中山大學
SUN YAT-SEN UNIVERSITY

本科生毕业论文（设计）

题　　目：基于 MF RC500 和 STM32 的 RFID 读写器设计与实现

院　　系：数据科学与计算机学院

专　　业：软件工程（移动信息工程）

学生姓名：陈胜杰

学　　号：12353022

指导教师：胡建国 博士

二〇一六 年 五 月

附表一、毕业论文开题报告

论文（设计）题目：基于 MF RC500 和 STM32 的 RFID 读写器

近几年来，随着电子技术的不断发展，物联网技术在产业发展中占着越来越重要的角色，其中，RFID 技术作为物联网技术的基础，应用范围越来越广，应用场合也越来越多，因此，对其进行研究具有很大的现实意义。此外，市面上普遍存在的基于 MFRC500 的低成本读卡器使用 51 单片机等低端微处理器，功能单一，不易扩展。考虑到一款基于 ARM®Cortex™-M3 32 位 RISC 内核的高性能微处理器——STM32，同样满足低成本要求，因此，设计一款基于 STM32 和 MFRC500 的读卡器具有研究和现实意义。

我国对低高频 RFID 技术的研究还是比较成熟的，13.56MHz 下的射频读卡模块的天线及其匹配电路设计具有较为完善的设计、调试流程，实现一款可行的读卡器存在可能。

综上，设计一款基于 MFRC500 和 STM32 的读卡器具有一定的可行性和挑战性，有利于锻炼学生的动手能力和创新能力，也是对本科四年所学的一个系统回顾和展示。

学生签名：　　年　月　日

指导教师意见：

1、同意开题（ ） 2、修改后开题（ ） 3、重新开题（ ）

指导教师签名：　　年　月　日

附表二、毕业论文过程检查情况记录表

指导教师分阶段检查论文的进展情况（要求过程检查记录不少于 3 次）：

第 1 次检查

学生总结：

指导教师意见：

第 2 次检查

学生总结：

指导教师意见：

第 3 次检查

学生总结：

指导教师意见：

第 4 次检查

学生总结：

指导教师意见：

学生签名：　　年　　月　　日

指导教师签名：　　年　　月　　日

总体完成情况	指导教师意见： 1、按计划完成，完成情况优（ ） 2、按计划完成，完成情况良（ ） 3、基本按计划完成，完成情况合格（ ） 4、完成情况不合格（ ） 指导教师签名：　　年　　月　　日

附表三、毕业论文答辩情况

答辩人		专 业	
论文（设计）题目			
答辩小组成员			

答辩记录：

记录人签名：　　年　　月　　日

学术诚信声明

本人所呈交的毕业论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料均真实可靠。除文中已经注明引用的内容外，本论文不包含任何其他人或集体已经发表或撰写过的作品或成果。对本论文的研究作出重要贡献的个人和集体，均已在文中以明确的方式标明。本毕业论文的知识产权归属于培养单位。本人完全意识到本声明的法律结果由本人承担。

本人签名： 日期：

论文题目：基于 MF RC500 和 STM32 的 RFID 读写器设计与实现

专　　业：软件工程（移动信息工程）

学生姓名：陈胜杰

学　　号：12353022

指导教师：胡建国 博士

摘 要

RFID(Ratio Frequency Identification)是一种非接触的自动识别技术，利用其射频信号空间耦合的传输特性能够识别高速移动的物体，能同时进行多目标物体识别，无需接触。RFID 标签可快速读写，能携带大量数据，磨损概率基本为零。正是由于这些优势，RFID 技术在日常生活中的应用越来越广泛，具有极大的研究价值。

目前市面上流行的基于 MFRC500 的读写器往往搭载在性能较弱的 51 单片机上，成本虽然不高，但是设备升级与扩展则举步维艰。本文延续低成本的需求，选择性能更为强大的 STM32 微控制器，从硬件到软件上对这样一款读写器进行设计与实现。

1) 完成了射频读卡模块的原理图设计，包括射频芯片外围电路设计、天线设计和天线匹配电路设计，并完成该模块的 PCB 设计与焊接，与 STM32 开发板组成本文设计的实验板。
2) 成功使用该读写器实验板结合所设计的下位机/上位机软件实现对 Mifare One 卡的密钥验证、读写卡和电子钱包等操作。
3) 设计并实现了一套用于此读写器应用系统开发的 API(Application Programming Interface)，方便快速进行二次开发。

实际运行结果表明，本文设计的 RFID 读写器能够完成 Mifare One 卡的所有操作，并且可以利用提供 API 快速开发诸如考勤系统这样的 RFID 应用系统。

关键词：RFID；MF RC500；STM32；API

Title: Design and Implementation of RFID Reader Based on MFRC500 and STM32
Major: Software Engineering (Mobile Information Engineering)
Name: Shengjie CHEN
Student ID: 12353022
Supervisor: Dr. Hu Jianguo

Abstract

RFID (Ratio Frequency Identification) is a non-contact automatic identification technology using the spatial coupling transmission characteristics of RF signals to identify fast-moving and multi-target objects (without any contact). In addition, RFID tags, which can carry large number of data, can be read quickly under the RFID technology, and its wear rate is almost zero. Thanks to these advantages, RFID technology is widely used in our daily life, and thus, is of great research value.

In the current market, RFID readers based on MFRC500 and microcontroller 51, with poor performance, are popularly. The cost is not high, but it is very difficult to upgrade and extend in those design. Follow the low cost demand, this paper, from hardware to software, introduces the design and implementation of a reader based on MFRC500 and the more powerful STM32 microcontroller. You will see the successful use of the reader breadboard, composed with a STM32 development board and a self-design and self-weld RF reader module, binding the designed embedded software and testing software to realize key authentication, read/write operation, electronic purse and all other operations of Mifare One card. Also, the design and implementation of API (Application Programming Interface) set for this RFID reader application system development will be introduced as well, making the secondary development more convenient.

The results show that the design of this RFID reader can complete all operations of Mifare One card. Besides, RFID applications like attendance system can be developed rapidly using provided API.

Keywords: RFID, MF RC500, STM32

目 录

摘 要 I

ABSTRACT II

第 1 章 引言 1

1.1 选题背景与意义 1
1.2 国内外研究现状和相关工作 2
1.3 本文的研究内容与主要工作 2
1.4 本文的论文结构与章节安排 2

第 2 章 RFID 系统理论基础 4

2.1 RFID 系统工作原理 4
2.2 编码解码 6
2.3 调制解调 7
2.4 ISO/IEC 14443 标准 10
2.5 本章小结 16

第 3 章 系统硬件设计 17

3.1 微控制器模块 17
3.2 射频读卡模块 18
3.3 系统 PCB 设计 28
3.4 本章小结 29

第 4 章 系统软件设计 30

4.1 分层原理下的软件设计 30
4.2 下位机软件设计 31
4.3 串口通信协议 39
4.4 应用程序 API 40
4.5 上位机应用程序设计 42
4.6 本章小结 43

第 5 章 系统功能测试 45

5.1 测试准备工作.....45
5.2 系统测试.....45
5.3 测试结果分析.....49
5.4 本章小结.....50

第 6 章 总结与展望.....51

6.1 工作总结.....51
6.2 研究展望.....51

参考文献.....53

致 谢.....54

第 1 章 引言

近几年来，随着电子技术的不断发展，物联网技术在产业发展中占据越来越重要的角色。在此趋势之下，各国齐头并进，相继推出区域战略规划，美、日、韩、欧盟等都正投入巨资深入研究探索物联网，并启动了以物联网为基础的"智慧地球"、"U-Japan"、"U-Korea"、"物联网行动计划"等国家性区域战略规划。目前，物联网技术的研究已经取得了明显进步，其中，RFID 技术作为物联网技术的基础，应用范围越来越广，应用场合也越来越多，因此，对其进行研究具有很大的现实意义。

1.1 选题背景与意义

RFID 是一种非接触的自动识别技术，利用其射频信号空间耦合的传输特性实现对物体的自动识别，识别过程无需光学可视，无需人工管理即可完成信息的录入和处理。

RFID 技术具有很多优点，例如能够识别高速移动的物体，能同时进行多目标物体识别，无接触的操作方式，磨损概率基本为零，电子标签具有较长的使用寿命等。此外，RFID 标签可快速读写、能携带大量数据，保密性强，且具有不怕污渍、灰尘，防水、防磁等较强的环境适应力，这是条码、磁卡、接触式 IC 卡、光学识别和生物识别技术(如虹膜、声音、指纹)等同期或早期的识别技术所无法比拟的。

正是由于这些优势，RFID 技术在产品生产、仓储物流、交通运输、金融支付、身份识别、安全和访问控制等领域有着广泛的应用，其实现的主要功能有电子车票、物品防伪、人员定位、图书管理、人员出入管理、汽车防盗、Factory Automation 等，应用前景十分可观，具有极大的研究价值。

如今，得益于摩尔定律的存在，RFID 标签的成本越来越低，存储容量更大，更易使用，各种 RFID 新技术得到迅速发展。随着 RFID 技术应用渗透到日常生活中，其发展也备受关注。RFID 标签从 RFID 读写器外部射频场中获取能量，标签

的低功耗技术成为关注的焦点；多标签和多读写器等的防冲突通信协议也是研究的热点；伴随着 RFID 技术在金融领域中的应用越来越广泛，RFID 技术安全性研究更是炙手可热。

1.2 国内外研究现状和相关工作

当今世界，美国的 RFID 技术最为先进，走在世界前列，有关 RFID 的标准建立，相关软硬件技术的研发及其相关产品的应用均处于世界领先位置[1]。欧洲国家的 RFID 技术几乎与美国并驾齐驱，但是相关的 RFID 技术标准仍然采用的是由美国制定的 EPC global 标准。在亚洲国家中，日本的 RFID 技术较为先进，已经制定出 UID 标准，但是还不具备成为国际标准的水平[2]。在全球范围内，美国、德国、英国、瑞典、瑞士、日本、南非目前均有比较成熟且先进的 RFID 系统。

虽然全球的 RFID 技术发展迅速，但我国起步较晚，整体研发水平不高，与发达国家的技术水平还存在较大的差距。目前，国内研发生产 RFID 产品的企业总数不算少，但关键是缺乏核心技术，尤其是在超高频 RFID 产品的研发方面，技术更是落后[3]。我国对低高频 RFID 技术的研究还是比较成熟的，不过关键技术还有待进一步完善，产品可靠性也需要进一步加强，其中 RFID 终端功能单一，操作不便，安全性低，仍需完善。

1.3 本文的研究内容与主要工作

本文在学习 RFID 系统理论的基础上，以低成本、高性能、可扩展为导向，设计并实现了一款基于 MFRC500 和 STM32 的读写器，打破市面上普遍存在的使用 51 单片机等低端微处理器的束缚。由于 STM32 微处理器在性能上的优势，本文设计的读写器在升级、扩展上也具有优势。此外，以动态链接库形式提供的 API，极大地方便了应用系统的开发。

1.4 本文的论文结构与章节安排

本文共分为六章，章节内容安排如下：

第一章主要介绍了课题的研究背景和意义，在阐述国内外 RFID 技术研究现状的基础上，提出了本文的研究内容。

第二章是对本文涉及的 RFID 系统理论基础的讲解，包括 RFID 系统的工作原理，编解码、调制解调等关键技术以及 ISO/IEC 14443 协议。

第三章重点讲解了本文设计的 RFID 读写器的硬件结构，针对射频读卡模块各部分电路设计及其涉及到的技术进行详细说明。

第四章重点对本文设计的 RFID 读写器的软件设计进行说明，从 STM32 底层板的下位机程序，上位机的通讯协议，应用程序 API 到上位机测试程序的设计与实现都进行了仔细的阐述。

第五章主要是对系统的整体测试，对本文设计需实现的功能的测试过程进行了详细说明。

第六章对全文进行总结，并对进一步的工作进行展望。

第 2 章　RFID 系统理论基础

为了更好地完成 RFID 读写器的设计与实现，我们需要先了解和学习有关 RFID 系统的理论知识。本章将从 RFID 系统的工作原理开始讲述，以此引出对编解码、调制解调两项关键技术的阐述。最后，对本设计中涉及的 ISO/IEC 14443 协议进行详细介绍和说明。

2.1 RFID 系统工作原理

本文设计的高频 RFID 系统符合 ISO/IEC 14443 标准，其能量传递和数据传输是基于电感耦合原理，电感耦合原理与变压器原理相同。通过共同的磁场空间，能量和数据从读写器天线传送到标签天线，遵循法拉第电磁感应定律，如下图所示。

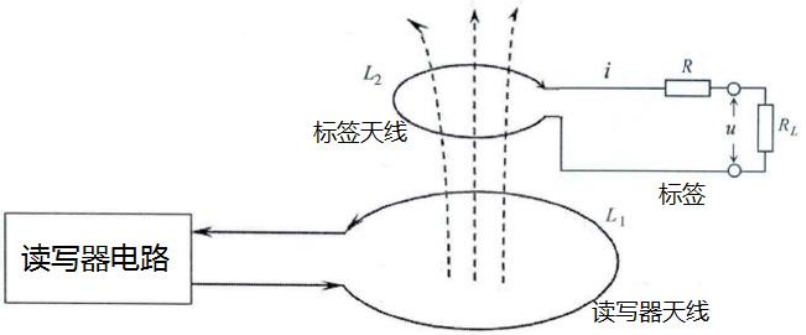


图 2-1：RFID 电感耦合方式的工作原理示意图

L_1 是读写器的发射天线，L_2 是标签的天线线圈，R 表示标签天线线圈的电阻值，R_L 表示标签负载电阻。高频的强电场由读写器的天线线圈产生，穿过线圈横截面积和线圈周围的空间。发射磁场的一部分磁场线穿过距读写器天线线圈一定距离的标签天线线圈，通过感应，在标签的天线线圈上产生感应电压 u，将其整流后作为标签芯片的初级电源，为标签工作提供能量[4]。

标签天线两端的感应电压 u 由读写器发射的磁场、标签天线自身参数和其内部电路消耗的电流共同决定。在标签与读写器天线线圈距离一定的情况下，要获得越大能量，就应当保证标签内部自身欧姆电阻越小越好，读写器天线发送电流越大越好。下图是更为复杂的射频模拟前端工作原理示意图。

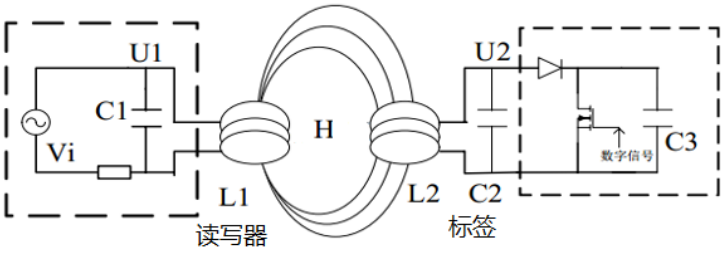


图 2-2：RFID 射频模拟前端工作原理示意图

读写器通过自身天线 L2 向四周发射高频的调制信号，产生高频的电磁场 H，使用的载波频率为 13.56MHz。发射磁场的很小一部分穿过标签天线 L2，并在 L2 上感应出一个交变电压 U2，U2 整流后作为标签芯片的初级电源。C1 表示与天线 L1 构成并联谐振回路的匹配电路部分，该回路的谐振使得 L1 上输出的电流最大化，以产生最大的场强驱动远距离标签工作；C2 代表与标签天线 L2 匹配的电路部分，调谐到载波频率，产生一个无限小的阻抗，以感应出一个最大化的电压 U2 来驱动标签内部电路的正常工作。

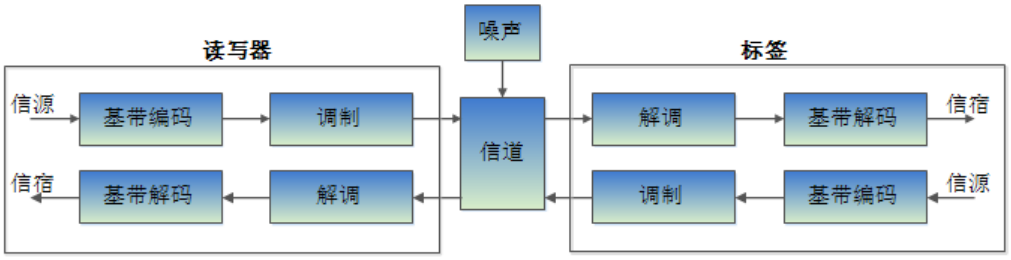


图 2-3：RFID 系统的基本通信过程

图 2-3 展示了读写器和标签之间数据传送的通信过程。读写器将要发送的消息，编码后经载波信号，传送给读写器天线 L1，读写器天线将信号发射给进入可识别区域的电子标签，与此同时，进入该区域的电子标签产生感应电流，获得能量并被激活，电子标签将自身数据通过标签天线 L2 发射给读写器；读写器天线接收到从标签发送来的信号后，将该信号传送到读写器信号处理模块，经解调和解码后将有效数据传送到应用系统进行相关处理[5]。由于读写器发射的信号为 ASK 调制信号，因此接收到的 U2 为带有调制信息的 ASK 信号，经过标签芯片模拟部分解调后即可送入数字部分处理；标签通过控制自身天线上负载电阻的导通与否对天线上的电压 U2 进行振幅键控，产生 ASK 调制信号。读写器天线检测该调制信号，对其进行解调后送入射频芯片的数字部分进行处理[6]。

本章接下来的部分，将对上述过程中的几个环节，包括编码、解码和调制、

解调作进一步说明，最后会在此基础上，介绍与本设计密切相关的 ISO/IEC14443-A 标准。

2.2 编码解码

为了使 RFID 系统传输的信号与空间信道的特性相匹配，需要对要传输的基带信号进行编码，基带编码以不同的电平形式表示二进制数"0"和"1"[7]。在工作频率为 13.56MHz 的高频 RFID 系统中，常用的基带编码形式包括电平不归零(NRZ-L)编码、曼彻斯特(Manchester)编码和米勒(Miller)编码。下面对本设计涉及的后两种编码进行简单说明。

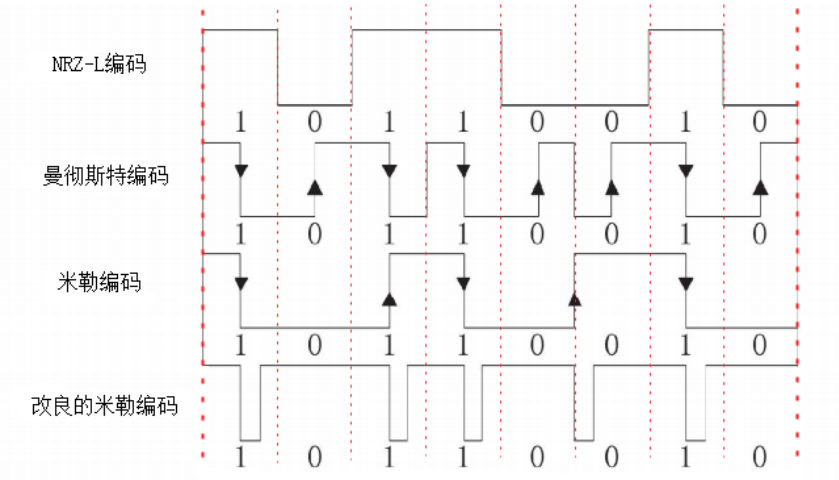


图 2-4：RFID 系统中常用的基带编码方式

Manchester 编码：通过某数据位半个周期时的电平变化表示该数据位的值，在半个周期时出现下降沿表示二进制数"1"，出现上升沿表示"0"。这种编码方式在每个数据位都有电平变化，不存在直流分量；通过这种电平变化，容易发现数据传输过程中出现的错误，若有多个标签同时发送数据，则可判断冲突出现的位置；同时，可以根据电平变化提取同步时钟。

Miller 编码：在半个周期时出现的任意电平变化表示二进制数"1"，在半个周期时电平不变化表示"0"；若有连续的"0"，则在周期开始时电平发生变化。在 RFID 系统中，为了保证标签能从读写器获取恒定的能量，采用的改进 Miller 编码用一个负脉冲代替了 Miller 编码中的边沿。

选取编码方式时需要特别考虑的是必须保证读写器到标签的能量供应不能中断，因此，通常采用信号电平在相邻数据位之间存在跳变的编码方式。此外，

调制后的信号带宽/频宽和对数据传输错误的敏感性也是需要考虑的方面[7]。

2.3 调制解调

为了按照特定的耦合方式,以特定的频率在空间信道传输 RFID 系统的信息，需要对经过基带编码后的信号进行调制[7]。由于读写器必须输出一个能提供足够能量并且信噪比很高的信号给标签；而标签由于成本、功耗和尺寸上的限制，片内只能有一个简单的接收器，所以读写器发送的调制信号也必须简单，容易被检测到。正是为了达到上述要求，RFID 系统中一般都采用振幅键控(ASK)的调制方法[4]。

2.3.1 读写器➔标签的数字调制

ASK 调制通过使载波的幅度随着调制信号变化实现数字调制。当调制信号为简单的二进制"1"、"0"数字信号时，即为 2ASK。2ASK 调制系统的调制度为：

$$m = 1 - \frac{A_{min}}{A_{max}} \quad (2.1).$$

其中，A_{max} 为传输二进制"1"时载波信号的幅度，A_{min} 为传输二进制"0"时的幅度，A_{min} 可以取 A_{max} 和 0 之间的值。若 2ASK 调制系统的调制度为 100%，则可使用数字开关简单地实现调制，这种调制系统称为通—断键控(On-Off Keying)。

在 RFID 系统中，需要保证读写器向标签发送的载波能量供应的可靠性和数据传输的可靠性。如果采用较大的调制度，为了防止读写器连续发送二进制"0"时载波能量过小，需要采用更为复杂的编码方法，这会使标签解码电路复杂化，提高标签成本,增大标签功耗;如果采用较小的调制度,读写器发送的二进制"0"、"1"信号则难以区分，使 RFID 系统的数据传输变得不可靠[7]。

通过带通滤波和包络检波，可以实现 2ASK 信号的解调。

2.3.2 标签➔读写器的副载波负载调制

标签由于尺寸、功耗等原因，不可能在内部设计一个类似读写器射频芯片的调制电路，所以不能使用传统的 ASK 调制方法，在 RFID 系统中，一般都采用负载调制的方式来返回数据[4]。如下图副载波负载调制原理示意图：

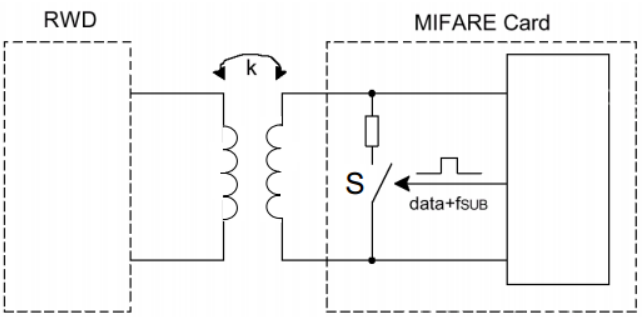


图 2-5：副载波负载调制原理示意图

标签没有信号调制时 S 断开，此时读写器天线线圈和标签天线线圈都有大小恒定的电流通过。标签开始调制时，S 闭合，标签整体等效电阻减小。将标签对读写器的影响可假设为一阻抗 Z_{T2}，如下图所示，此阻抗实由互感所引起的。

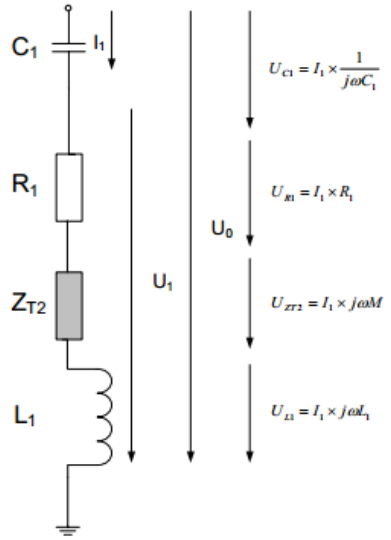


图 2-6：读写器工作时天线负载等效示意图

当频率为谐振频率时，读写器上的电感与电容的阻抗相消只剩下实部阻抗 R_1，标签芯片同样处于谐振状态，其阻抗也只剩下等效欧姆电阻部分[4]。该部分电阻减小，标签上的能量消耗发生改变，使读写器端出现电压变化，从而使读写器天线电流发生变化[8]。令开关 S 受传输数据控制，当信号为"1"时，开关开路；当信号为"0"时，开关闭合。同时在读写器天线处检测电流的变化情况，则可以得到标签返回的正确信号[4]。

此外，对于工作频率为 13.56MHz 的电感耦合 RFID 系统来说，读写器发射出的载波信号与接收到的负载调制信号的幅度差距很大，为了从负载调制信号中获取有用信息，需要检测其微弱的电压变化，因此需要采用副载波调制技术。将标签上经过基带编码的信号首先调制到副载波频率 f_{SUB} 上(一般是通过对 RFID 系统

的工作频率f_C进行二进制分频产生f_{SUB})。如下图所示，在使用副载波进行负载调制时，在 RFID 系统工作的中心频率f_C(= 13.56MHz)两侧距离±f_{SUB}(=f_C/16 = 847kHz)的位置上产生两个调制边带，通过带通滤波器可将调制边带与载波信号分离开。

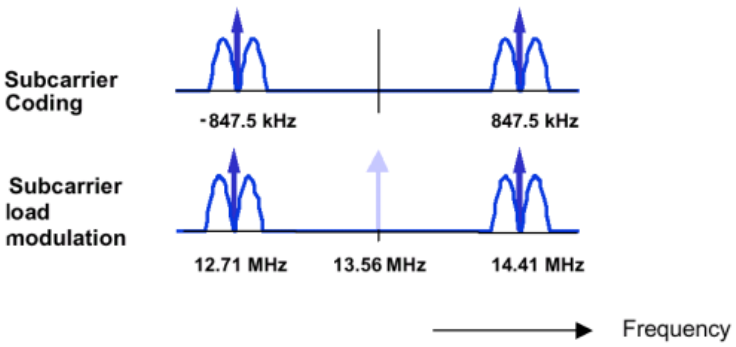


图 2-7：副载波及标签副载波负载调制信号频域分析图

以上分析已经提到，标签通过副载波负载调制返回的信号是十分微弱的，所以如果采用传统的非相干包络检波法很难解调出实际的数据，所以需要采用信号幅度、相位一起检测的相干检测方式。通常在 RFID 系统中，读写器使用正交解调处理标签返回的信号。

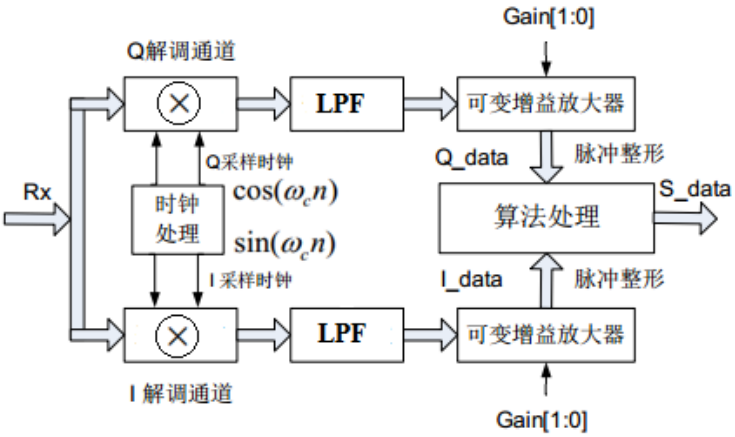


图 2-8：读写器正交解调示意图

正交解调使用两个不同的时钟 I-和 Q-时钟，他们之间的相位差为 90° 。与读写器天线线圈连接的 Rx 引脚接收到标签返回的信号，该信号进入系统后与两正交信号进行相乘；低通滤波器通过选择保留副载波频率的数据信号，滤除 13.56MHz 的载波信号，然后经过可变增益放大器放大到合适的幅度最后将 I、Q 两路信号通入算法处理模块(相关性电路)，根据解调算法(求出相关性结果)恢复出基带信号中的二进制码元信息，数字化后输入到数字模块进行处理和传送[4]。

2.4 ISO/IEC 14443 标准

非接触式 IC 卡根据作用距离的不同分为如下表所示的三种不同的国际标准。

标准	卡类型	作用距离（约）
ISO/IEC 10536	密耦合	0~10 MM
ISO/IEC 14443	近耦合	0~100 MM
ISO/IEC 15693	疏耦合	0~1000 MM

表 2-1：非接触式 IC 卡国际标准

本文使用的近耦合 IC 卡采用 ISO/IEC 14443 国际标准，作用距离在 10cm 左右，本节将着重介绍该标准内容[9]。ISO/IEC 14443 标准包含了以下四个部分：物理特性、射频能量和信号接口、初始化和防冲突、传输协议[10]。下文将对射频能量和信号接口、初始化和防冲突两部分进行说明。

2.4.1 射频能量与信号接口

在标准的这一部分中规定了需要供给能量的场的性质和特征，以及近耦合设备(PCDs)和近耦合卡(PICCs)之间的双向通信。

① 近耦合卡的初始化对话

PCD 和 PICC 之间的初始化对话通过下列连续操作进行：

- PCD 发出的射频场激活 PICC
- PICC 等待来自 PCD 的命令
- PCD 发出一个命令
- PICC 发回一个应答

这些操作使用下面段落中规定的射频能量和信号接口。

② 能量传送

PCD 产生耦合到 PICC 的 RF 电磁场，用以传送能量，并且被调制来用于通信。标准规定了射频工作频率f_c是 13.56MHz ± 7kHz。PCD 设备产生的磁场在阅读距离内任何地方的强度都不应小于最小场强和大于最大场强。在此范围内 PICC 应能不间断的连续工作。最小场强 H_{min}，其值为 1.5A/m(有效值)；最小场强 H_{max}，其值为 7.5A/m(有效值)。

③ 信号接口

有两种通信接口 Type-A 和 Type-B。符合标准的读写器必须同时支持这两种

方式，以便支持所有的 IC 卡[10]。读写器应在空闲状态时根据检测到的卡的类型是 Type-A 或 Type-B 而变换调制方式，而只允许其中一种通信的信号接口可以被激活。本文设计的读写器涉及到的是 Type-A，所以会对 Type-A 的通信接口进行重点介绍。

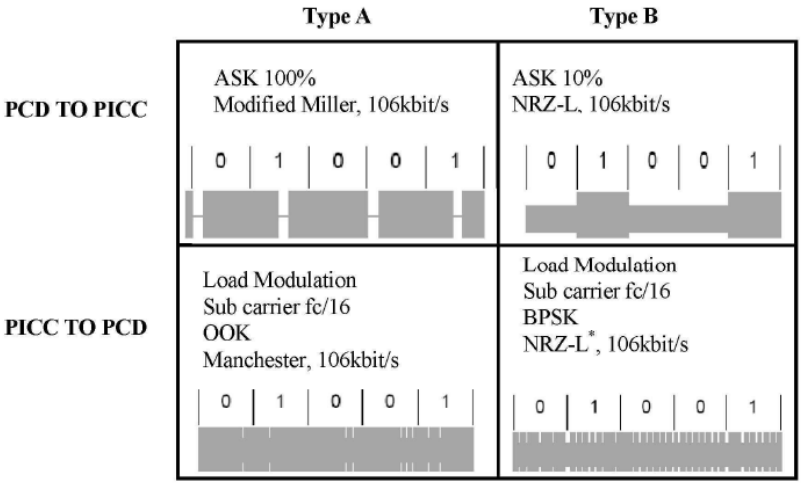


图 2-9：Type-A 和 Type-B 接口的通信信号举例

对于 Type-A 标准，

A. PCD 传送到 PICC 的信号

1. 数据传输率

载波频率f_c为 13.56MHz，数据传输率为 13.56MHz/128 = 106kbit/s，1 bit 数据所占的时间周期为 9.44us(128 个周期的载波信号)。

2. 调制方式

采用 ASK 调制方式，调制度为 100%，在 RF 场中产生"空隙" (Pause)来传送二进制数据。

3. 位的表示与编码

从 PCD 到 PICC 的通信采用改进的 Miller 编码方式，定义三种时序：

时序 X：一个"空隙"在半个位周期后产生；

时序 Y：在整个位周期期间不产生"空隙"；

时序 Z：在位周期的开始处产生一个"空隙"。

用以上时序进行以下信息的编码：逻辑"1"用时序 X 表示，逻辑"0"用时序 Y 表示，但有两种例外情况：(1)、假如相邻有 2 个或更多的"0"，从第 2 个"0"开始(包括其后面的"0")，采用时序 Z 表示；(2)、假如在帧的起始位后的第 1 位为"0"，则用时序 Z 来表示这一位和直接跟随其后的"0"。

通信开始时用时序 Z 表示，通信结束时用逻辑"0"后跟时序 Y 表示(这正是例外情况(2)存在的原因)，无信息用至少两个时序 Y 表示。

B. PICC 传送到 PCD 的信号

1. 负载调制

PICC 通过电感耦合区与 PCD 进行通信。在 PICC 中，利用 PCD 发射的载波频率生成副载波，副载波是在 PICC 中通过连接/断开负载的方法实现调制的，其中负载调制的幅度应至少为 $30H^{1.2}$ mV(峰值)，H 是以 A/m 为单位的磁场强度的有效值。副载波的频率 f_{SUB} 为 $f_c/16$(约 847kHz)，在初始化和防冲突阶段，1bit 码字时间等于 8 个副载波周期。

2. 数据传输率

在初始化和防冲突阶段，1bit 数据由 8 个周期副载波信号表示，数据传输率为 $f_c/16/8 = f_c/128$，也是 106kbit/s。

3. 位的表示与编码

采用 Manchester 编码方式，定义如下三种时序：

时序 D：载波被副载波在位宽度的前半部(50%)调制；

时序 E：载波被副载波在位宽度的后半部(50%)调制；

时序 F：在整个位宽度内载波不被副载波调制。

用以上时序进行以下信息编码：逻辑"1"用时序 D 表示，逻辑"0"用时序 E 表示，通信开始用时序 D 表示，通信结束用时序 F 表示，无信息无副载波。

表 2-3 和表 2-3 对两种标准的通信接口进行比较。

PCD ➔ PICC	Type-A	Type-B
调制	ASK 100%	ASK 10%
位编码	改进的 Miller 编码	NRZ-L 编码
同步	位级同步(帧起始、帧结束标记)	每个字节有一个起始位和结束位
波特率	106kbit/s	106kbit/s

表 2-2：读写器(PCD)到卡(PICC)的数据传输

PICC ➔ PCD	Type-A	Type-B
调制	用振幅键控调制 847kHz 负载调制的副载波	用相位键控调制 847kHz 负载调制的副载波
位编码	Manchester 编码	NRZ-L 编码
同步	1 位帧同步(帧起始、帧结束标记)	每个字节有一个起始位和结束位
波特率	106kbit/s	106kbit/s

表 2-3：卡(PICC)到读写器(PCD)的数据传输

对于 Type-B 标准，PCD 到 PICC 的通信也采用 ASK 调制方式，调制度为 10%，数据传输速率也为f_c/128，采用 NRZ-L 编码方式，高电平表示逻辑"1"，低电平表示逻辑"0"。PICC 到 PCD 的通信也采用副载波的负载调制方式，副载波频率和负载调制的幅度与 Type-A 相同，但调制和编码方式与 Type-A 存在区别。Type-B 采用 BPSK 调制方式，通过相对于参考时刻Φ_0的相位变化表示逻辑"1"和逻辑"0"，相位不变化表示逻辑"1"，相位变化 180° 表示逻辑"0"。此外，PICC 到 PCD 的通信也采用 NRZ-L 编码方式。

2.4.2 初始化和防冲突

在标准的这一部分中规定了 PCD 和 PICC 之间通信的初始化阶段所使用的帧格式以及使用的防冲突机制。对于标准中规定的两种信号接口，帧格式和防冲突机制都是不同的，同样地，下文会对 Type-A 的通信接口进行重点介绍。

当一个 A 型卡到达了读写器的作用范围内，并且有足够的供应电能，IC 卡将执行一些预置程序后进入 IDLE 状态。处于 IDLE 状态的 IC 卡不能对读写器传输给其他 IC 卡的数据起响应。IC 卡在 IDLE 状态接收到有效 REQA 命令，则回送对请求的应答 ATQA。当 IC 卡对 REQA 命令作了应答后，IC 卡处于 READY 状态[10]。

对于 Type-A 标准，采用二进制搜索算法实现防冲突。读写器首先发送 REQA 命令，所有处于读写器工作范围内的标签将同时发送 ATQA 应答，进入 READY 状态。读写器接收到多个标签的 ATQA 应答，发送 ANTICOLLISION 命令，进入防冲突循环，如下图 2-10 所示。

ANTICOLLISION 命令由三部分组成，包括选择代码 SEL、有效数字位数 NVB 以及由 NVB 指定的有效数据位 UID CLn(下面会详细介绍)。读写器首先发送 ANTICOLLISION 命令中的 SEL 和 NVB，所有处于 READY 状态的标签发送完整的 UID

应答，若有超过一个标签发送了 UID 应答，则说明产生了冲突，读写器识别出产生冲突的数据位置，发送 ANTICOLLISION 命令中的 SEL、NVB 和 UID CLn，以继续选择标签。只有符合读写器发送的 UID CLn 的标签才发送 UID 中的其他位。若仍存在冲突，则继续发送 ANTICOLLISION 命令，直到读出完整的标签 UID，标签将返回 SAK，表示接受选择由 READY 状态进入 ACTIVE 状态，完成防冲突循环。

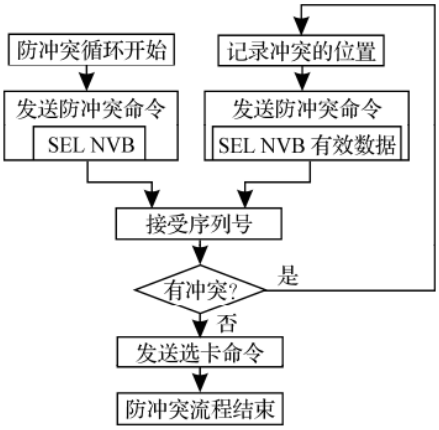


图 2-10：防冲突循环流程图

1. 帧的格式和时序

① REQA 帧格式

S	LSB-MSB	E

包括以下内容：

- 通信起始位 S；
- 7 位数据的 REQA 命令代码(0x26 或 0x52WUPA 命令)，低位先发送；
- 通信结束位 E。

② ATQA 帧格式

b_{16}	b_{15}	b_{14}	b_{13}	b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1
RFU								UID 大小		RFU	比特帧防冲突				

b_8 b_7表示 UID 比特帧(将在③中介绍)的大小。UID 大小不是固定的，可以由 1、2 或 3 部分组成，对应的 b_8 b_7分别为 00、01 和 10。b_5-b_1有且只有一位置为 1，表示采用的是比特帧防冲突的方式。RFU 为保留位，均置为 0。

③ ANTICOLLISION 命令帧格式

SEL	NVB	UID CLn	BCC

1 字节	1 字节	0-40bits	1 字节

SEL 表示 UID CLn 的层叠级数，其编码如下表：

b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	说明
1	0	0	1	0	0	1	1	"93"选择 UID CL1
1	0	0	1	0	1	0	1	"95"选择 UID CL2
1	0	0	1	0	1	1	1	"97"选择 UID CL3

NVB 表示 UID CLn 中的有效数据位数。BCC 只有在 UID CLn 为 40bit 才有，是前面 5 个字节的异或，此时为 SELECT 命令(NVB=0x70)；不足 40bit，则为 ANTICOLLISION 命令。

④ SAK 帧格式

b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1
RFU	RFU		RFU	RFU		RFU	RFU

其中 b_3(层叠位)表示 UID 是否完整，0 表示完整，即 PICC 的 UID 已被 PCD 所确认；1 表示还有部分 UID CLn(n=2 或 3)未经确认。RFU 全置为 0。

2. 防冲突流程

① PCD 根据当前级联防冲突的层数(UID CL1、CL2、CL3)指定 ANTICOLLISION 命令中 SEL 的代码。

② PCD 指定 NVB 的值为 0x20，此值表示 PCD 不发出 UID CLn 的任一部分，而迫使所有在 PCD 工作范围内的 PICC 发回完整的 UID CLn 作为应答。

③ PCD 发送经过①和②设置的 SEL 和 NVB。

④ 所有在 PCD 工作范围内的 PICC 发回完整的 UID CLn 作为应答。

⑤ 假如多于一张 PICC 返回应答，则发生了冲突；假如不发生冲突，可跳过⑥-⑩。

⑥ PCD 应认出产生第一个冲突的位置。

⑦ PCD 指示 NVB 的值说明 UID CLn 中的有效位数目，这些有效位是接收到的 UID CLn 发生冲突之前的部分，后面再由 PCD 添加一位"1"或"0"。

⑧ PCD 发送 SEL、NVB 和有效数据位。

⑨ 只有 UID CLn 部分与 PCD 发送的有效数据位内容相等的 PICC，才发送出 UID CLn 的其余位。

⑩ 假如还有冲突发生，重复⑥-⑨，最大循环啊次数为 32。

⑪ 假如没有再发生冲突，PCD 指定 NVB 为 0x70，此值表示 PCD 将发送完整的 UID CLn。

⑫ PCD 发送 SEL 和 NVB，接着发送 40bit 的 UID CLn，后面是其 CRC 校验码。

⑬ 与 40bit 的 UID CLn 匹配的 PICC，以 SAK 作为应答。

⑭ 如果 UID 是完整的，PICC 将发送带有层叠位为"0"的 SAK，同时从 READY 状态转换到 ACTIVE 状态。

⑮ 如果 PCD 检查层叠位为"1"，将 CLn 的 n 加 1，进入下一层防冲突循环。

对于 Type-B 标准，采用动态时隙 ALOHA 算法实现防冲突。当一个 B 型卡被置入读写器的作用范围内，IC 卡将执行一些预置程序后进入 IDLE 状态，等待接收有效的 REQB 命令；读写器首先发送 REQB 命令，所有处于读写器工作范围内的标签进入 READY 状态，根据 REQB 命令中的参数计算时隙数，并随机选择一个时隙等待发送 ATQB 应答；若在一个时隙内只有一个标签应答，则结束防冲突循环；若在一个时隙内存在多个标签应答，则重新开始新的防冲突循环。

2.5 本章小结

本章对本设计所涉及的所有理论基础进行了详细的阐述，包括 RFID 系统的工作原理介绍，与下位机软件设计中的场配置息息相关的编解码、调制解调核心技术的理论知识讲解。此外，重点对 ISO14443-A 协议进行详细说明，包括通讯帧格式以及初始化和防冲突流程，有助于对下文软件设计中 ISO14443 协议函数的理解。

第 3 章 系统硬件设计

为了实现对 Mifare One 卡的所有操作，在学习了 RFID 系统理论的基础上，采用 NXP 公司生产的 MFRC500 作为射频芯片，设计了读写器中的射频读卡模块，其硬件电路主要包括射频芯片外围电路、PCB 天线、天线匹配电路等。采用 ST 公司生产的 STM32 微处理器作为处理模块，整个 RFID 读写器系统的硬件架构图如下：

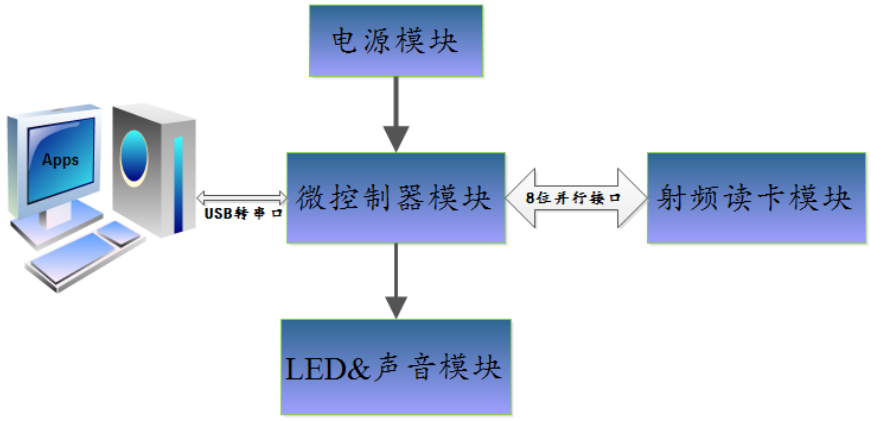


图 3-1：系统硬件架构图

PC 机通过 USB 转串口控制 STM32 微处理器，后者通过 8 位并行接口操作射频读卡模块完成对 M1 卡的各种操作，并将操作结果和数据通过 8 位并行接口传输到 STM32，进而返回给 PC 机做后续处理。

3.1 微控制器模块

项目微控制器模块暂使用的是野火 STM32 的开发板，以完成下位机软件编写与调试，并测试射频读卡模块硬件电路。

3.1.1 STM32

STM32 单片机是由 ST(意法半导体)公司生产的一款基于 ARM®Cortex™-M3 32 位 RISC 内核的高性能单片机，最近几年在工业自动化控制中应用非常广泛。开发板使用的 STM32F103VET6 芯片属于 STM32F103xE 增强型系列，使用高性能的 ARM®Cortex™-M3 32 位的 RISC 内核。工作频率为 72MHz，内置高速存储器(高达

512K 字节的 Flash 和 64K 字节的 SRAM)，丰富的增强 I/O 端口和连接到两条 APB 总线的外设。包含 3 个 12 位的 ADC、4 个通用的 16 位定时器和 2 个 PWM 定时器。还包含标准和先进的通信接口：多达 2 个 I2C、3 个 SPI、2 个 I2S、1 个 SDIO、5 个 USART、一个 USB 和一个 CAN。供电电压 2.0V 至 3.6V，一系列的省电模式保证低功耗应用的要求。

3.1.2 微控制器模块功能描述

微处理器模块的主要功能有：完成射频模块的初始化；对与 PC 端通讯的串口进行初始化；控制串口完成与上位机的数据通讯，接收并解析上位机命令，并通过控制射频模块完成对射频标签的各项操作，并将从射频模块获取的数据按照一定格式通过串口传输到 PC 端。

3.2 射频读卡模块

RFID 标签由于成本因素，结构较为简单，信号传输能力较弱，因此 RFID 读写器的射频接口是 RFID 系统中最关键的技术之一。在读写器识别标签时，应尽量使读写器天线和标签天线同时工作在相同频率(13.56MHz)的谐振状态下，这样有利于读写器天线传输的有效性，同时，可减少读写器与标签自身的能量损耗。此外，读写器的天线用于产生磁通量，以供应标签工作电源，并在读写器和标签之间传递信息。

射频读卡模块的设计主要涉及射频芯片的选型、射频芯片外围电路设计、天线设计、天线匹配电路设计这几个方面。

3.2.1 MF RC500

本文设计的射频读卡模块采用的射频芯片是 NXP 公司生产的 MF RC500 射频芯片。MF RC500 是应用于 13.56MHz 非接触式通信中高集成读卡 IC 系列中的一员。该读卡 IC 系列利用了先进的调制和解调概念，完全集成了在 13.56MHz 下所有类型的被动非接触式通信方式和协议。

MF RC500 支持 ISO14443A 所有的层。

内部的发送器部分不需要增加有源电路就能够直接驱动近操作距离的天线

(可达 100mm)。

接收器部分提供一个坚固而有效的解调和解码电路，用于 ISO14443A 兼容的应答器信号。

数字部分处理 ISO14443A 帧和错误检测(奇偶&CRC)。此外，它还支持快速 CRYPTO1 加密算法用于验证 MIFARE®系列产品。

方便的并行接口可直接连接到任何 8 位微处理器，这样给读卡器/终端的设计提供了极大的灵活性。

3.2.2 射频芯片外围电路设计

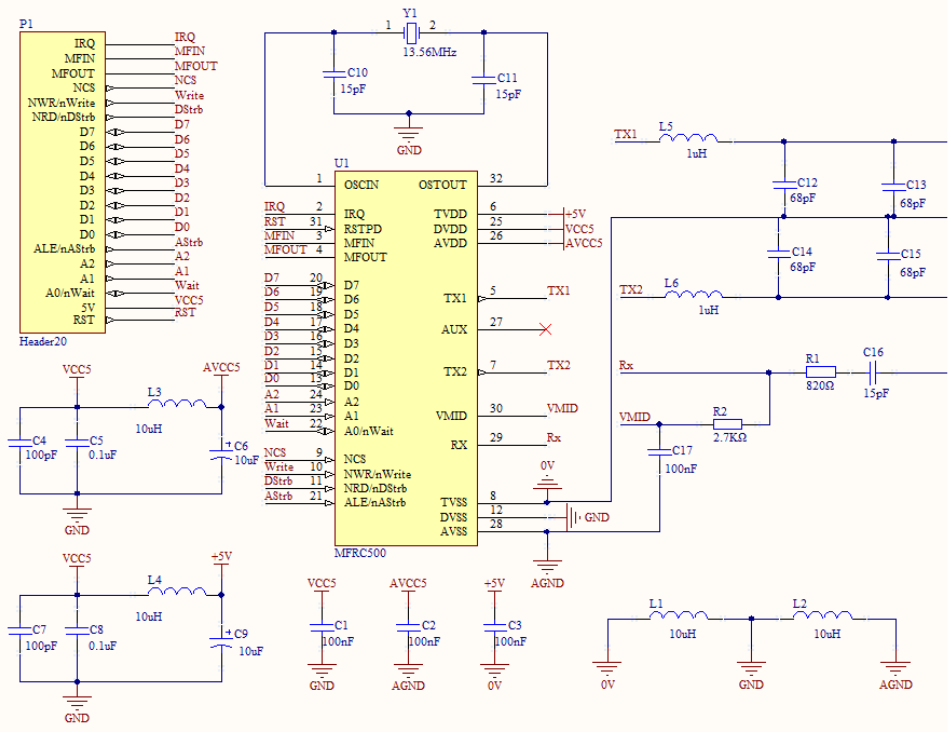


图 3-2：MF RC500 外围电路

射频芯片外围电路设计如上图所示，电路包含了电源、天线接口、复位电路、时钟电路、并行接口几个部分[7]。

① 电源

DVDD+DVSS、AVDD+AVSS、TVDD+TVSS 分别为数字部分、模拟部分和天线激励部分提供电源。1)、数字部分对数字信号进行处理和传输，一般为方波，含有大量的高频谐波信号；2)、模拟部分对从射频标签传输的微小信号进行处理，容易受到数字信号中的高频分量的干扰；3)、天线激励部分将调制后的信号进行放大输出，可能会干扰到模拟部分。

因此，数字电路中，一般要给每个芯片的电源引脚并联一个容量约为 0.01uF-0.1uF 的陶瓷电容到地，越接近芯片越好，一方面是蓄能元件，提供和吸收 IC(芯片模块)开关时的充放电能量，另一方面是旁路高频噪声。与数字电路相同，模拟器件也需要为电源提供高质量的去耦旁路，但同时也需要低频电源旁路，因为模拟器件对噪声的干扰更为敏感。

如何使用电容、电感提高射频模块的 EMC 性能，是本文设计的重点和难点。

在实际项目中，经常采用大电容来滤除低频噪声，小电容来滤除高频噪声。大容量电容，如，铝电解电容和钽电容，呈电感性，电容取值取决于 PCB 板上的瞬态电流要求，一般在 10-470uF；常使用的小电容为 0.1uF 的陶瓷电容，当频率更高时，还可并联更小的电容，例如几 pF，几百 pF 的。多个去耦电容并联，能提供更宽的频谱分布范围，降低电源网络产生的开关噪声。此外，经验表明，电源进来后，一般先过大电容，再过小电容，能达到最佳的滤波效果。

电感也是常用的小元器件。电感是一种蓄能元件，常用在 LC 振荡电路中低频的滤波电路等，其应用频率很少超过 50MHz；电感在电流脉冲期间电源板和模拟电源管脚之间实现了一个相对的高阻抗路径，加强滤波，可以有效降低数字电路部分中的高频谐波。

本文设计中，数字部分、模拟部分和天线激励部分的电源均接入了 0.1uF 的陶瓷电容作为去耦电容，以减少高频噪声的干扰；模拟部分和天线激励部分的电源，则另外接入一个 10uF 的铝电解电容来滤除数字电源中的低频噪声，和一个 10uH 的绕线电感降低其中的高频谐波，保证直流电压的稳定性。

芯片将数字、模拟和天线激励部分的电源分开，DVSS、AVSS、TVSS 连接的数字地和模拟地也最好分开，模拟地和数字地的串接可以采用磁珠、电感(几 uH 到几十 uH)或最好使用 0 欧电阻的方式。本文设计中采用 10uH 的绕线电感进行连接。

② EMC 低通滤波电路

系统的工作频率是 13.56MHz，这个频率要用一个石英振荡器发生，同时也产生了高次谐波，13.56MHz 中的三次、五次和高次谐波要被良好地抑制。13.56MHz 作为载波频率驱动天线，为了满足国际 EMC 标准规定的辐射能量幅度，需要对从 TX1、TX2 输出的信号进行滤波。

L5、C12、C13 和 L6、C14、C15 形成两个 LC 滤波电路。为了获得更好的性能，EMC 滤波器的谐振频率约为 14.41MHz，该频率为从标签接收到的副载波信号的上边带中心频率。目的是增大接收信号的信噪比，改善接收性能，同时还可以减小传输脉冲的过冲幅度，改善传输信号的质量[11]。根据汤姆逊公式以及并联电容容值公式，EMC 低通滤波器的谐振频率为：

$$f_0 = \frac{1}{2\pi\sqrt{L_5(C_{12}+C_{13})}} = \frac{1}{2\pi\sqrt{L_6(C_{14}+C_{15})}} \quad (3.1).$$

官方设计指南推荐参数为：L=2.2uH，C=L_{12}||L_{13}= L_{14}||L_{15}=47pF。本文设计中取电感 L_5 = L_6 = 1uH，代入计算，C ≈122pF。考虑到电容并联能够降低 ESL(等效串联电感)的影响，加强滤波效果，故采用容值均为 68pF 的陶瓷电容并联的方式，也方便元器件购置。

此外亟需注意的是，导线本身存在寄生电容，故 PCB 布局时上述 L 和 C 应尽量靠近 TX1 和 TX2 管脚，C 的接地线尽量短，减少对 TVSS 管脚的阻抗。

③ 接收电路

MF RC500 集成了一个正交解调电路，该电路从输入到 RX 脚的 13.56MHz ASK 调制信号中解析出射频标签副载波负载调制信号的两个边带，不需要增加外部滤波电路。

VMID 管脚是内部的工作参考电压，为保证芯片正常工作，使用内部产生的 VMID 电势作为 RX 管脚的输入电势；VMID 管脚一定要有个 0.1uF 的去耦电容，可减少干扰，提供一个稳定的参考电压；电阻 R1、R2 作为分压电阻，保证 RX 管脚的直流工作点电压与 VMID 电势尽量相同，避免电压值过高造成接收失败。R2=820Ω 为定值，R1 一般范围为 820Ω-2.7kΩ，通过调整电阻 R1 的值，保证 RX 管脚的直流工作点电压。建议在天线线圈和分压电阻之间串联一个电容，容值一般为 15pF。

④ 复位电路

RSTPD 复位引脚为高电平时，MF RC500 内部灌电流关闭，振荡器停止，输入端与外部断开，该管脚的下降沿启动内部复位。

⑤ 时钟电路

MF RC500 的晶体振荡部分包括 OSCIN 和 OSCOUT 引脚，内部集成了 13.56MHz 的石英晶体振荡器，OSCIN 为振荡器反相放大器输入，OSCOUT 为振荡器反相放大

器输出。时钟电路作为同步系统编码器和解码器的时基。无源晶振频率与系统工作频率相同，为 13.56MHz，其负载电容为 15pF。

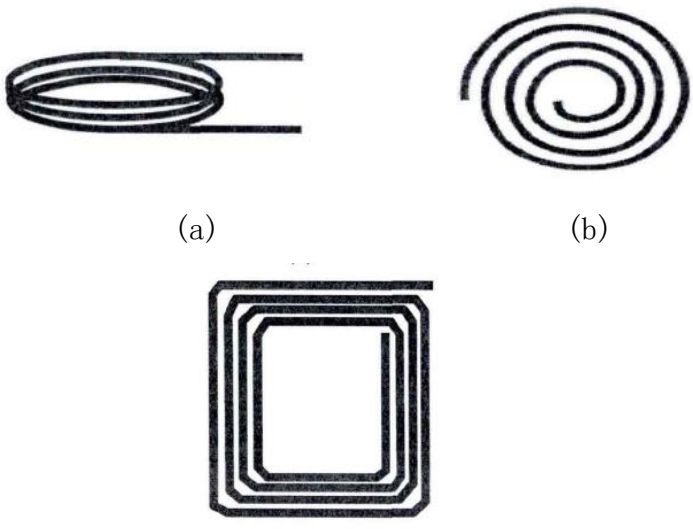
⑥ 并行接口

MF RC500 支持多种 8 位并行微处理器接口模式，当上电或复位后，MF RC500 将根据控制脚的逻辑电平识别微处理器接口。本文设计中采用独立的读/写选通，复用地址总线的模式。片选 NCS 引脚作为微控制器选择和激活 MF RC500 的接口；读/写选通 NRD/NWR 引脚作为选通 MF RC500 寄存器进行数据读写的接口；地址锁存使能 ALE 引脚为高电平时，MF RC500 将地址/数据复用总线 AD0-AD5 锁存为内部地址；其他时候 AD0-AD7 为 8 位双向并口输出/输入；中断事件输出引脚 IRQ 可作为中断源以方便微控制器进行控制。

3.2.3 天线设计

天线是 RFID 系统中能量传递和信息传输的唯一工具，天线设计的好坏将很大程度影响整个系统的性能。天线设计主要包括天线结构的选择以及天线尺寸的设计。

常见的天线线圈结构有短圆柱形(用导线绕制)、圆形和方形(PCB 平面印制)等，本设计中采用 PCB 平面印制的方形天线[12]。



(a) (b)

(c)

图 3-3：天线线圈结构

选定天线结构之后，就需要对天线的尺寸进行设计，即方形天线的边长、匝数、线宽和线间距。

天线周围产生的磁场强度大小 H 与天线线圈上的电流 I、天线线圈匝数 N、天线线圈的半径 R 以及距离天线线圈的垂直距离 x 有关，为：

$$H = \frac{IN R^2}{2\sqrt{(R^2+x^2)^3}} \quad (3.2).$$

假定 I 和 x 不变，当 R=x 时，H 有最大值，因此，读写器的读写距离和天线线圈的半径大致相等。

使用读写器时，通常可以理解为标签紧贴着读写器，即 x=0，此时：

$$H(x=0) = \frac{IN}{2R} \quad (3.3).$$

可以看出，天线线圈的半径越大，该情况下标签所在的磁场强度越小。

此外，需要注意的是，若读写器的读写距离过大，可能会出现多个读写器可以同时操作同一个标签的情况，一般要尽量避免。

综合考虑上述因素，我们采用如下图的 PCB 平面印制矩形线圈，该线圈的等效边长约为 50mm。

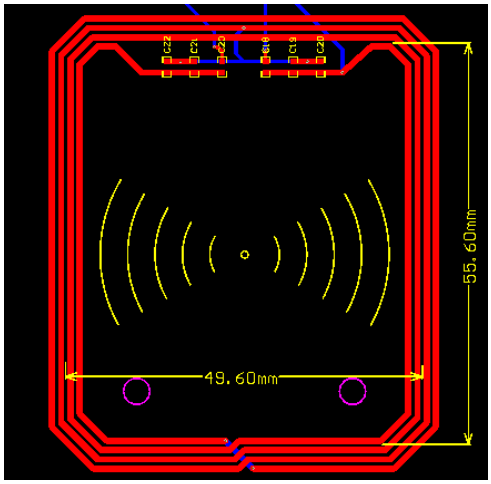


图 3-4：PCB 平面印制矩形线圈

天线线圈的中间有一抽头连接到地端，构成两个单端电感嵌套结构，其电感线圈的两个信号端口(TX1 管脚和 TX2 管脚)的信号等幅反相，即差分电感线圈。两个信号端口的信号成差分形式，这样在嵌套结构中二者的电流方向就是一致的，增加了互感量；此外，差分电感线圈能有效减少天线线圈的寄生电容，获得更高的磁能，改善天线线圈的 EMC 性能，提高天线线圈的能量效率[13]。

3.2.2 天线匹配电路设计

选择好天线的结构，设计完天线的尺寸，确定好天线线圈的匝数、线宽、线

间距，天线本身设计已完成。但要使天线工作在最佳状态，还需要设计与天线相匹配的外围电路，使天线的阻抗与传输线、射频模块的阻抗相匹配，使天线发射和接收能量的效率获得最大值，并最小程度地较少能量在天线上的热耗散。

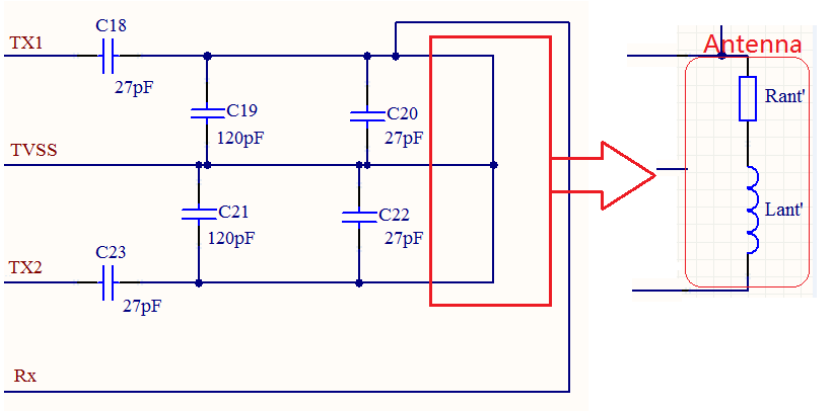


图 3-5：天线匹配电路

① 确定天线线圈等效电路

为了设计上述天线匹配电路中的电容值，需要知道天线线圈的等效电路，如下图。天线线圈的等效电路由等效电阻 R_{ant}、等效电感 L_{ant} 还有等效电容 C_{ant} 组成。

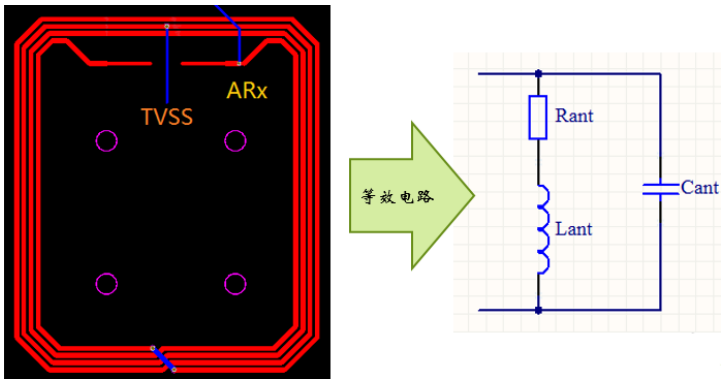


图 3-6：天线线圈等效电路示意图

等效电感 L_{ant} 可以通过下列公式推算：

$$L_{ant}[\mathrm{nH}] = 2l_1[cm]\left(\ln\left(\frac{l1}{D1}\right) - K\right)N_1^{1.8} \quad (3.4).$$

其中各参数含义如下：

l_1 ：一匝导线环的长度；

D_1 ：线圈直径或 PCB 导线的宽度；

K ：天线形状参数，环形天线，K=1.07；方形天线，K=1.47；

N_1 ：线圈的匝数；

ln ：自然对数函数。

本设计中，天线线圈的等效电感：

$$L_{ant}[\mathrm{nH}] = 2 * 20[cm] * \left(\ln\left(\frac{20}{0.1}\right) - 1.47\right) 4^{1.8} \approx 1856.84[nH]$$

线圈的等效电容 C_{ant}≈0.1pF 在计算天线匹配电路中的电容值时可以忽略。

此外，天线匹配电路使得天线工作在 13.56MHz，这个频率下，电阻的集肤效应(skin effect)损耗不可忽略，因此不能只使用 DC 阻抗描述天线线圈，可用下面公式估算：

$$R_{ant}[\Omega] = 5R_{DC} \quad (3.5).$$

本设计中，天线线圈具体设计参数如下：

铜箔厚度(导体厚度) T： 35um；

线 宽(导体宽度) W ： 35mils；

线 长L ： 4*(4*5) =80cm；

利用 PCB 走线电阻在线计算，通过方块统计的方法，利用铜的电阻与重量的关系，计算出天线线圈的直流电阻

$$R_{DC} = 0.44\Omega,$$

故等效电阻

$$R_{ant}[\Omega] = 5R_{DC} = 5 * 0.44 = 2.2[\Omega]。$$

在已知天线线圈的等效电路的情况下，我们可以通过计算来确定天线匹配电路中电容的取值，使匹配电路与天线线圈组成 LC 谐振电路，通过此谐振电路，以实现天线与 MF RC500 芯片达到良好阻抗匹配，使得天线的能量利用效率最高，提升读写器性能。计算天线匹配电路中电容的取值时，需要考虑以下三个方面：

1) 谐振频率。

保证匹配电路与天线线圈组成的 LC 谐振电路的谐振频率为传输载波频率的 13.56MHz。

2) 品质因子。

品质因子 Q 值用来衡量读写器和标签的性能，定义为谐振回路中总的存储能量与每周期中消耗在电阻(天线线圈本身具有内阻，加上连接电阻和其他的器件)上的能量之比(或无功功率的绝对值与有功功率之比，是用来体现电容、电感等其他储能元器件或电路存储能力的一种质量指标)。

$$Q = \frac{\omega_R \cdot L_{ant}}{R_{total}} = \frac{2\pi f \cdot L_{ant}}{R_{total}} \quad (3.6).$$

Q 值越高，说明天线线圈中的电流强度大些，天线线圈存储的能量大，改善读写器对标签的能量传输[4]。

3）阻抗匹配。

有以下两种方式设计天线匹配电路，一种是采用 50Ω匹配天线，用于读写器与天线距离较远(10m 左右)，通常需要通过同轴电缆连接的情况；一种是直接匹配天线，用于读写器和天线距离较近(<30mm)的小型设备，其阻抗匹配的典型值是 700Ω。

② 改善天线的品质因子

品质因子 Q 值不是越大越好，天线的传输带宽 B 与 Q 值成反比，Q 值过高将减少天线带宽，可能导致标签无法解调读写器发送的信号。在 MIFARE®卡应用中，数据从读写器传输到标签使用脉宽 T=3us 的 Miller 编码，由时间和带宽的乘积规定，

$$B \cdot T \geq 1 \quad (3.7).$$

又带宽和品质因子的关系，

$$B = \frac{f_R}{Q} \quad (3.8).$$

可得出本文设计中品质因子的取值范围：

$$Q \leq f_R \cdot T = 13.56MHz * 3us = 40.68$$

考虑到元件的容差和对温度的依赖，本文设计中采用官方推荐的典型值 Q=35。天线线圈的 Q 值通常都在 50-100 之间，要使信息能够正常传输，可以在天线线圈两端接入外部电阻 R_{ext} 的方式降低原始的品质因子。

因为本设计中的天线线圈采用中间抽头接地的嵌套结构，所以外部电阻 R_{ext} 被分为阻值相等的两部分，分别接入在 TX1、TX2 管脚的匹配电路中。

外部电阻 R_{ext} 可由以下公式估算：

$$R_{ext} = \frac{\omega_R \cdot L_{ant}}{Q} - R_{ant} = \frac{2\pi f_R \cdot L_{ant}}{Q} - R_{ant} \quad (3.9).$$

将天线线圈等效电路各参数代入，

$$R_{ext/2} = (\frac{2\pi \cdot 13.56M\,[\mathrm{Hz}] \cdot 1857[nH]}{35} - 2.2[\Omega])/2 = 1.1602[\Omega],$$

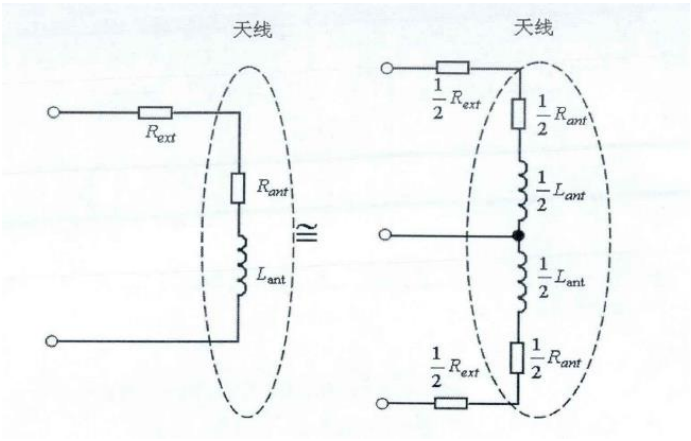


图 3-7：接有外部电阻的天线线圈等效电路

最终根据经验和实际设计情况，本设计中的天线线圈无需引入外部电阻即能满足信息正常传输的 Q 值要求。

③ 计算匹配电路中电容的取值

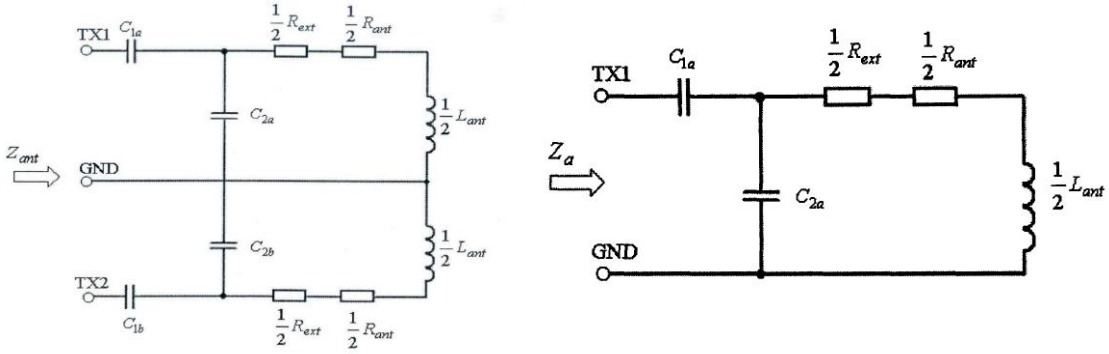


图 3-8：直接匹配天线电路简化(深色为简化后的电路)

本设计中天线线圈采用中间抽头接地的嵌套结构，由电路的对称性，能够得到用于分析的简化匹配电路。对该 LC 谐振电路进行分析，根据阻抗匹配经过一系列较为复杂的推导可以得到估算匹配电容容值的公式：

并联匹配电容容值：

$$C_{2a} = \frac{1}{\omega\sqrt{\left(\frac{\omega \cdot L_{ant}}{1-\frac{R}{Z_a}}\right)^2 - \frac{R^2+\omega^2 \cdot {L_{ant}}^2}{1-\frac{R}{Z_a}} + \frac{\omega^2 \cdot L_{ant}}{1-\frac{R}{Z_a}}}} \quad (3.10).$$

串联匹配电容容值：

$$C_{1a} = \frac{R^2 + (\omega \cdot L_{ant} - \frac{1}{\omega \cdot C_{2a}})^2}{\frac{\omega \cdot L_{ant}}{C_{2a}}\left(\frac{1}{\omega \cdot C_{2a}} - \omega \cdot L_{ant}\right) - \frac{R^2}{{C_{2a}}^2}} \quad (3.11).$$

其中，

$$\omega = 2\pi f_R = 2\pi * 13.56M[Hz] \quad (3.12).$$

输入阻抗 Z_{ant} 的典型值为 700Ω，Z_a=350Ω。

从并联电容容值推出串联电容过程中需要考虑天线匹配电路的谐振频率。再次使用汤姆逊公式，谐振频率：

$$f_R = \frac{1}{2\pi\sqrt{L \cdot C}} = 13.56\text{M[Hz]} \quad (3.13).$$

L 和 C 的取值需要参考天线线圈的等效电路。天线线圈等效电感取值范围 0.5-3uH，电容匹配较容易实现，一般天线线圈的等效电感设计在 1-2uH 之间，如本设计中就为 1857nH。

将上述各参数代入，使用 MATLAB 进行计算，得到：

并联匹配电容容值：$C_{2a} = C_{2b} = 137.1066pF.$

串联匹配电容容值：$C_{1a} = C_{1b} = 12.7504pF.$

参考 NXP 提供的官方计算 Excel 表 *NXP Semiconductors. Directly matched Antenna Excel calculation.xls*，本设计中采用的串联电容 C_{1x}=C_{18}=C_{23}=27pF，并联电容 C_{2x}=C_{19}||C_{20}=C_{21}||C_{22}，即 120pF 和 27pF 的电容并联作为并联匹配电容部分，以实现更好的谐振效果。

3.3 系统 PCB 设计

本文设计 RFID 读卡模块在 PCB 布局布线过程中遵循以下要求：相互间可能存在干扰的器件不能距离过近；PCB 上的模拟部分和数字部分应分开，对于数字电路，应采用多点就近接地的方式或者对地敷铜的方式以减少引线电感；地线应较粗，以减少其上的电平变化。在每个芯片的电源输入和地之间均应连接一个去耦电容。微控制器的无用引脚应接地或高电平或通过软件设置其为输出端。各器件的时钟产生端应距离器件足够近，通过地线敷铜将时钟引线包围起来，时钟引线应尽量垂直器件 I/O 线。在进行 I/O 线的布线时，线长不能过长，且应注意近距离平行走线所带来的干扰[14]。

对于平面印制 PCB 天线线圈，采用中心抽头接地的接线方式可以获得更好的 EMC 效果，此时左右两部分线圈的面积应尽量相等。天线匹配电容应尽量靠近天线且接口接线尽量粗短，以避免产生寄生电阻。

3.4 本章小结

本章阐述了本文设计的 RFID 读写器的硬件设计部分，主要是射频读卡模块的原理图和 PCB 设计，包括射频芯片 MFRC500 外围电路、天线设计、天线匹配电路等几部分。此外，在第 2 章系统理论的基础上，详细介绍了天线匹配电路设计中重要元器件各参数的由来。在设计阶段应尽可能多的弄清楚参数的来龙去脉，从理论依据出发进行设计，才能提高硬件电路的可行性和可靠性。

第 4 章 系统软件设计

在完成系统硬件设计与搭建后，我们需要协同进行软件的设计与开发，包括底层板 STM32 下位机程序、应用程序 API 以及用于测试的上位机软件。其中，下位机软件应实现适用于 M1 卡的 ISO14443-A 协议函数。具体的软件设计架构将在本章开头做概要性阐述并在后续内容中做详细说明。

4.1 分层原理下的软件设计

参考 OSI 模型分层原理设计的思想，对整个系统软件数据传输进行概况，可抽象出物理层、数据链路层、应用层这样的两个三层结构，如下图所示[15]。

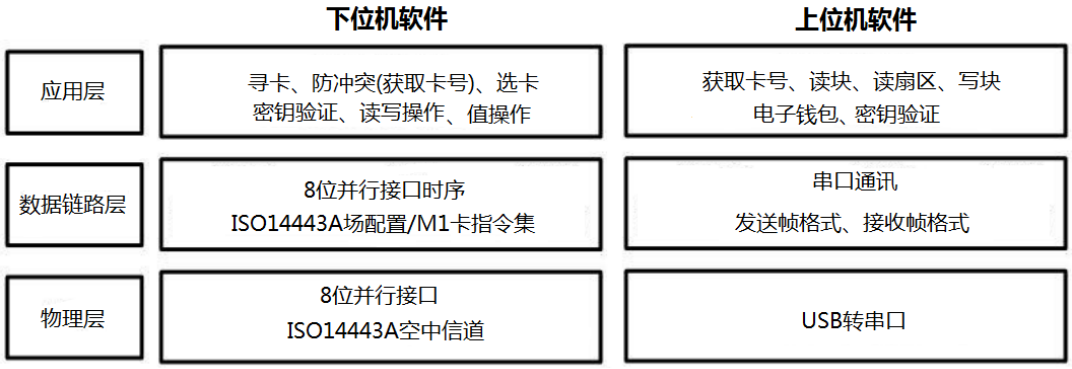


图 4-1：分层原理下的软件设计

物理层的实现在双方通信过程中起着"桥梁"的作用，在下位机软件中担此重任的角色正是 MFRC500 射频芯片。射频芯片通过天线向标签传输能量和传递信息，也通过天线接收从标签返回的数据。在程序初始化时，配置 MFRC500 的相关寄存器，使芯片工作在标签能够兼容的 ISO14443-Type A 模式。另外 MF RC500 发送信息时必须按照标签要求的格式进行发送，即 M1 卡命令及其对应的参数，根据 M1 卡操作流程控制标签；同时，按照 M1 卡命令的返回值格式完成数据接

收。此外，MFRC500 不支持独立编程，需由 STM32 单片机控制，二者间通过简单的 8 位并口通讯，通讯过程须严格按照并行接口时序进行。

在上位机软件中，则通过串口实现 PC 端到 STM32 单片机的数据传输。在程序初始化时也需要对其进行配置，包括波特率、校验位等。应用程序操作标签，需要按照串口通讯发送帧的格式发送数据给 STM32 单片机，单片机从串口接收到数据，按照格式对数据进行解析，获取其中的命令和参数，然后通过射频芯片这一媒介操作标签，获取到所需的数据并最终按照串口通信接收帧的格式返回给 PC 端， PC 端按照对应的格式对返回的数据进行解析。应用程序提供丰富的方式将数据呈现给用户或利用数据提供具体服务。

4.2 下位机软件设计

下位机软件按照任务调度的方式进行编程，对涉及到的各个模块进行模块化编程，各个模块的变量及子程序各自独立，模块与模块之间可以相互调用。

① 编写并行接口驱动

MF RC500 和 M1 卡都有各自的指令集(分别为 Command 指令和 M1 指令)，STM32 单片机通过 Command 指令控制 MFRC500，来完成 M1 指令从而实现读写卡的目的。如何通过对 MFRC500 的控制达到读写卡的目的是整个下位机软件设计的主要内容。STM32 单片机对 MFRC500 的控制有三种方式：

1) 执行命令来初始化函数和控制数据操作，即将 Command 指令写入 MFRC500 的命令寄存器，通过 MFRC500 的 FIFO 缓冲区来传递参数和交换数据；

2) 通过设置配置位来设置电气和函数的行为，即设置 MF RC500 的寄存器的相应位；

3) 读状态标志位监控 MFRC500 的状态，即通过读 MFRC500 的寄存器来监控

MFRC500 的状态。

本质上，这三种方式都是通过 MFRC500 的寄存器来实现的。对 MFRC500 寄存器的读写操作，通过 8 位并行接口实现。

MFRC500 有多种并行接口类型，本设计中采用独立的读写选通/复用地址总线的连接配置，连接方式如下图所示。

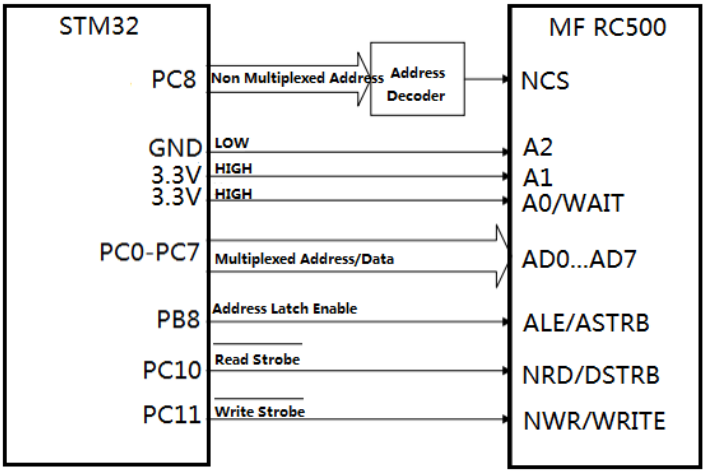
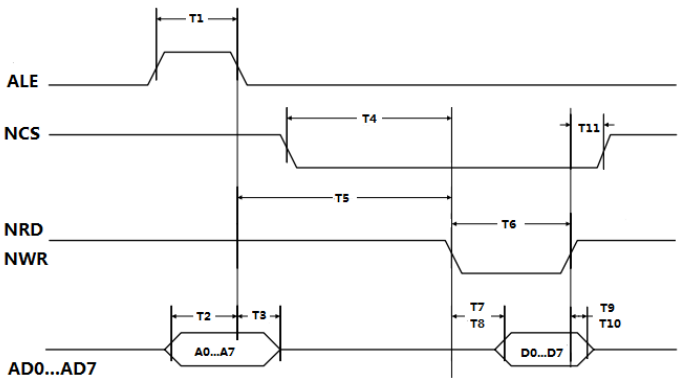


图 4-2：STM32 连接 MFRC500

按照上图的连接方式进行连接后，需要对相应的 IO 口进行配置，主要涉及 IO 口时钟配置以及 IO 口设置(输入输出模式等)，配置函数如下：

- *void MFRC500_GPIO_Config(void);*

此外并行接口驱动必须满足并行接口时序图中定义的时序。



符号	内容	参数(单位: ns)
T1	ALE 脉宽	≥20
T2	地址建立时间	≥15
T3	地址保持时间	≥8
T4	片选信号有效到读写信号有效	≥0
T5	地址锁存到读写有效	≥15
T6	读写有效信号脉宽	≥65
T7	写信号有效到数据有效	≤35
T8	读信号有效到读数据有效	≤65
T9	写数据保持时间	≥8
T10	读数据有效时间	≤20
T11	读写信号无效到片选失效	≥0

图 4-3：8 位并行接口时序图

最终的并行接口驱动函数如下：

- unsigned char ReadRawRC(unsigned char Address);

+ 函数说明：读取 MF RC500 指定寄存器的值
+ 参数说明：读取的寄存器地址(0x00-0x3F)
+ 返 回 值：读取的寄存器的值

- void WriteRawRC(unsigned char Address, unsigned char value);

+ 函数说明：向 MF RC500 指定寄存器中写值
+ 参数说明：寄存器地址(0x00-0x3F)；写入的值
+ 返 回 值：无

② 编写 PCD/PICC 通讯函数

M1 卡(PICC)与读写器(PCD)之间的典型通讯过程是：STM32 单片机将待发送数据(包括标准 M1 卡指令)按一定格式写入 FIFOData 寄存器后，再写 Command 指令到 Command 寄存器，以此触发通信的开始(有些指令还需对一些寄存器操作后才开始通信)。卡若接收到数据，先读取数据中的 M1 指令，再将余下数据作为 M1 指令的操作对象进行 M1 指令指示的操作，并将结果返回 FIFOData 寄存器。卡内部的操作是自发的，将 Command 指令写入 Command 寄存器中，一段时间后通过读取 FIFOData 寄存器中卡的返回值来确定 M1 卡是否成功地完成了制定操作。Command 指令在 MF RC500 内识别，标准 M1 指令在其中相当于一些普通数据，它的识别过程在 M1 卡中进行。

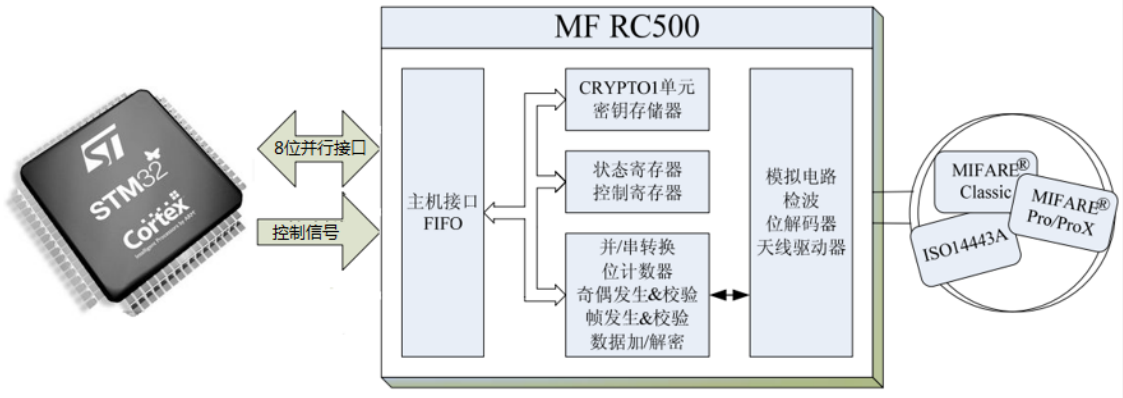


图 4-4：M1 卡与读写器通讯方框图

最终的 PCD 指令处理函数如下，主要是控制 PCD 与 PICC 进行通讯：

- *signed char PcdCmdProcess(struct TranSciveBuffer *pi);*

+ 函数说明：控制 MF RC500 处理相关指令，与 M1 卡进行通讯

+ 参数说明：用于传递操作指令和数据以及返回操作结果和数据的结构体指针

```
struct TranSciveBuffer{
    unsigned char MfCommand;
                    // MF RC500 命令字(与 M1 卡通讯时为 TRANSCEIVE 指令)
    unsigned int  MfLength;
                    // 发送数据长度或接收数据长度
    unsigned char MfData[128];
                    // 发送数据或接收数据临时缓冲区
    unsigned int mfcurrent;
                    // 当前数据在缓冲区中的下标
};
```

+ 返 回 值：执行状态

```
#define MI_OK                          0
#define MI_NOTAGERR                   (-1)
#define MI_CRCERR                      (-2)
#define MI_EMPTY                       (-3)
#define MI_AUTHERR                     (-4)
…
```

③ 编写 ISO14443-A 协议函数

在了解了 M1 卡与读写器之间的通讯过程后，我们还需要了解读写器操作 M1 卡的流程，只有符合这个流程，才能访问到卡并最终完成相应的卡操作，获取所需的数据。

1) 复位初始化 RC500 之后，首先要进行寻卡。寻卡主要用于搜寻天线射频场一定范围内是否存在 M1 卡。当 Mifare 卡处在读写器的天线工作范围之内时，程序控制射频芯片向卡片发出 REQUEST ALL(或 REQUEST IDLE)命令,此时卡片的 ATR 将启动,并将卡片 Block0 中的卡片类型(TagType)，共两个字节数据传送到读写器，从而建立卡片与读写器的第一步通信联络。具体的寻卡函数如下：

```
signed char PiccRequest(unsigned char req_code, unsigned char *pTagType);
```

+ 函数参数：*req_code* 指示 Request All 还是 Request Idle，
pTagType 返回 M1 卡类型

+ 返 回 值：寻卡状态

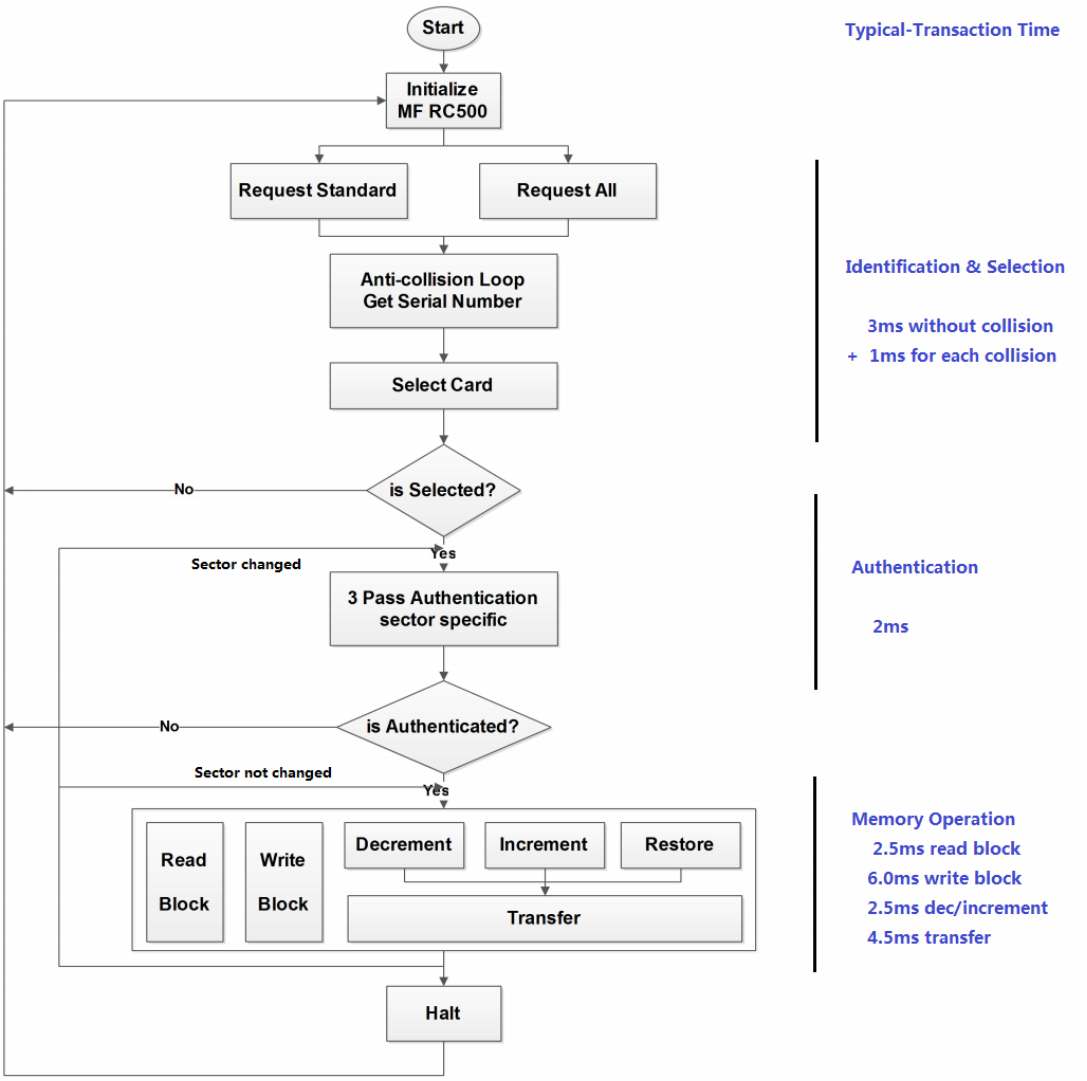


图 4-5：读写器操作 M1 卡流程图

2) 寻卡成功便可以进行防冲突检测。防冲突是整个流程中较为复杂的环节，主要功能是在若干张 M1 卡中按一定的算法(二进制检索树)获取其中 1 张 M1 卡的序列号。M1 卡防冲突命令[SEL NVB]中 SEL 为 0x93，NVB 为 1byte 数据，高 4 位表示本次待发送数据的有效字节数，低 4 位为发送数据最后一个字节的有效位数。成功接收到读写器发送的防冲突指令后，有效范围内的所有卡均以其序列号响应。一旦发生冲突，则应读取相应寄存器(CollPos，0x0B)的值确定冲突位，之后通过不断更新 NVB 的值、接收到的有效位数与接收到的有效数据来更新防冲突指令传送的数据，直至再

无冲突发生。具体的防冲突循环函数如下：

signed char PiccAnticollisionLoop(unsigned char *pSnr, unsigned char SelType);

+ 函数参数：通过防冲突循环获取的由 *SelType* 指定的当前 *SelType* 级别的序列号

+ 返 回 值：防冲突循环执行状态

3) 选卡。利用防冲突返回的卡号进行选卡，发送选卡命令，执行该操作后，将返回卡上的 SIZE 字节。

signed char PiccSelect(unsigned char *pSnr,
unsigned char *pSAK, unsigned char SelType);

+ 函数参数：当前级别 *SelType*，防冲突循环得到的序列号 *pSnr*，
进行选卡后返回的 *pSAK* 信号

+ 返 回 值：选卡操作执行状态

4) 级联防冲突流程包括上述的防冲突循环和选卡两部分，具体函数如下：

signed char PiccCascAnticollision(unsigned char *pUid, unsigned char *pLen);

+ 函数说明：调用防冲突循环函数和选卡函数完成级联防冲突流程

+ 函数参数：通过级联防冲突得到的卡号及其对应长度(4、7 或 10 个字节)

+ 返 回 值：级联防冲突执行状态

5) 密钥验证。密钥验证以扇区为单位，其结果在于开启通信加密单元，成功后的数据传输都将加密，且只有密钥验证之后才可以对卡进行读、写、增值、减值操作。具体的密钥验证函数如下：

signed char PcdAuthState(unsigned char auth_mode,
unsigned char blockNum, unsigned char *pSnr);

+ 函数参数：*auth_mode* 验证方式(验证 A 密钥或者 B 密钥)，
blockNum 要验证的绝对块号(0-63)，
pSnr 通过级联防冲突得到的卡的序列号

+ 返 回 值：密钥验证执行状态

6) 块读写操作及值操作。读块和写块是对 M1 卡的基本操作，值操作主要包括增值、减值操作，在电子钱包中有大量应用。具体函数如下：

signed char PiccRead(unsigned char addr, unsigned char *pReaddata);

+ 函数参数：*addr* 要读块的绝对块号，*pReaddata* 从该块中读出的 16 字节数据

+ 返 回 值：读块操作执行状态

signed char PiccWrite(unsigned char addr, unsigned char *pWritedata);

+ 函数参数：*addr* 要写入块的绝对块号，*pWritedata* 16 字节的写入数据

+ 返 回 值：写块操作执行状态

signed char PiccValues(unsigned char dd_mode,
unsigned char addr, unsigned char *pValue);

+ 函数参数：*dd_mode* 操作方式(增值还是减值)，*addr* 要操作块的绝对块号(0-63)，*pValue* 带符号 4 字节操作值，低字节在前

+ 返 回 值：值操作执行状态

7) 结束。卡操作完毕后，可使卡进入 Halt 状态，此时只有 REQUEAT ALL 指令可唤醒该卡。

signed char PiccHalt();

+ 函数参数：无

+ 返 回 值：执行状态

④ 编写应用层函数

应用层函数利用并行接口驱动函数、PCD/PICC 通讯函数以及 ISO1443A 协议函数，封装了本设计所有功能需要的应用函数。应用函数相互独立，均以模块的形式提供，方便下位机主程序随时调用。应用层函数按照功能划分为以下几个模块：获取卡号、密钥验证、读块、写块、电子钱包初始化、查询余额、充值、扣款。具体函数如下：

signed char ComM1RequestA(unsigned char* TypeA_Uid, unsigned char* len);

+ 函数说明：寻 Mifare_One 卡，完成冲突检查，获取其 UID

+ 函数参数：*TypeA_Uid* 获取的卡号及其对应长度 *len* (4、7 或 10)

+ 返 回 值：执行状态

signed char ComM1Authentication(unsigned char SectorNum,
unsigned char KeyAB, unsigned char *pKeys);

+ 函数说明：Mifare_One 卡扇区密钥验证，完成寻卡、选卡、3-Pass 验证

+ 函数参数：验证扇区号 *SectorNum*(0-15)，
密钥类型 *KeyAB* (A 或 B 密钥)以及 6 字节密钥 *pKeys*

+ 返 回 值：执行状态

signed char ComM1BlockRead(unsigned char SectorNum, unsigned char BlockNum,
unsigned char KeyAB, unsigned char *pKeys, unsigned char *pReadData);

+ 函数说明：读取 Mifare_One 卡中指定块数据，完成寻卡、选卡、3-Pass 验证及后续的读块操作

+ 函数参数：扇区号 *SectorNum*、块号 *BlockNum*(0-3)，

密钥类型 *KeyAB*、密钥 *pKeys* 以及读取到的 16 字节块数据 *pReadData*

+ 返 回 值：执行状态

signed char ComM1BlockWrite(unsigned char SectorNum, unsigned char BlockNum, unsigned char KeyAB, unsigned char *pKeys, unsigned char *pWriteData);

+ 函数说明：Mifare_One 卡中指定块写入数据，完成寻卡、选卡、3-Pass 验证及后续的写块操作

+ 函数参数：扇区号 *SectorNum*、块号 *BlockNum*，

密钥类型 *KeyAB*、密钥 *pKeys* 以及写入的 16 字节块数据 *pWriteData*

+ 返 回 值：执行状态

signed char ComM1WalletInit(unsigned char SectorNum, unsigned char BlockNum, unsigned char KeyAB, unsigned char *pKeys, unsigned char *pInitVal);

+ 函数说明：电子钱包初始化，完成寻卡、选卡、3-Pass 验证及后续的写块操作，将该块初始化为用于电子钱包的特殊数据块

+ 函数参数：扇区号 *SectorNum*、块号 *BlockNum*，

密钥类型 *KeyAB*、密钥 *pKeys* 以及带符号的 4 字节初始化金额(低字节在前)

+ 返 回 值：执行状态

signed char ComM1WalletCheck(unsigned char SectorNum, unsigned char BlockNum, unsigned char KeyAB, unsigned char *pKeys, unsigned char *pCheckVal);

+ 函数说明：电子钱包余额查询，完成寻卡、选卡、3-Pass 验证及后续的读块操作，获取保存在电子钱包特殊数据块中的金额

+ 函数参数：扇区号 *SectorNum*，块号 *BlockNum*，

密钥类型 *KeyAB*、密钥 *pKeys* 以及查询到的余额

+ 返 回 值：执行状态

signed char ComM1WalletRecharge(unsigned char SectorNum, unsigned char BlockNum, unsigned char KeyAB, unsigned char *pKeys, unsigned char *pIncreVal);

+ 函数说明：电子钱包充值，完成寻卡、选卡、3-Pass 验证及后续的值操作(增值)

+ 函数参数：扇区号 *SectorNum*、块号 *BlockNum*，

密钥类型 *KeyAB*、密钥 *pKeys* 以及充值金额

+ 返 回 值：执行状态

signed char ComM1WalletDebit(unsigned char SectorNum, unsigned char BlockNum, unsigned char KeyAB, unsigned char *pKeys, unsigned char *pDecreVal);

+ 函数说明：电子钱包扣款，完成寻卡、选卡、3-Pass 验证及后续的值操作(减值)

+ 函数参数：扇区号 *SectorNum*、块号 *BlockNum*，

密钥类型 *KeyAB*、密钥 *pKeys* 以及消费金额

+ 返 回 值：执行状态

⑤ 编写串口处理函数

串口处理函数是整个下位机软件的主程序，用以接收 PC 端上位机软件通过串口传输的命令，调度相应的应用模块进行处理，同时将处理结果以串口的方式回送给上位机软件。

命令的接收通过中断实现。串口接收中断响应函数将数据存储在主程序缓冲区中，并告知主程序接收完成。

主程序通过轮询的方式运转，在命令接收完成后，对其进行解析，调用相应的应用模块进行处理，并将结果直接以串口的方式回送。

命令的接收及结果的回送需遵循串口通信协议进行。

4.3 串口通信协议

PC 端通过 USB 转串口的方式与 STM32 单片机通讯，主要传输标签操作控制指令和标签数据等信息。为确保传输的有效性和稳定性，二者除遵循统一的串口设置外，还必须对传输数据设定统一的格式打包发送和接收[16]。

4.3.1 发送格式(PC 端➔STM32 单片机)

	起始位	命令字节	数据长度	数据字节	BCC 校验位	终止位
格式说明	0xBC	命令代码	N	命令参数	包校验	0xFE
字节数	1	1	1	N	1	1

表 4-1：串口通信协议发送格式

发送格式说明：

命令字节：命令代码具体数值及含义见下表：

命令字	含义	参数(字节数)	返回数据(字节数)
0x01	获取卡号	无	卡号(4)
0x03	密钥验证	扇区号(1)、密钥类型(1)、密钥(6)	密钥类型(1)
0x05	读块操作	扇区号(1)、块号(1)、密钥类型(1)、密钥(6)	块数据(16)
0x07	写块操作	扇区号(1)、块号(1)、密钥类型(1)、密钥(1)、写入数据(16)	无
0x09	钱包初始化	扇区号(1)、块号(1)、密钥类型(1)、密钥(1)、初始金额(4)	余额(4)

0x0B	余额查询	扇区号(1)、块号(1)、密钥类型(1)、密钥(1)	余额(4)
0x0D	钱包充值	扇区号(1)、块号(1)、密钥类型(1)、密钥(1)、充值金额(4)	余额(4)
0x0F	钱包扣款	扇区号(1)、块号(1)、密钥类型(1)、密钥(1)、消费金额(4)	余额(4)

表 4-2：串口通信协议命令集

数据长度：带参数的命令发送的参数数据长度 n；

数据字节：n 个字节的参数数据，多参数的情况按照参数顺序要求依次发送；

BCC 校验：从命令字节到最后一个数据字节按位异或的结果。

4.3.2 接收格式

	起始位	状态字节	数据长度	数据字节	BCC 校验位	终止位
格式说明	固定值	命令操作状态	n	命令参数	包校验	固定值
字节数	0xBC	1	1	N	1	0xFE

表 4-3：串口通信协议接收格式

接收格式说明：

状态字节：0x0 表示操作成功；负值表示操作失败，常见的状态及含义如下表：

数值	含义	数值	含义	数值	含义
0	执行成功	-4	3-pass 验证失败	-24	防冲突循环失败
-1	寻卡失败	-5	通信奇偶校验失败	-125	MF RC500 通讯超时
-2	通信 CRC 校验失败	-11	返回数据位不够		

表 4-4：串口通信协议常见状态

4.4 应用程序 API

应用程序 API 将上述协议封装成动态链接库，提供一系列 PC 端通过串口控制 STM32 单片机完成卡操作的 API，实现了卡操作、串口等细节对上位机应用程序的屏蔽，方便了上位机应用程序的快速二次开发。

基于 Windows API 实现对 PC 端串口的控制，主要流程如下：

① 使用 Windows API 中的 CreateFile 函数创建串口句柄，指定端口号；

② 使用 Get/SetCommState 对串口进行配置，其中比较重要的参数是波特率

(BaudRate)、数据位(DataBits)、奇偶校验(Parity)、停止位(StopBits)；

③ 使用 ReadFile 或者 WriteFile 函数对串口进行相应的读写操作；

④ 读写结束后，要使用 CloseFile 函数关闭串口句柄。

与下位机的串口处理函数类似，这里的串口发送操作也是以串行的形式实现，不过串口的接收操作，则是通过多线程实现。当串口句柄被创建时，接收线程也被启动，并以轮询的方式查询串口是否有数据接收，并在接收完成后以事件消息的方式通知 API 函数。API 函数按照协议通过串口发送命令后便等待处理结果的返回，直到该接收事件消息发生，才进行下一步返回数据的处理。

在控制串口的基础上，遵循上述协议，面向本设计功能需求，最终提供如下的 API：

```
// 串口打开&关闭
RC500_API signed char OpenPort(int portNum);
RC500_API signed char ClosePort(void);
// 获取卡号
RC500_API signed char Card_GetUID(unsigned char *pUID, unsigned char &len);
// 密钥验证
RC500_API signed char Card_Authentication(unsigned char sectorNum,
                          unsigned char keyType, const unsigned char *pKey);
//读块操作
RC500_API signed char Card_Read(unsigned char sectorNum,
                          unsigned char blockNum, unsigned char keyType,
                          const unsigned char *pKey, unsigned char *prdata);
// 写块操作
RC500_API signed char Card_Write(unsigned char sectorNum,
                     unsigned char blockNum, unsigned char keyType,
                     const unsigned char *pKey, const unsigned char *pwdata);
// 电子钱包初始化
RC500_API signed char Card_WalletInit(unsigned char sectorNum,
                     unsigned char blockNum, unsigned char keyType,
                     const unsigned char *pKey, signed long initVal);
// 电子钱包余额查询
```

```
RC500_API signed char Card_WalletCheck(unsigned char sectorNum,
                    unsigned char blockNum, unsigned char keyType,
                    const unsigned char *pKey, signed long &remVal);
// 电子钱包充值操作
RC500_API signed char Card_WalletRecharge(unsigned char sectorNum,
                    unsigned char blockNum, unsigned char keyType,
                    const unsigned char *pKey, signed long &value);
// 电子钱包消费操作
RC500_API signed char Card_WalletDebit(unsigned char sectorNum,
                    unsigned char blockNum, unsigned char keyType,
                    const unsigned char *pKey, signed long &value);
// 打印错误状态信息
RC500_API char* PrintErrInfos(signed char errCode);
```

4.5 上位机应用程序设计

上位机应用程序基本功能有获取卡号、密钥验证、块读写操作、电子钱包等。本节将简述如何利用前面实现的串口通信中间件进行二次开发，实现所需功能。

4.5.1 开发工具 VC++6.0

上位机应用程序的开发及串口通信中间件的开发均在 Microsoft©的 VC++6.0 上进行，该 IDE 支持 Win32 DLL 开发，并提供便捷的 MFC AppWizard[exe]开发，满足本文设计需求。

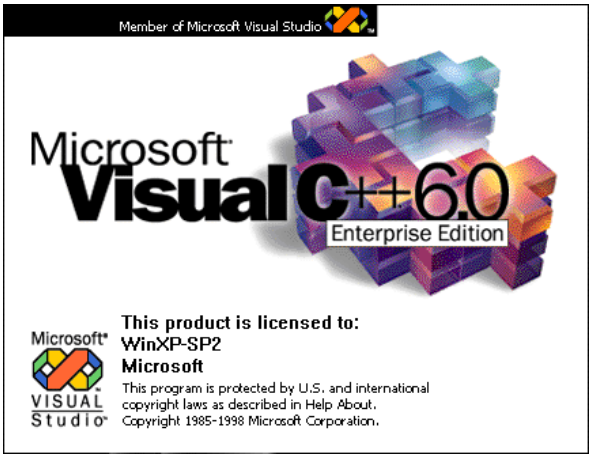


图 4-6：Visual C++6.0

4.5.2 使用应用程序 API

应用程序 API 除了提供一个 DLL 文件外，还包含进行二次开发所需的引入库文件(.lib)和头文件(.h)。我们只需要将这两个文件导入工程：①通过#include 预编译头的方式包含头文件，②在 Project Setting 中将该引入库文件作为项目的 Input 链接选项，这样就可以在项目中进行调用，并完成构建。最后，将构建生成的项目可执行程序与 DLL 放在相同目录下即可。

4.5.3 基于 MFC 的上位机软件的展示与分析

图 4-7：MFRC500_DEMO

上位机软件的实现按照以下流程进行：

① 界面设计，包括串口、开启卡片、读写块操作、密钥设置、电子钱包功能的布局以及各个功能需要呈现的信息，用到的控件包括下拉列表、按钮等。

② 围绕各按钮具体意义及其响应函数，调用相应的 API。从下拉列表、编辑框获取用户输入的 API 所需参数，API 的返回值则通过编辑框和弹窗等呈现给用户。

4.6 本章小结

本章详细介绍了基于 MFRC500 和 STM32 的读写器的软件设计，按照分层原

理，从最底层的并行接口驱动程序、下位机软件到基于串口的应用程序 API，以及用于调试的上位机软件。主要围绕软件设计架构以及各层软件设计中的关键技术进行阐述。并行接口驱动部分定义了 STM32 与射频读卡模块的连接方式并对并行接口的时序做详细说明。下位机软件则着重讲述了基于 ISO/IEC 14443A 协议的 M1 卡密钥验证、读写块等操作函数的具体实现和封装。最后，围绕 PC 端与 STM32 的串口通信，定义了一套串口通信协议，并基于 Windows API 实现了基于串口的应用程序 API。

第 5 章 系统功能测试

在基于STM32与MFRC500 RFID读写器软硬件设计的基础上,实现了MFRC500射频读写模块，实现了 STM32 下位机软件程序，封装了应用程序 API，并利用该 API 进行二次开发，开发了用于系统测试的上位机软件。本章内容讲述如何利用我们实现的系统测试上位机软件，对本文设计实现的软硬件进行测试及分析，确保我们的系统能够正常完成 M1 卡的所有操作。

5.1 测试准备工作

将模块与 STM32 开发板通过杜邦线连接，开发板与 PC 则通过 USB 线连接，连接的方式是串口，所以需要在 PC 上安装 USB 转串口驱动。

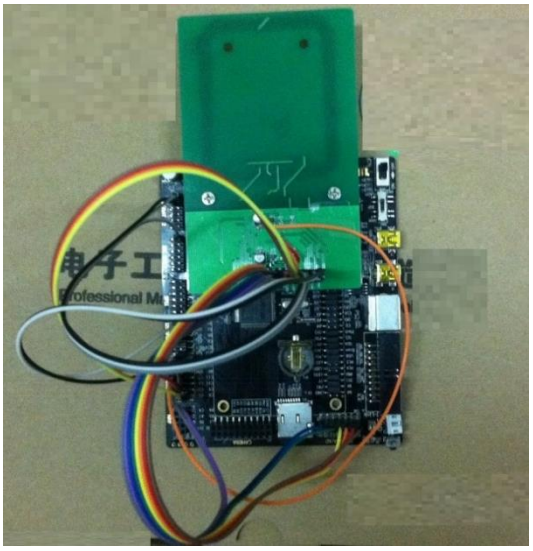


图 5-1：模块-开发板连接

5.2 系统测试

硬件设备连接后，即可拨开开发板开关，运行上位机软件进行测试。

5.2.1 串口

设置好波特率等信息后就可以点击"打开串口"按钮，之后会有弹窗提示打开成功与否。成功打开后，"打开串口"按钮会变成"关闭串口"按钮，串口设置也会置灰，不允许再修改配置。

串口打开失败的原因，可能是 USB 线没连接好或者串口没选对等。

串口成功开启后才能继续后续的操作。

图 5-2：调试-串口

5.2.2 开启卡片

只有一张 M1 卡，寻卡成功；多张 M1 卡产生冲突，也能寻卡成功。

由于杜邦线连接可能出现松散现象，有时会出现寻卡失败的情况。另外，建议给模块罩上一厘米左右外壳，卡片与天线距离过近通常会造成寻卡失败。

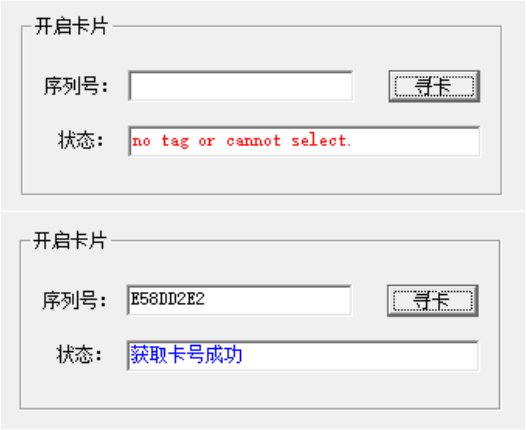


图 5-3：调试-开启卡片

5.2.3 密钥验证

寻卡成功即可进入读写块或者电子钱包操作，为避免密钥错误，可以先进行密钥验证。M1 卡的默认密钥是 0xFFFFFFFFFFFF(A 密钥、B 密钥)。

图 5-4：调试-密钥验证

图 5-5：调试-读块操作

5.2.4 读写块操作

密钥验证成功后便可以进行读写块操作。读扇区会将整个扇区 4 个块数据读出，读块则会将指定块数据读出。块 3 信息与整个扇区息息相关，依次存放扇区的 A 密钥、存储控制、B 密钥，密钥验证正是与该块信息进行比对。在下图测试中，由于 A 密钥的隐藏机制，通过读块操作得到的该段信息始终为"0x00

0000000000"，即使我们已经通过密钥验证确定该扇区的 A 密钥为默认密钥"0xFF FFFFFFFFFF"。

通过写块能将用户输入的数据写入卡片。上位机软件在实现过程中对于不满 16 字节的用户输入，会使用写入块原先的数据补足 16 字节；对于超过 16 字节的部分，则直接舍弃。

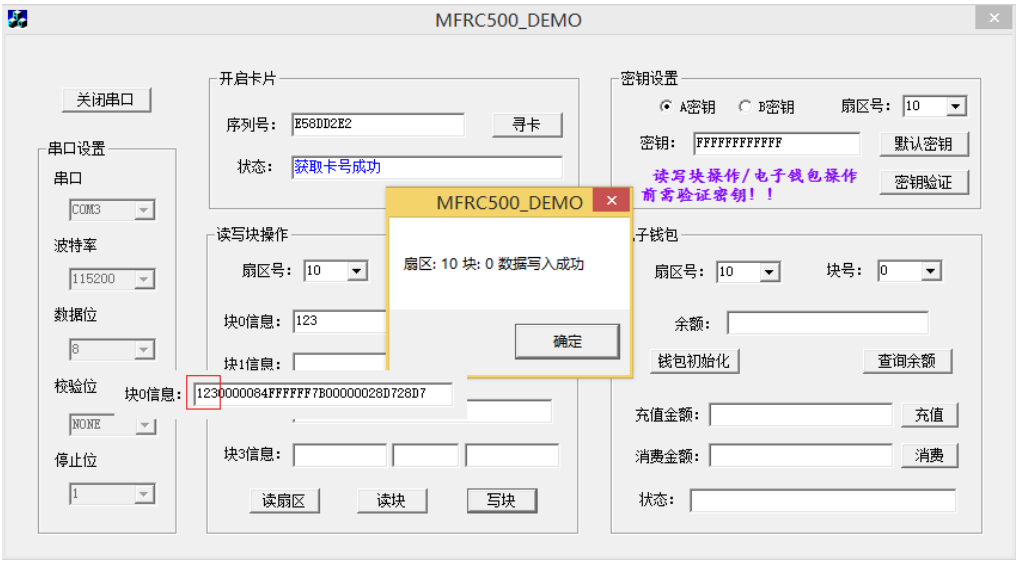


图 5-6：调试-写块操作

可以通过修改块 3 中 A 密钥、B 密钥的数据达到修改密钥的目的。由于块 3 的重要性，上位机在实现中特别添加了保护，在修改密钥时，建议只对一个字段进行修改。出于 A、B 密钥相对独立，这样可以通过另一密钥类型找回密钥或者修改回默认密钥，避免因密钥丢失损失该扇区。

5.2.5 电子钱包

电子钱包在进行初始化后便能进行余额查询、充值、消费操作。上图中依次通过 100 元初始化电子钱包，然后查询钱包余额，紧接着充值 100、消费 100 以验证电子钱包功能的准确性。

电子钱包以块为单位，除块 3 存储密钥信息外，其他块均能作为电子钱包使用。

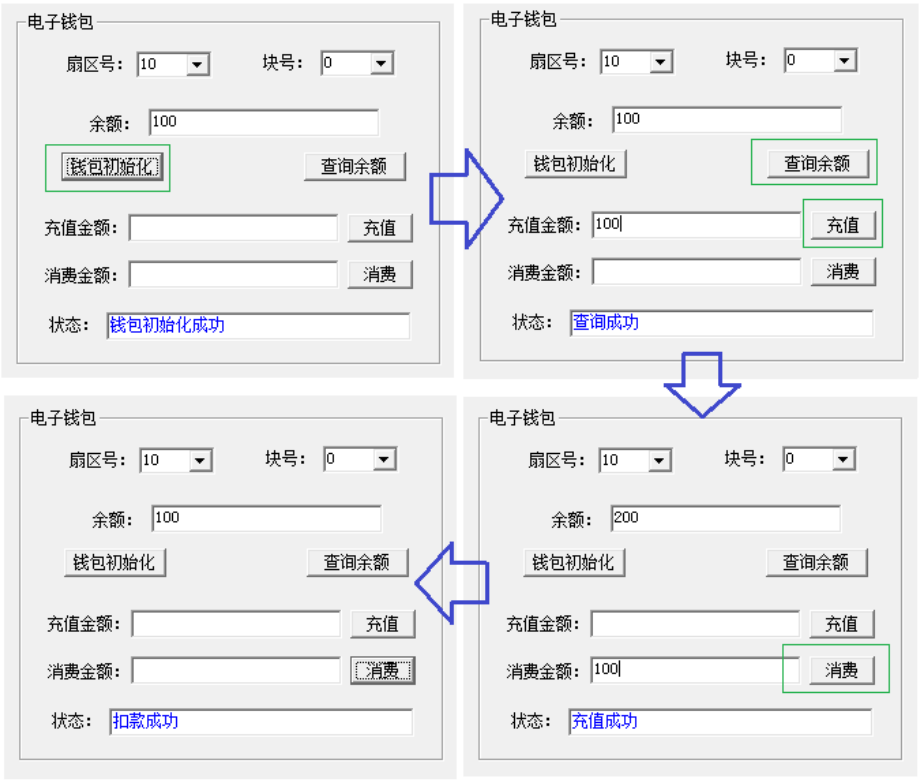


图 5-7：调试-电子钱包

5.3 测试结果分析

表 5-1：测试结果汇总

功能	测试结果
串口	正常开启、关闭；能够处理异常关闭等情况
获取卡号	正常获取；防冲突顺利进行；操作距离 2-10cm
密钥验证	正常进行密钥验证，密钥类型、控制字节符合 M1 卡规范
读块操作	读扇区、读块正常进行，块 3 密钥信息正确显示与保护
写块操作	正常写入，通过修改块 3 实现密钥更改
电子钱包	钱包初始化、充值、扣款、余额查询能正确执行

经过对本设计软硬件实现的测试，测试结果表明：射频读卡模块能够正确操作 M1 卡，完成所需的功能。射频读卡模块具有一定的穿透能力，工作距离满足

日常的使用需求。通过自定义的串口通信协议，STM32 能够控制射频读卡模块，并通过 USB 接口与上位机进行通信，完成所需数据的传输。在此基础上，搭建考勤系统、货物追踪等应用型系统将不是问题。

5.4 本章小结

本章主要介绍了系统的功能测试，包括测试所需的软件以及硬件如何连接使用。测试结果表明，本文设计的基于 MFRC500 和 STM32 的 RFID 读写器软硬件均工作正常，能够完成获取卡号、密钥验证、读写块操作、电子钱包操作等 M1 卡操作，满足本文设计的功能需求。

第 6 章 总结与展望

6.1 工作总结

本文广泛地查阅了资料，在了解 RFID 系统理论的基础上，完成了一套 RFID 读写器从硬件到软件的设计与实现，进行了以下工作：

(1)、系统学习了 RFID 系统理论基础，包括其工作原理、编码与调制以及本设计工作频率 13.56MHz 下的 ISO/IEC 144443 国际标准。

(2)、设计了基于 NXP MFRC500 的射频读卡模块，主要是天线线圈以及天线匹配电路的设计，并通过 STM32 微控制器实现对其的控制。

(3)、在射频读卡模块设计的基础上，设计了 PCB 实验板，并完成该实验板的焊接。

(4)、实现了基于 STM32 的下位机软件，包括射频读卡模块并行接口驱动、PCD/PICC 通讯函数、ISO14443A 协议函数。

(5)、设计了用于 PC 与 STM32 微控制器通讯的串口协议，保证二者数据传输的可靠性与准确性。

(6)、在串口协议的基础上，封装了一套基于串口的 RFID 读写器 API，完成本文设计功能需求的所有接口。

(7)、在 RFID 读写器 API 的基础上进行二次开发，实现了测试使用的上位机软件。测试结果表明，本文设计完成了设想的所有功能。

6.2 研究展望

本文设计的 RFID 读写器在已有设计的基础上可做如下的改进与扩展：

(1)、本文只完成较为复杂的射频读卡模块的设计与实现，后续可进一步完成基于 STM32 微控制器的底板的设计与实现。射频读卡模块可以无缝接入，另外通过添加蜂鸣器、液晶屏提供更为强大的交互方式。

(2)、本文设计的 RFID 读写器只支持 ISO14443-A，后续可以扩展为多协议读写器，包括工作频率 13.56MHz 下的 ISO14443-B 以及 ISO15693。

(3)、通过添加摄像头，本文设计的 RFID 读写器能够进一步整合成一个三合一 RFID 终端，能够对射频标签、条码以及二维码进行识别。

参考文献

[1] 王艳华，RFID 技术的发展与应用，科技资讯，（29），2011，15
[2] 贝毅君，RFID 技术在物联网中的应用，北京，人民邮电出版社，2013
[3] 李麟川，符合 EPCC1G2 标准的阅读器数字基带模块的设计与实现，硕士学位论文，南京，邮电大学，2011 年 5 月
[4] 戴逸飞，13.56MHz 阅读器芯片射频接口的研究和设计，硕士学位论文，武汉，华中科技大学，2007 年 5 月
[5] 郑洁，徐晶，RFID 读写器天线的研究与设计，微计算机信息(嵌入式与 SOC)，23(8-2)，2007,228-229
[6] Chia-Chun Tsai, Sheng-Bin Dai. The RF Circuit Design for Magnetic Power and Data Transmission, IEEE Radio Frequency Integrated Circuits Symposium, 2004, 6:12-18
[7] 陈博，基于 CLRC632 和 STM32 的 RFID 读卡器电路设计，硕士学位论文，天津，天津大学，2012 年 11 月
[8] NXP Semiconductor, Design of MF RC500 Matching Circuits and Antennas, Product data sheet, May 2000
[9] 张慧元，基于 MFRC500 的非接触式 IC 卡读写器的设计与实现，硕士学位论文，包头，内蒙古科技大学，2006 年 6 月
[10] ISO/IEC, ISO/IEC 14443, Identification cards -- Contactless integrated circuit cards, 2008, 2011
[11] Philips Semiconductors, Micore Reader IC Family: Directly Matched Antenna Design, 2006
[12] 戴彩艳，13.56MHz RFID 读写器天线的研究与设计，硕士学位论文，福州，福建师范大学，2013 年 6 月
[13] 菅洪彦，射频集成电路上电感的分析与优化设计，硕士学位论文，上海，复旦大学，2005 年 6 月
[14] 周润景，袁伟亭，张鹏飞，Cadence 高速电路板设计与仿真，北京，电子工业出版社，2009,418~427
[15] 张玉川，王彬，非接触式 CPU 卡的空中传输协议的软硬件设计，《单片机与嵌入式系统应用》，15(3)，2015
[16] 张冲，基于 ARM 和 RFID 技术的物联网广告机系统的研究，硕士学位论文，广州，华南理工大学，2014 年 5 月

致 谢

本论文的顺利完成，首先要由衷感谢我的导师胡建国博士。胡老师从论文的选题到论文的修改，都给予极大的指导与支持，倾注了大量心血。在此，谨向胡老师表示我最诚挚的谢意。

特别感谢实验室的吴老师，手把手教我元器件的焊接，最终顺利完成射频模块的设计与实现。吴老师的耐心教导，对本系统的硬件设计提供了极大的帮助。

感谢我的父母对我的养育之恩，他们潜移默化地塑造了我的性格。感谢我的家人对我学习的帮助和支持，使我能以积极的心态完成本科四年的学业。

最后谨向参与我论文评审的各位老师致谢。

年 月

毕业论文成绩评定记录

指导教师评语：   成绩评定： 指导教师签名：　　年　　月　　日
答辩小组或专业负责人意见：   成绩评定： 签名（章）：　　年　　月　　日
院系负责人意见：   成绩评定： 签名（章）：　　年　　月　　日