

摘 要

随着我国经济的高速发展，城市用电量的日益增加，配电网的安全可靠性也日益重要。它关系到千家万户的用电，特别随着电力市场的逐步建立，用电质量成为一个城市供电的重要参数，对配电变压器的运行参数进行监测是实现配电自动化的基础。uClinux 操作系统是从 Linux 衍生出来的一种操作系统，它是专为无 MMU 的微控制器开发的嵌入式 Linux 操作系统。它支持众多嵌入式处理器类型，具有完善的各类驱动支持。

本文根据我国配网现状及配电自动化的需求，在分析国内外配变监测终端研究与发展状况的基础上，提出一种基于嵌入式 uClinux 操作系统，集电表功能和通信功能于一体的配变监测终端的设计方案。硬件电路设计方面，主控板采用 ARM 功能强大的 32 位处理器 AT91M40800，全隔离的标准 RS485/RS232 接口、汉字显示 LCD；电能表数据采集部分采用基于 DSP 技术且带有谐波测量功能的高精度电能测量芯片 ATT7022B；软件设计方面主要讨论了系统开发环境的建立，包括 Bootloader 程序设计、内核移植和裁剪、设备驱动程序设计，终端各个模块应用程序设计。

经过调试、运行本终端具有测量精度高、性能稳定、体积小、重量轻、功耗低、操作简便、一表多用等特点，各项技术指标符合《GB/T 17883-1999》《GB/T 17882-1999》《GB/T 17215-2002》《DL/T 614-1997》和《DL/T 645-1997》等国家标准以及行业标准，适用于电能测量及用电自动化管理领域，较好的符合了目前国内的电网状况。

关键词：AT91M40800，监测终端，uClinux，ATT7022B，ARM

Abstract

With the rapid development of our country's economy, the demand for energy is growing. The safety and reliability of power distribution network has become more and more important. It is vital to millions of households the use of electricity, particularly with the gradual establishment of the electricity market, the quality of electricity supply as a city of the important parameters, monitoring to the distribution transformer working parameters is the base to realize the distribution. So the level of distribution transformer monitoring is one of the major standards to measure the level of distribution automation. uClinux is derived from Linux and inherits lots of virtues of Linux such as open source, it has special implements in the case of not having MMU and supports wide range processors and many drivers.

A new design of intelligent distribution monitoring terminal was proposed in this paper. The new design is based on the analysis of the current situation of distribution power system, the needs of distribution automation in our country and the international and domestic TTU research. This paper presents a transformer monitoring terminal design plan which is integrated function-meter and communications features based on uClinux embedded operating systems, dual-CPU architecture. In hardware circuit designing, the control panel used the 32-bit embedded microprocessors AT91M40800, all isolated standard RS485/RS232 communication interface and LCD display Chinese characters to design. ATT7022B is adopted in power meter data acquisition based on DSP thchnology and used multifunctional three-phase energy metering. In the aspect of software programming, this paper discusse the establishment of system development environment, including bootloader program design, the replant uClinux, establish file system and device driver design, programme modules-applications on terminal.

Through debugging and working, the terminal has high accuracy, small size, light wight, low power consumption, easy to operate, various technical parameters to meet《GB/T 17883-1999》《GB/T 17882-1999》《GB/T 17215-2002》《DL/T 614-1997》《DL/T 645-1997》 national standards and industry standards. And it makes appropriate to the field of automation and electric energy measurement, which is better to the present domestic power networks.

Key Words: AT91M40800, distribution monitoring termianl, uClinux, ATT7022B, ARM

独创性声明

本人声明，所呈交的论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得武汉理工大学或其它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。

签名：师娟娟 日期：2008.5.19

关于论文使用授权的说明

本人完全了解武汉理工大学有关保留、使用学位论文的规定，即学校有权保留、送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内 容，可以采用影印、缩印或其他复制手段保存论文。

(保密的论文在解密后应遵守此规定)

签名：师娟娟 导师签名： 日期：2008.5.19

第 1 章 绪论

1.1 课题背景及意义

配电自动化(DA-Distribution Automation)是一项集计算机技术、通信技术、控制技术和现代化设备及管理于一体的综合信息管理系统,其目的是提高电网的供电可靠性,改进电能质量,向用户提供优质服务,降低运行费用,减轻运行人员的劳动强度。配电网自动化从广义上讲,是指利用现代先进的电子技术,计算机网络通信技术,实现对配电网正常运行时的控制、检测和故障时的快速处理,以及配电的生产管理、设备管理的自动化。从狭义上讲,是指配电网故障的自动化处理,即包括故障检测,故障定位、隔离和非故障区的供电恢复整个过程。

我国配电网自动化的发展是电力市场和经济建设的必然结果,长期以来配电网的建设未得到应有的重视,建设资金短缺,设备技术性能落后,事故频繁发生,严重影响了人民生活和经济建设的发展,随着电力的发展和电力市场的建立,配电网的薄弱环节显得越来越突出,形成电力需求与电网设施不协调的局面。由于配电网中的监控设备点多面广,许多情况下不能把所有的监控设备终端都直接连接到配网主站上,特别是在有线通信方式下更是如此。因此,配电自动化系统一般由配电主站——子站——终端三层结构组成,各站端之间通过通信媒介完成通信和控制,结构如图 1-1。

配电主站从各个配电子站获取配电网信息,对配电网进行全面监测和控制,分析配电网的运行状态,实现故障诊断、故障隔离和非故障区的供电恢复,对整个配电系统进行有效的管理。

配电子站负责本区域内配电网的信息采集处理,完成本区域内故障处理与控制等功能,并根据主站的要求将信息转发至主站系统;同时,接收主站下发的信息,在特殊情况下与主站配合完成配电网的调度管理任务。对于较大规模的城市,由于配电网信息分散复杂,直接将信息采集至主站系统会造成主站系统通信端口数目太多以及主站系统负荷压力大的局面,为了保证系统实时性,配电网一般设置子站来解决上述问题。

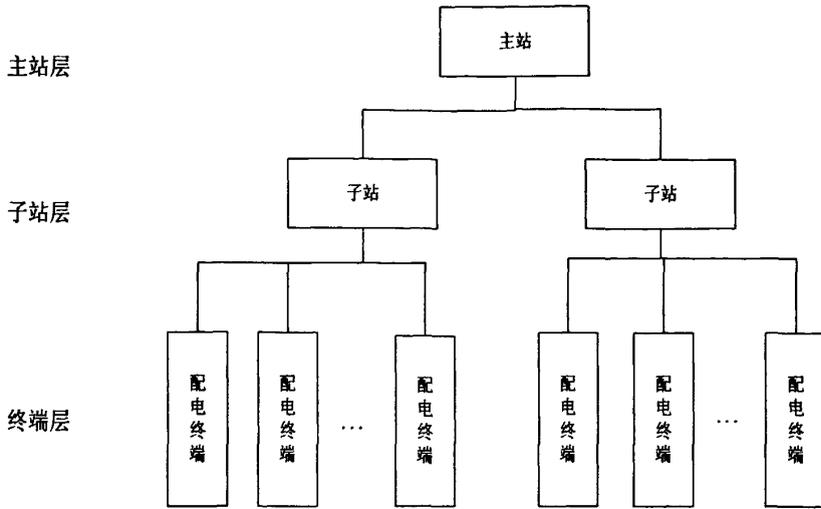


图 1-1 配电自动化系统结构图

配电终端层设备主要完成对配电网设备各种数据的直接采集、实时控制以及数据通信等功能。该层包括开关监控、配变监测单元、环网柜、开闭所、馈线通信设备等。随着超大规模集成电路技术的不端发展和新型通信技术特别是无线通信技术的不断出现，配电自动化系统的三层结构在许多情况下也可以由两层结构代替，即仅保留主站层和终端层，省去了子站层，达到简化系统结构，降低投资成本，提高系统运行效率的目的。本文主要研究的是位于配变自动化系统终端层的设备之一配变监测终端。

1.2 国内外研究的现状

配变监测终端用于配电变压器的各种运行参数的监视、测量与控制的远方终端。它是配电网监控终端层，负责数据的采集和操作命令的执行。配变监测终端在配电自动化领域，处在一个非常重要的地位。国内最早的配变自动化终端装置一般都依赖进口设备，但是，随着国内自动化水平的提高，配电自动化的关键设备由依赖进口逐步转向相信国产设备，配变终端已有了国产的入网许可产品，其功能与性能价格比更有利于各供电部门选用。

国外有很多较先进的配变监测终端的产品。如加拿大 Power Measurement Ltd.生产的 3700 系列高级电力综合监控仪。其中 3720ACM 集采集、测量、控制、通信功能于一体，装置也采用了高性能的 DSP 芯片用于数字信号处理。近

年来，该公司又推出了新产品，增加了微机继电保护功能。但是，国外产品由于其面板按键操作较复杂，技术实现上有所差异，加之价格昂贵，也不完全适合我国配电网的现状和要求。

配变监测终端的发展同配电自动化息息相关，因为配电自动化没有什么发展，在 80 年代末，配变监测终端的研究工作几乎为零。随着，配电自动化在我国不断的发展，配变监测终端的研制工作也有很大的进步。由于微处理机具有速度快、功能强、应用灵活等特点，intel18086, intel80286, intel18098, MCS-51 等高性能微处理机广泛地应用于配变监测终端中。随着电力调度自动化的发展，尤其是配电自动化的发展，要求配变监测终端有更大的容量和更丰富的功能，新型的，高性能的产品不断推出。

随着我国经济的高速发展，城市用电量的日益增加，配电网的安全可靠性也日益重要。它关系到千家万户的用电，特别随着电力市场的逐步建立，用电质量成为一个城市供电的重要参数，而用电质量的好坏通常要通过对配电变压器的运行状态进行实时监测。因此配变监测水平也是衡量配电自动化水平的一个主要标准之一。配电监测终端是从传统的电气测控等装置基础上发展起来的，电测量理论及仪表技术的发展历经了早期、初期、中期和近期四个阶段。早期和中期的电测量技术主要是以模拟量测量为主。20 世纪 50 年代，数字电子技术和微电子技术的引入，促进了电测量及其仪表技术的发展，模拟式电测仪表逐渐在越来越多的场合被数字式仪表所代替。1974 年出现的电压、电流波形等间隔采样技术，使数字电子技术在测量领域作用日益增大，成为电测与仪表技术步入中期发展阶段的重要标志。在这一阶段，以微计算机、独立操作系统、各种标准式总线结构为特征，可相互通讯、可扩展式仪器和自动测试系统以及相应的测量技术得到了蓬勃发展，并逐渐走向成熟。20 世纪 80 年代中期以来，电测与仪表技术进入了迅猛发展的近期阶段，仪器具有了控制、存储、运算、逻辑判断及自动操作等智能化特点，并在测量准确度、灵敏度、可靠性、自动化程度及解决测量问题的广度和深度等方面均有了明显的进步。

到了 90 年代中期，计算机和微电子技术的发展以及配电网自动化系统的实施，都进一步促进了电能质量问题的研究及其监测装置的研制。尤其是嵌入式芯片从 8 位发展到 16 位、32 位，处理速度与集成度越来越高，为装置的设计提供了强大的技术支持。传统的电力系统测量、监控、远动等硬件资源相互独立，往往造成二次设备的重复投资，而且接线复杂也降低了系统的可靠性。随着微

电子技术等先进技术的运用,出现了新一代智能监控单元,它是集保护、测量、控制、通信等多种功能于一体的综合化智能装置。就本文所研究的配变终端而言,它可取代传统的感应式电表、变送器及 RTU,实时地提供电网的电压、电流、有功功率、无功功率、电度等参数,并具有一些就地控制功能;另一方面,它还可以与远程终端技术相结合,通过通信将系统的计量信息传送到同一电力局的不同部门以及其它电力局,用于系统调度、电能控制等。

1.3 本文主要内容与结构

本课题采用 ARM 技术,基于嵌入式 uClinux 操作系统,提出一种集电表功能和通信功能于一体的配变监测终端的设计方案,在低成本的前提下满足配变监测终端各种参数的功能需求。论文共分为六章,主要的内容概要如下:

第 1 章 阐述了本课题的背景和意义,简要的说明了国内外研究的现状,阐述了论文的内容与组织结构。

第 2 章 研究了配变监测终端的工作原理,多功能电能表协议,uClinux 操作系统,根据系统功能需求提出了一种带有表计和通信功能的基于 uClinux 操作系统的配变监测终端的总体设计方案。

第 3 章 着重讨论了配变监测终端主控板硬件各个模块电路设计,详细讨论了计量模块的原理及校表原理与硬件设计电路,以及部分模块的抗干扰设计。

第 4 章 配变监测终端的软件设计。主要包括系统开发平台的建立,应用程序基于 uClinux 操作系统,采用模块化设计思想,给出了软件结构和设计流程。

第 5 章 终端硬件软件调试。主要包括主控板调试、交采板调试、整机调试。

第 6 章 总结和展望。

第 2 章 技术研究基础与系统总体设计

2.1 配变监测的原理

配变监测系统由配变监测终端和主站管理软件两部分组成。配变监测终端在电力系统配电自动化中担负着配电网参数的监测、数据处理、数据传输和控制的任務。因此，配变监测终端一般由信号及数据采集电路、数据分析与处理电路、数据通信接口电路、控制输出电路以及人机接口电路等组成，其原理结构如图 2-1 所示。

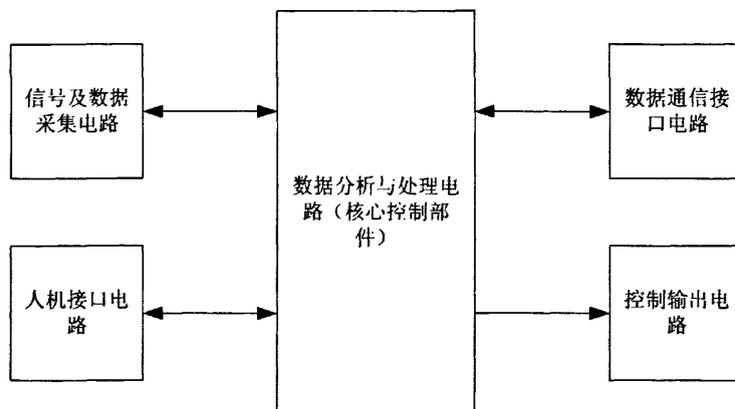


图 2-1 配变监测终端原理结构

配变监测终端利用信号及数据采集电路实时地采集供电系统的各项电力参数(如:三相电压、三相电流、有功功率、无功功率、功率因数、线频率等)及状态信息(断路器、隔离开关等设备的工况以及告警之类信号)。然后，终端将所采集的数据送到核心控制部件微处理器进行数据分析与计算处理:核心控制部件根据系统的要求，一方面将计算、处理与统计的数据进行存储以备查询或下次调用，或者通过数据通信接口将数据传输给配电站(或直接传给控制主站)管理系统及其它终端设备；另一方面，根据数据采集与处理的结果实时地对补偿功率因数，或输出控制信号。同时，核心控制部件还可以接收来自系统控制主站的控制命令，以完成系统的数据传输、控制功能及实现系统的联网。人机接口电

路主要由键盘与显示电路组成，用于终端的参数设置、数据修改、数据显示及查询等。数据通信接口电路用于终端与子站(或主站)及其它设备之间的数据传输，该电路通常设计成多个接口电路，以适应电力系统不同通信方式的需要，控制输出电路主要用于系统无功投切和紧急或特殊情况下的开关分合。

配变监测管理系统将所有的分布在配电变压器处的配变监测终端联接成一个系统，并将所有配变的运行状态、配电网路上所有负荷点的电能计量数据在配电网模型基础上进行统一管理，形成配电管理系统本身及其它配电管理子系统共享的数据资源，从而实现配变运行状态分析、全网无功优化、线损分析统计考核、电力市场分析、系统经营指标分析等功能^[32]。

2.2 多功能电表协议

多功能电能表通信协议（Multi-function watt-hour meter communication protocol）是由中华人民共和国电力工业部 1998 年批准实施的电力行业通信标准。制定协议的目的是为统一和规范多功能电能表的费率装置与数据终端设备进行数据交换时的物理连接和协议。多功能电能表通信协议代号为 DL/T 645-1997，行业通称 645 协议，以下简称 645 协议。

645 协议适用于本地系统中多功能电能表的费率装置与手持单元（HHU）或其它数据终端设备进行点对点或一主多从的数据交换方式，规定了它们之间的物理连接、通信链路及应用技术规范^[5]。

（1）物理层

645 协议适用的物理层有以下几种：接触式光学接口、调制型红外光学接口、RS-485 标准串行电气接口。

（2）链路层

645 协议为主—从结构的半双工通信方式。手持单元或其它数据终端为主站，费率装置为从站。每个费率装置均有各自的地址编码。通信链路的建立与解除均由主站发出的信息帧来控制。每帧由帧起始符、从站地址域、控制码、数据长度、数据域、帧信息纵向校验码及帧结束符等 7 个部分组成。每部分由若干字节组成。

① 字节格式

每字节含 8 位二进制码，传输时加上一个起始位(0)、一个偶校验位和一个停止位(1)，共 11 位。其传输序列如图 2-2 。D0 是字节的最低有效位，D7 是字节的最高有效位。先传低位，后传高位。

②帧格式

帧是传送信息的基本单元，它的格式如图 2-3 所示。帧起始符 68H，标识一帧信息的开始，其值为 68H=01101000B。地址域 A0~A5：地址域由 6 个字节构成，每字节 2 位 BCD 码。地址长度可 12 位十进制数，可以为表号、资产号、用户号、设备号等。具体使用可由用户行决定。当使用的地址码长度不足 6 字节时，用十六进制 AAH 补足 6 字节。低址位在先，高地址位在后。当地址为 999999999999H 时，为广播地址。

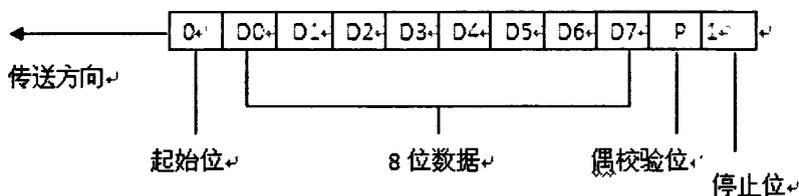


图 2-2 字节传输序列

说明	代码
帧起始符	68H
地址域	A0
	A1
	A2
	A3
	A4
	A5
帧起始符	68H
控制码	C
数据长度域	L
数据域	DATA
校验码	CS
结束符	16H

图 2-3 帧格式

控制码 C：控制码的格式如图 2-4 所示：

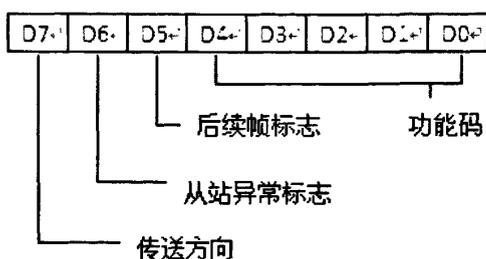


图 2-4 控制码格式

D7=0: 由主站发出的命令帧

D7=1: 由从站发出的应答帧

D6=0: 从站正确应答

D6=1: 从站对异常信息的应答

D5=0: 无后续数据帧

D5=1: 有后续数据帧

D4~D0: 请求及应答功能码

00000: 保留; 00001: 读数据; 00010: 读后续数据; 00011: 重读数据; 00100: 写数据; 01000: 广播校时; 01010: 写设备地址; 01100: 更改通信速率; 01111: 修改密码; 10000: 最大需量清零。

数据长度 L: L 为数据域的字节数。读数据时 $L \leq 200$, 写数据时 $L \leq 50$, $L=0$ 表示无数据域。

数据域 DATA: 数据域包括数据标识、数据和密码等, 其结构随控制码的功能而改变。传输时发送方按字节进行加 33H 处理, 接收方按字节进行减 33H 处理。

校验码 CS: 从帧起始符开始到校验码之前的所有各字节的模 256 的和, 即各字节二进制算术和, 不计超过 256 的溢出值。

(3) 传输层

①前导字节: 在发送帧信息之前, 先发送 1~4 个字节 FEH, 以唤醒接收方。

②传输次序: 所有数据项均先传送低位字节, 后传送高位字节。

③差错控制: 字节校验为偶校验, 帧校验为纵向信息校验和, 接收方无论检测到偶校验出错或纵向信息校验和出错, 均放弃该信息帧, 不予响应。

④数据标识

除测量值以外，本标准将计数值，最大需量发生时间，瞬时电压、电流、功率值等归为变量类，将日历、时间、用户设置值、费率装置的特征字、状态字、费率时段等归为参变量类。费率装置中有各种不同类型、不同属性的数据。本标准采用四级树状结构的标识法来表示这些数据。用 2 个字节的 4 个字段分别标识数据的类型和属性，这 2 个字节为 DI_1 和 DI_0 ，4 个字段分别为 DI_{1H} 、 DI_{1L} 、 DI_{0H} 和 DI_{0L} ，其中 DI_{0L} 为最低级标识字段， DI_{1H} 为最高级标识字段。

用 DI_{1H} 标识数据的类型，其标识如图 2-5：

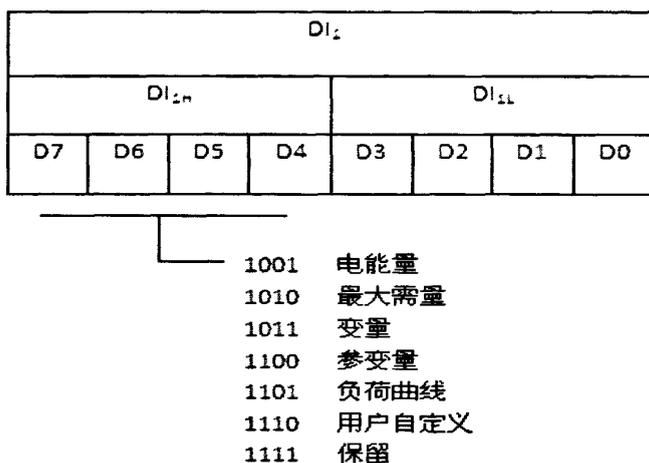


图 2-5 数据标识结构图

用 DI_{1L} 、 DI_{0H} 、 DI_{0L} 标识数据的不同属性时，对于电能量和最大需量数据，由于其具有多个属性，如时域性（当前值、上月值、上上月值等）、分类属性（有功、无功）、供电方向属性（正向、反向）、费率属性（总量、不同费率的量）等，具体详细的数据标识可查询 DL/T 645-1997 协议。

（4）应用层

应用层表示用户在实际通信应用层面上需要的命令，它反映了协议所应用的范围和支持的功能。645 协议应用层主要支持读数据、读后续数据、重读数据、写数据、广播校时、写设备地址、更改通信速率、修改密码等九种命令。

下面以读数据举例说明：

①主站请求帧

功能：请求读数据

控制码：C=01H

数据长度：L=02H

帧格式如图 2-6 所示:

68H	A0	...	A5	68H	01H	02H	DI ₀	DI ₁	CS	16H
-----	----	-----	----	-----	-----	-----	-----------------	-----------------	----	-----

图 2-6 读数据请求帧格式

②从站正常应答帧

功能: 从站正常应答

控制码: C=81H, 无后续数据帧

C=A1H, 有后续数据帧

数据长度: L=02H+m (数据长度)

无后续数据帧格式, 如图 2-7 所示:

68H	A0	...	A5	68H	81H	L	DI ₀	DI ₁	N ₁	...	N _m	CS	16H
-----	----	-----	----	-----	-----	---	-----------------	-----------------	----------------	-----	----------------	----	-----

图 2-7 无后续数据帧格式

有后续数据帧格式, 如图 2-8 所示:

68H	A0	...	A5	68H	A1H	L	DI ₀	DI ₁	N ₁	...	N _m	CS	16H
-----	----	-----	----	-----	-----	---	-----------------	-----------------	----------------	-----	----------------	----	-----

图 2-8 有后续数据帧格式

③从站异常应答帧

功能: 从站收到非法的数据请求或无此项数据

控制码: C=C1H

数据长度: L=01H

帧格式如图 2-9 所示:

68H	A0	...	A5	68H	C1H	01H	ERR	CS	16H
-----	----	-----	----	-----	-----	-----	-----	----	-----

图 2-9 从站异常应答帧格式

2.3 uClinux 操作系统

2.3.1 uClinux 简介

uClinux 从 Linux2.0/2.4 内核派生而来, 沿袭了主流 Linux 的绝大部分特性。

它是专门针对没有 MMU 的 CPU，并且为嵌入式系统做了许多小型化的工作。适用于没有虚拟内存或内存管理单元(MMU)的处理器。它保留了操作系统的所有特性，为硬件平台更好的运行各种程序提供了保证。在 GNU 通用公共许可证 (GNU GPL)的保证下，运行 uClinux 操作系统的用户可以使用几乎所有的 Linux API 函数，不会因为缺少 MMU 而受到影响。由于 uClinux 在标准的 Linux 基础上进行了适当的裁剪和优化，形成了一个高度优化的、代码紧凑的嵌入式 Linux，虽然它的体积很小，uClinux 仍然保留了 Linux 的大多数的优点:稳定、良好的移植性、优秀的网络功能、完备的对各种文件系统的支持、以及标准丰富的 API^[11]。下图 2-10 为 uClinux 的基本构架：

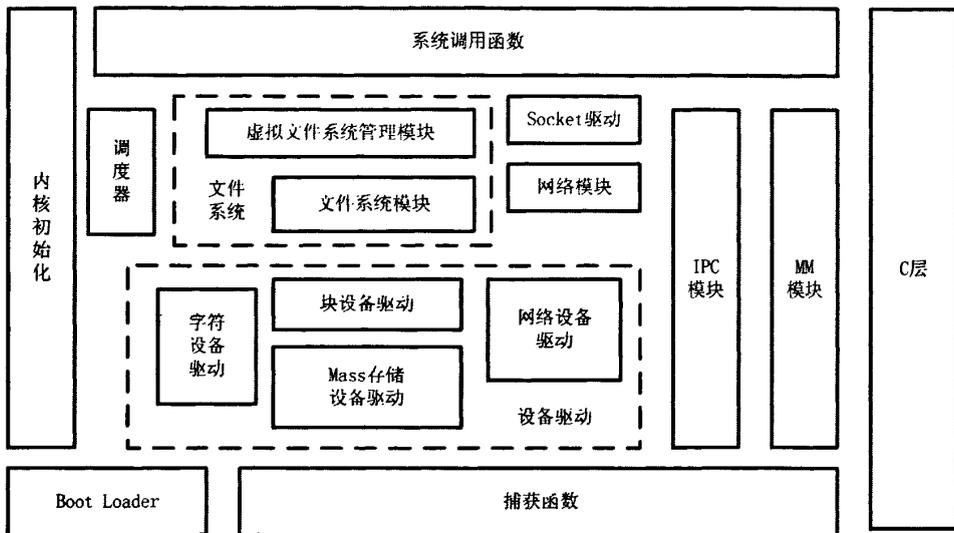


图 2-10 uClinux 的基本构架

2.3.2 uClinux 目录结构

在 uClinux 的发行包的顶层目录中包含以下及个目录。

- linux-2.0.x: uClinux 的内核，与 Linux-2.0.x 内核兼容。
- linux-2.4.x: uClinux 的内核，与 Linux-2.4.x 内核兼容。
- freeswan: 来自 www.freeswan.org 的软件。
- lib: 用户层应用程序用到的一些库，包括 libc、libm 等。
- uClibc: 一个 libc 库，但它比通常使用的 glibc 更小，更精简。
- user: uClinux 自带的 用户库的应用程序，来自开源社区，可从互联网上

免费获取。

下面几个目录包含了编译后的映像文件及输出结果。

- **config**: 一个基本的 Linux 配置脚本，用来进行默认的配置。
- **vendors**: 包含平台特定的一些编译信息和一些支持文件。这里还存放了对每一个 uClinux 支持的平台的默认文件。
- **romfs**: ROM 文件系统的目录结构。
- **images**: 包含编译好的二进制内核文件，文件系统的映像文件及一个包含文件系统和内核的映像。

其中在 Linux-2.4 目录下，包含了与 uClinux 内核相关的所有文件，其目录如下。

- **arch**: 包含硬件相关的代码，也就是 uClinux 内核的硬件抽象层。因为 uClinux 要支持许多不同的硬件，而它们的结构差别很大，所以就有一些代码屏蔽掉这些差别，让上层代码可以不受硬件差异性的影响。
- **kernel**: 包含 uClinux 内核的一些核心功能，最终用于生成内核映像文件。
- **fs**: uClinux 内核的文件系统。
- **init**: uClinux 内核的初始化代码。
- **include**: uClinux 内核的头文件。其中 `include/asm-<arch>` 是硬件相关的头文件，`include/linux` 是内核核心使用的头文件。
- **mm**: uClinux 内核的内存管理。
- **ipc**: uClinux 内核的进程间通信。
- **lib**: uClinux 内核需要使用的库。
- **net**: uClinux 内核的网络系统。其中 `net/ipv4` 就是 TCP/IP 的协议栈。
- **Documentation**: uClinux 内核的帮助文档。

2.3.3 uClinux 内存管理

uClinux 同标准 Linux 的最大区别就在于内存管理。一方面，标准 Linux 使用虚拟存储器技术，这种技术用于提供比计算机系统中实际使用的物理内存大得多的内存空间，使用者将感觉到好像程序可以使用非常大的内存空间，从而使编程人员在写程序时不用考虑计算机中的物理内存的实际容量。而 uClinux 是针对没有 MMU 的处理器设计的，不能使用处理器的虚拟内存管理技术。虚拟存储器采用基于局部性原理由存储器管理机制和一个大容量的快速硬盘存储

器支持来实现。为了支持虚拟存储管理器的管理，Linux 系统采用分页的方式来载入进程。另一方面，标准 Linux 是针对有内存管理单元的处理器设计的。在这种处理器上，虚拟地址被送到内存管理单元（MMU），把虚拟地址映射为物理地址。通过赋予每个任务不同的虚拟-物理地址转换映射，支持不同任务之间的保护。MMU 寄存器是有特权的，只能在内核状态才能访问。这就保证了一个进程只能访问自己用户空间内的地址，而不会访问和修改其它进程的空间，而对于多进程管理当处理器进行进程切换并执行一个新任务时，一个重要部分就是为新任务切换任务转换表。

由上可知，uClinux 的内存管理和 Linux 的内存管理主要区别在两个方面。Linux 采用虚拟存储系统是以占用 CPU 时间为代价的。内存管理需要地址转换表和其他一些数据结构，留给程序的内存减少了，地址转换增加了每一条指令的执行时间，而对于有额外内存操作的指令会更严重。当进程访问不在内存的页面时，系统发生处理失效，系统处理失效，并将页面加载到内存中，这是极耗时间的磁盘 I/O 操作。因此内存管理活动占用了相当一部分 CPU 时间。uClinux 的内存管理和 Linux 的内存管理相同之处在于 uClinux 采用存储器的分页管理，系统在启动时把实际存储器进行分页，在加载应用程序时程序分页加载；但是由于没有 MMU 管理，所以实际上 uClinux 采用实存储器管理策略。uClinux 系统对于内存的访问是直接的，它对地址的访问不需要经过 MMU，而是直接送到地址线上输出，所有程序中访问的地址都是实际的物理地址。操作系统对内存空间没有保护，各个进程实际上共享一个运行空间。一个进程在执行前，系统必须为进程分配足够的连续地址空间，然后全部载入主存储器的连续空间中。此外磁盘交换空间也是无法使用的，系统执行时如果缺少内存将无法通过磁盘交换来得到改善。如果从易用性这一点来说，uClinux 的内存管理是一种倒退，退回了到了 UNIX 早期或是 DOS 系统时代，开发人员不得不参与系统的内存管理，因此 uClinux 对内存的管理减少就给开发人员提出了更高的要求。虽然 uClinux 的内存管理与标准 Linux 系统相比功能相差很多，但应该说这是嵌入式设备的选择。在嵌入式设备中，由于成本等敏感因素的影响，普遍的采用不带有 MMU 的处理器，这决定了系统没有足够的硬件支持实现虚拟存储管理技术。从嵌入式设备实现的功能来看，嵌入式设备通常在某一特定的环境下运行，只要实现特定的功能，其功能相对简单，内存管理的要求完全可以由开发人员考虑^[11]。

2.3.4 uClinux 多进程管理

Linux 作为一种通用的操作系统，其最初的设计是用于桌面系统或者小型服务器。在将 Linux 用于嵌入式系统中时，由于受到嵌入式软/硬件环境的制约，需要对 Linux 内核做一些改进，使它更好地为嵌入式系统服务。在这些改进中，进程管理是相当重要的一环。

uClinux 是专门针对无 MMU 处理器的嵌入式设备的 Linux 变种。uClinux 没有 MMU，在实现多进程时 (fork 调用生成子进程) 须实现数据保护。由于 uClinux 的多进程管理是通过 vfork 来实现的，因此 fork 等于 vfork。这意味着 uClinux 系统 fork 调用完成后，或者子进程代替父进程执行 (此时父进程已经 sleep)，直到子进程调用 exit 退出；或者调用 exec 执行一个新的进程，此时将产生可执行文件的加载，即使这个进程只是父进程的拷贝，这个过程也是不可避免的。当子进程执行 exit 或 exec 后，子进程使用 wakeup 把父进程唤醒，使父进程继续往下执行。

uClinux 的这种多进程实现机制与它的内存管理紧密相连。uClinux 针对没有 MMU 的处理器开发，所以必须使用一种 flat 方式的内存管理模式。在启动新的应用程序时，系统必须为应用程序分配存储空间，并立即把应用程序加载到内存。缺少了 MMU 的内存重映射机制，uClinux 必须在可执行文件加载阶段对可执行文件进行 reloc 处理，使得程序执行时能直接使用物理内存^[11]。

2.3.5 设备驱动概述

系统调用是操作系统内核和应用程序之间的接口，而设备驱动程序则是系统内核与硬件设备之间的接口。在嵌入式系统中，设备驱动程序屏蔽了各种设备之间的细节，为用户提供统一的操作方式，应用程序可以像操作普通文件一样对硬件设备进行打开、关闭、读取或写入操作。作为系统内核的重要组成部分，Linux 设备驱动程序主要完成以下功能：

- (1) 对设备进行初始化和释放；
- (2) 将数据从内核传送给设备和从设备读取数据；
- (3) 对设备进行管理和控制；
- (4) 检测和处理设备出现的错误。

Linux 系统支持三种类型的硬件设备：字符设备、块设备和网络设备。字符设备和块设备均可通过 open、close、read、write 等操作来访问。二者之间的主

要区别在于管理方式的不同。字符设备以字节为单位进行读写，无须经过缓冲区，而块设备则必须通过一块缓冲区来访问。其次，字符设备不可随机读写，只能顺序存取。而块设备支持随机存取，可以读取任意位置上的整数倍块长的数据。网络设备既具有字符设备的特点，又具有块设备的特点。对网络设备的存取是流式的，但是其输入输出又是成块的，块的大小不固定，小到几个字节，大到数百甚至数千字节。因此，网络设备不再像字符设备和块设备驱动那样通过 `read`、`write` 等操作来访问，而是通过 `socket` 接口来进行。

设备驱动程序可以分为三个主要组成部分：

(1) 自动配置和初始化子程序。负责检测所要驱动的硬件设备是否存在和是否能正常工作。如果该设备正常，则对这个设备及其相关的、设备驱动程序需要的软件状态进行初始化。这部分驱动程序仅在初始化的时候被调用一次。

(2) 服务于 I/O 请求的子程序，又称为驱动程序得上半部分。调用这部分是由于系统调用的结果。这部分程序在执行的时候，系统仍认为是和进行调用的进程属于同一个进程，只是由用户态变成了核心态，具有进行此系统调用的用户程序的运行环境。因此可以在其中调用 `sleep()` 等与进程运行环境有关的函数。

(3) 中断服务子程序，又称为驱动程序得下半部分。在 Linux 系统中，并不是直接从中断向量表中调用设备驱动程序的中断服务子程序，而是由 Linux 系统来接收硬件中断，再由系统调用中断服务子程序。中断可以产生在任何一个进程运行的时候，因此在中断服务程序被调用的时候，不能依赖于任何进程的状态，也就不能调用任何与进程运行环境有关的函数。因为设备驱动程序一般支持同一类型的若干设备，所以一般在系统调用中断服务子程序的时候，都带有一个或多个参数，以唯一标识请求服务的设备。

Linux 的设备管理与文件系统紧密联系，用户进程是通过设备文件与具体硬件打交道。各种设备文件都以文件的形式存放在 `/dev` 目录下。在设备管理中除了设备类型外，内核还需要一对参数来确定唯一的设备。这对参数就是主设备号与次设备号。主设备号可以用来区分不同种类的设备，而次设备号用来区分同一种类型的多个设备。可以使用操作宏 `MAJOR()` 来获取主设备号，使用 `MINOR()` 来获取次设备号，使用 `MKDEV()` 来根据主次设备号合成设备号。

在系统内部，I/O 设备的存取通过一组固定的入口点来进行，这组入口点是由每个设备的驱动程序提供的。具体到 Linux 系统，设备驱动程序所提供的这组入口点由一个文件操作结构来向系统进行说明。定义于 `linux/fs.h` 文件中的

`file_operation` 结构就是这样一组入口点, 这个结构的每一个成员的名字都对应着一个系统调用。用户进程利用系统调用在对设备文件进行诸如: `read/write` 操作时, 系统调用通过设备文件的主设备号找到相应的设备驱动程序, 然后读取这个数据结构相应的函数指针, 接着把控制权交给该函数。这是 linux 的设备驱动程序工作的基本原理。因此, 编写设备驱动程序的主要工作就是编写子函数, 并填充 `file_operations` 的各个域。`file_operations` 结构如下:

```
struct file_operations
{
    int (*lseek)(struct inode *,struct file *,off_t,int);
    int (*read) (struct inode *,struct file *,char *,int);
    int (*write) (struct inode *,struct file *,const char *,int);
    int (*readdir) (struct inode *,struct file *,void *,filldir_t);
    int (*select) (struct inode *,struct file *,int,select_table);
    int (*iocrl)
    int (*mmap)
    int (*open)
    int(*release)
    int (*fsync)
    int(*fasync)
    int (*check_media_change)(kdev_t dev);
    int (*revalidate)(kdev_t dev);
};
```

这个结构中又包含另外两个重要的数据结构, 一个是 `struct_inode`, 它提供了关于特别设备文件/`dev/driver` 的信息; 另一个重要的结构是 `struct_file`, 主要用于与文件系统对应的设备驱动程序使用。当然, 其它设备驱动程序也可以使用它。它提供关于被打开的文件的信息。

2.4 配变监测终端的总体设计

2.4.1 系统功能需求

设计要求终端具有电表功能和用电管理终端的功能, 具体功能如下:

(1) 计量功能

计量正向有功电能（反向电能计入正向）、一/四象限无功电能（三象限计入一象限，二象限计入四象限），分别计量总及各费率（尖峰平谷）下的电量值，并贮存其数据。无功四象限累加组合方式： $I+IV$ ，反向无功计入正向。

能够贮存上一个抄表周期内的数据，转存分界时间为每月月末 24 时（月初零时）或其它抄表日的任意时刻（日、时）。存储 6 个月的电量。

（2）瞬时值测量

自动测量并显示电网的实时瞬时值。具体测量内容包括：总及/A/B/C 相有功功率、总及/A/B/C 相无功功率、总及/A/B/C 相功率因数、各相电压、各相电流、电网频率。

（3）脉冲输出功能

同时提供 2 路固定脉冲输出，分别是有功脉冲和无功脉冲（脉冲宽度为 80ms 左右），此脉冲的脉冲常数为铭牌标注的脉冲常数（8000imp/Kwh）。同时还提供时钟晶体振荡测试脉冲，用于测试时钟的准确度。

（4）抄表功能

通过 RS485 接口或红外接口，可以与智能抄表终端设备或红外手持抄表设备实现抄表功能。支持以下几种抄表方式：PDA 掌机抄表、手工抄表、PC 机抄表、停电抄表。

（5）用电管理终端功能

可作为用电管理终端使用，用于抄读其它电能表的数据，对数据进行统计、保存。支持多种通信协议，包括国标协议、鄂规 V1.0 协议、ABB 表协议、威胜协议、EDMI 协议、兰吉尔协议等。通讯波特率可设置为 300, 600, 1200, 2400, 4800、9600（根据规约进行设置）。此外，通过 GPRS 模块可与控制中心通信。

（6）报警功能

具有光控报警和声控报警两种功能。用户可通过光控及声控报警查询报警的事件，如失压报警、失流报警、逆相报警、超负荷报警、电压欠压报警、过载报警、过流报警、过压报警、功率反向报警、电池欠压报警以及其他故障报警（包括表内故障）等。声报警生效后，可通过按键关闭报警。当事件恢复正常后报警自动结束。

（7）负荷记录

负荷记录是指根据用户的实时用电情况，按照一定的时间间隔连续记录用户现场的实际用电参数。负荷记录时间间隔可由用户设置（1 至 60 分钟可设置），

按照设定的时间间隔对选定的内容进行滚动数据记录。

(8) 显示功能

两套轮显数据，分为自动轮显、按键轮显两种方式，通过红外可进行切换，同时支持按键以及红外遥控器的按键轮显显示，带液晶背光功能，通过按键激活背光显示，连续显示 10 秒钟后关闭背景光，进入省电模式。

终端还应具有失压记录、失流记录、过载记录、冻结记录、编程事件记录等功能。

2.4.2 系统设计方案

配变监测系统采用三层体系结构如图 2-11，第一层是服务器层（抄表主站），安装在客户服务中心，其主要作用是负责存储多功能电表的数据、实现对监测终端的远程监控、远程控制。

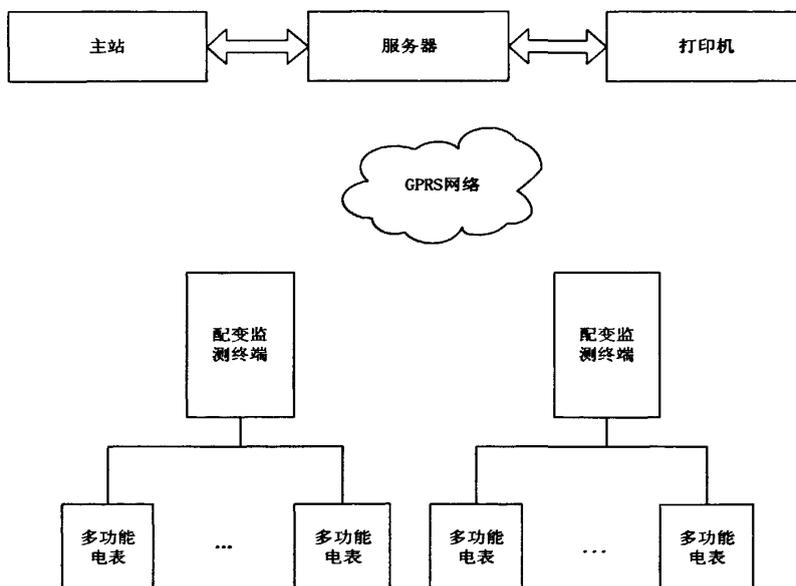


图 2-11 配变监测系统结构图

第二层是控制层，控制层的核心器件是配变监测终端，其主要功能是对多功能电表的数据采集和参数设置，并把采集到的数据通过 GPRS 网络传输到服务器层。本系统采用通信模块 MC39i，通过该通信模块，终端可直接与 GPRS 无限网络连接，而 GPRS 网络与 Internet 又可实现无缝连接。这样连接在 Internet 上的服务器就可与控制层相连，另外终端与第三层的多功能电表通过 RS-485 总

线连接。

第三层是多功能电表层，多功能电表实时抄收配电变压器的运行数据，通过 RS-485 总线向控制层传输数据。

本系统设计的配变监测终端带有电表功能，即它可作为终端抄自身内部的表，也可以抄其它外部多功能表，又可以作为一块电表独立使用。采用这种设计方案大大方便了用户的使用。下面终端硬件和软件部分的总体设计。

(1) 硬件部分设计

终端由主控板、通信模块（含天线）、显示模块、红外通信板、交采板、电源板及机壳等部件和整件组成。各模块的相互关系见图 2-12。主控板部分的设计是整个系统的核心模块，也是本文详细介绍的部分。其中，处理器是整个终端的管理、通信、运算、显示、统计的核心，采用 ARM 来实现这一部分的功能。

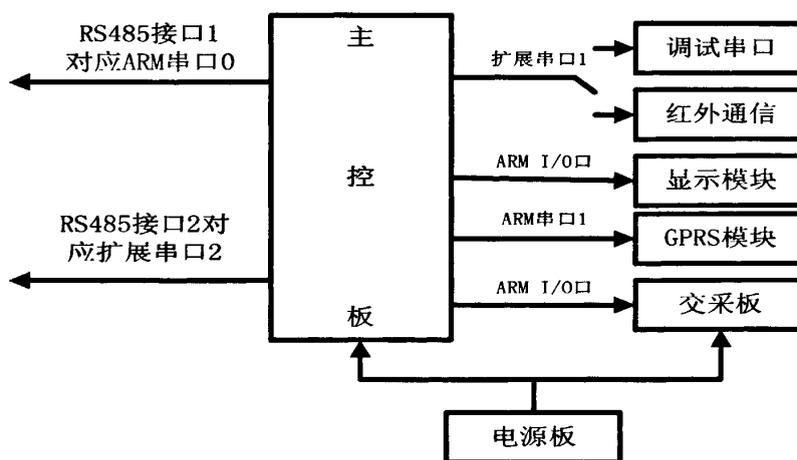


图 2-12 终端各模块关系图

(2) 软件部分设计

软件平台基于 uClinux 内核 v2.0.38, C 库 uC-libc-0.9.1.3, uC-libm-0.9.1.3, 交叉编译器 gcc-2.95.3, 系统软件框架如图 2-13, 应用程序处于程序的最顶端, 并不直接操作硬件物理层; 即便应用程序之间进行数据交换 (通信), 需要通过操作系统进行, 某一应用程序因意外原因中止运行, 也不会造成整个系统的瘫痪; 同样, 当终端应用程序需要升级时, 只需要升级相应的应用程序块而无须升级整个应用层, 大大提高了远程下载的效率。

当终端上电后,立即运行 BootLoader(也就是常说的 BIOS)程序。BootLoader 对外围器件的数据宽度、基地址及地址空间的大小进行配置,并将固化在 FLASH 中的操作系统和应用程序复制到 SDRAM 中并运行。

应用程序	通信	电表	抄表
操作系统	uCLinux 操作系统内核			
	设备驱动程序			
硬件物理层	远程通信串口	红外/本地串口	RS-485 端口

图 2-13 系统软件框架

第 3 章 配变监测终端的硬件设计

3.1 主控板原理与设计

本终端采用高性能的 32 位处理器 ARM,采用 ATMLE 公司推出的基于 ARM7TDMI 核的微处理器 AT91M40800,该处理器的核心为高性能的 32 位 RISC 体系结构,并具有高密度的 16 位指令集和极低的功耗,另外,为控制内部的大量分组寄存器适于快速中断处理,以满足实时控制的需要,主控板硬件结构图如图 3-1。



图 3-1 主控板硬件结构图

硬件实时时钟,所有根据时间进行的操作和数据时标都以此为基准。SDRAM 用来缓存可执行程序代码,系统启动后将 FLASH 里的程序代码复制到 SDRAM 里执行。FLASH 用来存放可执行程序代码。SRAM 用来存放需要保存的数据、记录及临时数据。EEPROM 用来存放终端地址、TV 变比 K_v 、TA 变比 K_i 以及电能表脉冲常数 K_p 、主站 IP 地址和端口、设备编号等参数。LCD 为字段型液晶显示器。GPRS 模块提供远程通信信道,通过 RS-232 接口和处理器

通信。红外通信接口用来和 PDA 通信，通过 RS-232 与处理器连接。本地串口供调试使用，和红外通信接口通过跳线二选一。由于一般处理器串口有限（通常为 2 个），供抄表和被抄的 RS-485 接口需要扩展，使用扩展 2 个串口的芯片来实现。CPLD 用来实现地址译码及逻辑控制。

3.1.1 AT91X40 系列微控制器

AT91X40 系列为 ATMEL 公司基于 ARM7TDMI 核的 AT91 16/32 位微控制器家族成员，其核心为高性能的 32 位 RISC 体系结构，并具有高密度的 16 位指令集和极低的功耗。此外，该系列微控制器内部的大量分组寄存器适于快速中断处理，以满足实时控制应用的要求。AT91X40 系列通过完全可编程的外部总线接口（EBI）直接连到包括 Flash 在内的各种片外存储器。8 个优先级向量中断控制器和片内外围数据控制器则显著提高了器件的实时性能。AT91X40 系列器件以 ATMEL 的高密度 CMOS 工艺制造。通过将 ARM7TDMI 处理器核与高速片内存储器以及各种外围功能模块集成于单一硅片，使得 AT91X40 系列为各种需要大计算量的嵌入式系统提供了一个灵活的、性价比高的解决方案。

AT91X40 系列的特点如下：

- 集成了 ARM7TDMI ARM Thumb 处理器内核。
- 8KB 片内 SRAM。
- 完全可编程的外部总线接口（EBI）。
- 8 优先级、可单独屏蔽的向量中断控制器。
- 32 个可编程 I/O 口线。
- 3 通道 16 位定时器/计数器。
- 两个 USART。
- 可编程的看门狗定时器。
- 先进的省电特性。
- 全静态工作。
- 1.8~3.6V 的工作电压范围。
- 100-lead 的 TQFP 封装。

系统的功能框图如图 3-2，AT91X40 系列微控制器集成了 ARM7TDMI 核、嵌入式 ICE 接口、存储器以及外围部件。系列结构有两条主要总线：先进系统总线 ASB(Advanced System Bus)和先进外围总线 APB (Advanced Peripheral

Bus)。ARM7TDMI 通过 ASB 接口来实现与片内 32 位存储器、外部总线接口 EBI 以及 AMBA 桥的互连。AMBA 桥用来驱动 APB；APB 用来访问片内外围，优化系统功耗。

AT91X40 系列产品用专用引脚实现了 ARM7TDMI 的 ICE 接口，为目标板仿真提供了一个完全的、低成本的和容易使用的调试方案。

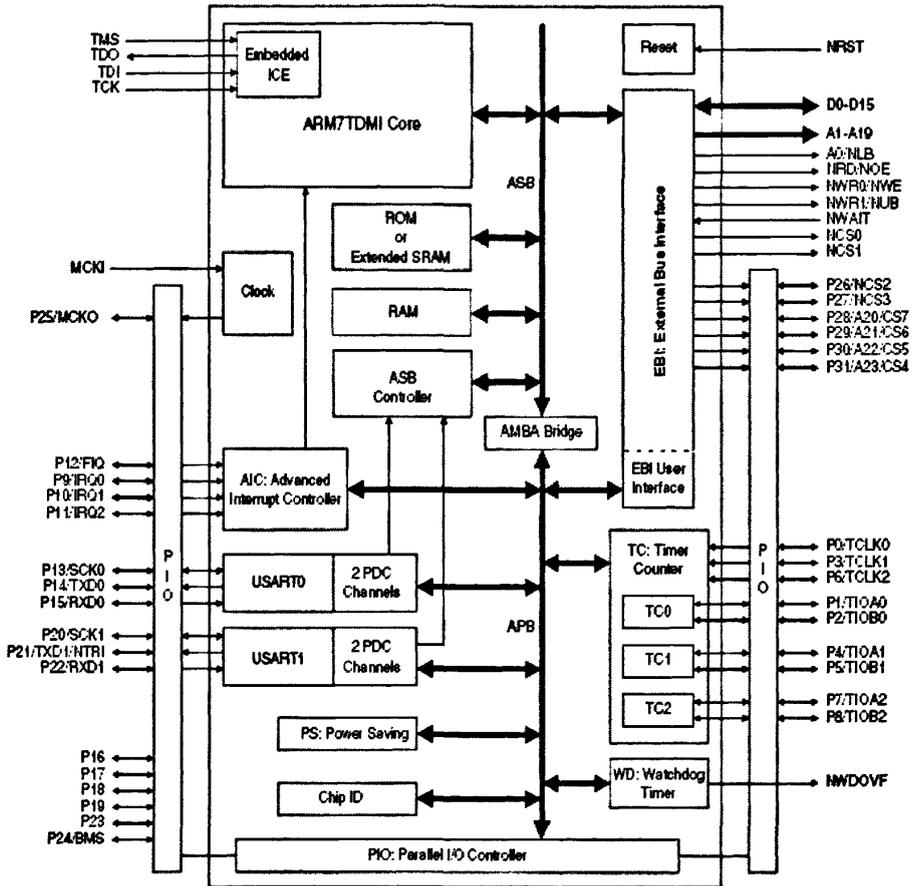


图 3-2 系统内部框图

3.1.2 存储器模块

存储器用来存放终端工作时必需的数据和程序,存储器系统性能的好坏将很大程度上影响终端的性能。本系统存储器模块主要包括一片 8Mbyte 的 Nor Flash、一片 16Mbyte 的 SDRAM、一片 1Mbyte 的 SRAM、还有一片 128byte 的 EEPROM。

(1) Nor Flash 接口设计

Flash 存储器是一种可在系统上进行电擦写、掉电后信息不丢失的存储器。它具有低功耗、大容量、擦写速度快、可整片或分扇区在系统编程（烧写）、擦除等特点。作为一种非易失性存储器，Flash 在系统中通常用于存放程序代码、常量表，以及一些在系统掉电后需要保存的用户数据等。

本系统 Nor Flash 采用 Intel 公司的 TE28F640J3C-120, 4M×16bits, 用来存储应用程序和用户数据。它是 56 引脚的 TSOP 封装器件, 3.3V 工作电压。具体接口连接如图 3-3。

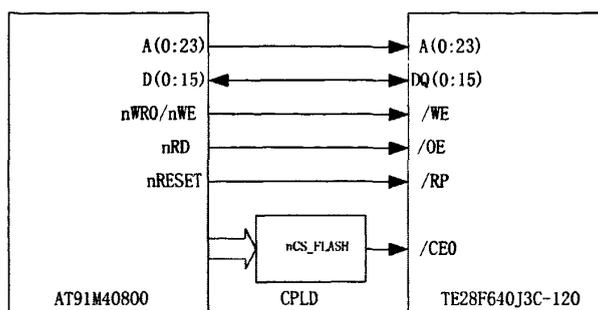


图 3-3 Norflash 接口连接框图

(2) SDRAM 接口设计

与 Flash 存储器相比，SDRAM 不具有掉电保持数据的特性，但由于其存取速度大大高于 Flash 存储器，且具有读/写的属性，因此，SDRAM 在系统中主要用做程序的运行空间、数据及堆栈区。当系统启动时，CPU 首先从复位地址 0x0 处读取启动代码，在完成系统的初始化后，程序代码一般应调入 SDRAM 中运行，以提高系统运行的速度。同时，系统及用户堆栈、运行数据也都放在 SDRAM 中。

本系统采用 Micron 公司的 MT48LC8M16 SDRAM, 4Banks×2M×16bits, 它是一款 54 引脚的 TSOP 封装器件, 3.3V 工作电压。因为 AT91M40800 处理器本身不具有 SDRAM 控制器, 因此需要使 CPLD 做为控制器, 具体接口连接框图如图 3-4。

(3) SRAM 接口设计

SRAM 为静态 RAM 存储器，具有极高的读写速度, 在本系统中用 SRAM 来存放需要保存的数据、记录及临时数据。采用美国 ISSI 公司的高速 SRAM 器件 IS62WV51216BLL-55TI, 采用 CMOS 技术。IS62WV51216BLL-55TI 存储容量

为 1Mbyte,44 引脚 TSOP 封装,具有 16 位数据线,19 位地址线,工作电压为 3.3V,具体接口连接框图如图 3-5。

(4) EPPOM 接口设计

EPPOM 属于一种非易失性永久存储器,即使在电源切断的情况下,其存储数据也不丢失,也永久保存。由于 EPPOM 是 CMOS 或 NMOS 材料制成的,它通过对作为每个存储单元的场效应管的栅极上加进正向电压充电和加负向电压放电的方便,十分方便地进行信息的擦写记录,并且断电后,由于栅极的高绝缘性、电荷不会自行消失,记录的信息长期保存。

本系统采用 EPPROM 用来存放终端地址、TV 变比 K_v 、TA 变比 K_i 以及电能表脉冲常数 K_p 、主站 IP 地址和端口、设备编号等参数。系统采用 SST 公司的 SST39VF010-70-4C-NHE,32 引脚 TSOP 封装芯片,工作电压为 3.3V,存储容量为 1Mbyte,8 条地址线,17 条数据线,具体接口连接框图如图 3-6。

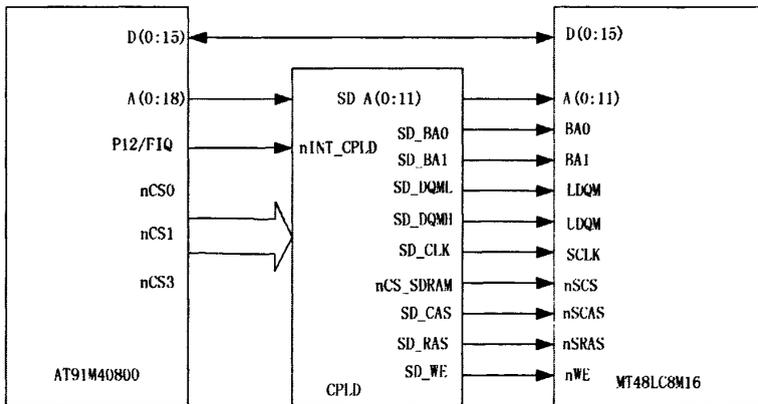


图 3-4 SDRAM 接口连接框图

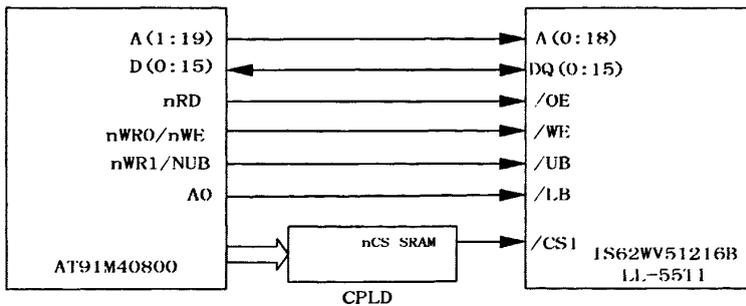


图 3-5 SRAM 接口连接框图

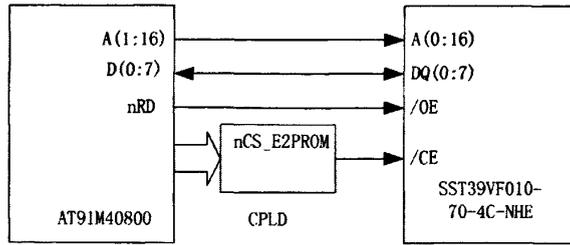


图 3-6 EEPROM 接口连接框图

3.1.3 以太网模块

AT91M40800 本身没有以太网控制器模块，所以需要有一个以太网控制芯片。这里我们采用 RTL8019AS,它是一种高度集成的以太网控制芯片，是 8/16 位 ISA 总线的网卡，集成了 MAC 和物理层的功能，能简单地实现 Plug and Play 并兼容 NE2000、掉电等特性。在全双工模式下，如果是连接到一个同样是全双工的交换机或集线器，就可实现同时接收和发送。这个特性虽然不能把传输速率从 10Mbps 提高到 20Mbps，但是在执行以太网 CSMA/CD 协议时，可以避免更多的冲突发生。而 Microsoft's Plug and Play 功能就可以为用户减轻对资源配置的烦恼(如 IRQ、I/Oaddress 等)。在一些特殊的场合，为了和一些不支持 Microsoft's Plug and Play 的器件兼容，RTL8019AS 还可以选择跳线模式或非跳线模式，其内部结构图如图 3-7。

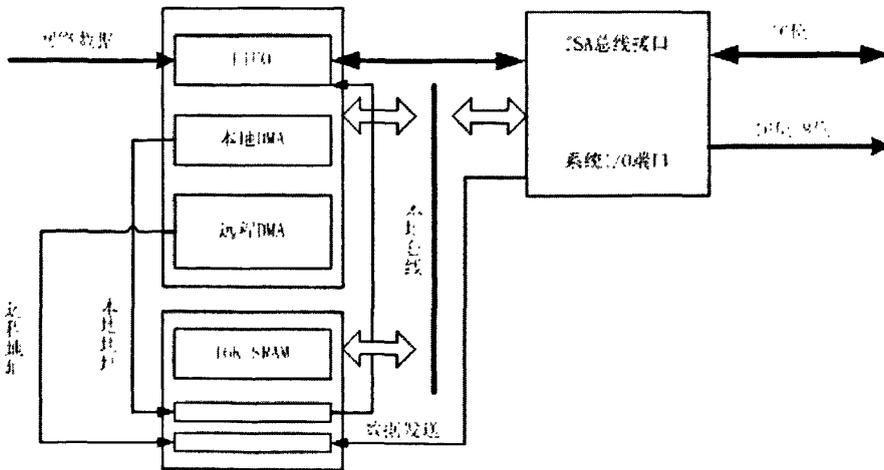


图 3-7 RTL8019AS 内部原理框图

RTL8019AS 芯片内部集成了远程 DMA 接口、本地 DMA 接口、MAC(介质访问控制)逻辑、数据编码解码逻辑和其他端口。MAC 逻辑完成以下功能:当 CPU 向网上发送数据时,先将一帧数据通过远程 DMA 通道送到 RTL8019AS 的发送缓存区,然后发出传送命令;当 RTL8019AS 完成了上一帧的发送后,再开始此帧的发送。RTL8019AS 接收到的数据通过 MAC 比较和 CRC 校验后,由 FIFO 存到接收缓冲区。收满一帧后,以中断或寄存器标志的方式通知主处理器, FIFO 逻辑对收发数据作 16 字节的缓冲,以减少对本地 DMA 请求的频率。

内部有两块 RAM 区,一块 16K 字节,地址为 0x4000~ 0x7FFF;另一块 32 字节,地址为 0x0000~0x001F。RAM 按页存储,每 256 字节为一页,一般将 RAM 的前 12 页(即 0x4000~0x4BFF)存储区作为发送缓冲区,后 52 页(即 0x4C00~0x7FFF)存储区作为接收缓冲区,第 0 页叫 PROM 页,只有 32 字节,地址为 0x0000~0x001F,用于存储以太网物理地址。要接收和发送数据包就必须通过 DMA 读写 RTL8019AS 内部的 16KBRAM,它实际上是双端口的 RAM,是指有两套总线连接到该 RAM,一套总线是 RTL8019AS 读或写该 RAM,即本地 DMA;另一套总线是 CPU 读或写该 RAM,即远程 DMA。

RTL8019AS 具有 32 个输入输出地址,地址偏移量为 00H1FH,其中 00H~0FH 共 16 个地址,为寄存器地址。寄存器分为 4 页:PACE0、PACE1、PACE2、PACE3,由 RTL8019As 的命令寄存器(CR, Common Dreiser)中的 PS0 和 PS1 位来决定要访问的页,但与 NE2000 兼容的寄存器只有前 3 页,PACE3 是 RTL8019As 自己定义的,对于其他兼容 NE2000 的芯片(如 DM9008)无效。远程 DMA 地址包括 IOH~17H,都可以用来做远程 DMA 端口,只要用其中的一个就可以了。复位端口包括 18H~1FH 共 8 个地址,功能一样,用于 RTL8019AS 复位。AT91M40800 与 RTL8019AS 的接口图如图 3-8。

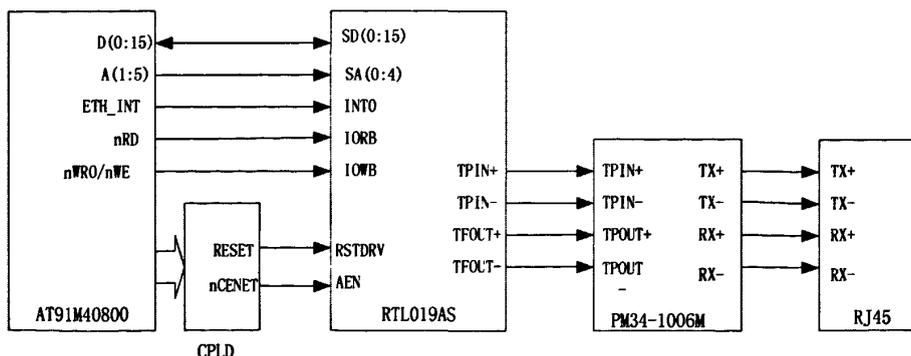


图 3-8 AT91M40800 与 RTL8019AS 接口连接框图

3.1.4 串口扩展

AT91M40800 集成了两个完全相同的全双工通用同步/异步收发器(USART)。USART 接到 APB 并与外围数据控制器 PDC 连接。

终端作为电表使用时提供一个 RS-485 接口与用电管理终端或其它具有 RS485 接口的设备相连, 获取电表数据或对电表进行设置。该 RS-485 接口由的 AT91M40800 串口 0 通过 TTL/RS-485 变换而成,RS-485 接口可通过红外切换为抄表口, 用来抄读其它电表的数据。AT91M40800 的串口 1 通过 TTL/RS-232 变换后直接与 GPRS 通信模块 (MC39i) 连接。

此外, 通过串口扩展芯片 16ST2550 扩展了两个串口。一个经 TTL/RS-485 转换后用来与无功补偿控制器连接 (终端为 RS-485 的从); 一个用来作红外通信口或经 TTL/RS-232 后作本地串口使用, 通过自锁开关切换。串口扩展原理框图如图 3-9。

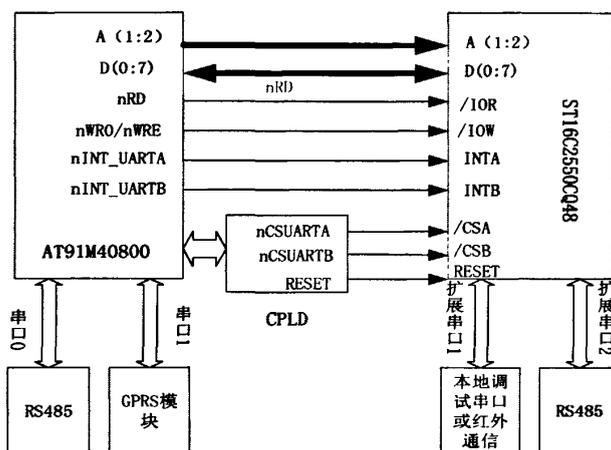


图 3-9 串口扩展原理框图

3.1.5 通信接口

配变监测终端与外部的数据通信主要包括三部分, 一终端通过 RS485 总线与电表进行通信, 读取电表的电能计量信息; 二通过红外通信接口与手持的抄表终端进行通信; 三终端可以通过 GPRS 网络与主站进行远程通信。

(1) 485 通讯接口

RS485 因硬件设计简单、控制方便、成本低廉等优点广泛应用。RS485 是一

种串行总线接口标准，它以差分平衡方式传输信号，具有很强的抗共模干扰能力，其最大传输距离可达 4000 英尺（约 1219m），并且 RS-485 接口芯片在使用、焊接或设备的运输途中都有可能受到静电的冲击而损坏。终端在传输线架设于户外的使用场合，接口芯片乃至整个系统还有可能遭致雷电的袭击。选用防静电或抗雷击的芯片可有效避免此类损失，本系统采用 TI 的一款芯片 SN75LBC184，它不但能抗雷电的冲击而且能承受高达 8kV 的静电放电冲击。硬件设计需要注意以下几点，具体电路图设计如图 3-10。

①光电隔离。在工业控制领域，由于现场情况十分复杂，各个节点之存在很高的共模电压。虽然 RS-485 接口采用的是差分传输方式，具有一定的抗共模干扰的能力，但当共模电压超过 RS-485 接收器的极限接收电压，即大于+12V 或小于-7V 时，接收器就无法正常工作了，严重时候甚至会烧毁芯片和仪器设备，因此系统采用光隔 TLP181-1-GB。

②总线隔离。RS-485 总线为并接式二线制接口，一旦有一只芯片故障就可能将总线“拉死”，因此对其二线口 VA、VB 与总线之间应加以隔离。本系统采用 TVS 二极管，并在末端接保险丝，以消除线路浪涌干扰。

③配置上拉电阻。异步通信数据以字节的方式传送，在每一个字节传送之前，先要通过一个低电平起始位实现握手。为防止干扰信号误触发接收器产生负跳变，使接收端进入接收状态，RXD0 端接 2K 上拉电阻，TXD0 接 5.1K 上拉电阻。

④总线匹配。采用二极管进行匹配，虽然为实现真正的匹配，但是利用二极管的钳位作用，迅速削弱反射信号达到改善信号质量的目的，节能效果显著。

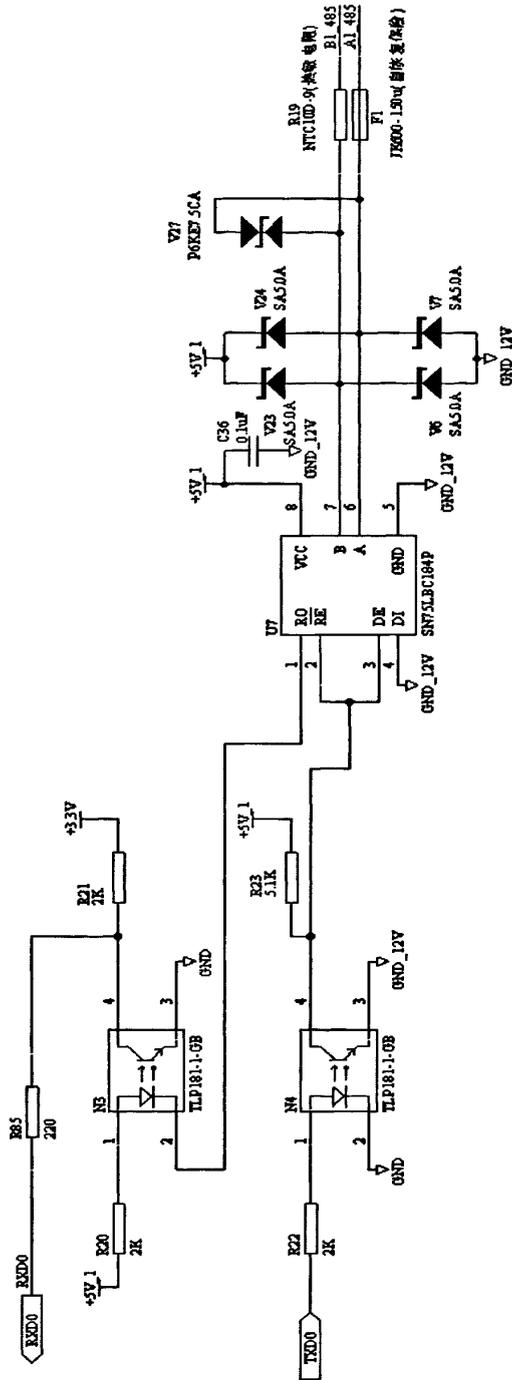


图 3-10 485 接口电路设计

(2) 红外通讯接口

红外通信的基本原理是发送端将基带二进制信号调制为一系列的脉冲信号，再通过红外发射管发射红外信号。串行红外传输采用特定的脉冲编码标准，这种标准与 RS232 串行传输标准不同。若两设备之间进行串行红外通讯，就需要进行 RS232 编码和 IrDA 编码之间的转换。红外通讯接口由红外收发器和红外编码解码器构成。

红外收发器包括发送器和接收器两部分。发送器将从 I/O 或 ENDEC 接收来的位调制后的脉冲转换为红外脉冲发出。接收器检测到红外光脉冲，并将其转换为 TTL 或 CMOS 电脉冲。红外通讯系统结构如图所示 3-11。系统采用现以 Agilent 公司的 HSDL7001 和 HSDL3201 芯片。

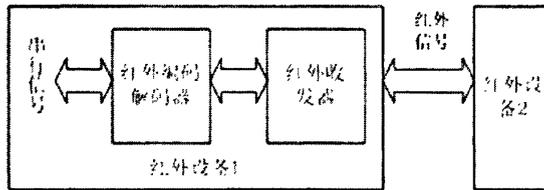


图 3-11 红外通信结构图

(3) GPRS 通讯接口

本终端采用 GPRS 通讯方式与主站进行通信，实现配变监测终端的远程控制与数据通信。系统采用 Siemens MC39i 通讯模块，集成完整的射频电路和 GSM 的基带处理器，为语音传输、短信（SMS）和数据业务(GPRS)提供无线接口，工作电压为 3.3V~4.8V。AT91M40800 通过串口 1 控制 GPRS 模块，方便、快捷地完成串行数据和 IP 数据包之间的转换，实现远程数据收发。

3.1.6 时钟保持电路

为了保持在系统掉电或者复位的情况下时钟不丢失，要设计一个时钟保持电路。DS3231 是 Maxim / Dallas 公司于 2005 年推出的一款低成本、超高精度 I2C 实时时钟芯片，它在单芯片封装内集成了温补振荡器、实时时钟和一个 32.768kHz 的晶体，不仅能够在 -40℃ ~+85℃ 温度范围内提供优于 ±2 分钟 / 年的计时精度，而且可以让用户省去制造过程中的晶体安装和在线校准工序。DS3231 的寄存器能保存秒、分、时、星期、日期、月、年和闹钟设置等信息少于 31 天的月份，可自动调整月末日期，包括闰年补偿。时钟的工作格式为 24 小

时或带 AM / PM 指示的 12 小时格式，提供两个可编程日历闹钟和一路可编程方波输出，地址与数据通过 I2C 双向串行总线传输。

DS3231 采用 I2C 总线与系统微处理器进行通信，因为 AT91M40800 未提供专门的 I2C 接口，因此将两个通用的 I/O 口线分别用作数据线和时钟线，并通过软件来模拟 I2C 时序，以实现对其内部寄存器的访问。由于 DS3231 的 SDA 和 SCL 均为漏极开路的引脚，所以必须外接上拉电阻。VBAT 引脚与地之间接后备电池，在主电源断电时自动切换到电池供电。

3.1.7 显示模块设计

在配电自动化系统中，许多场合用户需要在现场了解变压器实时情况，考虑到这种情况，在终端的基本功能上扩展了液晶显示这一功能，为用户可以提供极大的方便，特别有利于现场的维护和检测。根据系统需求，要求既可以显示汉字有可以显示数码管。由于 AT91M40800 处理器内部没有集成 LCD 控制器，因而需要配备专用的显示控制器，才能实现 LCD 显示。本系统选用了 HOLTEK 公司的显示控制芯片 HT1622 用于控制本系统中 LCD 显示。

HT1622 是一种外围器件，专用于扩展 I/O 型微控制器的显示能力。该器件的最大显示段为 256 patterns (32×8)。它还支持串行接口、蜂鸣器声音、看门狗或时基定时器功能。HT1622 是一个存储器映射 (memory mapping) 和多功能 LCD 控制器。HT1622 的软件配置特性使得它适合于多 LCD 应用，包括 LCD 模块和显示子系统。AT91M40800 和 HT1622 之间的接口设计如图 3-12。

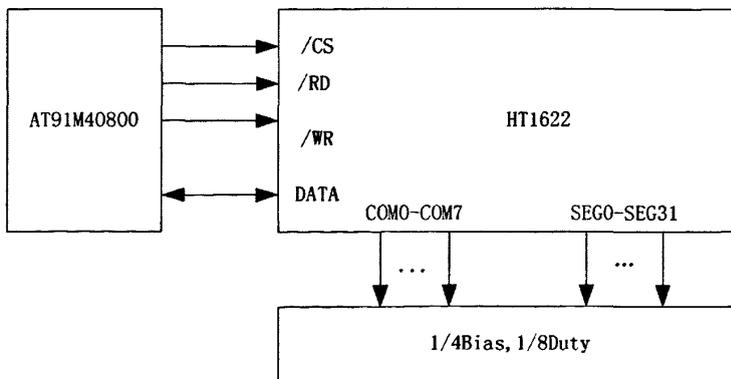


图 3-12 AT91M40800 与 HT1622 连接框图

HT1622 内部有 64×4 位显示 RAM，直接映射到 LCD 显示器的每一段。显

示 RAM 由 64 个地址连续的单元组成,从地址为 0 的单元到地址为 63 的单元,分别对应段电极输出 SEG0 至 SEG31,其中每一 RAM 单元又分为 8 位,从低位至高位分别对应背电极输出 COM0 至 COM7,也就是说 64×4 位 RAM 中的每一位对应一个 LCD 显示段,因此 HT1622 可驱动多达 256 段的 LCD 显示器。显示 RAM 中的内容可直接映射至 LCD 驱动单元。RAM 中的数据可由 READ、WRITE 和 READ-MODIFY-WRITE 命令进行访问。

3.1.8 JTAG 接口设计

JTAG(Joint Test Action Group)是一种国际标准测试协议,主要用于芯片内部测试以及对系统进行仿真、调试。JTAG 技术是一种嵌入式调试技术,它在芯片内部封装了专门的测试电路 TAP(Test Access Port, 测试访问口),通过专用的 JTAG 测试工具对内部节点进行测试。目前大多数比较复杂的器件都支持 JTAG 协议,如 ARM、DSP、FPGA 等。标准的 JTAG 接口有 20 针和 14 针两种,其中重要的信号线主要有以下 4 个:TMS、TCK、TDI、TDO,分别为测试模式选择、测试时钟、测试数据输入和测试数据输出信号线。

AT91M40800 支持 JTAG 调试功能,并且提供了 TMS、TCK、TDI、TDO,等调试信号线,因此 JTAG 调试接口的扩展比较容易,只需要将相应的引脚连接到标准 20 针 JTAG 接口上即可。

3.2 交采模块的设计

3.2.1 ATT7022B 功能简介及内部结构图

ATT7022B 是一颗精度高且功能强的多功能防窃电基波谐波三相电能专用计量芯片,它集成了七路二阶 $\sigma\text{-}\Delta$ ADC,其中三路用于三相电压采样,三路用于电流采样,还有一路可用于零线电流或其他防窃电参数的采样,输出采样数据和有效值,使用十分方便,该芯片适合于三相三线和三相四线应用。

该芯片还集成了参考电压电路以及所有包括基波、谐波和全波的各项电参数测量的数字信号处理电路,能够测量各相及合相包括基波、谐波和全波有功功率、无功功率、视在功率、有功能量以及无功能量,同时还能测量频率、各相电流及电压有效值、功率因数、相角等参数,提供两种视在电能(PQS、RMS),充分满足三相多功能电能表以及基波谐波电能表制作的要求。芯片主要特性如下:^[27]

- 高精度,在输入动态工作范围(1000: 1)内,非线性测量误差小于 0.1%;
- 有功测量满足 0.2 S、0.5 S,支持 IEC 62053-22,GB/T17883—1998;
- 无功测量满足 2 级、3 级持 IEC 62053,GB/T 17882—1999;
- 内置温度测量传感器;
- 可测量到 21 次以上谐波的有功和无功功率;
- 支持增益和相位补偿,小电流非线性补偿;
- 具有 SPI 接口,方便与外部 MCU 通讯。

ATT7022B 的内部结构框图如图 3-13,可大致分为 A/D 模数转换模块、数字信号处理模块 (DSP)、SPI 通信接口模块、脉冲生成模块等几部分,如图所示。图中 V1P/V1N、V3P/V3N、V5P/V5N 为三相电的电流信道正、负模拟输入引脚, V2P/V2N、V4P/V4N、V6P/V6N 为三相电的电压信道的正、负模拟输入引脚;输入的模拟电信号分别经过电流 ADC (模拟数字转换器)和电压 ADC;转换后的数字信号再经过数字信号处理模块 (DSP) 转换成需要的电能数据,同时生成电脉冲,用来进行校表;最后通过 SPI 通信口与外部进行数据通信。

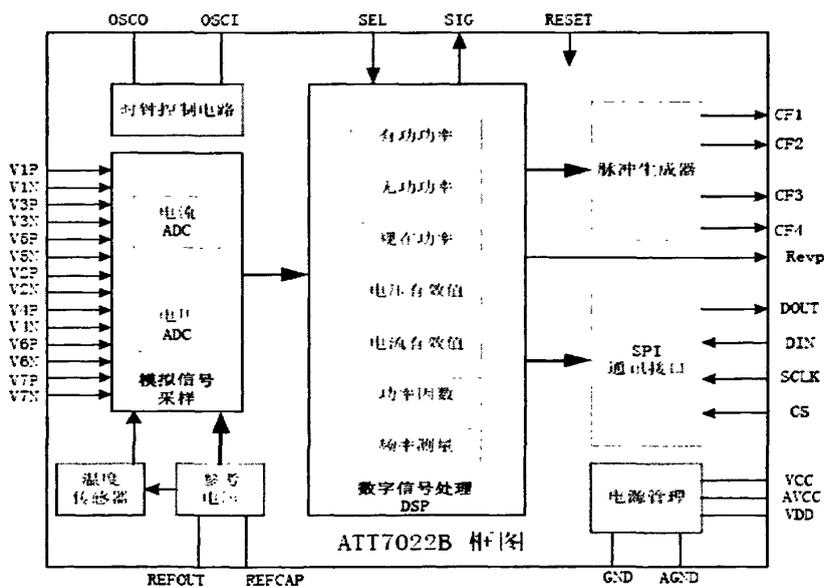


图 3-13 ATT7022B 内部结构框图

3.2.2 计量模块

(1) 数字信号处理框图

数字信号处理框图如图 3-14。

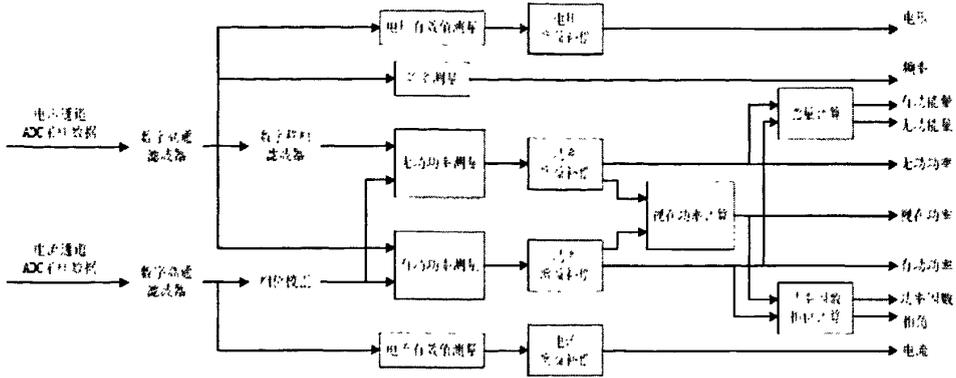


图 3-14 数字信号处理框图

(2) 计量原理简介:

1) 电路基本参数的测量原理^[27]

$$\text{电压有效值: } U_{\text{rms}} = \sqrt{\frac{1}{T} \int_0^T u^2(t) dt} \quad (3-1)$$

$$\text{电流有效值: } I_{\text{rms}} = \sqrt{\frac{1}{T} \int_0^T i^2(t) dt} \quad (3-2)$$

$$\text{功率因数: } PF = \text{sign}(Q) \times \frac{\text{abs}(P)}{\text{abs}(S)} \quad (3-3)$$

三相三线合相功率:

$$P_3 = P_{AB} I_A + P_{CB} I_C \quad (3-4)$$

$$Q_3 = U_{AB} I_A < 90^\circ + U_{CB} I_C < 90^\circ \quad (3-5)$$

$$S_3 = \sqrt{P_3^2 + Q_3^2} \quad (3-6)$$

单相有功平均功率:

$$P = \frac{1}{T} \int_0^T u(t) i(t) dt \quad (3-7)$$

单项无功平均功率:

$$U_q(t) = U(t) < 90^\circ \quad (3-8)$$

$$Q = \frac{1}{T} \int_0^T u_q(t) i(t) dt \quad (3-9)$$

有功能量: $E_p = \int p(t) dt$ (3-10)

无功能量: $E_p = \int q(t) dt$ (3-11)

2) 信号处理模块说明^[27]

①数字高通滤波器

主要是用于去除电流、电压采样数据中的直流分量。

②数字移相滤波器

主要完成对电压 90 度的信号处理。在保证信号幅频相应不衰减的前提下，能够对 30-1500Hz 的采样信号进行移相 90 度的处理。因此无功计量的带宽限制在 1500Hz 以内。下图 3-15 是移相滤波器的幅相特性。

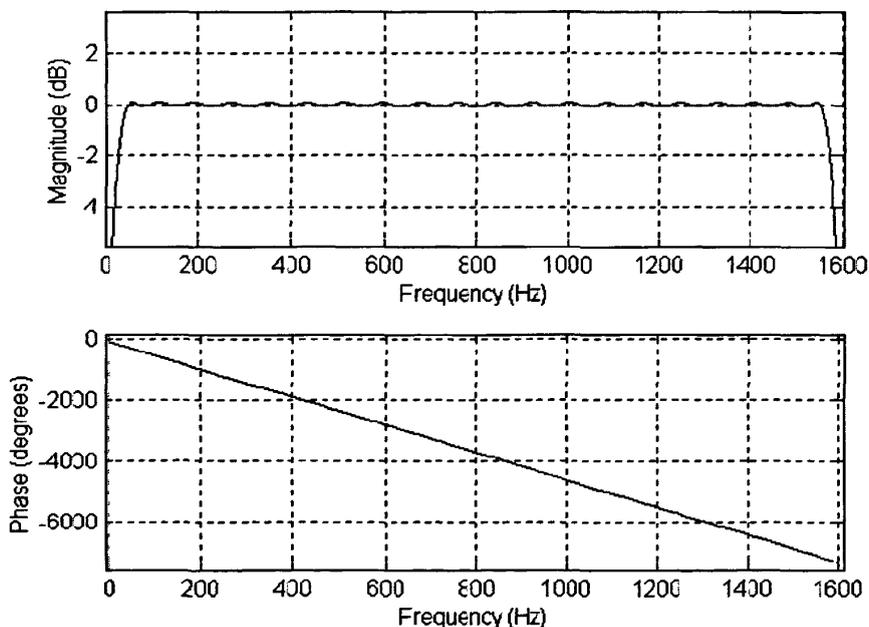


图 3-15 移相滤波器的幅相特性

③有功功率计量

各相的有功功率是通过对去直流分量后的直流、电压信号进行乘法、加法、数字滤波等一系列数字信号处理后得到的。

前端 sigma-delta ADC 采用过采样技术，可充分保证电流、电压采样速率，

根据采样定理可知电流、电压采样数据中包含高达 21 次的谐波信息，有功功率也至少包含 21 次谐波信息。

④无功功率计量

无功功率计量算法与有功类似，只是电压信号采用移相 90 度之后的。测量带宽主要受到数字移相滤波器的带宽限制，所以无功功率的测量也可高达 21 次谐波。

⑤四象限功率测量

有功功率既有输入也有输出的。输入有功标识为+，表示用户消耗的电能，而输出有功标识为-，表示用户发出的电能。

无功功率有容性无功和感性无功，感性无功标识为+，而容性无功标识为-。ATT7022B 可以提供图 3-16 的有功、无功参数，能够方便实现四象限无功电能测量。

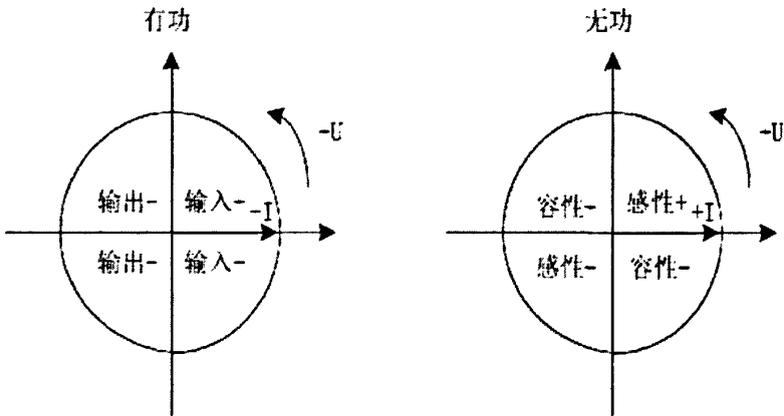


图 3-16 四象限无功测量

⑥视在功率、功率因数、相角测量

基于上述测量的有功功率和无功功率，通过开方、除法等运算就可得到这些参数。

⑦有效值测量

通过对电流、电压采样值进行平方、开方以及数字滤波等一系列运算得到。误差小于 0.1%。

⑧能量计算

将功率信号对时间进行积分就可以得到能量。

(3) 脉冲输出

通过高频脉冲输出寄存器可以设定脉冲输出频率，可用于仪表的校验。

3.2.3 校表原理

ATT7022B 支持全数字校表，即软件校表。经过校正的仪表，有功精度可高达 0.2s，无功精度 2 级。功率校正方面主要是对比差以及角差进行修正。比差校正主要是对互感器的比差进行修正。互感器比差的非线性在高精度计量时也是不可忽视的。ATT7022B 可以将比差进行分段补偿。ATT7022B 提供寄存器 Iregchg，可以根据电流大小设置补偿分段区域。如下图 3-17 所示。

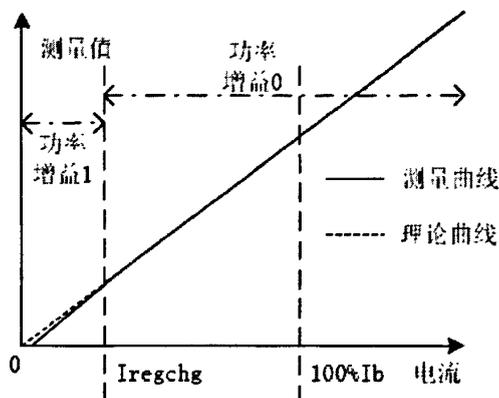


图 3-17 比差校正

通常将 Iregchg 设置在 35%Ib 处，即 $Iregchg=35\%Ib$ 。当 Iregchg 设定之后，需要对这两个区域分别进行比差校正。如果将 Iregchg 设为 0，则说明在整个电流区域内使用一次比差补偿，这时只需根据误差对功率增益 0 进行设置即可。同比差修正类似，角差修正也采用分段修正。ATT7022B 最多可以提供五个角差补偿区域，如下图 3-18 所示。

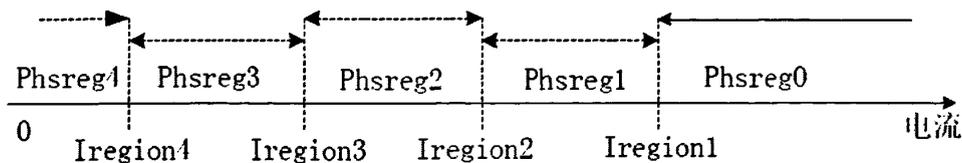


图 3-18 ATT7022B 角差校正图

当然，实际应用中角差的补偿并不需要将电流区域分得这么多段，在一级表或者 0.5 级表应用时只需要分成两段补偿即可达到要求，在设计 0.2 级或者

更高精度的电能表时才可能需要用到多于 2 段的补偿方法。完成比差修正之后，再做角差的修正，角差修正在 0.5L 时完成。ATT7022B 还可以通过寄存器对电流、电压有效值进行修正。ATT7022B 脉冲输出频率可以通过 HFConst 寄存器进行设定。ATT7022B 启动电流通过寄存器 Istartup 进行设置^[27]。

3.2.4 ATT7022B 与 CPU 的连接

ATT7022B 采用 44 脚 QFP 封装，外围硬件电路主要包括电源、电压及电流模拟输入、脉冲输出及 SPI 通讯接口等电路。

(1) 模拟输入电路

ATT7022B 内部集成了 7 路 16 位 A/D 转换器，电流通道有效值在 2mV 至 1V 的范围内线性误差小于 0.5%；电压通道有效值在 10mV 至 1V 的范围内线性误差小于 0.5%；电压取值在 0.2V 到 0.6V，电流取值在 2mV 至 1V，电压、电流采样输入电路如 3-19。

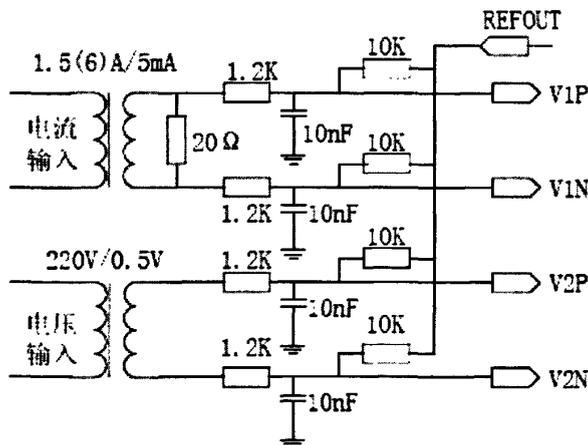


图 3-19 电压、电流采样输入电路

(2) 脉冲输出电路

电能脉冲输出口 CF1、CF2、CF3、CF4 上所串联电阻靠近管脚，如引线较长，可考虑在脉冲输出口管脚处接去耦电容。

(3) SPI 通讯接口电路

1) SPI 通讯连线应尽可能短，并且周围用地线包围起来，避免 SPI 传输信号可能受到干扰。可以在 SPI 信号线上串联一个 10Ω 电阻并在信号输入端加一个去耦电容，这个电阻靠近 IC 的输入端并与所接去耦电容结合起来可构成一个

低通滤波器，从而可以消除信号的高频干扰，CS、SCLK、DIN 所串电阻和所并的电容要尽量靠近芯片，DOUT 所串电阻和所并电容靠近 CPU，SPI 连接图如图 3-20。

2) 在上电或者 ATT7022B 受干扰复位，必须由外部 CPU 通过 SPI 口对校表数据进行更新，以保证计量的准确性。在 CPU 的 SIG 输入口处外接 0.01 μ F 的去耦电容，增强抗干扰能力。

3) 为了在上电和 CPU 复位后，ATT7022B 能与 CPU 同步的工作，ATT7022B 的 RESET 信号需要有 CPU 控制，复位过程为 RESET 信号保持大约 20 μ s 低电平，芯片复位，此时 SIG 输出高电平，然后 CPU 将 RESET 信号拉高，大约经 500 μ s 左右，ATT7022B 完成初始化，SIG 输出低电平信号此后才能进行 SPI 操作。在 ATT7022B 的 RESET 端口处接 0.1 μ F 的去耦电容，增强其抗干扰能力。

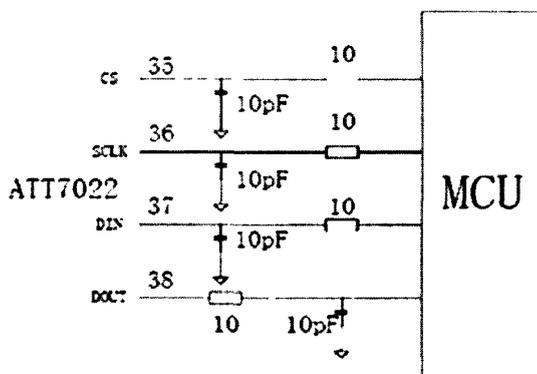


图 3-20 SPI 连线图

第 4 章 配变监测终端软件设计

4.1 系统开发环境的建立

4.1.1 交叉开发环境的建立

构建嵌入式 uClinux 系统软件平台的任务是通过交叉开发方式向嵌入式硬件平台移植 uClinux 操作系统。交叉开发是嵌入式开发过程中的一项重要技术，它是首先在宿主机上编写程序，然后通过交叉编译、链接生成针对目标机的可执行代码，最后通过各种通信手段将其下载到目标机上运行和调试的开发方式。目前，嵌入式系统多数采用交叉开发方式来进行，其原因主要在于多数嵌入式目标系统不能提供足够的软硬件资源供编译过程使用，只能将编译工程转移到高性能的宿主机上进行。在本系统中，使用由 ARM Linux 提供的交叉编译工具 gcc-2.95.3；宿主机操作系统采用 Redhat9.0。

通常，一个基本的嵌入式 Linux 系统可以分为 4 个部分：

(1) 引导加载程序。包括固化在固件(Firmware)中的启动代码和 BootLoader 两部分。引导程序即 BootLoader 是在系统上电后执行，负责系统的初始化并加载真正的应用程序或操作系统的软件。本课题采用 U-Boot 作为 BootLoader 程序；

(2) Linux 内核。定制适合本嵌入式平台的内核。本系统选用 uClinux2.0.38 内核，在其基础上定制。

(3) 文件系统。包括根文件系统和建立于 Flash 内存设备之上文件系统。根文件系统是内核启动后加载的第一个文件系统，所有其它的文件系统只能挂载在根文件系统的目录里。所以根文件系统必须包含 Linux 内核正常运行所必须的配置文件和应用程序的库以及常用命令，同时还包括部分驱动程序等。

(4) 用户应用程序。特定于用户的应用程序。

4.1.2 Bootloader 程序设计

一个嵌入式 Linux 系统从软件的角度看通常可以分为四个层次：引导加载程序、Linux 内核、文件系统、用户应用程序。

简单地说，Bootloader 就是在操作系统内核运行前运行地一段小程序。通过

这段小程序，我们可以初始化必要的硬件设备，创建内核需要的一些信息并将这些信息通过相关机制传递给内核，从而将系统的软硬件环境带到一个合适的状态，最终调用操作系统内核，真正起到引导和加载内核的作用。

Bootloader 的启动过程可以是单阶段的，也可以是多阶段的。通常多阶段的 bootloader 能提供更为复杂的功能，以及更好的可移植性。从固态存储设备上启动的 bootloader 大多数是二阶段的启动过程，也即启动过程可以分为 stage 1 和 stage 2 两部分。

在本系统，我们将 bootloader 分为两个阶段：阶段 1 和阶段 2。

阶段一的代码主要用汇编语言，主要进行与 CPU 核以及存储设备密切相关的处理工作，进行必要的初始化工作，使一些依赖 CPU 体系结构的代码，一般随具体的目标系统和开发系统有所区别，通常包括以下部分：

- 定义入口点；
- 设置中断/异常向量
- 初始化存储系统（包括地址重映射）
- 初始化堆栈指针寄存器
- 初始化中断中用到的变量
- 开中断
- 必要时改变处理器的模式
- 必要时改变处理器的状态
- 初始化 C 程序用到的存储区
- 进入 C 程序

(1) 在 ARM 汇编语言源文件中，开始必须通过 AREA 伪指令标记并且设置它的属性。属性被放置在名字之后，通过逗号分割。以下代码示例定义一个名为 RESET 的只读代码段。一个可执行映像必须有且只有一个入口指针。被放置在 ROM 中的可执行映像通常

在 0x00 放置一个入口指针。在初始化代码中使用汇编伪指令定义一个入口指针。

```
AREA reset, CODE, READONLY           ;区域定义
ENTRY                               ;定义入口指针
```

(2) 设置中断向量表

ARM7 中断向量表设置在从 0 地址开始，连续 8x4 字节的空间，分别是复位、未定义指令错误、软件中断、预取指令错误、数据存取错误、IRQ、FIQ 和

一个保留的中断向量。当把启动代码烧录到 Flash 中运行时就需要把 ROM 的地址定义到零地址。如果 ROM 定位于 0 地址，向量表包含一系列指令跳转到中断服务程序，否则向量必须被动态初始化。可以在启动程序中添加一段代码，使其在运行时将向量表拷贝到 0 地址开始的存储器空间。对于各未用中断，使其指向一个只含返回指令的哑函数，以防止错误中断引起系统的混乱。

这是上电瞬间 REMAP 之前所执行的中断向量，实际上是要被连接到 FLASH 的地址；但是在 REMAP 之前是 0 地址的地方。可以认为应用程序总是从复位异常处理程序开始执行的，因此复位是异常处理程序但不需要返回。

	B	InitReset	;复位
undefvec	B	undefvec	;未定义指令
swivec	B	swivec	;软件中断
pbatvec	B	pabtvec	;预取中止
dabtvec	B	dabtvec	;数据中止
rsvdvec	B	rsvdvec	;保留
irqvec	B	irqvec	;外部中断
fiqvec	B	fiqvec	;快速中断

(3) 初始化存储器系统

ARM 具有非常灵活的存储器地址分配特性，就是启动 ROM 的地址重映射 (remap)。系统在上电之后，紧接着就从地址 0 开始取得第一条指令的代码执行，所以地址 0 必须在上电的时候就存在可执行的正确代码，也就是地址 0 的地方应该是 ROM 区，至少在上电之后的一小段时间内应该是 ROM 区。对于 ARM 而言，中断和异常的入口地址是 0~20H。但是因为 ROM 的访问速度相对较慢，每次中断发生后，都要从读取 ROM 上面的向量表开始，影响了中断速度。因此，ARM 处理器提供一种非常灵活的地址重映射方法，把 0 地址重新指向到 RAM 中去，即启动地址重映射 (remap)。

为保证重映射之后提供正确的中断入口地址，在重映射之前就必须把中断和异常向量表拷贝到内部 RAM 中。其程序实现如下：

```

mov r8, #RAM_BASE_BOOT          ; RAM_BASE_BOOT 是重映射
                                前内部 RAM 区地址
add r9, pc, #-(8+.-VectorTable) ; VectorTable 是异常向量表入口
ldmia r9!, {r0-r7}              ; 读 8 个异常向量
stmia r8!, {r0-r7}              ; 保存 8 个异常向量到 RAM 区
    
```

```
ldmia    r9!, {r0-r4}           ; 读 5 个异常处理程序绝对地址
stmia    r8!, {r0-r4}           ; 保存 5 个异常处理程序绝对地址到
                                   RAM 区
```

AT91M40800 提供了一个 REMAP 命令来实现重映射。上电复位的时候，0 地址的地方映射到 NCS0 所接的外围器件，内部 RAM 映射到 0x00300000 处。当把 EBI 的重映射控制寄存器的重映射命令位 (EBI_RCR 的 RCB 位) 置 1 之后，内部 RAM 就映射到 0 的位置，NCS0 所接的外围器件映射到 0x01000000 处。如果这个时候中断和异常向量表已经被复制了，则上电复位的一段时间内，CPU 可以在 0 地址取到可执行的有效的指令代码，然后在应用程序执行时能够以更快的速度响应中断和异常，并且可以通过指令的执行动态地进行改动。其程序实现如下：

```
sub    r10, pc, # (8+.-InitTableEBI) ; InitTableEBI 是片选寄存器和重映射寄
                                   存器入口
ldr    r12, PtInitRemap             ; PtInitRemap 是重映射后跳转地址
ldmia  r10!, {r0-r9,r11}            ; 加载片选寄存器和重映射寄存器
stmia  r11!, {r0-r9}                ; 进行重映射并进行寄存器配置
mov    pc, r12                      ; 跳转并停止流水线
DCD    InitRemap                    ; 重映射之后跳转目的地址
InitRemap
```

(4) 初始化堆栈、中断寄存器

ARM 处理器具有 7 种处理器模式：用户模式、FIQ 模式、IRQ 模式、管理模式、中止模式、未定义模式和系统模式。除用户模式外的其它模式称为特权模式。特权模式是为了响应中断或异常，或访问系统保护的资源。特权模式可以自由地访问系统资源和改变模式。

每一种模式都有独立的堆栈指针寄存器 (SP)，因此对使用到的模式都要定义堆栈地址给堆栈指针寄存器 SP。改变状态寄存器 (CPSR) 内的状态位，使处理器切换到不同模式，然后给 SP 赋值，就可以实现堆栈的初始化。堆栈的大小根据需要而定。例如 FIQ 模式堆栈的初始化程序如下：

```
ldr    r0,    =TOP_EXCEPTION_STACK      ; 设置栈顶
msr    CPSR_c, #ARM_MODE_FIQ | I_BIT | F_BIT ; 切换到 FIQ 模式
```

```
mov    r13, r0                ; 进行堆栈初始化
sub    r0, r0, #FIQ_STACK_SIZE ; 得到栈底地址
```

(5) 初始化应用环境

在 ARM 处理器中任何初始化变量的初始化值必须从 ROM 中拷贝到 RAM 中，所有其他变量必须初始化为 0。程序可执行映像一开始总是存储在 ROM/FLASH 里面的，其 ROM 部分既可以在 ROM/FLASH 里面执行，也可以转移到速度更快的 RAM 中去；而 RW 和 ZI 这两部分是必须转移到可写的 RAM 里去。系统使用 FLASH，对 FLASH 的擦除和写入操作也一定要在 RAM 里运行。所谓应用程序执行环境的初始化，就是完成必要的从 ROM 到 RAM 的数据传输和内容清零。

ARM 连接器定义了生成映像文件时的各执行域的标识符号，在启动代码中利用这些符号完成各域的地址重新定位。

```
IMPORT |Image$$RO$$Limit|          ;ROM 中代码段结尾
IMPORT |Image$$RW$$Base|           ;初始化的 RAM 基址
IMPORT |Image$$ZI$$Base|           ;零初始化段的基址
IMPORT |Image$$ZI$$Limit|          ;零初始化段的结尾
```

程序先把 ROM 里|Image\$\$RO\$\$Limit|开始的 RW 初始数据拷贝到 RAM |Image\$\$RW\$\$Base| 开始的地址，当 RAM 这边的目标地址到达|Image\$\$ZI\$\$Base|后就表示 RW 区的结束和 ZI 区的开始，接下去就对这片 ZI 区进行清零操作，直到遇到结束地址|Image\$\$ZILimit|。

(6) 进入 C 程序

当所有的系统初始化工作完成之后，在进入主程序之前，需要确定主程序代码的编译模式是 ARM 还是 THUMB，由此决定相应的跳转指令。系统采用 ARM 模式，假定用户主程序入口函数是 main()那么就直接写成下面这样：

```
IMPORT      main
B          ;跳转到主程序
```

阶段二的代码通常用 C 语言来实现，以便于实现更复杂的功能和取得更好的代码可读性。下面给出了设计 Bootloader 第二阶段代码的总体框架如图 4-1。

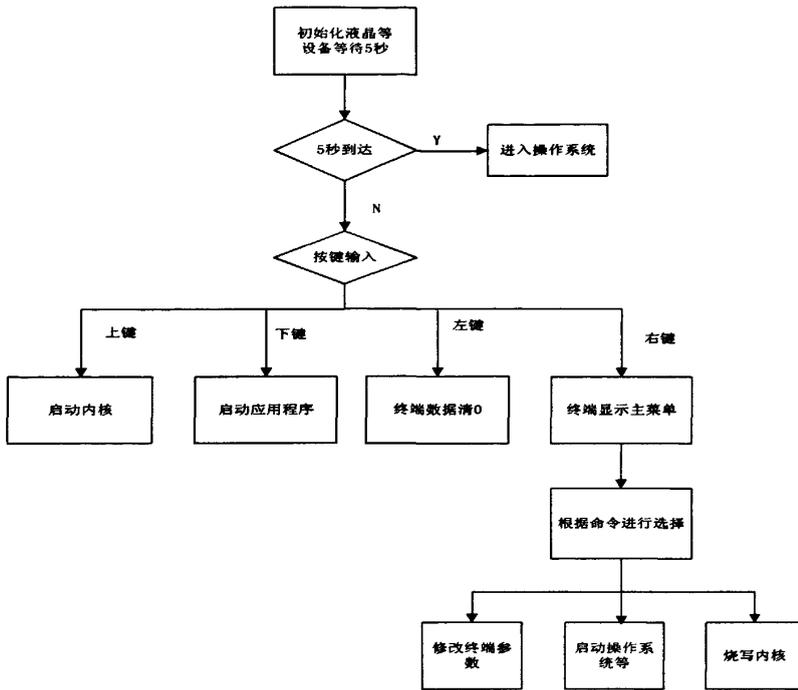


图 4-1 stage2 程序流程图

本系统的 Bootloader 程序代码均在 ADS1.2 开发环境中完成，ADS1.2 是一个使用方便的集成开发环境，全称是 ARM Developer Suite v1.2。它是由 ARM 公司提供的专门用于 ARM 相关应用开发和调试的综合性软件。在功能和应用性上比较 SDT 都有提高，是一款功能强大又易于使用的开发工具。

ADS 囊括了一系列的应用，并有相关的文档和实例的支持。使用者可以用它来编写和调试各种基于 ARM 家族 RISC 处理器的应用。可以用 ADS 来开发、编译、调试采用包括 C、C++和 ARM 汇编语言编写的程序。

ADS 主要由以下部件构成：

- 命令行开发工具；
- 图形界面开发工具；
- 各种辅助工具；
- 支持软件。

其中重点介绍一下图形界面开发工具。

- AXD 提供给基于 Windows 和 UNIX 使用的 ARM 调试器。它提供了一个完全的 Windows 和 UNIX 环境来调试你的 C,C++, 和汇编语言及的代

码。

- CodeWarrior IDE 提供基于 Windows 使用的工程管理工具。它的使用使源码文件的管理和编译工程变得非常方便。但 CodeWarrior IDE 在 UNIX 下不能使用。

下图为 ADS1.2 开发环境基本图形界面如图 4-2。

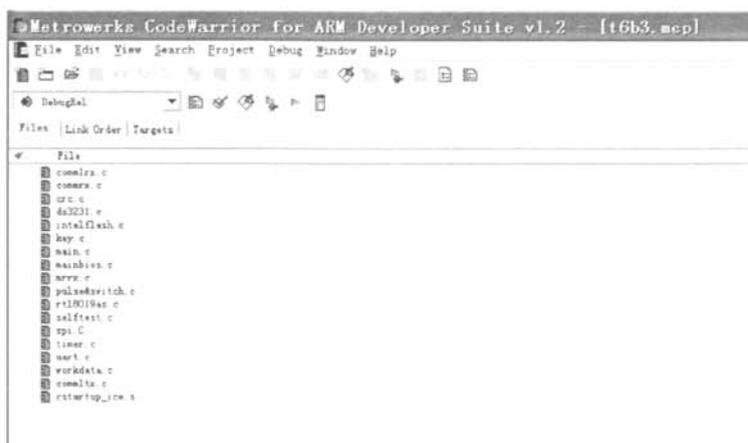


图 4-2 ADS 开发环境示意图

4.1.3 内核移植和裁剪

uClinux 的移植要按照其本身已经支持的硬件平台的结构，来划分所做的工作，具体可分为 3 个层次：基于体系结构的移植、基于硬件平台的移植和基于具体硬件电路的板及移植。

本人在移植 uClinux 时，所采用的 uClinux 系统参考已经有运行于 AT91M40800 芯片上的版本，在此基础上主要考虑修改与硬件开发平台相关的具体因素，如内存类型和大小、外围设备的配置等等这样极大的节省开发的工作量。对内核源代码需要在以下地方做修改：

(1) 在../linux/arch/armnommu/config.in 中

找到支持 AT91M40800 的部分 EB40LS，添加有关支持系统硬件的 DRAM 参数，

```
if [ "$CONFIG_EB40LS" = "y" ]; then
    define_int DRAM_BASE      0x02040000
    define_int DRAM_SIZE     0x00ec0000
```

添加选择系统工作的主时钟频率，

```
choice 'Frequency'
    32.768M          XXX_FRQ32768
if [ "$XXX_FRQ32768" = "y" ]; then
    define_int CONFIG_ARM_CLK      32768000
```

添加 flash 的参数，

```
choice 'FlashSize'
    8M          XXX_FSIZE8M
define_int FLASH_MEM_BASE 0x01040000
if [ "$XXX_FSIZE4M" = "y" ]; then
    define_int FLASH_SIZE      0x001a0000
```

(2) .../linux/include/asm-armnommu/arch-atmel 目录文件中，主要包括与 CPU 内核直接相关的指令、变量、参数的定义，包括一系列.h 文件。

(3) 在 CPU 已得到支持的前提下，还需考虑一些影响内核的因素，如与中断处理相关的函数等，在../linux/arch/armnommu/lib 目录下的相关文件中，主要包括，irqs-atmel.c、io-atmel.c 等；

(4) .../linux/arch/armnommu/kernel/head-arm-atmel.S 修改启动代码

(5) .../linux/arch/armnommu/Makefile 是控制如何链接各个目录下的目标文件形成一个镜像文件。在此文件里设置内核的起始运行地址。

```
ifeq ($(CONFIG_ARCH_ATMEL), y)
HEAD      := arch/armnommu/kernel/head-arm-atmel.o
COMPRESSED_HEAD = head.o
MACHINE = atmel
TEXTADDR   = 0x02080000
```

TEXTADDR 是定义内核起始运行地址的标号

(6) .../linux/Makefile,这是内核编译控制文件。在此文件中定义 CPU 体系结构、交叉编辑工具名称。

```
arch:=armnommu
CROSS_COMPILE:=arm-elf-
```

(7) ROM 文件系统的配置。配置 uClinux 往往需要在文件系统目录/etc 下写一些文件，uClinux 源码包会在 make 时自动建立整套 ROMFS，所以需要在编

译 uClinux 之前把这些配置文件写好, 存放在正确的地方, 然后由 make 拷贝到 romfs 里再打包生成 romfs.img。

在对内核源文件修改完后, 就要进行内核的配置。对内核的配置有 3 种方法: make config、make menuconfig、make xconfig。本文选用 make menuconfig 基于图形界面的配置, 运行命令 make menuconfig, 出现如图 4-3 配置界面。

- 在 Platform dependent setup (开发平台支持) 对话框中 CPU 选项选择 ARM7、Platform 选项中选择 Lineo_EB40LS、Frequency 选项中选择 32.768M、FlashSize 选项选择 8M、Kernel executes form 选项选择 ROM, 如图 4-3。
- 在 General setup(总体设置)对话框中选择 Use new compilation options(for GCC 2.8 and above)、Debug kernel errors、Networking support 以提供网络支持、Less wastful kmalloc.c/page_alloc.c(EXPERIMENTAL)、Support compressed binflat applications。
- 在 Floppy IDE and other block devices 对话框中选中 ROMdisk memory block device、FLASH device suport、Fixed Ram device、Lcd Block device。
- 在 Character devices (字符设备) 对话框中选中 support for extended UART、Console on extended UART、Atmel AT91 serial support。
- 在 Networking options(网络选项)对话框中选中 Network firewalls、Tcp/Ip networking、IP:forwarding/getewaying、IP:syn cookies。
- 在 Filesystems(文件系统)对话框中选中 Second extended fs support、/proc filesystem support、NFS filesystem support。
- 在 ZYDZ Configuration 对话框选中 support for JIAOCAI。

完成内核配置后, 就开始编译内核, 一般来说要依次执行如下命令。

make dep、make lib_only、make user_only、make romfs、make image、make。依据上述配置产生的结果文件, 寻找各文件之间的依赖关系, 建立编译依赖规则, 决定哪些需要编译、哪些不能编译, 为编译文件、更新目标文件做准备。编译完成后会在目录下生成文件 image.ram、image.rom、romfs.img, image.rom 是一个压缩的内核, 可以烧写到 flash 中, 从 flash 启动内核, 把 image 解压到 RAM 中的指定位置, 然后运行。Image.ram 是包含有调试信息的内核, 可以直接下载到开发板的 RAM 中运行, romfs.img 是文件系统映像文件。将编译好的内核映像文件烧到 Flash 上, 上电启动后通过超级终端看到输出信息如图 4-4。

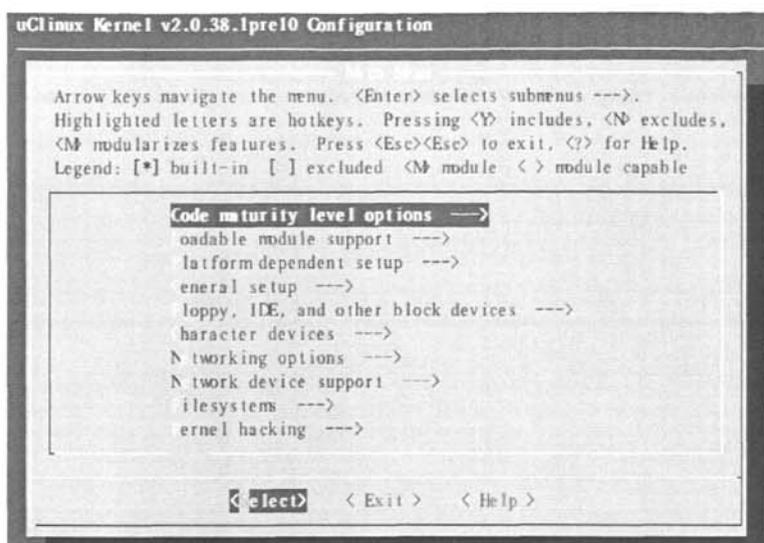


图 4-3 内核配置界面



图 4-4 超级终端输出内核启动信息

4.1.4 LCD 设备驱动程序

HT1622 中 WRITE 命令的操作时序如下图 4-5 所示，根据时序图编写 LCD 写函数。


```

int lcd_init(void)
{
    int rc;
    rc=register_blkdev(MAJOR_NR, DEVICE_NAME, &lcd_block_fops);
    if(rc<0)
    {
        printk(KERN_ERR DEVICE_NAME ": Unable to get major %d\n",
                MAJOR_NR);
        return -EBUSY;
    }
    printk("LCD block device , (c) \n");
    SendCmd(SYSEN);           //开启系统振荡器
    SendCmd(LCDON);          //开启 LCD 偏压产生器
    LCD_ON();                 //点亮 LCD
}

```

下面将 LCD 设备驱动加到 uClinux 内核中，编写完设备驱动程序后，需要添加到内核中了，这需要修改源码，然后重新编译内核，具体步骤如下：

(1) 将设备驱动文件复制到 kernel/drivers/char 目录下，该目录下保存了字符型设备的驱动程序。

(2) 在 kernel/drivers/char 目录下的 makefile 文件中添加如下代码：

```

ifeq ($(CONFIG_LCD_DRIVER),y)
L_OBJS+=LCD_DRIVER.o
endif

```

或

```

if ($obj-$(CONFIG_DEVICE_DRIVER))+=LCD_DRIVER.o

```

如果在配置内核时已经选择了支持该设备，则在编译内核时，将把 LCD_DRIVER.c 编译成 LCD_DRIVER.o

(3) 在 kernel/drivers/char 目录下修改 config.in 文件。在 comment 'character devices' 下面添加

```

Bool'support for DEVICE_DRIVER'CONFIG_DEVICE_DRIVER

```

这样在运行 make menuconfig 配置字符文件设备时就会出现 support for LCD_DRIVER 的字样。如果选中并编译通过，驱动程序就加到内核中了

(4) 设备文件都包含在 /dev 目录下，使用 mknod 命令创建设备文件。

4.2 校表程序设计

校表是电能表设计中非常重要的环节,ATT7022B 上电复位后校表寄存器的初始数据为默认值,此时读出的计量参数值和实际参数值不符,因而需要对校表寄存器进行设置以将测量值减小到误差范围之内,校表可按高频输出参数设置、比差补偿区域设置、角差补偿区域设置、功率增益校正、相位校正、启动电流设置、功率增益校正、参量累加模式设置、电压校正、电流校正的先后顺序进行。校表流程如图 4-6。

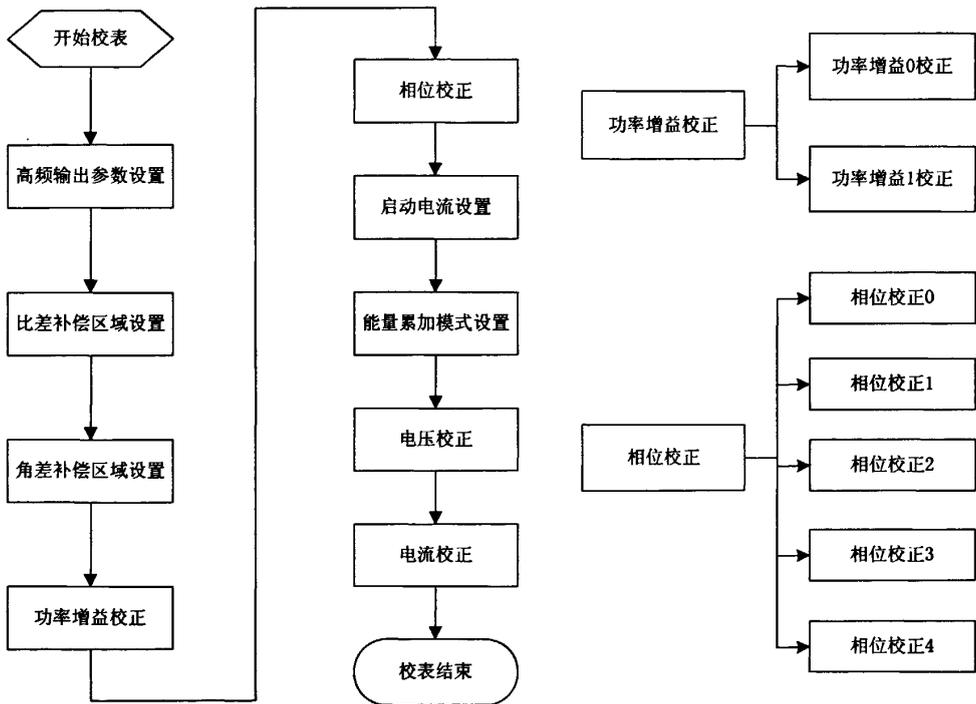


图 4-6 校表流程图

4.3 终端应用程序设计

4.3.1 应用程序总体设计

终端应用程序基于 uClinux 操作系统,各个模块由进程独立实现。模块关系如图 4-7。主进程 tserver 主要负责信号初始化,创建各个子进程,循环等待子进

程消息，对子进程进行管理。电表通信模块，子进程 `ammeter` 负责处理电表通讯，包括数据返回、设置。抄表模块，子进程 `meter` 通过与各种电表通讯取得需要的数据。红外模块，子进程 `infrared` 用于红外通讯给电表设置参数，并且用于切换 A/B 屏显示。电表交采模块，子进程 `amsample` 通过交采芯片 ATT7022B 取得电表所需要的各种数据，并处理相应的事件记录。通信模块，子进程 `comm` 控制与主站进行远程通信，返回主站所需要的各种类型数据。数据处理模块，子进程 `datacal` 定义了所有全局变量并通过计算处理终端所需要的各种类型数据。按键和显示模块，子进程 `menu` 通过按键控制显示每一屏，按上键向上翻一屏并开启背光，按下键向下翻一屏开启背光，不按键每隔 10 秒翻下一屏并关闭背光。定时任务模块，子进程 `time` 控制其它子进程定时任务，起到定时器的任务。下面介绍部分模块的程序设计。

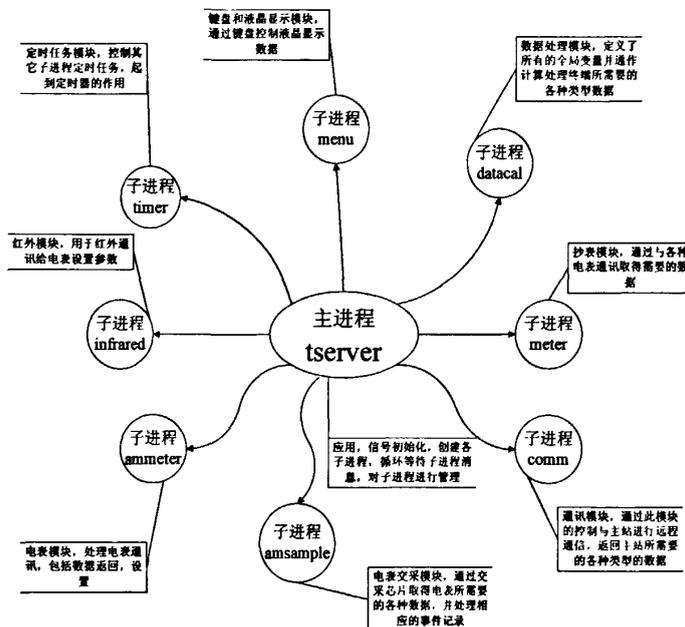


图 4-7 各个模块关系图

4.3.2 主进程模块

主进程 `tserver` 模块主要完成对终端应用程序各个模块的同步、控制：

- 终端硬件检测
 - 系统级硬件检测

- 启动应用程序模块
创建模块通信管道 IPC，舒适化全局数据，按序启动各个模块。
- 事件记录
各个模块版本信息、终端上电记录、终端掉电记录和终端故障记录等等以文件形式保存。

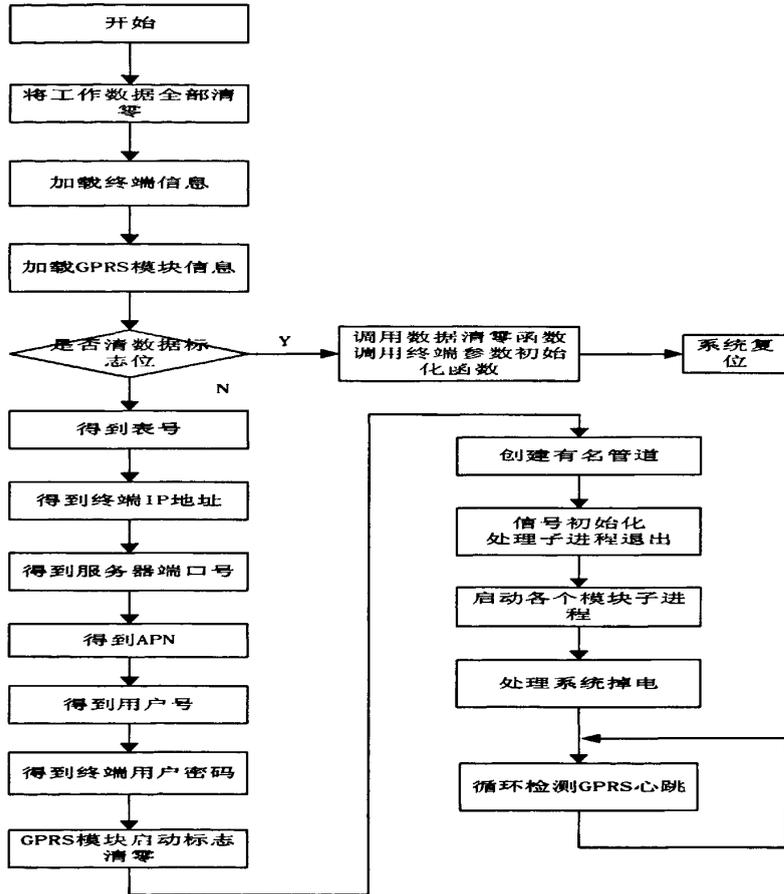


图 4-8 主进程流程图

首先将 SRAM 中的工作数据全部清零，从 flash 中加载终端和 GPRS 信息，如果有终端清零标志位，终端数据初始化，终端参数初始化，终端参数初始化主要包括设置抄表间隔、购电量、重拨 PPP 间隔、永久在线模式、电压上上限、电压合格上限、电压合格下限、欠压门限、电流上上限、电流上限、视在功率上上限、视在功率上限等，然后系统复位。如果没有终端清零标志位，则加载

表号、终端 IP 地址、服务器端口号等一些终端参数，接着创建管道，信号初始化，启动各个模块的子进程，并循环等待子进程的消息，循环监测 GPRS 心跳，具体流程如图 4-8。

4.3.3 按键和显示模块

根据按键和显示需求，显示分 A/B 屏进行显示，A 屏显示为供电人员抄表所需计量数据，B 屏显示记录的各项事件包括时段等编程内容，终端参变量显示，比如通信参数（主站 IP 地址和端口号、虚拟专网用户名、密码、终端地址及行政区划码等）、通信状态（信号强度）、终端软件版本等。显示分为两套轮显数据，自动轮显、按键轮显两种方式。通过红外和右键可进行 A/B 屏切换，带液晶背光功能，通过按键激活背光显示，连续显示 10 秒钟后关闭背景光，进入省电模式。按键显示流程图如图 4-9 所示。

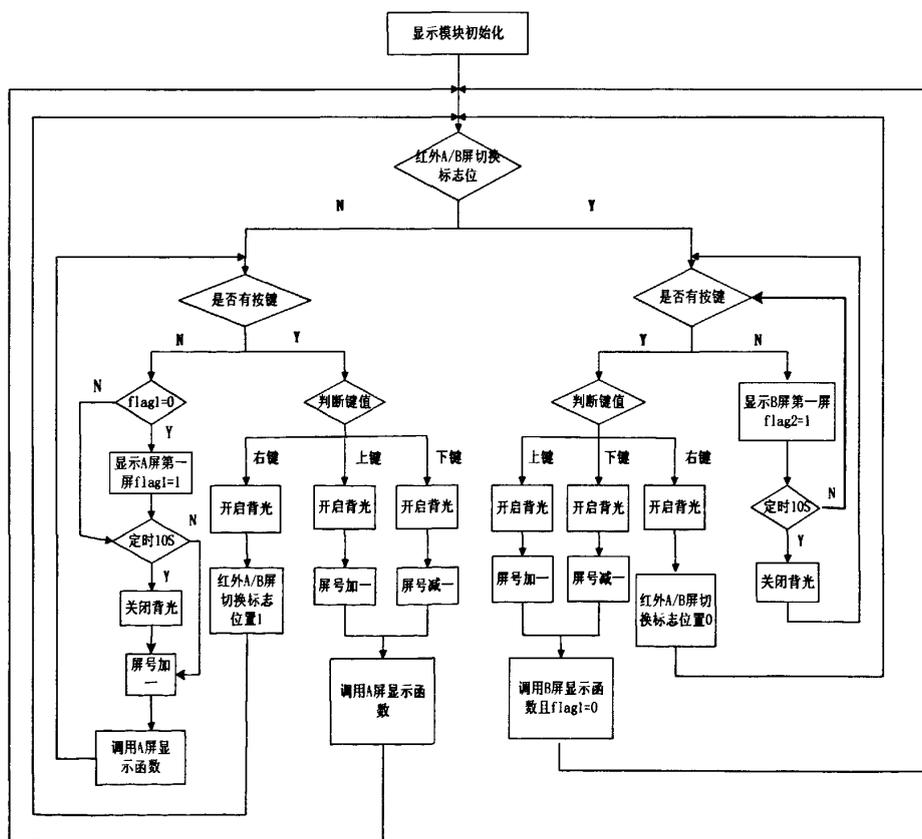


图 4-9 按键显示模块流程图

4.3.4 电表通信模块

电表需提供的数据包含配变终端所需要的计量数据及作为《湖北电网电能表通讯规约 V1.0》中所规定的简易多功能表应具备的数据，外部设备能够通过 485 接口抄得这些数据，所用协议为《湖北电网电能表通讯规约 V1.0》和《DLT645-1997 多功能电能表通信规约》。

根据设计需求，终端应具有简易多功能电能表的功能，其它终端通过 485 抄表口可以与其通信，必须满足通信国际规约 DLT645 协议，电表模块整体示意图如 4-10，流程图如图 4-11。对电表模块进行初始化，本终端有两个 485 接口，确定哪个接口作为通信接口，并同时初始化 485 接口，打开数据计算模块管道。循环等待是否接收到数据，接收到数据调用检查帧函数进行解帧。根据 DLT645 协议定义电表发送帧和接收帧结构体如下。

```
struct MeterTxBufStruct
{
    U8 Buffer[128];
    U16 BufferLength;
};

struct MeterRecFrameStruct
{
    U8 Addr[6];        //帧地址域
    U8 Length;        //帧长度
    U8 DI[2];         //数据标识
    U8 DC;            //控制码
    U8 Data[128];     //数据域
    U8 CS;            //校验码
};
```

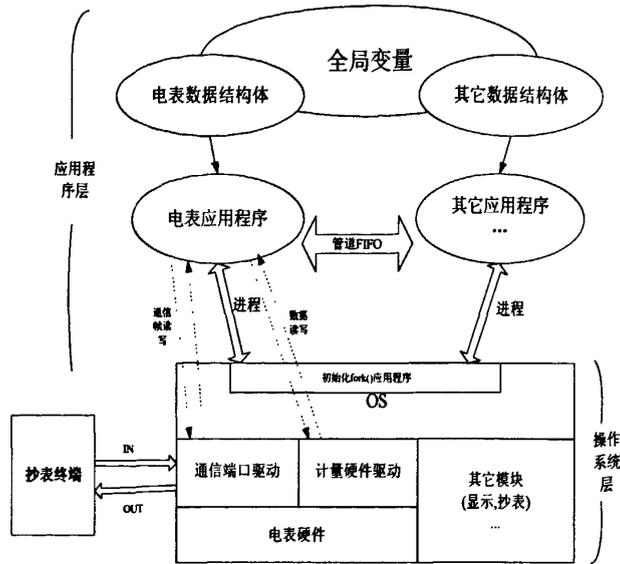


图 4-10 电表模块整体示意图

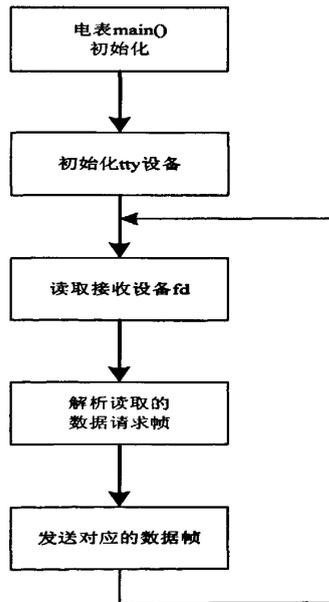


图 4-11 电表模块流程图

4.3.5 抄表模块

抄表模块用于抄读电能表的数据，对数据进行统计、保存。支持多种通信协议，包括国标协议、鄂规 V1.0 协议、ABB 表协议、威胜协议、EDMI 协议、兰吉尔协议等。通讯波特率可设置为 300, 600, 1200, 2400, 4800、9600（根据规约进行设置）。终端所采集的数据都来自于按终端参数设置的抄表间隔定时抄电能表。电能表分为外部表和内部表，终端可通过 485 接口使用多种电能表通信协议抄外部表的数据；对于内部表，终端可直接读取电表数据区的数据；判断当前所抄电表为外部表或内部表的依据是检测电表参数中的通信地址是否为内部表的通信地址。

使用不同协议的电能表，抄表数据项会有所不同，终端需要根据所抄电表的协议来决定哪些数据能抄哪些数据不能抄，对于不能抄的数据项，相应数据区作清 0 处理。其中抄表模块流程图如图 4-12，抄表任务流程图如图 4-13。

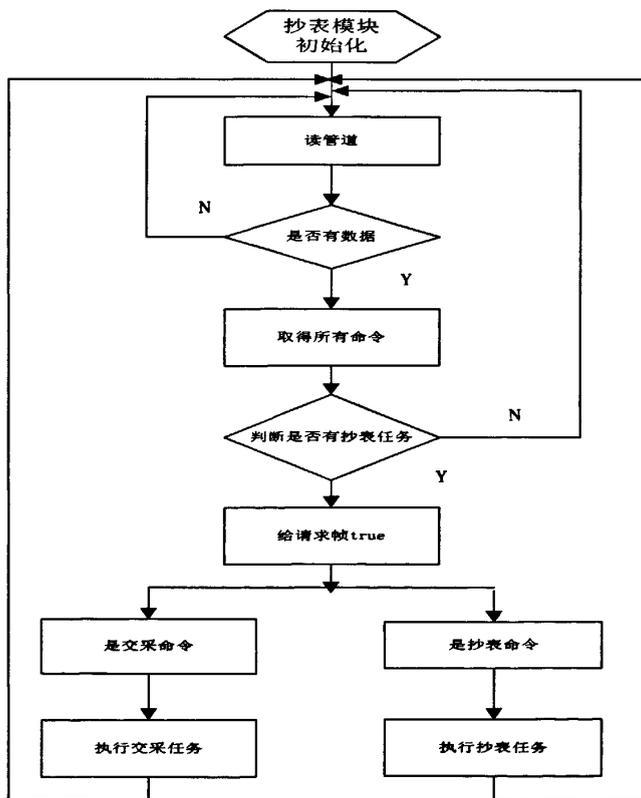


图 4-12 抄表模块流程图

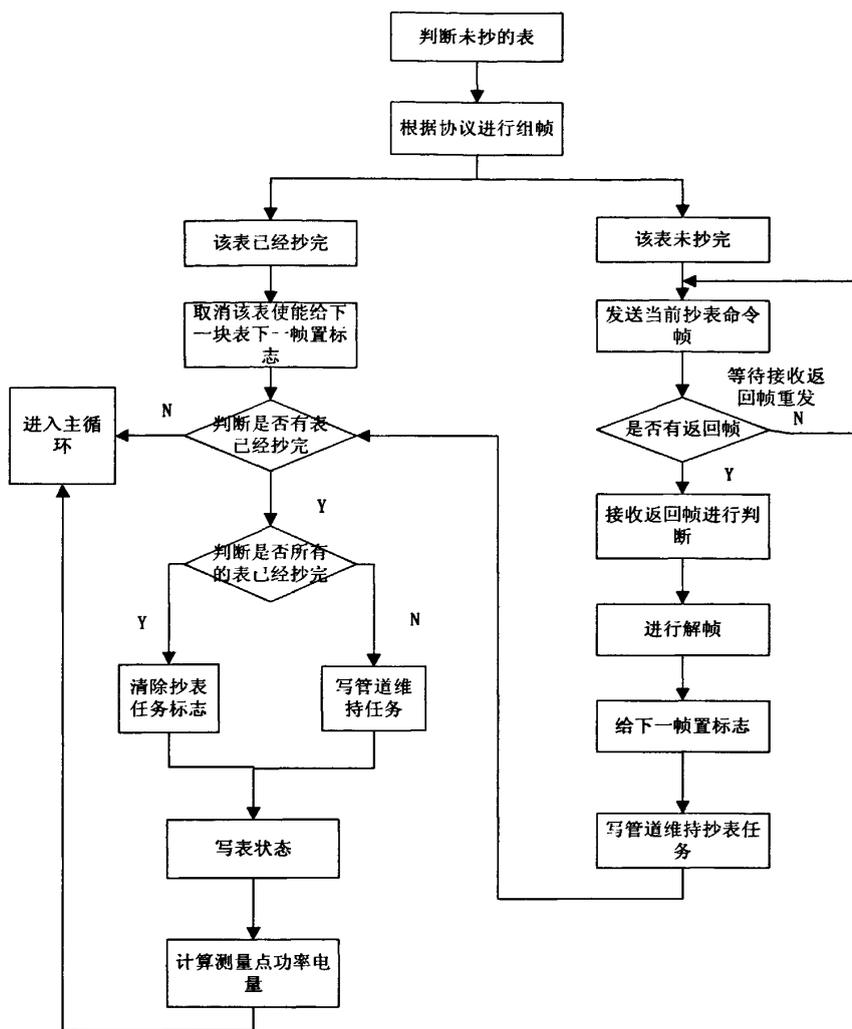


图 4-13 抄表任务流程图

4.3.6 交采模块

电表交采模块，子进程 `amsample` 通过交采芯片 `ATT7022B` 取得电表所需要的各种数据，根据数据缓冲区采集的数据以及各个参数的计算方法分别计算出电网的电压、电流有效值、有功功率、无功功率、功率因数、正向有功电量等参数，分析电网性能并确定采取措施，并处理相应的事件记录，流程图如 4-14。

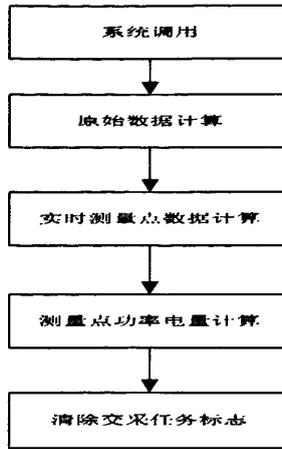


图 4-14 交采任务模块流程图

原始数据计算，ATT7022B 的一些数据寄存器采用补码形式给出，因此要对这些原始数据做一些转化。例如：功率寄存器采用补码形式给出，最高位是符号位。

A/B/C 分相功率参数：X

X：24 位数据，补码形式，如果 大于 2^{23} ，则 $XX=X-2^{24}$ ，否则 $XX=X$

实际的 A/B/C 分相功率参数为： $XXX=XX*2^{15}/2^{23}$

A/B/C 合相功率参数：T

T：24 位数据，补码形式，如果大于 2^{23} ，则 $TT=T-2^{24}$ ，否则 $TT=T$

实际的 A/B/C 合相功率参数为： $TTT=TT*2^{17}/2^{23}$

功率单位是瓦，与脉冲常数有关系。按照上述公式算的的功率值是基于 3200imp/kwh，如果设置脉冲常数为 EC，则实际功率值等于计算值再乘以 3200/EC 的系数。

第 5 章 配变监测终端调试

5.1 测试环境要求

(1) 大气环境条件

性能与功能实验应按标准大气条件进行，即：

①环境温度：+15~+35℃，

②相对湿度：45%~75%，

③大气压力：86~108kPa，

在每一项试验期间，环境条件应相对稳定。

(2) 电源条件

试验时电源条件为：

①频率：50Hz，允许偏差：-2%~+1%，

②电压：220V，允许偏±20%。

5.2 主控板调试

(1) 在不上电的情况下用万用表测量+3.3V 对地、+5V 对地以及+3.3V 和 +5V 之间是否短路，测量 U7 的 Pin5 和 Pin8 之间是否短路。如果短路，须查明原因并排除，不短路，则继续。

(2) 加上电源，用万用表测量+3.3V 是否正常（偏离值在±5%范围内），隔离电源+5V 是否正常（U7 的 Pin5 和 Pin8 之间，偏离值在±5%范围内）。

(3) 掉电后将 CPLD 下载电缆一端插入到 PC 机，一端插入到 XS8，使用 QUARTUS 将编译好 CPLD 文件下载到 U16，下载完成后 V16-V18 应立即点亮成红色。

(4) 将 USB 接口转串口电缆一端插入到 PC，一端插入到 T6B 主控板 XS4 的下面一个 USB 接口并按下 SA2 自锁开关。打开超级终端，串口设置为 9600，8bits，无校验。运行 ADS 软件，打开 T6B 的工程文件并点击“debug”图标或按 F5，弹出如图 5-1 界面：

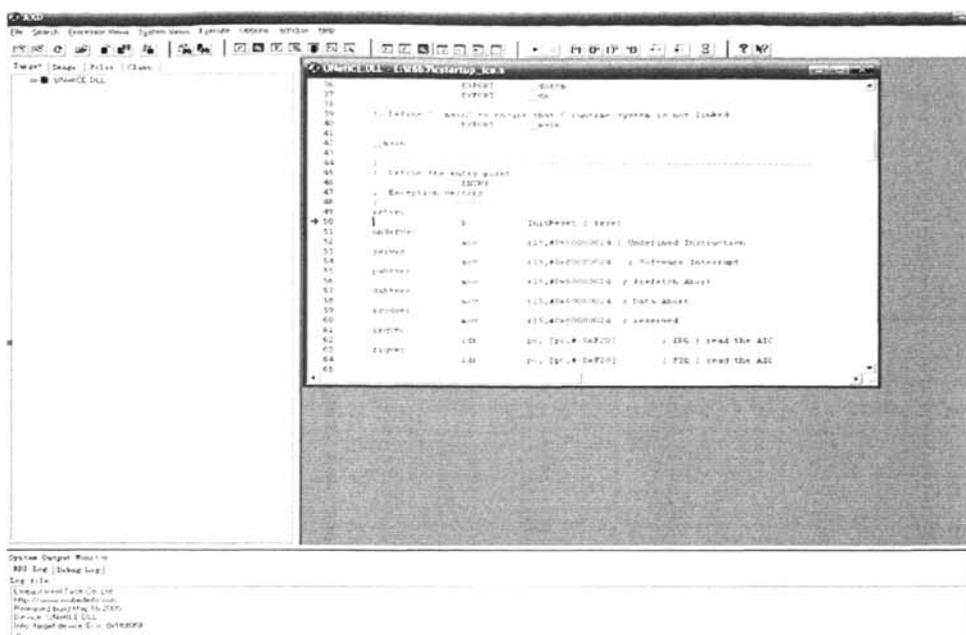


图 5-1 ADS 下调试界面

点击“run”图标，超级终端会显示如下界面：

BootLoader Mode!

ok

Tiny Bootloader for ZY-T6B
[Aug 02 2007 10:55:10][32168L]

Please Enter anykey to stop startup OS!

\$

按“H”进入调试主菜单，出现如下界面：

Menu

- 1 - Download image to ram ...
- 4 - Program OS image to flash ...
- 5 - Go from ram ...
- 6 - Go from flash ...
- 7 - Program Bloader image to flash ...
- 8 - Self Test ...
- 9 - NetPara Set...
- 0 - ReStart...
- H - Show this menu ...

\$

按暂停图标停止运行仿真器，在“File”菜单内选择“Load Memory From File”，会弹出如图 5-2 界面：

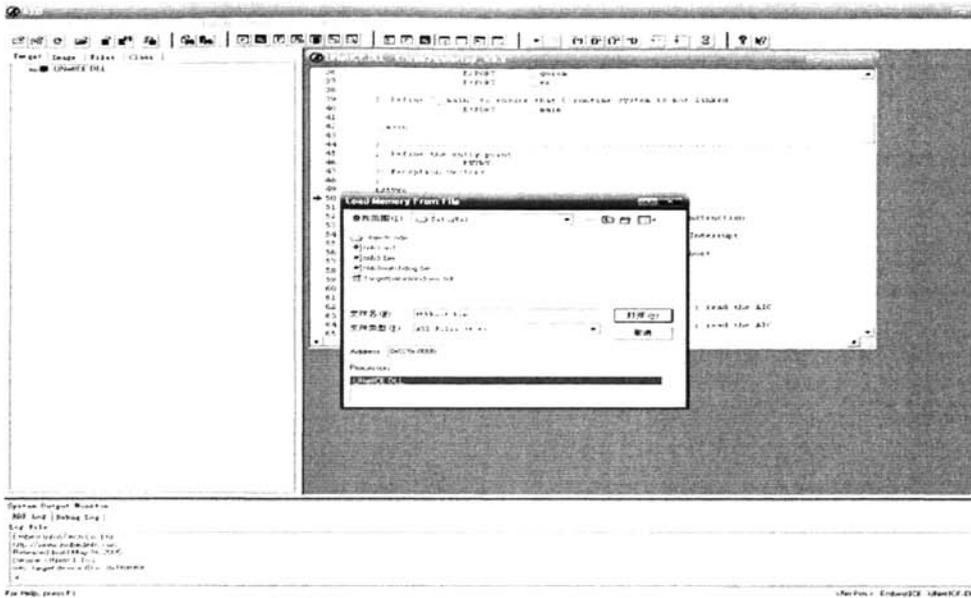


图 5-2 ADS 下烧写 Bootloader 界面

在“Address”处填写“0x020c0000”，选择所要固化的 Bootloader 文件，然后按“打开”，文件载入后，重新运行 AXD，在超级终端界面选择“7”，然后根据提示按“y”，固化完成后会显示“ok”。用相同的方法将操作系统映像写入到 FLASH。然后将拨码开关 S1 的第 1 位拨到“ON”

(5) 按下右键 SB4，进入到 Bootloader Mode。按“H”进入到调试菜单，再按“8”进入到自测菜单，根据菜单提示，按下“4”键测试实时时钟，屏幕应能显示出主控板的时钟，再次按下“4”键，看到时钟的秒正常增加。

(6) 用导线将两个 RS-485 接口的 A、B 分别对接，在自测菜单按下“5”键，指示灯 V10、V11 应红绿交替闪烁，屏幕有“RS4851 OK!”和“RS4852 OK!”字样。

(7) 测量电池两端电压，应与额定值一致。测量 U23 的 Pin6，高电平说明检测到的电压正常。

(8) 重新进入到 BootLoader Mode 模式，在主菜单按“9”进入到网络参数设置，然后选择“1”，超级终端将提示设置网络参数。将命令“058.048.110.010 02004 010.000.000.011 255.255.255.000 010.000.000.101”复制并粘贴到超级终端的提示下，终端会返回“ok”字样，然后参数设置菜单里选择“7”查看所有参数，返回值与所设置的一致。“058.048.110.010”是登网主机 IP（根据实际值设置），“02004”

是端口号，“010.000.000.101”是设置到终端的 IP 地址，也可以设置为与其它 PC 机 IP 不重复的值。

(9) 将网络接口板插入 J5、J6，用交叉网线连接装有 Linux 操作系统的 PC 和网络接口板，同时按下复位按键 SB6 和上键 SB2，超级将显示“Debug Mode”，通过网络接口板在 PC 机 Linux 操作系统下将应用程序文件写入到 FLASH。

(10) 单板调试过程中，查看指示灯状态。指示灯状态说明见表 5-1，经调试指示灯状态正确。

表 5-1 指示灯状态说明

位号	指示信息	状态说明
V19	终端运行	点亮 1 秒，熄灭 1 秒
V12	GPRS 通信	绿色接收，红色发送
V10	RS-485 接口 1 通信	绿色接收，红色发送
V11	RS-485 接口 2 通信	绿色接收，红色发送
V13	本地串口/红外通信	绿色接收，红色发送
V16	A 相告警信息	暂时不用
V17	B 相告警信息	暂时不用
V18	C 相告警信息	暂时不用
V14	有功脉冲输出	根据有功功率大小决定是否 闪烁及闪烁快慢
V15	无功脉冲输出	根据无功功率大小决定是否 闪烁及闪烁快慢

5.3 交采板调试

交采板顶层俯视图见图 5-3。

- (1) 检查元器件是否有漏焊、错焊。
- (2) 在不上电的情况下用万用表测量+3.3V 对地、+5V 对地以及+3.3V 和 +5V 之间是否短路，如果短路，查明原因并排除。
- (3) 将交采板插入到电源底座上，将调试过的主控板安装好，加上电源后，

在 BOOTLOAD 模式下根据菜单对交采板进行测试，用超级终端监视打印信息，给终端供电的一相应有电压。

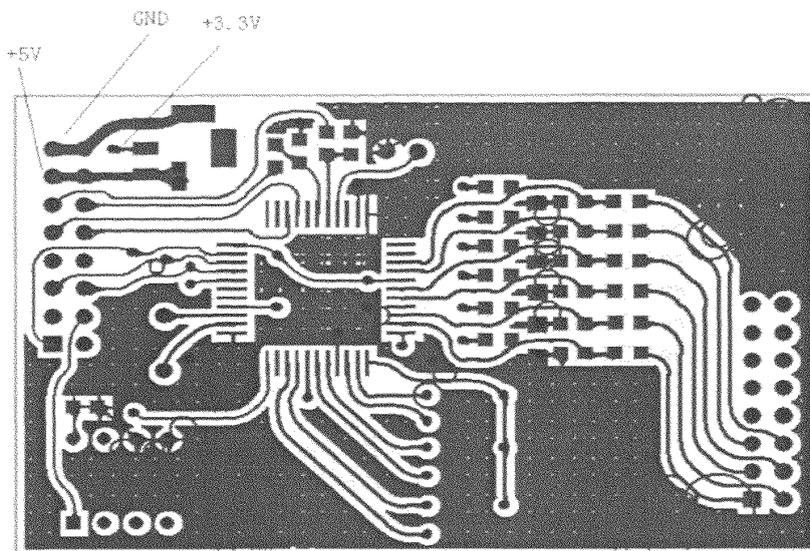


图 5-3 交采板顶层俯视图

5.4 整机调试

5.4.1 功能调试

(1) 同时按下复位键和下键，先放开复位键，然后放开下键，运行灯 (V19, 绿色) 闪烁，闪烁频率为点亮 1 秒，熄灭 1 秒；LCD 背景光点亮，所有字符全部显示，几秒钟后，背景光应熄灭，约 10 秒后 LCD 显示屏进入滚动显示状态，除运行灯外的其它指示灯全部熄灭。

(2) 将红外切换开关处于红外通信位置，拨码开关处于红外编程允许状态，用手持红外遥控器切换到 B 屏，然后设置 IP 地址、APN、端口号、行政区划码、终端地址、表号等参数，通过上下键查看终端显示值与所设置的是否一致。然后切换到 A 屏。

(3) 查看第 000 屏的高位是否为“1”，为“1”表示登录成功（正常情况下，上电 5 分钟后应能登录成功），通过协同主站测试软件对终端校时并查看终端显示时间是否正确，然后设置抄表参数并查询，看返回值与设置是否一致。

(4) 查询终端状态量，能反映出当前表盖开启状态。

(5) 用串口测试软件通过 RS-232/RS-485 转接头组帧查询当前电压、电流（符合无功补偿控制器发送帧协议）与 RS-485 接口 2 通信，终端返回的电压、电流应与显示值一致。

(6) 用多功能电表协议调试器通过 RS-232/RS-485 转接头与 RS-485 接口 1 通信，终端应正确返回数据（可通过抄读实时电压并与显示器的显示值比较）。

(7) 去掉 RS-232/RS-485 转接头，将 380V 分别加在 RS-485 接口的 A、B 端子上 2 分钟，然后去掉 380V，重新测试 RS-485 接口通信，应正确无误。

(8) 观察显示屏，电池符号应不显示。按下键或下键，背景光应点亮且显示内容应变化。查看显示屏时钟走时是否准确

(9) 将可充电电池插入 XS7，在没有交流电源供电的情况下，按“向下”键，电池应投入使用，终端能正常启动并运行。加上交流电源后，在终端应正常运行的情况下去掉交流电源，5 分钟后终端应自动关闭。在确认电池能正常切换后可直接拔掉电池，不必等待 5 分钟。

(10) 将终端安装到校表架上，接上三相电压和零线，并将本地串口与校表架上的 RS-232 串口连接，有功脉冲输出和无功脉冲输出分别与校表架上的有功脉冲接口和无功脉冲接口对接。将功率源的脉冲常数设为 8000。从校表控制台将三相电压设置为 220V，电流设置为 1.5A，使终端进入 Bootloader 模式，在电压、电流校准界面按下控制台菜单上的“设定”，终端即开始自动校准电压、电流、功率，校准完成后将返回校准完成信息，然后从控制台读取终端测量数据并与标准表比较。如果不准确，则重新校准。

(11) 将三相电压设置为 220V，电流设置为 1.5A，功率因数设置为 0.5L，按下控制台菜单上相位的“设定”，终端即开始自动校准功率因数。校准完成后将返回校准完成信息，然后从控制台读取终端测量数据并与标准表比较。如果不准确，则重新校准。校准完成后从控制台对终端进行检测，对终端在不同电压、电流、不同功率因素时的值进行测量，显示结果是否全部合格。在对终端进行检测的过程中，终端的脉冲指示灯应闪烁，校表架上的脉冲指示等也应闪烁。

5.4.2 绝缘、冲击实验

(1) 绝缘电阻

在正常条件下，按表 5-2 的测试电压(额定绝缘电压 $U \leq 60V$ 时，测试电压为 250V；额定绝缘电压 $60V < U \leq 250V$ 时，测试电压为 500V)在终端的端子处测试

各电气回路对地的绝缘电阻。

表 5-2 绝缘电阻实验电压

测试回路	测试电压
485 回路对地	1kV
电源回路对地	1kV

实验结果：经过测量阻值大于等于 10MΩ。

(2) 绝缘强度（交流电压）试验

用 50Hz 正弦波电压对以下回路进行试验。漏电流≤5mA。试验条件见表 5-3。

表 5-3 绝缘强度试验条件

试验回路	试验电压	试验时间
电压电流回路对地	2kV	1min
电压电流回路之间	2kV	1min

实验结果：无击穿和拉弧现象。

(3) 冲击耐压试验

用 1.2/50μs 的标准冲击波按表 5-4 要求对以下回路分别进行试验。

表 5-4 冲击耐压试验要求

试验回路	试验电压	备注
电压电流回路对地	6KV	耐受的冲击电压峰值为 6kV，正、负极性耐压试验各 10 次，两次试验之间最少间隔 3s。
电压电流回路之间		

实验结果：无击穿和拉弧现象。

5.4.3 高低温实验

高低温试验按先低温后高温，然后常温复测的顺序进行试验，低温或高温不合格的产品，经过修复后，应按上述顺序重新进行。高、低温调试项目和要求见表 5-5。

表 5-5 高、低温调试项目和要求

序号	试验项目	实验结果
1	通信功能检查	正确
2	表盖开启状态检查	正确
3	历史数据及参数保存	无丢失现象
4	RS-485 功能检查	正常
5	时钟检查	A)校时误差小于 1s; B)校时 10h 后,误差不大于 1s, 掉电后时钟走时正常;
6	按键、液晶显示	正常
7	红外通信功能	正常

(1) 低温

温度为-27℃,(或根据用户要求的特别温度), 保温 4h, 然后通电 0.5 小时后立即取出按表 5-4 的要求进行检测。

(2) 高温

温度为+57℃,(或根据用户要求的特别温度), 保温 4h, 然后通电 0.5 小时后立即取出按表 5-4 的要求进行检测。

(3) 常温复测及通电老化

经过低、高温试验合格的产品, 恢复到常温后, 对绝缘、冲击性能进行复测。复测合格的产品, 在常温条件下, 加电老化 72h。

常温恢复, 复测不合格的, 修复后, 应重新进行高、低温测试。

第 6 章 总结和展望

6.1 本文总结

随着嵌入式系统复杂程度不断的提高，对 CPU 处理能力的要求越来越高，在 CPU 上开发程度的复杂度使得以前针对嵌入式系统的简单控制流程不适合系统的要求，这就迫使开发人员使用操作系统，开发人员使用操作系统可以直接调用操作系统使用的函数编写自己的应用程序，这就大大缩短的开发周期，提高了开发流程。

本文成功采用 ARM 技术，基于嵌入式 uClinux 操作系统，提出一种集电表功能和通信功能于一体的配变监测终端的设计方案，并通过实际挂网运行，性能稳定，适用于电能测量及用电自动化管理领域，较好的符合了目前国内的电网状况，有较高的实用性和推广价值，本产品已经中标投放市场。本文主要研究了如下内容：

(1) 结合配电自动化系统的特点和对国内外配变监测终端应用现状的分析，提出了一种采用 32 位 ARM，AT91M40800 为核心，计量部分采用高精度电能测量芯片 ATT7022B，带有通信功能的设计方案。

(2) 研究了 uClinux 的源代码，分析了其在内存管理、进程管理设备、驱动方面的特点。

(3) 通过具体项目开发，掌握了基于 ARM7-AT91M40800 的嵌入式系统硬件电路系统设计方法，并研究了 uClinux 内核的配置和编译方法，并将它移植到目标板。

(4) 根据系统需求，编写 Bootloader，并移植设备驱动程序。研究多功能电能表协议，电力负荷管理系统数据传输规约，根据需求编写基于 uClinux 的系统应用程序。

(5) 完成了配变监测终端设计与调试，制作了样机，如附录一图。样机调试、运行试挂情况表明，各项技术指标均达到设计要求。

6.2 研究展望

由于嵌入式操作系统 Linux 是近几年才发展起来的新事物，它为 IT 业注入了新的活力，成为后 PC 时代的标志之一。但从嵌入式 Linux 系统本身上看，应该在如下方面继续加强：首先，在实时性上，uClinux 要进一步的改进，向硬实时性方向发展；其次，在应用开发的模式上，要吸取 windows 开发平台的优点，例如易学性、标准统一的开发环境；最后，嵌入式 Linux 技术急需制定相关的标准，使其统一化、规范化，使嵌入式 Linux 发挥更大的潜能。在配变监测方面，仍有许多工作有待进一步研究和完善，对于本系统可以考虑采用使用更高速的处理器，操作系统采用更高、更稳定的内核版本。配变监测终端硬件部分可以考虑设计 USB 接口，方便系统软件升级。

参考文献

- [1] 马忠梅. AT91 系列 ARM 核微控制器结构与开发[M]. 北京: 北京航空航天大学出版社, 2002.
- [2] 王学龙. 嵌入式 Linux 系统设计与应用[M]. 北京: 清华大学出版社. 2001.
- [3] 中华人民共和国国家电网公司.Q/GDW 129-2005.电力负荷管理系统通用技术条件.北京:中国电力出版社 2006.
- [4] 中华人民共和国国家电网公司.Q/GDW 130-2005.电力负荷管理系统数据传输规约.北京:中国电力出版社 2006.
- [5] 中华人民共和国电力工业部.DL/T 645-1997.多功能电能表通信规约.北京:中国电力出版社 1998.
- [6] 张建国, 王晓春, 吴国良.配电自动化技术的发展及系统构成[J] 河北电力技术 1998 年第 6 期 27-30.
- [7] 谢大为, 杨晓忠.配变监控终端在配电自动化中的应用[J],现代电力 Vol No.6 Dec.2002: 70-72.
- [8] 金明. 配电变压器监控终端的开发[D]. [硕士学位论文]. 合肥:合肥工业大学. 2004.
- [9] 刘健, 倪建立等.配电自动化系统[M].北京:中国水利电力出版社,1999.
- [10] 黄清波. 基于 AT91M40800 的嵌入式工业控制器设计[D]. [硕士学位论文]. 浙江: 浙江大学信息科学与工程学院. 2003.
- [11] 田泽.嵌入式系统开发与应用[M].北京:北京航空航天大学出版社,2005:599-605.
- [12] 杨光. 基于 ARM 的嵌入式系统软件平台的研究[D].[硕士学位论文].成都: 西南交通大学计算机学院.2006.
- [13] 宋振宇. 基于 uClinux 的嵌入式系统系统的构建与应用[D]:[硕士学位论文]. 大连:大连理工大学计算机学院. 2004.
- [14] 周立功, 陈明计等. ARM 嵌入式 Linux 系统构建与驱动开发范例[M]. 北京:北京航空航天大学出版社, 2006: 42-56.
- [15] 毛德操,胡希明. Linux 内核源代码情景分析(上册)[M]. 浙江:浙江大学出版社. 2005.10
- [16] 张记凯. 基于 PL3201 的多功能电能表设计[D]:[硕士学位论文]. 北京:北方交通大学社,2007.
- [17] 谭浩强. C 程序设计[M],清华大学出版社,1999.
- [18] 阎石. 数字电子技术基础[M],北京:高等教育出版社,1998.

- [19] 康华光, 陈大钦等.电子技术基础模拟部分[M],北京: 高等教育出版社, 1999.
- [20] 陈丽君. Linux 操作系统内核分析[M].北京:人民邮电出版社,2000.
- [21] 高志强,李业德. 基于 MSP430 单片机的配变远程监测终端的设计[J]. 山东理工大学学报(自然科学版) Mar.2007, Vol.21 No.2:84-87.
- [22] 黄胜,王斌,江修. 基于 ATT7022B 的配变检测仪的设计[J].电脑与信息技术. Apr.2007, Vol.15 No.2:13-14.
- [23] 林俊辉, 岑旭.基于 GPRS 远程监控的配变实时监测及无功补偿系统[J].福建电力与电工.2005.9:51-54.
- [24] 张建.电能量远程采集系统终端的研究[D]:[硕士学位论文].合肥:合肥工业大学,2006.
- [25] 肖鲲, 李晓明, 董丽娟.电能量远程数据采集系统的设计与实现[J].电力系统及其自动化学报.2002,14(2):19-22.
- [26] 魏永静. 基于 ARM 的电力负荷管理终端的设计[D]:[硕士学位论文].山东:山东大学电气学院.2006.
- [27] 珠海炬力.ATT7022B 用户手册[R].炬力集成电路设计有限公司,2005.3.
- [28] 王振华,张学平.电磁兼容技术在 PCB 抗干扰设计中的应用[J].中国无线电, 2006, 2: 60-63.
- [29] 陈晓民.一种新型配变监测装的设计与实现[J].湖南电力.2006.6:37-39.
- [30] 孙成宝, 李广泽.配电网实用技术[M].北京:中国水利水电出版社, 1998.
- [31] 诸小鹏.基于 80C196 的配变监测终端的设计[J].淮阴工学院学报, Jun,2004,Vol.13 No.3:48-53.
- [32] 张速.配电综合监测即无功动补系统开发研究[D]:[硕士学位论文].山东:山东大学电气学院.2005.
- [33] 曹建平.电力系统配电变压器监测终端的研究与实现[D]:[硕士学位论文].南京:南京理工大学.2006.6-7.
- [34] 顾樱华, 顾春艳.采用 DSP 技术的电能质量在线监测[J].电测与仪表, 2004, 3: 14-16.
- [35] 宋士通.配电自动化终端设备及其应用[J].山东理工大学学报(自然科学版), 2003, 7: 97-100.
- [36] 路灿等.一种多功能配变监测与动态无功补偿综合控制器[J].电工技术杂志, 2003, 4: 860.
- [37] 刘晨.配电自动化系统的通信方案设计[J].电力通信系统, 2003, 10: 16-18.
- [38] 蔡锐丹, 叶力勤.基于移动通信网络的配电参数远程监控系统[J].华北电力技术, 2004, 4: 33-35.
- [39] 王金富, 刘平, 高胜法.基于 RS-485 总线的数据采集与控制系统[J].自动化仪表, 2003, 24 (9): 57-60.

- [40] L.Cristaldi,A.Ferrero,S.Salicone. A Distributed System for Electric Power Quality Measurement[J]. IEEE Transactions on Instrumentation and Measurement. 2002, 51(4): 776-780.
- [41] Etienne Moulin.Measuring Reactive Power in Energy Meters[J].Metering International. 2002,10(1):489-490.
- [42] Daniel Bover.Marco Cesati, Andy Oram.Understanding the Linux Kernel[M]. www.oreilly.com/catalog/linuxkernel/, 1999.9.
- [43] What is uClinux.<http://www.uclinux.org/description/>
- [44] ATMEL. AT91RM40800 Data Sheet.Atmel Corporation[R], 2004.
- [45] Alessandro Rubini,Jonath Corbet.Linux device drivers.2nd Edition[M].America:O'Reilly &Associates,2001.
- [46] Rui Wang,Shiyuan Yang.The design of a rapid prototype platform for ARM based embeded systm.Consumer Electronics,IEEE Transaction on, Volume 50,Issue 2,May 2004:746-751.
- [47] Ke Cui,Zhen Wu.Reseach and implementation of remote monitoring system based on real-time uClinux.Services Systems and Services Management,2005.Proceedings of ICSSM'05.2005 International Conference on Volume 2,13-15 June 2005: 1182-1187 vol.
- [48] McGranaghan Mark.Trends in Power Quality Monitoring.IEEE Power Engineering Review,2001,21(110):3-10.

攻读硕士学位期间发表的论文及参加的项目

发表的论文

- [1] 师娟娟, 彭迪. 基于 ARM9 的嵌入式 Linux 移植[J]. 武汉理工大学学报(信息与管理工程版). 2008.2: 205-208.
- [2] 彭迪, 师娟娟. 基于 SH_Mobile 平台的 SD 卡存储方案的设计与实现[J]. 微计算机应用. 2008.4: 88-91.

参与的项目:

- [1] 2006. 9 至 2007. 10, 参与武汉理工大学与武汉中原电子有限公司合作项目基于 uClinux 的配变监测终端研发。
- [2] 2007. 3 至 2007. 10, 参与 ARM9 的电力负荷管理终端开发平台的建立。

致 谢

三年的硕士学习生活即将结束，论文课题也终于顺利完成，在此谨对所有关心、帮助和指导过我的亲人、老师、同学和朋友们致以诚挚的感谢。

感谢导师陈伟教授，他自始至终以极大的热情和严谨认真的态度给予作者指导和关心，在论文撰写的过程中，得到了陈伟老师的悉心指导，他仔细的修改作者的论文，并提出了许多宝贵的意见。他不仅教授了作者专业的知识，更传授了做人的道理。他广博的学识、一丝不苟的工作作风、严谨的治学态度都令作者终生难忘。

感谢信息学院黄秋元副教授，武汉中原电子信息技术有限公司，是他们提供了良好的实习环境、软硬件条件以及各种无私的指导和帮助。

同时还要感谢工程师张恒坊、何俊等同事，感谢你们对于我在完成课题和论文的过程中提出的宝贵意见和建议。

感谢我的父母和家人无私的养育之恩和细心的教导之心，使我得以健康地成长，父母正直的人格和顽强的敬业精神也是我学习和生活的楷模，如果没有他们就不会有今天的我。

感谢武汉理工大学通信技术研究所的所有同学，通信技术研究所是一个充满活力、积极上进的集体，在这个集体中，我学到了很多知识，取得很大的进步，度过了一段美好难忘的时光。

师娟娟

2008年4月

附录一：配变监测终端实物图

