

摘 要

随着信息技术不断进步和变电站自动化水平的不断提高,对数字式母线保护的通讯能力也提出了更高的要求。通讯系统作为保护装置的重要组成部分,一方面承担着监控整套保护装置的任务;另一方面是实现对外通讯联系的桥梁。论文首先给出了通讯系统的软硬件平台设计方案;然后详细阐述了主要接口电路的硬件设计;接着详细阐述了嵌入式操作系统的移植、各个通信接口驱动程序的编写;最后,给出了 103 规约在 RS-485 和以太网上的实现。研究设计的通讯系统已经应用于工程实际,丰富的通讯接口、标准的 103 规约、便捷的本地调试完全符合实际需求。

关键词: 母线保护, 通讯, 103 规约, 现场总线, 以太网

ABSTRACT

With the progress of information technology, it require higher demand for the communication ability of the busbar protection device. As an important part of the device, on the one hand communication system is in charge of supervising the entire equipment. On the other hand, the system is a bridge of communicating with outer terminal or automation system. First, a design scheme of the hardware platform and the software platform is presented in the paper. Second, the design for main interface circuits is stated in detail. Third ,The methods of transplanting RTOS and programinnng drivers for each interface are explained in detail. Finally,the realization of 103-protocol based on RS-485 and Ethernet is presented.The research have been put into real application and have good effects. Enrich communication interfaces, standard IEC 60870-5-103 and convenient locality debugging meet the practical demand completely.

Shi Meichuan(communication and information system)

Directed by prof. Xie Zhiyuan

KEY WORDS: busbar protection ,Communication , IEC 60870-103,field bus,ethernet

摘 要

随着信息技术不断进步和变电站自动化水平的不断提高,对数字式母线保护的通讯能力也提出了更高的要求。通讯系统作为保护装置的重要组成部分,一方面承担着监控整套保护装置的任务;另一方面是实现对外通讯联系的桥梁。论文首先给出了通讯系统的软硬件平台设计方案;然后详细阐述了主要接口电路的硬件设计;接着详细阐述了嵌入式操作系统的移植、各个通信接口驱动程序的编写;最后,给出了 103 规约在 RS-485 和以太网上的实现。研究设计的通讯系统已经应用于工程实际,丰富的通讯接口、标准的 103 规约、便捷的本地调试完全符合实际需求。

关键词: 母线保护, 通讯, 103 规约, 现场总线, 以太网

ABSTRACT

With the progress of information technology, it require higher demand for the communication ability of the busbar protection device. As an important part of the device, on the one hand communication system is in charge of supervising the entire equipment. On the other hand, the system is a bridge of communicating with outer terminal or automation system. First, a design scheme of the hardware platform and the software platform is presented in the paper. Second, the design for main interface circuits is stated in detail. Third ,The methods of transplanting RTOS and programinng drivers for each interface are explained in detail. Finally,the realization of 103-protocol based on RS-485 and Ethernet is presented.The research have been put into real application and have good effects. Enrich communication interfaces, standard IEC 60870-5-103 and convenient locality debugging meet the practical demand completely.

Shi Meichuan(communication and information system)

Directed by prof. Xie Zhiyuan

KEY WORDS: busbar protection ,Communication , IEC 60870-103,field bus,ethernet

声 明

本人郑重声明：此处所提交的硕士学位论文《数字式母线保护装置通讯系统研究》，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：石美佳 日 期：2009.3.5

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学校有权保管、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：石美佳

导师签名：谢志远

日 期：2009.3.5

日 期：2009.3.10

第一章 绪 论

1.1 课题背景及意义

母线是电力系统的重要组成部分，起着汇集和分配电流的作用，它上面连接着许多电力系统的重要设备，如变压器、发电机和输电线路、配电线路等。当母线发生故障时，如果不能快速、准确地切除故障，将对整个系统的运行产生严重影响。特别是随着我国电力系统的迅猛发展，系统容量迅速增加，75kV及其以上电压等级的超高压母线也逐渐投入运行，母线故障对电气设备的损害及对电力系统的稳定性影响越来越严重。因而，对母线保护的要求也越来越高。根据《继电保护和安全自动装置技术规程》规定，重要的35kV及其以上电压等级的发电厂或变电所的母线上，都需要装设专用的母线保护装置。

随着我国电力工业的发展和城乡电网改造的推进，对电网的安全、经济运行要求愈来愈高，变电站综合自动化系统、继电保护及故障信息系统正得到越来越广泛的应用。传统的具有单一保护功能的母线保护装置已经很难满足发展的需要，装置还必须提供提供事故追忆、故障记录、数据库的维护以及保护装置远方整定和维护等功能，而这些功能的实现更多的是依赖通讯，通讯接口已成为母线保护装置的基本组成部分。

同时，随着微机、通讯、信号处理等信息技术的进步，高性能的微处理器芯片层出不穷，母线保护装置的性能也不断得到改善。母线保护系统正向智能化、网络化、保护、控制、测量、数据通信一体化和标准化方向发展。由于保护装置底层功能的增强以及信息化的发展趋势，进而对通讯系统也提出了更高的要求。因此，根据母线故障特点和运行要求，借鉴相关微机保护的开发和运行经验，研制新一代母线保护装置的通讯系统，具有十分重要的现实意义。

通讯系统作为母线保护装置的重要组成部分，一方面承担着监控保护单元的任务；另一方面也是对外通讯联系的桥梁，其主要功能如下：

(1) 通过内部通讯对保护单元的工作情况及装置本身状态进行实时监测。

(2) 建立底层保护单元与 PC 机监控分析软件之间的通讯连接，实现对保护的操作、调试、以及获取装置数据信息、故障数据分析等功能。

(3) 通过外部通讯接口与其它系统连网通讯，实现变电站自动化系统或专用监控管理系统对保护装置的操作和监测。

1.2 母线保护装置及其通讯系统的发展现状

母线保护的发展经历了电磁型、晶体管型、集成电路型到微机型几个发展阶段。其中，电磁型、晶体管型母线保护已逐步被淘汰。目前，母线保护正由集成电路型逐步向微机型过渡。

数字式母线保护，也就是微机型母线保护，是采用以微处理器为核心的软硬件模块来完成数据采集、数据处理、保护功能算法、开关量逻辑判断、出口跳闸逻辑等功能。数字式母线保护最主要特点是计算能力强，能方便地实现各种保护原理；抗CT饱和能力强；具有自适应能力，可识别母线运行方式。同时数字式母线保护具有自检功能，可以进一步提高可靠性。而且，数字式母线保护装置具有强大的通讯能力，可以满足变电站自动化系统对继电保护装置的要求，与变电站自动化系统(或监控系统)互联，完成信息的远传与远控，与其它保护、控制装置和调度联网以共享全系统数据、信息和网络资源，实现自动化。

数字式母线保护中，由于需要进行采样的电流通道很多，另外为避免元器件损坏引起的误动，往往采用多个CPU来共同实现继电保护的功能，装置内部多个CPU之间仅靠串口进行通讯，其通讯速度和可靠性在通讯手段高速发展的今天，面临着巨大的挑战。

目前，我国电力系统现场运行的各厂家的母线保护装置对外通信接口不统一，有RS485/RS422/RS232标准串口、CAN或LonWorks现场总线、光纤通讯接口、以太网口等。采用的通讯协议也不统一，有的采用厂商自定义规约，有的采用103通信规约。由于保护装置提供的对外通讯接口单一、速率低、采用的通讯规约不标准，造成与变电站自动化系统或专用监控管理系统连接困难，不能及时地上传有关信息和报文，无法满足变电站自动化系统的要求。

现有母线保护装置的现场调试一般通过 RS-232 实现本地 PC 机连接调试，RS-232 通讯速率较低，不支持热插拔，影响了故障分析和装置的调试运行工作。

1.3 本文研究的意义和主要内容

1.3.1 本文研究的意义

在充分了解当前数字式母线保护的发展现状以及数字式母线保护对通讯接口类型、数据传输速度、通信协议等方面的要求的基础上，本文拟开发设计一个用于数字母线保护的通用通信系统，这样可以大大缩短母线保护装置的开发周期，并实现系列化和标准化，便于运行部门的运行维护。在对外通信的设置上，此通信系统要提供多个对外通信接口，尽可能实现用户的可选配置（例如RS-485和以太网等网络标准的可选配置），以满足不同的变电站监控系统的要求，通讯协议采用目前应用最广泛的103规约。同时，为了方便装置的现场安装和调试，要提供一个与PC机

高速且便利的接口。具备即插即用特性的USB (Universal Serial Bus, 通用串行总线)技术以其高速、灵活和便于使用的特点,可以成为一种新的良好解决方案。

1.3.2 本文研究的主要内容

论文首先分析了几种常用的网络通信方式,确定了本系统的硬件总体设计方案和软件总体设计方案。在此基础上,根据嵌入式系统的开发方法,详细介绍了硬件电路的设计、嵌入式操作系统的移植、各个通信接口驱动程序的编写以及 IEC 60870-103 规约在 RS-485 和以太网上的实现。

论文的具体章节安排如下:

第一章 对数字式母线保护通讯系统做了功能分析,并对发展现状进行了综述,明确了论文的主要工作。

第二章 概述了保护装置的情况,给出了其通信系统的软硬件总体设计方案。

第三章 详细介绍了硬件系统主要模块具体实现方法,并给出了各单元的原理图。

第四章 详细介绍了嵌入式操作系统的移植过程,并给出了各个通讯接口驱动程序设计思想及其层次结构,结合部分源代码及流程图,详细介绍了设计方法。

第五章 103 通讯规约的实现。详细研究 IEC 60870-5-103 规约的应用规约数据单元、传输帧格式、链路传输规则和应用层功能,并将其与 RS-485 结合,设计了实现 103 规约的间隔单元端的通讯程序;将 103 规约与 TCP/IP 协议相结合,设计了一种能实现 103 规约的应用层功能,又能在以太网上传输数据的网络通信协议。

第六章 总结

第二章 保护装置概况和通讯系统方案设计

2.1 保护装置概况

2.1.1 保护装置的设计原则

数字保护装置的设计借鉴了以往微机保护装置的开发经验，充分考虑了运行现场对装置的保护功能、通信能力、状态监测、故障记录与分析、装置本身的调试维护等要求，在装置的设计中，主要遵循了以下基本设计原则：

(1) 采用多个CPU组成的并行运行系统结构

采用多个CPU并行智能处理技术，一个CPU专门负责通信，一个CPU完成保护功能，一个CPU负责启动、录波，保护出口采用“启动+出口”的方式，可以防止因CPU损坏造成的保护误动作，CPU与出口信号具有彼此独立的电源，具有较强的抗干扰能力。

(2) 采用高性能硬件平台

CPU插件主处理器采用32位浮点Power PC体系微处理器，数字信号处理器为ADI公司的16位DSP，数据转换器为16位并行高速A/D转换器。

(3) 基于嵌入式操作系统的软件设计

嵌入式操作系统使得应用程序的设计和扩展变得容易，不需要大的改动就可以增加新的功能。软件按任务进行模块化设计，每个任务的调试、修改几乎不影响其他任务。按这种方法设计的软件实时性强、易于维护，提高了系统的稳定性和可靠性。

(4) 丰富的通讯接口、完整的数据记录与分析功能

该数字式母线保护装置除了具有常规的保护功能以外，还加入对外通信、调试管理、状态监测、启动录波、数据存储管理等诸多辅助功能，使得整个系统功能更加强大，智能化程度更高。

2.1.2 保护装置的硬件结构

保护装置由多个插件组成：

(1) 交流插件、开入插件、信号插件、电源插件、出口插件

这些插件主要实现模拟量、开入量的采集、装置供电及出口等功能。

(2) CPU插件

本装置共有2个CPU插件，硬件完全相同，一个完成保护功能，另一个完成故障启动及故障录波、驱动保护出口电源功能。

(3) 通信管理插件

通信管理插件，是母线保护装置与使用者或外界沟通信息的桥梁，一方面担负着保护装置对外进行通讯的任务；另一方面起着监控保护装置的作用。其主要功能包括：通过装置内部的CAN总线收集CPU板的事故或事件报告，存储故障记录，并通过对外通信接口送至监控系统或变电站自动化系统；通过调试接口与外部PC机通信，完成定值整定，数据分析，装置调试等功能。

其功能如图2-1所示。

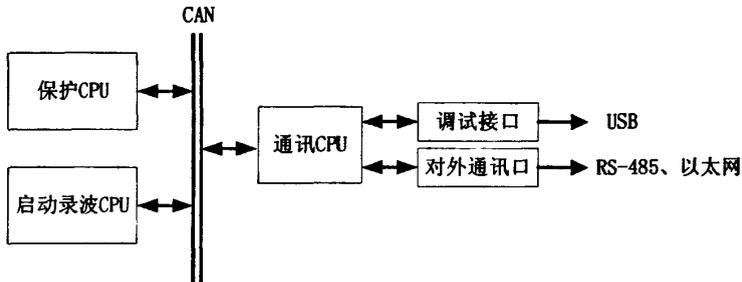


图 2-1 通讯系统的功能图

2.2 几种常用的通信方式

通讯系统应提供多种通讯方式，以方便地接入变电站自动化系统或监控系统。目前，现场运行的母线保护装置采用的通讯方式多种多样，常用的通讯方式主要有RS-485、现场总线方式（CAN总线、LonWorks总线）、以太网等方式。

2.2.1 串行通信

常用的串行通信接口的电气标准有EIA RS-232、RS-422、RS-485。

其中RS-232只能用于点对点的通信，且传输距离短、速率低，因此不适合一对多微机保护的通信。

RS-422A是为弥补RS-232的不足而提出的，RS-422A定义了一种平衡通信接口，最大传输速率为1Mb/s，最大传输距离为400英尺，并允许在一条平衡线上最多连接10个接收器。RS-422是一种单机发送、多机接收的单向、平衡传输规范。

RS-485与RS-422A一样，也是一种平衡传输方式的串行接口标准，它和RS-422A兼容，扩展了RS-422A的功能，增加了多点、双向通信能力，即允许多个发送器连接到同一条总线上，同时增加了发送器的驱动能力和冲突保护特性，扩展了总线共

模范围。在平衡电缆上最多连接 32 个收发器/接收器对。RS-485 标准抗干扰能力强，传输速率高，传输距离远。采用双绞线，不用 Modem 的情况下，在 100kb/s 时，可传送的距离为 1.2km，若速率降到 9600bit/s，则传送距离可达 15km，它允许的最大速率可达 10Mb/s（传送 15m）。

目前电力系统中，微机保护装置、微机监控装置和其他微机型的自控装置间的通讯，很多都是通过 RS-485 通讯实现的。RS-485 接线简单、可靠性高、成本低。物理层采用 RS-485 串行通信链路的继电保护设备信息接口配套标准（IEC60870-5-103）在电力系统中应用非常广泛。

2.2.2 现场总线通讯

上世纪 80 年代出现了现场总线，并制定了相应的标准。比较著名的有：德国 Siemens 公司制定的过程现场总线 ProfiBus（Process Field Bus）标准、由现场总线基金会提出的基金会现场总线 FF（Field bus Foundation）、美国 EcheLon 公司推出的局部操作网络 LonWorks（Local Operating Network）、德国 Bosch 公司推出的 CAN（Controller Area Network）总线等。根据国际电工委员会 IEC 和美国仪表协会 ISA 的定义，现场总线是连接智能现场设备和自动化系统的数字式、双向传输、多分支结构的通信网络，它的关键标志是能支持双向、多节点、总线式的全数字通信。它按国际标准化组织 ISO 和开放系统互连 OSI 提供网络服务，可靠性高、稳定性好、抗干扰能力强、通信速率快、造价低、维护成本低。

在众多的现场总线中，CAN 总线具有通信速率高、可靠性高、实现方式简单、性价比高等诸多优点。CAN 总线（Controller Area Network）是控制器局域网的简称，是 20 世纪 80 年代德国 BOSCH 公司为解决现代汽车中众多的控制与测试仪器之间的数据交换而开发的一种串行数据通信协议，是目前国际上应用最广泛的现场总线之一，被广泛用于自动化和工业控制领域。由于其采用了许多新技术及独特的设计，与一般通信总线相比，CAN 总线在可靠性、实时性和灵活性方面具有很多技术优势，主要特点如下：

(1) CAN 总线属于多主总线，网络上任一节点均可在任意时刻主动的向网络上其它节点发送信息，而不分主从，通信方式灵活。

(2) CAN 总线上的节点信息分成不同的优先级，可满足不同的实时要求。

(3) CAN 总线采用非破坏性总线仲裁技术，当多个节点同时向总线发送信息时，优先级较低的节点会主动地退出发送，而最高优先级的节点可不受影响地继续传输数据，从而大大节省了总线冲突仲裁的时间。尤其是在网络负载很重的情况下也不会出现网络瘫痪的情况。

(4) CAN 总线只需通过报文滤波即可实现点对点，一对多点及全局广播等方式

传送数据，无需专门的“调度”。

(5) CAN总线的通信速率最高可达 1Mb/s(此时通信距离最远为 40m)，通信距离最远可达 10km(速率 5kb/s以下)。

(6) CAN总线上的节点数主要取决于总线驱动电路，例如，当使用 Philips P82C250 作为收发器时，同一网络中允许挂接 110 个节点。报文标识符可达 2048 种(CAN2.0A)，而扩展标准(CAN2.0B)的报文标识符几乎不受限制。

(7) CAN总线的的数据采用短帧结构，数据段最多为 8 个字节，不会占用总线时间过长、保证了通信的可靠性。

(8) CAN总线的每帧信息都有CRC校验及其他检错措施，这些特性增强了CAN总线的抗干扰能力。

(9) CAN总线的通信介质可为双绞线、同轴电缆或光纤，选择灵活。

(10) CAN节点在错误严重的情况下具有关闭输出功能，可以使总线上其它节点的操作不受影响。

2.2.3 以太网通讯

以太网是由 DEC、Intel 和 Xerox 三家公司在 20 世纪 70 年代开发研制的，采用总线拓扑结构，媒体访问方式为载波监听/冲突检测(CSMA/CD)的多路访问控制协议，不支持带优先级的实时访问。以太网由于其开放性好、应用广泛以及价格低廉等特点，基本垄断了商业领域的网络市场，在自动控制领域也得到了大规模的应用。目前许多大公司的自动化系统都是采用以太网来实现管理层的通讯。以太网及 TCP/IP 随着控制技术与信息技术的结合进入到电力系统中，将是今后变电站自动化系统通信的发展趋势。其优势主要体现在以下几个方面：

(1) 网络的通讯速率高

以太网支持的数据传输速率包括 10Mb/s, 100Mb/s 和 1Gb/s，比目前任何一种现场总线都快，即使用于超大规模系统也游刃有余。

(2) 开放性好

以太网是目前应用最广泛的局域网技术，遵循国际标准规范 IEEE 802.3，受到广泛的技术支持，采用以太网作为控制网络，可以保证有多种开发工具供选择。基于 TCP/IP 协议的以太网是一种标准的开放式网络，不同厂商的设备很容易互联，并可以方便的与广域网相连。这种特性非常适合于解决控制系统中不同厂商设备的兼容和互操作的问题。

(3) 支持多种的物理介质和拓扑结构

以太网可使用同轴电缆、双绞线、光纤等通信介质，甚至能在一个网络中混合使用，用户可根据带宽、距离、价格等因素作多种选择。网络拓扑结构可为总线型或星型，同样也可两者混用。

(4) 成本和费用低廉

由于以太网的应用较为广泛，受到了硬件开发商和生产厂商的高度重视与广泛支持，因此可以有多种硬件产品供选择；与目前众多的现场总线相比，价格也低廉得多。现有的大量资源可以极大地降低以太网系统的开发和维护费用，进而可有效降低系统的成本。

2.2.4 本地调试通讯

目前微机型继电保护装置的调试接口大都是 RS-232 接口，RS-232 通讯速率慢，当数据量较大时很难满足实时性要求。另外，RS-232 不支持热插拔，给现场的连接调试带来很多不便。

USB 是英文 Universal Serial Bus(通用串行总线)的缩写，是 1994 年底由 IBM、Intel 及 Microsoft 等几家公司共同开发的 PC 外设接口标准，由于其具有使用方便及成本低廉等优点，迅速得到业界的强力支持，目前已经成为主流的总线标准。其应用范围不断扩大，工业自动化仪器仪表正在成为 USB 应用的热点之一。USB 与 RS-232 接口相比主要有以下优点：

(1) 使用方便、即插即用

所有的 USB 设备可以随时接入主机系统，USB 主机能够动态地识别设备的状态，并自动地给接入的设备分配地址和配置参数。

(2) 连接灵活、易扩展

USB 接口具有很强的可扩展性，一个 USB 接口理论上可连接 127 个外设。连接方式也十分灵活，既可以使用串行连接，也可以使用集线器把多个设备连接在一起。

(3) 速度快

USB1.1 协议支持的低速模式的通讯速率为 1.5Mb/s，全速模式的通讯速率可达 12Mb/s。USB2.0 协议支持的传输速率最高可达 480Mb/s。不同的模式适用于不同的应用场合。

(4) 使用灵活

USB 有四种传输类型：控制传输、同步传输、中断传输和批量传输。可以适应不同设备对传输速率、响应时间等方面的不同需求。

目前绝大多数流行的操作系统均支持 USB 技术，这些操作系统内部自带 USB 驱

动程序(包括总线驱动程序和主机控制器驱动程序),因此,开发者在设计程序时,只须编写与应用程序通信的设备驱动程序,这给开发人员带来了极大的方便。

2.3 通信系统硬件设计方案

硬件平台是整个通讯系统的基础部分,是软件平台的载体。良好的硬件平台可以为软件的运行提供强有力的支撑,能够最大限度的发挥软件的性能,同时也便于以后软件的升级。

本通讯系统设计目的是为了用一个通用通讯系统来实现当前和未来几年数字式母线保护装置与不同的变电站自动化系统或监控系统的互连,同时还应满足装置现场安装调试简单方便的要求。

根据对当前国内外各大公司,例如:南瑞继保公司、烟台东方电子、国电南自,许继、ABB, SIMENS, ALSTOM, SEL等现有产品在通讯接口层面技术水平的分析和未来几年内发展走向的预计,我们认为在对外通讯接口上将是传统的RS-232/RS-485/RS-422等串口和以太网长期并存的局面,对任何一种接口方式的过分强调和推崇都是有失偏颇的。所以我们研制的通讯系统必须具备RS-485通讯口和以太网接口。

装置内部采用CAN总线作为通讯网络,将通讯单元和承担保护功能及启动录波功能的单元共同挂在CAN总线上,通过CAN总线进行高速数据信息的传递,通讯单元统一对信息进行管理,完成远方、本地与保护装置的交互。CAN总线具有高速率、高实时性、高可靠性、高抗干扰性的特点,非常适合在变电站这种电磁干扰大、工作环境恶劣、对数据实时性要求高的环境下工作。

USB接口以其传输速度快、支持热插拔、价格低廉以及良好的兼容性已经成为计算机的主流接口。本系统拟配备一个USB接口,通过与PC机上驱动程序和应用软件配合,可组成一个便利高速的USB接口调试系统,弥补传统调试系统的不足,极大地方便了装置的安装调试。

硬件平台核心处理器选用MPC555,这是Motorola公司PowerPC 500系列的代表性产品,是一款高性能高速的32位处理器。其内部资源极其丰富,片内集成了带浮点运算单元的32位RCPU核、26kbytes静态RAM、448kbytes FLASH、多通道串行模块、两个CAN2.0B(TouCAN)控制器等,特别适于做复杂的实时控制和处理系统。对于本系统来说,能非常容易地实现RS-485接口、CAN接口,而且不需外扩程序存储器。

通讯单元的硬件系统主要由微处理器及外围电路、存储模块、CAN接口模块、RS485接口模块、USB接口模块、以太网接口模块、电源管理模块、复位和看门狗

电路等几个部分构成，其电路结构如图 2-2 所示。

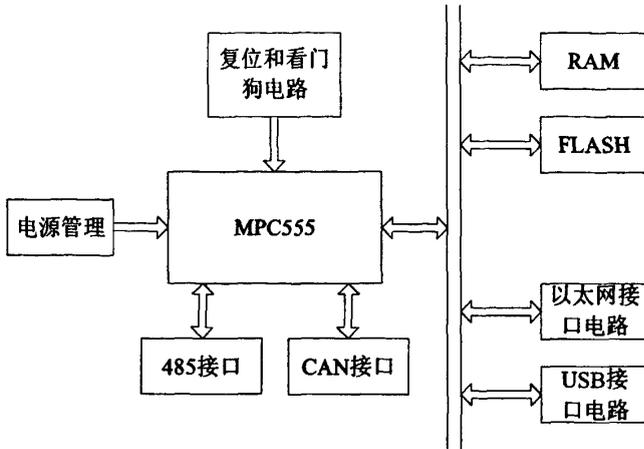


图 2-2 硬件结构框图

2.4 通讯系统软件设计方案

2.4.1 基于嵌入式实时操作系统的软件设计方案

软件是一套微机保护装置的核心和灵魂，各种原理的保护功能都是直接由软件来实现的，此外装置状态监测、故障记录与分析、对外通信等功能也依赖软件来实现。因此，软件设计好坏直接影响着保护装置的性能指标。

传统的微机保护软件通常采用前后台系统设计方法，后台系统是一个无限循环，循环中调用相应的函数完成相应的功能；前台系统也就是中断服务程序，用来处理异步事件。这种软件系统是单任务的顺序程序，整个软件即是一个任务，它始终独占系统的各种资源。处理器严格按照程序中所规定的顺序执行，程序的运行结果仅与初始条件有关而与执行速度无关。这种机制具有简单直观、易于控制的优点。然而，由于程序只能按照单一线索顺序执行，缺乏灵活性，实时性无法保证，在复杂系统中难以胜任。同时，由于使用的 CPU 资源较少，任务比较简单，往往采用汇编语言编写程序，汇编语言虽然简洁高效、占用空间少、运行速度快，但编写和调试都比较困难，不便于升级维护，因而开发效率和可移植性都较差。

如今，随着 32 位嵌入式微处理器为核心的嵌入式应用系统迅速推广普及，软件的规模逐步扩大，系统要求更强的数据处理能力、控制能力和更好的实时性能。嵌入式实时操作系统可以为软件开发提供一个高效和方便的平台，基于嵌入式实时操作系统的用 C 语言编程的软件设计方法已成为必然趋势。

嵌入式实时操作系统是事件驱动的能对外界的作用和信号在限定的时间内作出响应的软件平台，它具有较强的实时性、可靠性和灵活性。在此平台上可以有

个并行的程序在运行，每个运行的程序称之为任务。嵌入式实时操作系统为多任务创建运行环境，并完成任务的调度、通信、同步，以及内存管理、中断服务等功能。系统中每项任务都是一个无限循环的程序，等待特定的事件，然后执行相应的任务。嵌入式实时操作系统的使用，使得实时应用程序的设计和扩展变得容易，不需要大的改动就可以增加新的功能。复杂的应用程序可以分解为若干独立的任务，每个任务的调试修改几乎不影响其它模块。基于嵌入式实时操作系统的C语言程序具有极大的可移植性，优秀的嵌入式实时操作系统上跨处理器平台的程序移植只需要修改1~5%的内容。不仅如此，嵌入式实时操作系统还是一个可靠性和可信度很高的实时内核，从根本上保证了软件的可靠性和实时性。

2.4.2 嵌入式实时操作系统的选择

在嵌入式系统的开发中，嵌入式实时操作系统的选择对产品能否研制成功起着至关重要的作用。如果选择不当，会导致研制周期变长，研制成本增加，甚至会影响产品的性能指标。因此，在项目的方案初步设计阶段，首先要在详细调研、进行充分论证的基础上，选择合适的实时操作系统。

目前国际主流的嵌入式实时操作系统有：Nucleus、VxWorks、Windows CE、pSOS、Linux、 μ C/OS-II等。

综合操作系统选型的多方面因素，我们决定采用Accelerated Technology公司推出的新一代嵌入式操作系统Nucleus。选择Nucleus主要是基于以下几点来考虑的：

Nucleus提供注释严格的C源代码给每一个用户。用户能够了解底层内核的运作方式，可根据自己的特殊要求删减或改动系统软件，而且产品在批量生产时也不必支付许可证费用。用户可以完全控制内核而不必担心操作系统中可能会存在异常任务导致系统崩溃。

Nucleus可移植性好。Nucleus 95%的代码使用ANSI C编写的，只有与处理器相关的代码才使用汇编，硬件相关和硬件无关的代码是相对独立互不干扰的。这就使得它的可移植性较强，可以方便的移植到目标机上去。

Nucleus的内核很精巧，在典型的目标环境中，Nucleus的核心代码一般不超过20k字节大小。这个特点使得对于内核的彻底分析较为容易，相比之下，Linux的内核要复杂得多，要在有限的时间内深入透彻的分析困难较大。

Nucleus支持大部分流行CPU，如X86, 68xxx, 68HCxx, NECV25, ColdFire, MIPS, SPARClite, ARM6/7, StrongARM, PowerPC等。针对不同的CPU类型，Nucleus还提供编译器、动态连接器、多任务调试器等相应的工具来配置用户的开发环境，方便软件的开发和调试。

Nucleus的核心是一个实时的多任务内核——Nucleus PLUS，任务的调度采用基于优先级的抢占式调度算法，同时还采用时间片来实现同一优先级的任务公平合理地共享处理器，可以满足母线保护在实时性方面的要求。

2.4.3 通信系统软件的构成

通信系统的软件由三部分构成：（1）Nucleus PLUS 嵌入式实时操作系统内核；（2）硬件初始化代码及设备驱动程序；（3）对应应用任务的应用程序。其中，应用程序又由若干个中断服务子程序和任务程序模块组成。详见第四章和第五章。

2.5 本章小结

本章首先简要介绍了数字式母线保护装置的构成，然后介绍了微机保护装置中常用的几种通信方式，接着给出了通信系统的硬件和软件总体设计方案。

第三章 硬件电路设计

通讯单元的硬件系统主要由微处理器及外围电路、存储模块、CAN接口模块、RS485接口模块、USB接口模块、以太网接口模块、电源管理模块、复位和看门狗电路等几个部分构成，本章将详细阐述各模块的电路设计。

3.1 存储模块电路设计

考虑到本通信系统要存储大量的事件记录、操作记录等信息，并且这些信息在掉电后不能丢失，因此在系统设计中外扩了8M字节的FLASH。芯片选用的是AMD公司的MBM29DL64DF。该芯片有64Mbit（ $4M \times 16\text{bit}/8M \times 8\text{bit}$ ）空间大小，数据宽度可以配置为8位（字节）或16位（字）；单3.3V工作电源即可实现读、写、擦除等操作，无需另外提供擦除和写入电源。

另外，考虑到通讯单元要支持多个通信接口和多种通信协议，运行时需要很大的数据空间，又扩展了大容量的RAM。RAM选用的芯片是IS61LV25616，该芯片是256k \times 16bit的3.3V供电的高速异步静态RAM。图3-1中给出了存储模块和MPC555的连接关系。

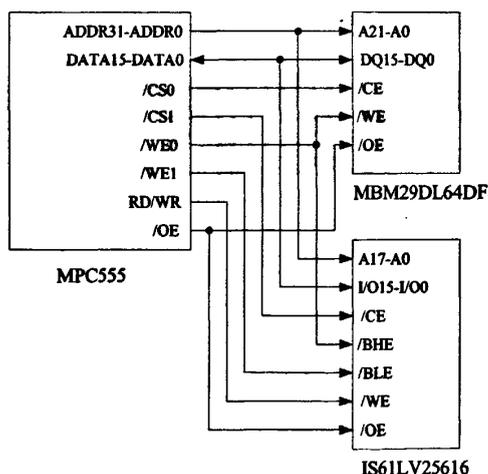


图 3-1 存储模块原理框图

3.2 RS-485 接口电路设计

实现RS-485 串口通信主要需要完成两部分工作：(1)实现RS-485电平和TTL/CMOS电平的转换；(2)接收并校验串行数据，将串行数据变成并行数据并提供给处理器处理；将处理器送来的并行数据变成串行数据，并加上校验码发送出去。实现电平转换可以用专门的接口芯片实现，考虑到系统的工作电压为5V，所以选用

了具有 5V 工作电源的 TTL/RS485 电平转换芯片--MAX485ESA。实现串并转换通常由通用异收发器 UART(Universal Asynchronous Receiver and Transmitter)来完成。MPC555 有两个 UART，分别是 SCI1 和 SCI2，在设计时选用了 SCI2。

RS485 通信接口相对来说比较简单，但是要注意接口芯片的保护问题，瞬态过电压和静电都可能造成接口损坏，尤其是在节点多、工作距离远的情况下。为此采取了以下措施：

1. 接口芯片选择了静电保护、过热保护、输入失效保护等保护措施较完善的 MAX485ESA；
2. AB 两线分别接 10k 的上下拉电阻；
3. AB 两线各对地接一瞬态抑制二极管；
4. AB 两线均串接一 250V/120UA 的自恢复保险丝。
5. 如果是线路终端，还要加接一个 120 欧的终端电阻。

具体电路图如图 3-2 所示。

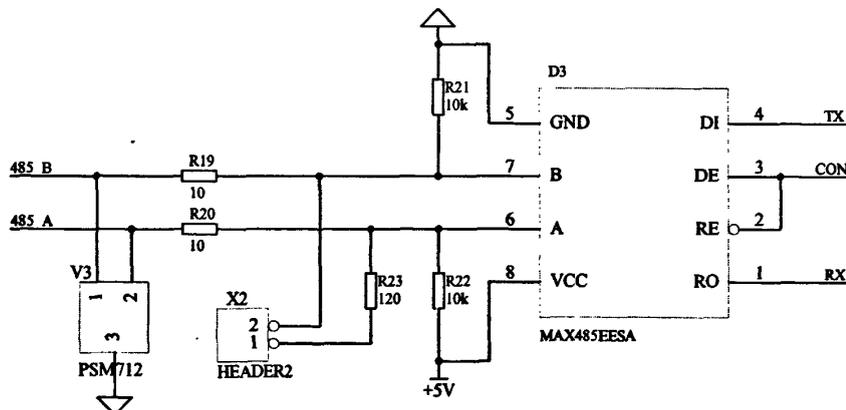


图 3-2 RS-485 接口电路图

3.3 以太网接口电路设计

以太网接口器件选用 Cirrus Logic 公司生产的 16 位 10Mb/s 的全双工以太网控制器 CS8900A，该芯片为工业级产品，内部带有编码/解码器、时钟恢复电路、10BASE-T 收发器和完整的 AUI 接口前置模拟电路等。网络接口 10BASE-T 收发器可以直接和隔离变压器相连，通过 RJ-45 连接各类屏蔽或非屏蔽双绞线，并支持接收极性倒置自动检测和校正。

CS8900A 主要功能模块包括：802.3 介质访问引擎 (MAC)，ISA 总线接口，4kB 的片上 RAM，串行 EEPROM 接口。CS8900A 通过总线与 MPC555 连接，接收 MPC555 发送到其内部 RAM 的数据，然后按 IEEE 802.3 标准格式发送出去；同时，CS8900A

从以太网上接收数据并缓存于 RAM 中，通过中断通知 MPC555 处理，从而实现通信功能。

CS8900 与 MPC555 采用 16 位 I/O 方式连接，具体电路如图 3-3 所示：

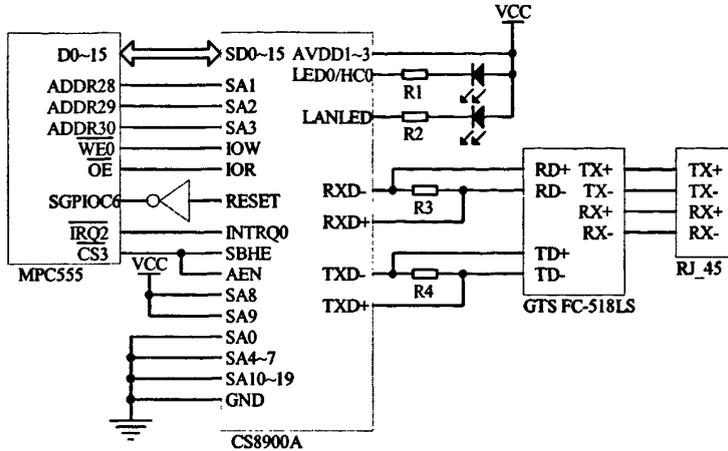


图 3-3 以太网接口电路

CS8900A 的 16 根数据线直接与 MPC555 的数据总线相连,CS8900A 的地址线 SA1、SA2、SA3 与 MPC555 的 ADDR28、ADDR29、ADDR30 相连, SA8 和 SA9 置高电平, 其它数据线置低电平, 这样在复位后 I/O 模式的基地址为 0x300。CS8900A 的读写控制连到 MPC555 的 $\overline{WE0}$ 和 \overline{OE} 管脚, RESET 连到 MPC555 的通用 I/O 口 SGPIOC6 上, 中断请求连接到 MPC555 的 $\overline{IRQ2}$ 上, MPC555 的 $\overline{CS3}$ 连接 CS8900A 的片选线, 这样就完成了 CS8900A 与 MPC555 间的接线。CS8900A 的 LED0 和 LANLED 在连接电阻和发光二极管后接到 VCC 上, 实现相应操作的指示功能。CS8900A 的接收和发送端通过隔离变压器 GTS FC-518LS 连接到 RJ45 接头上, 实现与网络的连接。隔离变压器的作用是把 CS8900A 送过来的差分信号用差模耦合的线圈滤波以增强信号, 并通过电磁场的转换, 耦合到连接网线的 RJ45 接头上。这样实现了网线和接口之间没有物理上的连接而能传递信号, 隔断了信号中的直流分量, 还可以在不同工作电平的设备中传输数据。另外, 隔离变压器本身就是设计为耐压 2kV~3kV 电压的, 也起到了防雷击的作用。

3.4 CAN接口电路设计

CAN 遵从 OSI 模型, 按照 OSI 基准模型, CAN 协议划分为两层: 数据链路层和物理层。其中数据链路层又划分为逻辑链路控制 (LLC-Logic Link Control) 和媒体访问控制 (MAC-Media Access Control) 两个子层。物理层定义信号是如何实际地传输的, 因此涉及到位时间、位编码、同步的解释。MAC 子层是 CAN 协议的核心。它把接收到的报文提供给 LLC 子层, 并接收来自 LLC 子层的报文。MAC 子层负责报文分帧、仲裁、

应答、错误检测和标定。LLC子层涉及报文滤波、过载通知、以及恢复管理。

CAN总线的通信协议由CAN总线控制器实现。CAN总线控制器分为两类:一种是独立的CAN总线控制器;另一种是和微控制器结合的芯片。负责CAN总线控制器和物理传输线路之间接口的器件是CAN总线收发器,该器件用来提高差动发送和差动接收的能力。

由于我们选定的MPC555微处理器内部有两个TouCAN控制器,这两个控制器完全符合CAN2.0B协议,只需外加CAN收发器即可。收发器选用了Philips公司PCA82C250。

为了增强CAN节点的抗干扰能力, TouCAN控制器的CNTX和CNRX并不是直接与PCA82C250的TXD和RXD相连,而是通过高速光耦6N137隔离后再与PCA82C250相连,这样就很好的实现了总线上各个CAN节点间的电气隔离。特别说明的是光耦部分电路所采用的两个电源VCC和VDD必须完全隔离,否则采用光耦隔离也就失去了意义。电源的完全隔离我们采用了一个DC-DC模块DUP750S05来实现。这些虽然增加了节点的复杂程度,但却提高了节点的稳定性和安全性。

在PCA82C250与CAN总线的接口部分也采取了一些抗干扰措施。CANH和CANL与地之间分别并联了一个1000pF/2kV的电容,可以起到滤除总线上的高频干扰的作用,而且有一定的防电磁辐射的能力。另外,在两根CAN总线接入端与地之间分别接了一个瞬态电压抑制二极管(TVS),保护PCA82C250免受ESD和EMI浪涌脉冲。通信信号传输到导线的端点时会发生反射,反射信号会干扰正常信号的传输,因而在PCA82C250的两根输出数据线之间加上120欧的终端电阻以消除反射信号。

电路原理图如图3-4所示:

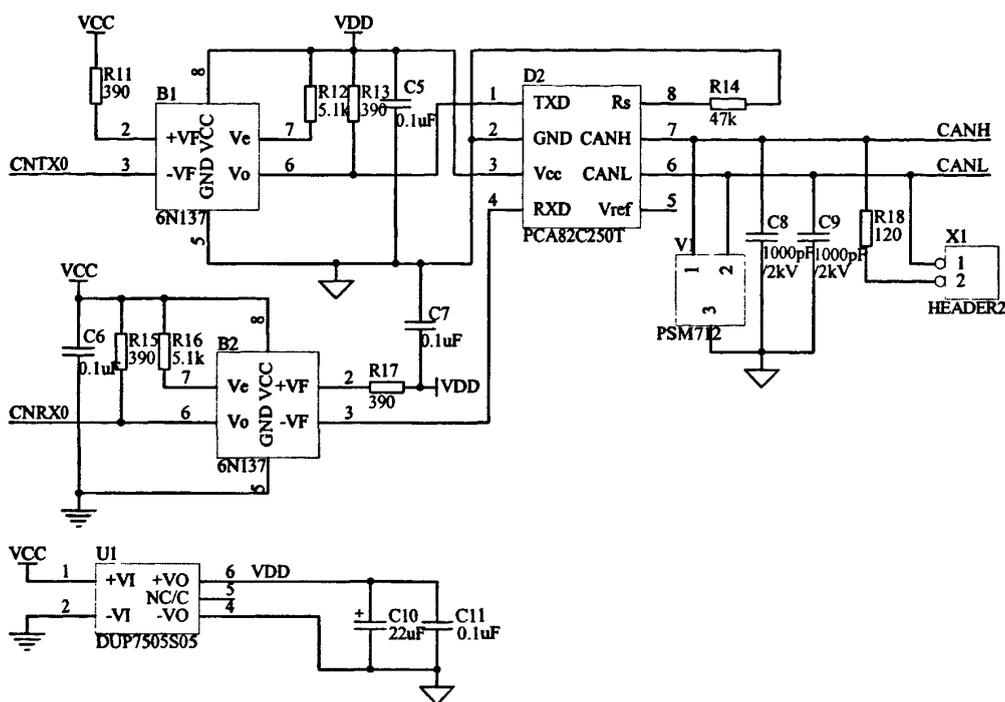


图 3-4 CAN接口电路图

3.5 USB接口电路设计

3.5.1 芯片选型

USB设备硬件设计的核心是USB控制器。USB控制器一般包括了USB SIE的基本功能以及USB数据收发功能。

市场上供应的USB控制器主要分为两类：专用USB控制器和通用USB控制器。专用USB控制器针对特定设备进行优化，基本不需开发者做特别的工作，如鼠标、键盘控制芯片等；通用USB控制器实现基本的USB协议，并为用户预留多个功能端点，可以满足不同的应用要求，显然我们选择后一种。

通用USB控制器按结构又可细分为带USB接口的处理器和通用USB接口器件两种。

(1) 带USB接口的处理器

这种方案的最大优点在于开发难度小，因为这样的处理器一般都是基于 8051 结构或其他常见的结构，同时有完备的开发辅助文档和代码实例，但开发一般需要专用的仿真器，价格也较高。

(2) 通用USB接口器件

通用USB接口器件仅处理USB总线相关的事务，必须有一个外部微处理器来进行协议处理和数据交换。这种方案的主要特点是成本低、可靠性高，具有很高的灵活性，缺点是开发者需要非常熟悉USB协议，以处理USB总线事务，同时要用微处理器控制接口器件的工作。

由于在我们的系统中已经有处理器，为节约成本，选择了通用USB接口器件，即Philips公司的PDIUSBD12。

3.5.2 USB接口芯片PDIUSBD12 介绍

PDIUSBD12是一款性价比很高的USB接口器件，完全符合USB1.1总线协议。PDIUSBD12所具有的低挂起功耗连同LazyClock输出可以满足使用ACPI、OnNOW和USB电源管理的要求。低的操作功耗可以应用于使用总线供电的外设。此外它还集成了许多特性，包括SoftConnect、GoodLink、可编程时钟输出、低频晶振和终止寄存器集合。所有这些特性都为系统显著节约了成本，同时使USB功能在外设上的应用变得容易。下面重点介绍PDIUSBD12芯片的四个特性：

- Philips串行接口引擎(SIE)

Philips SIE控制完成所有USB协议层的功能，这些功能完全由硬件来实现而不需要固件的参与。SIE模块主要完成以下功能：同步方式的识别，并串转换，位填充/解除填充，CRC校验/产生，分组标识(PID)校验/产生，地址识别和握手评估/产生。

- SoftConnet

与USB的连接是通过 $1.5k\Omega$ 上拉电阻将D+置为高实现的。 $1.5k\Omega$ 上拉电阻集成在PDIUSBD12片内，默认状态下不与VCC相连。连接的建立通过系统微控制器发送命令来实现。这就允许系统微控制器在决定与USB建立连接之前完成初始化时序。USB总线连接可以重新初始化而不需要拔出电缆。

- GoodLink

GoodLink技术可提供良好的USB连接指示，在枚举过程中，LED指示根据通信的状况间歇闪烁；当PDIUSBD12成功地枚举和配置后，LED指示将一直点亮一定时间，随后与PDIUSBD12之间成功的传输(带应答)将关闭LED；处于挂起状态时，LED将会关闭。该特性为USB器件、集线器和USB通信状态提供了用户友好的指示，作为一个诊断工具它对隔离故障的设备是很有用的。

- 存储器管理单元(MMU)和集成RAM

存储器管理单元MMU和集成RAM作为USB传输速度和微控制器或DMA控制器读写速度之间的缓冲区，这就允许微控制器以自己的速率对USB信息包进行读写。

题，所有微处理器均需要有一个复位逻辑，它负责将微控制器初始化为某个确定的状态。这个复位逻辑需要一个复位信号才能工作。这个复位信号的稳定性和可靠性对微控制器的正常工作有重大影响。另外，在程序受到干扰跑飞后，也需要一个复位信号，以使系统能够自动恢复正常运行。因此，在通讯系统硬件上设计了复位和看门狗电路。

在设计时，选择了专用的微处理电源监控芯片MAX705，该芯片可在系统上电、掉电以及供电电压降低时产生复位输出，并且还具具有手动复位输入和看门狗电路输出功能。

具体电路设计如图 3-6 所示。MAX705 的看门狗触发输入引脚WDI接MPC555 的I/O 口上，若WDI保持 1.6s没有输入，则MAX705 的看门狗输出引脚 \overline{WDO} 就产生一个复位信号来复位MPC555 的运行。MAX705 的复位输出引脚 \overline{RESET} 接到MPC555 的上电复位引脚 $\overline{PORESET}$ 上。这样，在系统上电时，MPC555 就可以接收到一个可靠的复位信号。

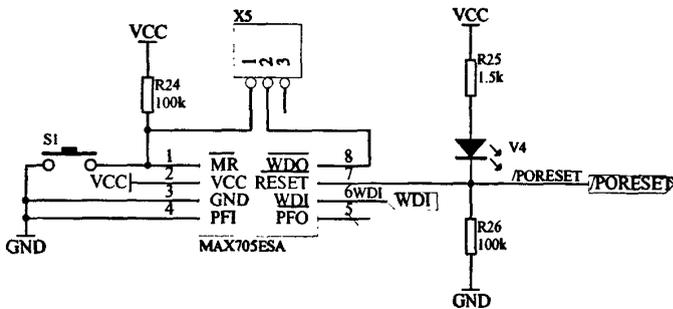


图 3-6 复位和看门狗电路

3.7 本章小结

本章具体的介绍了系统电路主要组成部分的设计内容，包括存储模块设计、各个通信接口电路设计、复位看门狗电路设计。给出了具体的参数指标，并对电路进行了分析和介绍。

第四章 Nucleus操作系统移植和驱动程序设计

4.1 Nucleus PLUS概述

Nucleus PLUS内核使用一种叫做软件组件的方法。所谓软件组件，就是具有明确目的或用途的由一个或数个C语言/汇编源文件组成的一段非独立执行应用程序，每个组件都提供一个定义好的外部接口，必须通过外部接口来使用组件。访问组件内的全局数据结构是被禁止的。使用了组件方法，软件组件既易于更新也易于重用。Nucleus Plus软件组件通常由定义数据类型、常量以及外部接口的头文件和一个或多个C语言/汇编源文件组成。Nucleus Plus提供如下组件：任务控制管理、内存管理、定时器管理、中断、系统诊断、I/O 驱动管理等共16个组件。

在Nucleus PLUS中，任务之间可以按照优先级和时间片方式来共享CPU资源，通过邮箱(mailbox)，队列(queue)和管道(pipe)进行通信，任务之间的同步和互斥可以通过信号量(semaphore)、事件组(event group)和信号(signal)来实现。Nucleus Plus提供动态和分区内存(dynamic/partition memory)两种存储器管理机制。Nucleus PLUS还提供定时器(timer)来处理周期性事件和任务的睡眠和挂起超时。Nucleus PLUS为每一个软件组件提供了一系列的系统调用，任务与Nucleus PLUS的交互是在系统调用的界面上进行的。

Nucleus PLUS内核的系统结构图如图4-1所示。

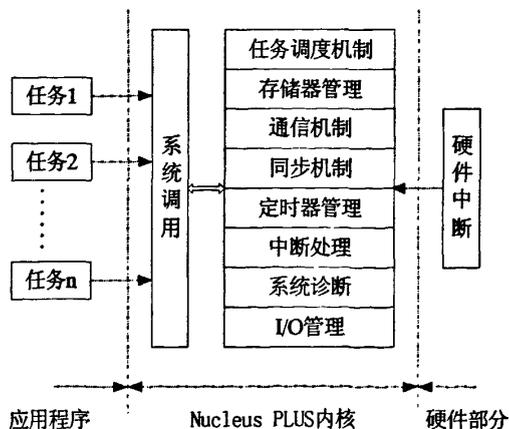


图 4-1 Nucleus PLUS系统结构

从图4-1可以看出，利用Nucleus Plus开发软件，只需编写任务(Task)代码和中断服务程序代码(ISR)。在任务代码(Task)和中断服务程序代码(ISR)中利用系统调用实现与Nucleus PLUS的交互，由Nucleus PLUS来调度多个任务并行执行，实现

CPU共享。

操作系统按照不同的任务对时间要求的不同来分配CPU资源，任务的调度采用基于优先级的抢占式调度算法，即系统中的每个任务都拥有一个优先级，任一时刻，内核将CPU分配给处于就绪状态的优先级最高的任务运行。系统内核一旦发现有一个优先级比当前正在运行的任务的优先级高的任务转变为就绪状态，内核立即保存当前任务的上下文，当前任务状态变为挂起，并且切换到这个高优先级的任务的上下文执行，当这个优先级较高的任务执行完后，原来被挂起的任务将被重新唤醒继续执行。同时，也采用时间片来实现同一优先级的任务公平合理地共享处理器。只要任务和中断服务程序的优先级和时间片设置得当，系统的实时性就能保障。

4.2 Nucleus PLUS的移植

4.2.1 系统启动和开发步骤

可以根据目标板的需要对Nucleus Plus进行裁剪和定制，生成适合目标系统的操作系统，尽量少占系统内存。ATI公司提供了一个名为PLUS.BAT的批处理文件，用户可以删除其中的模块、更改相关的选项或开关，然后执行该文件来生成与目标相关的操作系统，也就是根据系统需要生成定制的操作系统。

在Nucleus PLUS内核系统中，INT_Initialize是最先执行的函数，上电复位中断向量应该指向该函数的地址。INT_Initialize负责所有与目标系统硬件相关的初始化，包括建立中断向量、初始化栈指针、设置时钟频率、配置系统控制寄存器、分配存储空间及配置外围接口等。当INT_Initialize执行完后，将控制交给系统高级初始化函数INC_Initialize。

INC_Initialize调用各个组件的初始化函数。当所有系统组件初始化完成之后，INC_Initialize调用应用程序初始化函数Application_Initialize。

Application_Initialize是Nucleus内核对用户提供的-一个外部接口。用户通过自己编写该函数实体来完成应用程序环境的详细初始化。它包括任务、邮箱、队列、管道、事件集、内存池等的创建，中断的注册以及应用程序的初始化。

在Application_Initialize完成用户自定义初始化后，控制返回到INC_Initialize函数并调用TCT_Schedule，进入Nucleus Plus的任务调度循环。

Nucleus PLUS系统的启动过程如图4-2所示。

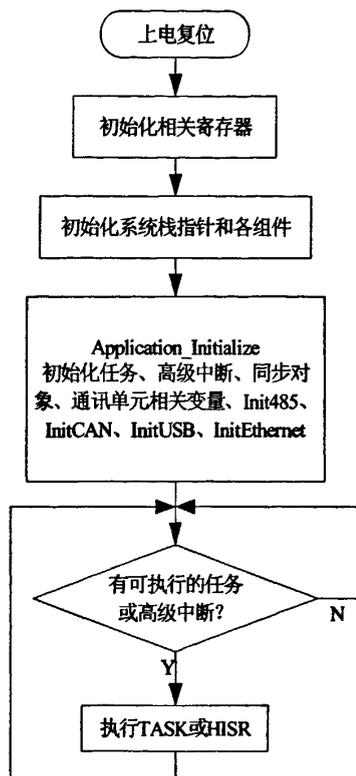


图4-2 系统启动过程

用Nucleus Plus开发嵌入式应用程序一般遵循如下步骤：

1. 修改INT.S文件，使之符合目标板的需要。该文件主要是一些与处理器相关的设置，包括对CPU异常向量表的设置，堆栈的初始化，CPU寄存器的初始化和中断设置等。

2. 为了提高系统的执行效率，在高级初始化函数INC_Initialize()中去掉目标系统中用不到的特征数据结构的初始化。

3. 定义Application_Initialize函数。包括创建内存池、任务、信号量、邮箱及高级中断HISR等。

4. 定义应用程序的任务。因为应用程序需要调用Nucleus Plus提供的服务，要把头文件NUCLEUS.H包含进去。

5. 编译/汇编所有的应用程序软件，并且包含系统低级初始化文件INT.S。

6. 链接所有的应用程序目标文件、Nucleus Plus库文件以及相关的开发工具库，生成可执行的目标代码。

7. 把生成的最终目标代码下载或烧写到目标板上执行。

4.2.2 开发注意事项

嵌入式开发在内存需求、任务管理等方面存在着特殊性，因此开发中对以下方面要特别加以注意：

- 任务优先级的分配要合理，否则会导致系统中的一些任务一直不断地在运行而另一些任务永远得不到运行，引起饥荒。

- 系统资源的分配要合理，否则会引起两个或多个任务为竞争某一系统资源而处于无限挂起现象。

- 内存泄露。内存分配过程中要尽量避免碎片并且要注意及时回收。

4.2.3 本项目中应用层软件构成

本项目中应用层软件由多个中断服务程序、高级中断服务程序和任务组成。

中断服务程序包括：RS485 通讯中断服务程序、以太网通讯中断服务程序、CAN 通讯中断服务程序、USB 通讯中断服务程序。高级中断服务程序包括：Hisr_485, Hisr_net, Hisr_usb, Hisr_can。任务的划分以功能为依据，包括处理以太网以及 TCP/IP 协议的任务 Task_NET、处理装置内部 CAN 通讯的任务 Task_CAN、处理 RS-485 通信的任务 Task_485、处理 USB 通讯的任务 Task_USB、系统自检任务 Task_CHECK。

其中，中断服务程序主要实现各个通讯接口数据的接收，为了避免中断服务程序占用 CPU 较多的处理时间，在接收到一个完整帧时，激活相应的高级中断，而由高级中断来激活任务，任务负责数据的处理、数据的发送以及通信协议的实现。

根据通讯系统要实现的功能可知，通讯单元要实现多种接口通讯，所以系统程序需要在多任务的环境下实现。在高性能的硬件平台下，通讯系统的实时性是否能够得到保障对软件的各个任务模块优先级的划分是关键。

表 4-1 任务优先级分配

任务	优先级	说明
Task_CAN	1	实现与保护CPU的通讯
Task_USB	2	实现装置的本地调试功能
Task_485	3	实现485接口的对外通讯功能
Task_NET	4	实现以太网接口的对外通讯功能
Task_CHECK	5	系统自检任务

通讯单元实现自身功能的前提条件是能够从保护单元及时可靠的获取数据信息，所以处理装置内部 CAN 通讯的任务 Task_CAN 的优先级最高为 1，这样可以保证通讯单元与保护单元进行 CAN 通讯时不会受其它任务的影响。

实现装置本地调试的任务 Task_USB 优先级为 2。实现对外通讯任务 Task_485、

任务 Task_NET 的优先级分别为 3、4。这三个任务都是采用高级中断触发机制运行的，任务平时任务处于挂起状态，当收到数据时由相应的接收中断调用系统函数 TCT_Activate_HISR(&HisrX) 来启动高级中断，高级中断服务程序负责任务的激活。在完成通信数据或协议处理后任务又调用系统函数 NU_Suspend_Task(&TaskX) 将自己挂起，等待下一次激活。

系统自检任务 Task_CHECK 优先级最低为 5，这是一个无限循环的任务，在创建后就激活运行了，主要完成硬件自检（包括 RAM、EEPROM 等，电源监视）、程序代码自检以及各种报告和扰动数据写入 flash 等功能，作为 CPU 空闲时循环执行的任务，随时可以被中断处理程序中断或者被优先级更高的任务抢占。

4.3 各个接口的驱动程序设计

4.3.1 RS-485 接口驱动程序

RS-485 接口由队列式串行多通道模块 QSMCM (queued serial multi-channel module) 中的串行接口 SCI2 实现，其驱动程序包括以下与硬件有关的操作：端口的初始化、数据发送和接收。

(1) 端口初始化：

端口初始化包括：设置通讯波特率、数据位、校验位、停止位；设置中断优先级，并开放相应级别的中断；设置中断屏蔽位并使能中断；最后使能接收器。

(2) 数据接收中断处理程序

在中断处理程序中，先读取 SCI2 状态寄存器，该寄存器包含指示 SCI2 运行状态的一些标志，通过读操作可清除这些标志。如果是接收中断，就读取 SCI2 数据寄存器并存入接收数据缓冲区，在接收到一个完整数据帧后屏蔽接收中断并启动高级中断处理。部分代码如下：

```
if(QSMCM.SC2SR.B.RDRF == 1)//为接收中断
{
    data=QSMCM.SC2DR.R;    //读取 SCI2 数据寄存器
    .....                //数据处理
    if(bFullFrame)        //判断是否接收到完整数据帧
    {
        QSMCM.SCC2R1.B.RIE = 0;        // 屏蔽接收中断
        TCT_Activate_HISR(&hisr_485);    // 启动高级中断处理
    }
}
```

```

}
}

```

(3) 数据发送子程序

发送数据时，先判断SCI2发送数据寄存器是否为空，然后将要发送的数据送入SCI2数据寄存器SC2DR，即可实现数据的发送。

4.3.2 以太网接口驱动程序

以太网通信处理主要包括链路层的以太网接口驱动程序，网络层的协议控制处理，传输层的协议控制处理三层。

由于所用的 Nucleus PLUS 只是操作系统内核，没包括网络协议的处理部分，所以移植了 lwIP 网络协议栈。

4.3.2.1 LwIP 介绍

LwIP 是一套用于嵌入式操作系统、开放源代码的 TCP/IP 协议栈。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运行。LwIP 协议的基础是在减少对硬件资源占用的前提下完成 TCP/IP 协议的主要功能。LwIP 的特性包括：支持多网络接口下的 IP 收发和转发，支持 ICMP 协议，支持 UDP 报文，支持 TCP 报文，具有超时时间估计、快速恢复和重发、窗口调整等功能。提供专门的内部回调接口 (Raw API) 用以提高应用程序性能等。LwIP 使用的进程模型是所有协议都驻留在一个单独的进程中，与操作系统内核是分离的。应用程序可以驻留在 LwIP 进程，也可以是分离的进程。LwIP 和应用程序之间的通讯，可以通过函数调用 (应用程序与 LwIP 共享进程时)，也可以通过更抽象的 API。这种与操作系统分开的进程模型可以方便在不同的操作系统上进行移植。LwIP 的进程模型如图 4-3 所示：

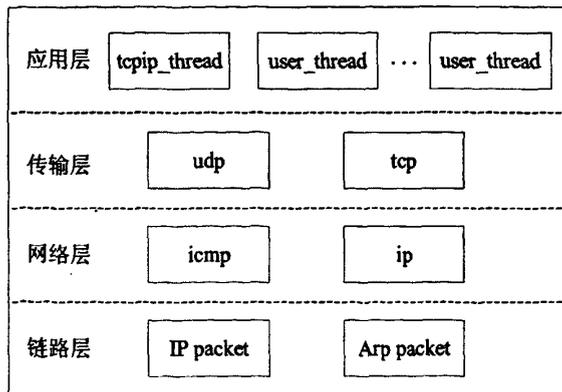


图 4-3 LwIP 的进程模型

LwIP 和 Nucleus PLUS 操作系统以及硬件板卡构成的系统架构如图 4-4 所示。本文移植的是 LwIP1.2.0 版本。

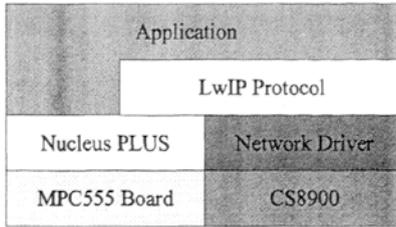


图 4-4 系统架构

4.3.2.2 操作系统模拟层说明

LwIP 协议在设计时就考虑了将来的移植问题，为了适应不同的操作系统，没有在代码中使用与任何操作系统相关的系统调用或数据结构，而是在 LwIP 和操作系统之间增加了一个操作系统的模拟层，该模拟层主要完成系统的初始化、任务的同步、时间管理和内存管理的功能。完成 LwIP 在 Nucleus PLUS 操作系统上的移植就是要调用 Nucleus PLUS 的系统函数实现上述几个部分的功能。

操作系统模拟层(sys_arch)在底层操作系统和 LwIP 之间提供了一般意义上的接口。主要目的是在移植 LwIP 到一个新的目标系统时，只需对一些新添加的头文件和一个新添加的操作系统模拟层(sys_arch)文件做一些简单的修改即可。当然不依赖底层操作系统的支持也可以实现这个操作系统模拟层(sys_arch)文件。

操作系统模拟层 sys_arch 需要给 LwIP 提供信号量(semaphore)和邮箱(mailboxe)两种通信机制。如果要得到 LwIP 的完整功能，还需要在 sys_arch 中实现多线程，但这对于只需实现 LwIP 的基本功能来说，不是必需的。除了上述提到的实现外，操作系统模拟层还需要提供几个头文件，在这个几个头文件中定义 LwIP 使用到的宏。这些文件和相应的宏定义稍后说明。

信号量既可以是计数信号量，也可以是二值信号量，这两种方式 LwIP 都可以正常工作。邮箱用于消息传递，用户既可以将其实现为一个队列，允许多条消息投递到这个邮箱，也可以每次只允许投递一个消息。这两种方式 LwIP 都可以正常运行。但是多条消息投递的邮箱可以使 LwIP 的运行更加有效。值得注意的是，投递到邮箱中的消息只能是一个指针。

信号量用“sys_sem_t”这种类型表示，在 sys-arch.h 文件中已经指定了。邮箱，同样道理，用“sys_mbox_t”类型来表示。具体如何来实现信号量和邮箱，LwIP 没有任何限制。

4.3.2.3 LwIP 移植的具体实现

1、与 CPU 或编译器相关的头文件

在 LwIP 源码的/src/arch/include/arch 目录下, cc.h、cpu.h、perf.h 中有一些与 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 Nucleus PLUS 相关环境的参数一致。通常 C 语言的结构体 struct 是 4 字节对齐的, 但是在处理数据包的时候, LwIP 是通过结构体中不同数据的长度来读取相应的数据的, 所以一定要在定义结构体 struct 的时候使用 `_packed` 关键字, 让编译器放弃结构体 struct 的 4 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需要使用 `PACK_STRUCT_XXX` 宏, 在移植的时候添加编译器所对应的 `_packed` 关键字。

2、sys_arch 操作系统相关部分

`sys_arch.c` 和 `sys_arch.h` 中的内容是与操作系统相关的一些结构和函数。现在, 要做的就是利用 Nucleus PLUS 操作系统的函数完成 LwIP 中信号量、消息队列、定时器和线程创建等功能的封装。

(1) `sys_sem_t` 信号量操作函数的实现

在 `sys_arch` 中应实现信号量结构体和处理函数:

```
--struct sys_sem_t
```

```
--sys_sem_t sys_sem_new(u8_t count)//建立并返回一个新的信号量, 参数 count 指定信号量的初始状态。
```

```
--void sys_sem_free(sys_sem_t sem)// 释放信号量。
```

```
--void sys_sem_signal(sys_sem_t sem)// 发送一个信号量。
```

```
--u32_t sys_arch_sem_wait(sys_sem_t sem, u32_t timeout)//等待指定的信号时阻塞线程。如果“timeout”参数值非 0, 线程只阻塞规定的时间(单位为毫秒)。在 timeout 参数值非 0 的情况下, 返回值为等待指定的信号所消耗的毫秒数。如果在指定的时间内并没有收到信号, 返回值为 SYS_ARCH_TIMEOUT(0xFFFFFFFF)。
```

由于 Nucleus PLUS 已经实现了信号量 `NU_SEMAPHORE` 结构及各种操作, 并且功能和 LwIP 上面几个函数的目的功能基本一样, 所以只要把 Nucleus PLUS 的系统函数重新包装成上面的函数, 就可以直接使用了。

(2) `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。

```
--struct sys_mbox_t
```

```
--sys_mbox_t sys_mbox_new(void)// 建立一个新邮箱。
```

```
--void sys_mbox_free(sys_mbox_t mbox)// 释放一个邮箱
```

```
--void sys_mbox_post(sys_mbox_t mbox, void *msg)//投递消息“msg”  
到指定的邮箱 mbox。
```

```
--u32_t sys_arch_mbox_fetch(sys_mbox_t mbox, void **msg; u32_t  
timeout)//阻塞线程直至邮箱收到至少一条消息。最长阻塞时间不能超过  
“timeout”毫秒。msg 是一个结果参数，用来保存邮箱中的消息指针，“msg”参  
数有可能为空，这表明当前这条消息应该被丢弃。返回值与 sys_arch_sem_wait()  
函数相同：SYS_ARCH_TIMEOUT 或者等待的毫秒数。
```

由于 Nucleus PLUS 已经实现了消息队列 NU_QUEUE 结构及其操作，并且功能和 LwIP 上面几个函数的目的功能基本一样，而且支持多个消息的投递，完全满足需求，所以只要把 Nucleus PLUS 的系统函数重新包装成上面的函数，就可以直接使用了。

(3) 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。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，所要实现的是如下函数：

```
--struct sys_timeouts *sys_arch_timeouts(void)//返回一个指向当前线程  
使用的 sys_timeouts 结构的指针。这个函数由 LwIP 的超时调度程序调用，并且不  
能返回空值(NULL)。单线程 sys_arch 实现中，这个函数只需简单返回一个指向保  
存在 sys_arch 模块中的 sys_timeouts 全局变量的指针即可。
```

(4) 与多线程相关的函数

LwIP 可以单线程运行，也可以多线程运行。单线程运行，即只有一个 tcpip 线程(tcpip_thread)，负责处理所有的 tcp/ucp 连接。单线程方式有优点也有缺点，优点是应用程序和 TCP/IP 协议栈驻留在同一个进程中，发送和接收数据时不必产生进程切换。缺点是程序代码晦涩难懂容易理解，而且一旦其它优先级高的任务开始执行，只要 TCP/IP 处理流程稍微被延迟，接收就会被阻塞，直接造成频繁丢包、响应不及时等问题。

本课题要实现的对外通讯相对复杂，为提高运行效率，降低编程复杂度，LwIP 以多线程方式运行，那么，就要实现下面的函数：

```
--sys_thread_t sys_thread_new(void(*thread)(void *arg), void *arg, int  
prio)//创建一个由函数指针“thread”指定的新线程，优先级由“prio”指定。  
“arg”将作为参数传递给 thread()函数。返回值为这个新线程的指针。
```

在 Nucleus PLUS 中，没有线程(thread)的概念，只有任务(Task)的概念。并且提供了创建新任务的系统函数 NU_Create_Task，因此把 NU_Create_Task 封装一下，就可以符合 sys_thread_new 的功能需求。

```
--sys_prot_t sys_arch_protect(void)
```

这个函数的功能是完成临界区域保护并返回先前的保护状态。该函数只在非常小的临界区域需要保护时才会被调用。支持 ISR 驱动的嵌入式系统可以通过禁止中断来实现这个函数。基于任务的系统可以通过互斥量和禁止任务来实现这个函数。该函数应该支持来自于同一个任务或中断的递归调用。换句话说，当该区域已经被保护，sys_arch_protect()函数依然能被调用。这时，函数的返回值会通知调用者该区域已经被保护。

Nucleus PLUS 属于支持 ISR 驱动的嵌入式系统，可以直接调用函数禁止中断来实现这个函数，具体实现如下：

```
sys_prot_t sys_arch_protect(void)
{
    return NU_Control_Interrupts(NU_DISABLE_INTERRUPTS);
}

--void sys_arch_unprotect(sys_prot_t pval)
```

这个可选函数的功能就是恢复受保护区域的先前保护状态，先前是受到保护还是没有受到保护由参数 pval 指定。

Nucleus PLUS 属于支持 ISR 驱动的嵌入式系统，可以直接调用函数禁止中断来实现这个函数，具体实现如下：

```
void sys_arch_unprotect(sys_prot_t pval)
{
    NU_Control_Interrupts(pval);
}
```

3、网络设备驱动程序的设计

在 LwIP 中可以有多个网络接口，每个网络接口都对应一个 struct netif，这个 netif 包含了相应网络接口的属性、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。在网络设备驱动程序中主要做的就是实现网络接口 CS8900A 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：

```
--err_t cs8900if_init(struct netif *netif) // CS8900A 初始化函数
```

CS8900A 初始化函数包括 CS8900A 芯片的复位和初始化内部各寄存器，arp 初始化，开启 ARP 表的定时更新处理。设置网络接口名称，设置网络物理地址，设置网络最大传输字节 MTU，允许处理网络广播通信；向 LwIP 注册数据发送函数 cs8900if_output，这样，当网络需要发送数据时，LwIP 就会自动调用 cs8900if_output 函数发送数据。

```
--static void cs8900if_input(struct netif *netif)// CS8900A 接收函数，  
从 CS8900A 接收以太网数据包并把其中的 IP 报文向 IP 层发送
```

```
--static err_t cs8900if_output(struct netif *netif, struct pbuf *p,  
struct ip_addr *ipaddr) // CS8900A 发送函数
```

如果系统需要发送数据，将数据通过 LwIP 协议栈处理后，自动调用 CS8900A 发送函数，具体就是给 IP 层传过来的 IP 报文加上以太网包头并并通过 CS8900A 发送出去。

```
--void cs8900_isr(void) // CS8900A 中断处理函数 ISR
```

当 CS8900A 接收到数据包时触发，在这个处理函数中启动高级中断处理函数 hisr_eth_comm，在 hisr_eth_comm 函数中用信号量通知接收任务进行数据接收并把数据提交给 LwIP 协议栈进行处理。

4、任务分配

专门建立一个接收任务来处理网络数据包的接收，接收到数据包后去掉以太网包头得到 IP 包，然后直接调用 tcpip_input() 函数将其投递到 mbox 邮箱。然后接收任务继续下一个数据包的接收，而被投递的 IP 包将由 tcpip_thread 线程继续处理。这样即使某个 IP 包的处理时间过长也不会造成频繁丢包现象的发生。这样更符合 LwIP 的设计原则，并且工作效率更高。此外接收任务的优先级应该较高，不至于使数据丢失，但系统繁忙时可能需要更多的内存。CS8900 中断和接收任务的关系如图 4-5 所示。

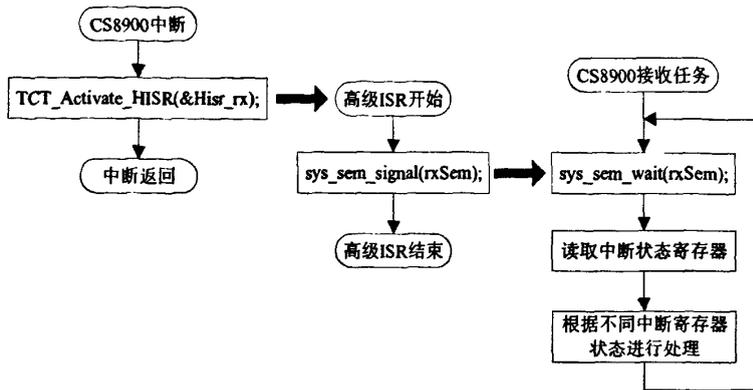


图 4-5 CS8900 中断和接收任务的关系

5、其它相关内容

文件 lwipopts.h 中包含 LwIP 的相关设置，如分配给 LwIP 的堆的空间大小、TCP 最大报文长度、TCP 接收窗口大小等，要根据实际情况来更改相关设置或选项。

4.3.3 CAN接口驱动程序

CAN 接口通讯处理主要由控制器模块（TouCAN）完成，TouCAN 模块包含 16 个消息缓冲器，用于实现发送和接收功能。也包含信息过滤器，当比较接收到的信息标识符和接收缓冲器标识符时用于验证这些标识符。CAN 接口驱动程序主要完成控制器模块的初始化和数据的接收、发送和中断处理功能。

(1) TouCAN 模块初始化

这部分程序主要是设置中断优先级、开放想要的中断、与位定时相关的设置（包括：同步段、传播段、相位缓冲段 1、相位缓冲段 2 以及重同步跳转宽度所占的时间长度，分频系数）、以及设置信号滤波器的屏蔽码等。

(2) 中断和数据接收处理程序

在中断处理程序中，首先读取错误/状态寄存器 ESTAT，如果是出错中断或离线中断就调用相应的处理子程序，否则读取中断标志寄存器确定是哪个消息缓冲器收到数据而触发的接收中断，然后从相应的消息缓冲器中读取信息。

接收数据时，先读取消息缓冲区的控制/状态寄存器，确定缓冲区中的数据长度，然后根据长度来读取缓冲区中的数据。部分代码如下：

```

len=0xf&(CAN_A.MBUFF[num].SCR.R);
for(i=0;i<len;i++)
    *(buf+i)=CAN_A.MBUFF[num].DATA[i].R;
  
```

(3) 数据发送子程序

发送数据时，先将数据送入发送消息缓冲区，然后设置发送缓冲区控制/状态寄存器的数据长度，最后设置发送码完成数据发送。部分代码如下：

```
for(i=0;i<len;i++)
    CAN_A.MBUFF[num].DATA[i].R=*(buf+i);
CAN_A.MBUFF[num].SCR.B.LENGTH=len;
CAN_A.MBUFF[num].SCR.B.CODE=0x0C;
```

4.3.4 USB接口驱动程序

USB通信可以分为两类：一类是用来设置USB设备的配置；另一类是用来实现应用程序的通信。其中，设置USB设备的配置是指主机了解设备的功能并且准备与其进行数据交换的过程，大部分通信是发生在开机或者连接设备时用来检测的过程中，即设备枚举过程中。而应用程序的通信则是发生在主机与应用程序的数据交换，也就是执行设备原始设计的功能。

本设计中只给出设备枚举功能的实现代码。

4.3.4.1 驱动程序结构

为了使驱动软件可移植性强、易维护，综合考虑USB协议、PDIUSB D12硬件接线、Nucleus PLUS的组织结构等因素，决定采用分层的方式编写，主要分为四层：硬件提取层、命令接口层、协议层和应用层。分层结构如图4-6所示。

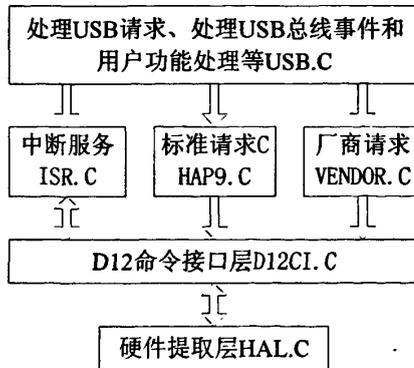


图 4-6 驱动软件分层结构图

(1) 硬件提取层

硬件提取层主要完成端口的初始化、数据或命令的发送以及数据的接收功能。硬件提取层只有一个文件HAL.C，提供对MPC555的I/O口、数据总线等硬件接口进行操作的函数，主要的函数列表如下：

表4-2 HAL. C中的主要函数

函数名称	功能
Void init_usb(void)	配置D12与MPC555的硬件接口
UINT8 inportb(UINT8 nPort)	从D12读入数据
void outportb(UINT8 nPort,UINT8 data)	向D12输出命令或数据

(2) 命令接口层

这一层定义了一套对D12芯片进行操作的命令接口函数，用来实现D12的功能，例如：地址使能、读取端点数据、把数据写入端点等等。主要是根据PDIUSBD12提供的命令(见附录0附表1)，调用硬件提取层的outportb()和inportb()这两个函数来实现。作为命令接口，这些函数提供给协议层和应用层程序调用。

(3) 协议层

协议层包括三个文件：ISR. C, CHAP9. C 和 VENDOR. C。

中断服务程序 ISR. C 设计的思想比较简单，就是根据中断源，调用相应的中断服务程序。当 ISR 收集了足够的数数据时，它通过高级中断来激活任务，任务检查事件标志以进入对应的处理子程序。

CHAP9. C 处理标准 USB 设备请求，VENDOR. C 处理厂商请求。

(4) 应用层

应用层定义了应用程序任务，它根据建立包的类型调用相应的协议处理程序来处理USB总线事务(如总线复位、总线挂起状态改变等)和实现用户功能(准备数据和数据处理)，每次处理完成后，把自己挂起。

4.3.4.2 USB 总线枚举过程

USB设备检测过程也称为USB总线枚举。

USB规范定义的设备状态共有6个，分别是连接、上电、默认、地址、配置和挂起。USB总线枚举过程中，USB设备需经历后4个状态。

USB集线器负责监测设备的连接情况，而USB主机会通过集线器轮流查询设备。当主机发现有一个新的设备时，会要求集线器建立与设备的通信通道，即设备的端点0(控制传输端点)。之后，主机会通过此通道向设备发送一系列的标准USB请求以获取所需的信息，设备必须响应这些请求并采取适当的动作。这个过程称为USB总线枚举。设备描述符包含的是设备信息，而USB请求是指USB主机和设备之间获取特定信息的格式，标准的USB请求有11个。

USB主机通过标准USB设备请求，设定和获取USB设备的有关信息，完成USB设备

的枚举。其过程如下：

1. 设备连接。USB设备接入USB总线。

2. 设备上电。USB设备可以利用USB总线供电，也可以使用外部电源供电，本系统中是外部电源供电。

3. 主机检测到设备，发出复位信号。设备连接到总线后，主机通过设备在总线上的上拉电阻检测到有新的设备连接，并获悉该设备是全速设备还是低速设备，然后主机向该端口发送一个复位信号。

4. 设备缺省状态。设备要从总线上接收到一个复位的信号后才可以对总线的处理操作做出响应，设备接收到复位信号后就使用缺省地址00H进行响应。

5. 地址分配。当主机接收到有设备对缺省地址的响应时，就发送设置地址的请求Set_Address对设备分配一个空闲地址，设备以后就只对该地址进行响应。

6. 主机发送获取描述符请求Get_Descriptor读取USB设备描述符、配置描述符，以确认USB设备的属性。

7. 查找驱动程序。在正确获取到所有描述符之后，主机开始查找驱动，如果未能搜索到驱动的话，主机系统会提示发现USB设备但未找到驱动。

8. 配置设备。当正确找到驱动后，主机依照读取的USB设备描述符来对USB设备进行配置，如果设备所需的资源得以满足，就发送Set_Configuration请求给USB设备，表示配置完成，这样才算是一个USB设备被正确枚举了。

4.4 本章小结

本章详细描述了基于MPC555硬件平台的Nucleus PLUS嵌入式操作系统的移植过程，同时给出了各个通信接口的驱动程序设计。

第五章 IEC 60870-5-103 网络通讯规约的实现

5.1 IEC 60870-5-103 规约介绍

IEC 60870-5-103 传输规约是国际电工委员会 TC57 技术委员会（电力系统控制和通信技术委员会）在 IEC 60870-5 系列基本标准的基础上所制定的可用于继电保护设备信息接口的配套标准，我国在等同采用 IEC 60870-5-103 标准的基础上，于 1999 年 10 月 1 日正式发布了电力行业相应的配套标准 DL/T 667-1999 规约。其目的是为了在变电站或厂站中，不同继电保护设备和控制系统之间交换信息，自推广以来在我国微机保护装置和监控系统中得到了广泛的应用，省去了不同厂家设备之间大量的规约转换工作，节省了人力物力，有利于不同的继电保护设备和控制系统之间互联。

103 规约的参考模型采用增强性能结构 EPA（Enhanced Performance Architecture），包括物理层、链路层与应用层 3 层。

物理层采用光纤系统或基于铜线的系统，它提供一个二进制对称和无记忆的传输。

链路层由一系列采用明确的链路规约控制信息（LPCI）的传输过程所组成，此链路规约控制信息可将一些应用服务数据单元（ASDUs）当作链路用户数据，链路层采用能保证所需的数据完整性、效率以及方便传输的帧格式的选集。

应用层包含一系列应用功能，它包含在源和目的之间应用服务数据单元的传输中。

5.1.1 IEC 60870-5-103 规约物理层说明

本规约既可以支持光纤系统，也可以是基于铜线的传输系统。在主控单元和继电保护设备之间可以采用基于铜线的传输系统。此种传输系统应符合 EIA RS-485 标准。由于 RS-485 标准的特性，在一个物理线路上最多能连接 32 个单元，实际应用中由于部分继电保护设备使用了转换头和现场电磁干扰的影响，最多所能接的单元数要少于 32。

5.1.2 IEC 60870-5-103 规约链路层说明

本规约采用非平衡传输。主控单元组成主站，继电保护设备为从站（子站）；即主控单元常常是始发站（启动站），继电保护设备常常是从动站。按照一主多从的严格查询方式进行通信。

传输速率可以为 9600bit/s 或 19.2kb/s。每个字符一位启动位，八位数据位，一位偶校验位，一位停止位。每个字符间无需线路空闲间隔，两帧之间的线路空闲间隔最少需 33 位。报文各字节在线路上传输顺序为低字节低位先发送。

当确认帧或响应帧受到干扰或超时没收到，则不改变帧计数位的状态重发原报文，重复帧传输的超时时间间隔为 50ms，最大重传次数为 3 次。

103 规约有 3 种帧格式：固定帧长帧格式、可变帧长帧格式、单个控制字符 E5H。最常用的两种格式为固定帧长帧格式、可变帧长帧格式。

表 5-1 固定帧长帧格式

启动字符 (10H)
控制域 (C)
地址域 (A)
帧校验和 (CS)
结束字符 (16H)

表 5-2 可变帧长帧格式

启动字符 (68H)	固定长度的 报文头
长度L	
长度L(重复)	
启动字符 (68H)	L个八位 位组
控制域 (C)	
地址域 (A)	
链路用户数据 (可变长度)	
帧校验和 (CS)	
结束字符 (16H)	

本规约采用的链路服务级别为 3 级：发送/无回答、发送/确认和请求/响应。发送/无回答由主控单元向继电保护设备发送广播命令；发送/确认由主控单元向保护设备设置参数等；请求/响应由主控单元向继电保护设备召唤数据或事件。

5.1.3 IEC 60870-5-103 规约应用层说明

本标准中链路规约数据单元 (LPDU) 不得包含多于一个的应用服务数据单元 (ASDU)。

如表 5-3 所示，一个应用服务数据单元是由一个数据单元标识符和唯一的一个信息体组成。

对于所有应用服务数据单元，其数据单元标识符有相同的结构，它由四个 8 位

位组所组成，它们是类型标识、可变结构限定词、传送原因和应用服务数据单元公共地址。

信息体由信息体标识和一组信息元素集组成，如果出现的话，还有时标。可变结构限定词规定了信息体中的信息元素集是由一个单个的信息元素、一个综合信息元素或一个顺序的信息元素组成。

信息元素、类型标识以及传送原因的定义信息可参考 IEC60870-5-103 传输规约中的相关内容。

表 5-3 一个应用服务数据单元的结构

应用服务数据单元	数据单元标识符	类型标识		数据单元类型	
		可变结构限定词			
		传送原因			
		应用服务数据单元公共地址			
	信息体	功能类型		信息体标识符	
		信息序号			
		信息元素集		信息体时标 (任选)	
		时标ms			
		IV	Res.		时标min
		SU	时标h		

103 规约中定义的基本应用功能包括初始化、总召唤、时钟同步、命令传送、扰动数据传输和通用分类服务等。

103 规约提供了两种信息交换方法：一种方法是基于严格规定的 ASDU 和标准化报文的传输应用过程，包括兼容范围和专用范围；另一种方法是使用通用分类服务，通用分类服务可以传输几乎所有可能的信息，但要求装置有较大的内存容量，要求传输通道有较高的传输速率。对于已定义的兼容范围的 ASDU 和应用过程要求各厂家强制性采用，对于所要求传输的信息，如果兼容范围还不能完全满足要求，对于国内已经生产的继电保护设备，在硬件方面改动比较困难，又较难适应通用分类服务的要求时，可使用专用范围实现，对于新的继电保护设备如果兼容范围还不能完全满足要求，不能满足要求的部分必须采用通用分类服务实现，不得使用专用范围。专用范围结合我国继电保护设备或间隔单元的情况，定义了大量保护相关信息对应的功能类型 (FUN) 与信息序号 (INF) 以及测控相关信息 (如遥测、遥信、步位置、

遥控和电度)的信息元素、ASDU与传输过程。

5.2 基于RS-485的103规约

5.2.1 103规约的软件实现

数字式保护装置对外通讯接口RS-485上采用103规约，程序流程图如图5-1所示。

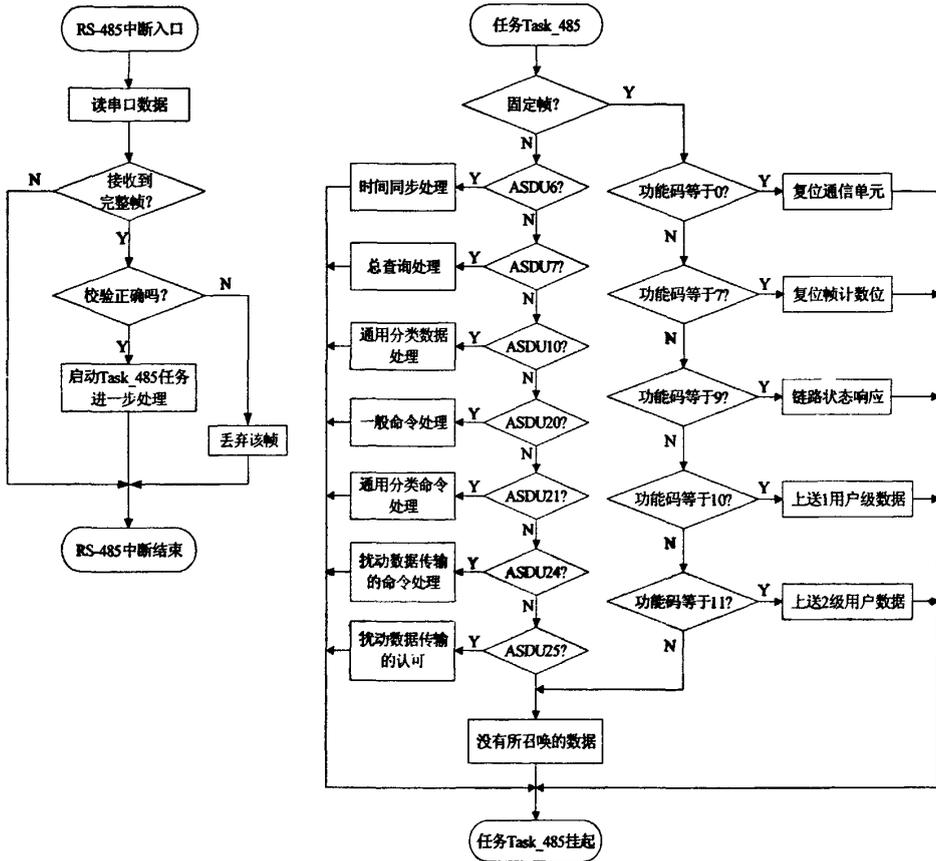


图 5-1 RS-485 中断和 103 规约处理流程图

图 5-1 中左侧的流程图是 RS-485 接收中断流程图，中断处理程序接收数据后判断是否是已经接收到一个完整的数据帧，并对完整的数据帧进行校验，若校验错误则丢弃该帧数据。若是接收到一个正确的完整帧则启动任务 Task_485 进行 103 规约的应用层处理，具体如图 5-1 右侧流程图所示。任务 Task_485 首先判断接收到数据帧是固定帧还是可变帧，然后根据功能码或类型标识调用不同的处理程序。

5.2.2 103 规约报文交换过程

本系统实现了初始化、时间同步、总召唤、召唤 1 级用户数据、召唤 2 级用户

数据等功能。以下为本文所用到的报文交换过程：

1. 初始化功能

装置上电或复位后，接收主站复位 CU 命令或复位 FCB 命令，上送 ASDU5（复位 CU 或复位 FCB），接着上送 ASDU5（启动/重启动）表示子站初始化完成。

主站：C_RCU_NA_3（复位通信单元）

子站：M_CON_NA_3（确认，ACD=1）

主站：C_PL1_NA_3（召唤 1 级数据）

子站：M_IRC_NA_3（复位通信单元）

主站：C_RFB_NA_3（复位帧计数位）

子站：M_CON_NA_3（确认，ACD=1）

主站：C_PL1_NA_3（召唤 1 级数据）

子站：M_IRF_NA_3（复位帧计数位）

2. 事件的获取功能

保护装置将开关变位、告警信息（ASDU1）、保护动作（ASDU2）作为突发事件上送（响应请求 1 级用户数据），报文中的时间标记表示事件发生瞬时时间。

主站：C_PL2_NA_3（召唤 2 级数据）

子站：M_NV_NA_3（有 1 级数据要传送，ACD=1）

主站：C_PL1_NA_3（召唤 1 级数据）

子站：ASDU2/ASDU1（突发信息，ACD=1）

.....

主站：C_PL1_NA_3（召唤 1 级数据）

子站：ASDU2/ASDU1（突发信息，ACD=0，1 级数据传送完毕）

3. 时间同步

主站发送时钟同步命令（ASDU6），保护装置在应用层进行 ASDU6 的响应（在采用广播方式，即采用发送/无回答链路服务且链路地址为 255 时，子站无链路层响应），并正确校正当地时钟；时钟同步命令后的时间采用新的经过校正的时标。时钟同步目的是为各个保护单元之间提供一个统一的时间参考。

主站：C_SYN_TA_3（时间同步）

子站：M_CON_NA_3（确认，ACD=1）

主站：C_PL1_NA_3（召唤 1 级数据）

子站：M_SYN_TA_3（时间同步，ACD=0）

4. 总召唤

主站发送总召唤命令（ASDU7），保护装置用 ASDU1 上送所有的状态信息，最后上送 ASDU8 表示总召唤结束；总召唤过程中允许传输突发事件或主站进行控制命令（ASDU20）。

主站：C_IGI_NA_3（启动总查询）

子站：M_CON_NA_3（确认，ACD=1）

主站：C_PL1_NA_3（召唤 1 级数据）

子站：ASDU2/ ASDU1（状态量，ACD=1）

主站：C_PL1_NA_3（召唤 1 级数据）

子站：ASDU1/ASDU2（状态量，ACD=1）

.....

主站：C_PL1_NA_3（召唤 1 级数据）

子站：M_TGI_NA_3（总召唤结束，ACD=0，1 级数据传送完毕）

5. 命令传输

主站发送 ASDU20（一般命令）进行信号复归（信息序号 INF=19），保护装置用 ASDU1 做肯定认可，然后上送由于执行这个命令引起的状态变位事件；如果主站发送其他的命令，则保护装置将进行否定认可，不做任何操作。

主站：C_GRC_NA_3（一般命令）

子站：M_CON_NA_3（确认，ACD=1）

主站：C_PL1_NA_3（召唤 1 级数据）

子站：ASDU1（命令的肯定认可，ACD=1）

主站：C_PL1_NA_3（召唤 1 级数据）

子站：ASDU1（命令引起的状态量变化，ACD=1）

.....

主站：C_PL1_NA_3（召唤 1 级数据）

子站：ASDU1（命令引起的状态量变化，ACD=0）

6. 通用分类服务

（1）通用分类命令

主站发送 ASDU21, INF=240 (读所有被定义组的标题), 保护装置用 ASDU10 上送其内部定义的各组的标题内容; 主站发送 ASDU21, INF=241 (读一组内所有条目的属性), 保护装置用 ASDU10 上送特定组内的全部条目的属性 (如描述、实际值、因子、量程); 主站发送 ASDU21, INF=244 (读单个条目的属性), 保护装置用 ASDU10 上送该条目的属性 (如描述、实际值、因子)。

主站: C_GC_NA_3 (通用分类命令)

子站: M_CON_NA_3 (确认, ACD=1)

主站: C_PL1_NA_3 (召唤 1 级数据)

子站: M_GD_N(T)A_3 (通用分类数据, ACD=1)

主站: C_PL1_NA_3 (召唤 1 级数据)

子站: M_GD_N(T)A_3 (通用分类数据, ACD=1)

.....

主站: C_PL1_NA_3 (召唤 1 级数据)

子站: M_GD_N(T)A_3 (通用分类数据, ACD=0, 传送完毕)

(2) 修改定值

主站发送 1 帧或连续多帧 ASDU10, INF=248 或 249 (写条目或带确认的写条目), 保护装置用 ASDU10 进行确认 (当主站 INF=249 时)。然后在主站发送 ASDU10、INF=250 (带执行的写条目) 时, 保护装置回复 ASDU10、COT=40 (肯定认可), 执行写操作; 如果主站发送 ASDU10、INF=251 (写条目终止) 时, 保护装置回复 ASDU10、COT=41, 取消所有写操作。

主站: C_GD_NA_3 (带确认的写条目)

子站: M_CON_NA_3 (确认, ACD=1)

主站: C_PL1_NA_3 (召唤 1 级数据)

子站: M_GD_NA_3 (响应带确认的写条目, 存储但不修改, COT=44)

主站: C_GD_NA_3 (带确认的写条目, 写另一个)

子站: M_CON_NA_3 (确认, ACD=1)

主站: C_PL1_NA_3 (召唤 1 级数据)

子站: M_GD_NA_3 (响应带确认的写条目, 存储但不修改, COT=44)

.....

所有定值都预选完毕

主站：C_GD_NA_3（带执行的写条目）

子站：M_CON_NA_3（确认，ACD=1）

主站：C_PL1_NA_3（召唤 1 级数据）

子站：M_GD_NA_3（修改，响应带确认的写条目，COT=40 或 COT=41）

（3）通用分类总召唤

主站发送 ASDU21，INF=245、COT=9（通用分类总召唤），保护装置用 ASDU10、COT=9 连续上送通用分类数据（状态量），最后上送通用分类总召唤结束 ASDU10、INF=245、COT=10。

主站：C_GC_NA_3（启动总召唤）

子站：M_CON_NA_3（确认，ACD=1）

主站：C_PL1_NA_3（召唤 1 级数据）

子站：M_GD_NA_3（状态量，ACD=1）

主站：C_PL1_NA_3（召唤 1 级数据）

子站：M_GD_NA_3（状态量，ACD=1）

.....

主站：C_PL1_NA_3（召唤 1 级数据）

子站：M_GD_NA_3（总召唤结束，ACD=0）

5.3 基于以太网的 103 规约

以太网具有传输速度快、可靠性高、适应性强，便于与其它自动化系统互连等优点，可较好满足变电站自动化系统的应用要求。目前继电保护装置基于以太网传输数据信息的规约还没有一个正式标准，在这种情况下采用了 IEC 60870-5-104 的第 1 节“范围与目的”中规定的那样：“本标准规定了 IEC 60870-5-101 的应用层与 TCP/IP 提供的传输功能的结合。在 TCP/IP 架构内，可以运用不同的网络类型。根据相同的定义，不同的 ASDU，包括 IEC60870-5 全部配套标准所定义的 ASDU，可以与 TCP/IP 相结合”。这相当于在 IEC 60870-5-104 的图 2 中将应用层的上层内容“选自 IEC 60870-5-101 的应用服务数据单元(ASDU)”改为“选自 IEC 608705-103 的应用服务数据单元(ASDU)”，具体如图 5-2 所示。

根据IEC60870-5-103从 IEC60870-5-5中选取的应用功能	初始化	用户进程
从IEC60870-5-103中选取的应用服务数据单元ASDU		应用层 (第7层)
APCI (应用规约控制信息) 传输接口 (用户到TCP的接口)		
TCP/IP协议子集 (RFC 2200)		传输层 (第4层)
		网络层 (第3层)
		链路层 (第2层)
		物理层 (第1层)
注: 第5、第6层未用。		

图 5-2 所定义的配套标准选择的标准版本

经过上述处理后 103 规约就可适用于以太网传输。根据采用以太网接口传输规约的要求，链路层采用网络传输规则，应用层采用 103 规约 ASDU 与 TCP/IP 网络组合方式，即 103 规约具有与 104 规约完全相同的 APCI 传输接口，相同的应答和重发机制。具体定义如图 5-3 所示。

5.3.1 应用规约数据单元 (APDU)

5.3.1.1 基本报文格式

基于以太网传输 103 规约没有为 ASDU 定义任何启动或者停止机制。为了检出 ASDU 的启动和结束，每个 APCI (见图 5-3) 包括下列的定界元素：一个启动字符，APDU 的长度，以及 4 个八位位组的控制域。可以传送一个完整的 APDU (或者出于控制目的，仅仅是传送 APCI 域) (见图 5-4)。

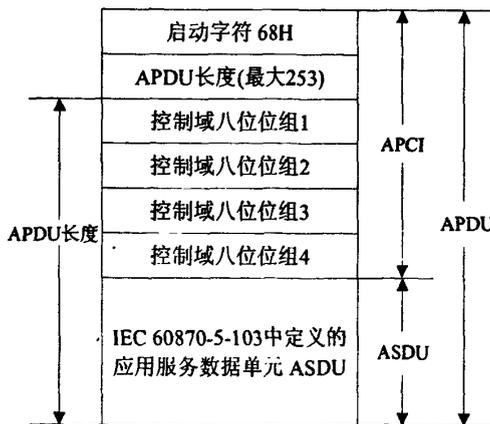


图 5-3 配套标准的 APDU 定义

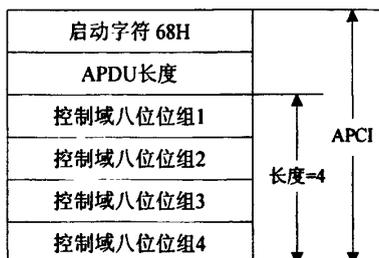


图 5-4 配套标准的 APCI 定义

启动字符 68H 定义了数据流中的起始点。

应用规约数据单元 APDU 的长度定义了应用规约数据单元 APDU 主体的长度，它由应用规约控制信息 APCI 的四个控制域八位位组和应用服务数据单元 ASDU 所组成。第一个被计数的八位位组为控制域的第一个八位位组，最后一个被计数的八位位组为应用服务数据单元 ASDU 的最后一个八位位组。应用服务数据单元 ASDU 的最大长度为 249，因为应用规约数据单元 APDU 的最大长度为 253 ($APDU_{max}=255-启动-长度$)，控制域的长度是 4 个八位位组，。

控制域定义了保护报文不至丢失和重复传送的控制信息、报文传送启动/停止以及传输连接的监视等控制信息。

5.3.1.2 三种类型报文格式介绍

这三种类型报文格式为：编号的信息传输格式（I 格式），编号的监视功能格式（S 格式）和未编号的控制功能格式（U 格式）。（编号的作用是防止报文丢失或重复传送，相当于 IEC 103 中的 FCB）

控制域第一个八位位组的第一位比特 = 0 并且第三个八位位组的第一位比特 = 0 定义了 I 格式，I 格式的 APDU 常常包含一个 ASDU。I 格式的控制信息如图 5-5 所示。



图 5-5 编号的信息传输格式（I 格式）的控制域

控制域第一个八位位组的第一位比特 = 1、第二位比特 = 0 并且第三个八位位组的第一位比特 = 0 定义了 S 格式。S 格式的 APDU 只包括 APCI。当报文接收方收到发送方的 I 格式报文后，如果没有 I 格式报文需要发送给对方，可以向对方发送 S 格式报文

以对所接收到的报文进行确认。S 格式的控制信息如图 5-6 所示。

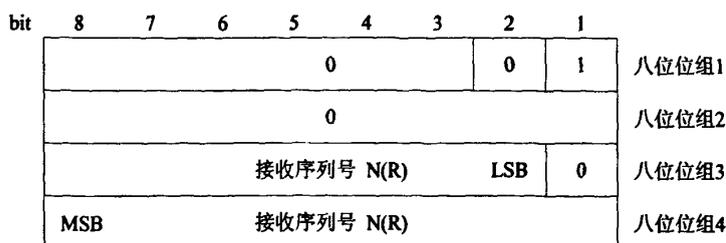


图 5-6 编号的监视功能格式 (S 格式) 的控制域

控制域第一个八位位组的第一位比特 = 1、第二位比特 = 1 并且第三个八位位组的第一位比特 = 0 定义了 U 格式。U 格式的 APDU 只包括 APCI。其作用主要在于实现 3 种控制功能，即启动子站进行数据传输 (STARTDT)、停止子站的数据传输 (STOPDT) 和 TCP 链路测试 (TESTFR); U 格式的控制信息如图 5-7 所示。在同一时刻，TESTFR, STOPDT 或 STARTDT 中只有一个功能可以被激活。STARTDT 的 U 格式报文中序列号 N 用来协商监视方向上送的 I 格式报文的发送序列号 N。TESTFR 和 STOPDT 中的序列号 N 无意义。

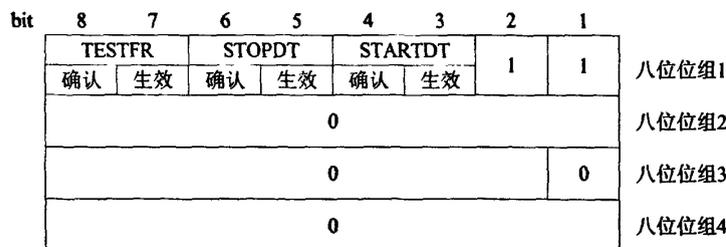


图 5-7 未编号的控制功能格式 (U 格式) 的控制域

5.3.2 报文传输控制机制

实际通讯时采用 STARTDT/STOPDT 机制，主站利用 STARTDT (启动数据传输) 和 STOPDT (停止数据传输) 来控制通讯系统以太网的数据传输。以太网是对等网络，所以通讯系统既可以主动上送报告信息，也可以响应主站定时召唤。为了防止 I 格式报文在传送过程中丢失或重复传送，在 I 格式报文的控制域定义发送序号 N(S) 和接收序号 R(S)，发送方每发送一个 I 格式报文，其发送序号应加 1，接收方每接收到一个与其接收序号相等的 I 格式报文后，其接收序号也应加 1。发送接收双方通过检查 N(S) 和 R(S) 确认报文的完整和有序，当 N(S) 和 R(S) 不同步时重发。需要注意，每次重新建立 TCP 连接后，主站和子站的接收序号和发送序号都应清零。当出现报文丢失、错序的情况时，通常意味着 TCP 连接出现了问题，发送方或接收方应关闭现在的 TCP 连接，然后再重新建立新的 TCP 连接，并在新的 TCP 连接上重新开始会话过程。某个 APDU 若被可靠接受，就返回这个被正确接收的序号。发送方在缓冲

区内保存所发送的 APDU，直到它收到和它自己的发送序号一样的接收序号，该接收序号是对所有发送序号小于或等于该号的 APDU 的有效确认，这时就可以删除缓冲区里已正确传送的 APDU。

基于以太网 103 规约的一个重要方面是重传机制，重传机制实质是“超时断开重传”。因为 TCP 连接是面向连接的，对于任何一次成功的连接，只要通信双方没有执行过断开操作，就认为一直是接通的。利用超时机制检测网络状态，是实现稳定传输数据的一个基本方法。规约默认定义当发送 12 个 APDU 未收到确认报文，应中止传输，而且定义收到 8 个 APDU 应回应发送方确认报文。因此，要分别对发送、接收报文建立计数器进行计数，按照规约定义进行超时断开处理。

为了能对 TCP 连接进行检查和维护，配套标准规定了几个超时时间，即 t_0 、 t_1 、 t_2 、 t_3 ，它们的取值范围为 1~255s，准确度为 1s。 t_0 规定了主站和子站建立一次 TCP 连接的最大允许时间， t_1 规定发送方发送一个 I 格式报文或 U 格式报文后，必须在 t_1 的时间内得到接收方的认可，否则发送方认为 TCP 连接出现问题并应重新建立连接。 t_2 规定接收方在接收到 I 格式报文后，若经过 t_2 时间未再收到新的 I 格式报文，则必须向发送方发送 S 格式报文对已经接收到的 I 格式报文进行认可，显然 t_2 必须小于 t_1 。 t_3 规定调度端或子站端每接收一帧 I 帧、S 帧或者 U 帧将重新触发计时器 t_3 ，若在 t_3 内未接收到任何报文，将向对方发送测试链路帧 TESTFR。

5.3.3 基于以太网的 103 规约的实现

母线保护装置侧作为服务器，在建立 TCP 连接前，应一直处于侦听状态并等待主站端的连接请求，当 TCP 连接已经建立，则应持续地监测 TCP 连接的状态，以便 TCP 连接被关闭后能重新进入侦听状态并初始化一些与 TCP 连接状态有关的变量；主站端作为客户机，在建立 TCP 连接前，应不断地向母线保护装置发出连接请求，一旦连接请求被接收，则应监测 TCP 连接的状态，以便 TCP 连接被关闭后重新发出连接请求。需要注意的是，每次连接被建立后，主站端和保护装置应将发送和接收序号清零。在收到主站发来的 U 格式的报文 STARTDT 后，母线保护装置响应该命令报文，然后开始传输数据，此时，应该立即上送变位信息和自发上送的数据（报告）。对于来自主站的各种命令报文，进行有效性检查：确定报文序号、通信站码、对象号等是否有效，并回应主站。规约中严格定义了各种超时时间界限，一般情况下，需要自己定义计时器，利用系统提供的函数建立自己的计数器进行计时，以达到精确性要求（精确到 1s）。

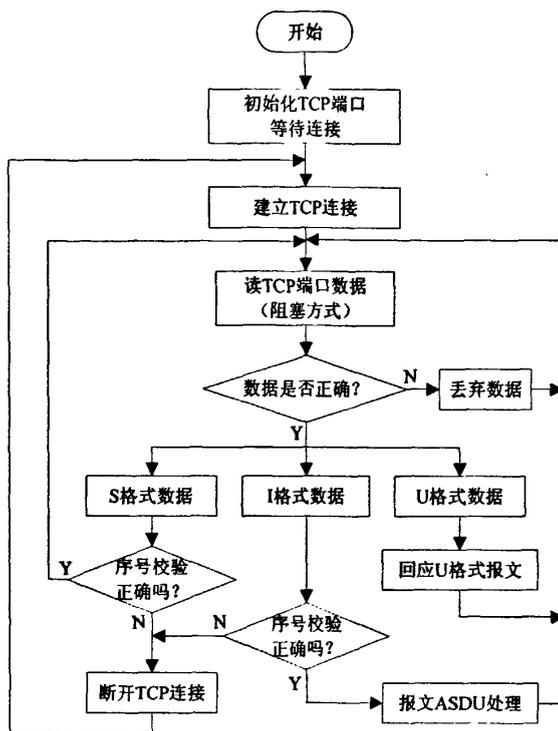


图 5-8 基于以太网传输 103 规约的通讯流程图

具体实现的 I 格式的通讯报文如下：

表 5-4 主站下发的报文

广播对时	C_SYN_TA_3	ASDU6
查询装置模拟量	C_GC_NA_3	ASDU21
查询装置定值	C_GC_NA_3	ASDU21
修改保护定值	C_GD_NA_3	ASDU10
广播对时	C_SYN_TA_3	ASDU6
查询装置模拟量	C_GC_NA_3	ASDU21
查询装置定值	C_GC_NA_3	ASDU21
修改保护定值	C_GD_NA_3	ASDU10
总召唤	C_IGI_NA_3	ASDU7
召唤装置通信工况	C_GC_NA_3	ASDU21

表 5-5 保护装置上送的报文

保护事件	M_TM_TA_3	ASDU1
	M_TMR_TA_3	ASDU2
自检信息	M_TM_TA_3	ASDU1
开关量变位	M_TM_TA_3	ASDU1
装置通信工况	M_GD_NA_3	ASDU1
装置的模拟量	M_GD_NA_3	ASDU10

装置的定值	M_GD_NA_3	ASDU10
修改保护定值	M_GD_NA_3	ASDU10
装置工况	M_GD_NA_3	ASDU1
总召唤结束	M_TGI_NA_3	ASDU8

第六章 总 结

经过近一年的努力，完成了母线保护通信系统的开发，实现了母线保护装置所要求的各种通信功能。主要包括以下几个方面：

(1) 可通过 CAN 总线收集保护 CPU、启动录波 CPU 的各种状态信息、动作信息和故障时的扰动数据记录。

(2) 保存装置操作记录、事件记录。

(3) 通过 USB 接口与 PC 机上软件的交换数据，实现了状态信息、开关量、模拟量的显示、定值上送和修改以及故障数据分析等功能，极大的方便了装置的安装调试和日常维护。

(4) 在基于 EIA RS-485 的传输系统上，实现了 IEC60870-103 规约的站初始化、时间同步、总查询、命令传输、通用分类服务等基本应用功能，并通过了规约一致性测试。

(5) 设计了一种能实现 103 规约的应用层功能，又能在以太网上传输数据的网络通信协议。

目前数字式母线保护装置的通讯系统的开发工作已经完成，并且整套装置已经顺利通过静模和动模测试以及机械工业部的国家鉴定。但从工业应用产品的角度来说，还存在一些需要继续研究和探讨的问题。

(1) 基于以太网实现的 103 规约传输没有正式出版相关的检测标准，所以无法做规约一致性的检测。此外随着 IEC61850 的普及推广，这部分通讯规约可能需要修改完善。

(2) 母线保护装置一般都处于有强电磁干扰的环境，存在电源、电击、跳闸等强电磁干扰和地电位差干扰。这些干扰通过一定的途径进入系统，会影响了系统的正常运行。在 PCB 的布线方面，最好请专家给出修改意见，使元件布置和走线更加规范，以提高系统的抗电磁干扰能力。

参 考 文 献

- [1] 贺家李. 电力系统继电保护技术的现状与发展[J], 中国电力, 1999, 32(10): 38-40.
- [2] 宋方方, 王增平. 母线保护的现状及发展趋势[J], 电力自动化设备, 2003, 7: 66-68.
- [3] 王春生, 卓乐友, 艾素兰. 母线保护[M], 中国电力出版社, 1997
- [4] 罗士萍. 微机保护实现原理及装置[M], 中国电力出版社, 2001
- [5] 中国电机工程学会继电保护运行技术分专委会. 继电保护及故障录波器信息处理系统技术规范, 2002.
- [6] 南瑞继保, 国电南自, 深圳南瑞, 四方继保, 许继集团. 继电保护装置技术使用说明书[Z].
- [7] 南瑞继保, 国电南自, 北京四方, 许继集团. 继电保护装置与监控系统通信规约[Z].
- [8] 阳宪惠. 现场总线技术及其应用[M], 清华大学出版社, 1999
- [9] 乌宽明. CAN 总线原理及应用系统设计[M], 北京航空航天大学出版社, 1995
- [10] 周立功等编. PDIUSB12 USB 固件编程与驱动开发[M], 北京: 北京航空航天大学出版社, 2003
- [11] Jan Axelson. USB 大全[M], 北京: 中国电力出版社, 2001
- [12] 王荣莉, 雷斌. 工业以太网技术的现状与发展[J], 四川工业学院学报, 2004, (3)
- [13] 吴在军, 胡敏强, 杜炎森. 嵌入式以太网在变电站通信系统中的应用[J], 电网技术, 2003, 27(1):71-75.
- [14] GILL H S, SIDHU T S, SACHDEV M S. Microprocessor-based busbar protection system[J]. IEEE Proc. Gener. Transm. Distrib. , 2000, 147(4): 252-260.
- [15] 田国政. 变电站自动化系统的通信网络及传输规约选择[J], 电网技术, 2003, 27 (9):66-68
- [16] 谭文恕. 电力系统无缝通信系统体系[J], 电力自动化设备, 2001, 25 (11):1-4
- [17] Lewis Shaw. Protocol Choices for the Substation[S], Transmission&Distribution World. 2000.
- [18] IEC60870-5-1. Transmission protocols-transmission frame format[S],

1990.

[19] IEC60870-5-2. Transmission protocols-Link transmission procedures (S), 1992.

[20] IEC60870-5-3. Transmission protocols-General structure of application data[S], 1990.

[21] IEC60870-5-4. Transmission protocols-Definition and coding of application information elements[S], 1993.

[22] IEC60870-5-5. Transmission protocols-Basic application function (S), 1995.

[23] DL/T 634-1997 idt IEC60870-5-101 基本远动任务配套标准[S], 中国电力出版社, 1998

[24] DL/T 667-1999 idt IEC60870-5-103 继电保护设备信息接口配套标准[S], 中国电力出版社, 1999

[25] DL/T 667-1999 idt IEC60870-5-104 继电保护设备信息接口配套标准[S], 中国电力出版社, 1999

[26] 刘玉民,石美传. 嵌入式实时操作系统 Nuclues PLUS 平台下 USB 接口设计[J], 电测与仪表, 2005, 5

[27] 柏嵩. IEC 870-5-103 规约的应用经验[J]. 电力系统自动化, 2000. 7: 68-70.

[28] 廖泽友, 蔡运清. IEC60870-5-103 和 IEC60870-5-104 协议应用经验[J], 电力系统自动化, 2003, 27(4):66-68.

[29] 徐立子. 变电站自动化系统IEC60870-5-103和IEC60870-5-104协议的分析 and 实施[J], 电网技术, 2002, 26(4):62-65.

[30] W. Richard Stevens, TCP/IP Illustrated, Volume 1:The Protocols, Addison-Wesley Professional, 1993

[31] W. Richard Stevens, TCP/IP Illustrated, Volume 2: The Implementation, Addison-Wesley Professional, 1995

[32] Jeremy Bentham, TCP/IP Lean: Web Servers for Embedded Systems Second Edition, CMP Books, 2002

[33] Walter Oney, Programming the Microsoft Windows Driver Model, Microsoft Press, 2002

[34] Bruce Powel Douglass, Real-Time Design Patterns, Addison-Wesley, 2002

- [35] Leen, Heffernan. TTCAN: a new time-triggered controller area network[J], *Microprocessors and Microsystems*. 2002, 26(2):77-94.
- [36] Annie Choquet-Geniet, Emmanuel Grolleau, Minimal schedulability interval for real-time systems of periodic tasks with offsets. *Theoretical Computer Science*, 2004, 310:117~134
- [37] Cena G, Valenzano A. An improved CAN fieldbus for industrial applications[J]. *IEEE Transactions on Industrial Electronics*, 1997, 44(4): 553~564
- [38] MPC555/MPC556 USER' S MANUAL. Motorola Revised 15, October 2000.
- [39] Philips Semiconductors Co, PCA82C250 CAN Controller Interface Data Sheet[Z]
- [40] USB 协议 1.1 规范 [Z]
- [41] PHILIPS 公司. PDIUSBD12 数据手册 [Z]

致 谢

本文是在导师谢志远教授的悉心指导下完成的。在攻读硕士学位期间，谢老师对我的学业和研究工作给予了很大的指导和帮助。他严谨的治学态度和高度负责的精神使我终生受益。在论文即将完成之际，谨向谢老师表示最衷心的感谢。

在论文的完成过程中，得到了高级工程师姜勇的很多帮助以及保定中力电力科技发展有限公司领导和员工的大力支持与协助，在此对他们表示衷心的感谢！

感谢我的家人，谢谢他们多年来给予我的无私关怀和支持！

感谢所有关心和帮助过我的人们！

在学期间发表的学术论文和参加科研情况

- [1] 石美传, 谢志远, 刘玉民. 主机接口(HPI)在嵌入式系统中的应用[J]. 现代电子技术, 2008, 12: 44-46