[摘要 iii](#_Toc478309261)

[Abstract iv](#_Toc478309262)

[第一章 绪论 6](#_Toc478309263)

[1.1课题来源 6](#_Toc478309264)

[1.2课题研究的目的和意义 6](#_Toc478309265)

[1.3 国内外研究概况 2](#_Toc478309266)

[1.3.1 国外对手术器械管理系统的研究 2](#_Toc478309267)

[1.3.2 国内对手术器械管理系统的研究 3](#_Toc478309268)

[1.4 论文的主要研究内容 4](#_Toc478309269)

[第二章 RFID射频技术原理 6](#_Toc478309270)

[2.1 RFID原理简介 6](#_Toc478309271)

[2.2 RFID应用介绍 10](#_Toc478309272)

[2.3 ISO15693标签 11](#_Toc478309273)

[2.3.1 ISO15693标签的协议分析 11](#_Toc478309274)

[2.3.2 ISO15693标签常用的命令集 15](#_Toc478309275)

[2.3.3 ISO15693标签特点分析 16](#_Toc478309276)

[2.3本章小结 17](#_Toc478309277)

[第三章 手术器械管理系统架构原理分析与设计 18](#_Toc478309278)

[3.1 系统整体结构规划设计 18](#_Toc478309279)

[3.1.1 手术器械管理系统的需求分析 18](#_Toc478309280)

[3.1.2 手术器械管理系统系统框图 19](#_Toc478309281)

[3.1.3 手术器械管理系统的实现方案 20](#_Toc478309282)

[3.2 RFID标签选型以及系统相关要求参数说明 21](#_Toc478309283)

[3.2.1 手术器械管理系统的相关参数要求 21](#_Toc478309284)

[3.2.1 手术器械管理系统的RFID标签选型 21](#_Toc478309285)

[3.3天线分布与设计概述 22](#_Toc478309286)

[3.3.1手术器械管理系统天线设计的基本要求 22](#_Toc478309287)

[3.3.2系统天线的结构 22](#_Toc478309288)

[3.3.3 天线尺寸的理论分析 23](#_Toc478309289)

[3.3.4 天线具体分布和形状 25](#_Toc478309290)

[3.4工控机概述与应用介绍 26](#_Toc478309291)

[3.4.1 工控机的概念 26](#_Toc478309292)

[3.4.2 工控机的特点 27](#_Toc478309293)

[3.4.3 工控机的应用 28](#_Toc478309294)

[3.5本章小结 28](#_Toc478309295)

[第四章 系统硬件部分 30](#_Toc478309296)

[4.1 射频前端电路设计 30](#_Toc478309297)

[4.1.1 发射电路设计 30](#_Toc478309298)

[4.1.2 接收电路设计 31](#_Toc478309299)

[4.2 FPGA编解码电路设计 31](#_Toc478309300)

[4.2.1 FPGA原理简述 31](#_Toc478309301)

[4.2.2 编解码电路的Verilog设计 33](#_Toc478309302)

[4.2.3 编解码电路的软件仿真 37](#_Toc478309303)

[4.3 MCU逻辑控制电路设计 38](#_Toc478309304)

[4.3.1 MCU原理简述与系统MCU芯片选型 38](#_Toc478309305)

[4.3.2 MCU电路设计 41](#_Toc478309306)

[4.3.3 MCU外围电路设计 42](#_Toc478309307)

[4.4 MCU与FPGA编解码电路接口设计 43](#_Toc478309308)

[4.5本章小结 44](#_Toc478309309)

[第五章 系统软件部分 45](#_Toc478309310)

[5.1 MCU软件设计 45](#_Toc478309311)

[5.1.1 STC单片机软件开发介绍 45](#_Toc478309312)

[5.1.2 MCU软件框架设计 47](#_Toc478309313)

[5.1.3 MCU核心代码的实现 47](#_Toc478309314)

[5.1.4 标签抗冲突代码实现 52](#_Toc478309315)

[5.2 上位机软件设计 53](#_Toc478309316)

[5.2.1 Qt技术简介 53](#_Toc478309317)

[5.2.2上位机软件框架设计 54](#_Toc478309318)

[5.2.3上位机操作界面的设计 55](#_Toc478309319)

[5.2.4后台控制程序的实现 56](#_Toc478309320)

[5.3 上位机与MCU的通信协议设计 60](#_Toc478309321)

[5.4 数据库的设计 61](#_Toc478309322)

[5.4.1数据库相关技术简介 61](#_Toc478309323)

[5.4.2数据库结构设计 62](#_Toc478309324)

[5.4.3系统中有关数据库程序 63](#_Toc478309325)

[5.5 本章小结 64](#_Toc478309326)

[第六章 系统可靠性的研究与改进 64](#_Toc478309327)

[6.1 系统的可靠性分析与改进 65](#_Toc478309328)

[6.2 系统在数据通信上的可靠性保障 65](#_Toc478309329)

[6.3 系统在数据读取上的可靠性保障 67](#_Toc478309330)

[6.4 系统的实物和运行界面展示 69](#_Toc478309331)

[6.5 本章小结 70](#_Toc478309332)

[第七章 结论与展望 71](#_Toc478309333)

[7.1结论 71](#_Toc478309334)

[7.2展望 72](#_Toc478309335)

[参考文献 74](#_Toc478309336)

[英文缩写词表 76](#_Toc478309337)

[作者在攻读硕士学位期间公开发表的论文及参加的项目 77](#_Toc478309338)

[A：在国内外刊物录用的论文 77](#_Toc478309339)

[B：参加的项目 77](#_Toc478309340)

[致 谢 78](#_Toc478309341)

基于ISO 15693芯片的RFID手术器械管理系统的研制

研究生：陆志峰 学科专业：通信与信息系统（RFID技术应用）

指导教师：景为平研究员

# 摘要

随着智能化、网络化生产生活的普及和生活质量的提高，促使人们对医疗设备提出更高更智能化的要求，促使了高新技术在医疗设备制造行业的普及，一方面提高了医疗设备的智能化水平；另一方面也增加了医疗设备的复杂性，对其安全性和可靠性提出了更高的要求。尤其是在手术过程中，规范的管理和使用手术器械，不仅是保障医师正常进行手术的基础，也是保证病人生命安全的必要条件，更是给手术中器械管理人员以及器械管理软件系统提出了越来越高的要求。

根据一些调查和研究结果显示，在手术中手术异物遗留导致手术失败的比例占绝大多数。特别是在高难度的手术中，由于参与手术的医疗人员比较多，使用的手术器材同样是多种多样，并且手术过程高度紧张，手术时间相当紧迫，争分夺秒；另外由于纱布和某些手术器械完全浸没与血肉间，给肉眼识别带来极大的困难。即便是在有多名医疗器械管理人员，对手术器械进行清点的情况下，发生手术器械残留在病人体内的案列也是屡见不鲜。因此给病人的健康，乃至生命带来极大的威胁。

本文针对RSIS的难题，提出了基于ISO15693射频标签的射频识别(RFID)手术器械管理系统。将ISO15693射频标签内置在手术器械中，采用三通道天线协调工作的RFID阅读器完成对手术器械的登记录入、回收统计和寻找扫描。借助Qt图形界面应用程序开发框架，实现对手术器械管理系统的上位机操作界面的设计。实验表明系统能够稳定、快速、准确地实现对手术器械使用和回收的智能化管理，有效地防止了手术器械遗留问题的发生，给手术安全提供了更加高效可靠的保障。

关键词：RFID；RSIS；ISO15693射频标签；手术器械；Qt

**Design of the RFID Operating Instruments Management System Based On ISO15693 Electronic Tags**

POSTGRADUATE:Lu zhifeng

SPECIALIZATION: Application of RFID

Directed by Prof.Jing Weiping

### Abstract

With the popularity of intelligent, networked production life and improving of quality of life, encouraging people to put forward higher and more intelligentized medical equipment, and promoting the high and new technology in the medical equipment manufacturing industry. On the one hand this improved the intelligent level of the medical equipment; On the other hand this added the the complexity of medical equipment, puted forward higher requirements on its safety and reliability. Especially in the operation process, and the use of standardized management of the medical equipment, is also a necessary condition to ensure the patient safety, it is put forward higher request to the operation medical equipment management personnel and software management of the medical equipment system.

According to some investigation and research, the Retained surgical items(RSIS) lead to the failure has the most proportion in the surgery. Especially in difficult surgery, as a result of the operation of the medical personnel is more, the use of surgical equipment is also varied, and the operation process, operation time is pressing, race against time. Because the gauze and some other surgical instruments total immersion and flesh and blood,it is difficult to identify by the naked eye. Even server medical equipment management personnel to count the surgical instruments, there are some case of the RSIS occuranced.All this bring great threat to the patient's health and life.

In order to solve the problem of retained surgical items (RSIS), the paper proposed a radio frequency identification (RFID) operating device management System based on ISO15693 electronic tags. The Surgical devices contains the ISO15693 electronic tags and the RFID reader having three channels cooperating to achieve the surgical instruments’ registration, recycling and scanning. Using Qt GUI application development framework to design the upper computer’s operation interface of the operating device management System. Experiments show that the system can stably, accurately and timely accomplish the intelligent management of the operation devices’ using and recycling, in order to prevent the problem of RSIS, actually the system can provide more reliable safeguard for The safety of operation. System now has the capability of practical application and is going on the medical application promotion.

**Key words:** RFID;RSIS;ISO15693 RFID Tag; Surgical devices;Qt

### 

### 第一章 绪论

### 1.1课题来源

本课题来源于低功耗射频识别标签芯片研制，江苏省科技支撑计划-工业部分（重点），项目编号BE2013008-3。

### 1.2课题研究的目的和意义

根据一些调查和研究结果显示，在手术中异物遗留导致手术失败的比例占绝大多数。尤其是在高难度的手术中，由于参与手术的医疗人员比较多，使用的手术器械更是多种多样，而且手术整个过程处于高度紧张，手术时间非常的紧迫，抢分夺秒；此外一些手术器械完全浸没在血肉间，给肉眼识别带来了极大的难度。即使是有多名医疗器械管理人员对手术器械进行清点，也难免会发生手术器械遗留在病人体内的事列。

2012年9月《纽约时报》报道了一位女士因手术异物遗留造成严重伤害的案例。数据显示，从2005年至2016年，联合委员会警讯事件数据库共收到772例手术异物遗留事件报告，其中造成16人直接死亡[1]。这表明手术异物遗留给患者不仅带来了不必要的伤害和疼痛的折磨，甚至直接威胁到他的生命和安全。在医患关系如此紧张的当今社会，手术异物遗留问题必须引起我们的重视。要解决手术异物遗留的问题，就在于对手术中手术器械的使用和回收要进行科学、严格的管理，杜绝任何手术器械遗漏的情况发生。通过调研发现,目前国内医院的手术为防止手术器械（如纱布、手术剪刀、血管钳等）遗留在病人体内，采用严格的三人四次清点制度完全依赖人工清点和回收，这不仅增加了手术的复杂度和管理成本，而且不能对手术中使用过的所有手术器械进行统一规范的管理统计[2]。

面对这样的难题与困境，基于对手术器械的使用和回收做校验的设计思想，发挥RFID技术的优势，在手术器械中采用RFID技术来识别、追踪和定位手术器械可以大大减少意外发生的概率，确保使用过的手术器械可以完全回收无遗漏，也使得手术器械管理员的工作任务更加的简单和安全可靠。这不仅可以让医疗保险事业的信心重振，也给病人的生命安全提供了最后一道可靠的保障。

### 1.3 国内外研究概况

### 1.3.1 国外对手术器械管理系统的研究

目前RFID技术在医疗领域的运用是多方面的，其在医疗领域的运用，提高了医院的管理效率及治疗效率，产生良好的经济效益和社会效益[3]。从全世界RFID技术在智能手术器械管理系统的研究来看，美国的一些公司对于手术器械管理系统的研究，设计及其应用都领先世界水平。目前，已经有三家美国公司先后成功发布了自己的产品，分别是Patient Safety Technologies Inc ，RF Surgical Systems Inc和ClearCount Medical Solutions这三家公司。

RF Surgical Systems Inc一家医疗设备公司，该公司的产品叫RF Assure Detection System X，包括两部分：嵌有1bit数据标签的纱布和检测扫描系统。该系统采用的是低频非电离性的无线电波去探测嵌有标签的纱布。运用低频是为了使信号的穿透性更好的目的，最终减少纱布遗留在病人体内的风险。医生可以根据该系统知道是否还有纱布在患者的体内，但是该系统的局限性在于不能识别出遗留的纱布具体数目以及该系统的识别距离太近。

Patient Safety Technologies Inc是一家加州的医疗器材公司，其主要产品涉及的领域是维护病患安全的医疗器材。该公司的the Safety-Sponge System 产品，可以统计手术时手术器械的消耗量和使用情况。主要包含三个部分：贴有条码的手术器械、数据处理系统和次数计数器（扫描器）。它采用小型的移动电脑来为手术提供一个实时的计数功能，扫描器不断地扫描计数，能够判别哪些类型的器械是否存在或者失踪，给手术提供了安全保障。但是它的局限性在于，对手术过程记录的信息太少。

ClearCount Medical Solutions 公司，他们销售的产品是Smart Sponge系统，主要是用来监测在手术中，医用耗材的使用情况。该系统将RFID标签内嵌到手术耗材内，在手术整个过程中使用RFID阅读器来确认手术耗材的序号和位置。据该公司称，Smart Sponge系统是全球第一个可以对手术中使用的纱布和海绵进行识别、扫描、读取和统计的RFID管理系统。该系统使用无线电磁波来标识自己，在血液和组织中依然可以快速找到丢失的手术器械，为手术的安全提供了可靠的保障。该系统的局限性在于成本太高，没有被大规模的应用。

### 1.3.2 国内对手术器械管理系统的研究

国内对RFID技术的研究起步比国外要晚很多，而且各个地方发展也不均衡。直到本世纪初，我国才对RFID技术开始大力的发张研究。到目前，国内才在北上广基本建立了RFID产业链雏形，其中上海以RFID的芯片研究为主，北京以系统的集成方案为主，而深圳着重发张应用方向（如远望谷等）。

目前国内手术对手术器械的管理和使用很重视，但由于国内医疗设备制造业与国外顶尖的医疗设备制造公司具有很大的示例差距，导致国内仍主要依赖人工清点核对。通过对国内的手术调研发现，为了确保手术器械无遗留，避免医疗事故的发生，手术中采用“三人四次”的严格清点制度。其中清点的人员由手术医生助理、手术器械护士和巡回护士三人组成，然后分成四次清点，并对清点的方法和时间有严格规范的规定。这样不仅增加人员的使用成本，而且使得整个手术室很繁忙杂乱。

国内对于智能手术器械管理方面的研究目前主要集中在广东深圳地区，有些公司已经进行了探索并推出了一些相应的产品，例如粤发实业有限公司研发的超高频智能电子纱布标签专门用于手术纱布的。但是，由于该公司的技术原因，整个产品性能不高，并没有达到高可靠性和稳定性的预想。此外还有一家规模比较大的公司，成立于2005年的深圳市新元素医疗技术开发有限公司，集医疗、IT与应用一体，专门致力于医疗应用系统方案解决、系统信息化集成，拥有自主的品牌和知识产权。该公司设计的电子纱布系统，能够准确地统计手术前准备的纱布总量，手术中使用了并完成回收的纱布数量，以及手术结束后剩余未使用的纱布总数，另外还可以探测到遗留在病人体内的纱布数量和位置。该系统使用的是RFID技术，在纱布内嵌入一个与硬币差不多大小的小型RFID标签，与普通的纱布几乎没有任何区别。但是大体上来讲，国内的只能手术器械管理系统还是相对比较薄弱的，还有很长的路需要走。

虽然这几年国内RFID技术在各个领域的应用都取得很大的进步，但是在医疗器械管理上还有很多需要探索研究的地方。就目前而言，国内的RFID技术仍然处于起步阶段，在医疗行业的应用上尤其是在手术器械管理系统上更是处于萌芽阶段，其发展空间、市场都是非常巨大，前景诱人的。

### 1.4 论文的主要研究内容

本文提出了一种基于ISO15693射频标签的射频识别手术器械管理系统。系统以RFID技术为核心，结合高效的微处理器MCU、精准的现场可编程门阵列器FPGA和友好的Qt图形交互界面，实现了一个稳定、高效的智能化手术器械管理系统。通过对手术器械使用和回收的智能化管理，能够达到有使用就有回收的标准，为手术器械的管理带来了极大的帮助，有效地避免了手术过程中异物遗留问题的发生。

第一章绪论对论文的课题来源、研究意义及国内外针对手术器械管理的医疗设备的研究动态做出概述，指出了本文的主要工作内容和研究目标，同时给出本文的组织结构。

第二章介绍RFID射频技术原理，以及RFID的一些相关应用。并结合ISO15693标签的协议标准和命令集分析其相关特性。

第三章结合系统设计开发的生命周期，以从上到下的设计理念，给出手术器械管理系统各部分的组成和整体架构，并对各个部分分别进行了简要分析和描述。

第四章具体介绍了手术器械管理系统硬件部分（RFID阅读部分）的设计。并分成射频前端电路、FPGA编解码和MCU逻辑控制电路三部分以及之间的连接部分进行详细分析和介绍。

第五章具体介绍了手术器械管理系统软件部分的设计。总体分为MCU端的软件和安装在工控机上的上位机控制软件。介绍的过程中顺带介绍了应用的相关技术和知识，可以使读者更快更易读懂。在第三小节还介绍了专门设计的上位机与MCU通信协议。

第六章介绍了在整个手术器械管理系统设计完成之后，根据一些可靠性检测的理论，对系统的准确度和可靠度进行了优化设计。

第七章作为总结和展望部分，对本文所完成的任务做了总结，指出了本文的工作量、创新点和取得的成果，并且提出了今后需要做的工作和下一步努力的方向。

### 第二章 RFID射频技术原理

### 2.1 RFID原理简介

射频识别(RFID)，又称无线射频识别技术，是无线电频率识别的简称，即通过无线电波进行识别。它是一种通信技术，即可以借助无线电讯号来识别目标，并进行数据的读写，读取存放在目标内的电子信息，却不需要识别系统和所识别的目标之间建立机械或者光学上的接触，整个过程使用无线通信来实现。

2.1.1 射频识别系统构成

采用射频识别技术的识别系统一般由两个部分构成：

(1)应答器，即标签芯片，放在被识别的物体上。应答器是射频识别系统的数据载体，通常，应答器由耦合元件以及微电子芯片组成。

(2)阅读器，一种可以读写标签芯片的装置。它通过发送请求命令，并等待应答器的应答，对应答信号进行读取。此外，许多阅读器还会附加各种接口，以便将从应答器读取的数据传输给上级系统，如PC等。

图2-1所示为RFID系统的基本组成单元，右边为具备耦合元件的阅读器，左边为含有耦合元件的应答器，二者通过各自的耦合元件建立起无线通信，通过电磁耦合的方式实现能量传输以及数据传输。



图2-1 RFID系统简图

2.1.2 射频识别系统分类

根据不同的分类标准，RFID系统可以有多种分类，以下是几种比较常见的分类。

1.根据标签的供电方式

根据标签供电方式的不同，行业将标签分为有源标签和无源标签。有源标签指的是在标签内部自身含有电池，标签的工作用电由电池供应。这样的RFID系统称为主动射频系统。有源标签适合远距离、高速度读写数据，而且可以主动发送射频信号和阅读器进行通信。但是此种标签体积大、价格高，而且使用寿命受限于电池而较短。无源标签则是内部没有电源，主要通过耦合阅读器的射频信号来获取能量。此种RFID系统也称为被动射频系统。无源标签读写距离较短，但是其体积小、价格低且使用寿命较长。

2.根据RFID系统工作频率

根据RFID系统的工作频段分，可以分为四种：低频系统（30-300KHz）、高频系统（3-30MHz）、超高频系统（300-2.45GHz）和微波系统（2.45GHz以上）如表2-1所示。

表2-1 RFID芯片分类

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 低频 | 高频 | | 超高频 | 微波 |
| 工作频率 | 125~134KHz | 13.56MHz | JM13.556MHz | 868~915MHz | 2.45~5.8GHz |
| 市场占有率 | 74% | 17% | 0% | 6% | 3% |
| 读取距离 | 1.2m | 1.2m | 1.2m | 4m（美国） | 15m（美国） |
| 速度 | 慢 | 中等 | 很快 | 快 | 很快 |
| 潮湿环境 | 无影响 | 无影响 | 无影响 | 影响较大 | 影响较大 |
| 方向性 | 无 | 无 | 无 | 部分 | 有 |
| 全球适用频率 | 是 | - | 是 | 部分（欧盟、美国） | 部分（非欧盟国家） |
| 现有ISO标准 | 11784/85、14223 | 18000-3/1  14443 | 18000-3/2  15693，A,B和C | EPC C0，C1,C2，G2 | 18000-4 |
| 主要应用范围 | 进出管理、固定设备，洗衣店 | 图书馆、产品跟踪、货架、运输 | 空运、邮局、医药、烟草 | 货架、卡车、拖车跟踪 | 收费站、集装箱 |

其中，在低频系统中常用的频率为125KHz和134KHz，适合近距离的场合，如门禁、动物耳标等。高频系统中常用的频率为13.56MHz，适合信用卡、资产管理等。超高频系统常用的频率为860MHz-915MHz，适用于远距离场合工作，例如车辆跟踪，物流管理等。微波系统中一般工作频率为2.45GHz和5.8GHz，比较适合较远距离的场合，高速公路收费系统等。

3.根据标签的存储器类型

根据标签存储单元的类型一般将RFID系统划分为：只读型、一次写入型、重复读写型。只读型为标签在出厂时信息已经固定，对于用户来说只能够读取标签信息，但是无权修改。该系统可用于门禁物流管理等。一次写入标签只允许用户写入数据一次，一旦写入后不得修改，只可读。此种系统一般用于资产管理等贵重物品管理。多次重复读写标签允许用户可以任意次读写标签数据，此类系统一般用于信用卡服务等。

2.1.3 RFID标签

RFID标签具有条形码所不具备的防水、防磁、耐高温、使用寿命长、读取距离大、标签信息加密、存储数量容量大、存储信息可更改等优点。常见的RFID标签如图2-2所示：



图2-2 常见的RFID标签

RFID标签主要性能参数：

* + 工作频率
  + 读/写能力
  + 编码调制方式
  + 数据传输速率
  + 信息数据存储容量
  + 工作距离
  + 多应答器识别能力（也称防冲突能力）
  + 安全性能（密钥、认证）等。

RFID标签的内部结构主要是RFID芯片和天线组成，具体如图2-3所示：



图2-3 RFID标签的内部结构

RFID芯片内部电路主要由射频前端的接收和发射电路、电源电路、逻辑控制电路和存储器件组成。基本结构如下图2-4所示：

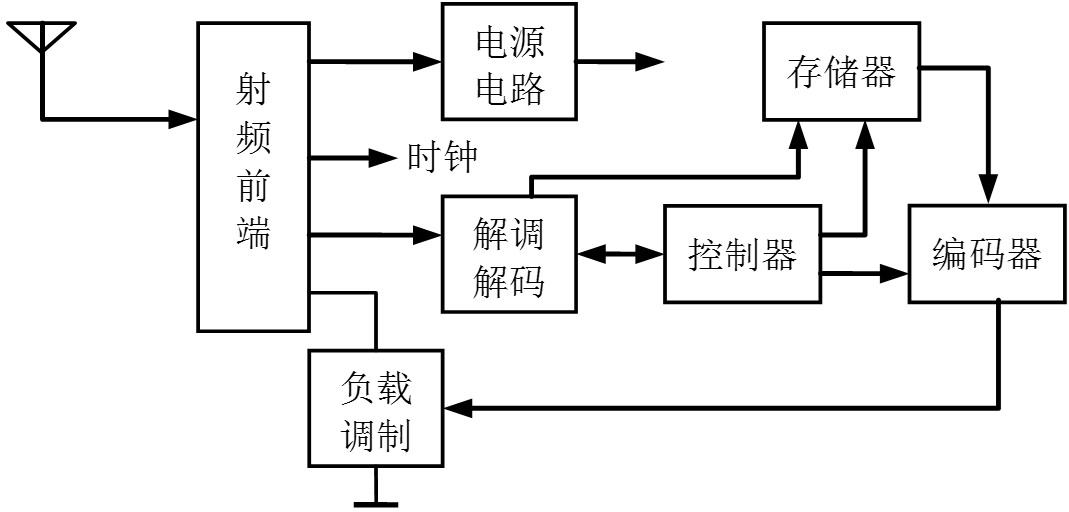


图2-4 RFID芯片内部结构图

2.1.4 RFID阅读器

一台常规的阅读器含有高频接收和发送模块、控制单元和与耦合元件。此外，许多阅读器还都有附加的接口（RS232、RJ45、WIFI等），以便将获取的数据向更高层系统传送。

阅读器的电路组成如图2-5所示：

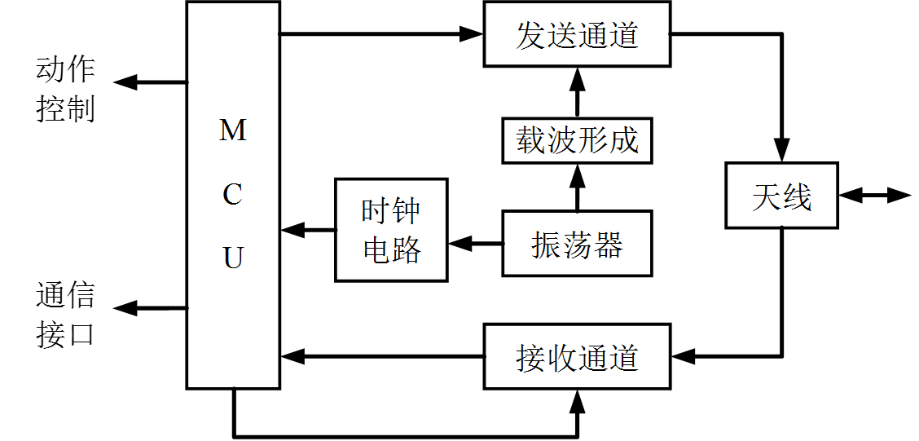


图2-5 阅读器电路结构图

阅读器的功能:

* 1. 以射频方式向应答器传输能量；
  2. 从应答器读取数据或者向应答器写入数据；
  3. 对读取的数据完成处理并执行相应的操作；
  4. 若有需要，应能和高层处理交互信息。

常见的阅读器如图2-6所示：



图2-6 常见的阅读器

### 2.2 RFID应用介绍

目前射频识别按载波频率分为低频（LF）、高频（HF）、超高频（UHF）和微波（MW）。因为不同波段的无线电波具有不同特性，各个频段的射频识别具有不同的应用场景。

低频的频率主要为：125kHz和134.2kHz。由于该频段在金属和液体背景下有较好的性能，目前主要应用于动物体内外标签，门禁及危险品钢瓶，蔬菜水果肉类运输标签。常用只读形式，也有少量读写芯片。该类芯片通常不加密。高频的频率主要为：13.56MHz。由于该频段天线简单体积小，有加密，主要用于证票类，一般为读写卡。该卡又分逻辑卡和CPU卡。逻辑卡是最常用，如身份证，门票，交通卡，它的简化版用于一次性车票，如铁路客票。由于高度信息安全CPU卡将用的越来越多，如身份证，交通卡，银行卡，而逻辑卡则逐步退到门禁等领域。在这个频段还有由于手机支付等应用的NFC（短距离无线通信）。超高频的频率主要为：433MHz和915MHz。常用于于有源形式。有十几米的识别距离。915MHz也常有无源形式，物流和车牌用的多。微波的频率主要为：2.45GHz、5.8GHz。该频段都为有源。识别距离远，方向性好。2.45GHz常用于集装箱，和水路。5.8GHz常用于道路收费。

### 2.3 ISO15693标签

目前，国际上有许多针对RFID芯片的协议。分别在不同的频率范围内都有相应的协议规范。在13.56MHz的频率下，有ISO/IEC-14443系列、ISO/IEC-15693等协议，各种协议下的RFID标签芯片可以适用在不同的场合。本文用到的RFID芯片是一款满足ISO/IEC-15693协议的高频芯片，在现代生活中，可以应用在产品供应链管理、图书管理、防伪等领域，具有广泛的应用市场。

### 2.3.1 ISO15693标签的协议分析

阅读器(Vicinity Coupling Device，VCD)和标签芯片（Vicinity Card，VICC）的通信采用命令帧格式传输。表2-2为VCD到VICC的请求命令帧格式，帧格式包括帧头SOF（Start of Frame），请求标志（Flag），命令码（Command），若干参数（Parameter）与数据（Data），最后是CRC（Cyclic Redundancy Check）校验和帧尾EOF（End of Frame）。根据每条命令的不同，帧格式中的参数和数据也不同。VICC到VCD的响应帧格式比请求帧格式中少了命令码项，其余部分格式相同。

表2-2 请求命令帧格式

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SOF | Flag | Command | Parameter | Data | CRC | EOF |
|  | 8 bits | 8 bits | m bits | n bits | 16 bits |  |

注: m、n 均为8的非负整数倍

协议规定了其通信命令帧的编码模式也有多种。以下简要介绍芯片通信的编码模式。

VCD到VICC的通信编码模式

VCD到VICC的通信编码模式有两种，一种是256取1，另外一种是4取1，两种编码模式通信速率不同，适用的场合也不同[33]。

1. VCD到VICC的SOF编码

如图2-7所示，为SOF的两种编码模式，上面的是256取1编码的SOF，下面一种为4取1编码的SOF，两种编码方式都是在第一个开始的9.44us产生一个暂停（pause）状态，并且都总共持续了两个37.76us。不同的是第二个pause产生的位置不同，从图中可以看到256取1的编码方式是在最后，不同于4取1编码。

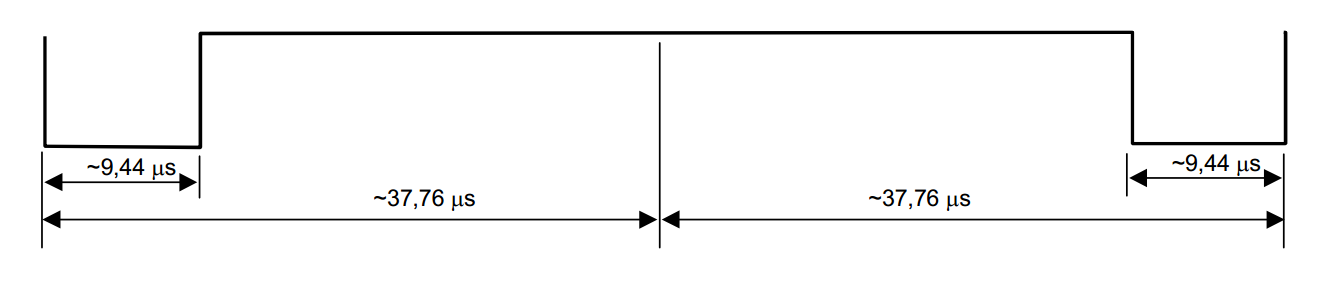
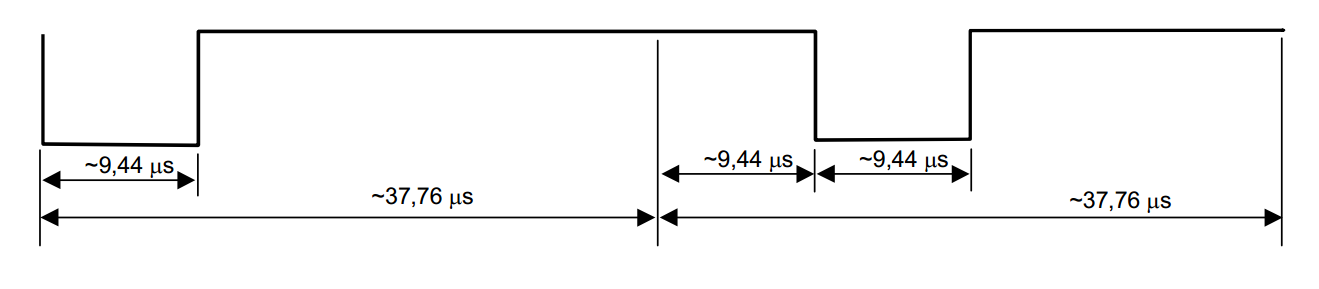
 

图2-7 VCD到VICC的SOF编码

2. VCD到VICC的EOF编码

这里的EOF编码，无论VCD采用何种编码方式，EOF只有一种形式，如图2-8所示。从图中可以看出，pause产生在倒数第二个9.44us处。

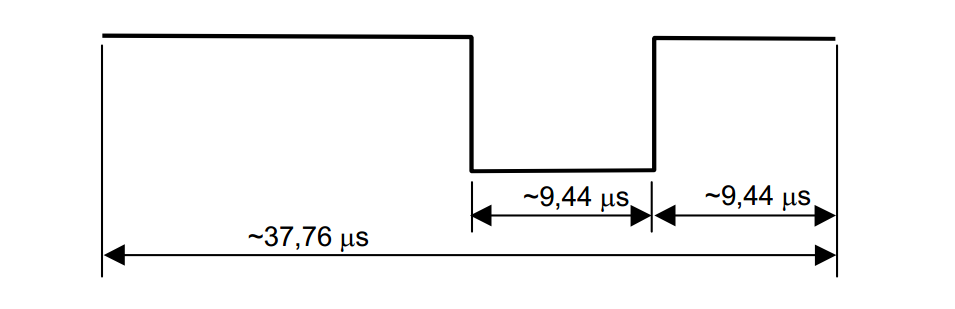


图2-8 VCD到VICC的EOF编码

3.VCD到VICC的数据编码

(1)256取1编码模式

此种编码模式采用一个字节产生一个pause的位置的方式来表示被编码的字节。所以，每次编码为一个字节，在256个连续的18.88us中，通过被编码的字节的数值，决定pause产生的位置。如数据“E1”，其十进制位225，则在第225个18.88us的后半部分产生一个pause，如下图2-9所示。这样传输一个字节需要4.833ms，数据速率为1.54kbits/s。

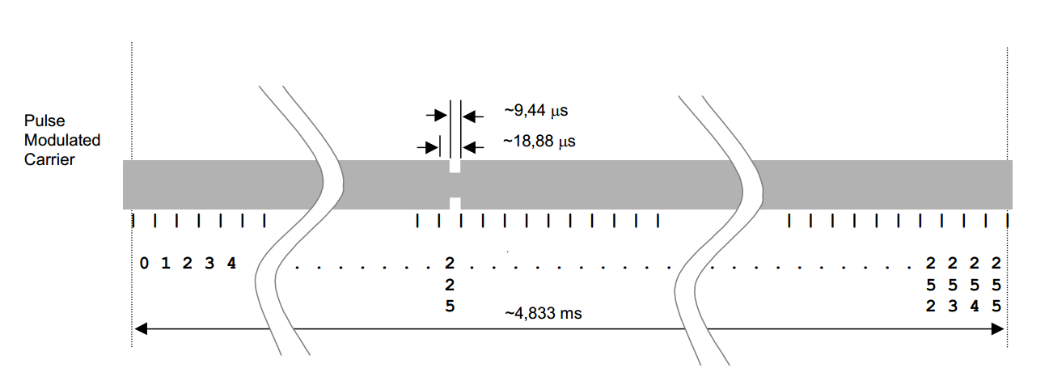


图2-9 256取1编码模式图

(2)4取1编码模式

此种编码方式为每2bits编码，在连续4个18.88us中，根据2bits值来决定产生pause的位置，如图3-5所示。从图2-10中可以看到，如果被编码的2bits为“00”，则在第0个18.88us的后半个位置处产生一个9.44us的pause，其余编码依次类推。这种编码模式一个bit需要2个18.88us，所以数据速率为26.48kbits/s。

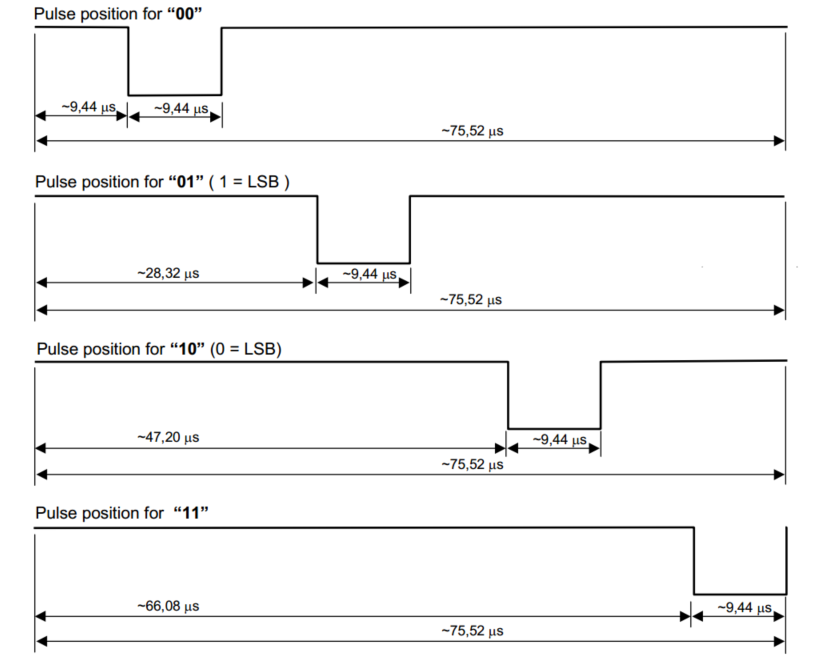


图2-10 4取1编码模式图

VICC到VCD的通信编码模式

从VICC到VCD的通信采用负载调制，通过载波来产生需要频率的副载波。在通讯中，有VCD控制VICC选择使用一种或者两种副载波，当使用一种副载波时，频率fs1为fc/32，当使用两种副载波时，其另外一种副载波频率fs2为fc/28。另外，VCD还可以通过请求命令的flag配置VICC的返回采用高速通信还是低速通信，下面介绍VICC到VCD的编码方式均是高速通信的编码方式，对于低速编码方式，则使用同样的副载波，但是脉冲数是原来的4倍。

1.VICC到VCD的数据编码

VICC到VCD的数据编码采用曼彻斯特编码，如图2-11所示。图上半部分是逻辑0的编码，以频率fs1的8个脉冲开始，接着是约18.88us的非调制时间。而下半部分是逻辑1的编码，与逻辑0的编码恰好相反。所以一个bit传输的时间是37.76us，即通信速率为26.48kbits/s。这是采用单副载波的编码，对于双副载波，即是对非调制部分，采用频率fs2的9个脉冲进行调制。计算得出传输速率为6.67 kbits/s。以下双副载波编码方式同样如此，不在赘述，仅介绍单副载波模式。

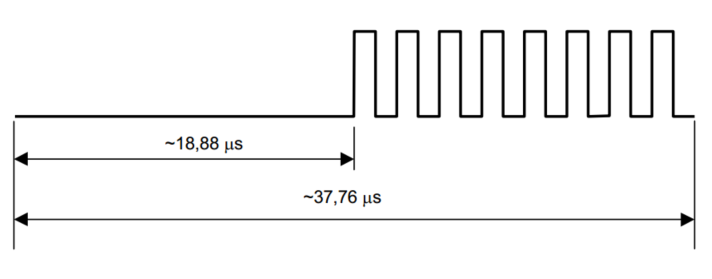
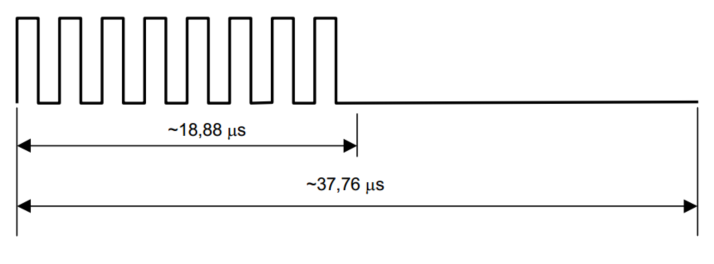


图2-11 VICC到VCD的单副载波数据编码

2.VICC到VCD的SOF

如图2-12所示，SOF包含三个部分，分别是一个56.64us的非调制时间、频率为fs1的24个脉冲和一个逻辑1。

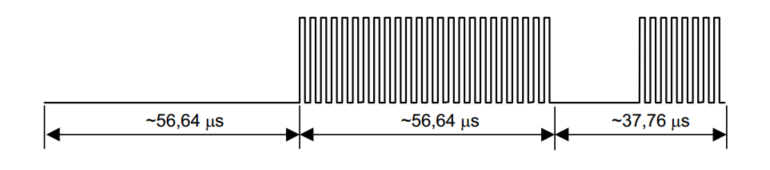


图2-12 VICC到VCD的单副载波SOF

3.VICC到VCD的EOF

如图2-13所示，EOF也包含三个部分，分别是一个逻辑0、频率为fs1的24个脉冲和一个56.64us的非调制时间。

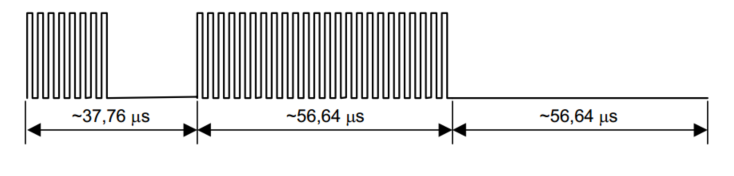


图2-13 VICC到VCD的单副载波EOF

### 2.3.2 ISO15693标签常用的命令集

标签在场强中工作，与阅读器的通信执行不同的动作是通过不同的命令来实现的，即表2-2所描述的命令帧中的Command，也就是说，一个标签必须具备一个命令集。ISO/IEC-15693协议就规定了一系列的命令集以及扩充命令集的规则。表2-3就是15693协议所规定的命令集。

表2-3 15693协议命令集

|  |  |  |
| --- | --- | --- |
| 命令码 | 类型 | 功能描述 |
| 01 | 强制 | 寻卡 |
| 02 | 强制 | 标签进入静默状态 |
| 03-1F | 强制 | 留作以后使用 |
| 20 | 可选 | 读单个块数据 |
| 21 | 可选 | 写单个块数据 |
| 22 | 可选 | 锁单个块数据 |
| 23 | 可选 | 读多个块数据 |
| 24 | 可选 | 写多个块数据 |
| 25 | 可选 | 标签进入选择状态 |
| 26 | 可选 | 标签进入复位状态 |
| 27 | 可选 | 写AFI数据 |
| 28 | 可选 | 锁AFI数据 |
| 29 | 可选 | 写DSFID数据 |
| 2A | 可选 | 锁DSFID数据 |
| 2B | 可选 | 获取标签信息 |
| 2C | 可选 | 获取标签多个块安全状态 |
| 2D-9F | 可选 | 留作以后使用 |
| A0-DF | 定制 | IC 制造商决定 |
| E0-FF | 私有 | IC 制造商决定 |

上表中，第一列为命令编码，使用一个Byte来表示，第二列为命令的类型，第三列为命令的功能描述。首先是01和02命令，是强制命令，也就是凡是满足此协议的标签芯片必须支持这两条命令。而03-1F命令也是强制命令，但是协议还没有规定功能，只是用来以后的协议扩充使用。20-2C这些命令对于标签芯片是可选择的。2D-9F同样留作以后使用。

### 2.3.3 ISO15693标签特点分析

ISO15693标签工作频率为：13.56MHz±7kHz，属于高频标签。ISO15693标签具有一个8字节共64bit的全球唯一序列号(UID),这个UID一方面可以使全球范围内的标签互相区别，更重要的是可以在多标签同时读写时用于防冲突。ISO用一个字节的AFI (Application family identifier)来区分不同行业中的电子标签。AFI的高半字节表示主要行业，低半字节表示主要行业中的细分行业。15693国际标准还规定了一个字节的可选的数据存储格式识别符(DSFID)，用来区分标签中不同的数据存储格式。如果标签支持DSFID，在清点命令中标签将返回一个非零的DSFID，读写器可据此判断射频场中的标签是否具有期望的数据格式。电子标签的内存最大可达8K字节，以数据块(Block)为单位进行管理，标签内最多可以有256个数据块，每个数据块最大可以有32字节。数据块的内容可以锁定以防止修改。标签具有4个相互转换的状态：Power off状态：在标签未进入到有效磁场区域时标签处于Power off状态；Ready 状态：被激活后选择表示未设立时，处理任何请求；Quiet 状态：不处理任何标签清点指令，可接受直接寻址的命令；Select状态：仅响应选择标识符设置的请求。

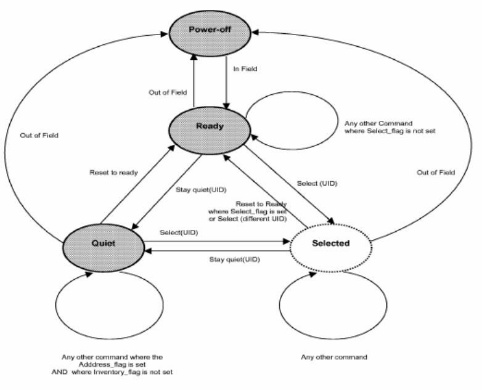


图2-14 ISO15693芯片内部状态转换图

综合来看此波段的特点为：读取距离在10cm到1m不等，芯片具有抗冲突特性，用于读取环境具有多张标签，距离适中的场景。典型应用包括：电子车票、电子闭锁防盗、大厦门禁系统、小区物业管理、物联网系统等。

### 2.3本章小结

本章开头从RFID原理开始分析和介绍，大体从RFID系统的组成、RFID的分类、各个频段RFID芯片和阅读器研究介绍，让读者对RFID系统原理和组成有一个大体的了解。后面对本文要用到的ISO15693协议进行了分析和研究，分析了协议中的编码解码规定，以及标签与阅读器之间的通信方式的规范，并结合ISO15693芯片的命令集含义进行介绍，这些重要的协议内容有助于后面对文章标签阅读部分的理解和依据。最后分析了ISO15693标签的物理特性和应用场景，变相的说明了本文手术器械管理系统为何选用ISO15693RFID标签的原因。本章总体上介绍了什么是RFID，RFID有哪些分类，RFID系统的组成，ISO15693协议规定了什么内容和为何本文选用ISO15693标签。

### 第三章 手术器械管理系统架构原理分析与设计

### 3.1 系统整体结构规划设计

手术器械管理系统的开发周期是指从该系统产生到该系统停止应用的整个周期。我们将手术器械管理系统生命周期划分为：手术器械管理的问题定义阶段、手术器械管理的需求分析阶段、手术器械管理的软件架构阶段、手术器械管理系统的编码与测试阶段、运行和维护阶段等。

### 3.1.1 手术器械管理系统的需求分析

将手术器械管理系统的开发周期大体上分为：分析需求并提出解决方案、实现设计和部署，测试评估。如果后续市场或者客户对该手术器械管理系统还有其他的需求更新或者功能需求，我们将会继续进行分析需求，提出实现方案…如此循环处理，这就是周期模式中的迭代式模式，我们每次“迭代”都会产生一个可以更新发布的产品。手术器械管理系统开发遵循的迭代思想可以用图3-1展示：

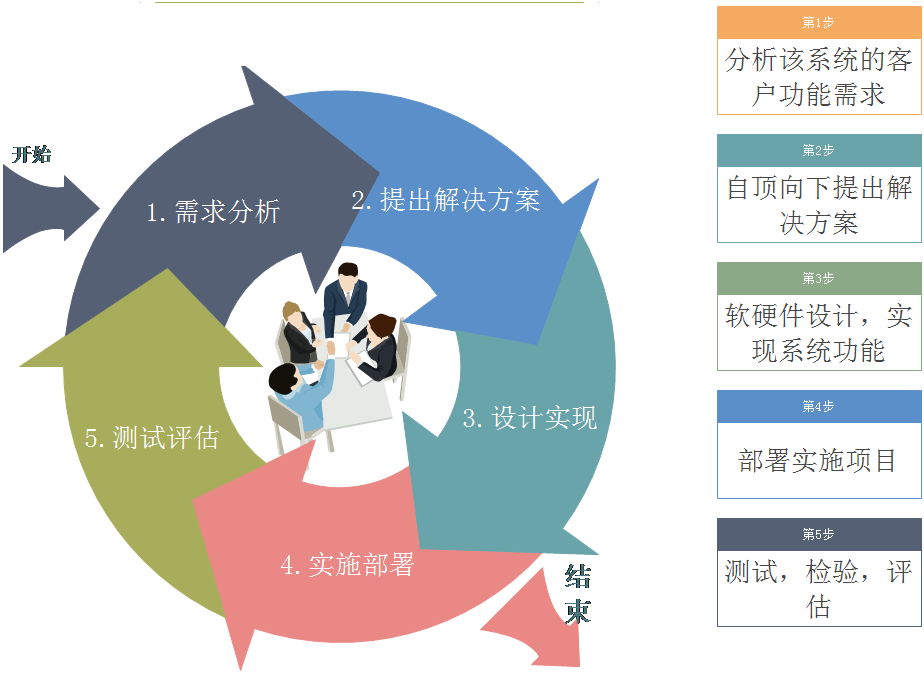


图3-1 手术器械管理系统开发迭代图

需求分析是该手术器械管理系统生命周期的第一阶段。需求分析主要是了解客户的需求，并对此总结出该系统的需要实现的功能来满足客户提出的要求。接着评估开发该系统的风险、成本和代价，最终产生一份系统开发计划。就手术器械管理系统来说，该阶段的主要目标是应用需求确认，项目的可行性研究，分析开发意义和投资与收益。

从实际的应用需求出发确认需求，手术器械管理系统要求可以准确地记录进入无菌区的手术器械和所使用的手术器械，确保没有手术器械残留在病人的体内，并给使用者提供美观而又友好的UI交互界面；可行性方面，可以借助RFID识别技术来实现对手术器械的识别和定位，并利用Qt技术开发应用程序来实现对众多手术器械数据信息的管理，以及为使用者提供方便、简洁和友好的系统操作界面；就目前我们实验室已经具备了独立设计RFID标签的能力，整合我们实验室其他方面的技术可以完成该手术器械管理系统的设计和开发，并将改系统推广应用，为手术提供保障，服务社会。

### 3.1.2 手术器械管理系统系统框图

在仔细分析了手术器械管理系统的应用需求后，我们便可以开始设计该系统的解决方案。由于整个手术器械管理系统涉及到的知识很多，功能也相对比较复杂，无论是该系统的硬件部分还是软件部分都有各自的不同点。例如系统的软件既有与底层硬件接口的程序，还有面向上层应用的面向对象程序；系统硬件部分既存在数字电路的设计还有射频模拟电路的设计。开发复杂系统一般采用自顶向下的开发设计方法，按照上述的分析对于手术器械管理系统我们也应该采用自顶向下的设计方法，先指定整体的系统规划，将整个系统划分为多个模块，从整体结构向下逐步完成各个模块的功能，图3-2展示了手术器械管理系统各个功能模块之间的关系和逻辑关系。

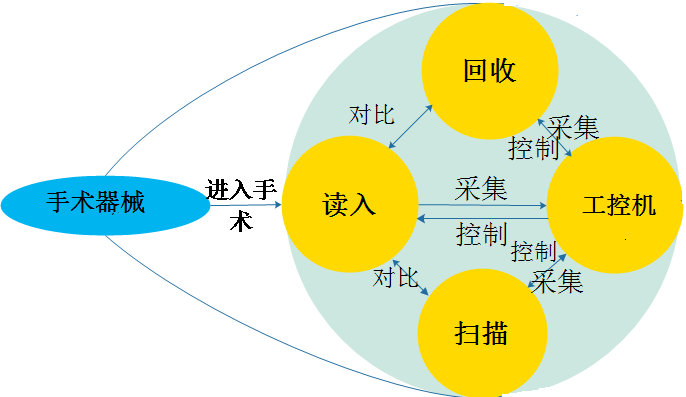


图3-2 自顶向下系统设计规划框图

按照自顶向下结合模块化的设计思想，可以增加系统的通用性和可扩展行，按上图所述，将手术器械管理系统各个功能模块划分为以下：嵌有RFID标签的手术器械，RFID阅读器部分，处理使用过的手术器械回收系统，检查扫描意外遗留在病人体内手术器械的寻找扫描部分，以上部分都是由安装在工控机上Qt开发的上位机软件来协调和控制。

### 3.1.3 手术器械管理系统的实现方案

实现设计阶段是根据提出的手术器械管理系统解决方案的落实和实施步骤。针对上述需求，本文提出了一种基于ISO15693射频标签的RFID手术器械管理系统。系统以射频识别（RFID）技术为核心，结合微处理器（MCU）的高效，利用现场可编程门阵列（FPGA）的精准，凭借Qt友好的图形交互界面，最终实现了一个稳定、高效的智能化手术器械管理系统。通过对手术器械使用和回收情况的智能化管理，达到有使用就有回收的标准，为手术器械的使用和回收管理带来了极大的帮助，有效地避免了手术过程中异物遗留问题的发生。

手术器械管理系统主要由内嵌射频识别标签的手术器械、读卡机系统、工控机等部分组成。阅读器系统由三通道天线、射频前端模块、FPGA编解码模块、MCU控制模块组成，主要完成对射频识别标签的登记录入、回收统计和寻找扫描。天线完成发射和接收磁场信号，射频前端模块实现电信号与磁场信号的相互转换，FPGA编解码模块负责将通信数据根据ISO15693协议进行编码和解码，MCU控制模块控制硬件电路的逻辑动作并与上位机进行通信。工控机端安装了Qt开发的上位机程序，通过RS232串口与读卡机系统通信，显示系统的操作管理界面并完成数据处理和分类。手术器械管理系统整体结构框图3-3如下：



图3-3手术器械管理系统整体结构框图

### 3.2 RFID标签选型以及系统相关要求参数说明

根据第二章中对RFID标签分类的介绍可知，RFID标签可以分为四种：低频系统（30-300KHz）、高频系统（3-30MHz）、超高频系统（300-2.45GHz）和微波系统（2.45GHz以上）。这四种因其工作频率的不同各有各的特性，因此适用的场合也不同。本系统根据相关实际应用需求进行选型。

### 3.2.1 手术器械管理系统的相关参数要求

手术器械管理系统整个设计的使用过程如下：手术开始打开手术管理系统，点击操作界面的登记录入按钮，将手术器械使用前在登记录入天线处扫描录入，器械使用完后扔进回收桶中。手术结束后点击回收按钮，系统自动统计回收桶中的器械。根据系统显示结果判断，如果有未回收的遗漏器械，点击扫描按钮用扫描天线去病人身体部位探测寻找，快速定位遗漏器械的位置。手术器械管理系统适用场景要求能够准确记录进入无菌区的手术器械和所使用的手术器械，确保没有手术器械残留与病人体内。在登记录入天线、回收桶天线和扫描天线处读取RFID时都会出现同时读取多张标签的情况，另外扫描天线需要读取距离在50cm左右这样才能确保整个人身体部位都能被读取，但读取距离又不能太远这样整个手术室内的标签都会被读取。总体上来说手术器械管理系统的参数要求为：1.能够同时读取多张RFID标签及RFID的抗冲突特性；2.读取距离小于70cm，其中扫描天线读取距离大于50cm。

### 3.2.1 手术器械管理系统的RFID标签选型

四种RFID标签一些应用特点如下：

1.低频系统中常用的频率为125KHz和134KHz，不具备抗冲突特性，适合近距离的场合，如门禁、动物耳标等。

2.高频系统中常用的频率为13.56MHz，具备抗冲突，读取距离适中，适合信用卡、资产管理等。

3.超高频系统常用的频率为860MHz-915MHz，具备抗冲突，适用于远距离场合工作，例如车辆跟踪，物流管理等。

4.微波系统中一般工作频率为2.45GHz和5.8GHz，具备抗冲突，比较适合较远距离的场合，高速公路收费系统等

根据上一节分析的手术器械管理系统的相关参数要求：1.能够同时读取多张RFID标签及RFID的抗冲突特性；2.读取距离小于70cm，其中扫描天线读取距离大于50cm。只有高频系统适用，所以选取ISO15693协议的RFID标签，作为手术器械管理系统的工作RFID标签用来标识和跟踪手术器械。

### 3.3天线分布与设计概述

### 3.3.1手术器械管理系统天线设计的基本要求

手术器械管理系统的阅读器和手术器械内的电子标签之间通过天线间的电磁耦合来传输能量和信息。手术器械管理系统阅读器的天线设计质量直接与阅读器和手术器械内标签的电磁耦合效果成正比。手术器械管理系统的天线在整个通信中处于主动地位，阅读器的天线主动产生磁通量，并给手术器械内的电子标签提供电源使其工作。结合上述的情况，手术器械管理系统阅读器的天线应该要实现以下几个要求：

1. 通过天线线圈的电流要尽量的大，这样可以产生最大的磁通量；
2. 良好的功率匹配，最大化的利用阅读器传送的能量；
3. 足够的带宽，能够保证载波信号可以正常发送。

在设计手术器械管理系统时，对手术器械管理系统的阅读器的识别距离也是有一定的要求。对手术器械管理系统天线识别距离产生影响的几个主要因素如下：

1. 手术器械管理系统阅读器天线和手术器械内的RFID标签天线的几何尺寸；
2. 天线的匹配电路设计；
3. 天线与匹配电路的Q值；
4. 手术器械管理系统的阅读器天线的输入功率；
5. 内外工作环境。

### 3.3.2系统天线的结构

手术器械管理系统是电感耦合型的RFID系统，手术器械管理系统阅读器的天线是工作在13.56Mhz下的。现在市场上销售的阅读器都采用线圈型的天线，因为该类型的天线结构简单，成本低等特点。线圈型的天线一般采用圆形或者矩形，性能根据线圈匝数，层数不同而不同。图3-4是两种常见的结构图。

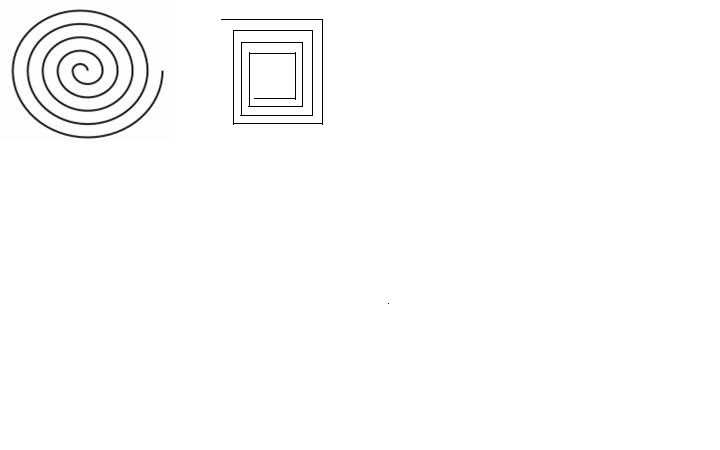


图3-4阅读器天线的主要结构

### 3.3.3 天线具体分布和形状

手术器械管理系统天线总共有三组天线组成：登记录入天线、回收桶天线和扫描天线。 该手术器械管理系统的登记录入天线和扫描天线都采用普通的单组圆形或者方形线圈天线。但是回收通天线为了能够读到回收桶内的任何角度、任何姿势、任何位置的手术器械内的RFID标签，采用特别形式的设计方法，在回收桶的四圈采用三个线圈天线分布，实现对手术器械回收桶的全方位、无死角覆盖，使得回收桶中没有任何盲区，达到实际的需求效果。回收桶天线的效果图如图3-8所示。

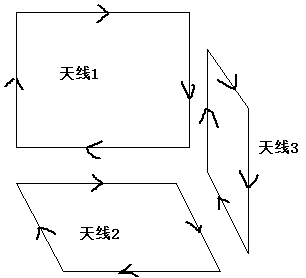


图3-8 回收桶天线效果图

### 3.4工控机概述与应用介绍

### 3.4.1 工控机的概念

工控机即[工业控制计算机](http://baike.baidu.com/view/1115231.htm)，是指对工业生产过程及其机电设备、工艺装备进行测量与控制用的计算机，工控机通俗地说就是专门为工业现场而设计的计算机。工控机如图3-9所示。



图3-9 工控机图

工控机是工业控制自动化和信息化的核心基础设备，起到了不可替代的作用。传统意义上，将用于工业生产过程的测量、控制和管理的计算机统称为工业控制计算机，包括计算机和过程输入与输出通道二部分。但是现在工业控制计算机的概念远远超过工业过程控制这个定义。工业控制计算机是指那些应用在各个发展和建设的领域，具有适应恶劣环境能力并可以长期在这样的环境中长期稳定工作的增强型的PC，所以也称为工控机。

工控机为何收到各行各业欢迎的根本原因是PC具有的开放性。它的硬件和软件资源非常丰富，而且被广大的用户和工程开发人员所熟知，基于PC的工程控制系统，以20%的速度增长，未来的发展趋势不可估计。

### 3.4.2 工控机的特点

工控机是加强版的PC具有更加牢固的外壳和散热性能，在工业恶劣的环境中可靠稳定地运行。在上世纪80年代的初期，美国的AD公司就已经推出了和工控机性能相似的MAC150工控机，接着IBM公司就正式推出了他们的第一款工业PC机IBM7532。由于工控机出众的性能、可靠的运行、价格廉价和丰富的软件资源，使得工控机在工业控制中异军突起，快速得到应用。当前，工控机已经广泛应用在工业现场控制、路桥收费、通信等领域，涉及到现代生活的各个方面。

工控机是依据工业生产的需求和相关特性而对应设计的计算机，应用在工业生产过程中，实现工业控制，自动化调度管理，达到实时、低功耗、安全可靠、减轻人为劳动、改善环境等目的，是自动化仪器仪表的一个重要分支，同时也是计算机行业的重要组成部分。工业生产现场通常是灰尘很多，具有强烈的震动，很强的电磁场干扰等特点，另外工厂是日复一日的工作的，没有任何停止和休息。所以工控机与普通的PC相比具有以下的一些特点：

1. 可靠性高。在整个运行期间不能停机检查和修理，万一发生了故障将会引起事故，所以要求工控机具备很好的可靠性。
2. 实时性好。工控机在生产过程中实时的监测和控制机器，所以工控机必须实时地响应各种控制操作。
3. 环境适应性强。工业控制现场的环境恶劣，严重的电磁干扰，因此要求工控机具有很强的环境适应能力。
4. 系统扩展性好。随着工厂自动化水平的提升，控制的规模也在不停扩大，所以要求工控机具备灵活的可扩展性。
5. 软硬件兼容性。支持多操作系统，多种编程开发语言，多任务操作，充分发挥PC丰富的软、硬件资源。

手术器械管理系统正是在这样的背景下考虑到工控机的可靠性高、实时性好、环境适应性强等优点，决定在系统控制处采用工控机作为控制软件的安装硬件，来实时可靠地控制整个手术器械管理系统。

### 3.5本章小结

本章在第一节根据系统的开发流程从系统需求分析开始介绍了手术器械管理系统需要实现的效果和功能，然后提出了手术器械管理系统的理论整体框图和规划，最后具体介绍了理论框图具体实现的方法和用到的相关技术。第二节介绍了上一节提及手术器械内嵌的RFID标签，从手术器械管理系统的实际需求的参数角度来解释了选择ISO15693RFID标签的原因。第三节介绍了手术器械管理系统天线相关的设计和原理。第四节介绍了手术器械管理系统涉及到的工控机的概念、优点和应用场景，变相地解释了手术器械管理系统选择工控机的原因和优势。

### 第四章 系统硬件部分

上一章介绍了系统的整体结构和各个组成部分，本章着重介绍手术器械管理系统的硬件电路部分。系统的硬件电路主要是RFID阅读器的三个部分：射频前端模块、FPGA编解码模块和MCU逻辑控制模块。MCU逻辑控制模块负责RFID阅读器的工作逻辑控制和将采集的信息上传至上位机程序；FPGA编解码模块依据ISO15693协议实现MCU和射频前端电路之间通信的编码和解码；射频前端电路对发射信号进行调制和放大，对接收信号进行整形滤波和解调。本章准备从阅读器的射频前端电路设计、FPGA编解码电路设计和MCU逻辑控制电路设计三部分来介绍。

### 4.1 射频前端电路设计

射频前段电路分为发射和接收电路两部分，均采用分立元件设计，使得硬件电路更加简单、成本更低而且性能更稳定。

### [4.1.1 发射电路设计](#_Toc421178788)

在发射电路的设计中，主要的实现步骤如下：首先由振荡器产生频率为13.56MHz的载波信号，本系统采用石英晶体振荡器具有很好的稳定性。然后和调制好的基带信号经过调制电路进行ASK调制，再通过功率放大器将信号放大，以此来达到天线对发射功率的要求，总共经过二级功率放大，最后将信号传送到天线，天线将调制的信息通过电磁波的形式向电子标签发射。

手术器械管理系统的前端发射电路的重要组成部分有：通过FPGA编码好的基带信号，13.56MHz的本地振荡器，二级的功率放大器和天线。电路主要的结构图如图4-1所示：

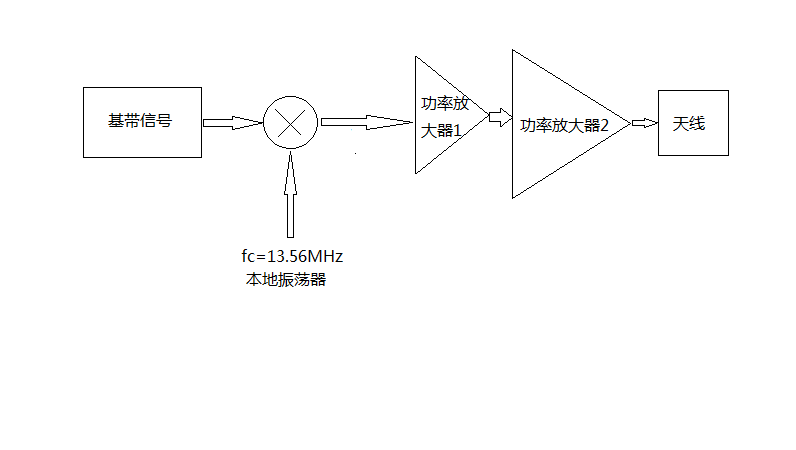


图4-1 发射电路构成图

### [4.1.2 接收电路设计](#_Toc421178788)

上一节介绍了射频前端发射电路，这小节介绍射频前端接收电路，接收电路利用AD8616芯片外接滤波器构成的两级放大滤波电路，将13.56MHz的载波信号滤除，提取出副载波信号，然后利用MAX9142双路比较器对提取的副载波信号进行规整和优化。其中AD8616芯片具有低失调、低噪声、极低的输入偏置电流和高速度的特性，配合滤波器可以完整高效的提取出副载波信号。双路比较器MAX9142具有低功耗、高速度、满摆幅特点可以快速稳定的对信号进行整形和优化。接收电路部分重要电路如图4-2所示。

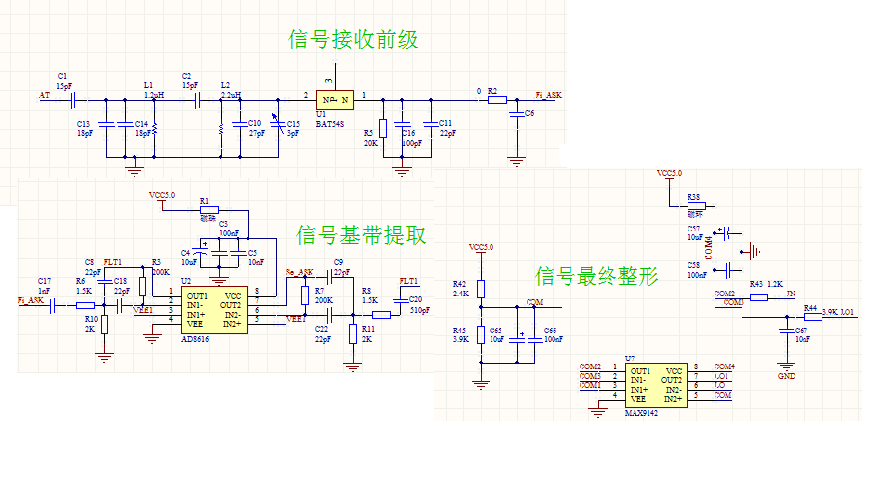


图4-2 接收电路部分电路

### [4.2 FPGA编解码电路设计](#_Toc421178797)

FPGA编解码电路负责将MCU传过来的数据编码成ISO15693协议中的4取1码发往射频前端电路，将接收到的射频前端电路的数据解码成二进制数据，传给MCU。整个系统中FPGA实现编码和解码的功能。

### [4.2.1 FPGA原理简述](#_Toc421178788)

FPGA是现场可编程门阵列的简称，是超大规模集成电路与计算机辅助设计技术发展的结果。这些可编辑元件，用来存储逻辑数据或作为电子开关。常用的可编程元件分为以下四种类型：熔丝型开关、反熔丝型开关、浮栅编程元件、基于SRAM的编程元件。FPGAs 就是可再编程芯片。使用预建的逻辑块和可编程路由资源，你根本不需要拿起面包板或者烙铁就可以使这些芯片具有特定的硬件功能。你可以在软件中开发数位运算系统并将其编译到配置文件或比特流，其包含元件接线的相关信息。此外,当你编译不同的配置电路时，完全可重设的FPGA可立即接受一个全新的设置。高级设计工具的兴起,不仅改变了FPGA编程的规则,而且新技术可以将图形化代码或者C代码转变为数字硬件电路。FPGA的集成度相当高，最多可达到数千万系统门的集成度，可以完成相当复杂的时序和组合逻辑电路的功能，适用于高集成度的高端数字逻辑电路的设计。

FPGA 技术的5大优点 :

1）效能– 依靠硬件并行的优点，FPGA 可以突破依序执行的固定运算，以此可以在每个时钟周期内完成更多的操作，从而超越了数字信号处理器 (DSP) 的计算功能。BDTI著名分析师和基准测试公司,公布的基准显示FPGA可以实现比数字信号处理器处理的一些应用程序多很多倍的处理能力。

2）上市时间– 就上市的时间来说，FPGA 技术具有灵活而且快速成型的能力。使用者不需经过定制ASIC的繁琐冗长的流程设计，即可在硬件电路中测试或验证某个设计想法。您只需要几小时而不是几周，就可以在FPGA设计上实现增量变化和替换。由此可见 FPGA 技术大大缩短了整个设计流程的时间和周期，提高了很大的开发效率。

3）成本– 定制的 ASIC 设计的一次性工程花费，远远超过基于FPGA 硬件解决方案的费用。ASIC技术开始阶段需要大量的投资定制各种硬件功能，用于开发中的各种系统。而可编程芯片的特性，代表低成本的制造，或长时间的装配。由于系统需求功能随时都在变化，因此若与 ASIC巨大的修改费用相比较的话，可以看出FPGA 设计的成本实在很少很少了。

4）可靠性– 而软件工具提供的编程环境,FPGA电路是一个真实的“硬”实施程序。处理器的系统往往涉及多个抽象的层次来帮助安排任务,并在多个进程共享资源。驱动层控制硬件资源，而操作系统则管理内存与处理器带宽。对于任何给定的处理器核心,一次只有一条指令可以执行，但是处理器的系统可以连续地处理一些重要的操作。FPGA 不需使用操作系统，并将产生问题的机率降至最低，以平行执行的方式和准确的硬件致力于每一个任务。

5）长期维护– 如之前所提，FPGA 晶片有即时升级的特性，不需浪费ASIC 重新设计的时间与费用。举例来说，数字通信协议，例如:规则随时间变化而变化， ASIC 的接口可能会产生维护和向下相容的问题。但FPGA 具有可重设性，可以随时根据以后不同的需求进行修改更新。

FPGA 的开发过程是利用EDA开发软件和编程工具对器件进行设计的过程，其中包括：设计准备、设计输入、设计处理、器件编程、功能仿真、时序仿真、器件测试等七个部分。

### [4.2.2 编解码电路的Verilog设计](#_Toc421178788)

编解码逻辑的设计主要是由一个状态机来控制的，此状态机采用三段式状态机，表面上代码虽然结构复杂，但是可以使有限状态机做到同步寄存器输出，消除了组合逻辑输出不稳定和毛刺的问题，更有利于时序路径的分组，利于在FPGA上的综合与布局布线。状态机图如图4-3所示。

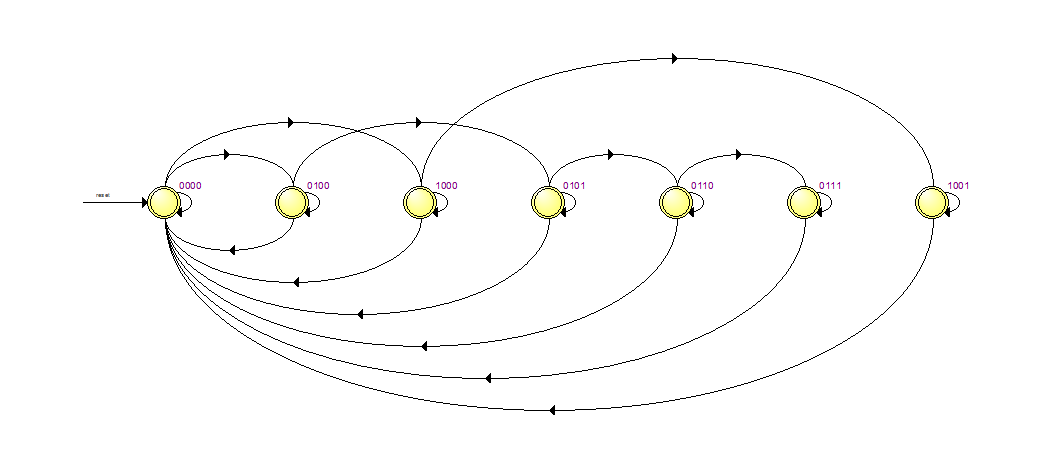


图4-3 编解码电路状态机图

此状态机总共有个七个状态：

STATE\_IDLE 4'b00\_00 初始状态

STATE\_CODER\_BGN\_DLY 4'b01\_00 编码开始状态

STATE\_CODER\_SOF 4'b01\_01 编码SOF状态

STATE\_CODER\_2BIT 4'b01\_10 编码数据状态

STATE\_CODER\_EOF 4'b01\_11 编码EOF状态

STATE\_DECODER\_SOF 4'b10\_00 解码SOF状态

STATE\_DECODER\_4BIT 4'b10\_01 解码数据状态

状态机的跳转Verilog程序如下：

always @(\*)

case(state)

`STATE\_IDLE: state\_next=Coder\_en\_sync ? `STATE\_CODER\_BGN\_DLY : `STATE\_DECODER\_SOF;

`STATE\_CODER\_BGN\_DLY:

state\_next = `STATE\_CODER\_SOF;

`STATE\_CODER\_SOF:

state\_next = `STATE\_CODER\_2BIT;

`STATE\_CODER\_2BIT:

state\_next = `STATE\_CODER\_EOF;

`STATE\_CODER\_EOF:

state\_next = `STATE\_IDLE;

`STATE\_DECODER\_SOF:

state\_next = sof\_found? `STATE\_DECODER\_4BIT : `STATE\_IDLE;

`STATE\_DECODER\_4BIT:

state\_next = `STATE\_IDLE;

default:

state\_next = `STATE\_IDLE;

endcase

因为整个FPGA编解码模块是半双工的，当编码时就不能解码，所以编解码放在一个状态机中，并通过Coder\_en\_sync来做指示。

编码部分主要的Verilog程序如下：

always @(negedge Clk)//编码输出

if(rst\_sync) begin

Code\_out <= #1 1'b1;

end

else if(state\_cnt[6:0] == 7'b111\_1111) begin

case(state)

`STATE\_CODER\_BGN\_DLY: Code\_out <= #1 1'b1;

`STATE\_CODER\_SOF: if(state\_cnt[9:7] == 3'b111 || state\_cnt[9:7] == 3'b010)

Code\_out <= #1 1'b0;

else

Code\_out <= #1 1'b1;

`STATE\_CODER\_2BIT: if(state\_cnt[9:7]== {code\_data\_convert,1'b0})

Code\_out <= #1 1'b0;

else

Code\_out <= #1 1'b1;

`STATE\_CODER\_EOF: if(state\_cnt[9:7] == 3'b001)

Code\_out <= #1 1'b0;

else

Code\_out <= #1 1'b1;

endcase

end

主要实现的功能是将二进制数据编码成ISO15693协议规定的四选一码，并在数据的开头和结尾分别加上SOF和EOF的码子。

解码部分的主要Verilog代码如下：

always @(negedge Clk)

if(rst\_sync) begin

bit\_1st\_half <= #1 1'b0;

bit\_2nd\_half <= #1 1'b0;

end

else if(bit\_en) begin

if(state\_cnt[10]) begin

bit\_1st\_half <= #1 ck32\_cnt\_threshould;

end

else begin

bit\_2nd\_half <= #1 ck32\_cnt\_threshould;

end

end

reg bit\_value ;

reg bit\_err ;

always @(\*)

case({bit\_1st\_half, bit\_2nd\_half})

2'b10: begin

bit\_value = 1'b0;

bit\_err= 1'b0;

end

2'b01: begin

bit\_value = 1'b1;

bit\_err= 1'b0;

end

2'b00,

2'b11: begin

bit\_value = 1'b0;

bit\_err= 1'b1;

end

endcase

主要实现的功能是将前端电路接收的曼切斯特码解码成二进制数据，根据曼切斯特码的特点下降沿为零，上升沿为一的特点，将接收的载波分成两部分检测，先高后低就是0，先低后高就是1，其余11或者00都置bit\_err= 1 报错。

### [4.2.3 编解码电路的软件仿真](#_Toc421178788)

编解码逻辑电路设计的仿真分析分成两部分；编码和解码。此部分借助ModelSim时序仿真软件进行仿真分析，分析Verilog的设计效果。

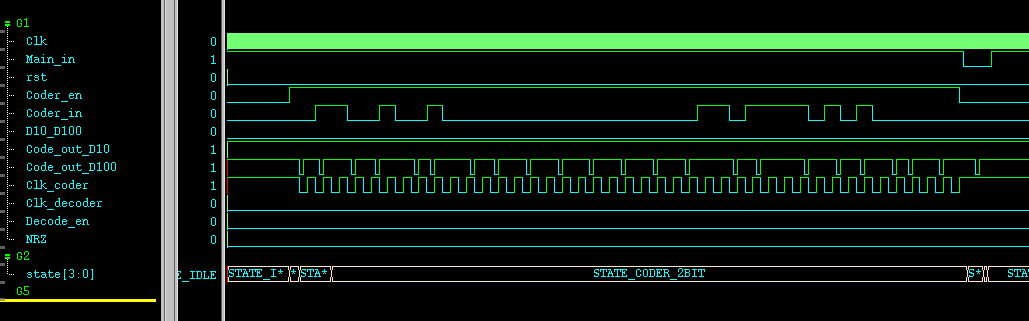


图4-4 编码电路的时序仿真图

编码电路部分的时序仿真如图4-4所示，clk是13.56Mhz的时钟信号；rst是FPGA程序的复位信号；coder\_en是编码使能信号；coder\_in是接收MCU传入数据的端口；D10\_100 是ISO15693中10%和100%两种不同调制，本系统选用的100%调制；Code\_out\_D100是100%调制的编码输出；Clk\_coder是产生的周期时钟信号用于与MCU同步接收MCU 的传入数据；state[3:0]是整个编解码电路的状态机的状态。从上面的时序波形可以清楚的看到当coder\_en被置高后产生的编码使能信号使得状态机先进入STATE\_CODER\_BGN\_DLY状态进入编码开始状态，然后进入STATE\_CODER\_SOF状态在编码数据前加上SOF码，再进入STATE\_CODER\_2BIT状态进行四选一的编码，整个数据的编码都在STATE\_CODER\_2BIT状态中进行，最后进入STATE\_CODER\_EOF状态在数据的末尾加上EOF码，从而完成整个编码过程。最后对比输入的数据与编码后的数据是一致的，而且整个过程状态机的跳转也是正确的，可以判断编码电路的设计时可行的。

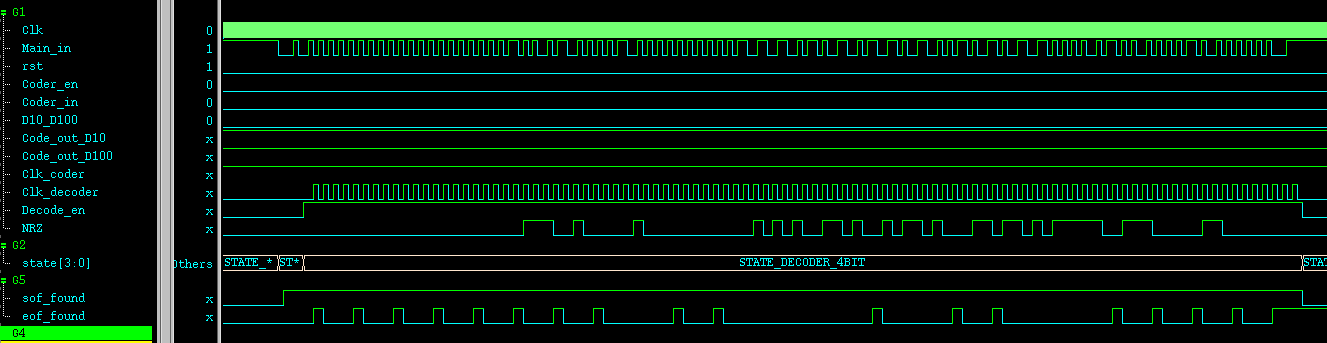


图4-5 解码电路的时序仿真图

编码电路部分的时序仿真图如图4-5所示，clk是13.56Mhz的时钟信号；Main\_in是射频前端电路返回的曼切斯特码数据；rst是FPGA程序的复位信号；Decoder\_en是编码使能信号；NRZ是向MCU发送解码后数据的端口；Clk\_decoder是产生的周期时钟信号用于与MCU同步向MCU 的传送数据；state[3:0]是整个编解码电路的状态机的状态。从面的波形仿真可以看出，当解码电路检测到前端电路的SOF信号后会进入STATE\_DECODER\_SOF状态，并随后进入STATE\_DECODER\_4BIT状态同时将Decoder\_en使能信号置高，通知MCU接收解码后的数据，延迟一段时间产生Clk\_decoder时钟信号来向MCU传递解码后的数据，整个解码过程都在状态机的STATE\_DECODER\_4BIT状态中进行。按照以上分析与一开始的解码设计时一样的，完成了设计任务。

### [4.3 MCU逻辑控制电路设计](#_Toc421178798)

MCU逻辑控制电路主要负责将通过串口接收上位机传来的操作命令，解析后转成对应的ISO15693协议上的通信帧发往FPGA编码；将FPGA解码好的标签返回数据包装成与上位机通信的协议返回到上位机中。整个系统中起到了逻辑控制和判断的作用。

### [4.3.1 MCU原理简述与系统MCU芯片选型](#_Toc421178799)

MCU（Microcontroller Unit）微控制单元，简称为单片机。从字面上就可以知道，“单片”是指一个芯片的意思，“机”是小型计算机系统的意思。该小型计算机系统主要的功能组件包括中央处理单元、程序存储器、随机访问存储器、中断系统、定时器/计数器、外部设备接口模块，以及连接各个功能部件的总线。

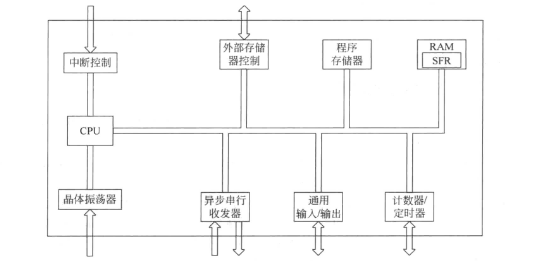


图4-6 MCU内部结构图

1.中央处理单元

中央处理单元及CPU是单片机系统的处理中心，相当于人的大脑，可以实现以下的一些基本功能：

1. 存储器交换信息。通过对存储器的读写操作，来实现CPU与存储器之间的信息交换。
2. 执行逻辑和算术指令。一些常用的基本指令包含：加/减运算、逻辑按位与/或/异或运算、移位运算等。更具体地说，其实CPU是由更小的一些功能模块组成，这些重要的模块包含：程序计数器、指令译码单元、算术逻辑单元和控制寄存器和状态寄存器组。

2.程序存储器

程序存储器是用来存储单片机将要执行的程序代码。一般程序存储器是非易失性的，一旦程序固化到存储器上，就会一直保存着，除非重新对存储器进行固化。即使单片机掉电后，下一次启动，原本的程序也还会存储字程序存储器内。一般单片机程序存储器都会采用Flash存储工艺，支持多次擦写固化程序。

3.随机访问存储器

随机访问存储器又称RAM，用来保存程序中所需要用到的一些数据。

4.中断系统

中断系统是CPU用来处理一些级别更高的事件机制。当CPU正在正常地执行当前的程序，突然外部设备向CPU发送了一个紧急的事件，通俗地说就是所谓的中断请求，当CPU开启了中断允许请求，就会打断当前的程序，优先去处理当前的中断事件，用于处理紧急事件的程序称为中断服务程序。

5.定时器/计数器

定时器/计数器对于单片机来说是最基本的功能单元。可以通过这些单元，对不同的事件进行同步。最简单的例子就是当定时器达到设定的初值时，便会产生一个定时器中断信号，利用这个中断信号便可以同步其他的设备。

6.外部设备接口模块

外部接口因不同的单片机厂商而不一样，但基本上都提供了通用IO(general purpose input and output，GPIO)和RS-232接口等。

7.总线

总线是一组相关逻辑信号的集合。目前计算机大多采用总线架构，总线一般包括控制总线、地址总线和数据总线。单片机系统总线分为内部总线和外部总线，内部总线连接芯片内的各个模块单元，外部总线连接单片机外的一些外围设备。

参照单片机的选型原则，考虑成本价格、工作环境、封装形式、工作频率、程序存储器容量、片内RAM容量等参数，结合实际的应用需求选择一款合适的单片机很重要。本系统根据实际手术器械管理系统的阅读器工作需求，从市场中挑选了STC的一款性价比较高的IAP15W4K61S4单片机，具有30MHz的最大处理器频率，保证其能够快速的完成ISO15693协议中的抗冲突算法，实现快速准确识别多张标签的要求。

### [4.3.2 MCU电路设计](#_Toc421178799)

单片机通过MAX232串口芯片实现与工控机串口通信。在与FPGA编解码模块通信时利用单片机自带的外部中断机制，将单片机中断设置为下降沿触发，FPGA设计成上升沿触发，实现一个方波信号传递一位数据的快速稳定的通信方法。单片机采用P1.2、P1.3、P1.4、P1.5、P3.5、P3.2、P3.3这7个端口与FPGA完成通信，其中P3.2和P3.3为单片机的外部中断0和外部中断1引脚。单片机考虑到稳定性和可靠性采用外部24MHz晶振。各个引脚的意义如下：

SW1——SW4：用于切换三组天线的标志

coder\_en: 给FPGA的编码使能信号

code\_out: 给FPGA发需要编码的数据口

FPGA\_rst: FPGA复位信号

decode\_en:FPGA给MCU的解码使能信号

NRZ: FPGA将解码后的数据返回至MCU

clk\_coder: FPGA编码时，MCU向FPGA发数据的同步信号

clk\_decoder: FPGA解码时往MCU传数据的同步信号

MCU电路图如图4-7所示。

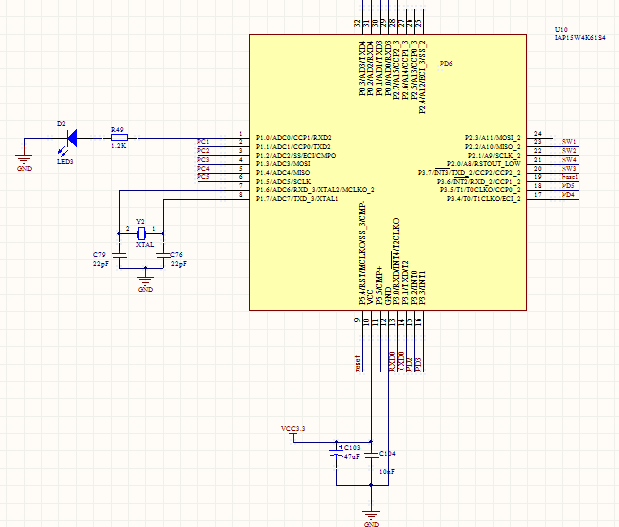


图4-7 MCU电路图

### 4.3.3 MCU外围电路设计

MCU外围电路主要是串口电路。手术器械管理系统的阅读器需要与上位机软件进行数据通信，采用最常用的串口作为通信接口，具有稳定、快速、设计简单等优点。串口电路采用MAX232芯片作为收发器，其专用低压差发送器输出级在供电电压为3.0v至5.5v之间时，利用双电荷泵实现真正的RS-232性能，电荷泵起高压仅需四个0.1uf的外部小尺寸电荷泵电容。MAX232芯片在最差的条件下，依然可以保证有120kps 的数据传输率，同时输出电平保持为RS-232的输出电平。

MAX232具有二路接收和二路驱动器，可以提供1µA关断模式，有效地降低功耗，在关断的模式下接收器依然可以保持有效状态，对外部设备进行检测，此期间仅消耗1µA电源电流，给MCU和上位机之间提供稳定的数据传输通道。串口电路如图4-8所示。

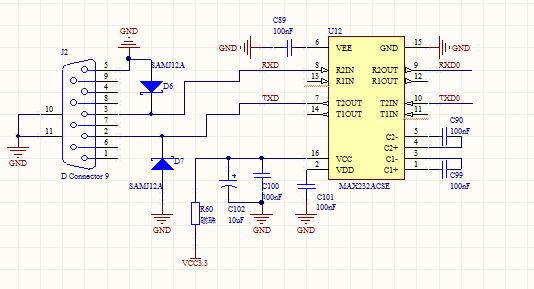


图4-8 串口电路

### 4.4 [MCU与FPGA编解码电路接口设计](#_Toc421178803)

单片机采用P1.2、P1.3、P1.4、P1.5、P3.5、P3.2、P3.3这7个端口与FPGA完成通信，其中P3.2和P3.3为单片机的外部中断0和外部中断1引脚，并设成下降沿触发。

P1.2 coder\_en: 给FPGA的编码使能信号

P1.3 code\_out: 给FPGA发需要编码的数据口

P1.5 FPGA\_rst: FPGA复位信号

P3.5 decode\_en:FPGA给MCU的解码使能信号

P1.4 NRZ: FPGA将解码后的数据返回至MCU

P3.2 clk\_coder: FPGA编码时，MCU向FPGA发数据的同步信号

P3.3 clk\_decoder: FPGA解码时往MCU传数据的同步信号

编码MCU向FPGA传递数据的过程：



图 4-9 MCU向FPGA传递数据

单片机向FPGA发送FPGA\_rst复位信号，接着将coder\_en信号置高表示FPGA可以开始进行编码，FPGA产生clk\_coder周期信号，单片机检测到clk\_coder信号的下降沿时向code\_out口传送数据，FPGA在clk\_coder信号的上升沿时接收code\_out口的数据，一个clk\_coder信号传递1bit数据。MCU数据传递结束后将coder\_en置低，数据传递结束。

解码FPGA向MCU传递数据的过程：



图 4-10 FPGA向MCU传递数据

当MCU向FPGA数据传完之后，向FPGA发送一个FPGA\_rst复位信号，FPGA开始对射频前端返回的数据进行SFO帧开头检测，当检测到SFO后将decode\_en使能信号置高，告诉MCU准备接收解码后的数据，并同时产生clk\_deocde的周期信号用来传递解调后的数据，在clk\_deocde的上升沿时FPGA给NRZ口置数，在clk\_deocde的下降沿时MCU接收NRZ的数据，一个clk\_coder信号传递1bit数据。直到FPGA检测到EOF帧结束标志，将decode\_en置低，结束数据传输。

### 4.5本章小结

本章着重介绍了手术器械管理系统的硬件电路部分的设计，第一节大体分析了射频前端发射和接收部分的设计思想和思路；第二节着重分析了编解码电路FPGA的开发设计的过程和Verilog程序，并根据编码和解码部分的时序波形仿真图进行了分析介绍；第三章重点介绍了MCU理论和相应的本系统的电路设计和外围电路的组成；第四节分析了MCU电路和FPGA电路两部分的数据通信方式和设计思想。总体来说本章对手术器械管理系统的硬件介绍是非常全面到位的。

### 第五章 系统软件部分

系统软件设计采用模块化的思想设计程序，设计时注意各个模块之间的数据传输方式。MCU控制软件是基于单片机IAP15W4K61S4采用C语言设计的，多模块结合使得程序逻辑结构清晰，代码量少。上位机程序设计采用C++语言基于Qt Creator图形界面应用程序开发软件开发的，该软件具有成熟的库函数，移植性强，运行环境多样等特点，支持Linux，Windows等操作系统可以快速简单的开发出友好简洁的交互界面。

### 5.1 MCU软件设计

### 5.1.1 STC单片机软件开发介绍

STC单片机采用C51内核作为单片机CPU内核的，支持汇编语言和C语言两种开发模式。C语言程序并不依赖于机器本身的硬件系统，一般可以不需要任何修改就可以从不同的系统中的移植并直接使用。C语言提供了很多数学函数并支持浮点运算，开发效率高，可极大地缩短开放时间，增加程序可读性和可维护性。单片机的C语言编程与用汇编编程相比，有以下几个优点：

1. 可以直接使用C语言进行单片机开发，无需对单片机的指令系统有过多的了解。
2. 一些细节上的寄存器分配，存储器寻址和数据类型等都由编译器自动处理。
3. 程序可以通过结构化设计，分成不同的函数。
4. 可以直接引用库中的一些标准程序，用来处理数据，使得开发更加快捷方便。

根据以上C语言开发的优点分析，系统开发采用C语言开发。

STC单片机软件开发的流程如下：

1.明确软件需要实现的功能

对于程序设计人员来说，需要完成下面的工作:

（1）了解所提供硬件的性能。

（2）了解STC单片机的软件开发环境Keil uVision。

（3）设计出软件的数据流图和程序流图。

（4）明确程序的概要设计和详细设计方案。

2. C代码的编写

通过Keil uVision软件提供的C语言编辑器进行代码编写设计。

3.将设计完的C语言代码通过编译器进行编译

4.通过库管理器生成库文件

在开发设计中允许调用库管理器，创建一个库，并在库中添加或者删除相应的文件。

5.使用链接器生成绝对目标文件

使用Keil uVision提供的链接器/定位器，将之前C编译器编译创建的目标模块链接在一起生成一个绝对目标文件。

6.使用转换器将绝对目标文件转换成十六进制文件

使用Keil uVivion开发工具提供转换器，将上一步生成的绝对目标文件转换为Intel格式的十六进制.HEX文件。

7.使用STC-ISP软件将十六进制文件固化到STC单片机的程序存储器中

当硬件和软件的整个开发流程都结束以后，使用STC的STC-ISP软件将HEX文件通过串口线下载到对应型号的单片机中去。

### 5.1.2 MCU软件框架设计

首先，对各个模块初始化和单片机IAP15W4K61S4引脚配置。当检测到串口中断后，进入串口中断服务程序，根据事先设定的通信协议识别帧头，开始接收数据。接收完成后进行循环冗余检查（CRC）校验，校验正确则将接收完成标志置为正。当主程序检测到接收完成标志位被置为正，对接收的数据进行解析判断，根据接收到的命令向FPGA中发送对应的命令帧。随后接收FPGA中返回的数据帧，进行CRC校验，通过后依据串口通信协议对接收数据进行打包，最后通过串口向上位机返回数据。一次通信完成后，程序回到主程序接收完成标志的检测函数，进行循环检测直到检测到下一次完成标志，进行上面同样的步骤。MCU程序具体流程如图5-1所示。



图5-1 MCU程序流程图

### 5.1.3 MCU核心代码的实现

上一小节介绍了整个MCU程序设计的流程图，整个程序主要由main主程序，myUART串口程序，command命令处理程序和与FPGA通信的程序四个主要部分。

main主函数程序进行一些端口和寄存器的初始化，并包含一个while（）死循环，用来让单片机一直处于进行工作状态。为了防止程序跑偏，或者在某个环节卡死，为主函数设计一个4s的看门狗程序。主函数的程序如下：

void main()

{

P0M0=0x00; P0M1=0x00;

P1M0=0x00; P1M1=0x00;

P2M0=0x00; P2M1=0x00;

P3M0=0x00; P3M1=0x00;

P5M0=0x00; P5M1=0x00; //初始化IO口为准双向口模式

LED\_PORT=0;

SW1\_PORT=1;

SW2\_PORT=0;

SW3\_PORT=0;

SW4\_PORT=0;

SW5\_PORT=1;

BELL\_PORT=0;

POWER\_PORT=0;

TEST1\_PORT=0;

Uart0\_Init() ; //串口初始化

FPGA\_Initial(); //FPGA初始化

Timer1\_init() ;

watchdog\_init(); //初始化看门狗

EA=1; //全局中断使能

while(1)

{

command();

LED\_cn++;

if(LED\_cn>0xb0000)

{

Led\_on(40);

LED\_cn=0;

}

WDT\_CONTR=0x37; //喂看门狗

}

}

myUART串口程序主要由接收和发射电路组成，串口使用单片机串口1的工作模式1，8位UART，波特率可变，定时器2作为波特率发生器，波特率为115200。具体的寄存器参数设置如下：

SCON = 0x50; //8位数据,可变波特率

AUXR |= 0x01; //串口1选择定时器2为波特率发生器

AUXR |= 0x04; //定时器2时钟为Fosc,即1T

T2L = 0xCC; //设定定时初值

T2H = 0xFF; //设定定时初值

AUXR |= 0x10; //启动定时器2

command命令处理程序在主函数的while循环函数内，通过一个全局变量Rev\_flag来判断是否完整的接收了一帧串口数据，只有当串口函数接收一帧完整的数据才会将Rev\_flag置为1，这样才会进入command命令处理程序对这一帧数据进行相应的处理。

if(Rev\_flag) //表示接收上位机命令结束，开始处理命令；或等待状态

{

Rev\_flag=0;

REN=0; //禁止串口1接收

message\_box[4]=UART0\_RX\_BUF[1];

//天线地址（通道地址）除去0x11、0X22、0x33头的第2位表示通道

POWER\_PORT=0;//RF关闭

delay\_ms(1);

POWER\_PORT=1;//RF打开

delay\_ms(2);

switch(UART0\_RX\_BUF[2])

{

case 0x01:

anticollision();

break;

case 0x02:

Reset\_Select\_quiet(); //静默 没写

break;

.

.

default:

message\_box[6]=0x90; //不支持指令

response();

break;

}

}

POWER\_PORT=0;//RF关闭

REN=1; //打开串口1

}

MCU与FPGA通信的程序，主要是通过外部中断0和外部中断1来模拟SPI 的传输方式实现的2个硬件之间同步传输数据，不直接使用SPI这样的好处是使得2个硬件之间的数据传递更加简洁，稳定，快速，不受约束（通信过程请参考第四章系统硬件部分的4.4小节MCU与FPGA编解码电路接口设计）。外部中断0和中断1都设置为下降沿触发， 寄存器设置为：IT1=1; IT0=1; 。外部中断0负责向FPGA发送数据，每一个中断信号发送1bit，具体代码如下：

void INT0() interrupt 0

{

if((send\_data & 0x01)==1)

coder\_out\_PORT=1;

else

coder\_out\_PORT=0;

send\_data=send\_data>>1;

int\_0++;

}

外部中断1负责接收FPGA发送的数据，每一个中断信号接收1bit，具体代码如下：

void INT1() interrupt 2

{

static unsigned char redata=0;

int\_1++;

cn1=0;

cn2++;

if(NRZ\_PORT)

redata+=0x80; //从低位开始接收

if(int\_1==8)

{

int\_1=0;

RXD[length\_R++]=redata;//数据丢到RXD里面暂时存放

redata=0;

}

redata=redata>>1;

}

### 5.1.4 标签抗冲突代码实现

抗冲突是ISO15693协议标签的重要功能，及能同时读取场内多张射频标签的特性。其实现的原理是在标签端和阅读器端都进行特殊处理，本小节将着重介绍阅读器端的抗冲突实现过程。

ISO15693协议中介绍的16slot的抗冲突如图5-2所示的步骤如下：

1. VCD发送一次Inventory请求，在一帧请求中，由EOF结束。slots的数量是16。
2. VICC 1在 slot 0发送其响应。假设它是唯一一个返回响应的VICC，不会发生冲突，VCD将记下它的UID。
3. VCD发送一个EOF，意思是接通到下一slot。
4. 在 slot 1，假设两个 VICC 2和3同时返回它们的响应，产生一次冲突。
5. VCD发送一个 EOF，意思是接通至下一个 slot。
6. 在 slot 2，没有VICC传输响应。VCD检测不到VICC的SOF，从而发送一个EOF，接通至下一个slot。

注：中断防冲突序列的决定权在 VCD。它可以持续发送 EOF，直到遍历至 slot 15，然后发送请求给 VICC 1。其中VICC：标签，VCD：阅读器。

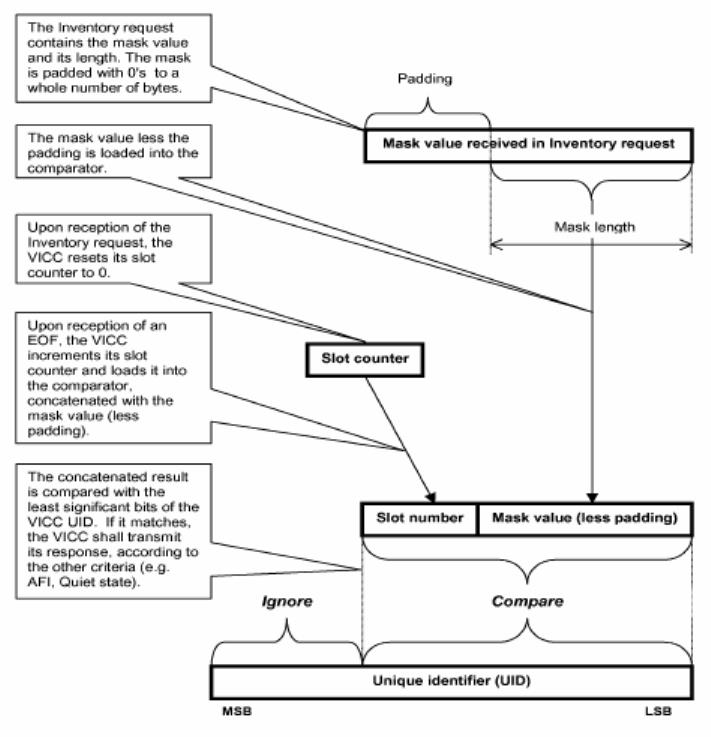


图5-2 ISO15693协议中16slot的抗冲突步骤

根据协议的定义和实际的手术器械管理系统的情况设计了一套简化的但效率高的抗冲突实现方法。结合协议中的抗冲突方法，第一步从slot0开始发送Mask值为0x00（因不足8位补齐8位），Mask长度为4，如果没有产生冲突（如果有返回接收返回的标签UID），并发送EOF接通到下一个slot（Mask值为0x01，Mask长度为4），这样一直下去，直到0x0F结束整个抗冲突过程。这个过程中一旦产生冲突在此时Mask值变为0x0Y（Y为产生冲突的slot，Y值：0-15），Mask长度变为8，及在原来的基础上加长4位进行比较，如果还产生冲突在往上加4位，Mask值为0x000Y，Mask值为12，直到没有冲突产生，但最高Mask长度可以加到16位（考虑到抗冲突过程的时间）。这种抗冲突的方式因为减省了与标签的一部分通信，可以在标签不是很多的情况快速读取（因为手术器械管理系统要求一次读取的标签数在50张左右），这样可以大大提高抗冲突的效率，从而提高了对标签的识别效率。

### 5.2 上位机软件设计

上位机程序是整个系统的开始点和结束点，是系统与人的交互点，因此界面不仅要简洁明了，更要保证操作快捷方便。利用Qt特有的信号与槽的机制使得系统操作者只需点击界面按钮，便可触发对应的操作，完成数据的获取。

### 5.2.1 Qt技术简介

Qt是一种支持跨平台的UI应用程序开发框架，只需一次开发，无需重新编写源代码就可以实现在不同的操作系统上部署该应用程序。Qt软件的前身是创立于1994年的奇趣科技，在2008年奇趣科技被诺基亚收购，作为其跨平台开放战略的重要部分。Qt Creator 是用于Qt技术开发的一款IDE软件，既可以单独使用，也能够与Qt库和其他开发工具组合使用。其中包括：高级C++代码编写器、项目和生成管理工具、集成的上下文相关的帮助系统、图形调试器、代码管理和浏览工具。

Qt Creator 具有两个很方便开发的特点：

1. Qt中UI开发可以直接通过拖拽现有的部件库元件置对应的开发界面上，在通过简单的布局设计就可以快速完成一个美观，快捷的UI界面开发。
2. 信号与槽的机制，可以完成对象之间的协同操作。简单的说，信号与槽都是函数，比如单机窗口上的一个按钮想要弹出一个对话框，那么就可以将这个按钮的单击click（）信号与定义的槽函数关联起来，当点击这个按钮时发出click（）信号，当前界面检测到该按钮的click（）信号后就会自动调用定义的槽函数来创建一个对话框。这种机制可以快速、方便的将界面的操作和后台执行程序绑定，实现相应的功能。

### 5.2.2上位机软件框架设计

整个上位机的业务需求：手术开始打开手术管理系统，点击操作界面的登记录入（COUNT IN）按钮，将手术器械使用前在登记录入处扫描录入，器械使用完后扔进回收桶中。手术结束后点击回收（COUNT OUT）按钮，系统自动统计回收桶中的器械。根据显示结果判断，如果有未回收的遗漏器械，且不在可视范围内，点击扫描（SCAN）按钮用扫描器去病人身体部位探测寻找，快速定位遗漏器械的位置。最后点击详情（DETAIIED）按钮进入详细数据界面保存数据，以便日后查询手术记录查看手术器械使用情况。

根据业务需求整个系统的上位机程序框架设计如下图5-3所示：



图5-3上位机程序框架图

### 5.2.3上位机操作界面的设计

用户UI模块就是系统数据显示与用户交互的界面，采用Qt自带的控件进行拖拽式布局设计，快速完成交互界面的开发。其中用户的点击按钮采用Qt的button控件结合信号与槽的机制，实现按钮点击后的后台程序触发。系统数据显示采用TableView控件，结合它的视图模型可以简单快速的显示数据库表的数据。UI界面设计图如图5-4所示。

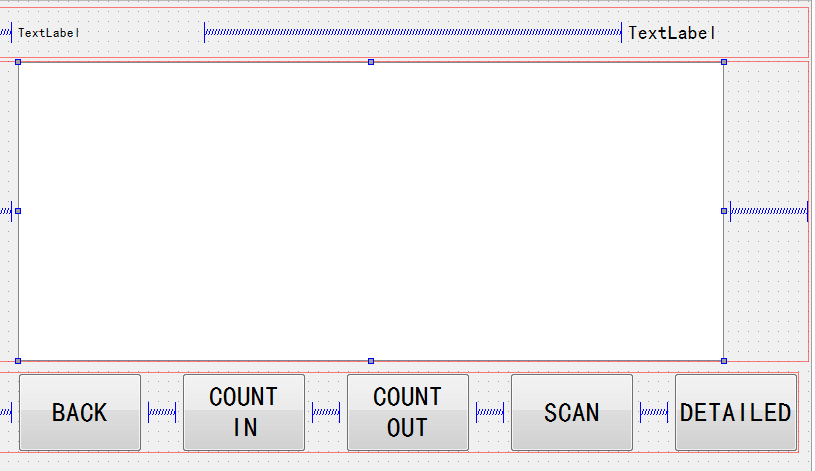


图5-4 UI界面设计图

上图中控件通过QLayout类提供的布局管理器，进行了整理布局，使得界面更加整洁美观，界面按水平布局为主，大体分成了三部分。具体如图5-5所示：

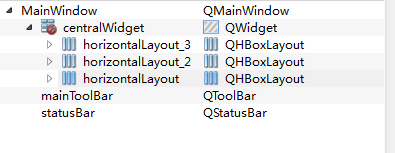


图5-5 UI界面分布图

### 5.2.4后台控制程序的实现

后台控制程序主要分成两个部分：串口通信程序和数据处理程序。

串口通信程序主要负责与手术器械管理系统的阅读器进行通信，将操作命令发送到系统阅读器，并接收阅读器返回的数据。由于系统要确保所接收阅读器数据的安全性和实时性；在安全性方面，采用CRC16校验确保每一帧数据的正确传输（此部分将在下一章节着重介绍）；在实时性方面，采用经典的“生产者消费者模型”的多线程机制来确保串口接收到的数据可以得到实时性处理。

Qt没有现成的串口通信类库调用，借助第三方的串口通信类qextserialport实现，该类分为Win\_QextSerialPort和Posix\_QextSerialPort分别支持Windows和Linux两种环境。在项目管理文件.Pro文件中进行如下配置：

unix:{

DEFINES = \_TTY\_POSIX\_

HEADERS += qextserialport/posix\_qextserialport.h

SOURCES += qextserialport/posix\_qextserialport.cpp

}

win32 {

DEFINES = \_TTY\_WIN\_

HEADERS += qextserialport/win\_qextserialport.h

SOURCES += qextserialport/win\_qextserialport.cpp

}

然后在Qt框架下来实现串口的发送数据就比较简单了，首先先将需要发送的一帧数据按照发送数据的格式整理好，然后通过HexStrToByteArray(QString str)函数将String型的字符串转化成十六进制的码子，再通过引用的串口库函数write()函数将转化好的十六进制数据帧发送出去。字符串转成十六进制数据的HexStrToByteArray(QString str)函数代码如下：

QByteArray HexStrToByteArray(QString str)

{

QByteArray senddata;//字符转十六进制后存放在senddata变量中

bool ok; char data;

QStringList list;

list = str.split(" ");//将需要转换的字符以空格划分开

for(int i = 0; i < list.count(); i++)

{ if(list.at(i) == " ") continue;

if(list.at(i).isEmpty()) continue;

data = (char)list.at(i).toInt(&ok, 16);//将分割开字符格式化成十六进制

if(!ok){QMessageBox::information(this, tr("提示"), tr("输入数据有误！"),

QMessageBox::Ok); break;}

senddata.append(data);

}

return senddata;

}

整个串口的发送程序只需要在主函数中编写两行代码就可以将字符串数据帧转换成的十六进制数据帧并发送出去发送，非常方便，快捷。代码如下：

data =assistance::HexStrToByteArray(writeData);

int count = port->write(data, data.length());

但串口接收部分的实现，就相对比较的复杂了。为了提高串口接收程序的工作效率和可扩展性，串口接收的一帧数据需要经过从串口接收到的数据需要经过物理层、缓冲层、校验层、通信协议层和数据处理层，五个层次的处理，最后才能被该系统接收下来存入数据库中。这五层之间的关系和功能如图5-6所示。

物理接收层

数据校验层

环形缓冲层

通信协议层

数据处理层

确保接收一帧完整的数据

经CRC校验，确保接收的数据正确

多线程中的“生产者消费者”模型，实时性

解析出响应帧中纱布使用的信息和数据

经过运算处理后，信息存储于数据库

图5-6 Qt串口接收数据的流程图

串口接收数据过程中，经常会出现接收数据的不均匀传送，传送期间存在空白或者时间间隔，这给数据处理提出了很高的挑战。

如果直接使用串口的数据接收函数来接收MCU发送的数据，则可能会因为上述的情况导致数据接收不完整，或者混乱。为此在设计串口接收程序时，可以在两批数据之间间隔时间T后再接收。当接收到第一批串口数据后等待T时间，如果有数据传入再接收第二批数据，如此循环直到没有数据传入后跳出循环。时间T的选择一般是接收一帧数据时最大时间间隔的1.5倍，如果选择过大可能会将几帧数据合并为一帧，选择过小可能会将一帧数据分成几段。只有选择合适的时间间隔T才可以保证数据的完整性和可靠性，提高串口接收数据的效率。串口接收程序的流程图如图5-7所示。



图5-7 系统串口接收程序的流程图

void ReceiveThread::run()//启动接收数据的线程

{ int count=0;

Forever

{

msleep(3);

mutex.lock();//加线程锁，确保读取操作的原子性

count = comport->bytesAvailable();//获取当前可读取的数据长度

mutex.unlock();//线程解锁

if (0 < count)

{lsloop:msleep(5);//等待5us后

if(count<comport->bytesAvailable())//如果判断串口是否还有数据到来

{count = comport->bytesAvailable();//获取可读的字节数

goto lsloop;}//跳转到lsloop等待5us后接着判断

emit newDataInPortThread(count);

}

}

}

当上面程序接收完一帧完整的数据后，会发射一个信号newDataInPortThread(count)，该信号中包含了接收到数据的帧长度，然后触发槽函数newDataInPortSlot(int count)将串口缓存区中数据帧取出来并再次发射一个newDataInPortSignal(data,readcount)信号，将接收的数据传到数据处理部分。接收数据的槽函数的代码如下：

void newDataInPortSlot(int count)

{char data[size];

int readcount = readData(data, count);//读取该帧数据

emit newDataInPortSignal(data,readcount);}

在系统中通过信号与槽的机制通过

connect(port,SIGNAL(newDataInPortSignal(const unsigned char \*, const int)),this, SLOT(receiveMsg(const unsigned char \*, const int)));

将接收完成信号newDataInPortSignal (data,readcount) 与主程序的void MainWindow::receiveMsg(const unsigned char \*data, const int size)槽函数绑定，当检测到newDataInPortSignal（）信号就执行receiveMsg（）函数，从而执行数据处理程序。

数据处理程序主要是对接收到的数据进行校验和解析，提取出MCU返回的一些执行动作的读取结果，本系统主要是切换天线和读取各个天线读到的RFID标签的UID。具体的代码很多不再叙述分析。

### 5.3 上位机与MCU的通信协议设计

本协议是系统上位机与阅读器中MCU进行通信的协议规范。在手术器械管理系统中，设计的通信协议要根据ISO15693协议，实现一些通用的系统的控制命令，负责设置阅读器的工作参数、控制阅读器的射频和输入输出，以及天线切换等。另外还需实现一些普通的ISO15693 标签操作命令集，比如读取标签的UID、数据块外等。整个通信协议是上位机与阅读器之间可靠通信规范约定和保障。通信协议分为请求帧和响应帧两部分。

请求帧由通信过程中处于主动地位的设备发出，本系统处于主动地位的是上位机及工控机，两个请求数据帧间隔时间应该不小于50毫秒。表5-1为请求帧的格式：

表5-1 **请求帧**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *帧头* | *帧长* | *通道地址* | *命令码* | *参数* | *CRC* |
| 3bytes | 1bytes | 1bytes | 1bytes | nbytes | 2bytes |

帧头是指示一帧数据的开始，值为0x112233，用于接收方判别数据的开头。长度是从通道地址开始到CRC结束，通道地址用于指示不同的天线工作，命令码指示不同的通信目的，保留默认为0x00。参数是命令携带的数据，长度和数据随不同的命令而变。CRC为从帧长度开始到参数区结束的CRC校验码。

响应帧由通信过程中处于被动地位的设备响应返回，本系统处于被动地位的是系统阅读器，响应帧应该会在阅读器收到请求帧数据后的200 毫秒内返回。表 5-2 为响应帧的格式。命令码指示不同的通信行为。参数包含执行响应命令后返回的读取数据，因各个命令的不同返回数据不同。

表 5-2 **响应帧**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *帧头* | *帧长* | *通道地址* | *命令码* | *参数* | *CRC* |
| 3bytes | 1bytes | 2bytes | 1bytes | nbytes | 2bytes |

### 5.4 数据库的设计

本系统在上位机中对通过串口接收到的返回数据处理，采用了数据库技术来存储记录读取到的数据，然后经过一些数据库的条件查询语句来筛选和分类数据，形成业务需求所要得到的手术器械使用结果的信息，然后通过UI界面中TableView控件来显示。

### 5.4.1数据库相关技术简介

数据库是一种按照数据结构来组织、存储和管理数据的建立在计算机存储设备上的仓库。数据库是长期储存在计算机内的有组织的、可共享得数据集合。

数据库具有以下特点：

1. 实现数据共享。所有用户可同时用各种方式通过接口读写数据库，提供数据共享。
2. 减少数据的冗余度。数据库可以实现数据的共享，这样避免各个用户各自建立数据文件，减少了大量重复的数据。
3. 数据的独立性。数据的独立性是指数据库的逻辑结构和应用程序之间相互独立，互补不影响。
4. 数据实现集中控制。不同的用户或者同一用户在不同的处理中，文件之间是毫不相干的，通过数据模型表示各种数据之间的联系。
5. 数据的一致性和可维护性。防止数据丢失和越权使用，保证数据的正确性和可靠性，可以进行数据的恢复和还原。

SQL语句是数据库的通用语言，数据库通过SQL语句来实现方便地查询、定义、操作和控制数据库中的数据。下面简单的介绍一下本系统中用到的一些语句：

1. SELECT语句

SELECT 列名称 FROM 表名称

1. INSERT语句

INSERT INTO 表名称 VALUES (值1, 值2,....)

1. UPDATE语句

UPDATE 表名称 SET 列名称 = 新值 WHERE 列名称 = 某值

1. DELETE语句

DELETE FROM 表名称 WHERE 列名称 = 值

数据库有很多比如MySQL、SQL Server、Oracle、SQLite等等，本系统综合考虑后决定使用SQLite数据，因为SQLite代码量特别小，是一个轻量级的数据库系统，处理速度快，占用资源少。考虑手术器械管理系统工控机端的数据处理量基本很小，无需使用大型的数据库系统这样显得很笨重占用资源，另外SQLite数据库支持跨平台，可以在上位机因系统环境变化时无需多大改动就能使用。

### 5.4.2数据库结构设计

数据库由2张表组成：⑴Device表。记录所有器械的使用和回收情况；⑵Type表。记录所有器械对应类型的使用和回收情况。Device表与Type表之间具有多对一的关联关系。数据库表关系图如图5-8所示。



图5-8 数据库表关系图

接收数据的处理流程如下：①手术开始后系统自动创建一张与手术患者有关的Device表。②登记天线读取到手术器械内标签的UID，先在表中查询是否已存在此UID记录。如果没有则创建ID为此UID的一条行记录，并将登记天线值写为true，回收天线和扫描天线值写为false，器械类型根据读取到标签的AFI判别。如果存在此UID记录就忽略。③回收天线读取到UID，在表中查询是否表中有此UID记录，如果有就将回收天线的值改为true，状态的值也改为true，否则提示该标签未经登记进入使用。④扫描天线读取到UID，在表中查询是否表中有此UID记录以及回收天线的值是否为false，如果以上条件都满足就将扫描天线值改为true，否则忽略此UID。⑥手术结束，备份保存数据表以便日后查询手术中器械使用的记录，关闭数据库。

Type表每当Device表有数据操作就对其做一次数据统计并更新Type表的内容。界面根据Type表的内容，统计显示当前各种类型手术器械使用和回收的数量情况。

### 5.4.3系统中有关数据库程序

Qt中的QtSql库提供了对数据的支持，只需在项目的.pro文件中加入：QT +=sql就可以将库引入，直接使用一些对数据库的连接和语句。其中的QSqlDatabase类用来创建与SQLite数据库的连接通道，QSqlQuery类用来提供SQL语句与数据库的交互；用户接口层的几个类实现了将数据库中的数据链接到窗口部件上。程序中通过createConnection（）函数来实现与数据库通道的连接，代码如下：

createConnection()

{

QSqlDatabase db = QSqlDatabase::addDatabase("QSQLITE");//创建sqlite数据库db.setDatabaseName("record.db");//创建的sqlite数据库命名为record.db

if (!db.open())//打开数据库

{QMessageBox::critical(0, "open db fail","connect fail", QMessageBox::Cancel);

return false;}

QSqlQuery query;// 创建表单

query.exec(QString("create table Device (id varchar primary key, "

"R1 varchar,R2 varchar,R3 varchar,state varchar,type varchar)"));

return true;

}

另外在程序后续的数据处理中通过使用一些query语句来对数据库表中的数据进行怎、删、改、查，实现业务对数据的要求，例如以下几句query语句：

query.exec(QString("select type from Device where type = '%1'").arg(type));

//到Device表中查询是否已存储此类型

query.prepare("insert into Device (type,R1,R2,R3,R4) values(:type,:R1,:R2,:R3,:R4)");

//将读取的数据插入到Device

表中query.exec(QString("update Type set state = 'yes' where id = '%1'").arg(sqlTmp));

//更新Type表中的一些类型的数据。

因为数据处理相对复杂涉及到很多语句和query语句，这里就不一一介绍。

### 5.5 本章小结

本章详细的分析介绍了手术器械管理系统的软件部分的设计。第一节介绍了阅读器端MCU上的控制程序设计，介绍了整个MCU程序的大体结构和分析重要部分的代码实现过程；第二节介绍了工控机端的基于Qt技术的上位机程序开发过程，先从整体结构设计开始介绍，再介绍了Qt 中本系统的UI界面开发细节，然后结合流程图分析了一些重要环节的代码实现和原理。第三节介绍了阅读器端的MCU和上位机之间的通信协议规范。第四节单独介绍了本系统在上位机程序中涉及到的数据库知识和应用，使得读者更加清楚本系统对数据的处理方法。总体上本章详细地分析了手术器械管理系统的软件设计过程和原理，使得读者能够很好的了解系统的软件部分。

### 第六章 系统可靠性的研究与改进

手术器械管理系统从需求开始就是一个对安全可靠性要求很高的手术辅助器械，不允许系统的任何意外漏读出现（除人为操作失误外），否则后果不堪设想。本系统在整体系统设计完成之后，根据理论分析和实际测量结果，对系统做了相应的可靠性分析和改进，确保系统的安全可靠。

### 6.1 系统的可靠性分析与改进

从系统的整体设计结构上来分析，可靠性需要得到保证的环节主要分为两个：系统在读取RFID标签时的绝对无误和系统在数据传递时的完整无误。

手术器械管理系统在登记录入天线和扫描天线读取数据时同时读取的标签数有限不会很多，在目前的抗冲突读取中基本不会出现漏读的情况，即使这样本系统也做了防护措施，采用多次读取的方法，防止任何一次遗漏读取。在回收桶天线读取时难度比较大，因为手术器械用完后随手扔在桶中器械内嵌的RFID标签朝向各个方向都有，同时桶内的标签数也会比较多，这样的读取环境很容易造成遗漏，为此专门设计了读取算法来避免漏读。数据通信发生在RFID标签到系统的阅读器和阅读器到上位机程序，只要任何一个环节出了问题，数据都会出现错误，是一种串联的关系，一断则断。因此在通信中都采用CRC校验的方法，来确保每次通信数据的正确性。

### 6.2 系统在数据通信上的可靠性保障

过去通信由于技术的限制和通信信道的复杂变化，严重影响信号在信道中的传输，很容易就会产生误码，导致信息的可靠度下降。为了保证数据通信中的可靠性，就必须通过校验来识别数据接收的正确性。CRC循环冗余校验是串口通信中最常用的一种校验方法。它采用数学的运算方法对接收的数据进行分析判断，从通信中数据本身的角度来验证。如果校验正确就传入下一层，不正确就修复或者丢弃该数据。

CRC校验先将发送的数据帧转化成多项式的形式，然后对多项式采用特定的计算方法计算，将计算得到的结果加在数据帧后面。接收方接收到该数据帧后，同样采用CRC校验，转化成多项式进行计算，如果等于直接约定的值说明接收的数据正确无误，从而确保了接收的数据一定是正确的。CRC信息码由信息码（n个Byte）和校验码（k个Byte）组成，写成（n，k）码。编码规则是先将信息码左移r位，再对生产多项式模二除，计算得到的余数就是CRC的校验码。对于任何一帧通信码字，有且仅有一个R次多项式g（x），该多项式的数学表达式如式子6-1：

 (6-1)

其中m(x)是信息多项式，r(x)是校验多项式，g(x)是生成多项式如式子6-2：

 (6-2)

信源通过生成多项式产生CRC校验码，信宿则利用生成多项式来验证接收的信息，循环冗余校验根据“权”的不同，其生成多项式有CRC8、CRC12、CRC16、CRC32等。手术器械管理系统选用的是CRC16，该生成多项式如式子6-3：

 (6-3)

接下来我们详细地分析一下任意长度数据帧的CRC16校验码生成的步骤。将系统接收的数据分成若干个8bit的字符，并依次从低位往高位的顺序发送，来计算CRC16 的校验码。

生成校验码的过程中，采用一个16位初值为0x0000的寄存器来存放信息码。先将数据的低八位和寄存器的高八位相异或，结果便存放在寄存器的高八位上，然后寄存器向左移一位，在低位补零，如果移出的高位数据是零，就接着左移一位，直到移除数据为一。之后将生成多项式和寄存器进行异或，结果存放在寄存器中。按照上述步骤重复执行直到计算完成十六位数据，寄存器中存放的数据就是该信息码的CRC校验码。图6-1是CRC16的计算流程。

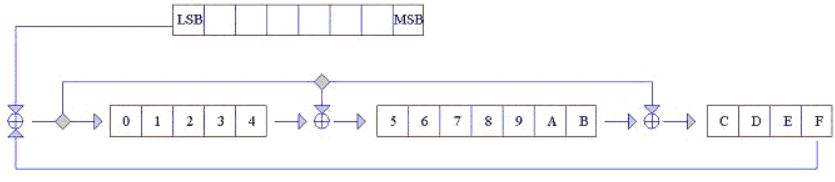


图6-1 CRC16的计算流程图

在手术器械管理系统的上位机中，串口每接收一帧数据都要通过calc\_crc()函数进行校验，该函数需要传入两个参数。第一个参数是需要校验数据的长度，第二个参数是要校验的数据的首地址。具体的实现代码如下：

int calc\_crc(unsigned int byte\_len, unsigned int \*data\_byte)

{ unsigned int current\_crc\_value;

unsigned int i,j;

current\_crc\_value=0xFFFF;

for(i=0;i<byte\_len;i++)

{current\_crc\_value=current\_crc\_value^data\_byte[i];

for(j=0;j<8;j++)

{if(current\_crc\_value&0x0001)

{current\_crc\_value=(current\_crc\_value>>1)^0x8408;}

else{

current\_crc\_value=(current\_crc\_value>>1);}}

current\_crc\_value = ~current\_crc\_value;

return(current\_crc\_value&0xFFFF);

}

只有通过CRC校验并得到正确的结果，程序才会把此次接收的数据往下一层传送，也就是上面所提及的环形缓冲层。

在MCU程序中采用字节查表法来进行CRC校验，在MCU硬件端可以大大提高计算速度。程序如下：

u16 table\_crc(u8 \*ptr,u16 len) // 字节查表法求 CRC

{

u16 crc=0xffff;

while(len--)

{

crc= (crc >> 8) ^ crc\_ta((crc ^ \*ptr) & 0xff); //FLASH

ptr ++;

}

return(~crc); // CRC校验码取反

}

crc\_ta是先前算好的CRC16的256个余式表数组，存放在flash内存中。通过uint table\_crc(uchar \*ptr,uchar len)函数来快速计算CRC，输入: uchar \*ptr : 数据；uchar len : 数据长度。输出: CRC16码。

通过以上2次的CRC校验，确保了在数据传递的环节中不会出现错误数据，杜绝了因数据传递出错了而引起的问题，做到有传递就有校验。

### 6.3 系统在数据读取上的可靠性保障

手术器械管理系统在登记录入天线和扫描天线读取数据时采用多次读取的方法，防止任何一次遗漏读取。当一次的漏读的概率为0.5%时，当执行5次读取时，漏读的概率为0.5%\*0.5%\*0.5%\*0.5%\*0.5%=0.03125% ，如果读取10次的话漏读的概率为0.0009765625%，按此推断，次数越多，发生漏读的概率越低。但由于不可能一直读下去，本系统设计成，当执行某个天线读取时，在没有执行其他的命令前，会一直循环做读取操作，并在程序中做校验，判断是否已经读到目前读取的数据，如果没有就添加。这样防止因一次漏读二造成的遗漏。

回收桶天线读取在已有的15693抗冲突算法的基础上，采用的改进的防冲突算法，提高回收桶中标签的抗冲突效率。天线同时工作会相互产生干扰，使得读取性能变差，因此回收桶的三个天线不是同时上电运行的。它们之间根据事先规定好的顺序依次执行，每一个天线覆盖回收桶的一部分区域。正常的情况下，当场内标签掉电后会重新进入抗冲突，所以当天线之间相互切换时势必会使场内的标签掉电，这样不管之前天线有没有读到该标签，都会再次参入下一次抗冲突。这种情况会加大系统的抗冲突运算的负担，减低读取率，增加抗冲突识别时间。为了避免以上情况的发生，针对手术器械管理系统的回收桶设计了一个标签抗冲突算法。基本思想是利用ISO15693芯片的静默功能，先做静默再进行抗冲突。每次读到一组RFID标签的UID后进行一定的运算，重新排序并剔除掉当前重复的UID，最后静默当前记录的重排后的标签，再次进行抗冲突读取，依此循环读取。

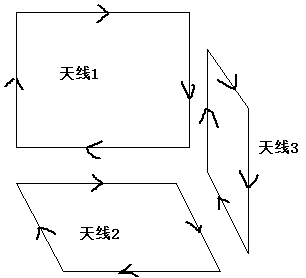


图6-2回收桶内的天线设计

回收桶内天线线圈的分布如上图6-2所示，整个抗冲突过程如下：先让天线1开始工作，进行一次Inventory命令，其覆盖的区域完成一次抗冲突，将读取的RFID标签的UID全部暂时缓存在一个UID缓冲池内；切换到天线2，接着对UID缓冲池内的RFID标签，发一遍静默命令后，让天线2进行一次Inventory命令对其覆盖的区域进行一次抗冲突，将此次读取的RFID标签的UID也存入UID缓冲池内，并进行运算排序，剔除掉重复的UID；然后切换到天线3，对UID缓冲池内的UID依次发送静默命令，让天线3再进行抗冲突，将读取的UID存入UID缓冲池，剔除重复部分后将UID缓冲池内的UID通过串口发送给上位机程序。为了能够确保系统的回收桶内RFID标签没有遗漏，在上述读取的情况下，也跟登记录入天线和扫描天线一样，采用重复多次读取的方法，不过需要注意的是在每次重复读取前要先清空UID缓冲池，再重复上述的读取过程。回收桶内三个天线的抗冲突流程图如图6-3所示。



图6-3 回收桶内三个天线抗冲突流程图

### 6.4 系统的实物和运行界面展示

为了验证系统的准确性和可靠性，选取了8张ISO15693协议的射频标签，分别放置在40\*40纱布、100\*60纱布、100\*100纱布和手术剪刀四种器械中，并将这8张标签的AFI写成对应型号的编码。然后模拟手术过程使用手术器械。

手术操作流程如下：手术开始打开手术管理系统，点击操作界面的登记录入（COUNT IN）按钮，将手术器械使用前在登记录入处扫描录入，器械使用完后扔进回收桶中。手术结束后点击回收（COUNT OUT）按钮，系统自动统计回收桶中的器械。根据显示结果判断，如果有未回收的遗漏器械，且不在可视范围内，点击扫描（SCAN）按钮用扫描器去病人身体部位探测寻找，快速定位遗漏器械的位置。最后点击详情（DETAIIED）按钮进入详细数据界面保存数据，以便日后查询手术记录查看手术器械使用情况。

实验过程中故意让1块100\*60型号和1块100\*100型号的纱布未放进回收桶模拟遗漏的情况，系统能够准确的显示这2种纱布各有1块未回收，实验结果如图6-4所示。然后用扫描器找到这2块纱布后，系统显示没有手术器械遗漏。

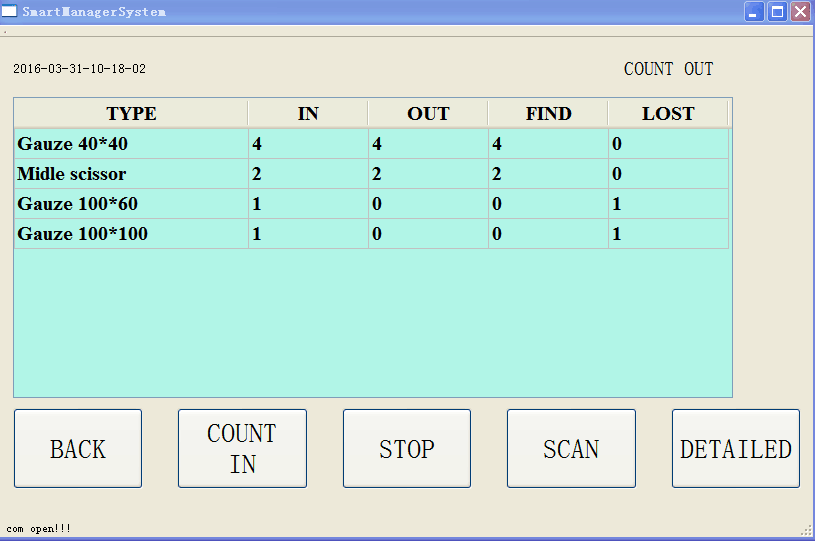


图6-4 实验结果界面

实验证明登记和回收的天线能够迅速没有遗漏地读取多个手术器械，扫描器的天线能够快速准确地探测到遗漏的器械。长时间工作，系统运行稳定，响应迅速。整个系统实物图如图6-5所示。



图6-5系统实物

### 6.5 本章小结

本章介绍了手术器械管理系统的可靠性分析和改进，先从整体上分析了需要改进的地方，以及改进的方法，然后分别从数据读取和数据传递的两个部分进行了理论分析和改进说明。本章所做的工作对于本系统属于画龙点睛之笔，只有有了可靠性，手术器械管理系统才具有实际的使用价值。

### 第七章 结论与展望

### 7.1结论

本文针对手术异物遗留的难题，结合ISO15693协议的RFID芯片，自主设计了一种RFID手术器械管理系统。该系统与之前手术器械管理方法相比：可以大大降低手术中手术器械清点者的压力和工作量，准确、稳定的显示手术器械的使用和回收情况，确保手术器械无遗漏，为手术安全提供最后一道防线。本文在深刻学习射频识别技术的理论基础上，结合手术室内手术器械使用的实际情况，对内嵌有RFID标签的手术器械进行规范、高效、安全可靠的系统的设计。本文具体的工作内容如下：

1) 首先从整个系统的业务需求入手结合事先掌握的射频识别技术的理论知识设计了手术器械管理系统的整体框架。再根据框架中涉及的理论知识和原理进行学习和掌握。

2) 完成了系统硬件电路中FPGA编解码电路、MCU电路设计和FPGA与MCU两个模块之间的接口设计。针对不同模块之间时序同步问题设计了通过单片机外部中断模仿SPI通信方式的方法来传递数据。基本完成了系统硬件部分对RFID标签读取的功能。

3) 开发了基于STC单片的C语言程序开发，完成了整个硬件电路的控制功能。基于Qt开发了系统的器械管理操作界面及系统的上位机程序，通过串口与STC单片机相连，传递数据和操作命令，将读取的数据存放在QSlite数据库中并进行相应需求的处理后通过UI界面进行显示。

4) 最后对手术器械管理系统的实际使用和测试后，分析系统的可靠性和需要改进的部分，进行了系统相应的提升和优化，有效的防止了手术器械的读取遗漏和数据传递校验。总体上达到了起初的设计需求。

### 7.2展望

本文紧密围绕手术器械管理系统的应用设计，积极开张了相关的研究工作，取得了一定的研究成果，本人也在改项目中深入学习了射频识别技术、FPGA硬件电路开发、MCU应用开发和基于Qt技术的UI应用程序开发等知识，但仍然感觉到一些该系统还有很多可以扩展和有待改善的地方，值得更加深入的去学习和提高。主要在于：

1)上位机上运行的手术器械信息数据的管理软件功能相对比较简单，实现了一些基本数据存储、备份和查表的功能，而当对数据做复杂的运算和生成报表功能时就具有一定的局限性。

2) 本系统使用的工控机因考虑到开发成本，选用了一些性能并不是很好的，价格相对比较低的品牌。但如果为了整个系统在控制上的稳定和可靠，可以考虑更换更加专业标准的工控机代替。

3) 对于该系统可以通过网络的扩展连接到医院的服务器上，将手术器械的使用情况上传至服务器，为智能化医院提供支持。比如：和医院的手术器械的库存量进行数据处理，可以得到当前医院手术器械可用的量，有利于医院对手术器械的库存管理。总之该系统可以作为手术室内的一个终端为医院大数据服务提供数据源采集。

### 参考文献

[1] 黄浩. 永不消失的纱布[J]. 中国信息化. 2011, 21(15): 62-63.

[2] 王珂. 医用病床的智能化设计研究——检查病床为例[D]. 上海:东华大学, 2011.

[3] 单红霞，雷云霞. 重视手术护理记录单的书写[J]. 新疆医学. 2009, 39(10): 110-113.

[4] 丁效军，郑理华，陈宇珂. RFID技术在医疗器械管理中的应用研究[J]. 医疗卫生装备. 2012, 33(8): 38-39.

[5] 于寅虎. 医疗电子解决方案开始决逐中高端市场[J]. 电子技术应用. 2015, 41(2): 14.

[6] 宁焕生. RFID产品研发及生产关键技术[M]. 北京: 电子工业出版社, 2007: 211.

[7] 张毅. 浅析需求分析在软件工程中的作用[J]. 技术与市场. 2011(08): 343-344.

[8] 李亚锋. ARM嵌入式Linux系统开发从入门到精通[M]. 清华大学出版社, 2007: 308.

[9] 译者萨默菲尔德. Qt高级编程[M]. 电子工业, 2011: 327.

[10] Karim，Yagbmour，Reilly O.，等. 构建嵌入式LINUX系统[M]. 中国电力出版社, 2004: 424.

[11] Jasmin, Blanchette, Mark, et al. C++ GUI Programming with Qt 4 (2nd Edition) (Prentice Hall Open Source Software Development Series)[M]. Prentice Hall, 2008: 752.

[12] 王松. RFID系统防冲突算法的研究与实现[D]. 天津:天津理工大学, 2012.

[13] 周立功. ARM嵌入式系统基础教程[M]. 北京航空航天大学出版社, 2008: 491.

[14] 李亚锋. ARM嵌入式Linux系统开发从入门到精通[M]. 清华大学出版社, 2007: 308.

[15] 胡文，冯剑，姜海涛，等. Windows CE嵌入式系统程序开发[M]. 机械工业出版社, 2013: 471.

[16] 刘恩博，田敏，李江全. 组态软件数据采集与串口通信测控应用实战[M]. 人民邮电, 2010: 299.

[17] 薛晓光. 智能电视三维触摸板遥控器的设计与实现[D]. 青岛:中国海洋大学, 2013.

[18] 王倩. 基于Linux和DM9000的网卡驱动设计[D]. 辽宁:辽宁大学, 2013.

[19] 李彦，李锵. 基于嵌入式Linux系统的双网卡大数据传输[J]. 电子测量与仪器学报. 2014(09): 1027-1032.

[20] 朱吉佳，蔡家麟. 基于Qt的业务监控系统界面设计与实现[J]. 计算机技术与发展. 2008(03): 236-239.

[21] 杨柳，岳坤，庞和明，等. Qt/Embedded及嵌入式Linux在智能监控系统控制中的应用[J]. 计算机应用. 2010(S1): 289-291.

[22] 申伟杰，彭楚武，胡辉红. 嵌入式Linux中基于Qt/Embeded触摸屏驱动的设计[J]. 中国仪器仪表. 2006(04): 48-51.

[23] 蒋伟. RFID与无线网络融合关键理论和技术研究[D]. 北京:北京邮电大学, 2008.

[24] 白鹤，吕红亮，王劲林. 进度显示的大文件上传组件的设计与实现[J]. 计算机工程与应用. 2009(05): 91-94.

[25] 阿秀梅，陈思敏. 手术中擦拭器械纱布细菌含量研究与分析[J]. 现代临床护理. 2006(02): 39-41.

[26] 宁焕生. RFID产品研发及生产关键技术[M]. 电子工业出版社, 2007: 211.

[27] 霍亚飞. Qt Creator快速入门[M]. 北京: 航空航天大学出版社, 2012: 465.

[28] 霍亚飞. Qt及Qt Quick开发实战精解[M]. 北京: 航空航天大学出版社, 2012: 259.

[29] 马玉春，李壮，张鲲，等. 计算机监控系统开发与实战[M]. 机械工业出版社, 2012: 371.

[30] 童启明. 控制系统数字仿真与监控组态软件应用[M]. 科学出版社, 2006: 291.

[31] 博斯. 信息论、编码与密码学[M]. 机械工业, 2010: 231.

[32] 张丽英，王世祥. 信息论与编码基础教程[M]. 清华大学, 2010: 191.

[33] 张秀娟. 生产者-消费者系统的建模与行为分析方法研究[J]. 微电子学与计算机. 2004(05): 97-100.

[34] 彭民德. 基于Web的生产者-消费者同步问题的实现技术[J]. 计算机工程与应用. 2006(22): 50-51.

[35] 李金忠，曾劲涛. 生产者/消费者经典同步问题的深入探究[J]. 电脑编程技巧与维护. 2008(01): 20-22.

[36] 孙杰林，王中训. 基于15693协议的射频识别读卡器的设计[J]. 电视技术. 2012, 36(3): 29-31.

[37] 胡晓，谈恩民，陈寿宏，等. 嵌入式数据库SQLite在边界扫描测试系统中的应用[J]. 微电子学与计算机. 2014(5): 188-192.

[38] 孙萍萍. 基于MIPS的嵌入式数据库SQLITE的移植及应用[J]. 计算机应用与软件. 2011, 28(1): 109-111, 153.

[39] Kevin, W., Fall, et al. TCP/IP Illustrated, Volume 1 (2nd Edition)[M]. Addison-Wesley Professional, 2011: 1056.

[40] 尹圣雨. TCP/IP网络编程[M]. 人民邮电出版社, 2014: 410.

[41] 任克强，刘苏，谢斌. 基于ARM11和Linux的网络视频监控系统[J]. 电视技术. 2014, 38(5): 156-159.

### 英文缩写词表

|  |  |  |  |
| --- | --- | --- | --- |
| 缩写 | 英文全称 | 中文 | |
| RSIS | Retained Surgical Items | | 手术异物遗留 |
| RFID | Radio Frequency Identification | | 射频识别 |
| MCU | Microcontroller Unit | | 微处理器 |
| FPGA | Field－Programmable Gate Array | | 现场可编程门阵列 |
| CPU | Central Processing Unit | | 中央处理器 |
| VCD | Vicinity Coupling Device | | 阅读器 |
| VICC | Vicinity Card | | 标签芯片 |
| SOF | Start of Frame | | 帧头 |
| EOF | End of Frame | | 帧尾 |
| CRC | Cyclic Redundancy Check | | 循环冗余校验 |
| SQL | Structured Query Language | | 结构化查询语言 |
| UID | Unique Identifier | | 唯一标识符 |
| IPC | Industrial Personal Computer | 工控机 | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |

### 作者在攻读硕士学位期间公开发表的论文及参加的项目

### A：在国内外刊物录用的论文

1. 基于ISO15693射频标签的RFID手术器械管理系统设计 一文被《实验室研究与探索》中文核心期刊录用

### B：参加的项目

1. 江苏省科技支撑计划-工业部分（重点），项目编号BE2013008-3。

致 谢

时光匆匆，毕业转眼就到了。在此，提笔感谢那些在研究生期间给与过我帮助的需要感谢的人。

首先，我要感谢这三年中专用集成电路重点实验室各位老师给予了我指导，正是有了您们的帮助和指导让我在这段学习生活中找到了正确的前进方向，在知识方面能够保持深入的学习，不断地成长。另外，我要特别感谢我的研究生导师景为平老师，感谢景老师在我学习和生活中的帮助和指导，您严瑾的科研态度，渊博的专业知识给我留下了深刻的印象，不仅让我学到了够硬的专业知识，更让我学到了处理问题的方法和态度，这将是我人生中的重要收获。

另外，我还要感谢计算机系的陈晓勇老师，在软件学习和开发中帮助我解决问题，使得我可以更加快速的完成本文系统UI程序的开发。感谢所有14级电子学院的研究生同学们，与我一起学习、科研生活。感谢我的师兄沈伯洵、龙硕、周蒙蒙、张慧雷和钱一文在我整个系统开发中的帮助和指导。感谢都平和范巍同学三年学习生活中的陪伴，和你们一起在景老师门下学习的经历我终身难忘，愿这份情谊长存。

最后，感谢我的父母对我的支持和关心，让我可以专心的投入到学习中，感谢我的妻子和儿子，在我遇到困难和挫折的时候你们是我的精神支柱，我爱你们！