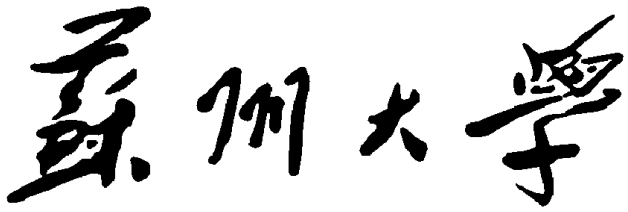


硕 士 学 位 论 文

（2008 届）

基于嵌入式以太网的 RFID 系统应用技术研究

Research on Application Technologies of RFID System based on Embedded Ethernet

研 究 生 姓 名 孙 鹏

指导教师姓名 王宜怀（教授）

专 业 名 称 计算机应用技术

研 究 方 向 嵌入式系统

论文提交日期 2008 年 4 月

中文摘要

射频识别是一种非接触式的自动识别技术，目前已广泛应用于物流、制造、公共信息服务等行业。而嵌入式系统的网络化也已成为当今后 PC 时代的重要特征。因此，RFID 技术与嵌入式以太网技术的结合已经成为学术研究的热点之一。

我国的 RFID 推广由于缺乏成熟的应用模式以及受成本制约，与国外相比还有很大差距。本文通过读取二代身份证中射频芯片的全球唯一序列号 UID，使用二代身份证代替传统只读电子标签，节省了制卡费用。并结合嵌入式以太网技术，给出了以二代身份证为载体的局域网内射频识别系统的应用模型。

文章首先分析了射频识别和嵌入式以太网技术的应用现状和工作原理。然后设计了基于嵌入式以太网的可读取二代身份证 UID 号的读写器设备，该读写器主要分为读写模块硬件中间件和具有以太网通信功能的主控系统两部分。读写模块硬件中间件被设计成 DIP40 标准封装形式，完成读卡操作并对外提供通信接口，用户通过该接口可直接获得电子标签信息而无需考虑射频实现细节，方便了 RFID 应用层的开发。主控系统采用飞思卡尔新一代 16 位微控制器 MC9S12NE64，实现了单芯片的以太网接入方案。在实现以太网模块驱动程序的基础上，设计了一个精简的嵌入式 TCP/IP 协议栈，适合无操作系统支持的嵌入式系统使用。论文的最后通过对应用实例的分析，给出了本文基于嵌入式以太网的 RFID 系统的应用方法和适用场合。

本文的研究成果对局域网内的嵌入式系统应用，现有只读卡系统的升级换代以及二代身份证的应用扩展具有参考价值和实际意义。

关键词：射频识别，嵌入式以太网，二代身份证，硬件中间件，TCP/IP

作　　者：孙 鹏

指导老师：王宜怀

Abstract

As a contactless automatic identification technology, radio frequency identification (RFID) is used for a wide variety of applications ranging from logistics to Manufacturing and public information service, etc. And the network of embedded system has already become an important characteristic of the post-PC era. Therefore, the academic research about RFID and embedded Ethernet has become a hot topic.

Because of cost restraint and the lack of application model in our country, there is a big gap in RFID extension compared with foreign countries. This paper focuses on the research of the RFID system, which proves that 2nd-Generation ID Card can be used instead of the traditional read-only card to cut the cost of RFID tag through identifying the global exclusive sequence number (UID) in 2nd-Generation ID Card; and at the same time, the application model for RFID system in local area network, with 2nd-Generation ID Card as a carrier, is supplied with Embedded Ethernet technology.

Firstly, the paper gives a detailed analysis of the application state and some basic manipulative principles related to both RFID and embedded Ethernet. Secondly, 2nd-Generation ID Card Reader device based on Embedded Ethernet is designed. The tag reader device is divided into two parts, the reader module hardware-middleware and the main control system with Ethernet Communication Function. Designed as standard 40-pin DIP footprint, the reader module hardware-middleware is programmed not only to implement read operation with RFID tag but also to provide communication interface. It is convenient that the RFID application developer can obtain the information of the RFID tag by the communication interface without considering the details of RFID. A single chip solution for Ethernet connectivity is implemented in the main control system with MC9S12NE64. On the Ethernet driver, a simplified embedded TCP/IP stack is designed for the embedded systems without operating system. Lastly, the application methods and occasions of RFID system based on embedded Ethernet are given via case analysis.

The result of this research provides some valuable references for the embedded Ethernet application, the upgrade of the traditional read-only card system and the extension of 2nd-Generation ID Card.

Key Words: RFID, Embedded Ethernet, 2nd-Generation ID Card, Hardware-Middleware, TCP/IP

Written by: Sun Peng
Supervised by: Wang Yihuai

目　录

第一章 绪论

RFID(Radio Frequency Identification，射频识别)技术是从上世纪 80 年代逐步走向成熟的一项自动识别技术，近年来发展十分迅速，并已广泛应用于身份识别、公共交通、智能楼宇、小区物业、考勤管理和商业供应链管理等领域，对提高现代化管理水平和人们生活质量具有重要的作用。传统的 RS232/RS485、USB(Universal Serial Bus，通用串行总线)等通信方式受自身限制，已无法满足不断发展的 RFID 应用系统对通信速度和传输距离的需求。随着“后 PC”时代的来临，越来越多的嵌入式系统开始通过以太网来传输数据。因此，将 RFID 技术与嵌入式以太网技术相结合，作为一种发展趋势，已经成为学术研究的热点问题之一。

本章首先分析了射频识别系统的发展历程与应用现状，然后提出了基于嵌入式以太网的 RFID 系统的设计思路，最后给出了本文的研究内容与论文结构。

1.1 课题背景

1.1.1 射频识别系统简介

射频识别技术是一种自动识别技术，利用射频方式进行非接触式双向通信，并达到识别目的[1]。一个典型的 RFID 系统由电子标签、读写器、计算机管理系统组成[2][3]，如图 1-1 所示。

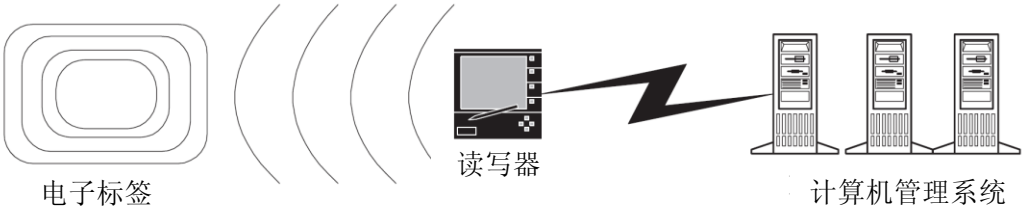


图 1-1 一个典型的 RFID 系统

RFID 技术通常是通过存储在电子标签内部芯片中的序列号来分辨标签所附着的人或物品。按电子标签获得能量的方法，一般可分为无源电子标签和有源电子标签两大类[4]。无源电子标签自身不带有电源，通过天线从读写器发出的能量中产生工作所需的电压，其特点是重量轻、体积小，寿命长，但是工作距离短。有源电子标签通过

电池供电，特点是识别距离长，但价格较高且寿命短。无源电子标签的识别过程是先由读写器发送射频信号，电子标签内的天线接收射频能量提供给内部芯片，电子标签回送识别信息给读写器；而有源电子标签则主动的发送射频信号。然后，读写器将电子标签返回的射频信号转换为数字信息，提供给计算机管理系统以完成识别过程。无源电子标签的结构图如图 1-2 所示。

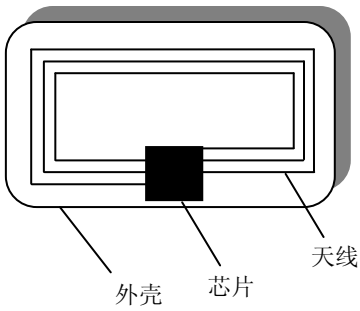


图 1-2 无源电子标签结构示意图（卡上带芯片和天线）

1.1.2 射频识别系统的发展及国内外应用现状

1. 射频系统的发展史

1948 年哈里斯托克曼发表的"利用反射功率的通信"奠定了射频识别射频识别技术的理论基础。RFID 在历史上的首次应用可以追溯到第二次世界大战期间，英国借此识别己方飞机和敌方飞机，此系统称为 IFF(Identify Friend or Foe，敌我识别)。随着芯片微型化封装技术的日趋成熟，RFID 技术已经可以投入生产，并广泛的应用于各个领域。按十年一个阶段，RFID 的发展历程如表 1-1 所示[5]。

表 1-1 RFID 发展史

年份	事件
1940-1950	雷达的改进和应用催生了射频识别技术，1948 年奠定了射频识别技术的理论基础。
1950-1960	早期射频识别技术的探索阶段，主要处于实验室实验研究。
1960-1970	射频识别技术的理论得到了发展，开始了一些应用尝试。
1970-1980	射频识别技术与产品研发处于一个大发展时期，各种射频识别技术测试得到加速。出现了一些最早的射频识别应用。
1980-1990	RFID 技术进入商业应用阶段。
1990-2000	射频识别技术标准化问题日趋得到重视，射频识别产品得到广泛采用，射频识别产品逐渐成为人们生活中的一部分。
2000 年后	标准化问题日趋为人们所重视，射频识别产品种类更加丰富，有源电子标签、无源电子标签及半无源电子标签均得到发展，电子标签成本不断降低，规模应用行业扩大。

2. 应用现状

1994 年我国启动金卡工程，这项工程以电子货币应用为重点，促进了各类卡基应用系统工程的发展，对提高现代化管理水平和人民的生活质量，推动整个社会信息化进程具有重要作用。在我国，RFID 技术主要应用于公共安全、生产管理与控制、现代物流和交通管理中[6]。利用 RFID 技术对高速移动物体识别的特点，可以将其使用在不停车收费系统中，以解决收费成本高、管理混乱以及停车排队引起的交通堵塞等问题。而在国内的一些城市(如上海)，已经开始使用基于 ISO(International Standardization Organization，国际标准化组织)/IEC(International Electrotechnical Commission，国际电工协会) 14443 A 标准的 13.56MHz 电子标签作为地铁、出租、公交的电子车票[7]。

电子标签还可以通过嵌入到身份证、护照、工作证等各种证件中，用作人员身份识别，是目前 RFID 技术应用最为广泛和成熟的领域之一。从 2004 年开始，我国启动了第二代身份证项目，这一项目将把目前的普通卡片式的身份证逐步更换为内嵌 ISO/IEC14443 B 标准的 13.56MHz 电子标签的身份证，可以与阅读身份证的仪器进行相互认证，并实现人口信息管理的现代化。为了提高信息安全，二代证中采用了加密技术，需要公安部授权的专用 SAM(Secure Access module，安全控制模块)才能读取芯片内存储的个人资料[8]。该项目可以说是国内乃至国际上最大的 RFID 应用的项目之一。根据公安部的规划，到 2008 年全国将有 10-12 亿个二代身份证，而根据专家推测，每 200 张二代身份证至少需要一台读卡器识别、验证。因此未来全国二代身份证识别、验证市场将大约需要 500～600 万台读卡器，目前市场上的二代身份证读卡器价格范围在人民币 2000 元～6000 元/台左右，因此这将为 IT 行业带来一次梦寐以求的暴富商机。

2003 年 6 月 19 日，在美国芝加哥召开的“零售业系统展览会”上，世界最大零售商沃尔玛宣布将采用 RFID 技术，以最终取代目前广泛使用的条形码，成为第一个公布正式采用该技术时间表的企业。按计划，该公司最大的 100 个供应商应从 2005 年 1 月 1 日开始在供应的货物包装箱托盘上粘贴 RFID 标签，并逐渐扩大到单件商品。这项措施为沃尔玛每年节约 13 亿美元的经营成本。和沃尔玛相似，欧洲的零售巨头麦德龙，2004 年底也开始了对于 RFID 系统的试用。西班牙马德里短角羚运动俱乐部的会员们现在使用 RFID 技术来当作门票，还可以作为获取服务、使用健身器材、购

买商品和服务的凭证。三星公司则是将 RFID 阅读芯片集成到手机产品中，为消费者提供来自产品标签上的服务信息，比如电影海报、衣服标签、博物馆或旅游票证等。

综上所述，随着技术的日趋成熟，RFID 已经深入到人们生活的各个领域中，而其美好光辉的市场前景也是毋庸质疑的。

1.1.3 嵌入式技术在以太网中的应用

1. 嵌入式系统概述

后 PC 时代的到来，使得人们开始越来越多地接触到一个新的概念—嵌入式系统。为了区别于通用计算机系统，通常把面向测控对象，嵌入到实际应用系统中，实现嵌入式应用的计算机称为嵌入式计算机系统，简称嵌入式系统[9]。

与通用计算机系统相比，嵌入式系统有一些自身的特点：

(1)资源有限

由于面向应用及实现功能单一等特点，嵌入式系统在计算与存储能力上都不如通用计算机系统。以 FLASH 空间为例，8 位 MCU(Micro Controller Unit，微控制器)的 FLASH 区域一般只有几十 KB，16 位和 32 位 MCU 的 FLASH 空间也只能达到几百 KB，这样的资源是无法与通用计算机系统相比的。

(2)可靠性与稳定性要求

嵌入式系统往往工作环境恶劣、受电噪声干扰较大。因此要特别注意系统的稳定性、电磁干扰与静电防护等方面的可靠性设计问题。在软件上一般要设定看门狗程序，在程序发生故障时重启系统。

(3)专用性强

嵌入式系统的设计以应用为核心，完成特定的功能。

(4)实时性要求

嵌入式系统有比较严格的实时性要求，对外部事件需要在给定时间内做出响应。

2. 嵌入式以太网

以太网技术是由 Xerox 公司创建并由 Xerox、Intel 和 DEC 公司联合开发的基带局域网规范，并且是当今现有局域网采用的最通用的通信协议标准。随着嵌入式系统应用的日益普及，以往单机控制系统数据量小，系统之间通信困难的缺陷也暴露出来。RS232/RS485 或 USB 等通信方式只适用于单机或区域使用，并且在传输距离和通信

速度上都存在一定的限制。嵌入式以太网技术就是在嵌入式系统的基础上实现网络化，将分散的嵌入式设备连接到以太网中，并通过以太网去控制这些智能设备。并且而随着成本的不断降低，以太网作为最通用的局域网技术在机关、企事业单位、家庭、学校中越来越普及。因此，设计具有以太网通信功能的嵌入式设备，可以利用现有的网络基础设施，省去了现场布线费用，且具有通信距离远，速度快等优点。

嵌入式系统网络化的一个主要技术障碍在于各种网络通信协议对于计算机存储空间、运算速度等方面有较高要求，而目前嵌入式系统中除部分 32 位处理器以外，普遍应用的是 8 位和 16 位 MCU，在这些 MCU 上运行 TCP/IP 等标准网络协议将占用大量的系统资源，由此将制约嵌入式系统的性能和网络通信的可靠性。因此，在嵌入系统上实现网络通信功能需要对标准协议进行裁减和优化。

1.2 设计思路与课题意义

1.2.1 设计思路

近几年，国外的 RFID 行业呈飞速发展，一些主流的 RFID 芯片厂商不仅加大了收购和合作的力度，还增加了研发投入，不断更新现有产品。我国的 RFID 推广由于缺乏成熟的应用模式以及受成本制约[10]，与国外相比还有很大差距。而且因为劳动力成本低，国内一些企业宁愿增加管理人员，也不使用 RFID 系统。

目前，我国正在全面换发二代身份证，其内嵌的 IC(Intergrated Circuit，集成电路)芯片符合 ISO14443 Type B 射频技术国际标准，并且每块芯片都拥有全球唯一的序列号也称 UID(Unique Identifier，唯一的标识符)。该 UID 由于未被加密，因此可以在无 SAM 模块的情况下被读取，并且不会涉及到二代身份证的保密性问题。

综合国内 RFID 应用现状和市场需求，本文在读取二代身份证 UID 的基础上，使用二代证代替原有的只读电子标签，节省了制作电子标签的费用；并通过与嵌入式以太网技术的结合，研究以二代身份证为载体的局域网内 RFID 系统的应用方法，不仅为现有只读电子标签管理系统的升级提供了一种参考模型，而且扩展了二代身份证的使用范围，实现了一卡多用。

本文设计的 RFID 系统由读写器和计算机管理系统组成。其中，读写器具有核心作用[11]，也是本课题研究的重点，它通过射频方式获得二代身份证的 UID 信息，并通以太网传送给计算机管理系统，对整个 RFID 系统的稳定性和安全性起决定作用；

计算机管理系统接收读写器传送的 UID 信息，针对不同应用做出相应的处理和控制。

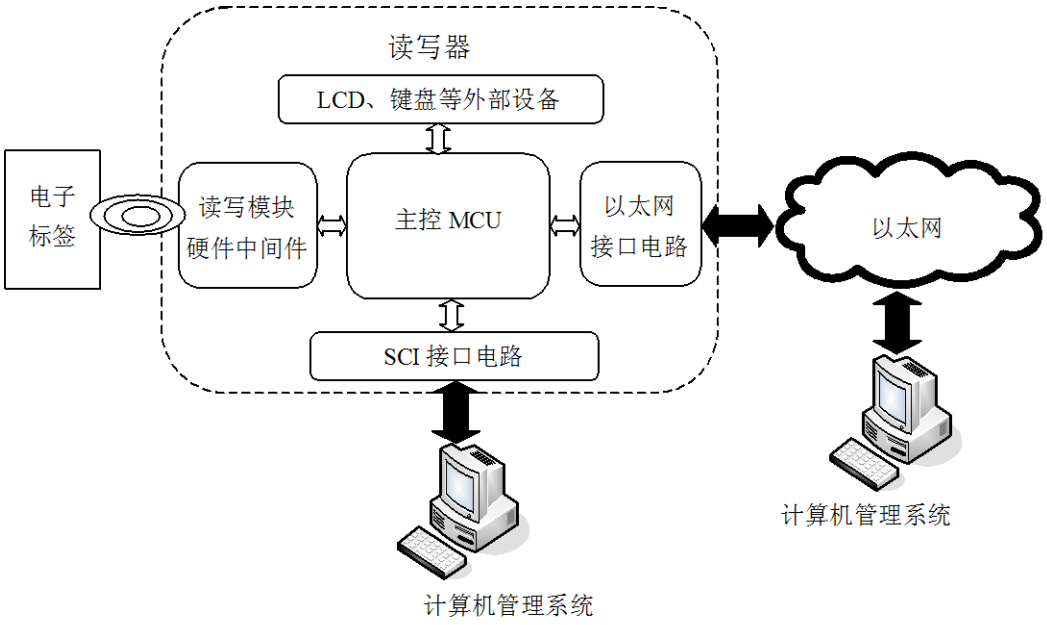


图 1-3 具有以太网通信功能的读写器结构

本文设计的读写器结构如图 1-3 所示，主要实现以下几个部分：

(1)读写模块硬件中间件

考虑到二代身份证内嵌 ISO/IEC14443 B 电子标签，而 ISO/IEC 14443 A 标准的电子标签在我国应用得最为广泛，为了提高系统的兼容性，本文设计的读写模块硬件中间件实现了读取 ISO/IEC14443 A & B 两种电子标签 UID 的功能。

由于使用射频基站芯片实现与电子标签的底层通信对一般应用开发人员来说还是比较困难的，需要花费大量时间和精力。因此，为了方便 RFID 应用层的开发，本文参照中间件通用定义，将读写模块设计为硬件中间件。该中间件完成读卡功能，并向外提供标准的软、硬件接口。这样，应用系统的开发人员就不需要了解射频基站芯片的底层驱动实现细节，而只需通过标准接口发送命令给读写模块硬件中间件，就可以获取电子标签的相关信息，大大减少了开发工作量。并且当底层的硬件更新换代时，如更换另外一种标准的电子标签，只要将中间件升级更新，并保持中间件对外的接口定义不变，应用软件几乎不需任何修改，从而保护了企业在应用软件开发和维护中的投资。

(2)主控 MCU 及以太网接口

主控 MCU 是读写器的数据处理核心。首先，它通过标准接口发送命令获取读写

模块硬件中间件中的电子标签 UID 信息；其次，实现以太网通信功能，将电子标签的相关信息通过以太网反馈给计算机管理系统；最后，接收计算机系统发送的命令信息，控制外部设备完成相应功能。

其中，在资源有限的 MCU 中实现以太网通信功能是需要解决的关键问题。不仅需要设计硬件电路，更重要的是实现精简的的 TCP/IP 协议栈，按照 TCP/IP 模型的分层结构，并结合嵌入式系统的特点，分割软件以实现各层的通信功能[12][13]。

(3)SCI 接口

为适应单机使用，保留了 SCI(Serial Communication Interface，串行通信接口)通信接口。

(4)LCD、键盘等外部设备。

键盘和 LCD(Liquid Crystal Display，液晶显示器)作为输入输出设备，方便对读写器的操作。

1.2.2 课题意义

本文设计将以太网通信与 RFID 技术相结合，由于吸收了两种技术的优点，因此具有一定的技术先进性与宽广的市场前景：

(1)使用二代身份证来代替普通只读电子标签，解决了制卡成本问题。

(2)给出了以二代身份证为载体的局域网内 RFID 系统的应用方法，为 RFID 系统应用的推广提供了一种参考模型。

(3)在嵌入式系统中实现精简的 TCP/IP 协议完成以太网通信的功能，不仅可以使用在 RFID 系统中，而且在其他领域中也有十分广阔的应用前景，如可以为远程测控系统提供数据通信平台。

(4)将 RFID 读写模块按照面向硬件设计的思想设计成硬件中间件，对外提供标准的软、硬件接口，给应用系统的开发带来了很大的灵活性，开发者不需要再去考虑读写模块的软、硬件细节，而把主要精力放在应用系统的功能实现上。

1.3 本文工作和结构

1.3.1 本文工作

本文工作安排如下：

(1)总体设计

为了规范整个开发设计过程，首先需要建立项目文档，将参考资料和软、硬件设计的内容等分类保存；然后进行系统的需求分析，制定设计开发流程；最后根据面向硬件对象设计的思想，将系统分割成功能独立的模块并分别设计实现。在芯片选型过程中，充分结合功能、价格、功耗、引脚资源、开发工具和封装等各种因素，选择最适合的芯片来实现本文的设计。

(2)读写模块硬件中间件

认真查找、理解 RFID 相关资料，特别是射频基站芯片 MF RC531 的数据手册，了解如何通过操作射频基站芯片的寄存器来完成读、写电子标签的过程。在 RFID 系统中，天线的设计也很重要，它直接关系到读写模块工作的稳定性，因此需要在了解相关电子知识的基础上进行不断调试。最后还要结合中间件的理论知识，完成对模块的封装。

(3)以太网通信

在查阅并研究 TCP/IP 协议的相关资料，并对现有的各种嵌入式以太网实现方案进行相关了解、比较后，决定采用 Freescale 公司 2004 年推出 16 位微控制器 MC9S12NE64 作为主控芯片来完成以太网通信功能。该芯片内部集成 EMAC 和 EPHY 模块，可配合第三方 TCP/IP 协议栈实现以太网的通信功能，从而实现单芯片的以太网连接方案。与多芯片的网络连接实现方案相比，具有设计简单，成本低廉，开发周期短等优点。这部分的工作重点是研究 OPENTCP、μIP 等成熟、开源 TCP/IP 协议栈的实现原理，结合 MC9S12NE64 芯片和实际应用的特点，设计一种精简、层级分割合理的嵌入式 TCP/IP 协议栈，完成以太网通信的功能。

(4)测试

功能的实现只是完成了设计的第一步，测试是保证系统可靠性的关键步骤。测试的目的就是发现各种错误和缺陷，不断的改进系统。大多数嵌入式系统系统没有显示器，这与普通的计算机系统有很大的不同，无法用一个最普通的“Hello World!”程序来表明系统的正常运行。因此测试过程中需要设置特定的输出，比如 LED 小灯或者蜂鸣器等，通过观察这些设备的状态来判断系统是否运作正常。

(5)论文

总结毕业设计的整个过程。

1.3.2 本文结构

全文共六章，各章的内容安排如下：

第一章简要叙述了 RFID 系统的发展历史和应用现状，分析了目前国内 RFID 推广的制约因素，给出了本文的研究内容与意义。

第二章分析与系统设计相关的理论基础知识。

第三章详细讲述各个功能模块的硬件设计过程，包括功能分析，元件选型、电路设计和硬件测试等步骤。

第四章阐述在硬件基础上的软件设计，包含射频读卡功能和精简的嵌入式 TCP/IP 协议栈的实现，并给出相应的软件测试结果。

第五章通过应用实例分析，给出了本文设计的 RFID 系统的应用方法和适用场合。

第六章对课题的工作进行总结，并分析了设计中存在的不足之处。

第二章 相关理论知识概要

了解相关的理论知识是完成设计的基础。本章首先阐述了射频识别的基本原理、信号的编码与调制、电子标签的分类以及相关国际标准；然后对网络参考模型、以太网技术和 TCP/IP 协议等基本概念作了简要的描述。

2.1 RFID 系统相关基础知识

2.1.1 RFID 系统基本原理

RFID 系统中，电子标签和读写器之间通过耦合元件实现射频信号的空间耦合；在耦合通道内，根据时序关系，完成能量的传递和数据的交换[1]。

耦合方式有两种：

(1) 电感耦合

通过空间高频交变磁场实现耦合。一般适合于中、低频工作的近距离射频识别系统。典型的工作频率有：125KHz、225KHz 和 13.56MHz。典型识别作用距离为 10～20cm。

(2) 电磁反向散射耦合。

发射出去的电磁波碰到目标后反射，同时带回目标信息。一般适合于高频/微波工作的远距离射频识别系统。典型的工作频率有：433MHz、915MHz、2.45GHz 和 5.8GHz。典型识别距离为 3～10m。

如表 2-1 所示，RFID 系统按工作频率的不同，应用的领域也有所区别[14]。

表 2-1 RFID 系统的频率使用范围

频率范围	特性及适应领域
125-134KHz	低频范围。识别距离小于 0.5m，数据传送速度低于 1Kb/S。应用于动物识别。
13.56MHz	识别距离可达 1.5m，数据传送速度大约 25Kb/S。多用于身份识别、门禁系统。
433-956MHz	高频范围。其中 433-864MHz 频段的识别范围长达 100m；而 865-956MHz 频段的识别范围从 0.5～5m。整个 433-956MHz 频段的数据传送速度为 100Kb/S。一般用于物流系统。
2.45GHz	最大可识别距离达到 10m，数据传送速率为 100Kb/S。一般用于车辆管理等。

2.1.2 RFID 系统中信号的编码与调制

从读写器向电子标签传输数据要经过三个主要的阶段：读写器中的信号编码和调制，通过传输介质，以及电子标签中的解调和信号译码[1]。

1. 编码

信号编码系统的作用是使要传输的信息和它的信号表示尽可能最佳的与传输通道的性能相匹配，这样处理可以对信息提供某种程度的保护，防止信息受干扰或者相碰撞。

RFID 系统中常用的的编码方法有：

(1)NRZ(Non Return to Zero，反向不归零制)编码

高电平表示 1，低电平表示 0，并且在表示完一个码元后，电压不需回到 0。

(2)曼彻斯特(Manchester)编码

曼彻斯特编码是一种自同步的编码方式，即时钟同步信号隐藏在数据波形内。在曼彻斯特编码中，每一位的中间有一跳变，位中间的跳变不仅作为时钟信号，还作为数据信号；从高到低的跳变表示"1"，从低到高的跳变表示"0"。还有一种是差分曼彻斯特编码，每位中间的跳变仅提供时钟定时，而用每位开始时有无跳变表示"0"或"1"，有跳变为"0"，无跳变为"1"。

(3)单极归零制编码

在第一个比特周期中的高电平表示“1”，而持续整个比特周期的低电平表示“0”。

(4)差动双相编码

在半个比特周期内的任意边沿跳变表示“0”，没有沿跳变表示“1”。

(5)米勒(Miller)编码

在半个比特周期内的任意边沿跳变表示“1”，而经过下一个比特周期中的“1”表示“0”。一连串的“0”在比特周期开始时产生电平交换。

(5)差动编码

用电平的跳变表示二进制“1”，信号保持不变表示“0”。

2. 调制

调制则是对信号源的编码信息进行处理，使其转变为适合传送的形式。一般是改变高频载波的信号处理，即使其振幅、频率或相位与调制的基带信号相关。

射频识别系统常用的调制方法有：

(1)ASK(Amplitude Shift Keying，振幅键控)

用数字调制信号控制载波的通断。“0”不发送载波，“1”发送载波。振幅键控实现简单，但抗干扰能力差。

(2)FSK(Frequency Shift Keying，频移键控)

频移键控是利用两个不同频率 F1 和 F2 的振荡源来代表信号“1”和“0”。移频键控实现简单，能区分通路，但抗干扰能力不如相移键控。

(3)PSK(Phase Shift Keying，相移键控)的数字调制法

二进制 PSK 相移键控是将编码状态的“0”和“1”转变成载波振荡相对基准相位的相应状态，即在相位状态 0 和 180 之间切换。相移键控抗干扰能力强，但在解调时需要有一个正确的参考相位，即需要相干解调。

解调是调制的逆过程，以再生基带信号。信号解码的任务是从基带编码的接收信号中恢复原来的信息，并识别和标识出传输错误。

2.1.3 电子标签的分类

电子标签具有防水、防磁、耐高温、使用寿命长、读取距离远、存储数据可加密和更改等优点[4]。在实际应用中，电子标签由标签天线、芯片等采用特殊封装工艺制造而成，附着于待识别物体的表面。标签内存储的数据量在几个字节到几千个字节之间，对于任何一个电子标签来说，都有一个唯一的序列号，这个序列号在标签的制造时由厂商固化写入，一旦写入后，这个号码是不能改变的。按照不同的分类标准，标签有许多不同的分类。

1. 有源电子标签与无源电子标签

在实际应用中，必须给电子标签供电它才能工作，虽然它的电能消耗是非常低的，按照标签获取电能的方式不同，可以把标签分成有源式标签与无源式标签。

有源式电子标签通过标签自带的内部电池进行供电，它的电能充足，工作可靠性高，信号传送的距离远。另外，有源式标签可以通过设计电池的不同寿命对标签的使用时间或使用次数进行限制，它可以用在需要限制数据传输量或者使用数据有限制的地方。有源式标签的缺点主要是价格高，体积大，标签的使用寿命受到限制，而且随着标签内电池电力的消耗，数据传输的距离会越来越小，影响系统正常工作。

无源式标签的内部不带电池，需要外界提供能量才能正常工作。无源式标签典型的产生电能的装置是天线与线圈，当标签进入系统的工作区域，天线接收到特定的电磁波，线圈就会产生感应电流给标签供电。无源式标签具有永久的使用期，常常用在标签信息需要每天读写或频繁读写多次的地方，而且无源式标签支持长时间的数据传输和永久性的数据存储。无源式标签的缺点主要是数据传输的距离要比有源式标签短。因为无源式标签依靠外部的电磁感应而供电，所以它的电能比较弱，数据传输的距离和信号强度受到限制，需要敏感性比较高的信号接收器才能可靠的识别与读取。但是它的价格、体积和易用性决定了它是电子标签的主流。

2. 只读标签与可读可写标签

根据内部使用存储器类型的不同，标签可以分成只读标签与可读可写标签。

只读标签内部只有 ROM（Read Only Memory，只读存储器）。ROM 中存储有标签的标识信息。这些信息可以在标签制造过程中由制造商写入，也可以在标签开始使用时由使用者根据特定的应用目的写入特殊的编码信息。这种信息只能是一次写入，多次读出。只读标签多数容量较小，一般可以用作标识标签，用于对特定的标识项目，如人、物、地点进行标识，关于被标识项目的详细的特定的信息，只能在与系统相连接的数据库中进行查找。

可读可写标签内部的存储器除了 ROM 之外，还有 EEPROM（Electrically Erasable ProgrammableRead Only Memory，电可擦除可编程只读存储器），它除了存储数据功能外，还具有在适当的条件下允许多次对原有数据的擦除以及重新写入数据的功能。可读可写标签还可能有 RAM（Random Access Memory，随机存取存储器），用于存储标签反应和数据传输过程中临时产生的数据。可读写标签存储的数据相对比较大，这种标签一般都是用户可编程的，标签中除了存储标识码外，还存储有大量的被标识项目的相关信息。另外在读标签的过程中，可以根据特定的应用目的控制数据的读出，实现在不同的情况下读出的数据部分不同。

2.1.4 近耦合 RFID 的国际标准 ISO/IEC 14443

ISO 和 IEC 是当今世界上两个最大的国际标准化机构。ISO/IEC 14443 由 4 部分组成，定义了 13.56MHz 的近耦合 RFID 国际标准，其内容包含了阅读器也称 PCD(Proximity Coupling Device，邻近耦合设备)和应答器也称 PICC(Proximity

Integrated Circuit Card，近耦合 IC 卡)。

ISO/IEC 14443 标准的 4 部分如下：

1. 物理特性

ISO/IEC 14443 中规定 PICC 的物理特性与尺寸应满足 ISO/IEC 7810 中规定的 ID-1 的需求，即 $85.72mm \times 54.03mm \times 0.76mm \pm$ 容差，与磁卡、接触型 IC 卡标准尺寸完全一致。此外，还对紫外线、X 射线、交流电场、交流磁场、静电、静磁场、工作温度、动态弯曲和扭曲等方面做了相应规定[15]。

2. 射频接口

ISO/IEC 14443 中规定 PCD 和 PICC 的操作顺序如下：

(1)PCD 的射频场激活 PICC；

(2)PICC 等待 PCD 的命令；

(3)PCD 发送一个命令；

(4)PICC 回送一个应答。

其中 PCD 发送的交变磁场其频率为 $13.56MHz \pm 7KHz$，强度在 1.5A/m ～ 7.5A/m 之间。国际标准 ISO/IEC 14443 中规定了两种 PCD 与 PICC 之间的数据传送方式：Type A 和 Type B，这两种方式的通信标准如表 2-2 所示[15]。一张 PICC 只需选择两种方式之一，而标准的阅读器必须同时支持这两种传送方式，以便支持所有的 PICC。PCD 在空闲状态时可以在两种方式中切换，但是在 PCD 与 PICC 的通信过程中不允许转变通信方式。

表 2-2 ISO/IEC 14443 Type A 和 Type B 通信方式比较

		Type A	Type B
PCD 到 PICC 的数据传输	信号调制方式	幅移键控(ASK) 100％	幅移键控(ASK) 10％
	数据编码格式	改进的米勒(Miller)编码	非归零编码(NRZ)
	同步化	位级同步	每组位元均有起始和终止位
	传输速度	106Kbit/S	106Kbit/S
PICC 到 PCD 的数据传输	信号调制方式	幅移键控(ASK)，次载波频率为 874KHz	二进位相移键控(BPSK)，次载波频率为 874KHz
	数据编码格式	曼彻斯特(Manchester)编码	非归零编码(NRZ)
	同步化	位级同步	每组位元均有起始和终止位
	传输速度	106Kbit/S	106Kbit/S

3. 初始化与防冲突

ISO/IEC14443 中规定的“初始化”是指 PICC 进入 PCD 所产生的射频场后，从射频场获得能量、与读写终端建立通信信道、确定通信协议所规定的参数，并进入应用状态的过程，包括 REQ 和 ATQ 命令的内容[15]。

“防冲突”是指在射频场中有两张以上的 PICC 存在时，各个 PICC 返回到 PCD 的信号将发生冲突。如在通信时的某一位上，一张 PICC 发送逻辑“1”，而另一张 PICC 发送逻辑“0”，则将在信号上产生冲突。为防止这种现象发生，PCD 按照规定的机制与各 PICC 建立通信信道，在每一时刻只与指定的一张 PICC 建立联系，以避免冲突的发生。

4. 传输协议

传输协议中分别按 Type A 及 Type B 两种方式，定义了由 PCD 发送数据的半双工传输协议和 PICC 的激活过程和解除活动过程[15]。

2.2 计算机网络知识简介

2.2.1 网络参考模型

计算机网络是通过通信设施，将地理上分散的具有自治功能的多个计算机系统互连起来，进行信息交换，实现资源共享、互操作和协同工作的系统。OSI(Open System Interconnect，开放式系统互联)参考模型由国际标准化组织 ISO 于 1981 年制定。这个模型把整个网络按功能分为 7 层，由低到高分别是物理层、数据链路层、网络层、传输层、会话层、表示层和应用层。每层完成一定的功能并直接为其上层提供服务。双方的通信是在对等层次上进行的，不能在不对称的层次上进行通信。

TCP/IP 模型通常被认为是一个四层的协议系统[16]，包括应用层、传输层、网络层及链路层，每层负责不同的功能。其中应用层处理特定的应用细节；传输层为两台主机上的应用程序提供端到端的通信；网络层解决主机到主机的通信问题；而链路层处理与电缆的物理接口细节。

OSI 模型与 TCP/IP 模型的对比以及相关协议如图 2-1 所示[17]。

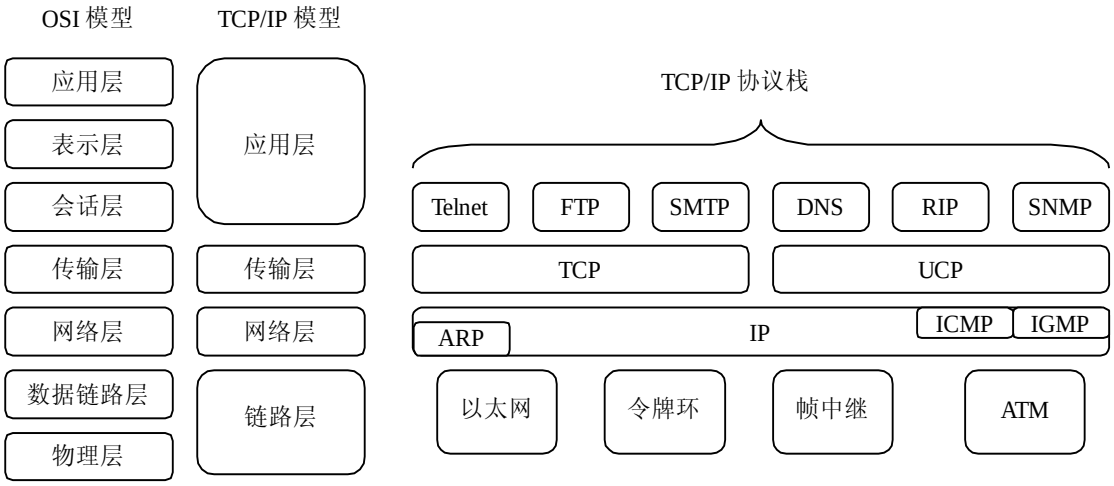


图 2-1 网络参考模型及相关协议

2.2.2 以太网技术

以太网是由 Xerox、Intel 和 DEC 公司在 1982 年联合公布的一个标准，并几乎垄断着现有的有线局域网市场[18]。IEEE 制定的 IEEE 802.3 标准给出了以太网的技术标准。它规定了包括物理层的连线、电信号和介质访问层协议的内容。以太网可以采用多种连接介质，包括同轴缆、双绞线和光纤等。

1. 以太帧结构

在以太网中，信息被封装成以太帧的格式进行传输，其格式如图 2-2 所示[19]。

前导位	帧起始位	目的 MAC 地址	源 MAC 地址	长度/类型	数据域	CRC 校验
7	1	6	6	2	46~1500	4

图 2-2 以太帧结构

前导位(7 字节)用来使局域网中的所有节点同步，帧起始位(1 字节)是帧的起始标志，这两部分一般都由硬件自动添加/删除。

接下来是目的 MAC 地址、源 MAC 地址。MAC(Media Access Control，介质访问控制)地址是识别 LAN(局域网)节点的标识。也就是说，在网络底层的物理传输过程中，是通过物理地址来识别主机的，它一般也是全球唯一的。在以太网中，物理地址是 48bit(比特位)的整数，如：F0-4E-77-8A-35-1D，前 24 位是由生产网卡的厂商向 IEEE 申请的厂商地址，后 24 位由厂商自行分配，这样的分配使得世界上任意一个拥有 48 位 MAC 地址的网卡都有唯一的标识。MAC 广播地址是全 1，即 FF-FF-FF-FF-FF-FF。

MAC 地址中，第一字节最低位为 1 表示多播地址，为 0 表示唯一地址；次低位为 1 表示局部唯一地址，为 0 表示全球唯一地址。

长度/类型位(2 字节)的值小于等于 1500 字节时，表示的是数据域里面的实际数据字节数。当接收以太帧时，将实际的数据域的数据字节数和长度/类型位的值进行比较，如果不匹配则报告错误。当长度/类型位的值大于等于 1536 字节时，这个域的值表示的是数据域内数据的类型，比如 0x0800，表示数据域内的是 IP 数据报。如果长度/类型位的值大于 1500 而小于 1536，则此以太帧无效。

数据域(46～1500 字节)这里是指 IP 数据报。以太网的最大传输单元(MTU)是 1500 字节，如果某个 IP 数据报超出这个界限，则需要将数据报进行分段。在嵌入式解决方案中，为了简化协议一般不使用超过 1500 字节的数据报，以避免分组。数据域的最小长度是 46 字节，当 IP 数据报小于这个值，则需要填充到 46 字节。通信时 IP 数据报可以根据其首部的长度字段来判断哪些是需要的数据，哪些是填充的数据。

CRC(4 字节)字段用于帧内后续字节差错的循环冗余码校验。

2. CSMA/CD

以太网通信是一种不可靠通信，实际它并不知道通信的对方有没有真正收到自己发出的数据。以太网在半双工模式下采用 CSMA/CD（Carrier Sense Multiple Access/Collision Detect），即载波监听多路访问/冲突检测方法[19]。CSMA/CD 是一种分布式介质访问控制协议，网中的各个站（节点）都能独立地决定数据帧的发送与接收。每个站在发送数据帧之前，首先要进行载波监听，只有介质空闲时，才允许发送帧。这时，如果两个以上的站点同时监听到介质空闲并发送帧，则会产生冲突现象，这使发送的帧都成为无效帧，发送随即宣告失败。每个站点必须有能力随时检测冲突是否发生，一旦发生冲突，则应停止发送，以免介质带宽因传送无效帧而被白白浪费。在随机延时一段时间后，再重新争用介质，重发送帧。

2.2.3 TCP/IP 协议原理

TCP/IP 协议允许不同厂家、运行不同操作系统的各类计算机进行相互通信[16]。当用户使用 TCP/IP 协议传输数据时，数据被送入到协议栈中，然后逐个通过每一层直到被当作一串比特流送入网络。每层对收到的数据都要增加一些首部信息，如图 2-3 所示。

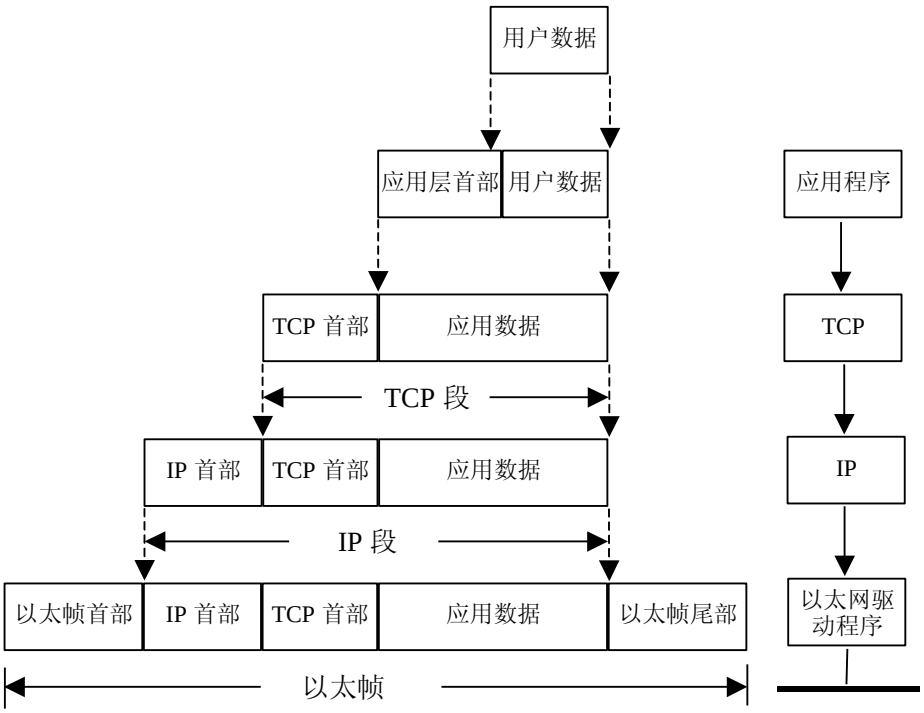


图 2-3 协议逐层封装过程

1. ARP协议分析

以太网中，ARP(Address Resolution Protocol，地址解析协议)的功能是在 32 位的 IP 地址和 48 位 MAC 地址之间提供动态映射，其请求和应答分组格式如图 2-4 所示[20]。

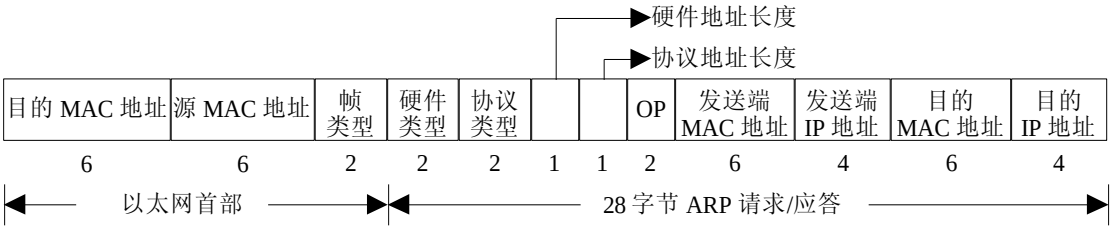


图 2-4 用于以太网的 ARP 请求或应答分组格式

帧类型字段为 0x0806，表示 ARP 请求或应答；硬件类型表示网络上硬件的类型，如果是以太网则为 1；协议类型表示映射的协议地址类型，0x0800 表示 IP 地址；硬件地址长度指 MAC 地址长度时为 6；使用 IPv4 时，协议地址长度为 4；操作号表示操作内容的数值，1 为 ARP 请求，2 为 ARP 响应，3 为 RARP 请求，4 为 RARP 响应；接下来的是发送端 MAC 地址、发送端 IP 地址、目的 MAC 地址和目的 IP 地址，其中对于 ARP 请求来说，要用全 0 来填充目的 MAC 地址部分，当目的主机收到 ARP 请求后，就把自己的 MAC 地址填充进去并返回给发送请求的主机。

2. IP协议分析

IP 协议(Internet Protocol，网际协议)的基本功能是提供无连接的数据报传送服务和数据报路由选择服务，但不保证服务的可靠性[21]。IP 协议是 TCP/IP 协议族中最核心的协议，TCP、UDP 和 ICMP 都以 IP 数据报的格式传输。

Internet 上的每台主机都有一个唯一的 IP 地址，IP 协议就是使用这个地址在主机之间传递信息，这是 Internet 能够运行的基础。IP 数据报的格式如图 2-5 所示，标准的 IP 首部为 20 个字节。

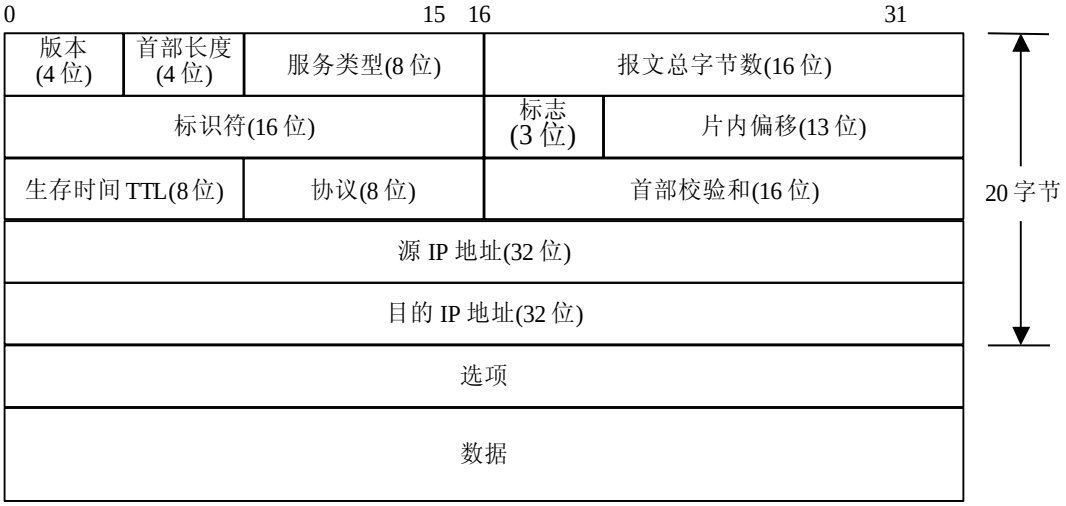


图 2-5 IP 首部

版本信息为 4 表示 IPv4，为 6 则表示 IPv6，一般使用 4；首部长度表示 IP 首部占 32 位的长度，将其乘以 4 得到首部字节数；服务类型不处理；报文总字节数表示整个 IP 数据报的长度，将其减去首部长度就可以得到数据长度；标识符唯一标识主机的每一份数据报；标志和片内偏移用于 IP 分片；生存周期设置了数据报可以经过的最多路由器数，每过一个路由减 1，为 0 则丢弃；协议字段表示示哪个高层协议调用了 IP 协议；首部校验和不包含数据部分；每个 IP 数据报都包含源 IP 地址和目的 IP 地址；选项字段一般不使用。

3. ICMP协议分析

ICMP(Internet Control Message Protocol，Internet 控制信息协议)协议传递差错报文，并被 IP 层或更高层协议（TCP 和 UDP）使用。ICMP 报文是在 IP 数据报内部被传输的，如图 2-6 所示[22]。

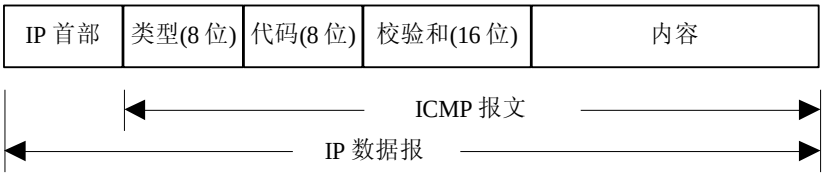


图 2-6 封装在 IP 数据报内的 ICMP 报文

类型字段可以有 15 个不同的值，表示不同特定类型的 ICMP 报文。当类型字段为 8，代码字段为 0 时，表示是一个 ICMP 回显请求报文。如果类型字段的值变成 0 则说明这是一个回显应答报文。

4. TCP协议分析

TCP(Transmission Control Protocol，传输控制协议)称为面向连接的协议[23]。两个进程使用 TCP 发送和接收数据之前，首先要通过 3 次握手建立进程所在的计算机之间的连接。在完成握手时，每台计算机都要确认握手中指定的端口可用于接收来自另一台计算机指定端口的通信。然后，双方都可以使用该连接向对方计算机发送 TCP 报文段。

在通过已建立的连接接收到数据时，目的主机响应，返回数据是否已正确到达、是否可以发送更多的数据信息。如果可以，还要返回目的主机能接收新数据的数量。

要关闭连接，双方都需要发送关闭连接的请求，并且等待对方对请求的确认。

5. UDP协议分析

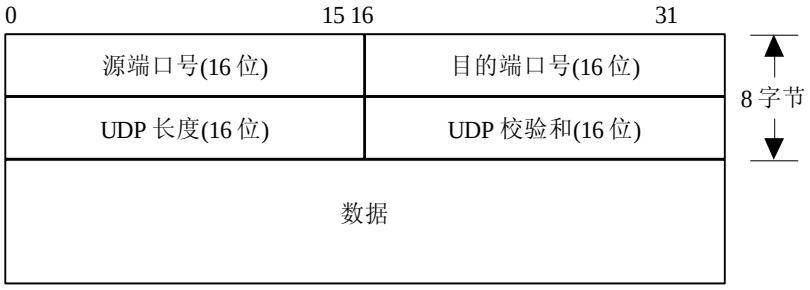


图 2-7 UDP 首部

UDP(User Datagram Protocol，用户数据报协议)是一种基本的通信协议，只在发送的报文中增加了端口寻址和可选的差错检测功能。它不是一种握手信息协议，不能确认接收到的数据或交换其他流量控制信息。UDP 是一种非连接协议，计算机在使用 UDP 发送报文之前，不要求通信的对方已联网或指定的目的端口可用于通信。正因为如此，将 UDP 称为不可靠协议，即如果只使用 UDP，则发送方不知道目的主机

何时是否接收到报文。

UDP 首部由 4 个字段组成，其后是要传输的数据，如图 2-7 所示[24]。

源端口号标识发送报文的计算机端口或进程，如果接收进程不需要知道发送数据报的进程，则该字段可置为 0；目的端口号标识接收报文的目的主机端口或进程；数据报长度指整个数据报的长度，以字节为单位，包括报头，最大值为 65535；UDP 检验和是根据 UDP 数据报和伪报头计算得到的差错检测值。

UDP 的大多数功能不如 TCP，所以 UDP 的实现要简单些，更适合于特定的应用场合。但是 UDP 可将报文发送到多个目的主机，包括向局域网内所有的 IP 地址以广播方式发送，或者向指定的 IP 地址以组播方式发送。对于 TCP 而言，广播和组播都不现实，因为源主机必须与所有目的主机握手。

2.3 本章小结

本章主要工作总结如下：

(1)简要描述了射频识别的基本原理，给出了相关调制和编码技术的定义以及电子标签的分类方式，对近耦合 RFID 国际标准 ISO/IEC 14443 的 4 部分内容分别进行了分析，为读取 ISO/IEC 14443 电子标签 UID 信息提供了理论参考。

(2)探讨了网络参考模型中 OSI 模型与 TCP/IP 模型的区别与联系。并从实际开发角度分析了以太帧格式与 CSMA/CD 机制，分层讨论了 TCP/IP 协议模型各层实现原理及相关首部信息，并对 UDP 协议与 TCP 协议各自的优缺点进行了总结与对比。

第三章 读写器硬件设计

硬件是整个系统的物理基础，为软件的运行提供了平台。根据系统功能需求，本文设计将读写器硬件系统划分为两个部分：一部分是读写模块硬件中间件；另一部分是具有以太网通信功能的主控系统，并包含一些外部设备。这样设计的优点是不仅可以方便地将两部分独立制板调试，还可以减少这两部分之间的信号干扰，并提高硬件模块的复用性。

本章将分别阐述这两部分的硬件设计，包括各自硬件系统的组成、芯片的选型、硬件接口与功能模块的硬件测试。

3.1 读写模块硬件中间件

3.1.1 中间件设计思路

中间件是位于操作系统与应用系统之间的服务软件，屏蔽了底层硬件、操作系统和数据库的差异，为应用软件开发构建了一个标准的平台[25]。

参考中间件的通用定义，并结合嵌入式系统特点，本文设计的读写模块硬件中间件结构如图 3-1 所示，向下屏蔽了电子标签、射频基站芯片等硬件环境的差异，向上为 RFID 应用层提供标准的软、硬件接口。应用层的开发基于该接口进行，而无需考虑 RFID 的实现细节，并且不管底层的硬件怎样更新换代，只需将中间件升级更新，并保持该中间件的对外接口定义不变，应用软件几乎不需做任何修改，从而提供了一个相对稳定的高层应用环境。

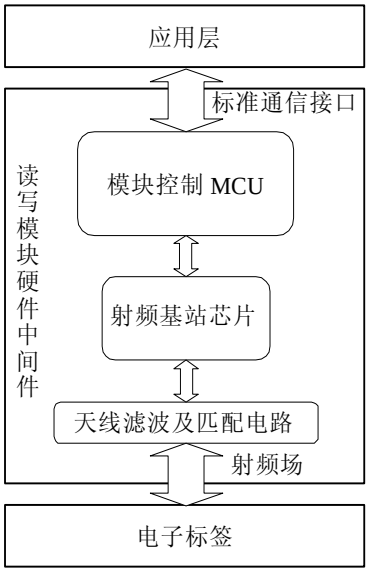


图 3-1 读写模块硬件中间件结构

因此，读写模块硬件中间件主要完成两项功能：一是通过 MCU 控制射频基站芯片工作，读取电子标签信息；二是为 RFID 应用层提供标准通信接口，由于嵌入式设计通常涉及硬件部分，所以除了软件接口外，硬件接口的设计也很重要。

3.1.2 芯片选型及功能概述

1. 选型原则

在嵌入式硬件设计中，一旦确定了系统功能，首先要做的就是芯片的选型。选型的原则如下：

(1)性能

嵌入式系统一般都是针对特定应用而专门设计的，因此芯片选型首要考虑的就是性能，在此基础上再考虑价格等其它因素。芯片的性能包含多个因素，比如位数、运算速度、内部存储器大小、I/O 引脚个数、集成何种外设模块等[9]。

(2)价格

对嵌入式系统设计来说，目标不是选择位数最高、速度最快、功能最强的 MCU，这样的 MCU 往往价格较高，增加了整个系统的成本。而在激烈的市场竞争中，价格因素对产品的销路具有很大的影响。除此之外，芯片的购买是否方便，市场供应量是否充足也是需要考虑的方面。

(3)熟悉程度

目前市场上出售的芯片种类繁多，而且各个厂家提供的开发工具一般互不通用。因此在实际开发中，尽量选择设计者熟悉和开发工具完备的芯片，可以减少开发周期，提高开发效率。

(4)功耗

嵌入式系统的功耗也是设计实际产品时需要考虑的[26]。尤其对需要电池供电的便携式系统，降低系统功耗可以延长电池的寿命，从而降低系统的运行成本。对芯片来说，可以通过选择内核简单、供电电压低、带低功耗模式的 MCU 来降低功耗。

(5)封装

一般来说，同一款芯片可能有多种封装形式。常见的芯片封装有 DIP(Dual In－line Package，双列直插式封装)、PQFP(Plastic Quad Flat Package，塑料方型扁平式封装)和 PLCC(Plastic Leaded Chip Carrier，塑封有引线芯片载体)等。DIP 是最普及的封装形式，并可以配合 DIP 结构的芯片插座，方便了芯片的更换，但一般面积和体积较大。而采用 SMD(Surface Mounted Devices，表面贴装器件)形式的封装具有体积小、可靠性高等优点，但拆卸时需要专用工具。因此，选型时对易损耗并需要更换的芯片多选择 DIP 封装，而当对芯片外形尺寸有限定时，则采用 SMD 形式的封装。

读写模块硬件中间件的硬件设计中，芯片选择的重点就是确定合适的射频基站芯片和模块控制 MCU。

2. 射频基站芯片MF RC531

射频基站芯片内部集成了能量传输线路给电子标签内的芯片提供工作电源，还集成了信号放大和天线驱动线路以完成与电子标签之间的信号发送与接收。同时它和 MCU 的连接也非常简便，只需极少量的外围驱动电路即可完成读写电子标签及与 MCU 通信的功能，可以最大限度地降低应用系统的生产成本和开发费用。

在我国，13.56MHz 的 RFID 由于国际标准相对成熟，所以应用得最为广泛。而且二代身份证也内嵌 ISO/IEC14443 B 标准 13.56MHz 的电子标签。因此，选择一款兼容 ISO/IEC14443 A & B 的射频基站芯片符合本文的设计需要。目前市场上可供选择的芯片有更名为 NXP 的飞利浦半导体公司出产的 MF RC531 和 CL RC632；复旦微电子的 FM171X 系列和 FM172X 系列；EM Microelectronic 公司的 EM4094 和 EM4294 等。鉴于国内市场使用 NXP 公司 Mifare 芯片的电子标签应用广泛，而且其 RC5XX 系列在国内应用比较早，占据市场主流，其性价比较高购买也方便，所以本文选用 RC531 作为读写模块硬件中间件中的射频基站芯片。

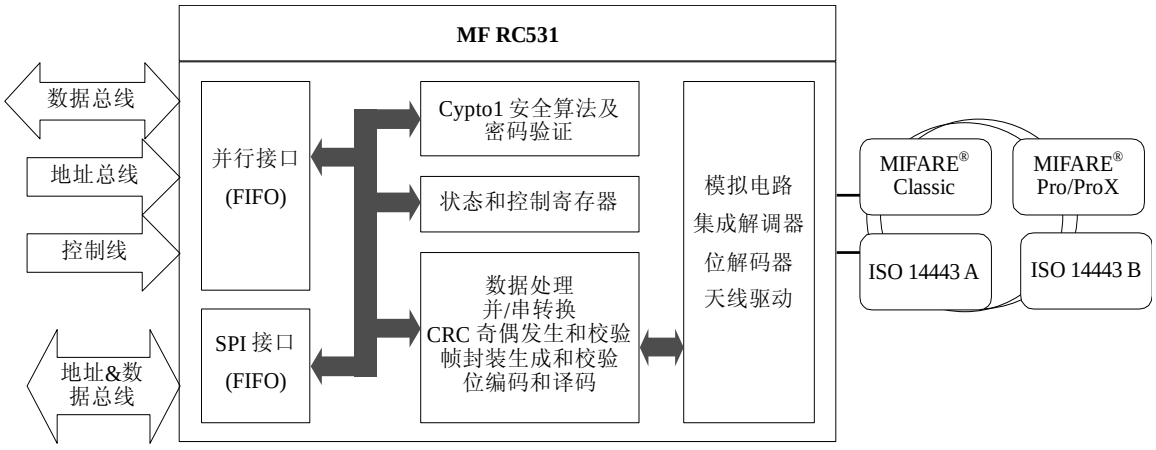


图 3-2 MF RC531 功能框图

MF RC531 是 NXP 公司生产的应用于 13.56MHz 非接触式通信的高集成 IC 读写芯片系列中的一员，其功能框图如图 3-2 所示。支持 ISO 14443 A&B 的所有层协议；内部的发送器模块不需要外加有源电路就能够直接驱动天线，操作距离可达 100mm；接收器模块接收天线传送来的信号，然后对该信号进行解调和解码；数字模块负责处理 ISO14443 帧并使用奇偶校验和 CRC 校验进行错误检测；RC531 与主机之间可采

用并行模式或 SPI(Serial Peripheral Interface，串行外围设备接口)模式进行通信[27][28]。

MF RC531 采用 32 脚 SO 封装，如图 3-3 所示。使用 3 个独立的电源以实现在 EMC 特性和信号解调方面达到最佳性能。

一些主要引脚按不同功能划分如下：

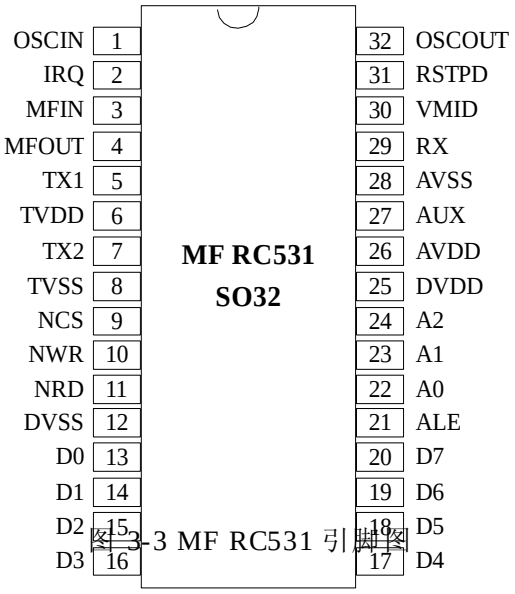


图 3-3 MF RC531 引脚图

(1)天线

为了驱动天线，MF RC531 通过 TX1(5 脚)和 TX2(7 脚)两个发送器引脚提供 13.56MHz 的能量载波，并根据寄存器的设定值发送命令信号。天线接收到电子标签的反馈信号后，经过天线匹配电路送到 RX 接收脚。TVDD(6 脚)、TVSS(8 脚)接电源和地用于对天线驱动部分供电。

(2)模拟电源与数字电源

AVDD(26 脚)、AVSS(28 脚)接电源和地用于对模拟部分如振荡器、模拟解调器和解码器等电路供电。

DVDD(25 脚)、DVSS(12 脚)接电源和地用于对数字处理部分供电。

(3)复位引脚

RSTPD(31 脚)为复位引脚。当引脚检测到一个下降沿的电平跳变时，RC531 产生复位；低电平时芯片正常工作。

(4)晶振电路

OSCIN(1 脚)、OSCOUT(32 脚)为晶振电路的输入、输出引脚，连接外部的 13.56MHz 晶振。当采用有源晶振时，也可以直接从 OSCIN 输入。

(5)SPI 接口

A0(22 脚)、A2(24 脚)、D0(13 脚)和 ALE(21 脚)分别作为 SPI 接口的 MOSI、SCK、MISO 和 NSS 引脚。

3. 模块微控制器MC68HC908JB8

目前市场上的主流 MCU 有 Freescale 系列、51 系列、AVR 系列、PIC 系列和 NEC 系列等。如此琳琅满目、让人眼花缭乱的 MCU 品种，给 MCU 应用的工程师提供了巨大的选择空间。相比较而言，美国公司的 MCU 技术仍处在领先的地位，特别是在

高端产品方面，高性能的 MCU 新产品不断推出，而日本和韩国公司的 MCU 产品在价格上占有一定优势。在研究生学习期间，作者使用 Freescale 公司的 MC68HC908GP32、MC68HC908JB8 和 MC9S12DG128 等 MCU 开发过不同的项目，熟悉 Freescale 公司 08 系列和 S12 系列 MCU 的性能、指令系统和开发工具。并且，在 Freescale 公司的官方网站可以免费申请样片和下载芯片手册。因此，选用 Freescale 系列 MCU 在开发周期和成本上都具有优势。根据系统需求，本文使用 Freescale 08 系列中的 MC68HC908JB8 作为读写模块硬件中间件中的模块控制 MCU。

MC68HC908JB8 是 Freescale 公司 08 系列的一款低价位、高性能的 MCU，其主要特点是内嵌了低速 USB 模块，同 USB1.1 协议兼容，支持 1.5Mbps 的传输速率[29]。JB8 总线频率达 3MHz，并且拥有 256 字节 RAM，8K 字节 FLASH。除此之外，芯片内部集成定时器、键盘中断、看门狗等功能模块，有多种封装形式，可用 I/O 引脚最多达 37 个。可见，MC68HC908JB8 的性能和资源完全可以满足控制射频基站芯片以及提供通信接口的需求。

3.1.3 读写模块硬件电路设计

1. 封装设计

读写模块的硬件设计即要考虑到 PCB 布板的大小，又要考虑到接口的通用性。通过多次比较所用元件和一些标准封装的的尺寸，并考虑到使用方便，本文将读写模块设计成标准的 DIP40 封装，并给出了引脚定义，如图 3-4 所示。设计中采用双面 PCB 布板，尽量选用尺寸较小的贴片元件并双面放置，如 MC68HC908JB8 就选用了 QFP44 封装。这样，开发者要使用读写模块硬件中间件时，只

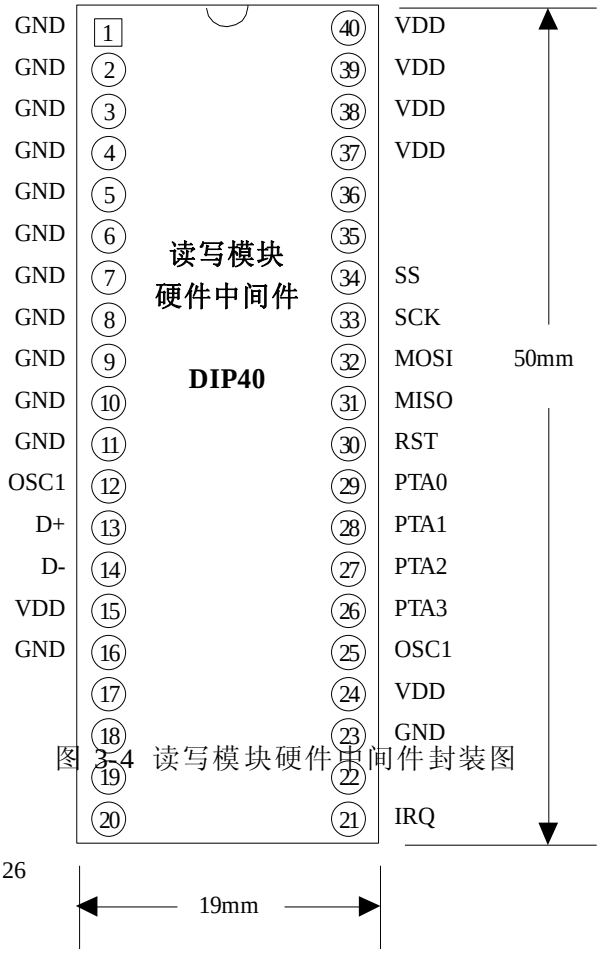


图 3-4 读写模块硬件中间件封装图

要将目标板的底座引脚按图 3-4 定义，读写模块硬件中间件就可以直接插到目标板上很方便的使用，实物图参见附录 B。

读写模块硬件中间件的引脚分为以下几类：

(1)电源

VDD、GND 接外部+5V 直流电源和地，为提高稳定性，增加多个冗余引脚。

(2)程序下载接口

21 脚～30 脚为 JB8 的 MON08 接口，供程序下载用。

(3)SPI 接口

31 脚～34 脚的 MISO、MOSI、SCK 和 SS 为读写模块的 SPI 从机接口。可以通过该接口获取读写模块中存储的电子标签信息，或者发送命令控制读写模块的操作。

(4)USB 接口

13 脚～16 脚的 D+、D-、VDD 和 GND 是 JB8 的 USB1.1 通信接口。

(5)复位与外部中断

MC68HC908JB8 正常工作时 RST 引脚为高电平，当该引脚产生一个下降沿电平跳变时 MC68HC908JB8 产生复位；IRQ 引脚为 MC68HC908JB8 提供外部中断信号，电平跳变下降沿产生中断。

限于篇幅，下文仅分析各功能模块，详细原理图参见附录 A.1。

2. MC68HC908JB8最小系统

嵌入式设计中，仅仅有一个 MCU 是无法正常工作的，它必须与相应的外围电路一起，才能构成一个最小系统。如图 3-5 所示，44 引脚 QPF 封装的 MC68HC908JB8 最小系统包括电源电路、晶振电路和复位电路[30]。

(1)电源

VDD、VSS 接外部+5V 直流电源和地，为芯片提供工作电源。接在电源和地之间的 0.1μF 为滤波电容，作用是去除电源噪声，在 PCB 布板时应尽可能地靠近芯片引脚。

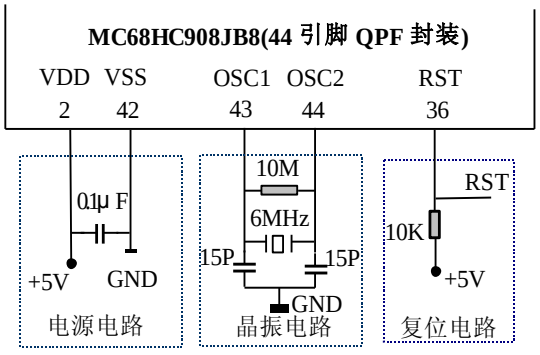


图 3-5 MC68HC908JB8 最小系统电路图

(2)晶振电路

晶振电路的设计参考 Freescale 公

司的 MC68HC908JB8 芯片手册，选用 6MHz 的石英晶振。

(3)复位电路

MC68HC908JB8 正常工作时，RST 复位引脚通过 10KΩ 电阻上拉到高电平。同时，该引脚复接到模块外部封装引脚，外部控制模块可以通过改变该引脚的电平状态，控制 MC68HC908JB8 复位。

3. 微控制器与射频基站芯片的连接

这是读写模块硬件中间件的核心部分。控制芯片 MC68HC908JB8 连接射频基站芯片 MF RC531，并控制其完成读取电子标签 UID 信息的工作。

MF RC531 与 MCU 的通信模式有 8 位并行接口和 SPI 两种，并且可以和个人电脑的 EPP(Enhanced Paralell Port，增强型并行端口)直接连接。在每次上电或硬件复位后，MF RC531 也复位其接口模式，并根据相关控制脚的逻辑电平识别当前 MCU 接口类型，这个过程是通过固定引脚连接的组合和一个专门的初始化程序来实现的[27]。

SPI 通信接口是由 Motorola 公司开发，用于在 MCU 和外围设备芯片之间提供一个低成本、易使用的接口，现已发展成为一种工业标准。它是一种高速、全双工和同步的通信总线，并且在芯片的引脚上只占用四根线，节约了引脚资源，同时为 PCB 的布局上节省空间。因此本文设计采用 SPI 接口来实现 MC68HC908JB8 与 MF RC531 连接，如图 3-6 所示。

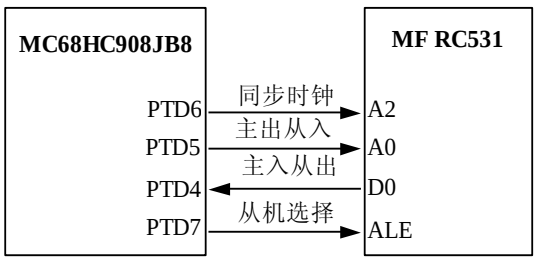


图 3-6 控制芯片 JB8 与射频芯片 RC531 的 SPI 连接

SPI 通信以主从方式工作，有主机(master)和从机(slave)的概念，至少需要 4 根线，主出从入(MOSI)、主入从出(MISO)、同步时钟(SCK)和从机选择(SS)。每次 SPI 通信双方交互 8 位数据，主机首先通过 SS 引脚将从机选中，然后从 SCK 引脚产生 8 个时钟信号，在该信号的控制下，主机的数据依次从 MOSI 引脚送出，同时从机的数据也通过 MISO 引脚传到主机中，通信的双方完成了数据交换。

MF RC531 支持 SPI 通信方式，在通信期间作为从机。要使 RC531 上电后能检测到 SPI 通信模式，并能正常的进行 SPI 通信，除了图 3-6 中的 4 根引脚连接到主机外，还需要对其它的一些引脚做相应处理。比如将 A1 引脚接低电平，NRD 和 NWR 引脚

接高电平，而 D7～D1 引脚则必须悬空。

控制芯片 MC68HC908JB8 内部并没有集成 SPI 模块，其 SPI 通信功能是在通用 I/O 口上以软件模拟的方式完成的。MC68HC908JB8 芯片供电电压为 5V，但是其 I/O 口输出电平只能达到 3.3V。为了增加引脚驱动能力并减少芯片功耗，在硬件设计时，特别选取了 MC68HC908JB8 中具有开漏输出(Open Drain)功能的 D 口作为 SPI 模块的接口。并且 SPI 接口中作为输出的引脚需要通过外部上拉电阻接+5V 来保持高电平，从机选择引脚的连接方式如图3-7所示，而主出从入和同步时钟引脚的接法与此相同。

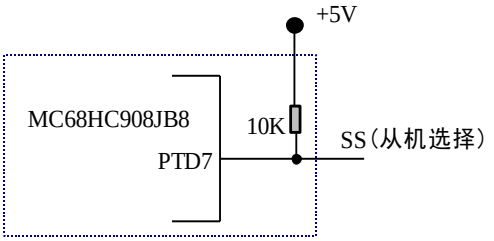


图 3-7 JB8 开漏输出引脚的上拉电阻连接方式

4. 天线设计

RFID 系统中，射频基站芯片如 MF RC531 通过天线发射能量，并与电子标签进行数据通信。因此，天线的设计的对通信的稳定性非常重要，其基本要求是[31][32]：

(1)最大化天线电路电流，以产生足够大的磁通量；

(2)功率匹配，以最大程度利用产生的磁通的可用能量；

(3)足够带宽，以保证载波信号的无失真传输。

RC531 不需要外加的功率放大器时，读写距离可以达到 10cm。根据具体的应用的不同，有两种不同的天线和匹配电路可以选择。如果电缆的长度大于 30mm，则使用 50Ω 匹配天线；而当电缆长度小于 30mm 时，则选择直接匹配天线来构造一个紧凑的读写器系统，如手持式读写器。读写器实际可以达到的读写距离依赖于读写器天线的尺寸，天线的匹配电路的性能和周围环境等因素。

本文采用直接匹配天线，电路设计如图 3-8 所示，包含以下三个部分：

(1)接收电路

为了减少干扰，RC531 使用内部 VMID 电压作为接收电路的参考电压，通过电容 C8 接地。在 RX 接收引脚和 VMID 引脚之间需要连接分压电阻 R6 和 R7，而读写天线和分压电路之间还要串联一个电容 C9。

(2)EMC 滤波电路

RC531 的工作频率是 13.56MHz，由石英振荡器产生，其中含有很多高次谐波。为了达到合格的 EMC(Electro Magnetic Compatibility，电磁兼容)标准，高次谐波必须被滤除。除了设计多层线路板改善 EMC 外，还必须使用低通滤波器。

(3)天线匹配电路

天线线圈被制作成 PCB 板，并由其电感值决定电容 C1、C2、C3、C4 的参数[33]。并且由于每块天线 PCB 板天线线圈的电感值总是会稍有差异，因此在天线匹配电路上设置了一个可调电容 C0，通过调整可调电容值将每块天线板的读写距离调整到最佳[34]。

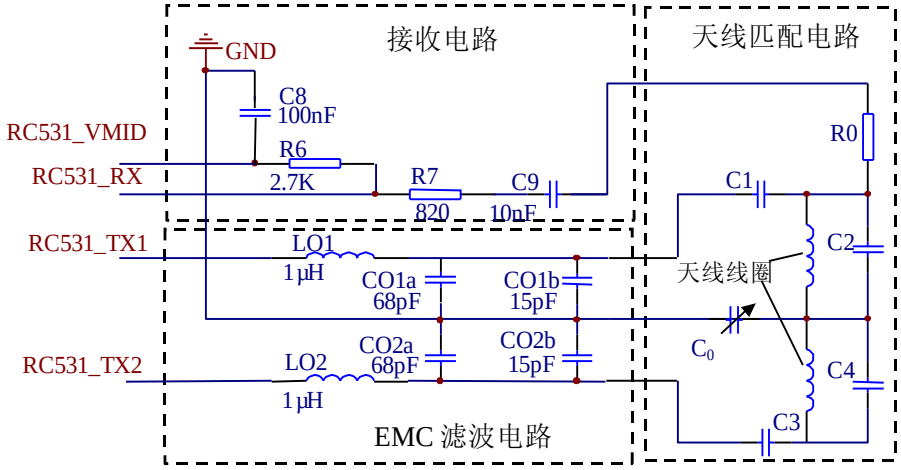


图 3-8 射频天线电路原理图

5. 通信接口

读写模块硬件中间件对外提供 USB 和 SPI 两种通信接口。其中 USB 通信接口使用 MC68HC908JB8 芯片的 USB1.1 模块，具有 1.5Mbps 的数据传送速率，可与其它支持 USB1.1 的设备如 PC 机进行通信；而 SPI 接口则需要使用 I/O 口模拟，实现 SPI 从机功能，为了提高通信的实时性，选用 MC68HC908JB8 芯片中具有键盘中断功能的 PTA 口来完成，具体实现见软件设计部分。

3.2 主控系统硬件设计

主控系统与读写模块通过 SPI 接口连接，获取该模块控制芯片内存中存储的电子标签信息，然后通过以太网通信接口将信息传送给计算机管理系统处理。因此，主控

系统的硬件设计首先要实现以太网通信硬件接口；然后是与读写模块的 SPI 通信接口；为了适应单机使用，还设计了 SCI 接口与 PC 进行串行通信，以及 LCD 等外围设备电路。主控系统的详细原理图参见附录 A.2。

3.2.1 以太网接入解决方案的选择

随着嵌入式技术与以太网技术的飞速发展，越来越多的嵌入式系统采用了以太网接口技术，将嵌入式设备接入到局域网中。这样可以利用已有的网络基础设施，节约布线成本，并提高了嵌入式系统的通信速度和传输距离。

目前可行的解决方案有下面两种[35][36]：

(1)由不集成以太网通信模块的通用 MCU 和专用以太网控制器，如 RTL8019、DM9008、AX88796L、CS8900A 或 LAN91C111 等组成，MCU 和以太网控制器通过总线方式连接。这种方案不受 MCU 功能的限制，通用性强，但是接口电路复杂，体积较大，而且整体价格比较昂贵。

(2)选择带有以太网接口的 MCU 实现单芯片接入方案。这种方案与多芯片的网络连接实现方案相比，具有设计简单，成本低廉，开发周期短等优点。

经过比较，本文采用第二种以太网接入解决方案，主控 MCU 选用 Freescale 公司的 16 位微控制器 MC9S12NE64。

3.2.2 主控芯片简介

MC9S12NE64 是 Freescale 公司 2004 年年底推出的 S12 系列 16 位 MCU 中的一款应用于以太网连接的产品。其内部集成 EMAC(Ethernet Media Access Controller，以太网媒体访问控制器)和 EPHY(Ethernet Physical Transceiver，以太网物理层收发器)，可配合第三方 TCP/IP 协议栈实现以太网的通信功能，从而实现单芯片的以太网连接方案。

MC9S12NE64 的 EMAC 模块和 EPHY 模块结构与接口如图 3-9 所示。其中 EMAC 模块和 EPHY 模块可以分开使用，通过 MII(Medium Independent Interface，介质无关接口)接口的 18 个引脚与外部 EMAC 模块和 EPHY 模块通信。本文设计中，EMAC 模块和 EPHY 模块均采用内部集成模块，省去了 MII 引脚，与外部只使用 RXN、RXP、TXN 和 TXP 这 4 根信号线进行通信。

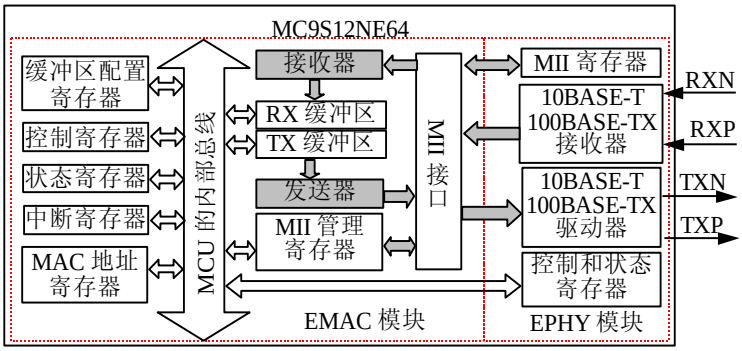


图 3-9 NE64 的 EMAC 模块和 EPHY 模块结构框图

(1)EMAC 模块

EMAC 模块兼容 IEEE802.3 标准，具有 10Mbps/100Mbps 的传输能力，实现了数据链路层的功能。该模块具有 MII 接口及其管理功能；支持全双工和半双工模式；使用 PAUSE 帧实现流量控制；支持 MAC 地址识别；具有两个接收缓冲区和一个发送缓冲区。

(2)EPHY 模块

EPHY 模块是物理接口收发器。支持 IEEE802.3 定义的 MII 介质无关接口。MII 层定义了 MAC 和各种物理层之间的标准电气和机械接口，它们使得 EPHY 与 EMAC 之间的数据通信、EPHY 的配置及通信状态的判断得以实现。EPHY 模块一端与 EMAC 通信，另一端与传输介质进行通信。该模块需要 25MHz 的晶振提供工作频率；可在半双工或全双工模式下工作；支持 10Mbps/100Mbps 速度；自动协商功能等。

除此之外，MC9S12NE64 还包含 64KB 的 FLASH 空间和 8KB 的 RAM 空间；2 个 SCI 模块，1 个 SPI 模块和 1 个 I^2C 模块，可用于芯片间通信；1 个 4 通道 16 位的时钟模块；1 个 8 通道 10 位的 A/D 模块。最大工作频率达 50MHz，拥有丰富的 IO 资源，112 脚封装版本中的 IO 引脚总数有 70 个[37]。因此，选用 MC9S12NE64 可以满足主控系统对控制芯片功能和资源的需求。

3.2.3 电源电路设计

对嵌入式设计来说，电源电路的好坏，直接影响到系统的稳定性和可靠性。设计一个实际的嵌入式系统电源部分时，需要综合芯片供电电压、工作电流和整体功耗等各项因素。

主控系统中，MC9S12NE64 的芯片工作电压为 DC 3.3V，而其他部分电路则工作

在 DC 5V。因此，可以使用外部 DC 5V 电源给主控系统供电，再设计一个电压转换电路将 DC 5V 转换成 DC 3.3V，为 MC9S12NE64 芯片提供工作电压。

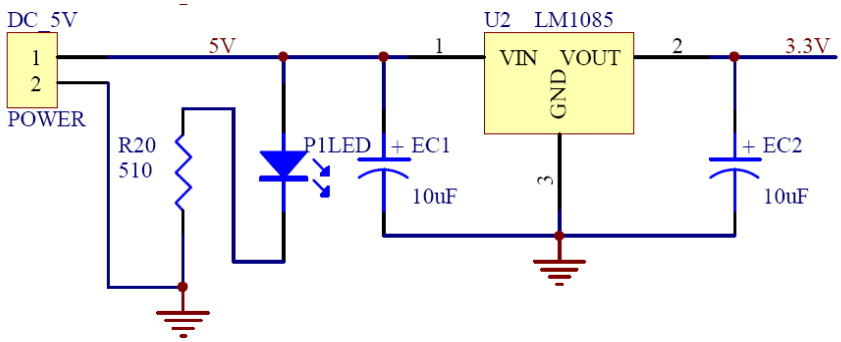


图 3-10 电源电路设计

根据 MC9S12NE64 芯片手册电气特性部分说明，该芯片运行时最大电流为 285mA。本文设计采用美国国家半导体公司的一款单芯片集成的电压转换器 LM1085。如图 3-10 所示，LM1085 将输入的 5V 电源转为 3.3V，并能提供最大 3A 电流输出，完全满足了主控系统中 3.3V 电路部分对电流的需求。同时，该芯片还具备过流保护，过温保护等功能，因此具有可靠的工作性能和较高的工作效率，为系统工作提供了强有力的保证。设计时还增加了发光二极管 P1LED 作为电源指示灯，为硬件测试提供方便。

3.2.4 MC9S12NE64 最小系统

与读写模块硬件中间件中的控制芯片 MC68HC908JB8 相比，MC9S12NE64 的最小系统要相对复杂一些，包含电源、PLL 电路、晶振电路和 BDM 电路[37]，如图 3-11 所示。

(1)电源

VDDR 接 3.3V 电源，为芯片提供工作电压；VDDX1 和 VDDX2 接电源，VSSX1 和 VSSX2 接地，为 I/O 口提供驱动电压。在电源和地之间接上 0.1 μF 的滤波电容以去除噪声。

(2)PLL 电路

芯片内部的 PLL(Phase Locked Loop，锁相环)电路具有放大频率的功能，可以将较低的外部时钟信号变换成较高的工作频率。在 PCB 布板时，为了减少电磁干扰，可以用地线将该电路圈起来。

(3)晶振电路

选用有源晶振，其 OUT 引脚可以直接输出时钟信号。由于本课题使用 BDM 写入器为 MC9S12NE64 下载程序，该 BDM 写入器要求芯片工作频率小于 20MHz 才能正确识别芯片，而 MC9S12NE64 的 EPHY 模块要求工作频率为 25MHz，所以设计了晶振选择电路，在写入时使用 8MHz 晶振，正常工作时使用 25MHz。晶振电路对芯片工作的稳定性非常重要，PCB 布板时应尽量靠近 MCU，并且与其他信号线隔离。

(4)复位电路

MC9S12NE64 正常工作时，$\overline{RESET}$ 引脚上拉到+3.3V，当该引脚接地时，芯片复位。

(5)BDM 电路

BDM(Background Debug Mode，背景调试模式)是 Freescale 公司定义的一种底层软硬件调试方法。可以通过 BDM 向目标 MCU 下载程序；读取 CPU 各个寄存器的内容；执行单片机内部资源的配置、程序的加密处理等操作。

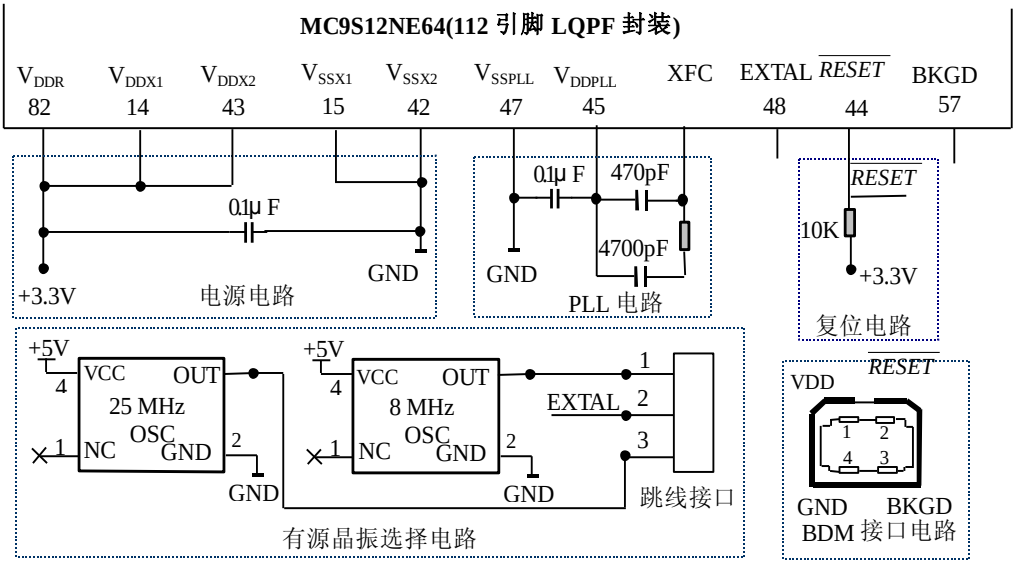


图 3-11 MC9S12NE64 最小系统

3.2.5 以太网接口硬件设计

MC9S12NE64 的 EPHY 模块采用 CMOS 工艺，由于电平信号的不同以及网络冲击信号的存在，不能直接连接到 RJ45 接口，两者之间需要添加隔离及电平信号转换的元件。

为了简化电路，在元件选型时采用了内置以太网隔离变压器的 RJ45 接口产品 PRJ-005A，其原理图如 3-12 所示[38][39]。其中,以太网隔离变压器起信号传输、阻抗匹配、波形修复、杂波抑制以及高电压隔离等作用，以保护系统的安全。MC9S12NE64 的 RXN 和 RXP 接收接口，TXN 和 TXP 发送接口分别接到 PRJ-005A 的 RD－/RD＋、TD－/TD＋上。经变压器变换后引到 RJ45 的 RX－/RX＋、TX－/TX＋引脚，以标准的 RJ45 接口与以太网相连接。

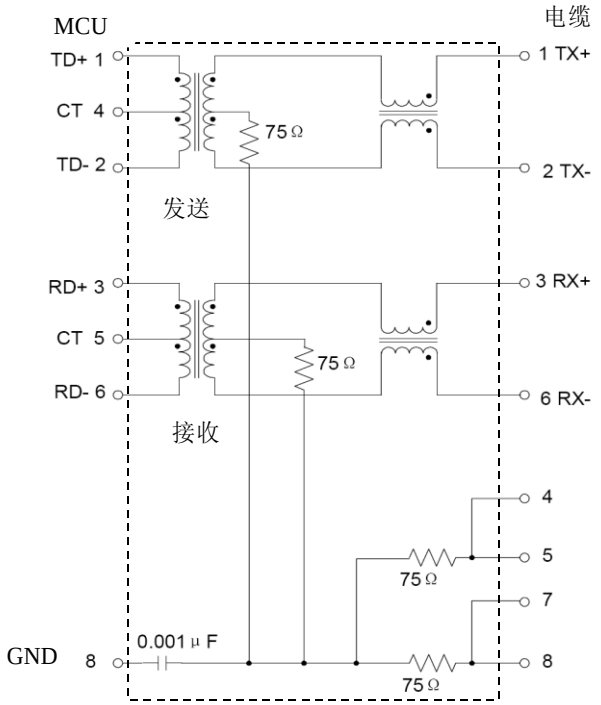


图 3-12 PRJ-005A 内部原理图

3.2.6 SPI 通信接口

本模块完成与读写模块硬件中间件交互信息的功能。MC9S12NE64 内部集成 SPI 模块，允许 MCU 与外围设备的全双工、同步、串行通信，并可通过软件配置成 SPI 主机或 SPI 从机模式。

硬件设计时，只需将 MC9S12NE64 的 SPI_MISO(主入从出)、SPI_MOSI(主出从入)、SPI_SCK(串行时钟)和 SPI_$\overline{SS}$ (从机选择)4 个引脚与读写模块底座的 HC_MISO、HC_MOSI、HC_SCK 和 HC_SS 相连接即可。

3.2.7 SCI 通信接口

目前几乎所有的台式电脑都配置了 9 芯串口，因此为了扩展系统的应用范围，本课题实现了 SCI 模块，完成读写器与 PC 的串行通信。

MC9S12NE64 中有 2 个 SCI 模块，可以通过外接的 SCI 电平转换电路，将 MCU 发送引脚的 TXD 和接收引脚 RXD 的 TTL 电平转化为 RS-232 电平，与外部串口设备通信。

外围电路中采用的 MAX3232 是 MAX232 的改进型，功耗更低。MAX232 的供电电压为 5V，耗电 5mA，需要外接 4 个 1uF 电容。而 MAX3232 供电电压可以为 5V 或 3.3V，耗电仅 0.3mA，外接 4 个 0.47uF 电容。

硬件设计时，使用 MC9S12NE64 的 SCI0 模块，将 MCU 的 SCI0_RXD 和 SCI0_TXD 分别接 MAX3232 的 R1OUT 和 T1IN。MAX3232 只起到转换电平的作用，进行 SCI 编程时只需对模块的相关寄存器进行读写操作。

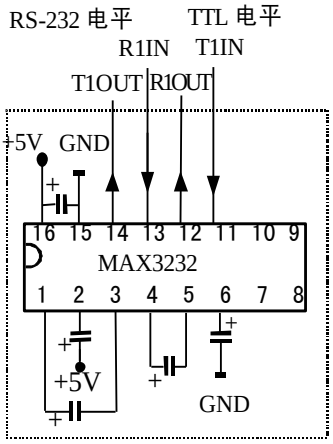


图 3-13 SCI 外围电路

3.2.8 LCD 显示电路

LCD 一般可以作为电子产品的显示器件。而 LCM(Liquid Crystal Module，液晶显示模块)通常包括 LCD 显示及驱动电路，接口电路等，往往做成一个模块的形式。设计中，选用台湾 DATA IMAGE 公司 LCM 汉字液晶模块 GX123200。该 LCM 像素点阵为 122×32，可视区域为 60.6×20(mm)，STN(Super Twisted Nematic，超扭曲向列型)黄绿屏幕。若显示 8×16 点阵的字符，一行最多可显示 15 个字符，最多可显示两行。

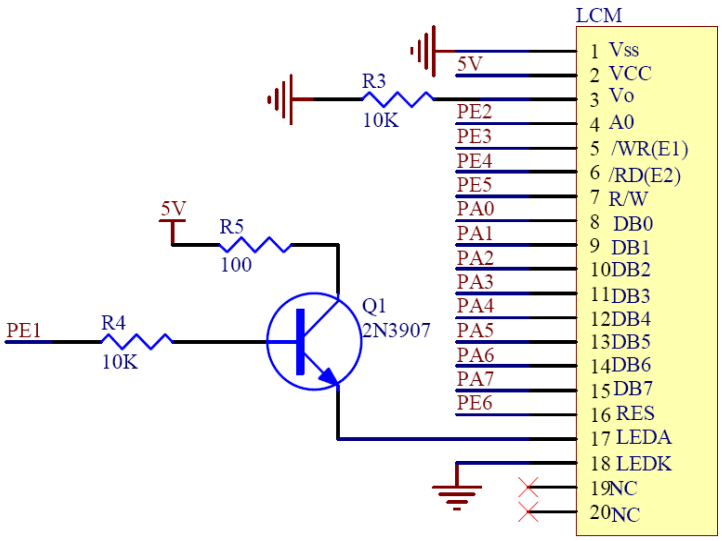


图 3-14 LCD 显示电路

MC9S12NE64 与 LCM 的连接如图 3-14 所示。MC9S12NE64 的 E 口用于 LCM 的

控制，其中 PE2 引脚接 LCM 的 A0，控制数据/指令通道选择；PE3、PE4 分别接 LCM 的 E1、E2，用于屏幕前、后区域选择；PE5 负责读写控制，接 LCM 的 R/W 引脚；PE6 接 RES，控制 LCM 复位；PE1 接 LEDA，控制 LCM 的背光开关。MC9S12NE64 的 A 口作为数据传输接口，PA0～PA7 引脚依次接 LCM 的 DB0～DB7 端口。

3.3 硬件测试及设计体会

3.3.1 硬件模块测试

稳定可靠的硬件是系统正常运行的前提。因此，在硬件设计完成后，要对各个功能模块分别进行测试以发现存在的问题。硬件测试需要使用万用表、示波器等工具，并结合模块驱动程序来完成。

整个系统的主要模块硬件测试按以下步骤进行：

(1)电源模块测试

在 PCB 板上焊接元件时，首先要完成电源电路部分，并进行检测。在确认电源电路正常后再焊接其他模块的元件，否则可能因为电源电路的故障而将系统的硬件电路烧坏。

(2)MC68HC908JB8 与 MC9S12NE64 最小系统测试

在 MC68HC908JB8 的最小系统焊接完成后，首先通过 MON08 接口检测是否能够正常的下载程序，以判别芯片是否良好；然后对芯片供电，测量 VDD、VSS 供电引脚电压是否正确，以及 RST 引脚是否为高电平；为了方便直观的判断 JB8 是否工作正常，可以在其一个通用 I/O 口上接 LED 指示灯，并编写一个简单的小灯闪烁程序下载到 JB8 中，因为当 JB8 未初始化成功时，其 I/O 口的输出状态是未知的，只是将该 I/O 口一直输出高电平或低电平并不能确定 MCU 工作正常；如果 JB8 供电部分正确，但是工作不正常，则需要使用示波器来检测其晶振电路的频率是否符合要求。

MC9S12NE64 最小系统的程序写入部分使用了 BDM 接口电路，测试步骤与 JB8 相同。

(3)MC9S12NE64 的 SCI 电路测试

SCI 电路相对简单，只要接线和元件正确就可以正常工作。首先在 NE64 和 PC 上分别编写串口通信程序，测试时 PC 主动向 NE64 发送数据，NE64 在接收到一个字节数据后立即将该数据返回给 PC。通过判断 PC 发送和接收的数据是否相符来判

断 SCI 电路工作是否正确。在最小系统完成后立即测试 SCI 电路，是考虑到 SCI 模块作为系统的显式数据输出，可以方便的观察 MC9S12NE64 的寄存器和程序变量的数值变化，成为调试其他模块的工具。

(4)以太网接口测试

由于采用了集成以太网隔离变压器的 RJ45 接口，以太网接口部分接线也比较简单，检查完线路后，功能测试需要通过相应的驱动程序并配合 SCI 模块来完成。详细的测试步骤参见软件设计中的以太网通信部分。

(5)MC9S12NE64 与 MC68HC908JB8 的 SPI 接口

为了测试读写模块硬件中间件，首先要完成 MC9S12NE64 与 MC68HC908JB8 之间的 SPI 通信。其中，MC9S12NE64 集成 SPI 模块，并作为 SPI 主机；MC68HC908JB8 通过 I/O 口模拟 SPI 从机。在编写 SPI 通信程序之前，要确定通信的一方是否能检测到对方的电平变化。SPI 通信中使用到 4 条线路，对其中一条线路进行测试时，要根据 SPI 通信中数据的传输方向，由输出方同时交替变化该线路和本方运行指示灯电平电压；输入方检测该线路电平，并将该引脚状态赋给其自身的运行指示灯。如果双方的运行指示灯同步闪烁，则说明电平检测成功。当 4 条线路都测试通过后，就可以开始 SPI 的编程。

其他部分电路不是很复杂，测试过程不再敷述。

3.3.2 硬件设计体会

经过研究生三年的学习，对硬件电路的设计有如下体会：

(1)需求分析

在进行硬件设计之前，首先要明确系统的功能，否则设计的系统可能和实际需求大相径庭。

(2)原理图设计

将整个系统按功能划分成各个子模块，对每个模块独立分析，画出原理图。任何模块都要做成一个包含输入和输出的系统，设计之后写一段原理性描述。电源电路的设计一定要考虑到整个系统的功耗。对一些成熟的电路，可以进行模仿借鉴；设计新电路时，一定要明确需求，对器件进行合理选型，可以先在面包板又称电路试验板上搭建电路，然后通过测试不断完善。

(2)PCB 板设计

PCB 布板时首先要考虑外形尺寸，然后将每个模块独立放置，用虚线框框好，并标注模块名，使整个 PCB 板分割清晰。考虑到电磁兼容性，芯片的电源引脚要接滤波电容，高频电路应与其他电路隔离，芯片的晶振电路元件应尽可能的靠近芯片引脚。电气上对一些重要的信号线可以进行复接，减少开路故障的发生，电源和地线尽可能宽。为了硬件测试和系统扩展的需求，要多留测试点和冗余接口。

(4)硬件测试的重要性

测试如同科学，也是一门艺术[40]。如果没有一定的逻辑方法和有条理的思维，即使使用昂贵的仪器也很难发现故障所在。因此，测试的过程也是锻炼思维能力的过程，故障解决能力越强，学到的知识就越多，经验也越丰富。

3.4 本章小结

本章主要工作总结如下：

(1)将读写器划分为读写模块硬件中间件和具有以太网通信功能的主控系统两部分，降低了硬件模块之间的耦合度，以提高硬件的复用性。

(2)从应用角度出发，对芯片选型时应考虑的主要因素做了详细分析。

(3)参照中间件通用定义，提出了读写模块硬件中间件的设计思路。将读写模块设计成标准的 DIP40 封装，并给出引脚定义。当读写模块插接在主控系统的底座上后，主控系统就可以直接通过 SPI 接口与其通信，实现了硬件模块的即插即用。

(4)通过功能分析、方案选择和电路设计等步骤在主控系统中实现了单芯片的以太网连接，并对主控系统中各外围模块的电路设计一一加以阐述。

(5)对读写器硬件系统各功能模块进行充分测试，并给出了硬件设计体会。

第四章 读写器软件设计

嵌入式系统的软件设计一般都要涉及底层硬件，并且硬件驱动程序的开发是与硬件设计协同完成的，在此基础上根据实际需求实现应用软件的开发。因此，理解软、硬件之间的关系及界限有助于驱动程序和功能程序的合理划分。本章结合面向硬件编程的思想，重点讨论了读写模块如何获得 TYPE A & B 电子标签 UID 以及主控系统以太网通信功能的实现。

由于设计中采用了两款不同的 MCU，因此分别采用实验室自主开发的 MT-IDE for Freescale HC08 和 Metrowerks 公司的 CodeWarrior IDE 4.6 两种集成开发环境编写 MC68HC908JB8 与 MC9S12NE64 的软件程序。而为了提高程序的易读性和可移植性，全部代码使用标准 C 语言实现。

4.1 读写模块

4.1.1 MC68HC908JB8 工程文件

图 4-1 JB8 的 C 程序工程在 MT-IDE 下的文件视图

MC68HC908JB8 实现读取 TYPE A & B 电子标签 UID 在 MT-IDE 开发环境下的 C

语言工程实例的文件组织如图 4-1 所示。整个工程设计中遵循面向硬件编程的思想。首先对每个硬件对象编写独立的.c 和.h 文件，并以该硬件名称作为文件名，如 SPI.c、SPI.h；然后将对硬件对象的直接操作封装成函数放在相应的.c 文件中，并且在.h 文件中声明，如 SPI0 模块初始化函数 void SPI0Init(void)，供外部程序调用。

4.1.2 主函数设计

读写模块被设计成硬件中间件，为应用系统和电子标签提供数据交互的接口。模块微控制器 MC68HC908JB8 的主函数流程图见图 4-2。

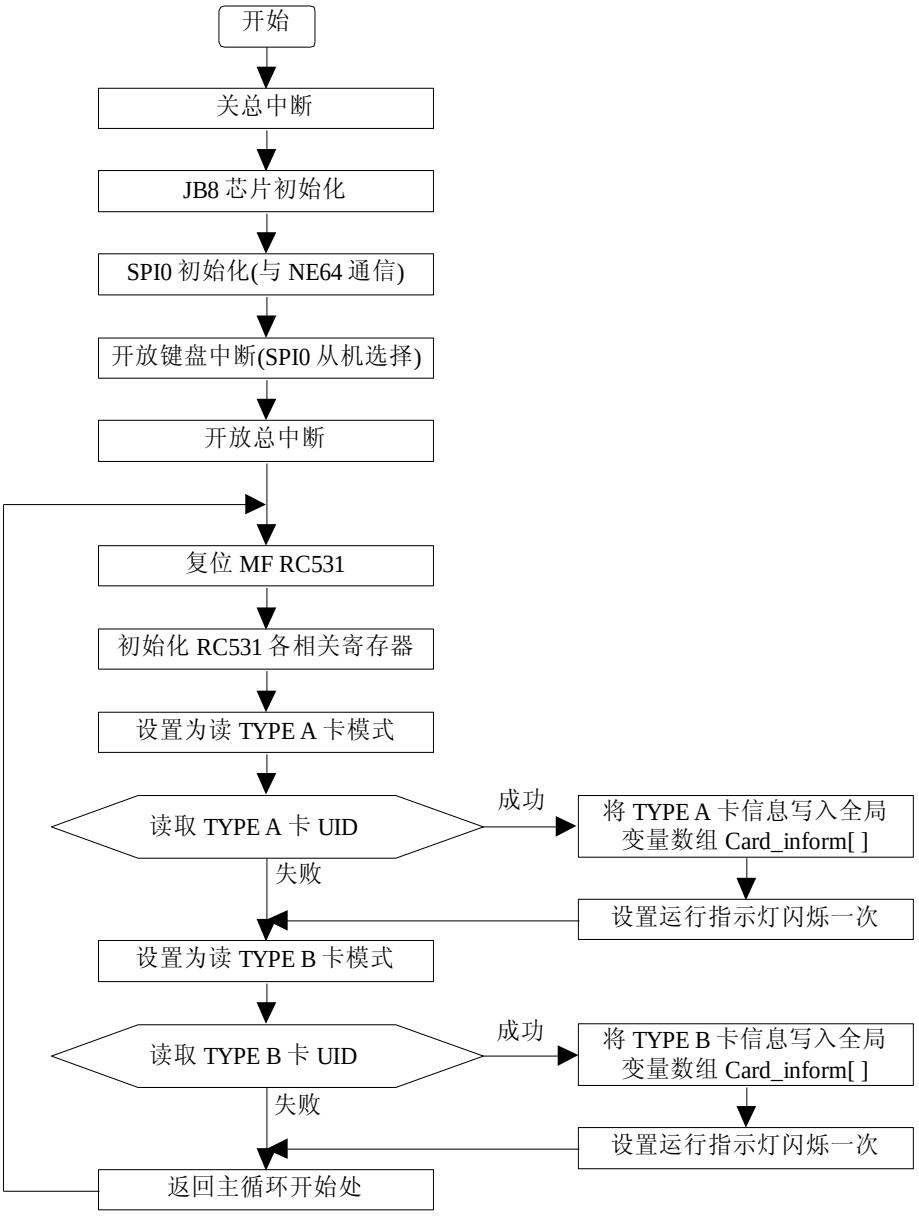


图 4-2 读写模块主函数流程图

系统上电后首先执行 MCU 以及相关模块的初始化操作，然后进入主循环。嵌入式系统的软件一般由主循环和若干中断服务例程组成，并且主循环通常被设计成无限循环一直运行下去，就如同操作系统一样。如图 4-2 所示，主循环中不断的切换读卡模式，满足读取 TYPE A & B 两种电子标签 UID 的需求。一旦读取成功，将 UID 赋给全局字节型数组变量 Card_inform[ ]，Card_inform[0]存放电子标签的类型，即“A”或“B”的 ASCII 码，其后依次存放该类型的 UID。主控系统可以通过读写模块硬件中间件的 SPI 通信接口获取 Card_inform[ ]中的内容，获取完毕后 JB8 将 Card_inform[ ]数组清零。

对 RC531 的复位和寄存器初始化操作原本放置在主循环之前，即每次 JB8 复位后只执行一次。但是在实际测试过程中，当 RC531 长时间运行后会出现不稳定的情况，因此将这两步操作放在主循环中，以提高系统的鲁棒性(Robustness，健壮性)。

4.1.3 I/O 口模拟 SPI

本文在 JB8 中用软件模拟实现了两个 SPI 模块 SPI0 和 SPI1，其中 SPI0 是读写模块提供的对外通信接口，而 SPI1 用来完成 JB8 与 RC531 的通信。

1. 读写模块SPI接口

由于 SPI 是一种同步通信方式，因此模拟时首先要确定对方的通信参数。MC9S12NE64 中 SPI 模块的工作方式、时钟极性和时钟相位等参数的初始化代码如下：

```
//SPIInit:SPI通信初始化-------------------------------------------------*
//功  能:SPI通信初始化                                                  *
//参  数:无                                                             *
//返  回:无                                                             *
//----------------------------------------------------------------------*
void SPIInit(void)
{
    SPICR1 = 0x50; //不产生中断,主机方式,时钟空闲低电平,SCK上升沿采样数据
    SPICR2 = 0x00; //
    SPIBR  = 0x77; //分频因子设定，设置波特率12.21KHz
}
```

MC68HC908JB8 的时钟频率为 3MHz，远大于 NE64 的 SPI 模块 12.21KHz 的数据传送速率，完全可以通过对通用 I/O 口的操作来模拟 SPI 从机通信。实时性是嵌入式系统的一个重要特征，中断方式显然比轮循方式有更快的响应速度。因此，读写模块硬件中间件的 SPI 模块被设计成从机模式，通过中断方式响应 SPI 主机

(MC9S12NE64)的通信请求。MC68HC908JB8 的 A 口与键盘中断模块复用，利用其中断功能完成对主机的从机选择信号的响应。

根据面向硬件对象编程的设计思想，将使用到的通用 I/O 口按 SPI0 模块引脚定义，对其所有操作封装函数形式供外部程序调用。

```
//SPI0通信寄存器及标志位定义
#define SPI0_P PTA
#define SPI0_D DDRA
#define SPI0_SS    3          //从机选择引脚位
#define SPI0_SCK   2          //时钟引脚
#define SPI0_MOSI  1          //主出从入
#define SPI0_MISO  0          //主入从出
//SPI0通信相关函数声明
void SPI0Init(void);              //SPI0通信初始化函数声明
INT8U SPI0SLAVECOM(INT8U snddata);//SPI0从机通信程序
```

SPI 从机方式通信子程序的流程图如图 4-3 所示。每次 SPI 通信主机与从机之间完成 1 个字节(8 位)的数据交换。其中，将 JB8 的发送数据提前上线是为了让主机 NE64 能及时采样到主入从出(MISO)线路上的信号。

电子标签的 UID 信息存储在 JB8 内存的全局字节型数组变量 Card_inform[ ]中。考虑到同步通信方式的特点，主机 NE64 首先向从机发送所需数据在该数组中的下标值，而从机 JB8 返回上次通信时主机请求的数据，并根据接收的下标值确定下次发送的数据。例如，当主机需要获取电子标签类型即 Card_inform[0]的数据时，需要连续向从机发送两次 0x00，取第二次的接收数据。这种数据通信格式接口清晰，操作简易，缺点是效率不高。

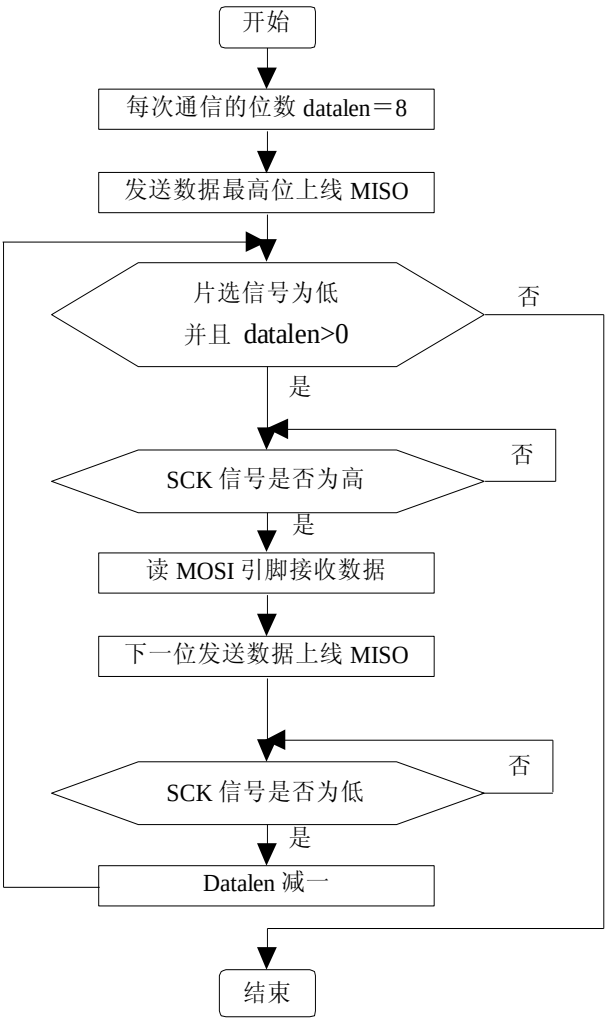


图 4-3 JB8SPI 从机程序流程图

2. MC68HC908JB8与射频芯片RC531的SPI通信

JB8 通过模拟的 SPI1 通信接口发送命令控制 MF RC531 工作并接收返回的电子标签信息，RC531 芯片集成 SPI 模块通信时作为从机[28]。图 4-4 描述了 RC531 芯片的 SPI 时序图，通过分析可以发现：主机时钟信号 SCK 空闲时需要保持低电平；RC531 在 SCK 的上升沿采样数据；通信时高位优先。

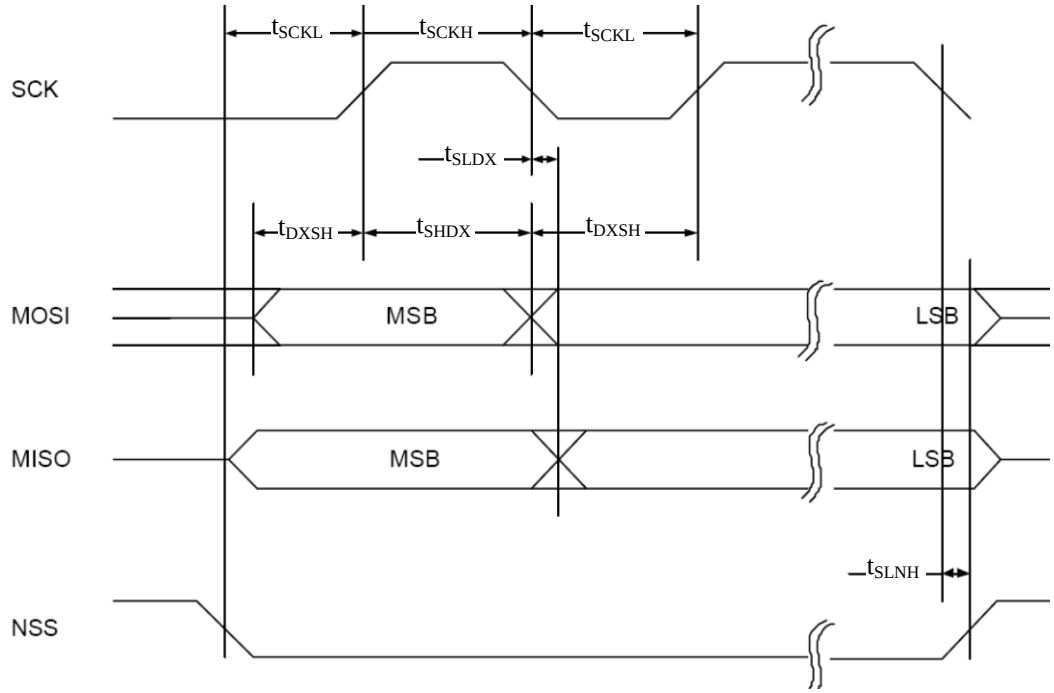


图 4-4 RC531 的 SPI 时序图

SPI 时序图中的时间参数要求见表 4-1。

表 4-1 RC531 的 SPI 时序参数表

名称	说明	最小值	最大值	单位
t_{SCKL}	SCK 脉冲低电平宽度	100	–	ns
t_{SCKH}	SCK 脉冲高电平宽度	100	–	ns
t_{SHDX}	SCK 脉冲高电平到数据变化	20	–	ns
t_{DXSH}	数据变化到 SCK 脉冲高电平	20	–	ns
t_{SLDX}	SCK 脉冲低电平到数据变化	–	15	ns
t_{SLNH}	SCK 脉冲低电平到 NSS 片选信号高	20	–	ns

JB8 的总线频率为 3MHz，即一个时钟周期 333ns，大于时序表最小值中的最长时间 100ns，因此在使用 I/O 口模拟 JB8 的 SPI 主机程序时，不需要添加延时等待操作。JB8 作为 SPI 主机与 RC531 进行通信，其程序实现与图 4-3 描述的 SPI 从机方式流程图类似，只是不需要将发送数据提前上线，在此不再敷述。

4.1.4 RC531 驱动

驱动程序提供了从硬件到应用软件的一个接口。RFID 应用程序通过 RC531 的驱动程序接口，控制射频芯片的工作，完成读取 ISO 14443 TYPE A & B 电子标签相关信息的功能。

1. 读写寄存器

对寄存器的读写操作是完成其他控制功能的基础。射频芯片 MF RC531 的内部共有 64 个寄存器寄存器，按功能分为 8 页，每页 8 个寄存器[28]。其寄存器的寻址模式分为页模式和线性模式。页模式通过页地址和页内偏移地址的组合来确定寄存器，当需要访问的下一个寄存器不在当前页时，需要先切换页，然后才能访问。而线性模式可以直接访问线性地址 0x00 到 0x3F 中的所有 64 个寄存器，因此执行效率更高。采用线性模式时，使用一个字节中的 6 位来描述寄存器地址。

读写 RC531 寄存器需要使用特定的 SPI 通信格式：

(1)读寄存器

如表 4-2 所示，SPI 主机可以连续读取 n 个 RC531 中的数据。SPI 主机向 RC531 发送需要访问的寄存器地址，并接收上次 SPI 通信时请求的数据。

表 4-2 读操作 SPI 通信方式

	字节 0	字节 1	字节 2	……	字节 n	字节 n+1
主出从入	地址 0	地址 1	地址 2	……	地址 n	00
主入从出	–	数据 0	数据 1	……	数据 n-1	数据 n

主机发送的首字节格式决定了通信模式以及地址信息。当主出从入线路发送的“地址 0”的最高位为 1 时，表明目前的 SPI 通信工作在读模式；位 6～位 1 存放 6 位的线性地址；最低位置 0。其后的“地址 1”到“地址 n”的最高位必须为 0，其他格式与“地址 0”相同。结束读操作时，主机发送 0x00。

为了程序接口清晰，本文设计的读 RC531 寄存器的函数牺牲了程序执行效率。该函数每次读取一个寄存器内容，入口参数为目标寄存器地址，返回值为该寄存器内容。处理步骤如下：

①将入口参数(目标地址)按“地址 0”格式设置；

②选中从机；

③调用 JB8 的 SPI1 主机程序，发送目标地址，接收数据丢弃；

④调用 JB8 的 SPI1 主机程序，发送 0x00，接收数据为目标寄存器内容；

⑤取消从机选中；

⑥返回寄存器数值。

(2)写寄存器

与读操作类似，对 RC531 寄存器的写操作可以一次向一个目标地址写入多个数据，为操作 RC531 的 FIFO 缓冲区提供了方便，通信格式如表 4-3 所示。“地址”的最高位置 0，表明目前的 SPI 通信工作在写模式；位 6～位 1 存放 6 位的线性地址；最低位置 0。写单个寄存器的函数的设计与读操作实现函数类似，通过先发送地址再发送数值的方式来实现，结束写操作则需要中断本次的 SPI 通信。

表 4-3 写操作 SPI 通信方式

	字节 0	字节 1	字节 2	……	字节 n	字节 n+1
主出从入	地址	数据 0	数据 1	……	数据 n-1	数据 n
主入从出	–	–	–	–	–	–

读写操作提供的接口函数声明如下：

```
void RFRegWrite(INT8U addr, INT8U Data); //向给定地址的寄存器写入一个字节数据
INT8U RFRegRead(INT8U addr);             //读出给定地址的寄存器的值并返回
```

2. 操作FIFO缓冲区

MF RC531 的 FIFO(First In First Out，先入先出)缓冲区有两个重要的作用：

(1)传递命令参数

应用程序向 FIFO 缓冲区中写入命令参数，而当 RC531 执行命令时会自动访问 FIFO 缓冲区以获取相关信息。

(2)传递电子标签信息

当 RC531 接收到电子标签发送的信息后，将其存放于 FIFO 缓冲区，等待被读取。

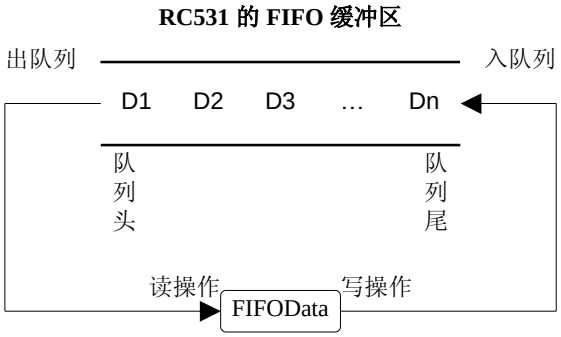


图 4-5 RC531 的 FIFO 缓冲区

FIFO 缓冲区在 RC531 的内部实现，应用程序只需读写 FIFOData 寄存器。如图 4-5 所示，数据 D1～Dn 存放在 FIFO 缓冲区中，D1 是队头元素，Dn 是队尾元素。写一个字节数据到 FIFOData 相当于执行了入队列操作，该数据被插入 Dn 之后成为新的队尾；而读取 FIFOData 则完成了一次出队列操作，队头元素 D1 退出 FIFO 缓冲区。缓冲区的长度可以通过软件设定，但最大不能超过 64 个字节。

为方便应用程序对缓冲区的操作，本文设计了三个接口函数：

```
void Write_FIFO(INT8U *Send_Buf, INT8U Length); //写length个字节数据到FIFO缓冲区
INT8U Read_FIFO(INT8U *Send_Buf); //根据缓冲区数据长度寄存器FIFOLenth的值读取FIFO中的数据
INT8U Clear_FIFO(void);           //清空FIFO缓冲区
```

3. 命令控制接口

RC531 的运行和操作是由内部的状态机来决定的，应用程序向 RC531 发送不同的命令实现状态机的状态转移，以达到控制目的。命令由命令代码和命令参数组成，向 RC531 发送命令首先要清空 FIFO 缓冲区，接着将命令参数写入到 FIFO 缓冲区中，最后再向命令寄存器写入命令代码来启动该命令。命令接口函数声明如下：

```
//Command_Send-------------------------------------------------------------*
//功  能: 向MF RC531发送命令                                                *
//参  数: Length  : 要写入的字节数                                          *
//        Send_Buf: 待发送数据应放在Send_Buf缓冲区中                        *
//        comm_set: 命令操作码                                              *
//返  回: 1 - 命令执行失败; 0 - 命令执行成功                                *
//-------------------------------------------------------------------------*
INT8U Command_Send(INT8U *Send_Buf, INT8U Length, INT8U comm_set);
```

4.1.5 TYPE A & B 电子标签的 UID 识别

读写模块的功能是识别 TYPE A & B 两种电子标签的 UID，由于标签类型的差异其识别过程的实现也有所不同。

1. 读卡模式设置

对射频基站芯片 MF RC531 的正确初始化是完成其他操作的前提。由于国际标准 ISO 14443 规定的 TYPE A 和 TYPE B 采用了不用的通信机制(见第二章表 2-2)，因此为了 RC531 能与不同类型的电子标签进行射频通信，必须在初始化 RC531 后对其相关寄存器做不同的设置。

(1)ISO14443 A 模式

对 TYPE A 类型的电子标签，RC531 需要做以下设置：

```
RFRegWrite(RegTxControl, 0x5b);          //幅移键控(ASK) 100%
RFRegWrite(RegCwConductance, 0x3f);      //设置输出驱动的电导系数
RFRegWrite(RegModConductance, 0x3f);     //调制比100%
RFRegWrite(RegCoderControl, 0x19);       //TYPE A模式,波特率106Kbit/S,米勒编码
RFRegWrite(RegTypeBFraming, 0x00);       //EGT长度为0
RFRegWrite(RegDecoderControl, 0x08);     //接收TYPE A曼彻斯特(Manchester)编码
RFRegWrite(RegRxThreshold, 0xff);        //可接收的最小信号强度
RFRegWrite(RegBPSKDemControl, 0x1e);     //忽略EOF,打开高通滤波
RFRegWrite(RegClockQControl, 0x00);      //Q时钟控制
RFRegWrite(RegRxWait, 0x06);             //设置接收延时
RFRegWrite(RegChannelRedundancy, 0x03);//TYPE A接收冗余校验
FRegWrite(RegCRCPresetLSB, 0x63);        //CRC预设值
RFRegWrite(RegCRCPresetMSB, 0x63);       //CRC预设值
```

(2)ISO14443 B 模式

如果与 TYPE B 电子标签通信，RC531 的寄存器设置如下：

```
RFRegWrite(RegTxControl, 0x4b);          //13.56MHz
RFRegWrite(RegCwConductance, 0x3f);      //设置输出驱动的电导系数
RFRegWrite(RegModConductance, 0x06);     //幅移键控(ASK) 10%
RFRegWrite(RegCoderControl, 0x20);       //TYPE B模式,波特率106Kbit/S,NRZ编码
RFRegWrite(RegTypeBFraming, 0x00);       //TYPE B帧格式
RFRegWrite(RegDecoderControl, 0x19);     //接收TYPE B,BPSK编码
RFRegWrite(RegRxThreshold, 0x44);        //可接收的最小信号强度
RFRegWrite(RegBPSKDemControl, 0x3e);     //忽略EOF,打开高通滤波
RFRegWrite(RegClockQControl, 0x00);      //Q时钟控制
RFRegWrite(RegRxWait, 0x06);             //设置接收延时
RFRegWrite(RegChannelRedundancy, 0x2c);//TYPE B接收冗余校验
FRegWrite(RegCRCPresetLSB, 0x63);        //CRC预设值
RFRegWrite(RegCRCPresetMSB, 0x63);       //CRC预设值
```

2. 状态图分析及UID识别

ISO14443 A & B 两种电子标签有着不同的状态机制，通过分析该机制可以得到获取 UID 的方法。

(1)ISO14443 A 电子标签

TYPE A 电子标签的状态图如图 4-6 所示。电子标签在进入读写器的射频场之前处于“掉电”状态；进入射频场后，电子标签获得能量并转为“空闲”状态，等待接收读写器发送的命令；当接收到询卡命令或唤醒命令后，电子标签转为“就绪”状态，在这个状态可以进行防冲突检测(多个电子标签同时响应询卡命令)操作，并获得标签的 UID 信息；经过防冲突检测后，读写器向选中的电子标签发送选卡命令，使之进入“激活”状态，此时可以进行一些高层的操作，如读写电子标签标签内信息；当电子标签接进入“挂起”状态后，只响应唤醒指令，其他操作命令对其无效。

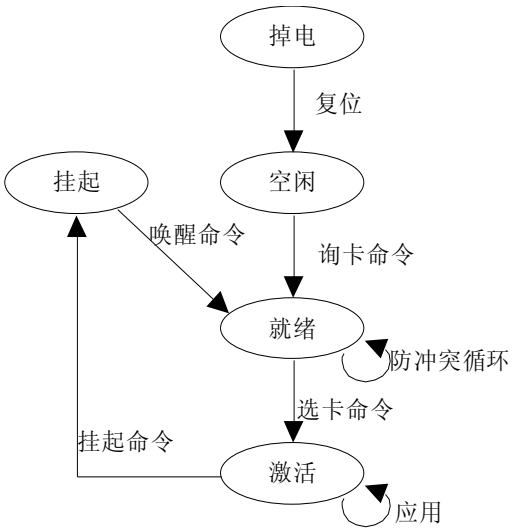


图 4-6 TYPE A 电子标签状态图

读取 TYPE A 电子标签 UID 的程序设计如下：

①读写器不断的发送询 A 卡命令 REQA；

②当 TYPE A 电子标签进入射频场后返回应答 ATQA，并进入“就绪”状态；

③读写器检验 ATQA 正确，发送防冲突指令 ANTICOLLISION，如果未出现冲突，读写器将获得 4 个字节 UID。

本文设计只实现了单卡操作，对多卡冲突不处理。测试使用的 TYPE A 电子标签为 NXP 公司的 Mifare 1 S50 卡，其 UID 为 4 字节。

(2)ISO14443 B 电子标签

与 TYPE A 相比，TYPE B 电子标签的状态图要复杂一些，如图 4-7 所示。

当 TYPE B 电子标签进入射频场后，状态由“掉电”变为“空闲”并监听询卡指令(REQB，格式与 REQA 不同)；REQB 中包含时隙参数，其范围是 1～16，当 RF 场内的多张电子标签接收到询卡指令后，每张确定一个唯一的时隙来发送应答(ATQB)，如果时隙值为 1，则发送 ATQB 并进入“声明就绪”，如果不是则进入“请求就绪”；在“请求就绪”状态，电子标签监听询卡命令和时隙命令，如果与时隙命令中的标记匹配，则转入“声明就绪”状态；“声明就绪”转到“挂起状态”的前提是接收到挂起命令；读写器通过激活命令中的 PUPI 参数指定一个电子标签进入“激活”状态；当处于激活状态时，可以进行高层的操作，也可以转移为“挂起”；“挂起”时只接收唤醒命令进入“空闲”，对其他命令不响应。

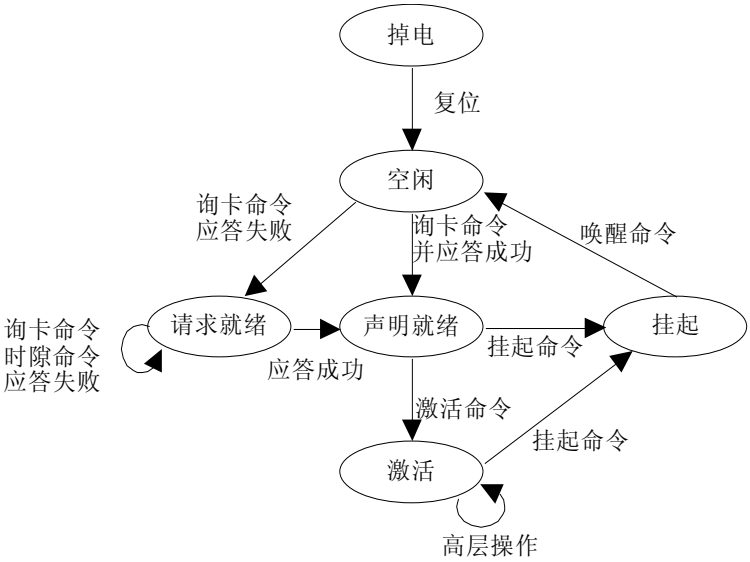


图 4-7 TYPE B 电子标签状态图

本文设计中使用的 TYPE B 电子标签是我国的“二代身份证”，其 UID 为 8 个字节。在不实现多卡冲突功能时，对单个“二代身份证”UID 读取过程设计如下：

①读写器不断的发送询 B 卡命令 REQB；

②当“二代身份证”进入射频场并接收到 REQB 后，在时隙 1 回送应答命令 ATQB，并进入“声明就绪”状态；

③读写器检验 ATQB 正确，发送激活命令 ATTRIB，“二代身份证”回送激活应答，表明成功进入了“激活状态”；

④当“二代身份证”处于激活状态时，读写器发送请求 UID 命令来获取 8 个字节的 UID。

4.2 嵌入式以太网

4.2.1 解决方案设计

MC9S12NE64 内部集成的 EMAC 和 EPHY 模块可以完成数据链路层的功能，如果要实现以太网的通信，还需要配合第三方的 TCP/IP 协议栈。

目前已经有一些开放源代码的嵌入式 TCP/IP 协议栈，如 OPENTCP、μIP、LwIP 和 μC/IP 等，这些协议栈由标准的 TCP/IP 协议栈简化而来，提供了移植接口，适用于各种处理器，但是一般需要操作系统的支持。而在绝大多数的嵌入式应用系统中，

其处理任务相对简单，而且系统资源比较有限，所以一般不使用操作系统。因此，直接移植这些协议栈并不是最合理的选择。

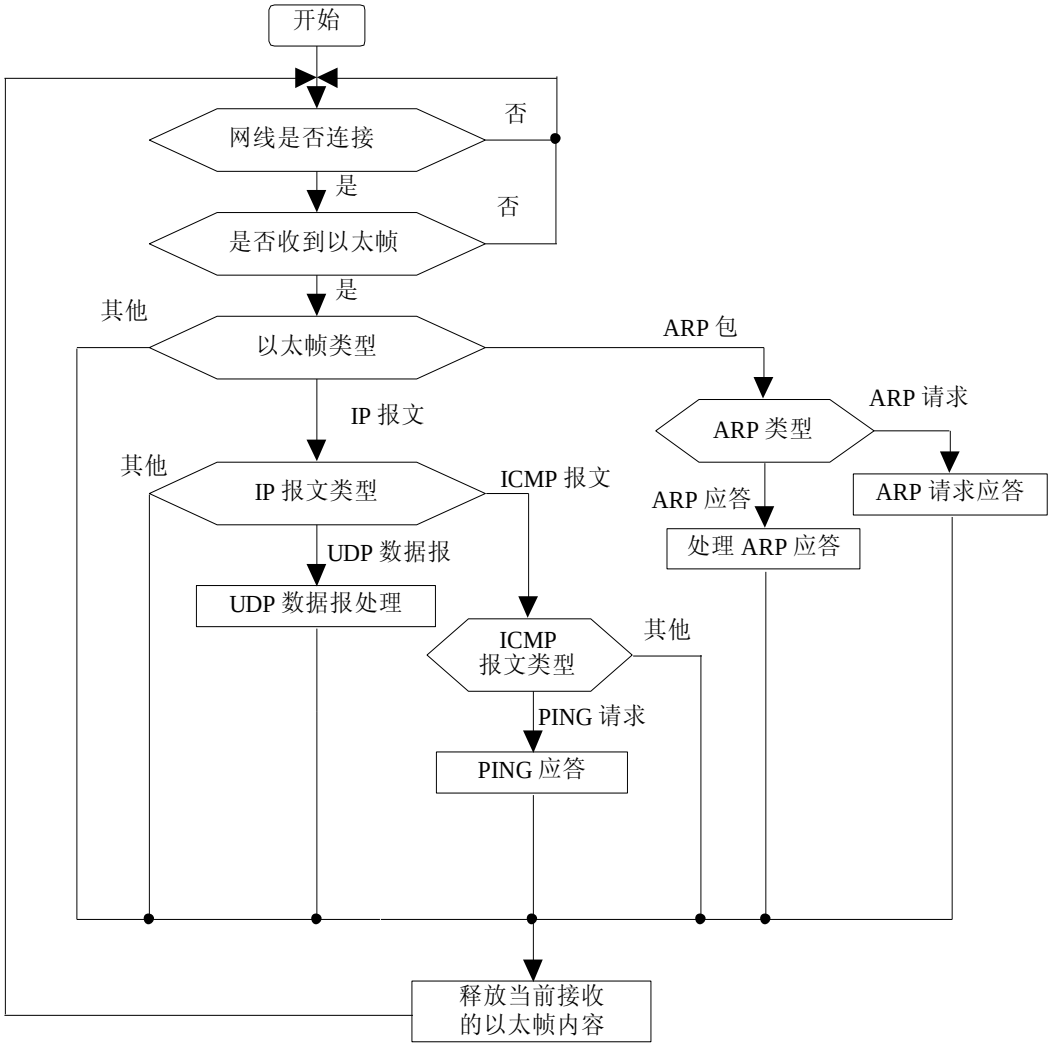


图 4-8 以太网通信流程图

标准的 TCP/IP 协议栈功能强大，协议内容复杂，而嵌入式系统一般作为终端设备来检测信号和控制对象，因此很多协议完全不需要采用，而且一些协议的功能也可以简化。在对目前成熟的一些嵌入式协议栈进行深入研究后，从嵌入式系统的特点出发，本文在完成 MC9S12NE64 以太网驱动的基础上，设计了一个无需操作系统的精简嵌入式 TCP/IP 协议栈，实现了 MC9S12NE64 的以太网接入功能。移植协议栈时，只要对驱动实现部分进行修改，而保持调用接口不变，上层协议可以不做改动。考虑到以太帧数据部分最大为 1500 个字节，如果用户程序的数据需要先拷贝到 TCP/IP 协议栈内部的缓冲区中，再通过该缓冲区传递给以太网模块，将会占用大量的内存空间

[41]。这样的系统开销对资源紧张的嵌入式系统来说代价太大，因此采用直接操作缓冲区的方式更符合实际需求。发送时用户数据不需要从应用拷贝到协议栈缓存，而是将数据首地址指针传递给协议栈，经过校验处理后协议栈调用以太网驱动接口把用户数据直接写入发送缓冲区；接收数据时各层进行相应协议处理并传递数据首地址指针给上层协议，通过避免协议栈内部数据复制和维护以提高系统的吞吐效率[42][43]。

整个协议栈以循环方式运行，MC9S12NE64 以中断方式接收以太帧，协议栈对接收到的以太帧自底向上进行过滤，并做相应处理，通信过程如图 4-8 所示。各层设计如下：

(1)链路层：MC9S12NE64 以太网驱动，ARP 协议；

(2)网络层：IP 协议，ICMP 回显应答服务；

(3)传输层：UDP 协议。

4.2.2 MC9S12NE64 以太网驱动

以太网驱动为整个协议栈提供了运行的基础，其最重要的功能是收发以太帧[44]，因为与硬件功能关系密切，所以不同的 MCU 实现起来可能有所差异。

1. 初始化

由于使用 NE64 内部集成 EPHY 和 EMAC 模块，因此初始化的过程应包含对两个模块相关寄存器的设置。其中 EPHY 模块 19 个有 MII 寄存器无法直接寻址访问，需要通过 MII 管理接口来操作。而且芯片初始化时要将总线频率设为 25Mhz，以满足 EPHY 模块的工作需要。

根据设计需求，并参照 MC9S12NE64 的芯片手册。初始化过程如下：

(1)关闭 EPHY 模块时钟。

(2)设置 EPHY 地址。这个地址在进行 MII 操作时会使用到，因为使用了内部 EPHY，所以地址为 0x00。

(3)使能 EPHY 指示灯。这样 EHPY 模块可以自动根据网络状况设置 ACTLED(以太网活动状态)、LNKLED(连接状况)、SPDLED(通信速度)、DUPLED(全/半双工)和 COLLED(冲突)等指示灯的状态。这项功能为调试 EPHY 模块提供方便。

(4)使能 EPHY 模块。使得 EPHY 地址有效，并允许操作 MII 接口。

(5)设置 EMAC 模块的 MDC 时钟为 2.5Mhz。

(6)设置 EMAC 模块的接收 A、B 缓冲区和发送缓冲区缓冲区大小都为最大值 1.5KB，在地址空间 0x0000～0x11FF 中依次存放。由于系统的 RAM 空间被划出一部分作为以太网的缓冲区，因此程序的运行 RAM 空间起始地址(默认 0x0400)要进行重新定位，以免与这部分地址重叠，影响系统正常运行。

(7)写 MAC 地址到 MACAD 寄存器。这个寄存器在复位后只能被修改一次，如果通过在线方式修改 NE64 设备的 MAC 地址，需要复位后才有效。

(8)设置接收帧的类型为接收所有帧。

(9)设置 MAC 地址过滤模式，接收唯一地址帧(发给本机)和广播帧(ARP 请求)。

(10)确定全半双工模式和通信速度 10/100Mbps。在通信速度要求不高的应用场合推荐使用 10Mbps，降低系统的功耗，并减少了高频信号干扰，有利于系统长时间稳定运行。

(11)允许 EMAC 接收中断。使得 NE64 通过中断方式来接收以太帧，满足了嵌入式系统对实时性的要求。

(12)允许 EPHY 中断。因为 EPHY 模块的中断发生条件是系统复位或者网络的通/断，所以本文设计在进入 EPHY 中断后，通过 PSR 寄存器判断以太网连接状态，并通过参数传递给 TCP/IP 协议栈参考。

(13)启动 EPHY 时钟。

至此，MC9S12NE64 的 EPHY 和 EMAC 模块已经初始化完毕。

2. 收发以太帧

(1)接收以太帧

MC9S12NE64 通过设置 A、B 两个接收缓冲区提高了以太网的数据吞吐量。当以太网线路上的数据帧通过 NE64 的 MAC 地址过滤，并且 A、B 缓冲区至少有一个为空闲，则该帧信息被接收，并产生相应的接收缓冲区中断。

```
//以太网帧首部信息结构体定义
struct ethernet_frame
{
INT8U   DSTMACAddr[6];   // 目标MAC地址
INT8U   SRCMACAddr[6];   // 源MAC地址
INT16U  Frame_len;       // 收到的以太网帧的大小
INT16U  Protocol_Type;   // 协议类型:IP-0x0800  ARP - 0x0806
INT16U  Data_index;      // 以太网帧中的数据地址
};
```

本文根据以太帧格式设计了以太帧结构体 ethernet_frame，并由此定义了结构体

变量 received_frame 作为当前需要处理的以太帧首部。在缓冲区 A、B 中断处理程序中，调用以太帧接收处理函数提取相应的以太帧首部信息传递给 received_frame，上层协议通过 received_frame 提取相关信息。因为接收处理只提取以太帧首部信息，并且只将以太帧数据首地址传递给上层协议，而数据内容一直放在缓冲区中，所以占用 MCU 的运算时间和系统的空间较少，运行效率高。当上层协议处理完以太帧后要清空当前处理的接收缓冲区，以接收下一帧数据。以太帧接收处理函数首部声明如下：

```
//-------------------------------------------------------------------------*
//程序名:FrameReceive                                                      *
//功  能:提取以太网帧首部信息赋值给全局变量received_frame                  *
//入  口:*Pbuffer-接收缓冲区首地址指针  Datalen-缓冲区数据长度             *
//       bufflag-接收缓冲区状态标志位                                      *
//返  回:操作结束返回0                                                     *
//-------------------------------------------------------------------------*
INT16U FrameReceive (INT16U *Pbuffer, INT16U Datalen, INT16U bufflag);
```

(2)发送以太帧

发送以太帧时，上层协议先通过以太网驱动的接口函数将数据及以太帧首部直接写入发送缓冲区，然后调用 EMAC 模块的发送命令，接下来的发送过程就由 NE64 自动完成。发送以太帧需要调用的接口函数声明如下：

```
void SNDBufWriteB (INT8U data);                    //向发送缓冲区写入一个字节数据
void SNDBufWriteW (INT16U data);                   //向发送缓冲区写入两个字节数据
void SNDBufWriteN (INT8U* pdata, INT16U datalen);//向发送缓冲区写入多个字节数据
void AddFrameH (struct ethernet_frame* sndframe);//向发送缓冲区写入以太帧首部
void FrameSend (INT16U datalen);                   //指定数据长度,并发送数据
```

4.2.3 嵌入式 TCP/IP 协议栈

嵌入式 TCP/IP 协议栈的实现与硬件无关，通过调用以太网驱动接口来完成相应功能，因此具有很好的可移植性。

1. ARP协议实现

以太网中，ARP 协议负责将 32 位的 IP 地址解析成 48 位的 MAC 地址。绝大部分的 TCP/IP 协议栈中都定义了 ARP 缓冲表，至少要含有 IP 地址及相对应的 MAC 地址两项。发送 IP 数据报时，首先根据包中的 IP 地址，在缓冲表中查出对应的 MAC 地址并填入包中相应位置；若表中没有相应 MAC 地址，则按照标准格式组装一个 ARP 请求包并发送，以得到对方 MAC 地址。

因此，对 ARP 缓冲表的维护是实现 ARP 协议的重点同时也是难点。不仅在接收到 ARP 应答包时，需要更新 ARP 缓存表，而且在主机运行一段时间后，因通信对方

的 IP 地址可能发生改变，也要刷新 ARP 缓存中的表项，这个时间就是表项的生存周期。上述的维护操作对有资源限制的嵌入式系统无疑代价很大，除了时空开销，还需要定时器模块的支持[45]。通过分析 ARP 协议原理，并结合嵌入式系统的应用特点，本文提出了一种无需 ARP 缓冲表的协议处理的方法，对嵌入式设备的主动与被动通信采用不同的处理方式，如图 4-9 所示。

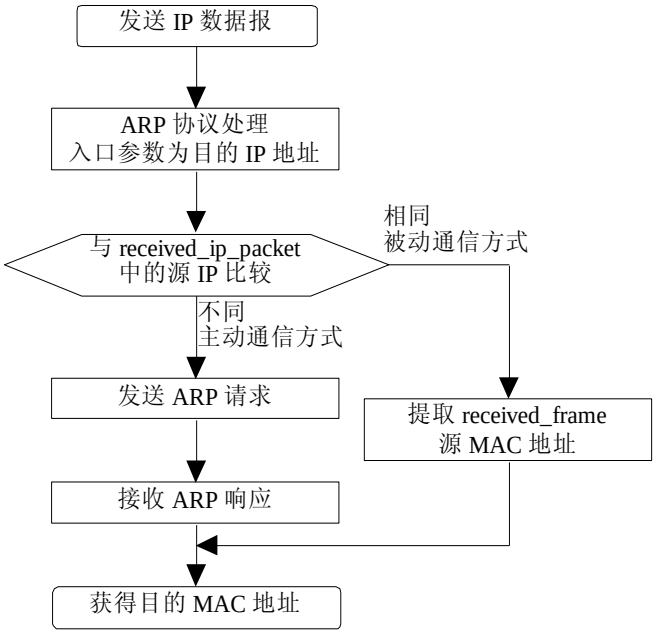


图 4-9 发送 IP 数据报前的 ARP 协议处理流程图

以太网驱动部分将接收到的以太帧首部信息放在全局变量 received_frame 结构体变量中。相应的，网络层也从该以太帧的数据中提取 IP 首部信息存放在全局变量 received_ip_packet 内，其中包含了目的与源 IP 地址信息。当嵌入式设备需要发送 IP 数据包时，目的 IP 地址作为参数传递给 ARP 协议处理程序；然后将输入参数(目的 IP 地址)与 received_ip_packet 中存储的源 IP 地址比较，判断本次通信是主动方式还是被动方式；如果是被动方式，则直接将 received_frame 中的源 MAC 地址返回给 IP 层，主动方式则需要通过发送 ARP 请求报文来获得目的 MAC 地址。

采用上述方法，嵌入式设备在被动通信时具有较高的效率；主动通信时每发送一个 IP 数据报之前都要先发送 ARP 请求，效率相对较低。嵌入式设备在实际使用中作为测控系统的终端，采用被动通信方式较多，而且考虑到增加 ARP 缓冲表维护操作为系统带来的负担，因此本文设计的 ARP 协议处理方法不失为一种合适的解决方案。

ARP 协议部分提供的接口函数声明如下：

```
INT8U ARPexecute(void * ipaddr, void * hwaddr);  //ARP协议处理,实现IP地址与MAC地址的转换
INT8U ARPanalyse(struct ethernet_frame* reframe);//分析接收到的ARP分组类型,并做不同处理
void  ARPSendReq(INT8U * DestIp);                //向目标IP地址发送ARP请求
void  ARPGetResp(void);                          //处理接收到的ARP应答
void  ARPSendResp(void);                         //发送ARP应答
```

2. IP协议实现

在 IP 协议设计中，根据嵌入式设备一般只作为通信终端的特点，裁减了 IP 选路功能，只实现了基本的收发 IP 数据报操作：

(1)IP 协议接收处理

以太帧成功接收后，链路层将以太帧首部信息 received_frame 作为入口参数传递给 IP 接收程序。received_frame 中的 Data_index 成员变量指向 IP 数据报的首地址，IP 接收处理程序根据这个地址提取 IP 首部信息，在校验通过且相关设置符合时，比如 IP 不分片或数据报是发往本地的等等，则将其数据部分提供给更高层的 UDP 或 ICMP 协议处理，否则丢弃。

(2)IP 协议发送处理

IP 协议在网络层，提供调用接口给其他协议，因此 IP 协议发送程序首先要判断上层的协议类型，并做相应初始化；然后调用 ARP 协议处理程序，获得目的 MAC 地址；最后将发送数据数据和以太帧首部直接写入发送缓冲区，并通过驱动层的接口来启动以太帧的发送。为了避免 IP 分片与重组，减少 IP 协议实现部分的复杂度，发送 IP 数据报的长度被限定在 1480 个字节之内，加上 20 个字节的 IP 首部，总长度不超过设置的以太帧最大数据长度 1500 字节。

IP 协议的接收和发送函数声明如下：

```
//-------------------------------------------------------------------------*
//程序名:IPReceive                                                         *
//功  能:处理接收的IP数据报,提取首部信息赋值给全局变量received_ip_packet  *
//入  口:reframe-接收的以太帧                                              *
//返  回:IP数据报不正确返回-1;否则返回IP数据包长度                         *
//-------------------------------------------------------------------------*
INT16S IPReceive (struct ethernet_frame* reframe);
//-------------------------------------------------------------------------*
//程序名:IPSend                                                            *
//功  能:发送IP数据报                                                      *
//入  口:ipadr-地址; pcol-IP协议类型号; tos-服务类型号; ttl-生存周期       *
//       *databuf-发送数据首地址数据; IPDatalen-IP的数据长度               *
//返  回:IP数据报不正确返回-1;否则返回IP数据包长度                         *
//-------------------------------------------------------------------------*
INT16S IPSend (INT8U * ipadr, INT8U pcol, INT8U tos, INT8U ttl,INT8U* databuf, INT16U IPDatalen);
```

3. ICMP协议实现

嵌入式协议栈一般只实现 ICMP 回显应答服务。因为嵌入式设备一般未实现访问控制和防火墙，所以PC机可以通过PING程序来测试该设备在网络层是否可达。ICMP协议处理比较简单，首先校验接收报文，然后将该报文中的类型字段由 8 改为 0，即由回显请求改为回显应答，最后重新计算发送信包校验和并调用 IP 协议发送程序回送应答报文给请求主机。

4. UDP协议实现

UDP 协议通过 IP 地址和端口号来确定通信的主机和进程，并把应用程序传送给协议栈的数据发送出去[46]。虽然 UDP 不保证数据能到达目的地，但是与 TCP 协议相比，省去了通信前的“三次握手”以及复杂的状态机机制，因此 UDP 协议更为简洁高效，更适合数据传输实时性要求较高的嵌入式系统应用。UDP 的发送过程是为应用程序数据添加 UDP 首部，并传递给下一层的 IP 协议，而接收过程则相反。为了增加传输数据的准确性，本文实现了 UDP 校验功能，重传机制则留给用户程序在应用层中实现。

UDP 校验算法为二进制反码求和[47]，内容包含 UDP 数据，UDP 首部和一个 12 字节的 UDP 伪首部。伪首部由以下部分组成：源 IP 地址(4 字节)、目的 IP 地址(4 字节)、数据 0(1 字节)、协议(1 字节)和 UDP 长度(2 字节)[16]。

校验和字段长度为 1 个字(2 字节)，而参与校验的数据并不是按字存储的，并且还包含单字节的数据。因此，需要按以下步骤对 UDP 校验和进行计算：

①将 16 位的累加和变量 Sum 清零，并分为高、低两个字节 SumH 与 SumL；

②设置输入数据地址计数变量 Addr，初始化为 0，每个字节计算完毕将 Addr 加一，使得高字节数据的地址计数值 Addr 为偶数，而低字节的为奇数；

③通过 Addr 值的奇偶来判断待校验数据是高字节还是低字节，并与 SumH 或 SumL 进行循环进位的加法运算；

④组合 SumH 与 SumL 得到累加和 Sum；

⑤对累加和 Sum 按位取反，得到最终的校验和结果。

循环进位加法运算的实现代码如下：

```
if( Addr & 0x01 )
{//奇数字节为低字节
    SumL = SumL + Data_in;
```

```
    if(SumL < Data_in )
    {//低字节产生进位
       if(++SumH == 0 )    //高字节加一,并判断是否有进位
           SumL++;         //高字节也产生进位,低字节加1
    }
}
else
{//偶数字节为高字节
    SumH = SumH + Data_in;
    if( SumH  < Data_in )
    {//高字节产生进位
        if(++SumL == 0 )    //低字节加一,并判断是否有进位
            SumH++;         //低字节也产生进位,高字节加1
    }
}
```

UDP 协议的接收和发送函数声明如下：

```
//-------------------------------------------------------------------------*
//程序名:UDPReceive                                                        *
//功  能:处理接收到的UDP数据报                                             *
//入  口:rePacket-接收到的IP数据报; datalen-数据长度                       *
//返  回:-1,发生错误;1,处理成功                                            *
//-------------------------------------------------------------------------*
INT16S UDPReceive (struct ip_packet* rePacket, INT16U datalen);
//-------------------------------------------------------------------------*
//程序名:UDPsend                                                           *
//功  能:发送UDP数据报给目的主机                                           *
//入  口:dstip-目的IP地址; dstport-目的端口号; *databuf-要发送数据首地址   *
//       datalen-要发送的数据长度(应小于1500-20-8=1472字节)                *
//返  回:-1,发生错误;1,处理成功                                            *
//-------------------------------------------------------------------------*
INT16S UDPsend (INT8U * dstip, INT16U dstport, INT8U* databuf, INT16U datalen);
```

4.2.4 网络参数的在线修改

嵌入式 TCP/IP 协议栈的运行需要设定一些参数，如本机 MAC 地址、本机 IP 地址和网关 IP 地址等。如果将这些参数设为常量，那么只能与整个程序代码一起，一次性写入到 MCU 中，当需要更改参数时，如增加设备或更改设备所在网段，都需要修改源文件，并将整个工程文件编译后重新下载到 MCU 中；设为变量时，修改后一旦 MCU 复位，被修改的参数又恢复成原值。

Freescale 的 S12 系列 MCU 提供了对 Flash 存储器在用户模式下的在线编程功能[30]。本文设计采用的 MC9S12NE64 就是 S12 系列中的一款，64KB 的 FLASH 空间分为 0x3C、0x3D、0x3E 和 0x3F 四页，每页大小为 16KB。其中 0x3E 和 0x3F 固定映射在 0x4000～0x7FFF、0xC000～0xFFFF 可以直接访问，其他两页需要通过选页机制来映射到 0x8000～0xBFFF 这段地址中。

因此，将网络参数写入到合适的 FLASH 地址中，MCU 运行后通过读取该地址的内容来获得参数内容，需要修改时通过 FLASH 在线擦写技术修改 FLASH 中的数据，保证了掉电后数据不丢失，提高了系统的稳定性。

向 MC9S12NE64 的 FLASH 空间写入数据每次至少 2 个字节,而擦除则至少为一个扇区(512 字节)，并且向已存储数据的地址写入必须先执行擦除操作。因此要将网络参数独立存放在一个扇区中，每次只对这个扇区进行修改，避免擦除操作影响到存储程序的区域。因为 NE64 的 EMAC 模块中的 MAC 地址寄存器只能在上电后写入一次，因此网络参数的初始化操作应在 EMAC 模块的初始化之前。当 NE64 接收到在线修改 MAC 地址的命令，在完成操作后需要调用一个无限循环程序，利用看门狗模块来自动复位 NE64，以使修改后的参数有效。

擦除程序操作步骤如下：

①判断是否设置时钟分频，若无分频，则先分频 FCLKDIV |= 0x50；

②清保护错误和访问错误标志位 FSTAT |= 0x30；

③选择块号，设置页寄存器；

④向要擦除的扇区首地址写任意值(0x0000 除外)；

⑤向命令寄存器写扇区擦除命令 FCMD = 0x40；

⑥在加高压期间,调用 RAM 区的机器码。

与上述过程相比，FLASH 写入程序操作在第④步时将数据写入指定地址，第⑤步调用写入命令 FCMD = 0x20，其他部分与擦除操作相同。

4.3 软件测试与设计体会

4.3.1 软件测试

软件测试的过程要先模块后整体，先测试能产生可视现象的模块，如运行指示灯的驱动，通信接口等，然后再完成其他功能模块的测试。

1. 以太网驱动测试

测试时将 NE64 设备通过交叉网线与 PC 机相连，此时 PC 机上相应的网卡应显示“网络电缆没有插好”；而当 NE64 设备上电并正确初始化后，PC 机网卡状态改为已连接上。通过观察 NE64 的 LNKLED 指示灯的状态变化，也可达到同样的测试目

的。

2. ARP协议与ICMP协议测试

将 NE64 设备的 IP 地址设为“192.168.149.132”，MAC 地址设为“0xF0 0x4E 0x77 0x8A 0x35 0x1D”，与测试用的 PC(192.168.149.153)连接于同一个网段。首先使用“arp -d”删除 PC 上的 ARP 缓冲表；然后 PING 目标设备，此时 PC 会先发送 ARP 请求，收到应答后再发送 ICMP 回显请求报文；最后通过“arp -a”显示 ARP 缓冲表。测试结果如图 4-10 所示，表明 NE64 设备可以完成 ARP 与 ICMP 应答服务。

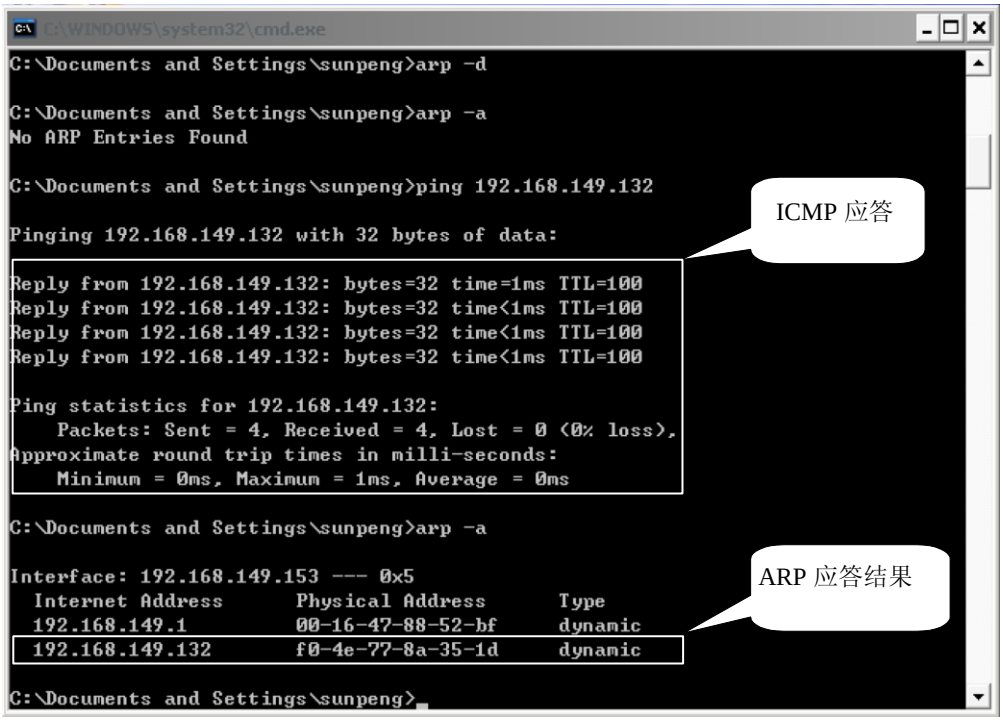


图 4-10 ARP 与 ICMP 协议测试结果

3. TYPE A & B电子标签的UID识别测试

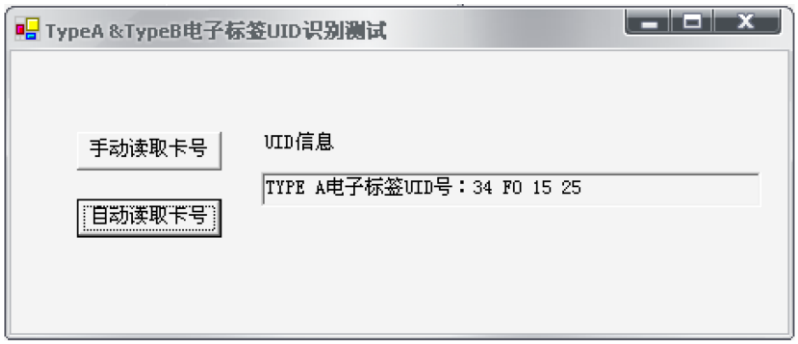


图 4-11 TYPE A 电子标签的 UID 识别测试结果

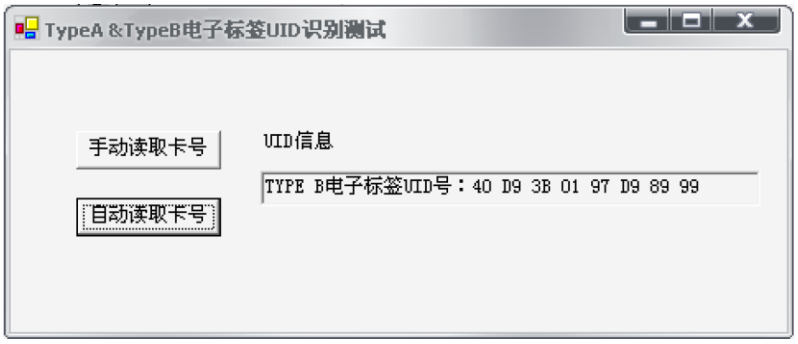


图 4-12 TYPE B 电子标签的 UID 识别测试结果

图 4-11、4-12 给出了识别 TYPE A & B 两种电子标签 UID 的测试结果。PC 机主动发送包含通信命令的 UDP 数据报给读写器，读写器接收到该数据报后，将获取的最新 UID 信息通过 UDP 协议返回给 PC 主机。

测试的各项具体参数见表 4-4。识别间隔时间是指电子标签处于读写器的有效识别距离内，读写器两次读取其 UID 的时间差值。因为读写器每次成功读取 UID 就会闪烁其硬件上的指示灯一次，测试时将电子标签置于 RF 场中一段时间，通过目测指示灯闪烁次数获得，存在一定误差，不包含网络延时。

表 4-4 UID 识别测试参数

电子标签	类型	能否识别	UID 长度(字节)	最大识别距离	识别间隔时间
Mifare 1 S50	TYPE A	可识别	4	7cm	小于 2 秒
二代身份证	TYPE B	可识别	8	5cm	小于 1 秒

4.3.2 软件设计体会

与硬件相比，软件设计具有较高的灵活性，并很大程度上决定了系统的整体性能。以下是作者对嵌入式软件开发的几点体会：

(1)精简代码。

由于嵌入式系统的运算能力和存储空间有限，使得嵌入式软件对时间和空间的开销比较敏感。因此，设计时应尽量精简代码，并减少对系统资源的占用。

(2)考虑实时性与可靠性

嵌入式系统往往对实时性有一定要求，而实时性的实现与软件的设计有很大关系，这涉及到合理的程序结构、高效的代码和中断处理等很多方面。同时，为了系统的稳定运行，需要对很多异常情况做出处理，比如在子程序入口做错误参数的判断处

理以及为系统增加看门狗程序等。

(3)规范程序设计

良好的编程风格可以提高程序的易读性和易理解性，方便了后期开发人员的维护。对与硬件相关的驱动程序应该封装严密，任何应用程序只能通过驱动接口操作硬件。子程序的设计应标明入口和出口参数，函数内部尽量不操作全局变量。

(4)提高可移植性

特别是一些功能性的程序，设计时应考虑到通用性，方便代码的移植，减少重复的开发工作。例如，本文在设计 ARP 协议处理函数时，充分考虑到无操作系统的嵌入式设备的普遍特点，取消了 ARP 缓冲区机制，提高了协议栈的可移植性。

(5)充分测试

代码编写完毕并不代表开发过程的结束，对各个模块要进行严格的测试是保证软件质量的基础，而且后期的测试现象对前期的测试结果往往有一定影响。作者在开发 I/O 口模拟 SPI 的子程序时，测试单个字节数据交换，数据准确性和通信速度都符合要求。但是在调试后期模块的过程中，如连续进行 8 个字节的 UID 信息的通信时，通信速度慢的现象很明显，最终发现是 SPI 主机等待数据的延时太长所致。

4.4 本章小结

本章主要工作总结如下：

(1)通过对 SPI 通信原理以及时序图的分析，在通用 I/O 口上以软件模拟的方式实现了 SPI 通信。

(2)按照自底向上、先驱动后功能的方法，结合流程图与关键代码，在完成 RC531 驱动层的基础上，通过对 ISO14443 A & B 两种电子标签不同状态机制的分析，阐述了 ISO14443 A & B 电子标签 UID 识别的软件实现。

(3)深入分析现有嵌入式 TCP/IP 协议栈结构及运行原理，结合无操作系统的嵌入式系统特点，设计了一个精简的 TCP/IP 协议栈，实现了主控系统的以太网接入功能。高层协议通过以太网驱动接口直接读写链路层收、发缓冲区，通过避免数据复制提高了系统性能；取消 ARP 缓冲区，对主动和被动通信采用不同的 ARP 协议处理方式。

(4)软件测试结果表明读写器运行结果与程序设计一致。

(5)根据软件开发过程中遇到的实际问题给出了设计体会。

第五章 应用实例设计与分析

完整的 RFID 系统是由电子标签、读写器和计算机管理系统组成的。本文设计实现的读写器具有以太网通信功能，可识别我国第二代居民身份证 UID 号，为 RFID 管理系统的开发提供了方便。

本章以智能大厦门禁管理系统为主，通过对应用实例的分析，讲述了在局域网内以二代身份证作为只读电子标签的 RFID 管理系统应用方法。

5.1 智能大厦门禁系统设计

5.1.1 门禁系统需求分析

智能大厦(Intelligent Building)是指实现了实现了楼宇自动化(Building Automation，缩写 BA)、办公自动化(Office Automation，OA)、通信自动化(Communication Automation，CA)及布线综合化的智能化大型建筑[48][49]。智能大厦是信息时代的产物，通过中央控制系统监控大楼内的各种控制设备、通讯设备、管理系统、消防系统、给排水系统等装置。

门禁系统顾名思义就是是管理人员进出门户的系统[50]，而采用射频技术来判别用户的身份是目前门禁系统市场的主流。

智能大厦的门禁系统应具有以下功能：

(1)电子钥匙

用户通过 IC 卡可自由进出满足权限的门，无需携带多把钥匙去开门。

(2)权限设置

按持卡人身份设置其有效的开门区域，有效杜绝人员的随意进出。当出现不符合权限的刷卡操作时，产生报警信号。

(3)时间设置

对特殊地点，设置访问的有效时间段，如公司的办公场所在非上班时间禁止开门。

(4)访问记录

记录每次的刷卡信息，生成报表方便管理部门查询，并为考勤提供参考。

(5)挂失管理

如果所持卡片遗失，需要立即办理挂失以使该卡在系统内失效，防止不必要的财产损失。

(6)安全疏散

在出现紧急情况时，通过控制系统打开所有通道门锁，确保人身安全。

上述功能要求门禁控制系统有很快的通信速度，以及合理的卡片管理体系。因此，采用以太网通信方式不仅可以满足智能大厦门禁系统对速度的要求，还可以利用其完善的网络基础设施，省去了现场布线的工作。同时，二代身份证已经全面发放，采用二代证作为门禁系统的识别卡，即节省了制卡成本又提高了卡片的防伪性。可见，智能大厦是应用本文所设计的 RFID 系统的理想场所。

5.1.2 门禁系统设计方案

1. 系统组成

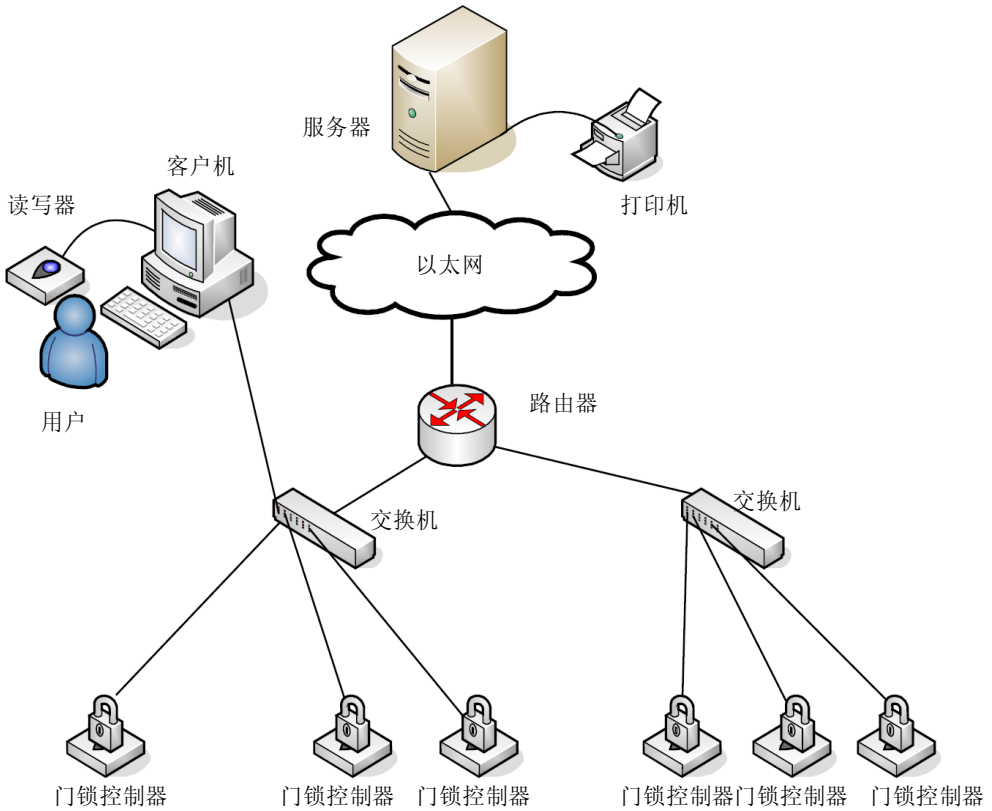


图 5-1 智能大厦门禁系统示意图

智能大厦的门禁系统的系统组成如图 5-1 所示。系统管理软件运行在服务器中，通过以太网与具有读卡功能的门锁控制器通信；门锁控制器用于开、关指定门；用户

可以通过客户机登记信息或管理系统。

2. 硬件设计

因为通信部分利用了现有的网络设施，所以系统硬件部分的重点是门锁控制器的设计。门锁控制器至少要完成读卡、网络通信和开关门的功能。本文设计的读写器除了符合前两项功能要求，并且在设计时预留了输入输出接口，因此可以通过增加一组输出，完成对门的开关控制，并接入门磁传感器来检测门的开关状态等。

门锁按通断电时的状态可分为以下两种[50]：

(1)断电闭门

顾名思义，在断电时为锁门状态，一旦锁体通电则完成开门操作。

(2)断电开门

断电时呈开门状态，当外部控制系统对锁体通电后，转为锁门状态。

设计门锁控制器时，需要考虑到适用场合。在财产保护性高的地点，应使用断电闭门锁。而对大厦的安全通道，应使用断电开门锁，保证紧急情况下人员的疏散。

3. 管理系统的软件架构

C/S(Client/Server，客户端/服务器)和 B/S(Brower/Server，浏览器/服务器)是两种不同的软件架构。

C/S 模式分客户机和服务器两层，其中客户机需要安装专用的客户端软件，并具有一定的数据处理能力。但是由于服务器连接个数和数据通信量的限制，这种结构的软件适于在用户数量不多的局域网内使用。并且 C/S 模式的维护和升级成本非常高，对客户机的软件环境一般也会有所限制。

B/S 模式是随着 Internet 技术的兴起，对 C/S 模式的一种改进的结构，其最大的优点就是客户机是通过浏览器来实现用户界面并完成与服务器的数据交互，而不用安装任何专门的软件。B/S 模式大大简化了客户端，并且减少了系统维护与升级的成本和工作量。

因此，智能大厦的门禁管理系统(RFID-ACS 管理系统)采用 B/S 模式，使用 ASP.NET 技术将其开发成 WEB 应用程序。系统管理人员在客户机上通过浏览器打开登录界面，再使用分配的用户名和密码登录到服务器上，就可以进行相应权限的管理操作，登录前后的浏览器界面如图 5-2 所示。

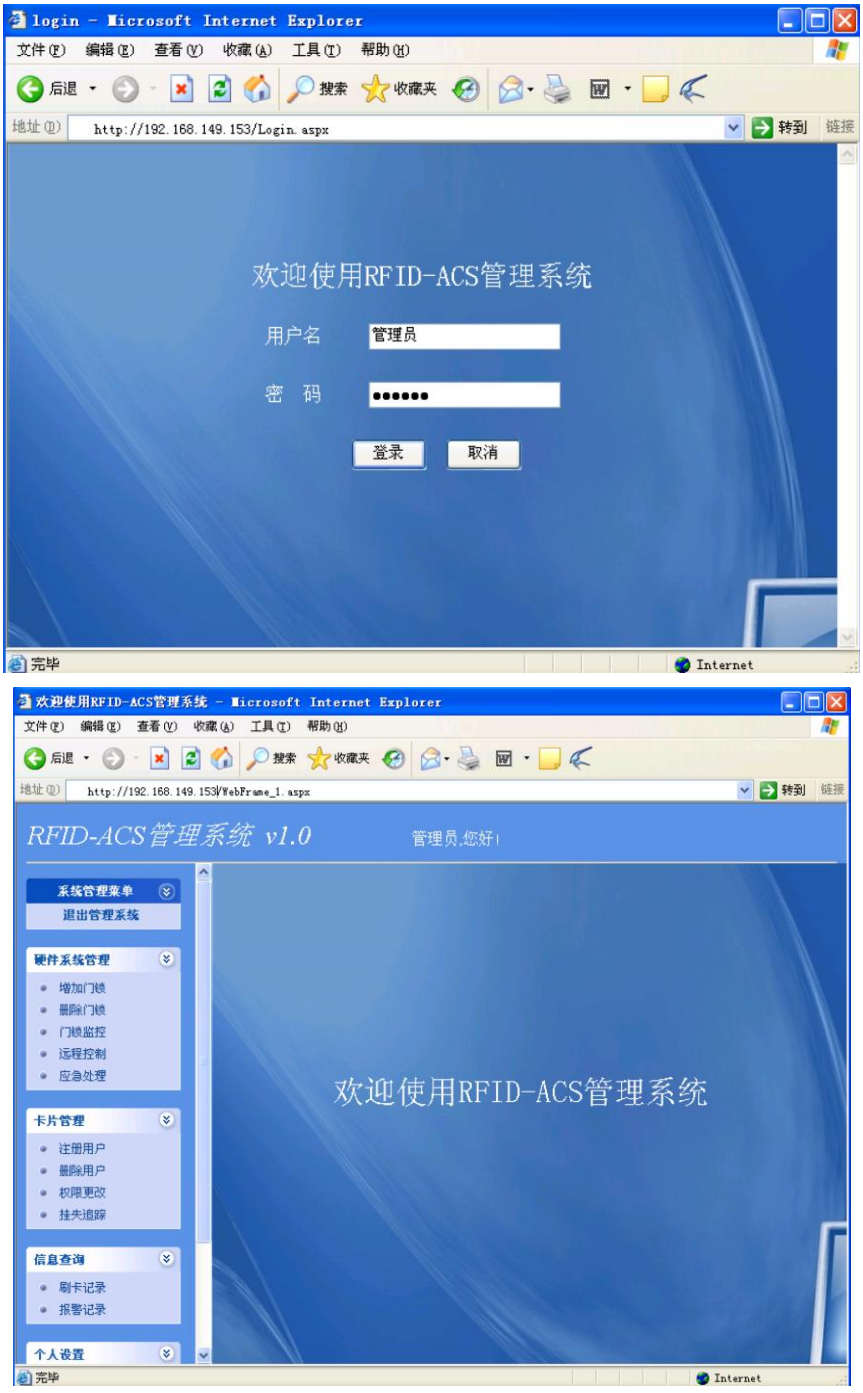


图 5-2 RFID-ACS 管理系统登录界面

4. 管理系统功能设计

经过需求分析，门禁管理系统的功能设计见表 5-1 所示。

表 5-1 门禁管理系统功能结构

功能		说明
硬件系统管理	初始化	设置所有门锁初始状态
	巡检	定时循环检测门锁控制器的网络连接状态
	增加门锁	增加一个门锁控制点，设置门锁控制器位置信息
	删除门锁	从系统中删除一个门锁控制点
	门锁监控	监控所有门锁的当前开、关状态
	远程控制	远程操作门锁的开关，应对用户身份证丢失等特殊情况
	应急处理	紧急状况时打开所有通道门锁
门禁实时控制		及时响应门锁刷卡数据包，判断用户身份权限后返回控制命令
卡片管理	注册用户	登记用户二代证 UID 号及相关个人信息，分配用户门禁权限
	删除用户	删除指定用户记录，取消其所有门禁权限
	权限更改	根据人员流动变化，修改用户的门禁权限
	挂失追踪	当用户的身份证挂失后，一旦发现还有人使用该卡，系统报警
信息查询	按用户查询	查询指定用户的刷卡信息
	按门锁查询	查询指定门锁控制器的刷卡信息
	按时间查询	查询指定时间段的刷卡信息
	报警信息	查询报警信息
	报表输出	为方便管理，提供刷卡信息的报表输出

整个管理系统的设计需要考虑到多种可能的情况。正常操作时，用户首先通过与客户机相连的读写器识别二代证的卡号，与个人相关信息一起输入管理系统，并上传到服务器中获得相应等级的进出权限。终端控制的门锁控制器具有读卡功能。当用户需要进出某个门时，只要拿出二代证在距离门锁控制器读卡接口 5cm 的地方轻轻掠过，刷卡信息立即发送给服务器以判断用户身份，门锁控制器接收服务器的回送命令，执行开关门的操作及控制警报器状态。如果用户忘带或丢失身份证，可以联系系统管理员进行远程开门。处于紧急情况时，如产生火警，系统将自动打开所有通道疏散人员。

并且为了方便管理，提供了按用户、门锁和时间段进行信息查询的功能，并能生成打印相应的统计报表。

5. 数据库设计

RFID-ACS 管理系统中，后台数据库的设计十分重要，合理的信息和范式分解是关键。根据系统需求，选用 SQL Server2000 数据库产品，并使用 ADO.NET 技术实现对数据库的访问。

根据对 RFID-ACS 管理系统的功能和需求进行分析，系统中各主要实体的 E-R

关系图如下：

(1)进出人员

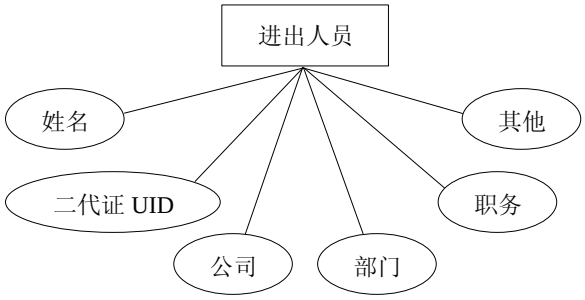


图 5-3 进出人员信息实体 E-R 图

进出人员信息实体的相关属性见图 5-3。因为每个公民只有一个有效身份证，因此将二代证的 UID 号作为主键唯一确定用户。其他属性作为系统管理的参考资料，如通过公司和部门属性进行进出人员的信息查询操作。

(2)门锁控制器

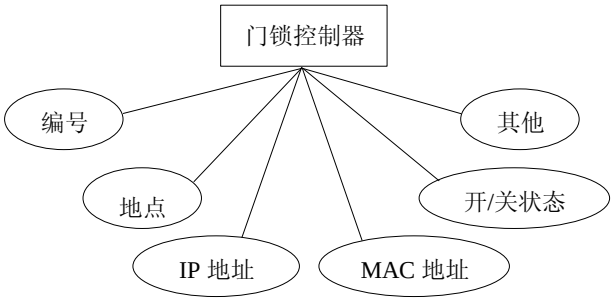


图 5-4 门锁控制器信息实体 E-R 图

门锁控制器作为 RFID-ACS 管理系统的终端控制设备，其相关属性如图 5-4 所示。每台门锁控制器用编号唯一确定，并设置地点、网络参数和开/关状态信息方便查询。

(3)权限设置

如图 5-5 所示，权限设置确定了进出人员信息主体和门锁控制器信息主体之间的关系模式为 m:n 的对应关系，即一个人员可以开多扇门，一扇门也可以由多个不同的人打开。门禁时间段限定了特殊地点的有效访问时间，如某些公司只允许在上班时间开门。

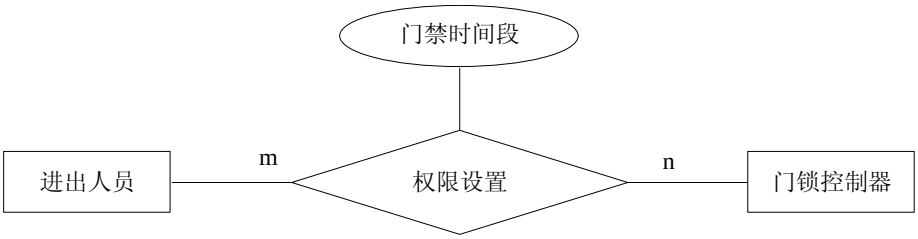


图 5-5 权限设置

6. 与门锁控制器的通信

由于门锁控制器实现了以太网通信功能，并且在运输层采用了 UDP 协议，因此 RFID-ACS 管理系统在服务器端通过 Microsoft WinSock Control 控件完成与门锁控制器的信息交互。因为 UDP 协议为非连接式的通信协议，所以使用 WinSock 控件在初始化时只需要指定本机 IP 地址和端口号，然后设置成 UDP Server 绑定端口监听门锁控制器发送的 UDP 数据报，处理完毕后向数据库中写刷卡记录，并返回包含命令信息的 UDP 数据报给指定的门锁控制器。刷卡测试结果如图 5-6 所示。

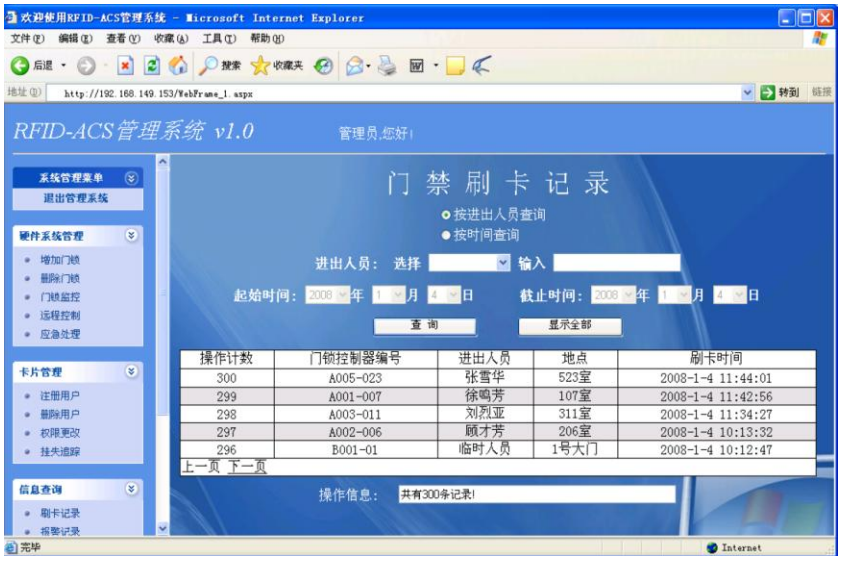


图 5-6 门禁刷卡记录

5.2 应用扩展

5.2.1 电子消费管理系统

“刷卡消费”目前已逐步成为一种时尚，不仅使用方便无需携带大量现金，而且卡片丢失后可以通过挂失的手段来提高安全程度。管理人员还可以通过刷卡信息统计

和管理消费情况，实现消费管理的信息化和自动化。二代身份证作为一种非接触式的电子标签，可以设置成“电子货币”应用在消费管理系统中。

由于本文设计的读写器实现了读取二代身份证 UID 的功能，但不能对电子标签进行信息写入，因此电子消费系统需要设置成在线收费模式。用户首先在服务台刷二代身份证建立账户并预存一定金额，消费时只需在具有读卡功能的 POS(Point Of Sale)机上再刷一次二代身份证，POS 机将该身份证的 UID 信息发送到服务器管理系统并完成扣款操作，服务器将账户余额返回给 POS 机并显示。

5.2.2 考勤管理系统

考勤是企业行政管理的重要考核手段之一。由于每个公民只有一张有效的二代身份证，并且身份证内的 UID 也是全球唯一的，因此使用二代身份证作为考勤卡可以唯一标识员工身份并且省去了发卡的费用。随着企业现代化的推进，其网络设施也越来越完善，利用现有的网络基础设施可以省去布线的费用。

使用本文设计的读写器作为考勤机，分布在各个考勤地点，并接入到公司局域网中。员工上下班时，只需要将二代身份证在考勤机的有效读取范围内晃一下，考勤机立即将该员工的身份证 UID 信息发送到服务器，服务器上运行的考勤管理软件核对信息后返回校验成功信息给考勤机，并将考勤记录写入到后台数据库中。管理人员可以通过考勤管理软件方便的进行查询、统计工作，并生成报表输出，实现考勤的自动化管理。

5.3 实例分析

智能大厦门禁系统、电子消费管理系统和考勤管理系统其实都是只读卡管理系统，终端设备只完成刷卡功能，所有控制需要通过后台管理系统来完成。这种控制方式为信息的集中管理提供了方便，但是对终端与服务器之间的通信速度有较高的要求，因此设计中采用了以太网通信方式。

综上所述，本文设计的 RFID 系统适合应用在二代身份证普及，人员数量多、流动频繁，并且对速度有要求、跨地域的工程中，如停车场收费、公司考勤和酒店住宿管理等。

5.4 本章小结

本章主要工作总结如下：

(1)通过系统需求分析，详细阐述了使用二代身份证作为只读电子标签，并采用以太网通信方式的智能大厦门禁系统的方案设计。

(2)简要描述了本文设计在电子消费管理系统和考勤管理系统中的应用。

(3)分析并总结应用案例，给出了本文设计的应用方法、特点以及适用场合。

第六章 总结与展望

6.1 全文总结

在沃尔玛等商业巨头的大力推动下，RFID 技术在很多行业开始得到应用，并且这个趋势愈演愈烈。然而在我国这样一个人口众多，拥有大量廉价劳动力的国家，电子标签的成本成为 RFID 普及的一个制约因素。本文设计的 RFID 系统以第二代身份证代替传统只读电子标签，为解决这一问题提供了参考。

本文主要工作总结如下：

(1)分析国内外 RFID 技术应用状况以及嵌入式以太网技术的发展趋势，结合我国二代身份证的全面发放，提出了本文的设计方案。

(2)从实际开发角度详细分析了 RFID 与嵌入式以太网技术的相关原理、国际标准以及通信协议。

(3)读写器硬件系统被划分为读写模块硬件中间件和具有以太网通信功能的主控系统两部分，通过降低硬件模块之间的耦合度提高了硬件的复用性。读写模块硬件中间件被设计成标准的 DIP40 封装并给出了引脚定义，实现了硬件模块的即插即用。主控系统则采用了单芯片的以太网接入方案，简化了电路设计并提高了系统的性价比。

(4)使用普通 I/O 口，以软件模拟的方式实现了 SPI 通信，从而增加了系统设计的灵活度。

(5)读卡模块硬件中间件实现了读取包含二代身份证在内的 ISO/IEC14443 A & B 两种电子标签 UID 的功能，并提供标准的软、硬件接口。应用程序开发人员可以通过该接口直接获得电子标签 UID 信息，而无需了解射频实现细节，缩短了 RFID 应用程序的开发周期。当更换 RFID 应用系统的电子标签时，只需将读卡模块硬件中间件升级更新，且保持外部通信接口不变，从而减少了 RFID 系统的维护和管理的工作量。

(6)实现精简的 TCP/IP 协议栈，适合无操作系统支持的嵌入式系统使用。整个协议栈以无限循环的方式运行，各层之间通过指针传递数据信息，高层协议调用以太网驱动接口直接读写链路层缓冲区，通过避免数据复制提高了系统的性能。取消 ARP 缓冲区，省去了对其复杂的维护机制，对嵌入式设备的主动和被动通信采用不同的

ARP 协议处理方式，并且在被动通信时具有较高的效率。

(7)对读写器的软、硬件模块进行充分测试，并通过修正测试过程中发现的问题实现对设计的持续改进。

(8)结合应用实例分析，给出了以二代身份证为载体的局域网内 RFID 系统的应用方法，并且为局域网内的嵌入式系统应用和现有只读卡系统的升级提供了参考模型。

6.2 课题展望

由于时间和现有条件的限制，本文在设计实现过程中存在不少不尽完善的地方，需要在后续工作进一步改进：

(1)在获取 TYPE A & B 电子标签 UID 的实现过程中，增加防冲突处理，实现同时识别多张电子标签的功能，这点对于考勤系统尤为重要。

(2)增加读卡模块的对外命令接口，应用程序开发人员可以通过发送命令使读卡模块完成更多的高层操作。

(3)开发实际项目时，对通过以太网通信接口传输的数据进行加密，保证信息安全。

与发达国家相比，我国的 RFID 由于起步相对较晚，在技术和产业发展上存在一定的差距。但是在国家金卡工程的推动下，伴随二代身份证、2008 奥运会带来的巨大市场契机，通过众多研发人员的共同努力，RFID 必将进入人们生活的各个领域，并为我国的电子信息产业带来新的增长点。

参考文献

[1] 游战清, 李苏剑. 无线射频识别技术(RFID)理论与应用[M]. 电子工业出版社, 2004.

[2] Ngai,E.W.T., et al. RFID rearch:An academic literature review (1995-2005) and future research directions[J]. Production Economics, 2008,112(2):510-520.

[3] C.M.Roberts. Radio Frequency Identification(RFID)[J]. Computers & Security, 2006,25:18-26.

[4] 王爱英. 智能卡技术—IC 卡[M]. 清华大学出版社, 2000.

[5] Jeremy Landt. Shrouds of Time: The History of RFID[R]. AIM Inc., 2001.

[6] 中华人民共和国科学技术部等十五部委. 中国射频识别(RFID)技术政策白皮书[Z]. 2006.

[7] 马庆容. 我国 RFID 发展与应用现状研究报告[J]. 金卡工程, 2007,11(4):46-49.

[8] 朱爱红. 二代证读卡装置及其相关技术的研究[D]. 山东大学, 2007.

[9] 王宜怀, 刘晓升. 嵌入式系统—使用 HCS12 微控制器的设计与应用[M]. 北京航空航天大学出版社, 2008.

[10] 李挺, 郑伟. 应用无线射频识别技术(RFID)存在的问题[J]. 商场现代化, 2006(25):10.

[11] 鲁公羽, 陈雄, 倪斌. 射频识别系统中读写器的开发与研究[J]. 计算机工程与应用, 2006(07):89-91.

[12] Jan Axelson. Embedded Ethernet and Internet Complete[M]. Lakeview Research, 2003.

[13] Jan L. Harrington. Ethernet Networking for the Small Office and Professional Home Office[M]. Elsevier Science & Technology Books, 2007.

[14] Konstantinos Domdouzis, et al. Radio-Frequency Identification(RFID) applications: A brief introduction[J]. Advanced Engineering Informatics, 2007(21):350-355.

[15] International Standard ISO/IEC, FDIS 14443. Identification Cards- Contactless Integrated Circuit(s) Cards-Proximity Cards[S]. 2000.

-Part 1: Physical Characteristics

-Part 2: Radio Frequency Power and Signal Interface

-Part 3: Initialization and Anticollision

-Part 4: Transmission Protocol

[16] W. Richard Stevens. TCP/IP 详解 卷 1:协议[M]. 机械工业出版社, 2000.

[17] 杨晓静, 张玉, 徐济仁, 吕久明. 协议 TCP/IP 和 OSI/RM 的深入分析[J]. 计算机工程, 2002,28(11):263-267.

[18] Minh Huynh, Prasant Mohapatra. Metropolitan Ethernet Network: A move from LAN to MAN[J]. Computer Networks, 2007(51):4867-4894.

[19] Institute of Electrical and Electronics Engineers. IEEE 802-3:2000 standard, Part 3: Carrier Sense Multiple Access with Collision Detection(CSMA/CD) and Physical Layer Specifications[S]. 2000.

[20] D. C. Plummer, An Ethernet address resolution protocol-or- converting Network protocol address to 48 bit Ethernet address for transmission on Ethernet hardware[S]. RFC 826, 1982.

[21] J. Postel. DoD Standard Internet Protocol[S]. RFC 760, 1980-01.

[22] J. Postel. Internet Control Message Protocol[S]. RFC 792, 1981-09.

[23] J. Postel. Transmission Control Protocol[S]. RFC 793, 1981-09.

[24] J. Postel, ISI. User Datagram Protocol[S]. RFC 768, 1980-08.

[25] 张云勇. 中间件技术原理与应用[M]. 清华大学出版社, 2004.

[26] 陈萌萌, 邵贝贝. 单片机系统的低功耗设计策略[J]. 单片机与嵌入式系统应用, 2006(03):5-7.

[27] NXP Semiconductors. Mifare MF RC531 ISO 14443 Reader IC Data Sheet[EB/OL]. http://www.nxp.com/, 2002.

[28] NXP Semiconductors. Mifare MF RC531 ISO 14443 Reader IC Short Form Specification[EB/OL]. http://www.nxp.com/, 2005.

[29] Freescale Semiconductor, Inc. MC68HC908JB8 MC68HC08JB8 MC68HC08JT8 Technical Data[EB/OL]. http://www.freescale.com/, 2002.

[30] 王宜怀, 刘晓升. 嵌入式技术基础与实践[M]. 清华大学出版社, 2007.

[31] 朱灿. 基于非接触式 IC 卡的读卡器的设计与开发[D]. 武汉理工大学, 2006.

[32] 韩益锋. 射频识别阅读器的研究和设计[D]. 复旦大学, 2005.

[33] PLC. 13.56MHz RFID Systems and Antennas Design Guide[EB/OL]. http://www.melexis.com/, Melexis Semiconductors, 2004.

[34] 徐丽华. 射频识别卡读写模块的设计与应用[D]. 苏州大学, 2005.

[35] 李国, 秦培龙, 李艳红. 基于 CP2200 的嵌入式以太网接口设计[J]. 微计算机信息, 2007,23(7-2):32-34.

[36] 张东来, 常春. 基于单芯片以太网协议栈的远程环境监测系统[J]. 仪器仪表学报,

2003,24(4):537-539.

[37] Freescale Semiconductor, Inc. MC9S12NE64 Data Sheet Rev.1.0[EB/OL]. http://www.freescale.com/, 2004.

[38] 深圳市磁创通讯技术有限公司. PRJ-005A Data Sheet[EB/OL]. http://www.pptchina.cn/, 2005.

[39] 郑洪静. 基于嵌入式 Web 服务器的测控系统的开发[D]. 苏州大学, 2006.

[40] John Catsoulis. Designing Embedded Hardware[M]. Oreilly & Associates Inc., 2005.

[41] Mei-Ling Chiang, Yun-Chen Li. LyraNET: A zero-copy TCP/IP protocol stack for embedded systems[J]. Real-Time Systems, 2006,34(1):5-18.

[42] 潘建平,顾冠群,沈苏彬. 区域网络 TCP/IP 协议栈的性能分析[J]. 计算机学报, 1999,22(5):513-518.

[43] 王力生, 梅岩, 曹南洋. 轻量级嵌入式 TCP/IP 协议栈的设计[J]. 计算机工程, 2007,33(02):246-248.

[44] 全成斌, 任秀丽, 范力军,李贵兴. 嵌入式系统以太网驱动程序的设计方法[J]. 小型微型计算机系统, 2002,23(9):1029-1032.

[45] Weissenrieder, U., Schlegel, C. Use of the TCP/IP internet protocol in embedded computer systems[J]. Elektronik Praxis, 2001(12):122-6.

[46] James F. Kurose, Keith W. Ross. 计算机网络—自顶向下方法与 Internet 特色[M]. 机械工业出版社, 2005.

[47] Braden R, ISI, Borman D, et al. Computing the Internet Checksum[S]. RFC 1071, 1988-09.

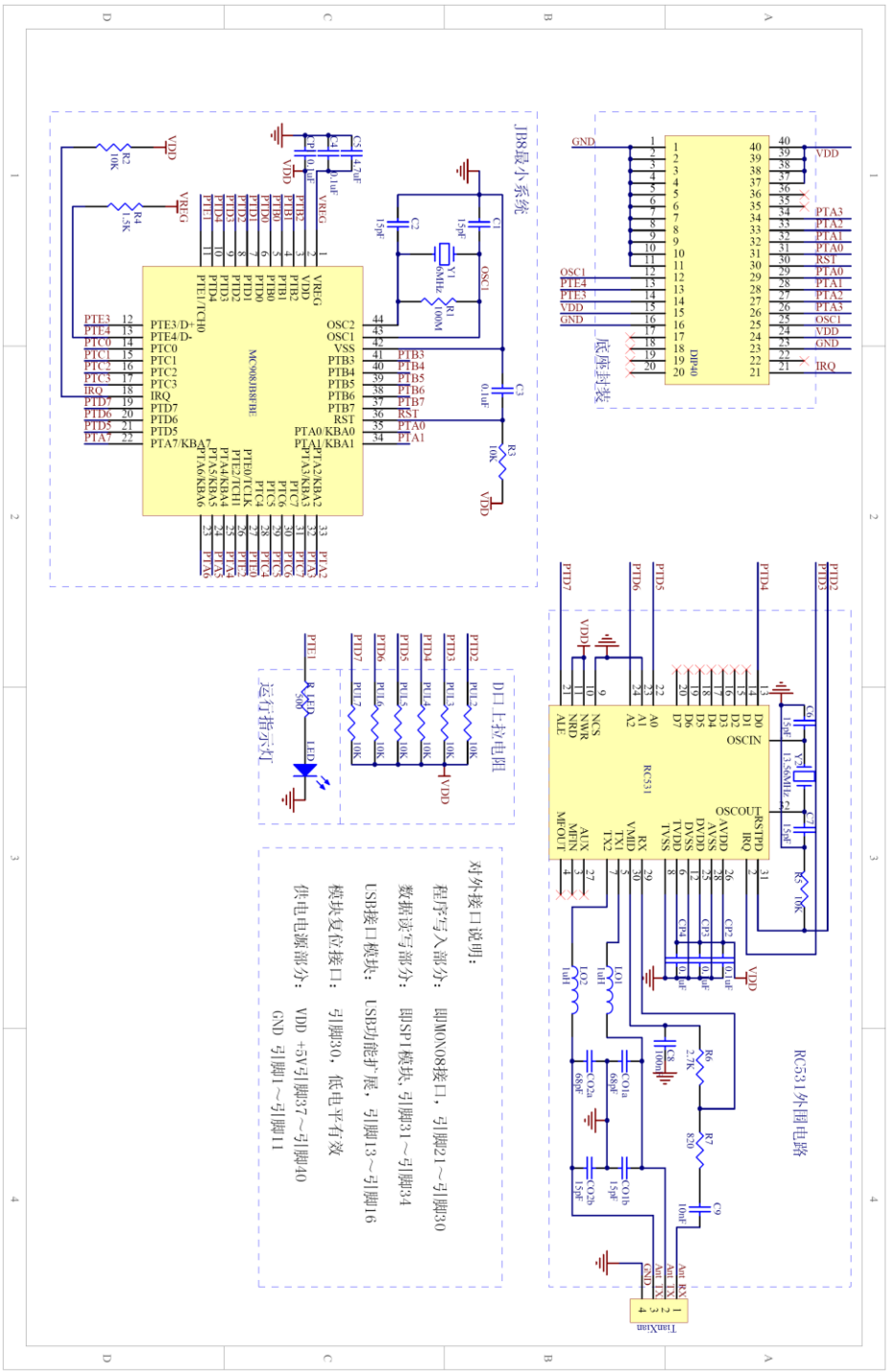
[48] 杨永辉. 智能大厦[M]. 北京邮电学院出版社, 2002.

[49] 朱学莉. 智能建筑网络通信系统[M]. 中国电力出版社, 2006.

[50] 王汝琳. 智能门禁控制系统[M]. 电子工业出版社, 2004.

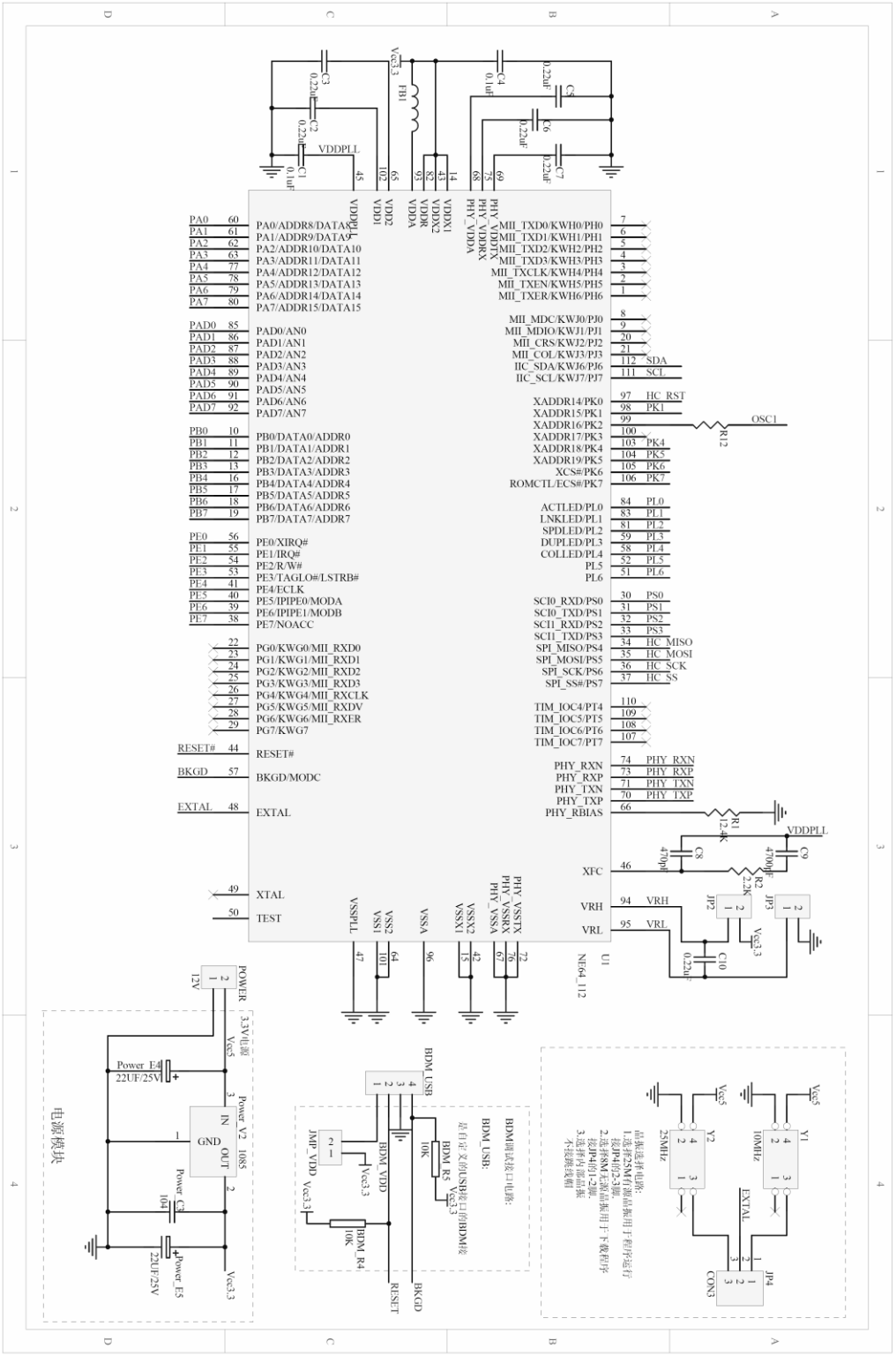
附录 A 读写器原理图

A.1 读写模块原理图

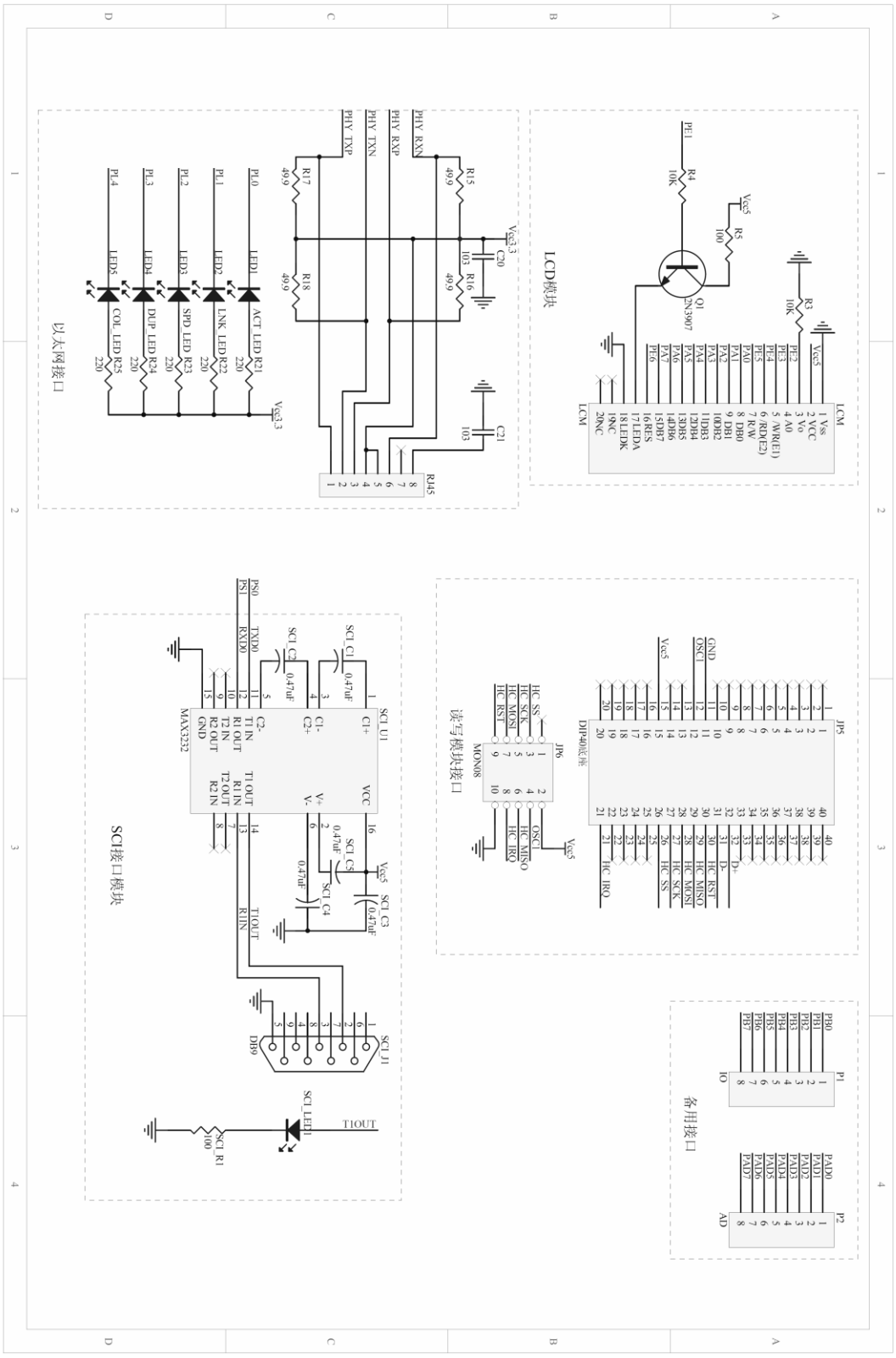


A.2 主控系统原理图

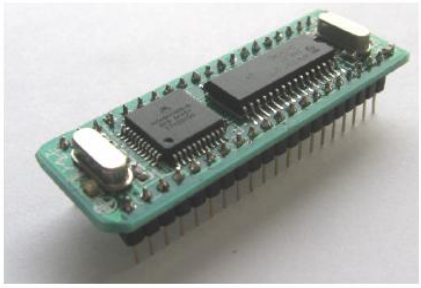
A.2.1 MC9S12NE64最小系统



A.2.2 主控系统外围电路



附录 B 读写器实物图



读写模块硬件中间件实物图

读写器实物图

攻读硕士学位期间公开发表的论文及参与的鉴定项目

[1] 孙鹏、王宜怀．基于嵌入式以太网的二代身份证读卡器设计．军民两用技术与产品，2008(已录用)

[2] 孙鹏、于鹏、祝叶、陈姝．串励、他励电机控制器测试平台的设计．苏州大学第九批大学生课外学术科研基金资助项目一等奖．项目编号：KY2006123B

[3] 参与王宜怀、刘晓升等编著的《嵌入式系统-使用 HCS12 微控制器的设计与应用》中第 10 章的撰写．北京航空航天大学出版社,2008

致 谢

三年的时光转瞬即逝，在此我首先要感谢我的导师王宜怀教授。在我攻读硕士学位期间，王老师不仅为我提供了良好的学习条件和实践锻炼机会，还非常注重对我学习方法的培养。在他的严格要求和悉心指导下，我逐步掌握了如何规范地进行嵌入式系统的开发，并且具备了一定的分析问题、解决问题的能力。王老师平时严谨的治学态度、执着的科研精神和朴素的生活作风，更是让即将走上工作岗位的我体会并懂得了做人的道理。

衷心感谢刘晓升老师对我生活上的关心和学习上的帮助。

真诚感谢实验室的兄弟姐妹们，感谢你们营造的浓厚学习氛围。感谢郑洪静、陈龙和张琴同学在本文研究过程中给予的无私帮助。感谢曹振华、葛强、孟忠伟和严健同学在论文修改期间提供的大力支持。

特别感谢我的父母，你们的爱是我前进的动力。

最后，向审阅本文的专家、教授致敬！