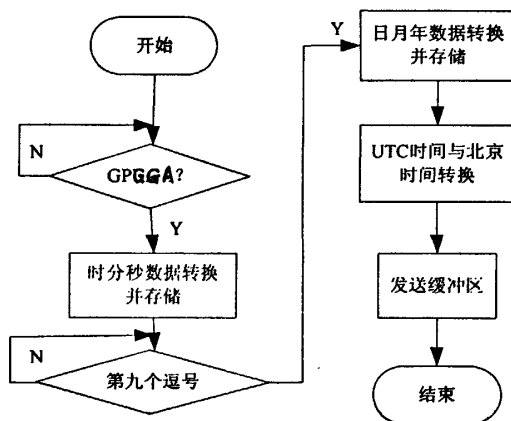


图 5-2 提取时间信息流程图

5.2.2 数据采集处理部分

模拟信号进入 ADC 进行模数转换前，首先要对 ADC 相关寄存器进行初始化，采用相关语句为：

```

rCLKCON=0x7ff8;           //时钟控制寄存器，使能 MCLK 作为 ADC 的时
                           钟源
rADCCON=0x1 | (0<<2);    //启动 A/D 转换
Delay(100);               //等待一定时间使 ADC 的参考电压稳定下来
rADCPSR=0x20;            //设置时钟预分频器
    
```

在实时内核 $\mu\text{C}/\text{OS-II}$ 下，驱动程序是通过中断来读取 AD 采样值的。方法如图 5-3 所示。

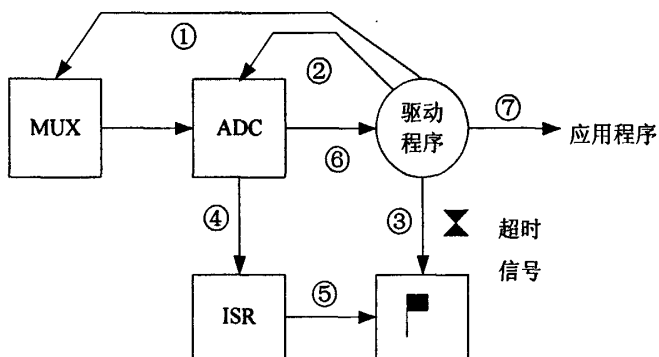


图 5-3 读取 ADC 采样值

应用程序调用驱动程序，并传递要读取的通道，驱动程序通过 MUX 选择要读取的模拟通道 (①) 开始读。接着，ADC 被触发开始转换 (②)。当模拟转换完成后，ADC 产生一个中断信号 (④)。若 ADC 转换完成，ISR 给信号量发一个信号 (⑤)，通知驱动程序，ADC 已经完成转换。驱动程序查询信号量 (③)。如果 ADC 按规定完成转换，驱动程序读取 ADC 转换结果 (⑥)。并将转换结果返回到应用程序 (⑦)。反之信号量超过系统设定的延时时间，则驱动程序不再等待下去，返回“信号错误”。

系统通过内部定时器中断启动 A/D 转换，中断周期被设置为每周波采样 128 点。其中断程序流程图如图 5-4 所示。A/D 转换完成后，电路触发中断信号给 ARM，实时数据由 ARM 通过连续读脉冲将数据存到内、外部扩展 RAM。当数据采样达到 128 点后，开始执行 FFT 单元，该单元是一个 128 点同址基 2 时间抽取 FFT 模块。

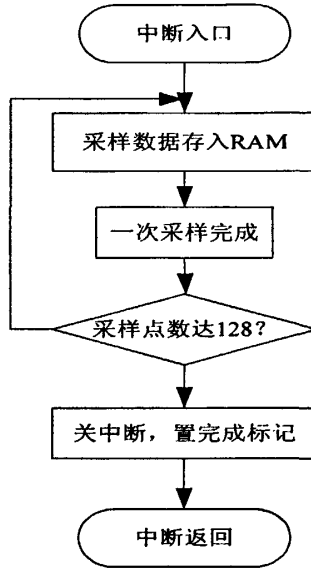


图 5-4 采样中断处理流程图

电能质量参数计算模块的核心部分为 FFT 处理单元，FFT 的原理已经在第二章进行了详细的说明。

FFT 算法通常将时域上采集的原始数据倒位序存储，最后按正常顺序输出结果 $X(0), X(1), \dots, X(k)$ 。假设一开始，数据在数组 `float dataR[128]` 中，我们将下标 i 表示为 $(b_5b_4b_3b_2b_1b_0)_b$ ，倒位序存放就是将原来第 i 个位置的元素存放到第 $(b_0b_1b_2b_3b_4b_5)_b$ 的位置上去。由于 C 语言的位操作能力很强，可以分别提取出 $b_5, b_4, b_3, b_2, b_1, b_0$ ，再重新组合成 $b_0, b_1, b_2, b_3, b_4, b_5$ ，即是倒位序的位置。

N 点 FFT 运算可以分成 $\text{LOGN}2$ 级，每一级都有 $N/2$ 个蝶形。DIT FFT 的基本思想是用 3 层循环完成全部运算 (N 点 FFT)。

5.2.3 LCD 显示部分

在系统初始化后，首先是对 LCD 控制寄存器及地址寄存器进行改写，设置一些 LCD 参数，配置 Buffer 的起始地址和 Buffer 大小等一些参数；然后是清屏，最后是显示。

实现上述操作最基本的函数是对液晶控制器指令的操作函数，其显示实现过程如图 5-5 所示。这些操作的主要功能为：

(1) LCD 初始化，主要包括对控制器的显示频率、显示行数及显示缓冲区地址的设置。

(2) LCD 清屏，由于系统上电时，显示缓冲区的数据是不固定的，显示出乱码，因此在液晶显示操作之前应将缓冲区清零。

(3) 数据显示，液晶初始化结束后，系统将采集来的信号通过处理后用文字、图形等显示到 LCD 上。

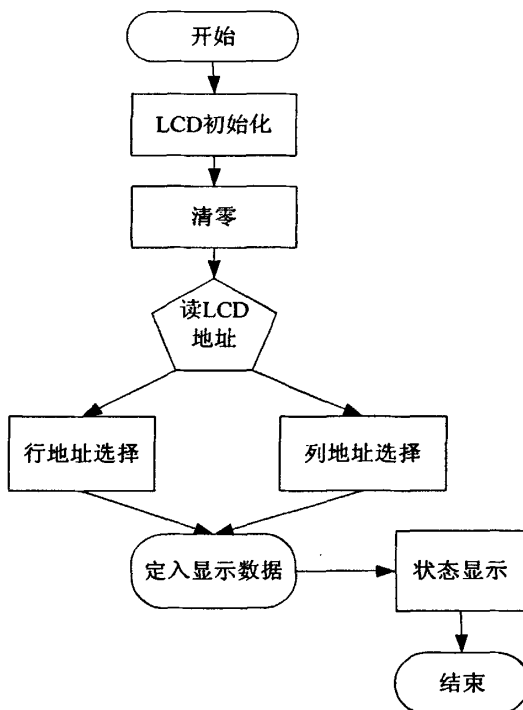


图 5-5 LCD 显示实现过程

5.2.4 键盘部分

键盘扫描程序分为主程序和读键子程序两个部分。主程序以一定的时间间隔反复调用读键子程序，从而实时响应键入。

主程序流程图如图 5-6 所示。

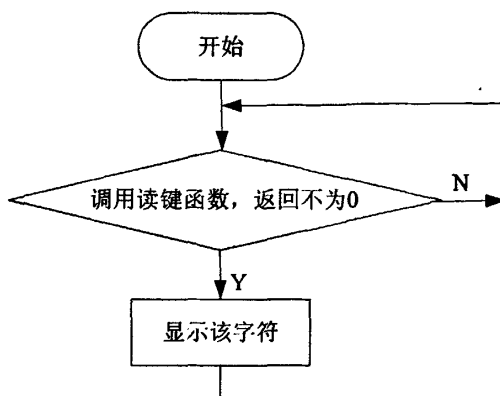


图 5-6 主程序流程图

系统子程序中先定义了键值表格如下：

```
const char Keyboard[4][4]
{
    {'7', '8', '9', '/'},
    {'4', '5', '6', '*'},
    {'1', '2', '3', '-'},
    {'0', '.', '+', '='}
};
```

然后定义扫描数码组为:

```
scanvalue[4]={0xfe,0xfd,0xfb,0xf7};
```

子程序流程图如图 5-7 所示。

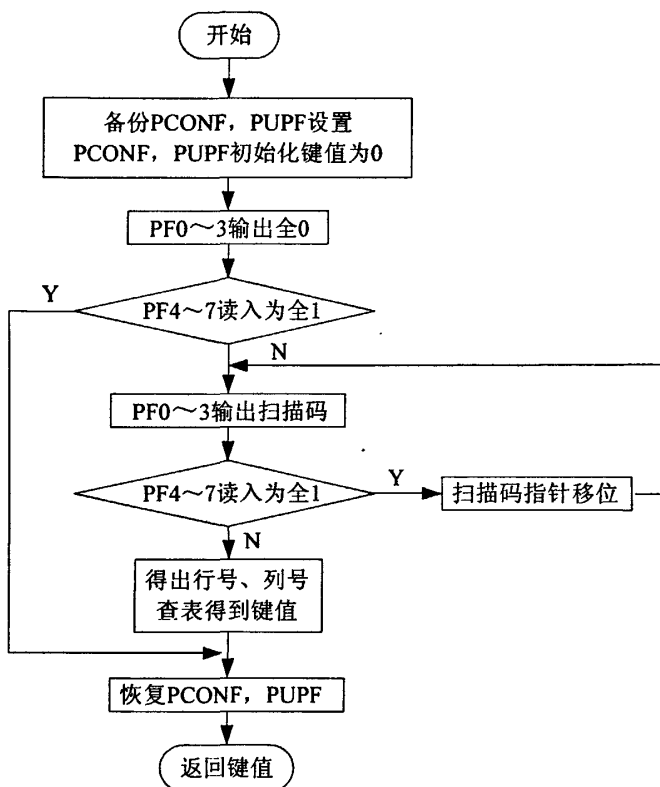


图 5-7 扫描按键子程序流程图

由于按键可能出现抖动，因此可采用下述方法来消除抖动，确认键真正被按下了。如果测到有键，并确定了键值后，延时一段时间再次测键，如果两次得到的键值相同，说明该键确实被按下了。

5.2.5 网络传输部分

(1) 串口通信部分

S3C44B0X 自带两个异步串行口控制器，每个控制器有 16 字节的 FIFO（先入先出寄存器），最大波特率 115.2kbps。每个 UART 有 7 种状态：溢出错误、校验错

误、帧错误、暂停态、接收缓冲区准备好、发送缓冲区空和发送移位缓冲区空。这些状态可由相应的 UARSTATn 或 UERSTATn 寄存器表示，并且与发送/接收缓冲区相对应的有错误缓冲区。

在串口程序设计中，要对串口进行初始化和端口配置，这样才能使串口正常工作。与 ARM 通信时，发送命令后返回的数据包是由模块自动发送的，并且正常工作时会自动发送数据包，因而串口的工作方式不能设定为查询方式，而只能设定为中断工作方式。

串口的基本操作有 3 个：串口初始化、发送数据和接收数据。这些操作都是通过串口控制寄存器进行的。

首先对 UART 口的可配置参数进行初始化，使其能够按照所要求的通讯方式进行通讯。UART 口进行初始化的设置程序为：

```

rUFCON0=0xF7;      //设置串口的 FIFO 属性，并清除 FIFO 中的内容
rUFCON0=0x0;      //关闭流控功能
rULCON0=0x3;      //设置 8 位数据位、1 位停止位、无奇偶校验串口模式
rUCON0=0x345;     //设置收发的中断模式为电平触发，关闭超时功能
rUBRDIV0=((int)(MCLK/16./baud+0.5)-1);
                    //根据波特率计算 UBRDIV0 的值
    
```

发送数据：

```

while(!(rUTRSTAT0&0x2));      //等待发送缓冲空
rUTXH0=data;                  //将数据写到数据端口
    
```

接收数据：

```

while(rUTRSTAT0&0x1==0x0);   //等待数据
data=rURXH0;                  //读取数据
    
```

串口通信程序流程图如图 5-8 所示。

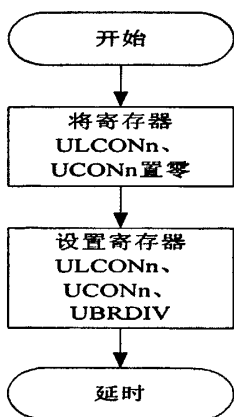


图 5-8(a) 串口初始化

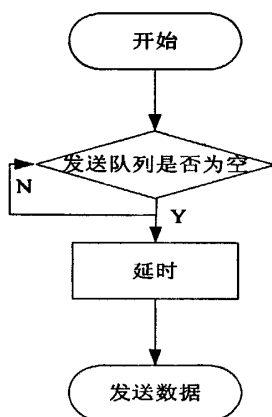


图 5-8(b) 串口发送数据

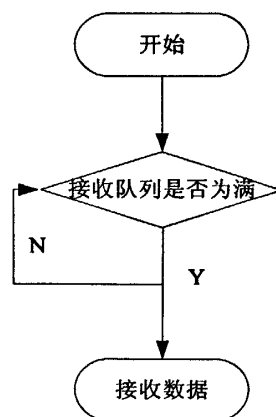


图 5-8(c) 串口接收数据

(2) 以太网通信部分^[27]

RTL8019AS 内部有 RAM，地址范围从 0x0000~0x7FFFF，其中 0x4000~0x7FFFF 用作接收和发送缓冲区。

在以太网通信中，首先要对以太网口进行初始化，初始化的第一步是复位以太网口，以太网口复位分为硬件复位和软件复位。硬件复位通过给 RTL8019AS 的 RST 引脚的一个脉冲来复位以太网口。软件复位通过写 ResetPort 达到复位，也就是给 18~1F 之间的任意一个寄存器写入任意一个数，就使得以太网口复位。第二步是设置一些寄存器的初始值。寄存器保存本机以太网的物理地址，只有和寄存器的物理地址相同的以太网帧才被接收（如果 RCR 中 PR0=0 的话）。

以太网口的第一次复位必须是硬件复位，也就是说使用以太网口前一定要先硬件复位。另外，硬件复位以后要经过大约 10ms 的等待才能对以太网口操作，特别是发送和接收操作。

发送数据包的基本步骤如下：

1、首先选择发送的起始页，一般是发送缓冲区内的页（0x40~0x4b），写入 StartPage 变量中。要发送的数据写入地址为 StartPage<<8 开始的缓冲区中，然后等待上一次发送结束。对于过大或者过小的数据包，不发送；最大最小帧的大小是以太网规定的，对于过小的帧，在发送时要填充。

TPSR 为发送起始寄存器，将 StartPage 写入 TPSR 寄存器，高字节写入 TBCRH(TBCR1)，低字节写入 TBCRL(TBCR0)。当写发送命令时，RTL8019AS 将从 TPSR<<8 地址开始发送 size 个字节的数据。

2、发送数据帧：发送缓冲区可以存储两个最大的以太网帧。规定发送缓冲区可存储两个帧，一个起始页为 SEND_START_PAGE0，另一个起始页为 SEND_START_PAGE1，两个交替使用。

接收数据包的步骤如下：

1、接收缓冲操作，当 RTL8019AS 接收到一个数据包以后，它自动将接收的数据包放到 CURR 页。如果一页放不下，则 CURR 加 1；如果 CURR = RECEIVE_STOP_PAGE，则 CURR 自动变成 RECEIVE_START_PAGE，继续写入接收的数据。

2、用户读取接收数据包，RTL8019AS 通过 Local DMA 把接收的数据写入接收缓冲区，并且自动改变 CURR 和识别缓冲区的界限，这些都不用用户干预。当一个无错的数据包接收完毕，则触发中断处理函数。接下来就是要读取数据包到分配的内存中，读取多少个可以从 ReceiveByteCount 得知。

然后将接收的数据写入网络接口层的输入队列，如果写入失败则释放内存。写入以后上层协议将提取这个数据包。

最后网卡通过中断控制器向 ARM 相应中断，中断完毕清除中断标志，使得后

来的同级和低级中断能够响应。

5.3 小节

在本章中，主要介绍了电能质量在线监测系统的软件设计部分，系统的软件设计包括驱动程序的编写和应用程序的编写两部分。对系统的驱动程序做了简单的介绍，主要说明系统应用程序的设计。在系统应用程序设计中，主要介绍了 GPS 接收部分、数据采集处理部分、LCD 显示部分、键盘部分和网络传输部分，并给出了各个部分的程序流程图，以上则构成了系统的软件设计部分。

第六章 总结与展望

1、部分实验数据分析

本文在实验过程中，对部分电能质量指标进行了测试，下面列出部分实验数据做出分析。

(1) 频率测量

用 ST3V658 型多功能计数器对频率进行多次测量并与使用本系统的测量值相比较，数据值见表 6-1。

表 6-1 频率测量数据

测量次数	频率计测量值 (Hz)	测量值 (Hz)
1	50.136	50.341
2	50.008	50.086
3	50.011	50.106
4	50.022	50.137
5	50.508	50.627
6	50.012	50.140
7	50.544	50.634

从表 6-1 可以得出频率测量的最大误差为：

$$\frac{(50.296 - 50.177)}{50.177} \times 100\% = 0.40\%$$

(2) 电压测量

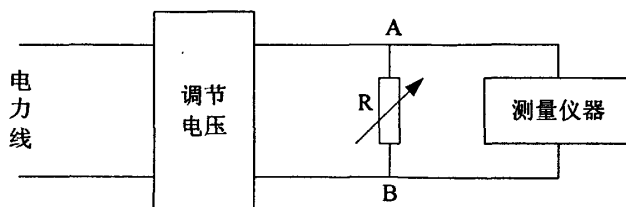


图 6-1 电压测量框图

如图 6-1 所示，用 UT56 型数字万用表对 A、B 两端电压进行多次测量，并与使用本系统测量得到的数据相比较，数据值见表 6-2。

表 6-2 电压测量数据

测量次数	电压表测量值	测量值
1	220.00	220.24
2	220.05	220.26
3	220.21	220.38
4	220.13	220.28
5	220.09	220.31
6	220.15	220.36
7	220.04	220.23

从表 6-2 可以得出电压测量的最大误差为：

$$\frac{(220.295 - 220.095)}{220.095} \times 100\% = 0.80\%$$

(3) 电流测量

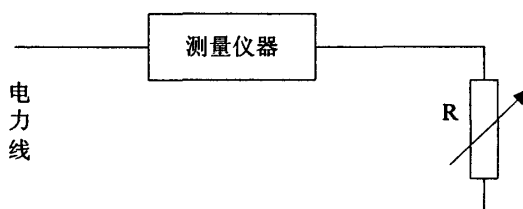


图 6-2 电流测量框图

如图 6-2 所示，用 UT56 型数字万用表 20A 电流档对电流进行测量，改变负载滑线变阻器的阻值，得到电流的多次测量值。与使用本系统测量得到的数据相比较，数据值见表 6-3。

表 6-3 电流测量数据

测量次数	电流表测量值	测量值
1	1.5	1.54
2	2.0	2.05
3	3.0	3.03
4	3.5	3.54
5	4.0	4.03
6	4.5	4.55
7	5.0	5.04

从表 6-3 可以得出电流测量的最大误差为：

$$\frac{(3.40 - 3.36)}{3.36} \times 100\% = 1.10\%$$

从以上数据可以看出, 本文设计的电能质量监测装置能够实现预期的功能, 精度达到了要求, 得到了比较满意的效果。

2、总结与展望

电力系统的电能质量在线监测是近年来得到迅速发展并受到电力系统有关运营、管理、科研等部门工程技术人员日益关注的一个新的研究领域。它的发展对电力系统的安全运行具有重要意义, 并具有明显的经济效益和社会效益。随着传感器技术、计算机技术、通信技术、智能技术等相关领域技术的发展以及电能质量监测技术本身的发展, 越来越多的新技术在电能质量监测中得到应用。本文主要就电能质量在线监测系统在硬件和软件上的设计进行了研究与实现。

本课题的具体工作总结如下:

(1) 对配电网电能质量参数五项指标的基本概念进行了介绍, 并对相关指标的测量方法进行了分析。

(2) 对系统的硬件平台进行了设计, 硬件部分以 ARM 处理器 S3C44B0X 为主 CPU, 针对电能质量监测中数据采集和处理的实际特点, 设计了各部分的电路。

(3) 在电能质量在线监测系统中, 同步时钟是一个重要组成部分, 利用 GPS 接收机提供一个高精度的同步时钟, 可以提高系统之间的对时精度, 降低因各种原因所造成的误差。本文分析了 GPS 接收机时间信号的提取与接收。

(4) 系统在实时性和精度上要求都比较高, 处理器要处理多个任务, 因此操作系统就为系统各个任务的合理调度提供了重要保障。 $\mu\text{C}/\text{OS-II}$ 实时操作系统结构简单, 易于移植, 所以为 S3C44B0X 处理器嵌入了 $\mu\text{C}/\text{OS-II}$ 操作系统。

(6) 以太网通信是网络通信中应用比较广泛的技术之一, 本文研究了一个 TCP/IP 协议栈 LWIP, 并说明了如何把 LWIP 协议栈移植到 $\mu\text{C}/\text{OS-II}$ 操作系统中去。

(5) 最后对系统的相关应用软件进行了分析说明。

然而, 由于时间和条件限制, 本系统还有很多待改进的地方:

(1) 系统硬件在加强稳定性和可靠性还可进一步提高。

(2) 系统的应用软件部分如采集后数据的处理, 采用了 FFT 算法, 而 FFT 有其固有的局限性, 它只适用于平稳信号的分析, 而配电网的电压、电流随时发生着变化, 所以在谐波测量上还可以引入小波变换方法。

(3) 整个系统是通过 GPS 信号进行同步授时的, GPS 接收机在长时间内的统计误差非常小, 但是在偶然情况下, 还是会出现比较大的偏差, 因而同步时钟的校正有待于进一步的研究, 并采取相应的改进措施。

总之, 提高电能质量在线监测技术水平, 加强电能质量管理是电力系统的一项重要工作, 而随着各种新技术的发展, 电能质量监测系统也必然会朝着便捷化、智能化、网络化的方向发展。

参考文献

- [1] 马维华. 嵌入式系统原理及应用. 北京: 北京邮电大学出版社, 2006.15~35
- [2] 周忠谟, 易洁. GPS 卫星测量原理与应用. 北京: 测绘出版社, 1997.5~23
- [3] 肖雁鸿, 毛筱, 周靖林. 电力系统谐波测量方法综述. 电网技术, 2002, 26(6): 61~64
- [4] 牛胜锁. 基于 GPS 的电力系统交流电量同步采集器的研究: [硕士学位论文]. 华北电力大学, 2004
- [5] 林海雪. 电力系统的三相不平衡. 北京: 中国电力出版社, 1998. 10~18
- [6] 蔡邻. 电力系统频率. 北京: 中国电力出版社, 1998. 8~20
- [7] 肖湘宁. 电能质量分析与控制. 北京: 中国电力出版社, 2004. 12~25
- [8] 杨坤. 电能质量监测与分析系统: [硕士学位论文]. 四川大学, 2004
- [9] A Lakshmikanth, Medhat M.Morcros. A power quality monitoring system a case study in DSP-based solutions for power electronics. 2001, 50(3): 724~730
- [10] McGranaghan M. Trends in power quality monitoring. IEEE power Engineering Review, 2001, 21(10): 3~9
- [11] McEachern A. Roles of intelligent systems in power quality monitoring: past, present and future. Power Engineering Society Summer Meeting, Vancouver, British Columbia, Canada, 2001: 1103~1105
- [12] 李兰芳. 电能质量在线监测系统: [硕士学位论文]. 武汉大学, 2004
- [13] 毕春跃, 基于 $\mu\text{C}/\text{OS-II}$ 的嵌入式系统研究与应用. 现代计算机, 2007, 2: 33~36
- [14] 刘俊飞, 刘宏达, 徐俊红. 基于锁相同步采样的谐波测量. 应用科技, 2005, 32(5): 27~29
- [15] 杨磊. 基于嵌入式实时操作系统的电能质量分析仪软件的研究: [硕士学位论文]. 华北电力大学(北京), 2005
- [16] 阙大顺, 王近涛. LWIP协议在 $\mu\text{C}/\text{OS-II}$ 系统上的移植和实现, 舰船电子工程, 2006, 4: 89~91
- [17] XU W, LIU Y. A method for determining customer and utility harmonic contributions at the point of common coupling. IEEE Trans on Power Delivery, 2000, 15(2): 804~811

- [18] 孙鹤林, 吕元龙, 田跃军. 基于 GPRS 与虚拟仪器的远程电能质量监测系统. 继电器, 2007, 35(1): 59~62.
- [19] 张宇, 陈明, 瞿赟. 基于 ARM7 的嵌入式数据采集系统设计. 电子测量技术, 2007, 30(1): 99~101
- [20] 赵刚, 施围, 林海雪. 闪变值计算方法的研究. 电网技术, 2001, 25(11): 16~18
- [21] 同向前, 余健明. 电网电压中波动信号检出方法的研究. 仪器仪表学报, 1998, 19(3): 235~238
- [22] Il-Yop Chung, Dong-Jun Won, Joong-Moon Kim, Seon-Ju Ahn, Seung-Il Moon, Jang-Cheol Seo, Jong-Woong Choe. Development of power quality, diagnosis system for power quality improvement, IEEE Power Engineering Society General Meeting, 2003,2(7): 13~17
- [23] 徐鹏煜. 基于ARM+Linux的高性能电能质量分析仪: [硕士学位论文]. 华北电力大学, 2006
- [24] 何香玲, 郑钢, 范秋华. GPS 定位信息的采集和显示. 自动化仪表, 2007, 28(3): 40~45
- [25] 潭子求. 基于DSP的电能质量在线监测装置的研究: [硕士学位论文]. 武汉大学, 2004
- [26] 张鹏, 王少荣. 电网状态监测系统GPS同步时钟的稳定性研究. 继电器, 2004, 32(23): 18~22
- [27] 曹宇, 巍丰, 胡士毅. 用 51 单片机控制 RTL8019AS 实现以太网通讯. 电子技术应用, 2003, 1: 21~23
- [28] 李学桥, 陈晓雷. 基于 ARM 的远程数据采集系统的设计与实现. 仪表技术与传感器. 2004, 2: 29~31.
- [29] Angrisani L, Daponte P, Apuzzo M D, et al. A measurement method based on the wavelet transform for power quality analysis. IEEE Trans on Power Delivery, 1998, 13(4): 990~998
- [30] 李分秋, 吴为麟. 基于ARM的电能质量监测控制器的研制. 江南大学学报(自然科学版), 2006, 5(2): 199~203
- [31] 李国庆, 王佐勋. 基于数字信号处理器的一种新型电能质量监测仪的设计. 东北电力学院学报, 2005, 25(1): 1~5
- [32] 付晓军, 夏应清. 电能质量监测系统的嵌入式以太网接口. 电能质量专辑, 2005, 10: 70~72

- [33] 胥静. 嵌入式系统设计于开发实例详解—基于ARM的应用. 北京: 北京航空航天大学出版社, 2005. 75~105
- [34] 张绮文, 谢建雄, 谢劲心. 嵌入式常用模块与综合系统设计实例精讲. 北京: 电子工业出版社, 2007. 207~245
- [35] Ananth I, Morcos M M. A power quality monitoring system: A case study in DSP based solutions for electric power industry. IEEE Power Engineering Review, 1999, 19(7): 47~50
- [36] Possion O, Rioual P, Meunier M, Detection and measurement of power Quality disturbances using wavelet transform. IEEE Trans on Powe Delivery, 2000, 15(3): 1039~1044.
- [37] 任先文, 刘福涛. 电能质量三项国家标准的在线监测与分析系统实现. 东北电力学院学报, 2001, 22(1): 68~72
- [38] 高翔, 陶炜. DSP系统的GPS高精度时钟设计和实现. 华北电力技术, 2003, 3: 1~3
- [39] 段成刚, 欧阳森. 新型在线实时电能质量监测设备的设计. 电网技术, 2004, 28(2): 60~63
- [40] 孙俊, 周俊华, 李正明, 陈为军. 基于嵌入式系统的远程电网电能质量监测系统. 2006, 4: 93~95
- [41] Daponte P, Di Penta M, Mercurio G, Transientmeter: a distributed measurement system for power quality monitoring. Proceedings Ninth International Conference on, Orlando, FL, USA, 2000: 1017~1022
- [42] S chen. Open Design of network power quality monitoring system. IEEE Transaction instrumentation and measurement, 2004, 53(2): 597~601
- [43] 于海生, 潘松峰, 吴贺荣. 基于复序列FFT和锁相原理的电参数测量. 电网技术, 2000, 24(3): 59~61
- [44] 陈晓平. 基于嵌入式的电能质量监测系统的研究于应用: [硕士学位论文]. 湖南大学. 2006
- [45] 孙俊, 周俊华, 李正明. 基于嵌入式系统的远程电网电能质量监测系统. 中国农村水利水电, 2006, 4: 93~95

致 谢

首先感谢华北电力大学给予了我学习的机会和条件，在这两年半的研究生学习中我不仅学到了丰富的知识，同时也在各位老师和同学的教导和帮助下学到了很多做人、做事的道理。

在本课题的研究过程中，我的导师谢志远教授给予了我热心的支持和帮助，本文是在导师悉心的指导下完成的。导师不但在论文的研究方面给予了细致的指导，提出了许多有益的建议，而且在生活上也给了我很多关心和帮助，是一位良师益友。谢老师学识渊博，学术观点新颖，治学作风严谨，对科学执着追求的精神都使学生终生难忘，谨在此对辛勤培养我的导师谢志远教授致以我崇高的敬意和衷心的感谢！

感谢学术上相互探讨进步的实验室同学赵敏华、许国强、樊丰华、时磊，以及研究生期间的各位同学，他们的鼓励和帮助，令我的学习生活增色不少。在此无法一一列举，一并谨表谢意！

最后，我要感谢我的家人，是他们的支持使我能够顺利的完成学业。

在学期间发表的学术论文和参加科研情况

- [1] 李航, 谢志远. 配电网电能质量监测中数据采集系统的研究. 河北经贸大学学报综合版, 2007, 7: 72~74
- [2] 赵敏华, 谢志远, 李航. 基于 TMS320LF2407 的馈线终端装置设计. 国外电子元件. 2007, 8: 25~29