杭州电子科技大学

硕 士 学 位 论 文

题　　目：基于 STM32 的超高频 RFID 读写器软件系统的设计实现

研 究 生　王　成

专　　业　微电子学与固体电子学

指导教师　李文钧　副教授

　　王　彬　教　授

完成日期　2013 年 2 月

杭州电子科技大学

学位论文原创性声明和使用授权说明

原创性声明

本人郑重声明： 所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。申请学位论文与资料若有不实之处，本人承担一切相关责任。

论文作者签名： 日期： 年 月 日

学位论文使用授权说明

本人完全了解杭州电子科技大学关于保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属杭州电子科技大学。本人保证毕业离校后，发表论文或使用论文工作成果时署名单位仍然为杭州电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。（保密论文在解密后遵守此规定）

论文作者签名： 日期： 年 月 日

指导教师签名： 日期： 年 月 日

杭州电子科技大学硕士学位论文

基于 STM32 的超高频 RFID 读写器软件系统的设计实现

研 究 生：王　　成

指导教师：李 文 钧　副教授

王　　彬　教　授

2013 年 2 月

Dissertation Submitted to Hangzhou Dianzi University

for the Degree of Master

UHF RFID Reader Software System Design and Implementation Base On STM32

Candidate: Wang Cheng

Supervisor: Prof. Li Wenjun

Prof. Wang Bin

February，2013

摘　要

射频识别技术是一种先进的自动识别技术，自上世纪兴起以后到本世纪已经得到了长足的发展，极大的改变了人们的生活方式。而 UHF 射频识别技术是当前 RFID 发展的前沿，它具有读写距离远，标签成本低，读取速度快等优点，已经被应用到包括物流管理、交通运输等各个方面。随着物联网发展的大潮，UHF 射频识别技术作为物联网技术的一部分也不断更新，推动着整个物联网行业应用向前发展。

在超高频 RFID 发展过程中，影响其广泛应用的主要因素是读写器的成本太高，所以解决问题的关键就是在不损失性能的情况下开发一款成本比较低的读写器。在读写器的开发过程中，软件系统起着非常重要的作用，它是读写器硬件和具体应用的桥梁，起着承上启下的作用。同时软件系统也是定制系统，随着硬件的不同而改变，所以一款读写器的成功与软件系统实现的效果有着不可分割的联系。甚至，读写器实现过程中比较繁琐工作也在软件系统。软件系统可以直接控制读写器，实现 UHF RFID 相关的国际标准通信协议，同时将信息发送至客户端或者上位机界面，将及时信息反馈给用户。

本设计主要研究基于 STM32 的低成本 RFID 读写器的软件系统，由于读写器硬件设计的特殊性，因此也与其他读写器的软件系统有所不同。超高频读写器必须符合 EPC Global 1 Class2 协议标准，才能与不同厂家的超高频标签通信，而市面上的读写器一般由硬件来实现协议处理，我们为了节约硬件成本，使用 MCU 来模拟实现协议规定的各项任务包括读写器读写标签流程、基带信号编解码等。在读写器应用过程中，更多时候是一个读写器来读写多张标签，因此读写器的多标签防冲突功能是读写器实现的必要组成部分，我们根据 Gen-2 协议标准的特点采用动态帧时隙 ALOHA 算法，很好的解决了多张标签同时读写的碰撞问题。由于 Gen-2 协议对于读写器与标签通信过程中时间限制比较严格，同时在读写器接收信号的时候，需要对基带信号进行采样，所以主频、采样速率等是我们选取主控芯片的时候考虑的主要因素，而意法半导体的 STM32F207 芯片很好的满足了我们的需求，因此作为我们的主控芯片。在具体实现过程中，优秀的程序设计方法是软件系统实现的关键所在，我们力求使程序简洁、高效，能够很好的和读写器硬件平台配合工作。

本设计历经多次试验和改版，实现了读标签距离 8 米（天线增益 5dBi）写标签 3 米的性能，最高单标签读取速度每秒 200 次和防冲突的性能，成本约为市场价格的三分之一，达到了设计指标，有较高的实用价值。

关键词：UHF，射频识别，读写器软件系统，低成本

ABSTRACT

Radio frequency identification technology is an advanced automatic identification technology, since the rise of the last century to this century it has got plenty of development and has greatly changed people's way of life. UHF radio frequency identification technology is the forefront of the current RFID development; It has many advantages such as read and write distance, low-cost tags, fast read speed, has been applied to all aspects of logistics management, transportation. With the tide of the development of Internet of things, UHF radio frequency identification technology as the part of Internet of things technology constantly updated to promote the whole of Internet of things moving forward.

In the UHF RFID development process, the factor affecting the development of the UHF radio frequency identification technology is the reader too costly, so the key to solving the problem is the development of the higher performance-price ratio UHF reader. System software plays the same important role of Reader hardware in RFID reader system; it is the bridge of the specific application and the reader and plays essential role. The software system is also customizable system base on different hardware, so software system is inextricably linked to the success of a reader. Even major cumbersome task and the key of the realization of the reader are in the software system. The software system can directly control the reader, realize UHF RFID international standard communication protocol, and send information to the client or host computer interface, timely feedback to the user.

This designe primarily study the low-cost reader software systems, due to the particularity of the hardware design of a low-cost reader software system and the general reader software system is different. Ultra-high frequency Reader must comply with EPC Global 1 Class2 protocol standard, so it can be match with different manufacturer UHF RFID tags, but the general reader on the market by the hardware to achieve protocol processing. In order to save the cost of hardware, we use of MCU to simulate the realization of all tasks of Gen-2 (including read and write label process, baseband signal decoding, etc). In the reader application, A Reader to read and write more than labels, so the anti-collision function is the necessity to realize the Reader. According to Gen-2 protocol standard features, we use dynamic frame timeslot ALOHA algorithm, it’s very good solution to solve reading and writing collision problems of multiple labels. Due to the Gen-2 protocol for reader and tag communication process in a strict time limit, so in the selection of main control chip, Core frequency and the number of interfaces are our main considered factors, while

STMicroelectronics STM32F207 chip is very good to meet our needs, so as our master control chip. In the concrete realization of the process, Good programming method is the key to realize the software system, we strive to make procedure simple and efficient, can work well with reader hardware platform.

This design through many test and revised, and realize the to read the label distance eight metres (antenna gain 5 dbi) write tag 3 meters, the performance of the highest single tag reading speed 200 times per second and anti-collision performance, reached the design target, it's cost is about one-third of the market price, have higher practical value.

Key words: UHF, Radio Frequency Identification, Reader System Software, Low cost

目　录

第一章 绪 论

1.1 研究背景

“物联网”是在“互联网”的基础上，将其用户端延伸和扩展到任何物品进行信息交换和通信的一种网络。物联网最初在美国被提出时，还只是停留在给全球每个物品一个代码，实现物品跟踪和信息传递的设想。如今，物联网被称为继计算机、互联网之后世界信息产业的第三次浪潮，将上升为国家战略，成为下一阶段 IT 产业的任务[1]。在物联网时代，人类在信息与通信的世界里将获得一个新的沟通维度，从任何时间、任何地点人与人之间的沟通和连接，扩展到任何时间、任何地点人与物、物与物之间的沟通和连接。

近年来物联网被视为全球经济复苏的技术引擎，世界上很多国家都在投入巨资建设物联网环境。物联网可分为感知网、传输层和应用层，而射频识别（RFID）技术即为物联网感知层的核心技术之一[1]。事实上，RFID 作为一项新兴的自动识别技术，已在全球范围内广泛应用于生产制造和装配、进出管理、门禁管理、车辆管理、防伪、票务、产品跟踪、仓库管理、邮件/快运包裹处理、文档追踪/图书管理等众多领域，与人们的生产、生活息息相关。对 RFID 技术和 RFID 产品进行深入研究，不论对于国民经济发展还是对于国家安全都具有重要的意义[2-4]。

RFID 系统的硬件包括电子标签和读写器两部分，读写器通过天线与电子标签进行无线通信，来实现对电子标签数据的读出和写入。读写器又可以与计算机网络进行连接，来完成对数据信息的存储、管理和控制[3]。从这个意义上说，读写器也是电子标签与计算机网络的连接通道。读写器是一种数据采集设备，其基本作用就是作为数据交换的一环，将前端电子标签所包含的信息，传递给后端的计算机网络。现在，射频识别系统已发展出了多种模式和标准[5]。UHF 射频识别技术是当前 RFID 的发展前沿，具有读写距离远、读取速度快、标签成本低和体积小等特点。目前该技术已被广泛应用在物流管理和交通运输等领域。然而，UHF 射频识别系统的读写器成本很高，对于该项技术的进一步普及非常不利。降低读写器的成本已成为 UHF 射频识别技术的一项关键问题。

1.2 RFID 系统简介

1.2.1 RFID 系统的基本组成

作为物联网的核心技术之一，RFID 技术的应用领域非常广泛。由于不同领域的应用需求不同，造成了目前多种标准和协议的 RFID 设备共存的局面，这就使得应用系统架构的复杂程度大为提高[6]。但是就基本的 RFID 系统来说，其组成相对简单而清晰，主要包括 RFID 标签、读写器、中间件和应用软件等四部分。

1、读写器

读写器（reader）又称读头、阅读器等，它在 RFID 系统中扮演着重要的角色，读写器的工作主要是负责与电子标签完成通信，按照主机发出的指令，完成对电子标签的访问。一般，RFID 系统的工作频率即载波频率由读写器来决定，同时标签与读写器之间通信距离也有读写器发射功率决定。读写器根据使用的结构和技术不同可以是读或读/写装置，它是 RFID 系统信息控制和处理中心[6]。读写器内部结构通常由射频模块、逻辑控制模块和天线三部分组成，其内部结构如图 1.1 所示。

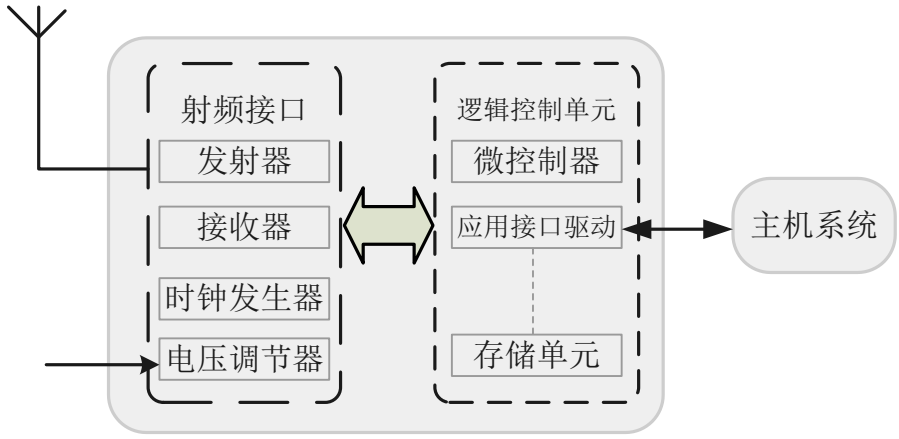

图 1.1 RFID 读写器内部结构图

（1）射频模块的主要任务是：

1） 产生高频发射能量，为电子标签提供能量并激活它。

2） 调制发射信号，将数据传输给电子标签。

3） 接收并解调来自标签的射频信号。

在射频模块中有两路分隔开的信号通路，分别应用于读写器和电子标签相互之间通信的两个不同方向数据传输。发射通路将基带数据调制以后发送出去，而接收通路将来自电子标签的信号解调后传送回基带[7]。

（2）逻辑控制模块的主要任务是：

1） 与应用系统软件通信，并执行从应用系统软件发送来的指令。

2） 控制电子标签与读写器之间通信的各个参数。

3） 基带信号的编码和解码。

4） 加密和解密读写器与标签之间的通信数据。

5） 完成标签的防碰撞算法。

（3）读写器天线

读写器天线能够将空中的电磁波转换为电流信号，或者将电流信号转换为能够发射出去的电磁波。在 RFID 系统中，读写器必须通过天线来发射能量，形成电磁场，通过电磁场对电子标签进行识别，所以可以说，读写器天线的电磁场辐射范围就是读写器的

可读区域[8]。

2、电子标签

电子标签（Electronic Tag）是指由 IC 芯片和天线组成的超小型的标签，天线的作用是用于与读写器进行通信[9]。RFID 系统工作的时候，读写器发出询问信号（能量），标签（能量）在接收到询问信号（能量）以后将一部分整流为直流电源供电子标签工作使用，然后电子标签通过反射另一部分能量，将自己数据信息携带发射回读写器中完成与读写器的通信。电子标签是 RFID 系统中真正的数据载体，在不同的应用场合里表现为不同的形态，比如在动物追踪领域称为动物标签或者电子狗牌、动物追踪标签；在车辆自动识别领域中称为电子牌照、车辆远距离 IC 卡或车辆远距离识别标签；在访问控制领域称为一卡通或者身份卡。电子标签的内部结构如 1.2 图所示。

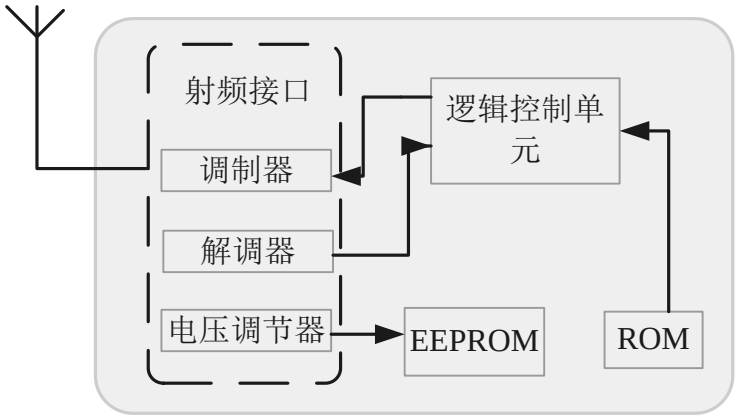

图 1.2 电子标签内部结构图

电子标签内部各个模块的功能如下

(1) 天线：用于接收读写器发送过来的信号，并将标签自己的信息发送回读写器。

(2) 电压调节模块：将读写器发射出的能力整流为直流电源，并由大电容存储起来，然后经过稳压电路后为标签提供稳定电源。

(3) 调制器：逻辑控制模块发送出的基带信号经过调制器调制以后通过天线发送出去。

(4) 解调器：去除载波信号将有用信号传送回逻辑控制模块。

(5) 逻辑控制模块：用来解码来自于读写器的信号，并根据协议流程将读写器所需信号传送回读写器。

(6) 存储单元：可以是 EEPROM 或 ROM，作为标签系统运行以及识别数据的位置。

3、中间件

中间件（Middleware）处于读写器与后台网络的中间，扮演 RFID 硬件和应用程序之间的中介角色，是 RFID 硬件和应用之间的通用服务，这些服务都具有标准的程序接口和协议，能实现网络与读写器之间的无缝连接[10-13]。中间件作为 RFID 系统运作的中枢，解决了硬件接口与应用系统连接的问题，即使标签数据增加，读写器种类增加时，应用端也不需要修改就能够处理数据。中间件的结构如图 1.3 所示。

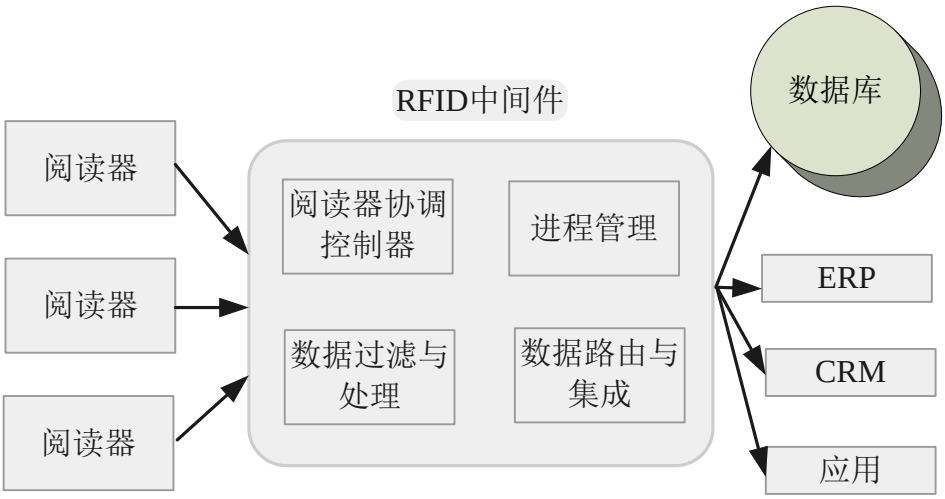

表 1.3 中间件结构图

中间件内部各个模块功能如下

(1) 程序模块集成器：程序模块集成器具有数据搜集、过滤、整合与传递等功能。

(2) 读写器接口：读写器接口遵循相应的读写器接口通信协议，与读写器进行通信连接。

(3) 应用程序接口：应用程序接口提供应用程序与程序模块集成器之间的接口。应用程序可以有多种表现形式，包括订单管理系统（OMS）/仓库管理系统（WMS）和物流管理系统（LMS）等。

(4) 网络访问接口：网络访问接口提供与互联网的连接，用来构建物联网名称解析服务 IOT-NS 和物联网信息发布服务 IOT-IS 的通道。

4、应用软件

RFID 应用系统软件针对不同的企业需求而开发的不同的应用软件，他可以有效的控制读写器对标签信息进行读写，并搜集目标信息进行统计和处理。RFID 应用系统软件可以嵌入到现有的电子政务和电子商务平台中，与 CRM、ERP 和 SCM 等系统综合起来可以提高各行业的运作效率[14-15]。

1.2.2 RFID 的主要频段和应用领域

RFID 系统工作频率主要考虑不能对其他无线电服务造成干扰，所以通常情况下，按照读写器发出的频率（系统工作频率或者载波频率）分为低频、高频和微波系统。

（1） 低频系统（LF）

低频系统的工作频率为 30KHz~300KHz，RFID 常见的低频工作频率有 135KHz 和 143.2KHz[15]。低频系统的特点是电子标签内保存的数据量较少，读写距离短，读写天线方向感不强。

低频标签的典型应用有工具识别、动物识别、容器识别、电子闭锁防盗（带有内置应答器的汽车钥匙）等。

（2） 高频系统（HF）

高频系统的工作频率范围为 3MHz~30MHz，其中常见的高频工作频率是 6.75MHz、13.56MHz 和 27.125MHz[15]。这是一个比较开发的频段，标签的读写距离最远大概 1M 左右。这个频段的标签大部分为无源的，靠与读写器之间电磁耦合来提供能量，而且应用比较成熟。我国的二代身份证、学生证优惠卡、公交卡等项目都采用这个频段的产品。

（3） 超高频系统（UHF）

这个频段主要分为 433MHz 和 860MHz~960MHz。这个频段的标签和读写器之间的有效通信距离最远。标签靠电磁反向散射耦合与读写器进行通信，数据传输速率更快，可以同时读取多个标签，但是穿透能力不强。其中本文主要介绍的 860MHz~960MHz 是 ISO 规定的无源超高频设备使用的频段。近年来发展迅猛，在酒类防伪、港口物流管理等领域应用较广。

（4） 微波（MW）

这个频段主要分为 2.45GHz 和 5.8GHz 两个频段。这个频段的优势在于受强电磁场的干扰较小，识别距离介于高频和超高频之间，同时标签可以设计的很小，但是成本较高。

1.2.3 RFID 系统的主要工作原理

RFID 系统的基本工作原理是：由读写器通过发射天线发送特定频段的射频信号，当电子标签进入读写器天线的有效辐射范围的时候，标签产生感应电流获得能量而被激活，标签解调出去载波信号，得到有效的基带信号，解码得到来自读写器的信息，按照协议流程进行处理以后进过编码调制以后通过标签自己的发射天线发送回读写器，读写器在限定时间内收到来自标签的信息，进过处理以后发送回主机（客户端界面），完成对标签信息的读写操作。客户端（一般为 PC 机）发送指令来指挥读写器对标签进行一系列的相应的操作。

从读写器到标签之间的通信及能量感应方式来看，RFID 系统一般分为两类，即电磁反向散射耦合系统和电感耦合系统。在 UHF RFID 系统中标签是通过电磁反向散射耦合从读写器获得能量。电磁反向散射耦合，即雷达原理模型，依据的是电磁波的空间传播，发射出去的电磁波碰到目标后反射，同时将目标信息带回给发射源。

1.2.4 RFID 国内外发展现状

1.2.4.1 国外发展现状

射频识别技术是 21 世纪最成功的技术之一，在工业自动化、商业自动化以及交通管理控制、物流控制等领域得到了大力发展，大幅提高了生产工作效率，目前新兴的 UHF 波段射频识别技术更是有着很大的发展潜力，一些成功的方案可能会改变人们的生活方式，提高生活质量。因此，世界各国都在 RFID 技术上投入了巨大的精力。现在，在美国专利局注册的关于 RFID 的技术已经有 339 项[16]。

作为一项高新技术，射频识别技术在国外尤其是发达国家的发展可谓非常迅猛，在各行各业都出现了应用。下面举一些国外比较成功的案例。

走在射频识别领域最前沿的是美国。最为重要的是美国在标准的制定方面起着主导作用，最具代表性的就是业界应用最为广泛的 EPC Global 标准，该标准在抗冲突和执行效率方面有

着绝对的技术优势。2003 年 11 月 5 日，沃尔玛百货正式宣布，到 2005 年，所有供应沃尔玛百货公司的货物包装箱上都要采用 RFID 技术的电子标签。因为沃尔玛公司认为这项技术可在很大程度上降低库存成本和操作错误率。据称沃尔玛公司在使用这项技术后每年节省成本高达 84 亿美元[17]。

欧洲地区的发展也很迅速。有不少欧洲国家率先把这项技术应用在高速公路电子收费站，将 RFID 标签粘在车上，经过收费站的时候自动扣款，无需停车。这样不但减少了通过时间而方便车主，还大幅降低了收费站的运营成本，提高了收费站的通过量。英国航空公司正在研究一种利用 RFID 技术简化登机流程并提高安全性的方案[16]。

而在亚洲，韩国也非常重视 RFID 技术的发展和产业化，尤其是韩国政府对于这项技术非常重视。技术开发领域不是韩国的强项，但在政府的大力支持下，韩再应用方案的研究上处于国际领先的水平。

1.2.4.2 国内发展现状

中国作为世界第二大经济体和世界经济发展的主要推动力已成为 RFID 技术最具潜力的市场，新兴的市场对新兴科技的需求是强烈的，因此我国也在大力的进行 RFID 技术的发展。随着国家对物联网技术的不断重视，作为物联网核心技术的 RFID 技术越来越受到国内研究机构和高新技术企业的重视。

在标准方面，我国目前还没有很多的发言权，但我国也已有超过一百家企业参与到标准的制定中，相信在政府的大力支持下一代 RFID 技术的标准制定中我国会有更大的发言权。

在硬件方面，我国已有很多研究机构和企业研发出了读写器和标签，在性能方面较发达国家水平仍有一定差距，但是差距在不断被缩小。我国在读写器芯片方面目前还处于研究试验阶段，读写器产品的核心大多还是国外的读写器芯片，但也有一些公司在声场拥有自主知识产权的分立器件读写器。标签方面，由于性能不及欧美公司的产品所以市场份额还非常小。

在应用方面，我国在本世纪初迎来了 RFID 技术普及的浪潮。各大城市开始使用射频识别卡作为公交卡，大大提升了公交系统的工作效率也方便了市民们的出行。各个中学、大学甚至小学使用了射频识别卡作为计费卡，学生们在学校的所有开销都可以用它进行付款，方便了同学也让家长们更安心。我过第二代身份证也采用了射频识别技术，在公共场合如酒店，车站等地方可以方便的调出个人信息，提高了效率的同时也杜绝了造假行为。在交通方面如高速公路收费站和加油站等设施也增加了对 RFID 系统的支持。另外，中国标准化协会 EPC 和物联网应用标准化工作组正在研究 RFID 在运输业、零售业和能源等产业上的应用，在不久的将来会进行大范围的试点应用[18]。

可以看出，我国对于 RFID 的重视程度是很高的。可以肯定的是在不久的将来，RFID 领域将出现越来越多中国人的身影，我国的 RFID 产业必将在一些方面引领世界的潮流。

1.3 课题的目的和意义

物联网技术的发展推动了 RFID 技术的革新，其中 UHF RFID 将在未来几年掀起巨大的

市场，在物流管理和盘存具有巨大的潜力[19]，但是目前 UHF RFID 的读写器成本还很高，在一定程度上阻碍了技术的普及，一些大公司例如 Impinj，其读写器产品售价动辄数万元。国内的一些企业如远望谷和瑞福科技等公司的读写器产品售价也都在五千元上下。

因此，在保证一定性能的情况下进一步降低读写器的成本将在一定程度上推动 UHF 射频识别技术的普及。读写器系统一般由软件和硬件两部分配合完成工作，由于为降低硬件成本而采取的特殊结构，使得软件系统也需要定制，一定程度将读写器开发的很大比重的工作落在了软件开发上， 所以软件系统的成功将决定读写器的成功与否。

1.4 论文的结构和内容

1.4.1 读写器软硬件开发环境简介

1.4.1.1 读写器硬件开发平台

为了降低读写器的成本，我们硬件平台中，射频前端采用分离器件搭建，同时为了满足超高频无线通信协议（EPC G1C2）的要求，这样将编码、解码和协议流程处理工作交给了微控制器，同时微控制器还承担与 PC 机通信的工作，为了以后便于扩展接口，所以选择具有多种类型接口（比如串口、并口、网口等）的 MCU，同时在解码的时候需要比较高的采用率，协议规定执行命令直接对时间有比较高的限制，基于上述考虑并兼顾价格，我们选择意法半导体的 STM32F207 作为主控芯片，此款芯片的简介如下：

STM32F207 是意法半导体推出全新 STM32 互连型（Connectivity）系列微控制器中的一款性能较强产品，整合了最高工作频率为 120MHz 的高性能 ARM ®Cortex™-M3 32 位 RISC 内核、高速嵌入式存储器(1MB Flash 存储器和 128+4KB SRAM)，集成了各种高性能工业标准接口，且 STM32 不同型号产品在引脚和软件上具有完美的兼容性，可以轻松适应更多的应用[20]。

此款 STM32 的标准外设包括 10 个定时器、三个 12 位 1-Msample/s A/D(模数转换器) (快速交替模式下 2M sample/s)、两个 12 位 D/A(数模转换器)、两个 I2C 接口、五个 USART 接口和三个 SPI 端口和高质量数字音频接口 IIS，另外 STM32F107 拥有全速 USB（OTG）接口，两路 CAN2.0B 接口，以及以太网 10/100 MAC 模块[20]。

1.4.1.2 读写器软件开发平台

RealView MDK 开发套件源自德国 Keil 公司，RealView MDK 是 ARM 公司最先推出的基于微控制器的专业嵌入式开发工具。它采用了 ARM 的最新技术编工具 RVCT，集成了享誉全球的 μ Vision IDE，因此特别易于使用，同时具备非常高的性能。RealView MDK 集成了 uVision4 集成开发环境与 RealView 编译器，支持各类不同核处理器包括 ARM7、ARM9 和 Cortex-M 系列，自动配置启动代码，集成了 Flash 下载功能，具有强大的硬件模拟仿真和性能分析等功能，与 ARM 之前的工具包 ADS 等相比，RealView 编译器的最新版本可将性能改善超过 20%[21]。

Keil 公司开发的 ARM 开发工具 MDK，是用来开发基于 ARM 核的系列微控制器的嵌入

式应用程序。它适合不同层次的开发者使用，包括专业的应用程序开发工程师和嵌入式软件开发的入门者。MDK 包含了工业标准的 Keil C 编译器、宏汇编器、调试器、实时内核等组件，支持所有基于 ARM 的设备，能帮助工程师按照计划完成项目[21]。

1.4.2 本文章节安排

本章主要分为六章：

第一章介绍了 RFID 系统的基本知识，包括 RFID 国内外发展现状，RFID 系统的基本组成和工作原理，最后说明了课题的目的和意义

第二章介绍了超高频 RFID 系统的特点及其工作原理，包括超高频 RFID 遵守的 Gen-2 协议标签的相关知识。

第三章介绍了超高频 RFID 读写器的三种实现方案，通过比较引出了本文所讲述的低成本读写器的设计方案。

第四章着重讲述了低成本读写器软件系统中超高频 RFID 读写器所遵从的 Gen-2 协议的实现，包括防冲突算法的实现。

第五章介绍了读写器软件系统中基带编码和解码方式及其实现方法。

第六章讲述了此款读写器的系统软件的功能性测试和稳定性测试。

最后一章总结全文的内容和本设计的关键结构、技术特点和改进方向。

第二章 UHF RFID 系统简介

2.1 UHF RFID 系统简介

UHF RFID 是指超高频射频识别系统的简称，主要有读写器、标签芯片、读写器天线、标签天线、上位机软件组成如图 2.1。国际上超高频主要工作在 860MHz~960MHz，由于这个频段在各国均被分配为移动通信专用频段，所以频谱资源比较紧张 ，不同国家质检会产生一定程度的频率冲突。我国无线电频率划分和产业发展的实际情况出发，另一方面则是与国际相关标准相衔接，所以 920MHz~925MHz 和 840MHz~845MHz 两个频段。由于超高频射频识别系统工作原理采用的是电磁反向散射耦合的形式，所以有读写器先发言的形式。读写器接收到 PC 上位机发来的指令，经过译码，编码，调制以后通过天线发送出去，在读写器天线有效辐射范围的标签通过整流电路产生供自己电路使用的电流，通过自己的天线解调，解码得到来自读写器的有效信息，进过一套严密的协议通信流程，完成读写器对标签信息的读取。因为超高频工作频段的缘故，电子标签一般做的很小，贴在物品上面。

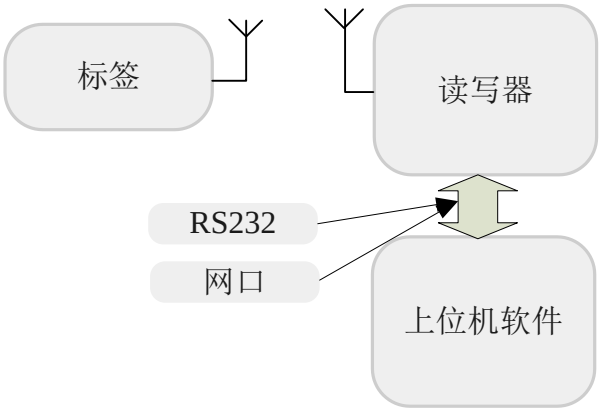

图 2.1 超高频射频识别系统

一般读写器调制方式，编码方式可以选择，因此通过 PC 机可以对读写器的工作方式进行设置。上位机（PC）一般与读写器的接口为串口、网口或/和 USB 接口等。读写器将自己的信息和标签信息通过上述接口传送到上位机界面，可视化界面能够及时人性化的将操作人员所需的信息反映出来。

2.2 UHF RFID 标签

UHF RFID 标签采用无源设计，具体体积小读写距离小，而且成本较低等优点。电子标签主要有标签天线、模拟前端、存储单元、数字基带处理模块等几部分组成[22-24]。现在国际上有好多国家都努力开发标签芯片，国内发展还比较缓慢，主要靠进口满足要求，其实物图如 2.2 所示。

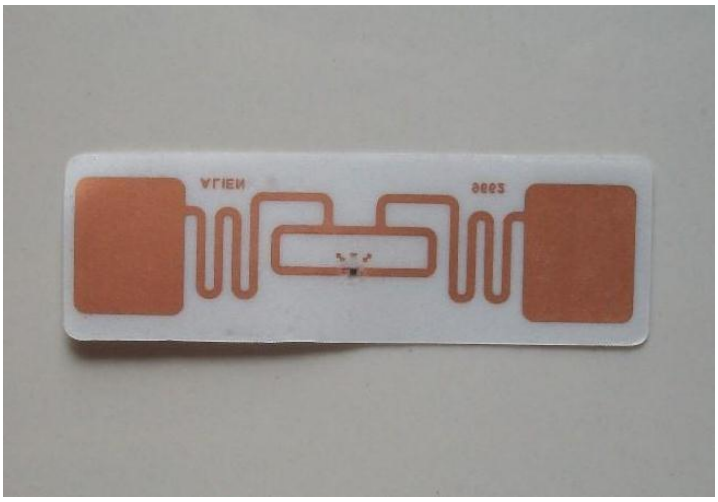

图 2.2 RFID 标签实物图

标签主要有标签芯片和天线两部分组成，标签天线主要有接收能量和反射能量，天线尺寸较小，一般为 1cm~5cm 长。标签芯片尺寸更小，整体面积主要由天线决定。标签芯片内部结构图如 2.3 所示，可以看到其主要有模拟部分、数字部分和存储器三部分组成。模拟部分负责调制和解调，整流为数字部分提供时钟。数字部分主要负责协议处理、编码和解码。存储器部分负责存储标签信息和数字部分处理代码。主要有 EEPROM 和 OTP 等储存技术[25,26]。

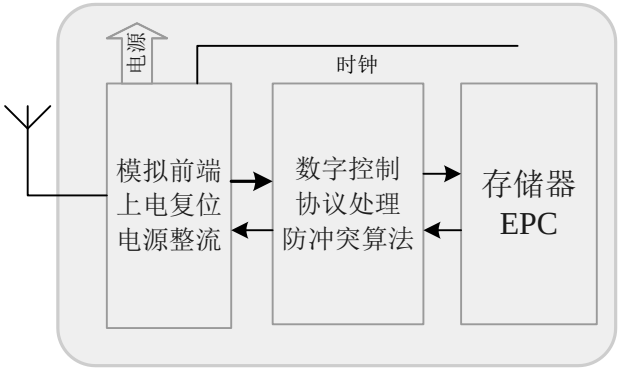

图 2.3 标签内部结构图

2.3 UHF RFID 读写器

2.3.1 UHF RFID 读写器原理

读写器作为 RFID 系统的关键部分，不仅负责与上位机通信，执行上位机发出的指令，通知承担超高频协议处理、产生数字基带信号、编码、解码、载波信号产生等工作。其结构如图 2.4 所示。

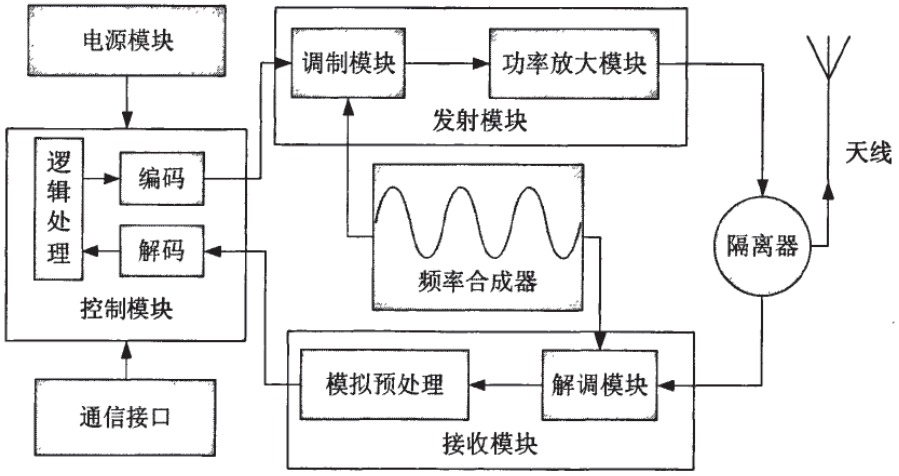

图 2.4 读写器内部结构图

发送部分的基本流程为：

(1) 上位机通过串口或者网口发送指令给读写器，读写器接收到指令后加密后传送到编码模块；

(2) 编码模块对数据进行相应的编码以后发送到射频模块；

(3) 射频模块与载波发生器合作，对基带信号进行调制，发送到放大器模块；

(4) 放大器对已调制信号进行放大处理，然后传输到天线；

(5) 天线模块将电信号转化为磁信号，发送出去。

接收部分的基本流程为：

(1) 标签检测到信号，然后正确验证通过自己天线反向散射能量反射回读写器天线；

(2) 读写器天线检测到由标签返回的能量，将磁信号转化为电信号传输到解调器；

(3) 高频信号进过解调器和滤波器以后，变为低频信号传输到基带处理端口；

(4) 基带模块对信号进行解码处理传输到逻辑控制模块，逻辑控制模块对信号进行验证然后根据协议流程是进行立即回复还是传送回上位机。

2.3.2 UHF RFID 读写器的发展现状

随着射频识别系统远距离无源应用的不断扩大，全球远距离无源读写器的市场也在迅速扩大。全球 RFID 读写器的市场正以 30%的速度增长。据了解，截止 2008 年全球超高频读写器市场规模已经达到 11.7 亿美元[27]。

在国际范围内，随着 RFID 技术将由萌芽状态日渐走向成熟，有关 RFID 的技术也不断创新、同时国际标准的也在日益完善。最早开始制定的关于 RFID 的国际标准是 ISO/IEC 18000 标准组织。他们按频段将 ISO/IEC 18000 标准划分为 7 个部分，目前支持 ISO/IEC 18000 标准的 RFID 读写器最多而且也比较成熟。美国 EPC Global（由 UCC 和 EAN 两大组织联合成立）在吸收了麻省理工 Auto ID 中心的研究成果之后推出了自己的一系列标准草案。EPC Globle 在不断的推广基于 EPC 编码标准的 RFID 产品,发展十分迅速，许多大公司如沃尔玛等都是

EPC 标准应用的坚定支持者。

我国政府已经充分认识到 RFID 产业的重要性，在 2004 年初正式成立了 RFID 国家标准工作组，制定中国自己的 RFID 标准，推动中国自己的 RFID 产业。到 2006 年 4 月底，中国企业也加入 RFID 的全球化标准组织 EPCglobal，同期 EPCglobalChina 也已成立。中国电子标签国家标准工作组正在考虑制定中国的 RFID 标准，包括 RFID 技术本身的标准，如芯片、天线、频率等方面，以及 RFID 的各种应用标准，如 RFID 在物流、身份识别、交通收费等各领域的应用标准[28]。

应用上，读写器芯片有奥地利微电子公司 AS3990 系列、美国 Impinj 公司的 IndyTM R1000（原 Intel R1000）系列和 Samsung 超高频读写器芯片。IndyTM R1000 读写器射频芯片是一款高集成、高效能的 UHF Gen2 读写器射频芯片，芯片采用 0.18μm SiGe BiCMOS 工艺设计，将典型 RFID 阅读器射频百分之九十的部件（包括接收、传递、基带、调制和解调功能）集成到一个芯片上，提供了空前的设计灵活性。能够支持从 860MHz~960MHz 的全频段。三星超高频读写器单芯片供移动电话应用，将射频识别芯片设计到插入手机中的卡式阅读器里，采用 0.18μm CMOS 工艺，结合了前端射频、基频数据机、处理器级记忆芯片。奥地利微电子公司携手系统方案提供商 IDS Microchip，合作开发出面向便携式阅读器市场的新型 UHF RFID 阅读器 IC，AS399x 系列读写器芯片具有高集成度支持多种协议，接口灵活等优点方便工程师二次开发。这些 UHF RFID 阅读器将用于识读第 2 代 RFID 标签，有助于在众多应用领域中实现货物或物品的非接触式识别[29,30]。

国内先施科技自己开发了 UHF RFID 的 S1871 是一款搞灵活度的集成芯片，作为超高频读写器模块它可以嵌入到标签打印机、手持读写器等多种设备中。深圳远望谷也不断开发自己的读写器，成功占领中国市场。广东恒睿 RMU900+UHF RFID 读写器模块也是许多系统集成公司开发考虑的对象。许多研究所高校也在加紧步伐赶上这波浪潮，复旦、清华、北邮等高校都在开发自己的读写器产品。中科院物联网研究所（杭州）基于奥微芯片开发了自己的读写器并购置了一套读写器标准性能验证平台[31]。

2.4 RFID 的 EPC Class1 Gen2 标准简介

无线射频识别（RFID）系统存在众多的通信的通信协议。其中一个最流行的协议就是 EPC Class1 Gen2 协议。这个协议在零售业中得到了广泛的使用。Gen-2 协议在物理层特性和链路层过程中提供了较大的灵活性，以适应不同环境。这个灵活性对于最大化吞吐量或每秒接入/读标签的数量来说，是非常重要的。

EPC Gen-2 标准是 RFID 系统中使用的标准之一。最近，Gen-2 系统被世界批准为 ISO 标准，ISO 18000 的 6C 部分是无源超高频 RFID 系统的一个主导标准。许多零售商，如沃尔玛、麦德龙等使用基于 Gen-2 标准的 RFID 系统[32-34]。

2.4.1 物理层通信特性

Gen-2 协议的物理通信接口与七层开发系统互连（OSI）模型的物理层的概念相似。读写

器控制 Gen-2 协议物理层的所有部分，并编码发给标签的所以命令的前端部分。在 Gen-2 协议中，存在着两个通信链路：读写器到标签的前向链路和标签到读写器的反向链路。这两个链路是互相独立的，存在着不同的数据编码、数据速率和数据调制方案。这两个通信链路的具体特性都有读写器控制。这两个通信链路的具体特性都有由读写器控制。

这可以是读写器根据环境的改变调节通信链路。例如，读写器在射频噪声较大的环境中，可以使用米勒编码来减少标签响应时错误位的数量。又或者是使用想快速获得大量标签的信息时，读写器可以使用读写器向标签链路和标签向读写器链路允许的最快的速率。

(1) 数据速率

Gen-2协议定义了两个通信链路。第一个链路是读写器向标签的链路，它被用来从读写器向标签发送命令。读写器发送命令，随后保持一个载波（CW）。这个载波是没有调制的信号，简化了向标签传输的能力。第二个链路是标签向读写器的链路，它被用来发送从标签回复的数据给读写器。在Gen-2中，标签与读写器通信使用反向散射的形式。

在暴露在射频环境中，所有的天线会吸收环境中的部分能量，并反射剩余的部分。反向散射定义为从任意天线反射的能量。标签具有在两种设置的天线特性之间转换的能力：反射非常少的能量或反射全部能量。

(2) 调制类型

调制定义为如何把数据在物理层面上编码到载波信号上。Gen-2支持两种类型的调制方式：振幅键控（ASK）调制和移相键控（PSK）调制。

如前所述，在Gen-2中有两个独立的通信链路。读写器到标签的前向链路的调制，可以采用ASK的三种调制方式中的任何一种：单边带幅移键控调制（SSB-ASK）、双边带幅移键控调制（DSB-ASK）、相位倒置幅移键控调制（PR-ASK）。

(3) 数据编码

读写器到标签的前向链路采用脉冲间隔编码（PIE）进行编码的。PIE使用两个长度不同的脉冲来代表数据0和数据1。在Gen-2中，数据0的脉冲宽度要比数据1的脉冲宽度窄。

标签到读写器的后向链路采用FM0或Miller两种编码方式。由于具体会在第五章讲编码方式的具体实现中详细讲述编码方式，所以在此不再赘余。

2.4.2 标签识别层特性

在 Gen-2 协议标准中，读写器对标签群访问主要通过三大类命令来完成，图 2.5 所示分别为 Select、Inventory、Access。标签内部有 Ready、Arbitrate、Reply、Acknowledge、Open、Secured、Killed 等七种状态[35]。

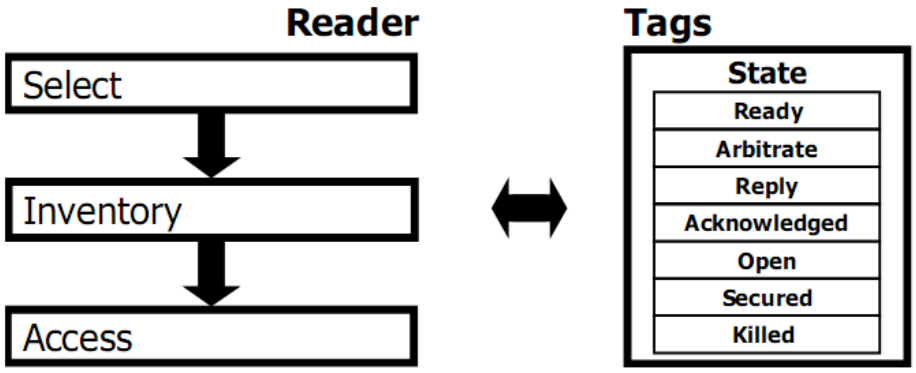

图 2.5 读写器访问标签命令表

Select 命令主要用于确定即将对哪类型标签进行 Inventory 和 Access 操作，对标签群进行分群分类标记。

Inventory 命令组是对标签群进行盘存操作，一个标签经过读写器盘存访问，最终将自己的 ID 回复给标签，其中防冲突算法就是利用 Inventory 命令组的各条命令来实现的。

Access 命令组是对与读写器建立连接的标签进行一系列访问操作使用的，对标签进行读、写、锁、杀等操作。

在第四章中对标签协议进行软件仿真和处理等研究，因此也不再此进行赘余。

2.5 本章小结

本章介绍了超高频 RFID 系统，包括超高频电子标签的工作原理，后半部分重点介绍了超高频读写器的工作原理和发展现状，以及超高频 RFID 协议标准的基本知识，引出下一章读写器方案的研究及其低成本读写器的方案。

第三章 UHF RFID 低成本读写器方案研究

3.1 UHF RFID 读写器方案研究

3.1.1 采用专用集成芯片的读写器

关于 UHF 频段的读写器射频芯片，已见报道的产品有美国 Impinj 公司的 Indy R1000，三星半导体公司的 NXP 读写器芯片和奥地利微电子公司的 AS399X 系列芯片，他们都具有高集成度、高兼容性等特点，对超高频协议具有较好的支持。同时留有多个选项接口，对协议相关的不同调制方式、编码方式都有很好的支持。同时能够根据用户应用场合的不同，能够设计出不同类型的读写器。例如，利用奥地利微电子读写器芯片集成的读写器，如果是手持设备的话要求读写距离较近，这样可以不用添加外部放大器。如果是固定式读写器的话，对读写距离要求比较高，这样可以在原有芯片的基础上再添加外部放大器，这样可以读到很远距离的标准标签[36]。

采用专用集成芯片的读写器大致结构如图 3.1 所示。

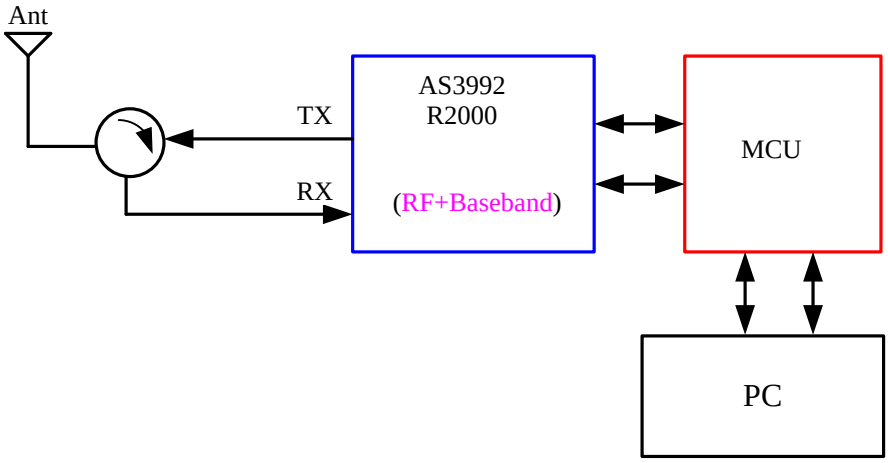

图 3.1 专用集成电路结构图

从图 3.1 可以看出，使用集成芯片的读写器具有结构简单、稳定性高等优点，逻辑控制模块与上位机或者客户端可以选择串口或网口等接口，集成芯片与天线那部分可以采用环形器或者定向耦合器来隔离发送和接收信号。同时根据应用场合的不同，可以选择是否需要添加外部放大器，同时 MCU 与集成芯片也可以选择串行通信或者并行通信。

可以看到，使用专用集成芯片的电路结构比较简单，对于工程师来讲，调试也比较方便，因为编码和解码、调制和解调都有专用芯片来完成，同时部分协议处理工作也由专用芯片来完成，所以 MCU 工作相对比较简单，工程师开发也相对简单。但是，这样缺点就是有点透明度不够，所以对于超高频 RFID 通信协议的支持并非十分全面，尽管芯片设计人员已经尽

可能努力来支持全部协议，但是依然有遗漏的部分。同时更主要的是，这种专用芯片价格非常高，对于市场的拓展非常不利。国际上能够生产专用芯片的就那几家公司，系统集成公司在开发芯片的过程中，由于芯片不透明，所以开发过程过分依赖芯片厂商，这样开发成本大大增加，对于读写器的价格也提出了挑战。

3.1.2 采用通用收发芯片的读写器

考虑到采用专用集成芯片成本的缘故，也有人提出使用通用收发芯片来设计读写器。

目前应用广泛的通用集成收发芯片有 ADI 公司推出的 ADF7020，TI 推出的 CC1100。这些芯片集成度高，应用广泛，成本低，给工程师开发带来很大的空间。但由于这些芯片还不是针对 RFID 通信应用开发使用，因此如果要实现兼容超高频 RFID 通信协议标准的读写器，需要小心的选取和设计[37]。

使用通用收发芯片设计超高频读写器的电路结构如图 3.2 所示。

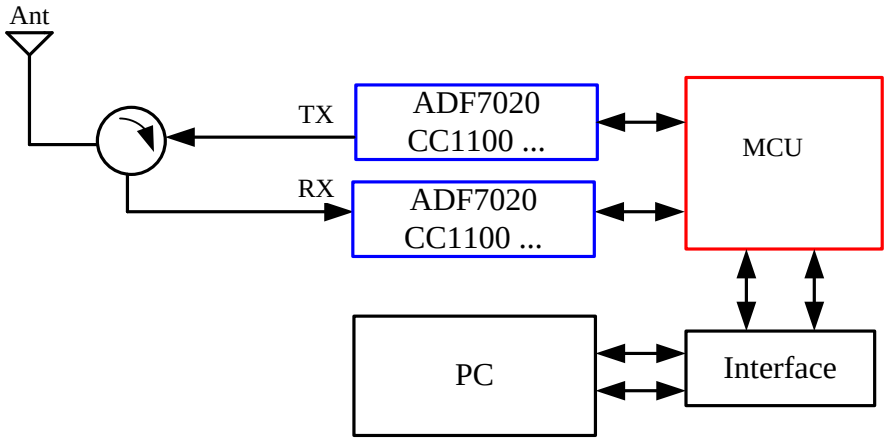

图 3.2 通用收发芯片读写器结构图

从图 3.2 可以看到，使用通用收发芯片设计的读写器电路结构与专用芯片差不多，只是在收发端分开的两路信号与 MCU 连接。这种设计方案由于环行器的泄露问题，而通用射频收发芯片并没有对类似的应用进行优化，故接收端的通用射频收发芯片可能由于接收到的泄露功率过强而无法正常工作。同时，由于通用芯片并未对标准协议进行支持，所以对于通用芯片的选取要求比较严，编码解码工作也需要 MCU 来完成，对 MCU 要求也比较高，对于成本的降低依然没有太好的效果。这种设计只是在文献或者实验室内部讨论中提到过，但是真正产品或者设计成功实例没有看到，因此对于设计一款成功的读写器依然有很长的路要走。

3.1.3 采用分离原件搭建的读写器

使用分离原件搭建的读写器，在射频输出端依然需要环形器或定向耦合器对输入输出信号进行隔离，同时在接收端采用低噪声放大器和混频器的结构。在数字基带部分可以采用 DSP+ARM+FPGA 的组合，这样可以设计出很高端的读写器，无论从协议指令透明、编码解

码还是传输速率等都能很容易满足，但是这样不仅开发周期较长，同时成本依然会很高。因此提出了本文所讲了另一种分离原件搭建的一款低成本的读写器，在没有明显损失读写器性能的情况下，读写器成本大大降低，满足了市场的基本需求。

3.2 低成本 RFID 读写器设计

基于上述几种读写器类型优缺点的比较，我们准备设计一款低成本的读写器，采用前端分离器件搭建的类型方式设计一款低成本的读写器，虽然性能有所损失，同时对协议的某些选项不满足，但是这作为以后开发加强项来讲，不失为一款高性价比的读写器。接下来重点介绍这款低成本读写器的硬件基本平台。

图 3.3 为这款读写器的硬件架构图。

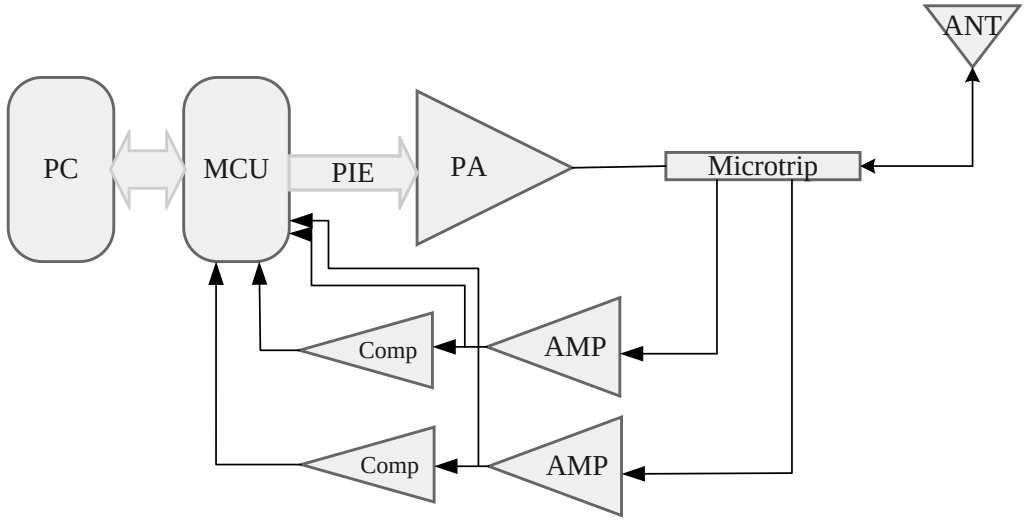

图 3.3 低成本读写器硬件架构图

硬件主要分为发射部分、接收部分、数字逻辑部分、电源部分。发射部分受数字逻辑部分控制，发射数字信号，接收部分将天线过来的高频信号通过接收部分变为数字信号，返回给数字逻辑部分。数字部分负责编码、解码、协议处理、与上位机通信及其协调其他各个部件等工作。电源部分为整个读写器的各个模块提供稳定电源。本文主要介绍的是数字部分的实现。

数字部分是整个读写器的大脑，控制着读写器的接收发送的权利及其各个部件之间的联系，下面介绍各个部件与数字逻辑控制部分的接口或连接：

(1) 射频部分（发射）。

发射部分是读写器向标签发送能量的重要部分，在本设计中发射部分主要由三个功能模块构成，分别是射频模块、射频功率放大器和射频开关芯片。

射频模块产生900MHz的左右的射频信号，是这个读写器的射频信号源。采用射频集成收发芯片，与数字部分通过SPI接口和使能信号接口连接，主要需要数字控制部分对射频集成收发芯片进行配置和使能工作，依据收发芯片的收据手册和配合射频工作组的需求对芯片进行高效的配置，使能信号能够控制收发芯片是否工作，这样可以按照自己意愿对

芯片使能进行配置，对读写器节约能量，越长读写器寿命具有很大的帮助。

射频功率放大器能够使读写器能够提供足够的功率，一般可达到30dBm，另一方面还有调制（ASK）的作用。射频功率放大器通过两个DA管脚与数字控制部分连接，一路控制功率放大倍数，另一路作为调制控制所用[35]。

射频开关芯片作为多路天线选择器来使用，本读写器有四路天线可供选择，所以使用数字控制部分的两个数字控制管脚来选通四路天线。

发射模块大致完成的功能流程为：首先数字控制部分对射频收发模块进行配置（包括频率、功率等），然后产生载波，强度为0dBm，功率放大器将载波放大到30dBm左右，然后通过微带线将射频信号送至射频选择开发，射频选择开关通过数字控制部分的配置，选择一路天线，将射频信号发送出去。

(2) 接收部分。

接收部分的作用是讲射频信号反射回来的微弱信号，经过放大、解调以后送至数字控制部分。接收部分大致可以分为射频前端部分、基带放大部分和整形部分。

射频前端部分采用检波二极管，由于本设计主要针对ASK设计，因此可以采用本设计，在完成混频器的功能之后也将成本大大降低，硬件部分的同事的设计对成本降低起了很大的作用。

基带放大部分的作用是有从标签发射回读写器的能量较小，信号较微弱，为了增加接收灵敏度，同时便于数字控制部分解码，需要基带放大部分对信号进行放大。而数字控制部分可以调节放大倍数，这样可以调节接收灵敏度。

整形部分的设计是为了将放大之后的基带信号整形成数字控制部分（MCU）可以处理的数字信号，由于空间电路对信号有较大的干扰，所以需要加上一个低通滤波器对信号进行滤波然后经过迟滞比较器，这样可以得到数字控制部分处理的信号。

接收信号进过射频前端（即检波二极管）对信号进行解调，同时滤除发射信号的干扰，经过基带放大部分对信号进行放大处理后再对信号进行整形滤波最后得到MCU可以处理的数字信号，然后数字控制部分对信号进行解码和接下来的一系列处理[33]。

(3) 电源部分。

由于各个部件对电源有不同的要求，同时发热对能量的损耗的，所以综合考虑各个部件的需求和性价比，考虑使用线性电源和开关电源结合的方式，作为整个电路的供电系统。

(4) 数字控制部分。

数字控制部分作为本文工作的重点，承担着中心核心的作用，不仅协调各个部件之间的相互联系，同时包括协议处理、编码、解码等繁重的工作都需要数字控制部分来完成，对于低成本读写器的设计起着至关重要的作用。数字控制部分即MCU，主要工作功能：

1) 控制射频收发芯片的配置驱动，同时配置载波频率和发射功率

2) 控制放大器的功率和调制信号，通过MCU内部两部DA转换器来达到对放大器的

控制。

3) 通过两路数字管脚对射频开关进行控制达到配置选通天线的作用。
4) 超高频协议处理、编码、解码等工作都有MCU来完成。
5) 通过控制蜂蜜器，来达到人性化设计的效果，当读到标签的时候，MCU驱动蜂鸣器来通知操作者。
6) 通过以太网接口或者串口与上位机进行通信，将及时信息反馈回上位机，使用户能够实时的看到读写器和标签的信息。

3.3 本章小结

本章通过分析读写器的各个方案，得出了本文设计的低成本读写器方案，通过简单介绍低成本读写器硬件电路模块，对于设计的完成和接下来软件系统的介绍起了至关重要的铺垫。

第四章 读写器软件系统设计及协议研究与实现

由上一章我们知道低成本读写器的硬件考虑尽量节省成本，然后将协议处理和编码、解码环节放到软件处理。软件系统的设计是基于硬件方案考虑的，通过硬件外围接口来完成对MCU 软件的设计，同时兼顾标准协议要求，力求软件系统运行速度快，能够高效稳定的运行，同时为了以后升级方便，软件系统中各个功能函数应该模块化，这样便于后期升级移植。在接下来我们将重点讲述我们的读写器软件系统，其中重点介绍了协议处理部分和防冲突算法部分的代码设计，为读写器完成重要一环。

4.1 读写器软件系统实现

读写器软件系统运行在 MCU 上，软件系统设计尽量考虑层次化、模块化以便于移植修改调试方便。整个软件系统整体方案是在硬件基础上设计得出的，其框架如图 4.1 所示。

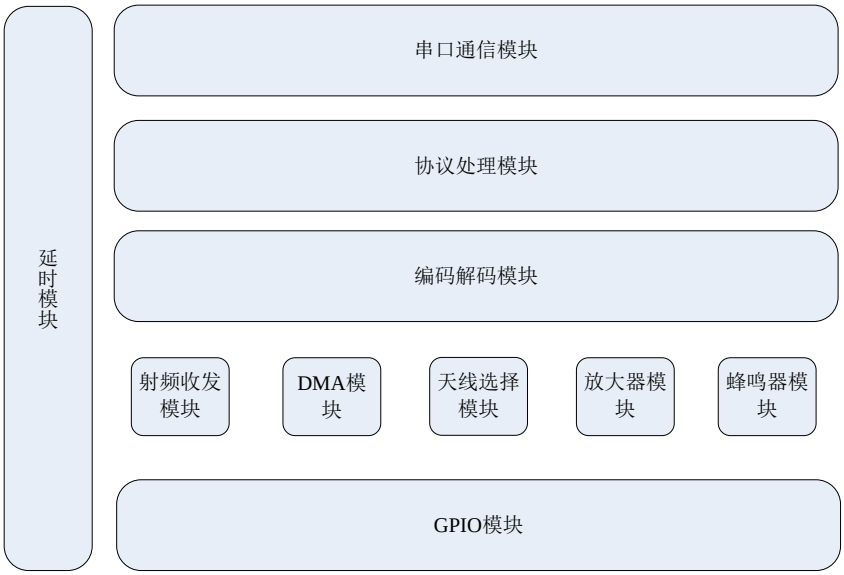

图 4.1 软件系统整体框架图

从图中可以看出，所有模块基于不同的外围组件，不同模块驱动不同外围组件，下面分块介绍不同模块的实现原理。

(1) 天线选择模块

这个模块是基于第三章讲述的射频开关芯片编写的。因为射频开关芯片有四路天线，为了节省 MCU 的 GPIO，以备以后开发所用，我们采用两个通用 IO 进行控制，我们知道两个二进制数组合可以得到四个不同的二进制数，正好满足我们的需求。表 4.1 显示了天线口与控制端对应规则。

表 4.1 天线接口选择与 MCU 的 GPIO 口对应表

Path	GPIO1	GPIO2
ANT-RF1	0	0
ANT-RF2	0	1
ANT-RF3	1	0
ANT-RF4	1	1

由 CC1101 使用手册知道，是通过 4 线的 SPI 兼容接口（SI、SO、SCLK、CSn）进行配置的。CC1101 作为从设备，同时读写缓冲数据的功能。SPI 上的数据都是以一个 header 字节开始，包括一个读写选择位（R/W），突发选择位（burst access），和六比特的地址位（A0-A6）。

按照图 4.2 所示的波形，在进行芯片配置的时候只有 MCU 实现此波形，才能够正确配置芯片，其中 CSn 在数据传输的时候必须保持低电平，我们按照此波形来进行芯片驱动编程，MCU 的四个 SPI 管脚与 CC1101 四个管脚对应起来，延时部分由软件定时器延时程序实现。这部分驱动实现较简单，在此不多做介绍。

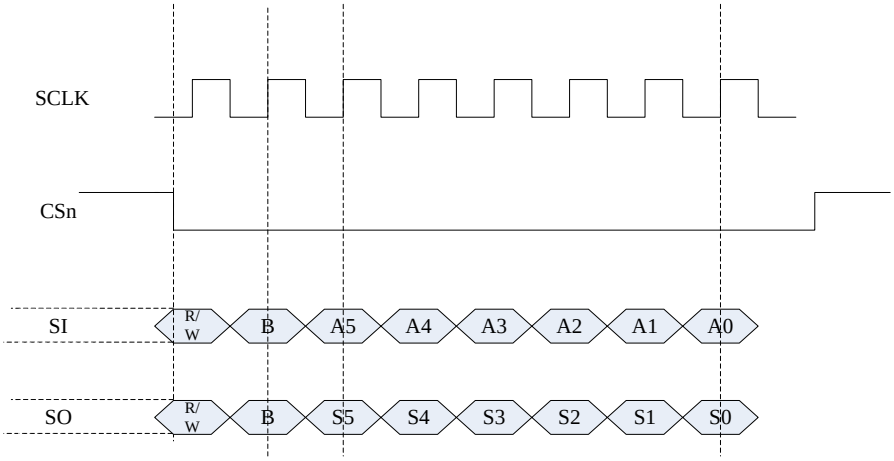

图 4.2 CC1101 配置波形图

(2) 放大器模块

放大器模块软件部分主要有 MCU 的 DAC 实现，DAC 编程依据 STM32 数据手册进行配置，然后根据其转换法则：mv = A×3300/4095，我们选择不同的数据 A，然后 DAC 模块输出不同的 mv，来驱动放大器实现不同的放大倍数。比如 mv = 3.3V 的时候放大器放大倍数最大，mv = 0V 的时候放大器不工作，相当于放大 0 倍。

(3) 蜂鸣器模块

蜂蜜器是友好的声音提示器件。我们采用一路通用 IO 来驱动蜂蜜器。设计了一个函数用于上电提醒和一个用于读写到标签的提示。这两个函数只是频率不同而已。在输出驱动蜂鸣器的方波的时候，高低电平采用通用 IO 的高低电平来表示，不同的频率依靠不同的延时间隔来实现。例如，上电的时候发出短促的“嘀嘀”的响声，蜂鸣器是电平低有效的方式，我们在四路高电平之间通过延时一定时间来实现上电提醒，而在读写到标签的时候，“嘀”一声代

表读写到一个标签，因为在两路高电平之间间隔较长的低电平，每次读到标签调用此函数来进行提醒。

(4) 延时模块

延时模块是整个读写器软件被调用最多的模块，因为调用此延时模块的部分并不需要非常精确的延时间隔，因此我们采用软件定时的方式。因为我们主系统时间为 120MHz，全速运行的时候，估计 1μs 的间隔大概是主系统时钟空循环 120 次，但是因为不同的命令执行周期差别，所以距离自己理想的间隔差别很大，我们利用示波器来尽量实现精确定时，在两个高电平之前，我们实现 1μs 的低电平延时，我们通过不断的调整循环的次数，最终得多循环 100 即为实行微妙延时函数的最精确经验值，但是可能我们带入微妙参数不同，误差会增大，但是由于我们对延时的精确值并未达到 10μs 的地步，因此这个函数还是可以很容易满足我们的要求。

(5) DMA模块

超高频协议对读写器盘存过程中，从标签回复到达读写器的波形的最后一个下降沿到读写器发出下一条命令的第一个上升沿有明确限制，这个也是读写器自己的处理时间，如果时间过长的话，标签认为通信失败，将放弃之前的通信进程，进入冲突等待阶段。所以读写器必须在这个间隔内完成自己的，我们设计解码方式的时候又需要对正交的两路信号进行比较后选择其中的一路进行解码，对两路的比较主要是幅度的比较，因为标签在不同的位置，回复的 IQ 信号幅度具有比较明显的区别，有一路信号质量会比较好，所以我们之前先选择其中一路，基于幅度的比较我们就需要模数转换，即需要 MCU 的 ADC 模块来实现检测功能，而 ADC 检测完这个前导码直到有效区分两路信号又需要比较长的处理时间，这样无形之中是对读写器解码时间的浪费，所以我们使用 STM32 的 DMA 来实现 ADC 功能。

DMA 被成为动态内存访问，它是独立于 MCU 内核的一个模块，使用的时候首先利用 MCU 内核对其进行相应的配置，然后启动 DMA，之后 MCU 主内核可以继续进行别的任务，而 DMA 模块可以按照之前配置进行数据传输，它可以和 SPI、I2C、ADC、DAC 等模块配合使用。在本例中，我们使用 DMA 的 P2M（外设到内存）功能，外设配置为 ADC 模块，将 ADC 采用的数据保存到内存，不仅可以是 MCU 主频从事别的任务，节省了主频的时间，在解码这对时间有特殊限制的模块中具有重要作用。STM32F207 的 ADC 模块采用率达到了 1M/S 的速度，而标签到读写器（T=>R）的后向链路速率基本为 64k/s 来讲，有十几个采用点，可以很好的后向链路波形。

(6) 串口接口模块

因为 MCU 有自带的串口接口，相当于硬件集成了串口，所以只需要将相应的管脚配置成串口模式，然后依据芯片使用手册编写相应的发送接收程序，然后每次与上位机通信调用这些函数即可。但是因为上位机与 MCU 之间通信，一次性会发送较多的信息，因此我们自定义了帧格式，将信息打包发送，这样可以是信息稳定。表 4.2 为数据帧的一般格式：

表 4.2 串口通信数据帧格式

帧头	数据类型	数据长度	有效数据	帧尾

帧头、帧尾由一些特殊符号构成，使它们区别于有效数据，在其中不会出现，比如“#”“&”等；数据类型表示上位机传输的命令时针对读写器配置还是发送给标签的指令；数据长度是相当于同步码，为了防止数据丢失，我们使用数据长度来标识数据总长度，根据数据长度来判断接收的数据是否与帧尾相对应来检测数据完成性；有效数据为实际用于读写器执行的数据。

主函数利用 C 语言 while 循环，等待上位机串口发来的命令，根据数据类型执行相应命令，执行完后再次进入 while 循环等待下一条命令的到来。图 4.3 显示了串口通信模块执行流程。

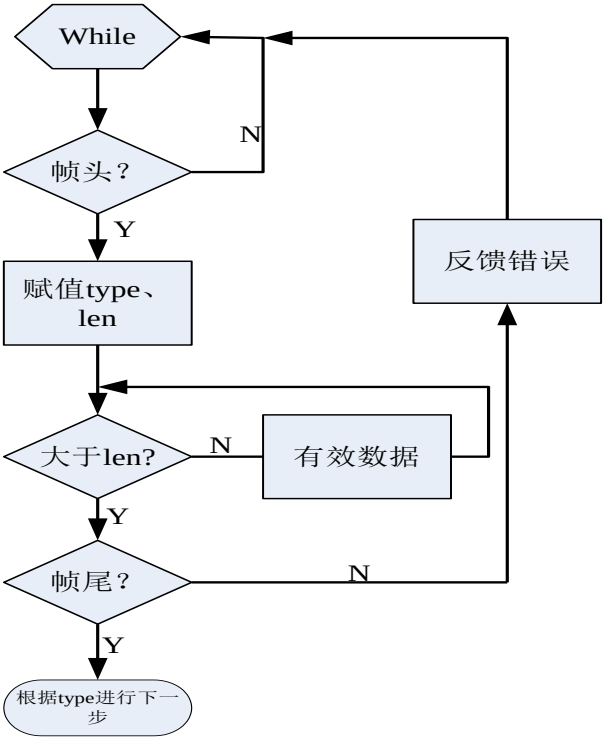

图 4.3 串口执行流程

(7) 协议处理模块

这个部分是整个程序的核心部分，把从上位机通过串口发送过来的数据，经过加工以后发送到编解码模块，当标签有回复过来的数据的时候，先经过编解码模块以后必须经过协议处理或者发送下一条命令到编解码模块或者经过串口接口模块返回给上位机。这一部分将在下一节重点分解讲述。

(8) 编解码模块

这个模块在整个软件系统中处理底层地位，但是也是调试过程中的难点，承担有将整条命令数据转换成编码电平，或者将编码后的高低电平转换成数据的重任，将在下一章的时候重点讲述。

(9) 通用IO（GPIO）

这个部分单独列出来是因为 MCU 的所有接口都可以作为 GPIO，再按照自己意愿使用成相应类型的时候都需要对接口进行一系列的配置。STM32 的 GPIO 选项包括类型选择项，这是指是否配置成 ADC 模式、DAC 抑或 UART 模式等，即使 GPIO 也包括输入输出、是否配置内部上拉下拉电阻、GPIO 的扫描频率、配置为输出的时候是上拉还是下拉等，所以我们编写了统一的配置函数，在对 GPIO 的时候，首先调用 GPIO 接口函数进行相应管脚配置才可以使用，既方便而且层次感强，变化理解。

上面已经介绍了软件系统各个模块的实现，现在我们介绍主体函数，图 4.4 显示了主函数执行流程：

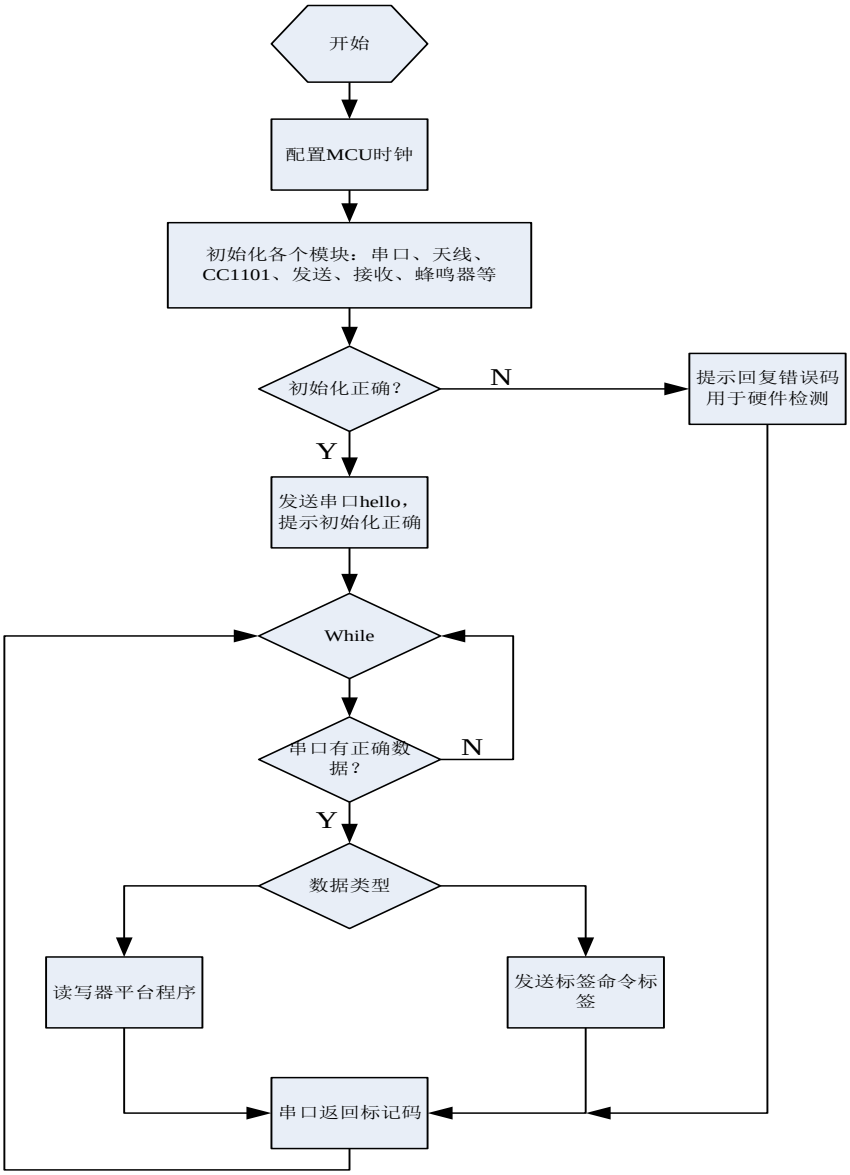

图 4.4 主函数执行流程图

主函数的任务主要配置 MCU 系统和初始化工作。其主要步骤介绍如下：

(1) MCU初始化为其系统初始化工作，主要是配置时钟系统，因为MCU时钟较为复杂，

是整个MCU和软件系统工作的驱动核心，所以时钟配置尤为重要。我们可以选择其时钟的频率，初始化的时候我们选择最高的120MHz；

(2) 模块初始化工作主要包括串口、CC1101、PA、定时器、DMA等上述设计到的各个模块的初始化工作，完成配置以后我们将PA关掉节省功耗；

(3) 如果回复上位机为00h则为提示初始化完成，如果非零则出错。我们定义16比特的错误码，各模块初始化失败错误码各不相同，我们根据错误码信息即可进行硬件基本检查；

(4) 初始化成功后驱动蜂蜜器提示用户程序进程；

(5) 然后进去while主循环，等待上位机发来的命令；

(6) 如果检测到有效数据则开始执行命令，根据命令类型可以分为读写器平台和用于标签通信两类，读写器平台为通信配置模块，包括修改发射功率、载波频率、放大倍数等。标签部分主要为对标签访问，其中包括协议处理、编码等；

(7) 执行完返回上位机信息或者回复标记码，如果执行成功，则返回0，错误的时候返回错误码；

(8) 执行完以后程序跳入步骤5等待上位机下条命令。

4.2 协议处理流程软件实现

超高频协议规定了读写器与标签之间的通信流程，EPC C1 G2（Gen-2）协议规定读写器对标签访问主要通过三大类命令来完成的，分别为 Select、Inventory、Access，而三大组命令里边又包含不同的命令集，由于在介绍协议处理的时候，主要是针对标签的状态和属性的改变，所以有必要先简单介绍一下标签的知识，然后再介绍三条命令，最后介绍他们的软件实现。

4.2.1 标签的相关属性介绍

超高频无源标签，符合 Gen-2 协议的标签本身有四个对话（session），分别为 S0、S1、S2、S3，在每次读写器与标签进行通信的时候，可以且只能在一个对话里进行，不同的标签可以使用不同的独立的对话与读写器进行通信。标签在于读写器进行一次完整的通信的时候必须保持在一个对话里。标签本身具有的四个对话，每个对话又有两个值，A 或 B，这两个值属于非此即彼的关系，比如对话 1 要么具有的值是 A，要是具有的值是 B，属于“且”的关系。同时，标签又具有一个被选择位（Select flag），简称 SL。这个位是独立于前面四个对话之外的一个标志位，它可以与前面四个对话配合起来使用对标签进行标记选择，但是它不能单独作为一个对话来使用，也就是说，四个对话配合一个选择位这种模式来对标签进行标记的[28]。

标签本身有一个随机数发生器，用于产生被称为“Q”的一个值，同时也由于产生一个随机数用于与读写器通信配对的时候使用。标签本身具有一个计数器，当接收到读写器发来的某些命令的时候，标签通过提取读写器发来命令里边的数据包含的有效参数，即 Q 值，利

用 Q 值随机数发生器在 0~2^{Q-1} 之间选择一个值作为计数器的值，然后根据读写器稍后的命令来改变计数器的值，同时在 0~2^{Q-1} 之间选择一个二进制 16 位的随机数值作为 RN16，返回给之前读写器的命令，或者可能作为读写器和标签之间通信的握手信号[37-40]。

标签自身存储着不同长度的 EPC 码，作为自身的标识（ID），在通信过程中会部分或全部发送回读写器。

标签自身具有几种状态，在读写器发送不同命令，标签在不同情况下同时结合自身的情况来改变状态，其状态转换如图 2.5 所示。

4.2.2 协议处理实现

Gen-2 命令集具有相同的属性，都是由命令码、参数表、校验位组成。其中不同命令的命令码的长度不同，四位、八位不等，参数表长度也不尽相同，有的参数较多，有的参数较少，而且即使同一条命令，每个参数的位数也不同，我们全按照二进制来讲，有些参数只有 1 位，而有写参数位数可达到很长，16、32 或者更长都有可能。

全部命令除了 ACK、QueryRep 这两条命令全部都需要循环冗余校验（CRC），其中只有 Query 是 CRC-5 剩下的全部都为 CRC-16。

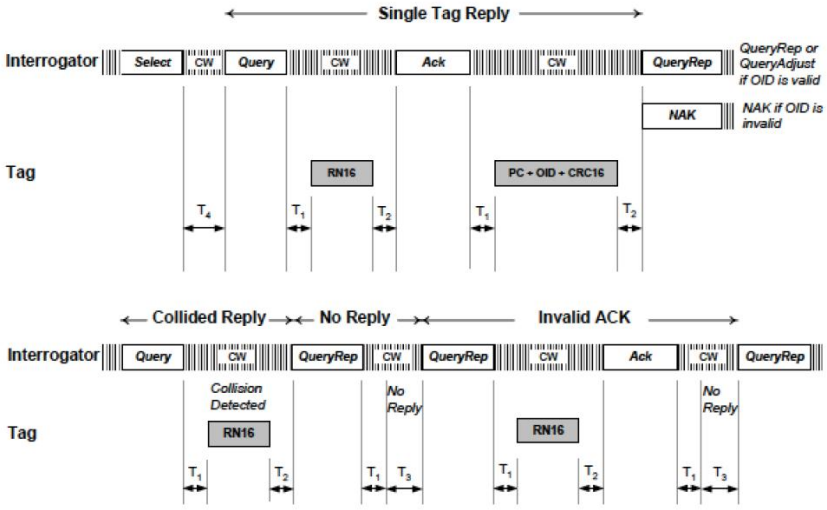

图 4.4：读写器与标签通讯时序图

图 4.4 描述了读写器和标签通讯的时序图。上图是正常的通讯时序，下图是对非正常通讯的处理。首先由读写器发送 Select 命令，发送长度 T4 时间的连续载波后发送 query 命令，经过 T1 时间有标签回复 RN16，再经过 T2 时间发送 ACK 命令，之后再经过 T1 时间标签回复 PC+OID+CRC16，再经过 T2 时间发送 queryrep 命令继续与其它标签通讯。我们在实现协议处理的时候就是按照上述流程来进行编写设计的，同时我们可以看到正因为 T2 实现的限制我们需要对代码的运行时间和速度有特殊的要求[28]。

基于上述不同点考虑，我们设计协议处理程序的时候都需要考虑的因素。

现在来介绍读写器与标签直接的射频通信过程，如图 4.6 所示。其中 Access 命令可以选

择不执行，这样读写器只是对标签的 ID 进行访问，而不对标签的信息进行读、写、锁、杀等操作。其实，读写器可以只是用 Inventory 命令来对读写器进行访问，而不是用 Select，但是这并不是一个好的策略，因为访问效率大大降低，对于一大群标签的扫描来讲，使用 Select 和 Inventory 配合使用可以快速的扫描盘存所有标签。

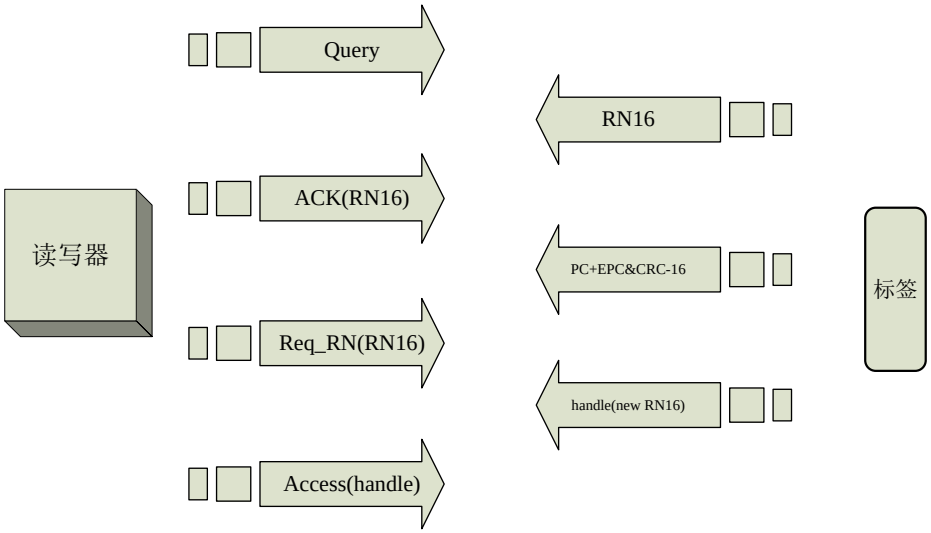

图 4.6 读写器访问标签的交互命令

单个标签的识别过程是这样进行的：首先，读写器发送 Query 命令来对标签进行访问，其中 Query 命令中包含了访问标签的类型，同时包括一个 Q 值，用于防冲突使用，唯一确定一个标签，标签接收到 Query 命令以后，通过 Q 值加载自己的计数器，如果计数器的值为 0 则，标签会立马回复给读写器一个 RN16 的随机数，读写器接收到 RN16 以后，然后发送 ACK 命令，在 ACK 命令中包含了标签发送过来的 RN16，标签接收到读写器发送过来的 ACK 命令以后，首先进行 CRC 校验，然后提取 RN16，与自己之前保存的 RN16 进行对比，如果完全相同，证明读写器正好访问的自己，然后将自己的 EPC 码回复给读写器，完成读写器对标签的盘存过程[3]。

期间标签并不是正好计数值为零，假如不为零，标签不会进行回复，此时读写器需要回复 QueryRep 命令再次访问此标签，标签接收到命令以后将自己的计数值减 1 通过不断的发送 QueryRep 命令直到计数值减为零，标签才进行回复。

协议处理流程考虑的是将上位机发送过来的命令经过协议解释然后发送出去，接收回来信息经过协议解释返回给上位机。为了能够比较清晰的说明协议处理部分过程，我将其分为两层来看，一层被称为命令级，即将每条命令按照协议说明变成二进制序列，这一步完成由上位机程序发送过来数据变成编码可以完成数据这一阶段；另一层是在前一层的基础上根据协议处理流程讲述命令与命令之间的关系，即各条命令如何配合来完成对标签的访问。我们分别以发送和接收两部分的形式来说明第一层，然后在说明下一层的关系。

4.2.2.1 读写器到标签的前向链路

为了便于讲述，我们以 Query 命令为例，说明一条命令是如果最后变为无确定含义的二

进制序列的。图 4.7 显示了 Query 命令的具体参数表及其回复命令。

	Command	DR	M	TRext	Sel	Session	Target	Q	CRC-5
# of bits	4	1	2	1	2	2	1	4	5
description	1000	0: DR=8 1: DR=64/3	00: M=1 01: M=2 10: M=4 11: M=8	0: No pilot tone 1: Use pilot tone	00: All 01: All 10: ~SL 11: SL	00: S0 01: S1 10: S2 11: S3	0: *A* 1: *B*	0–15	

	Response
# of bits	16
description	RN16

图 4.7 Query 命令参数表及其回复命令

上位机是以 1byte 形式发送到 MCU 的，当 MCU 收到上位机发来的命令以后，对帧进行分解，先把帧头，帧尾去掉，按照 Query 命令的形式赋值给其的每个参数。我们对 Query 的控制主要有这几种变量来完成的：

(1) Tx_Map 这个变量是用于说明一条命令里边所有每个参数包含多少bit位的，我们在协议处理文件的头文件里边把所有命令的每个命令具体参数位数，有多少参数列了出来，Query命令来讲其Tx_Map为{4，1，2，1，2，2，1，4，5}，其中第一个数字4代表Query命令码有4bit，1代表第一个参数的位数为1bit。

(2) Tx_Size 这个变量是用于说明每条命令总共有多少比特位。可能有些命令的比特位并不是固定的，这可以在发送的时候根据具体情况来改变。但是基本上大部分命令的位数是固定的。还是以Query来举例，它的SEQ_NUM_Query的位数为22=4+1+2+1+2+2+1+4+5来确定，我们在协议处理头文件里边都具体列出来。

(3) Tx_head 这个变量是因为在每条命令发送前都会有一段前导码用于同步使用，由于不同命令的前导码的位数不同，比如Query命令的前导码位数为7比特，而后面的ACK命令的前导码为5，所以我们采取头文件设置的方式，统一来管理，到时候统一列出来即可。

(4) Tx_fieldSize 变量是用于说明每条命令的包括命令码在内有多少参数位。由于绝大多数命令都有CRC校验位，所以我们在之前值符号以后计算CRC校验位的值，这个参数是命令参数位域的个数减去CRC校验位以后的值。比如Query命令为CRC-5校验，Query命令总共有9个参数位域，命令码是一个参数位域，DR也算一个参数位域，这样Query总共9个参数位域，由于Query是CRC-5所以只需要减去1，而ACK没有CRC校验位，所以就不需要减去，Select是CRC-16，所以需要减去2，然后最后赋值给Tx_fieldSize。我们采取的是预先在头文件设置后参数值，最后需要用的时候，直接调用常量即可。例如我们设置FIELD_NUM_Query = 9，FIELD_NUM_CRC5 = 1；当发送Query命令的时候需要给控制变量Tx_fieldSize赋值，我们只需要使用FIELD_NUM_Query --

FIELD_NUM_CRC5赋给变量即可使用，条理清晰。

(5) Tx_busy 这个是“忙”状态位，我们为了不让读写器前一条命令没有完成然后出现后一条命令，通过“忙”状态位来判断，只有在“忙”状态位为空闲的时候才可以发送下一条命令。

上面这几个变量作为发送一条命令时候的控制参数，通过他们的组合控制后变成一列二进制数。我们以发送一条 Query 命令举例来讲发送部分协议处理的命令。

(1) 发送Query命令时，首先对控制变量进行赋值，包括Tx_fieldsize、Tx_map、Tx_Size、Tx_head；

(2) 对Query进行赋值，将上位机传输过来的数据按照其参数先后顺利赋值给它，为了方便，我们首先会设置一个一个结构体，这样方便赋值，例如struct Query{ u8 Command; u8 DR; u8 M; u8 TRext; u8 Sel; u8 Session; u8 Target; u8 Q; u8 CRC_5;}，其中u8是我们自定义的变量类型符合，代表无符号整型类型的意思，DR与Query命令的第一个参数对应。我们对Query结构体按照上位机传输过来的数据进行赋值，其中CRC_5由于按照塔前面的数据位来计算，所以暂且不对其进行赋值；

(3) 添加CRC校验位，按照协议规定，对命令添加CRC校验位，我们将在下一小节中说明我们利用软件实现CRC校验的方法；

(4) 等待忙指示位变为“清闲”时开始发送Query命令；

(5) 判断发送指示位Tx_seq是否小于Tx_Size（Tx_seq表示Query命令二进制序列的每一位），如果小于进行步骤六，否则Tx_seq归零后进行步骤九；

(6) 判断X是否小于Tx_fieldsize（X表示Query命令参数位域的位置，开始的时候X = 0表示Query命令的命令码位），如果小于继续进行下一步，否则X归零后进行步骤九；

(7) 判断Tx_seq_map是否小于Tx_Map中的第X位的长度（Tx_seq_map表示Query命令中具体到每个参数位域的长度），如果小于则进行下一步，否则将X增1，然后转到步骤六，同时将Tx_seq_map归零；

(8) 通过0x1移位Tx_Size-Tx_seq-1位并与Query值进行“位与”运算，判断其结果是“0”还是“1”来确定Query命令的二进制序列的第Tx_seq位的值。然后是Tx_seq增1；

(9) 最后得到Query命令的二进制系列，用于编码模块对其进行编码。

上面讲述了协议处理发送部分中，将上位机发送来的数据转换成依照协议中命令说明的二进制序列，由于有些命令参数的特殊性，需要在这些地方进行特殊处理，例如 Select 命令，我们从 Select 命令图 4.8 可以看到，它的参数中 mask 比较特殊，因为它并不是固定值，在按照上面步骤处理的时候需要进行简单的处理：

(1) 我们根据上位机发送过来的Select参数中的mask length来分类。将8比特的整数倍添加到Select变量里边，不足8比特的余下的比特添加到后边即可；

(2) 将Tx_Map这个参数中，将mask value这个参数按照1byte步长来划分。即在Select的结构体定义中，将mask value分隔为5个byte；

(3) Tx_Size中根据mask length指示的mask value长度增加到Tx_size；

(4) Tx_fieldsize将mask value长度分无5个byte长度，因为其最长为32比特。我们按照最长来统计然后不够的按照零补齐即可，因为由mask length决定，而且Tx_Size也改变了，所以在实际发送的时候多余的部分会自动舍弃。

	Command	Target	Action	MemBank	Pointer	Length	Mask	Truncate	CRC-16
# of bits	4	3	3	2	EBV	8	Variable	1	16
description	1010	000: **Inventoried** (S0) 001: **Inventoried** (S1) 010: **Inventoried** (S2) 011: **Inventoried** (S3) 100: **SL** 101: RFU 110: RFU 111: RFU	See Table 6.20	00: RFU 01: EPC 10: TID 11: User	Starting Mask address	Mask length (bits)	Mask value	0: Disable truncation 1: Enable truncation	

图 4.8 Select 命令参数表

4.2.2.2 标签到读写器的后向链路

接收部分是将解码部分发送过来的二进制序列转换成有含义的数据。由于不同的命令回复的数据有很大的差别，但是基本转换方法是类同的。

因为协议对每条命令回复格式有具体的规定，所以只需按照格式将一些列二进制序列分隔即可。我们以 read 命令的回复为例来说明接收部分的流程，图 4.9 和图 4.10 为 read 命令预期回复的格式表：

	Header	Memory Words	RN	CRC-16
# of bits	1	Variable	16	16
description	0	Data	handle	

图 4.9“读”的正确时回复命令

	Header	Error Code	RN	CRC-16
# of bits	1	8	16	16
description	1	Error code	handle	

图 4.10“读”错误时的回复命令

(1) 首先从解码部分已经得到二进制序列Rx_buffer，对其进行CRC校验，正确再进行下一步，不正确直接跳出本流程；

(2) 然后0x1以为X位与Rx_buffer做“与”预算，然后X增1；

(3) 根据上一步“与”运算的结果，得到header的数据，如果header为“0”则进行步骤4，否则进入步骤5；

(4) 然后循环连续重复步骤2，次数根据发送read命令时候需要读取多少比特来决定，跳到步骤6；

(5) 然后连续重复步骤2八次，将得到的结果赋值给error code变量；

(6) 连续重复步骤2十六次得到handle变量的值并赋值给它。于是将二进制序列得到了read所有可能的回复数据。

4.2.2.3 CRC 校验的实现

CRC 全称循环冗余校验（cyclic-redundancy check），是数据通信领域最常用的一种差错校验码，其特征是信息字段和校验字段可以任意选择。在标签与读写器通信的时候，由于空间电磁波的干扰或者硬件电路的干扰，可能造成传输过程中信息数据的丢失，假如我们不知道信息是否完整而盲目的进行处理，可能会造成严重的后果，而且通信双方都没有察觉。为了解决这个问题 Gen-2 协议在通信过程中对通信数据进行 CRC 校验，保证了数据的完整性。当读写器一方发送数据的时候添加 CRC 校验位，当标签接收到读写器发送的命令时，首先对其进行 CRC 校验，去除 CRC 校验位的同时，也对数据完整性进行了检测。在实际应用中，实现 CRC 校验的方式各不相同，有硬件有软件，在本系统中，标签更多用硬件实现，而我们为了方便使用软件来实现 CRC 校验位的添加和对接收数据的 CRC 检测。Gen-2 协议规定了两种类型的 CRC 校验：CRC-5 和 CRC-16。我们首先介绍 CRC 校验的原理，然后介绍他们的实现方法。

CRC 是由两部分组成的，分别为信息位和校验位，信息位为需要校验的信息，校验位是依据统一校验依据而添加的校验信息。任意一个由二进制组成的代码序列都可以使用一个系数只能为“0”和“1”的多项式表示，二进制代码序列值与多项式一一对应。例如代码 100110 对应的多项式为 x^5+x^2+x，而多项式 x^6+x^3+x 对应的代码为 1001010。CRC 码集选择原则为：若代码字的长度为 N，信息字段为 K，校验字段为 R（N=K+R），则对应于 CRC 码集中的任意任一码字，存在且仅存在一个 R 次多项式 $g(x)$ 使得：

$$V(x)=A(x)g(x)=x^R m(x)+r(x) \tag{4.1}$$

其中：$m(x)$ 为 K 次信息多项式，$r(x)$ 为校验多项式，$g(x)$ 为生成多项式，发送方通过指定 $g(x)$ 产生 CRC 校验码，接收方则通过该 $g(x)$ 来对接收到的 CRC 校验码进行校验。

CRC-5：其生成多项式为 x^5+x^3+1，初始化代码值为 $(01001)_2$，长度为 5 比特。

CRC-16：其生成多项式为 $x^{16}+x^{12}+x^5+1$，初始化代码值为 FFFFh，长度为 16 比特。

CRC 校验是用信息码去除生成多项式，余数即为校验码，在二进制运算中，模 2 除就是模 2 加，而模 2 加就是我们熟悉的异或运算。我们在软件实现 CRC 校验的过程中就是依靠模 2 运算来最终得到校验位，为了方便我们以 CRC-16 来说明其基本方法：

(1) 预置CRC码为$FFFF_h$，信息码长度为len，生成码为1021_h；

(2) 判断slot是否大于len，slot初始值为0，移动检验信息码的每一比特位；

(3) 0x1移位len-slot-1位并与信息码做“异或”运算，如果结果为“0”则进行第5步，

如果为“1”第4步；

(4) CRC码与1021_h相“位与”，得到结果赋值与CRC码；

(5) slot自增1，然后跳到第2步；

(6) 最后将CRC码取反得到的值即为最终校验码。

其程序流程如图 4.11 所示：

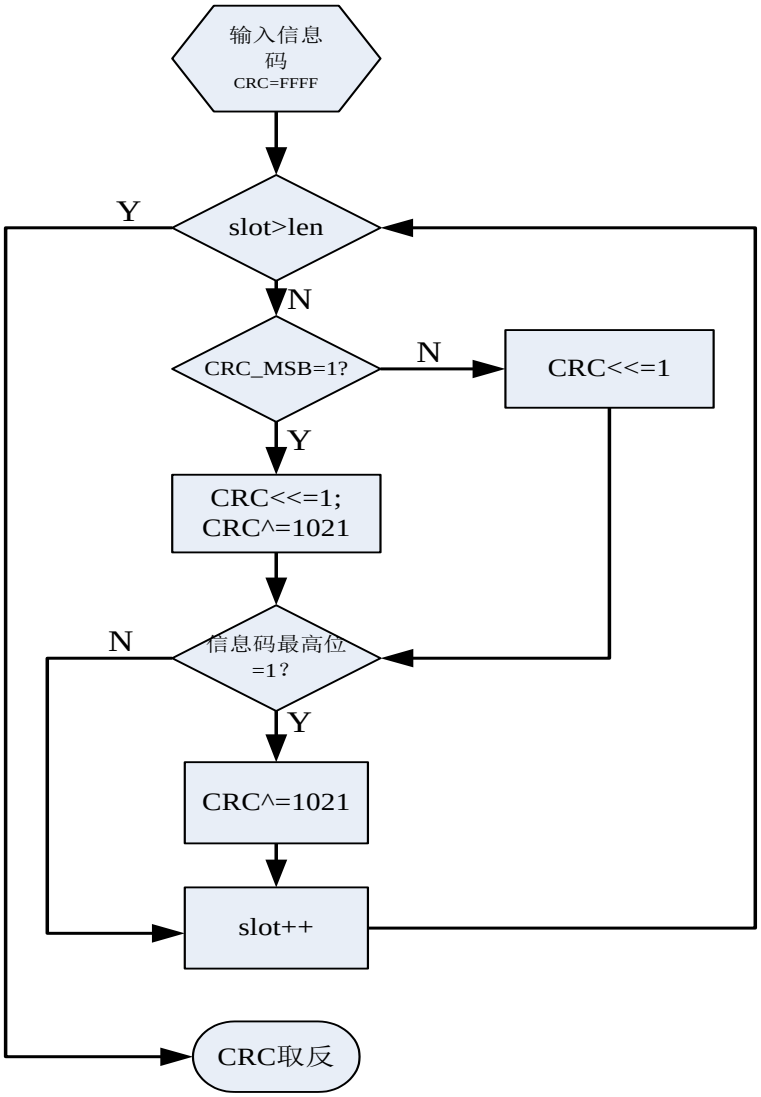

图 4.11 CRC 校验流程图

4.2.2.4 Gen-2 协议执行流程

我们编写协议处理流程主要是根据协议的要求来设计的。协议的规定是金标准，实现力求全面而且稳定。我们以流程图的形式来介绍协议处理流程。图 4.12 显示了 Query 命令实现由发送到接收的流程图：

(1) 首先，Query命令各个参数配置完成，调用上述发送部分的方法，将各参数转换成二进制序列，并且添加CRC校验位，用于解码；

(2) 检测读写器发送是否正处于Busy状态，如果“忙”则一直等待直到读写器发送模块

变为空闲状态为止，执行下一步；

(3) 启动定时器翻转功能，定时器开始执行解码；

(4) 解码完成后，配置接收模块，包括清除发送Busy状态位、接收循环缓冲数组、接收序列预期值、通道选择、超时检测等；

(5) 启动超时检测程序；

(6) 执行解码部分程序，如果累计缓冲数据超过最大值，则超时，跳出解码，否则完成解码程序；

(7) 对解码部分得到的序列CRC校验部分进行CRC校验，如果正确则返回0，如果错误则返回错误码；

(8) 将解码得到的二进制序列进行解析，即执行上述讲述接收部分的程序；

(9) 执行下一条发送命令或者返回信息给上位机。

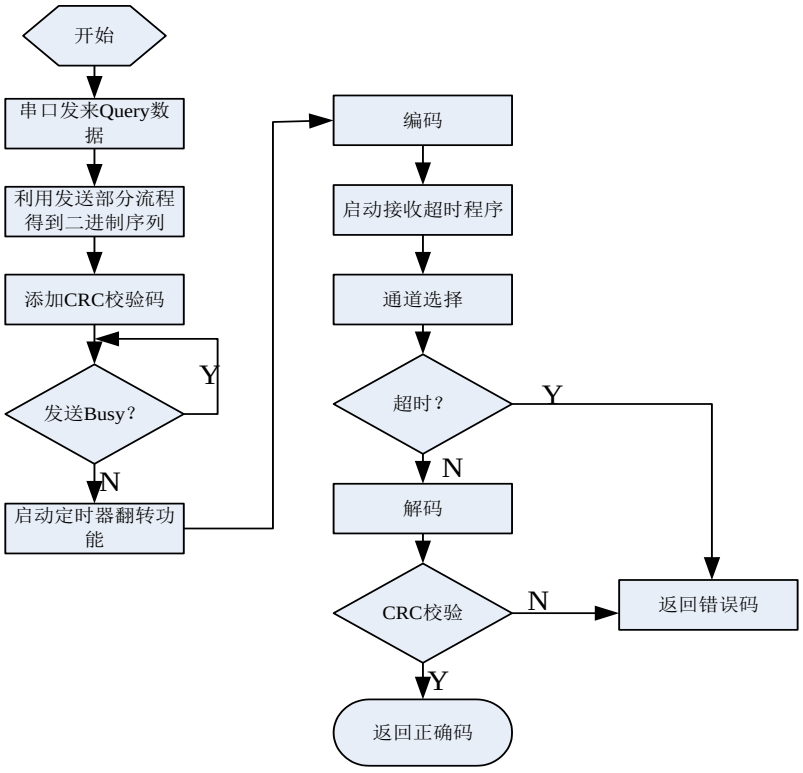

图 4.12 单条命令发送接收流程图

上面介绍了单个命令的执行流程，标签访问是由多条命令组合而成，最终完成对标签读写或者盘存，其中 Inventory 命令是对标签的盘存，得到标签的 EPC 码，同时标签进去 Open 状态，标签进入 Open 状态以后即可进行读写等操作。在协议里边并未对标签写入流程做具体规定，只是说明了写入标签所有的状态，这个与之前 handle 有关，我们认为，只有选中写入标签，然后才可以进行写入操作，所以在执行标签写命令的时候，通过之前的盘存命令，得到一堆标签的 EPC，而此时 Select 命令的作用发挥出来了，我们知道 Select 命令中 mask value 是关于利用标签内存中存有的数据，然后读写器发送这些数据作为选中标签的参数，标签接

收到命令以后，利用 mask value 数据与自己的内存中数据进行比对，然后改变自己 S0~S3 中的一个会话的标记又 A 跳到 B 或者由 B 跳到 A，改变 SL 的标记，然后通过 Query 中携带的会 Sx 会议参数和 SL 标记位来选中这个标签，然后进过 Inventory 命令组和 Req_RN 标签才进入 Open 状态。所以我们是每次标签读写之前都会再进行一次 Select 和 Inventory 命令，这样也可以避免由于外界干扰，比如在 Write 之前标签已经不在读写器辐射范围内，而使我们错误的任务 write 命令执行的错误。下面我们讲述读写器由盘存到写标签的执行流程，其涵盖了整个协议处理过程，其他命令与其过程相符，执行由上位机发送过来的执行指令不同而已。

(1) 在进行写命令之前首先需要盘存标签，使标签进去 Open 状态。在写之前我们先对一堆标签进行盘存命令得到他们的 EPC 码；
(2) 然后我们选中进行写的标签，因为每个标签都具有唯一的 EPC 码，所以通过选择 EPC 码，来选中要写入标签；
(3) 然后执行 Select 命令，我们根据 mask value、mask length 等参数来对标签进行分类；
(4) 发送 Query 命令，根据之前 Select 命令的执行的条件，将相应参数添加给 Query。因为当标签有上百张的时候，不可避免非我们选中的标签依然会响应读写器，造成冲突，此时，程序执行 Select 命令，将辐射范围的所有标签 S 参数和 SL 变成统一的别的标签，然后再执行一次 Select 命令，即可实现唯一标签的选中。其中，再次执行的 Select 命令参数中 mask length=0，由协议规定，当 mask length =0 的时候相当于选中所有标签，他们的标记位都会改变，这样直到没有冲突发生选中唯一的标签；
(5) 然后将返回的 RN16 带入 ACK 命令，到底标签的 EPC 码；
(6) 发送 Req_RN，经过两次之后，标签才能进入 Open 状态，而有协议知道，最后一次返回的 RN16 即为 handle，称为握手信号，作为 Access 命令组使用；
(7) 执行 write 命令，然后等待返回信息；
(8) 经过解码，解析等程序后得到应答信息，首先检测回复中 CRC 校验位是否正确，然后判断 handle 是否正确，如果错误则认为通信中出现错误，程序执行失败；
(9) 然后根据 write 返回的信息的头一位 header 来判断 write 命令是否执行成功，如果 header 为 1，则认为程序执行成功，只是 write 写入的内存的位置不对或者写入数据太长等。如果 header 为 0 则标签写入成功，返回给上位机。

图 4.13 显示 write 命令执行流程图。Read、lock、kill 命令与 write 命令相同，因此只需按照此执行流程即可对标签进行 Access 命令组范围。而 Inventory 命令组即为 write 命令组的前半部分，只是在检测到冲突的时候，执行防冲突算法即可。

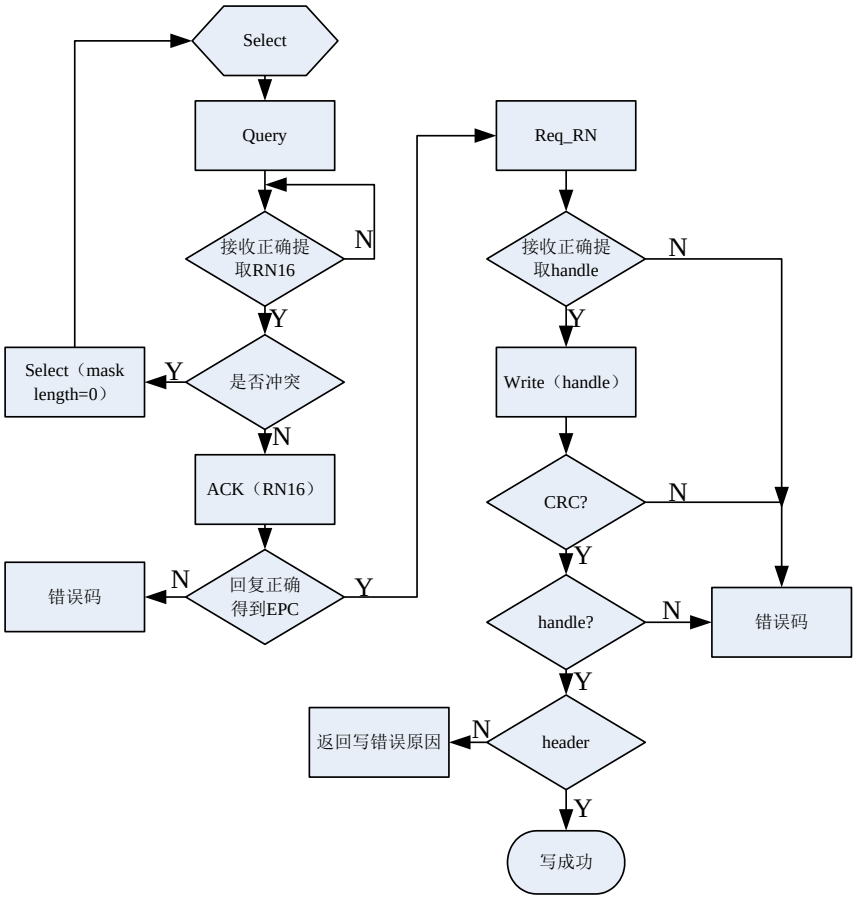

图 4.13 写命令执行流程

4.3 标签防冲突算法实现

在无线射频识别（RFID）系统中，标签存储这识别号基本都是唯一的，并且粘贴在物体上。像其他无线通信系统一样，RFID 系统也存在着信号干扰问题。其中主要有两类信号干扰，一类是读写器与读写器之间的信号碰撞问题，另一类是多个标签之间的信号的碰撞问题，由于我们现在研究的主要是单个读写器的问题，所以主要研究的是后者，标签与标签之间的信号干扰问题[38]。

在读写器发出访问指令的时候，在读写器射频范围内，可能存在多个标签，这样读写器可能会同时多个读写器的回复，但是读写器某一时间只能够与其中的一个标签进行通信，这样就有防冲突问题。为了减少标签之间的碰撞，多个标签防冲突的协议被提出来。这些协议主要包括：基于 ALOHA 的、基于计数器的协议和基于树的协议。而我们读写器采用的是基于 ALOHA 协议的[38-40]。

ALOHA 协议是最简单的基于 ALOHA 的标签防碰撞协议。当读写器发出请求指令的时候，在识别区内的每个标签都会自动选择一个回退时间，在这个回退时间后，标签再把自己的 ID 回复给读写器[41]。如果在标签发送自己 ID 到读写器期间没有发生与别的标签的碰撞，那标签能够成功发送自己 ID 到读写器，否则，标签只好重复的选择一个随机的回退时间发送它的 ID，直到 ID 被读写器识别为止。

时隙 ALOHA 协议中，随机的回退时间必须是多个预先设置的时隙时间。需要注意的是，时隙时间经常被设置成一个时间周期，这个时间必须足够长，不仅要标签在这个时间内可以发送完自己的 ID，同时也要保证读写器能够识别到标签的 ID。这要求读写器为识别区内的所有标签同步时隙时间。在一个时隙时间内，只能够有唯一的标签传输它的 ID，这样读写器才能够识别到标签，否则会产生冲突，没有被识别到的标签将会再次选择一个时隙来发送自己的 ID 号。据文献报道，时隙 ALOHA 协议的性能是纯 ALOHA 协议性能的两倍[42]。

帧时隙 ALOHA 协议，整个识别过程被分成一系列的帧，每个帧具有多个时隙。在标签接收到读写器的请求命令时，每个标签在某个帧期间选择一个时隙来发送自己的 ID 到标签。帧时隙 ALOHA 基本与时隙 ALOHA 协议相同，都需要同步时间。但是帧时隙 ALOHA 协议有一个缺点，在标签数大于时隙时间的时候，读取标签的时间会大大增加，标签数远小于时隙时间的时候又造成浪费[38-42]。

由于上面的问题，有人在帧时隙的基础上提出了动态帧时隙 ALOHA 算法，每帧的时隙个数是动态产生的，本设计中我们采用的就是动态帧时隙 ALOHA 算法[41]。

我们利用标签内部计数器以及利用读写器发送过来的 Q 值来产生一个回退时间，因为 Q 值是确定的，同时我们可以依据标签回复过来的信息可以动态的调整下次盘存标签的 Q 值，所以动态帧时隙 ALOHA 算法很好的满足了我们的要求，在代码调试过程中，通过我们测试，过程中，六个标签可以很容易在 1s 时间内读写到，图 4.14 显示了利用动态帧时隙 ALOHA 算法实现的标签防冲突算法的流程图。

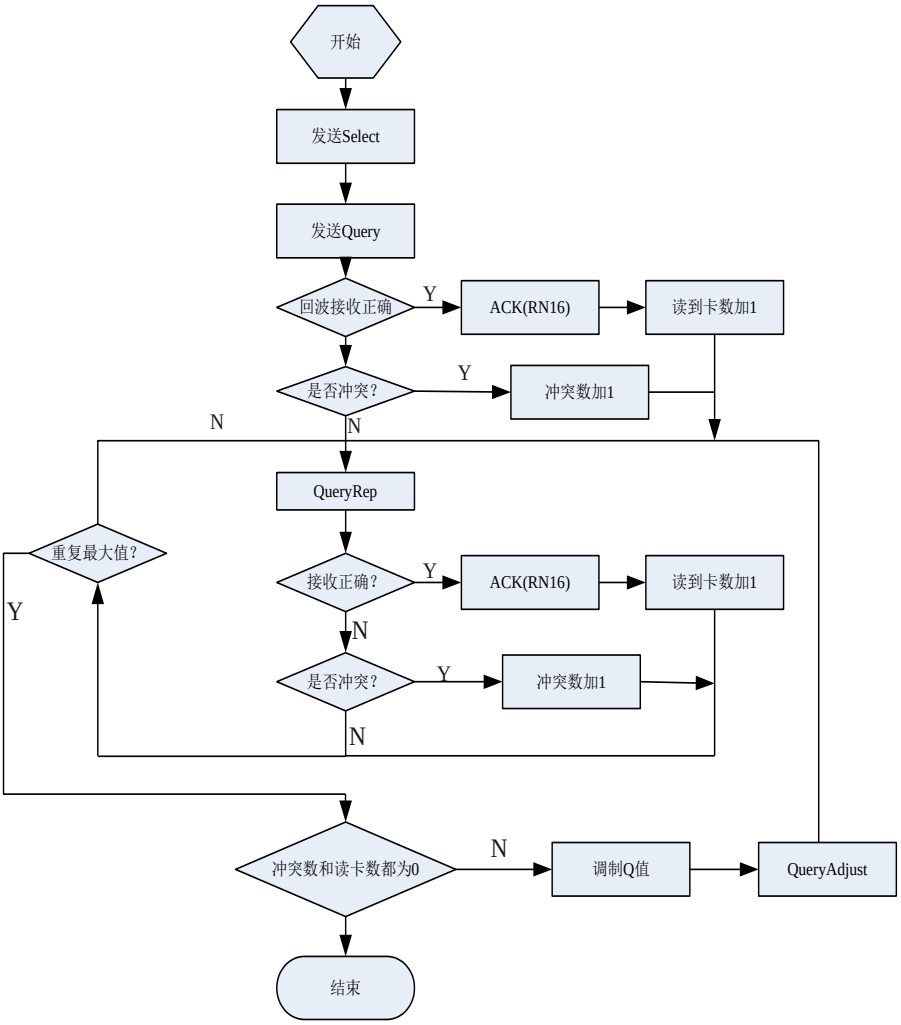

图 4.14 防冲突算法流程

4.4 本章小结

本章介绍了软件系统除编码、解码模块以外各个基本模块的实现原理，及其流程图。通过第六章的软件测试，证明我们实现方法很好的完成了协议物理层要求和用户层要求，可以稳定的读写到多个标签。

第五章 UHF RFID 读写器编解码的实现

5.1 协议规定编解码方式简介

协议对编码/解码方式做了具体规定，发送采用 PIE 编码，接收采用 miller 编码。其中 miller 编码分为 FM0、miller2、miller4、miller8。他们编码方式类似，知识数据速率不同而已，我们后向速率支持 80KB，所以我们在此选择介绍 FM0 编码方式。本小节先分别介绍这两类编码的理论知识，后边两节介绍具体实现。

5.1.1 PIE 编码方式简介

在前向链路（I=>R）即由读写器到标签的方向规定是 PIE 编码。有不同长度的高低脉冲代表不同的二进制数，有短一点的数据间隔代表数据 0，由长一点的数据间隔代表数据 1，并且数据 1 的总时长应在 1.5 到 2 个数据 0 总时长之间，其中 Tari 是读写器到标签的参考时间间隔，是数据 0 的时间间隔，其格式如图 5.1 所示。高电平由发送的未衰减载波表示，低电平由衰减的载波表示。脉冲宽度 PW 对于数据 0（data-0）与数据 1（data-1）是相同的[28]。

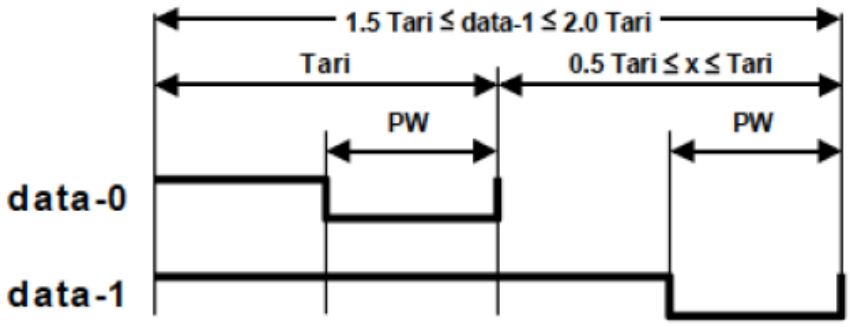

图 5.1 PIE 编码波形图

读写器通信使用的 Tari 值在 6.25 μ s 到 25 μ s 之间。

另外，在发送的时候会有一段前导码用于读写器与标签之间的同步使用。分为前导码（preamble）和同步码（frame-sync）。其中前导码用于发送 Query 命令的时候，它作为盘存命令的起始命令，是整个盘存命令的开始。其他所有的命令都有同步码开始。前导码应该包括一个固定的定界符、数据 0、数据 1、读写器到标签的校准信号（RTcal）和标签到读写器的校准信号（TRcal）。其格式如图 5.2 所示，它们都有编码模块来实现[43]。

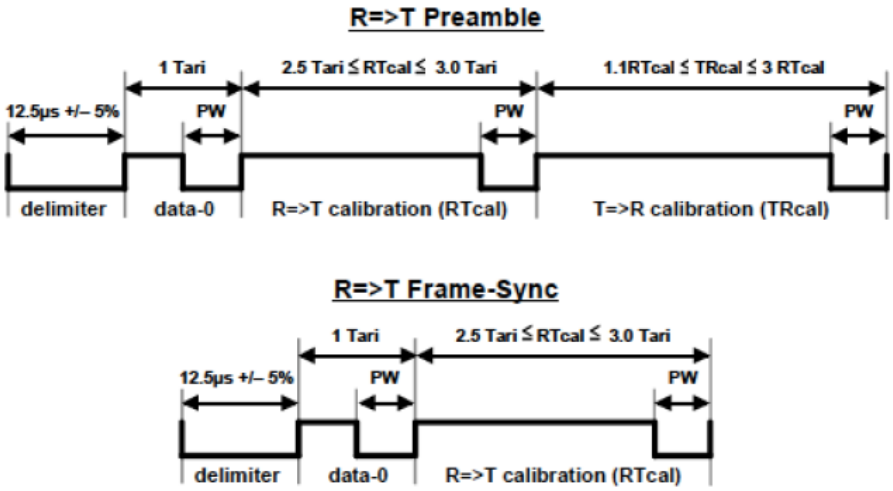

图 5.2 前向链路前导码和帧同步

RTcal：读写器可以设置长度 RTcal=0length+1length。因此，标签可以通过测量 RTcal 计算 pivot=RTcal/2。现在标签将其解释为数据 0 数据 1 比 pivot 符号较小。此外，标签认为大于 4 RTcal 的数据为错误数据[44,45]。

TRcal：一个 TRcal 长度是由读写器选择指定链接低频（LF）。标签使用 LF 确定其 FM0 的数据率。 LF 的计算见式（5.1）其中 DR 是由标签制造商指定的。

$$LF = \frac{DR}{TRcal} \tag{5.1}$$

5.1.2 FM0 编码方式简介

FM0 编码的全称为双相间隔码编码。射频标签通过调制来自读写器的射频能量，将之反向散射，从而将信息传送回读写器。

FM0 编码的信息格式：图 5.3 分别显示了 FM0 编码的基本格式和生成状态图。FM0 在每个脉宽周期倒转基带相位，而数据 0 脉宽中间会有附加相位倒转。S1-S4 状态标记代表四种可能的 FM0 编码格式，代表个 FM0 可能出现的基本格式的相位关系。这些状态标签之间的数字代表当传输波形由一个状态转换到另一个状态时，需要键入的数据类型（data0 或 data1）。其中，从状态 S2 转换到状态 S3 是不允许的，因为由此产生的传输在符号边界上没有相位转化。由于 FM0 编码序列选择依赖于先前比特波形，由此 FM0 编码需要存储器来存储前一编码比特的波形[46-48]。

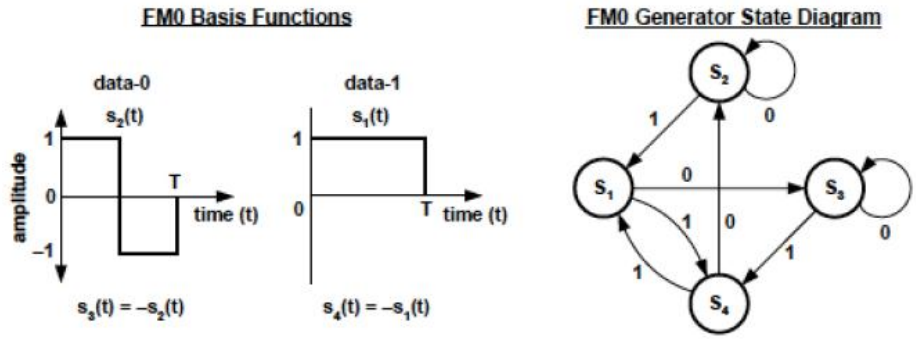

图 5.3 FM0 编码波形图及状态转换图

Gen-2 协议规定有标签到读写器有一段前导码用于标签与读写器之间的同步使用。依据 Query 命令中的参数 TRext 来选择哪种方式的前导码，因为我们希望可以准确的把握到反向散射信号的到来，所以我们选择 TRext=1 方式编写前导码。它是有 1010V1 做为前导码开始，可以看到 V 作为反 FM0 编码，可以很好的作为反向散射的波形的起始的检测标记，其格式如图 5.4 所示[48-50]。

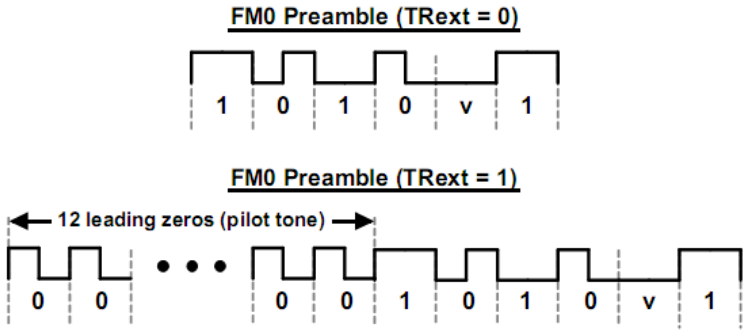

图 5.4 后向链路前导码波形图

5.2 PIE 编码（编码部分）的实现

我们要实现 PIE 编码，从 PIE 编码的特性着手，一 PIE 编码是高低电平交替间隔出现的组合同时高低电平交替时候要求脉冲上升沿下降沿时间不高于参考时间 Tari 的 1%，根据这些特性分析我们用 STM32 的定时器翻转模式来实现 Gen-2 规定的 PIE 编码。

我们知道每个数据 PIE 编码都是由高低电平组合出现的，其由不同长度脉冲宽度来代表不同的二进制数据，同时调制的时候要求脉冲宽度误差不大于 1%，为了方便我们以 Tari 值为 25μs 来讲，我们知道 25μs×1%=0.25μs，而 STM32F207 的定时器很好的满足了这个要求，120MHz 的分辨率大约为 0.001μs 左右，然后我们通过配置 MCU 定时器，即使 Gen-2 协议的最高标准我们 MCU 定时器也可以很好的满足[51,52]。

PIE 编码特性就是高低电平交替出现，即下一个数据起始电平由上一个数据的结束电平来决定的，同时由于总是高电平在先，低电平在后，且低电平的脉冲宽度相同，由高电平的脉宽来决定是数据 0 还是数据 1。我们知道 STM32 定时器不仅有定时计数功能，而且还能够

输出方波，只需要修改输出脉冲的宽度，然后加上定时器的翻转功能即可实现高低电平交替改变的特性。

我们实现数据 0 和数据 1 时，只需要设置定时器间隔即可。我们要求低电平脉冲宽度为 T1，而我们定时器的时钟周期为 T2，我们需要设置低电平计数值 C=T1/T2，就可得到精确的脉冲宽度[53]。

在发送每条命令之前，都会有前导码或者同步码用于读写器与标签通信时候同步使用，而且它们开始都是低电平，所以我们首先设置一个无关脉冲，同时定时器默认翻转开始为高电平，这样下一个电平标准前导码或者同步码可以实现标准低电平。同时前导码高电平脉冲宽度与数据 0 和数据 1 不同，我们需要预先设置同步码或前导码值 C 列表，在每条命令发送之前加上它们即可。

图 5.5 为编码数据执行流程。我们在上面基础上做了一个小变动，可以使我们编程简介很多，我们知道每个数据都包含高电平低电平，低电平的值相同，在所有数据序列中我可以认为奇数位为低电平，所有偶数位为高电平，这样是数据检测变得方便，而把前导码或同步码那部分列表中间隔着把每个低电平的相对值加上去即可。

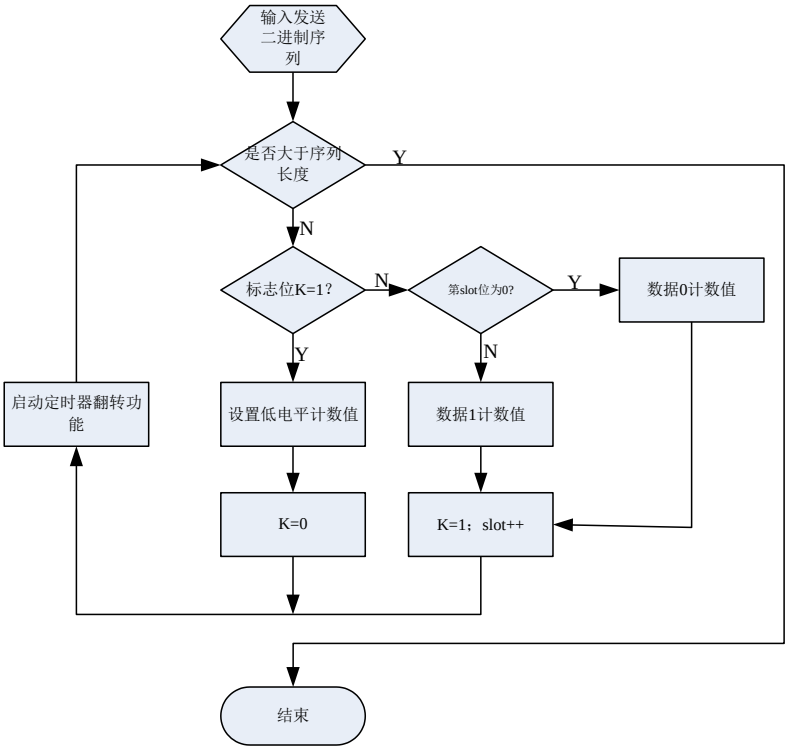

图 5.5 PIE 编码实现流程

5.3 FM0 编码（解码部分）的实现

接收部分数据 0 和数据 1 是根据脉宽中间是否有相位倒向来区分的。而由于电路和空间电磁波的干扰，会使得波形毛刺比较多，使得波形并不是那么完美有时候还会比较差，所以我们通过调制检测值，得到经验值使解码尽可能得到正确的数据。弥补硬件的缺失。解码部分我们主要包括通道选择、实际解码、超时检测、CRC 校验等四部分，CRC 校验由协议处理

部分介绍，在此不再赘余。主要介绍前三部分，特别是第二部分的内容。

通道选择是根据硬件提供的条件，我们可以选择两路中的一路来进行解码。在硬件电路中信号再通过微带线的时候分成 I、Q 两路信号，它们是正交的两路信号，由于标签在空中相对于读写器天线的位置不同，高频信号通过类混频器功能的微带线时候分为 I、Q 两路信号，导致这两路信号强度有所不同，但是这两路信号的数据一样，波形相反，强度不同，同时我们为了节省 MCU 资源提高时间效率我们只需选择其中一路来进行解码即可，所以在实际解码之前，根据他们的幅度大小来选择其中一路。我们利用 MCU 的 DMA 功能和 DAC 功能来实现，DMA 和 DAC 的特性在第四章第一小节介绍过，我们再次不在介绍。当 DAC 收集到一定数量 K 的数据的时候，K 的选择比较关键，它不仅能够要求这个数量的数据能够充分的区分出两路信号幅度的差别，而且不影响正常有用数据的解码，通过实验同时结合在读写器发送数据到标签回复数据这个时间我们得出了经验值为 K，通过计算这个值范围内波形的幅度可以准确的选择其中一路用于解码。对于 K1 值的 DAC 采样的各个点的值，我们不同简单的相加进行比较，依据统计学的知识，通过求两组数据的均方差来作为最后的比较数据，这样可以消除采样数据中突然超高的毛刺导致的影响。其流程为：

(1) 我们先收集两路信号K1个DAC采样值x_1、x_2……x_k，计算它们的平均值X。

(2) 然后利用（x_1-X）+（x_2-X）+……+(x_k-X)得出两路数据的均方差。

(3) 通过比较两路信号的均方差，来选择其中一路作为最终的解码数据。

超时检测是当我们长时间接收不到有效信号的时候，我们认为没有标签响应，我们应该结束解码，协议中 write 命令的接收数据与读写器发送数据的间隔时间最长，在此基础上我们增加有效短暂时间作为数据长度误差导致的影响，我们定出了超时时间，由于我们的解码程序总是在发送完成以后立马进行等待，所以我们将超时启动程序在每次解码程序开始时候，我们在设计的时候采用了两种方法来实现超时检测，通过比较来选择最终合适的方式：

(1) 第一种是我们在开始的时候利用定时器功能启动定时器，根据接收数据到达时间来设置延时时间。延时时间是我们考虑的重点，由于write、lock等命令有效数据来的时间较长，大概几毫秒时间，而Inventory命令组的各条命令等待时间只有几百微妙，我们设置等待时间的时候如果按照Inventory来设置的哈，时间短暂，可以很好的满足其要求，但是按照相同数据用在write命令时候会出现将无效数据作为解码数据来的现象，导致write应答信号检测不到；当我们按照write命令设置的时候，Inventory命令就会出现开始有效被忽略舍弃掉现象，这个比较严重，导致读写器根本读不到标签；同时write命令的参数错误的时候，标签也会回复应答信号，其中包含了错误原因，对于读写器修改下次访问参数有重要提示作用，同时其标签回复数据的时间也和Inventory的时间相符，也会导致错误码接收不到，以上总结这种方法用于只盘存读写器不用于读写的时候可以采用，或者由于write命令是在Inventory命令组之后才进行的命令，所以可以分情况进行解码，这样会使程序显得比较冗余，所以我们

提出了第二种方法。

(2) 基于第一种方法的缺陷，提出我们可以考虑采用一个比较大的数组，将读写器发送命令结束以后所有读到有效无效数据收集起来，然后我们在收集所有数据里边检测选择自己有效数据的头来解码，开始的时候由于在Inventory命令组的时候可以很好的执行，但是到write命令的时候出现一直检测不到有效数据头的问题，通过示波器辅助调试和推断得出是由于数据并未足够大的原因，然后我们将缓冲数组设置足够大很好的满足了write命令，同时新考虑出现了，这样的话缓冲数组设置很大很大，浪费了MCU很多存储空间，对于程序扩展影响很大，所以我们提出了第三种方法。

(3) 我们基于第二种方法基础上我们希望缓冲数组不能够太大，因此提出了循环数据的概念，即当数组满的时候，新数据会覆盖旧的数据。这种方法中循环数组的大小尤其重要，太大与第二种相同，太小则循环数据会讲老的有效数据覆盖掉，我们知道有效数据的最长长度，因此我们将循环数组设置成比最长长度大其1/3的值，可以很好的实现数据的接收。

解码部分的设计利用接收波形中同步码的特殊性来进行有效数据头的检测。因为经过 I、Q 信号选择以后我们选择其中一路进行解码，而通路选择的信号来源于没有经过比较器的，而经过比较器的信号将变成 MCU 可以识别的方波，因此我们采用 MCU 定时器边沿检测功能来记录两个边沿之间计数器值，同时在发送之前我们已经知道接收信号的频率，所以预先已经知道数据 0 和数据 1 的脉宽，通过比较脉宽来确定数据 1 还是数据 0 变成为通过比较计数器（脉宽经过转换的相对值）来得到数据类型。

我们先检测数据头，在反向链路中同步码中通过检测“V”来判断有效数据的出现，但是由于毛刺的影响，我们按照标准换算值得到两个边沿直接的计数值只能作为参考，因为波形治疗并非那么完美，所以我们需要更多附加的脉宽来帮助检测“V”。在标准同步码之前还有连续 12 位数据 0 作为前导码，我们根据 100 个数据零测量得出在连续数据 0 的序列中，我们的标准数据 0 的脉宽为 6500，但是由于毛刺影响，我们通过 100 个数据的统计得出大于 5000 小于 7500 的值都认为是数据 0。在接收的有效数据中比较少见连续 8 个以上的零，结合我们忽略计算的反向链路前导码，当我们检测到大于 8 的计数值在 5000 和 7500 之间时，认为前导码到了，然后我们开始同步码中“V”。从同步码的波形可以看到在 V 后边是一个数据 1，可能会与“V”相同的波形是一个数据 1 后边跟着一个数据 1，后边数据 1 波形发送歪曲，这样前边一个 1 会显得较宽，或者一个数据 1 后边跟着一个数据 0，数据 0 的前半部分脉宽较小导致前边数据 1 较长。不论前一种还是后一种，与实际“V”的情况有差别的就是在“V”出现之后的那个数据 1 即使发生变形，其值也应该比较大，大于数据 1 的 2/3，经过我们 100 次的统计，得出“V”后边的数据 1 发生变形，其值也在 8500 以上，满足了我们的推论，所以在我们检测“V”以后再检测后边一位，如果满足大于 8500 要求，则认为我们检测到标志位“V”开始进行实际有效数据的解码。

数据 0 和数据 1 波形比较好，这样我们容易区分，加入波形不好对于我们解码会产生误差。比如如果是数据 010 或者 011 的时候，由于后一个数据的偏差使我们对中间数据的检测产生影响，我们根据对 1000 个数据值的比较检测，发现假设我们周期脉宽为 1250，我们发现小于 900 的计数值的脉宽我们认为是数据 0，其实际为数据 0 的概率达到 97%，大于 1000 实际为数据 1 的概率为 99%，很好满足我们要求，而在 900 与 1000 直接的数据容易造成检测误差。所以我们考虑如果计数值在 900 和 1000 直接，我们继续检测下一位的脉宽，如果小于 650，认为符合数据 0 的格式，则判断这一位为数据 0，如果下一位大于 650 则认为是数据 1，这里边有一个情况是下一位如果还是数据 1，而且也发送了偏差，小于 650，但是这样的情况基本不会出现，如果出现这样的情况我们认为这类数据足够差，不能够解码应当舍弃再次发送之前命令。

图 5.6 显示了解码部分的执行流程。

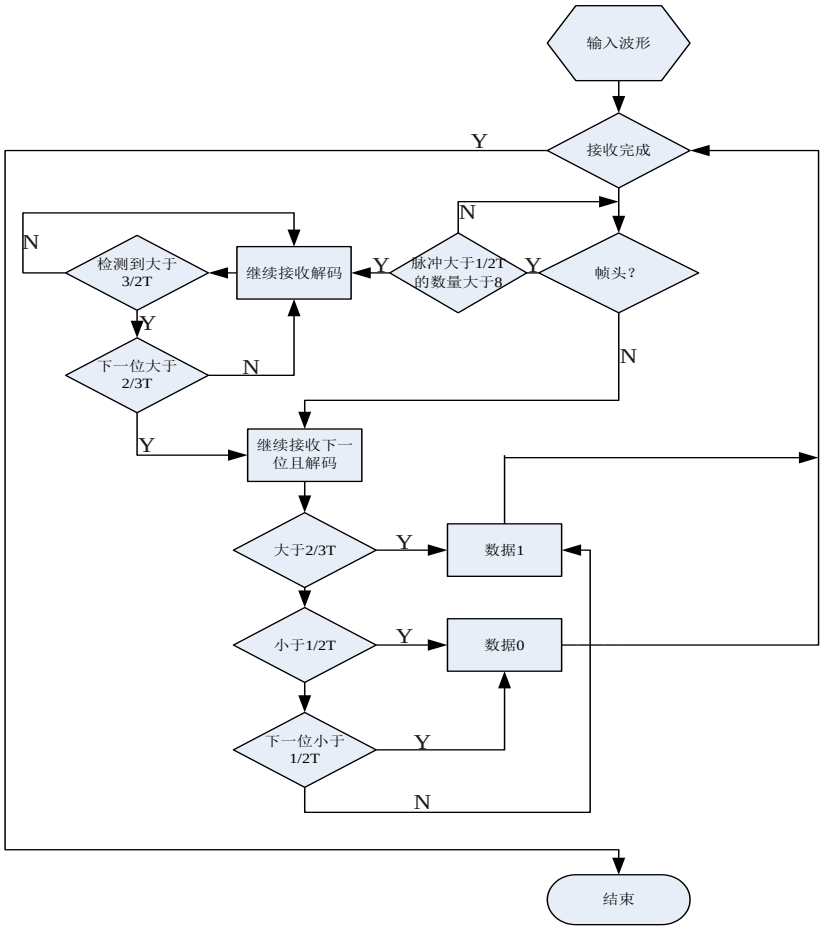

图 5.6 FM0 编码实现流程图

5.4 本章小结

本章主要介绍了软件系统中编码和解码部分的实现，在软件系统实现后经过不断的调试、修改最终完成，然后整机测试的时候能够很稳定的完成解码任务，下一章将介绍读写器的整机测试，它对我们评估读写器具有重要的意义。

第六章 UHF RFID 读写器程序的测试验证

一款读写器的成功实现需要进过不断的调试与测试才能最终完成，在实现读写器的过程中，如果不经过测试，问题没有反映出来，将会引起很大的问题很可能使整个项目完全废弃。在调试过程中，主要是进行了解码部分、读写器协议流程、整机应用测试等。根据读写器的要求对读写器进行一般性的测试，比如读写距离、是否能进行多标签读写等，这些数据对我们实现读写器具有至关重要的作用。

6.1 测试平台搭建

测试平台包括一台 PC 机、一个超高频天线、读写器、串口线、超高频标签、9V 适配器。串口线用于连接读写器与 PC 机。超高频标签我们使用了包括 Impinj 和 Alien 两个国际上最大两家生产的标签[2]。

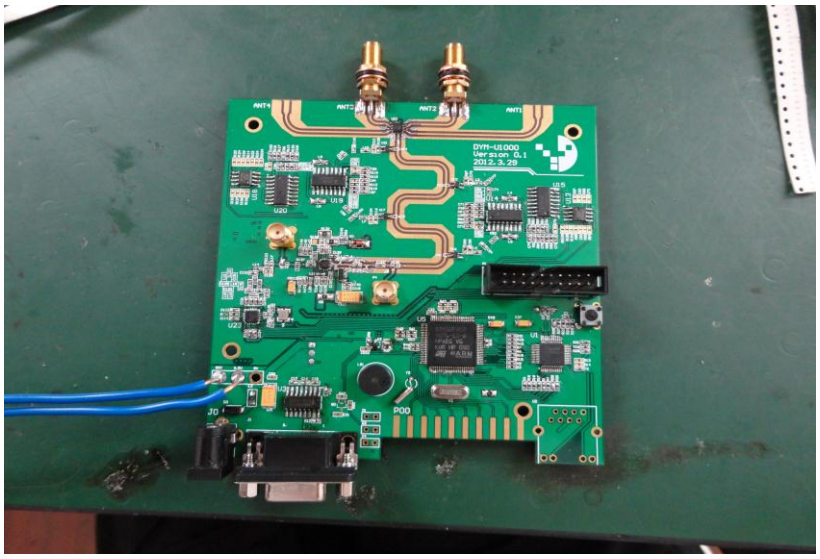

图 6.1 读写器实物图

6.1.1 测试环境及目标

本次测试需要 PC 机上位机软件配合使用，读写器采用此次设计的低成本读写器，其成品图如图 6.1 所示，上位机软件使用奥地利微电子提供的 DEMO 中的上位机软件，然后使用串口与 MCU 进行连接。串口帧格式采用奥地利微电子 DEMO 中 MCU 与上位机串口通信使用的帧格式，这样我们可以很好的匹配上位机软件，减少我们开发调试中的检查点。天线使用商用超高频读写器天线，在本次测试中我们选用两路天线，分别测试不同的天线端口。串口线使用标准九针口串口线，标签使用 4 张商用超高频标签，两张 impinj 和两张 Alien 的标签，因为市场上 90%的份额都被这两家公司占有，这样为我们的读写器兼容性做了很好的测试。供电系统使用 9V 适配器与 220V 连接为读写器提供电源。其平台搭建如图 6.2 所示，其

中两个天线与低成本读写器的四个天线接口中的两个连接，然后读写器 RS232 接口与 PC 主机的串口连接，上位机使用 AS399Xdemo。

硬件测试环境需要将读写器放大器设置最大，发射功率设置最大，载波频率为 920MHz，配合商用超高频天线。

图 6.2 读写器测试搭建平台

6.1.2 测试手段及数据

我们测试采用黑盒测试，因为软件系统的测试需要整体来进行，所以通过外部情况来反应读写器系统内部运行情况。通过读写器可以同时读到多张标签，来证明读写器协议处理和编解码等各个模块能够正常运行，通过上位机对读写器进行控制，通过观察读写器运行状况来证明读写器模块的情况[7]。

在程序调试过程中，需要示波器辅助调试，我们将示波器的两个探头接入基带解码部分，

6.1.3 测试步骤

(1) 首先将读写器软件主函数修改成读写器一个标签，通过蜂鸣器来标识是否读到卡；

(2) 然后连接读写器到上位机软件，通过上位机命令来控制读写器读多张卡；

(3) 在步骤一的基础上不断调整标签与读写器天线的距离，看读写器的性能；

(4) 然后修改读写器的各个参数，通过示波器观察波形进行验证；

(5) 对于防冲突的测试，我们通过两步来完成，通过示波器观察在同一周期内收到两次标签回复来证明读写器可以一次读写两张标签；通过上位机软件读出多张不同ID的标签，来证明读写器可以读到多张标签；

(6) 因为读写器读写标签的时候，可以修改各类不同的参数，包括内存地址、读写长度等，我们通过修改上位机中参数，然后观察读出数据是否正确，来证明修改参数是否成功，同时证明我们读写器下层软件协议实现是否成功。

6.2 功能性测试

由上述介绍我们知道，功能性测试也称为黑盒子测试，我们并不关心整个软件各部分内部结构和代码的设计实现，只需要针对各个功能，进行测试即可[4-6]。

6.2.1 与 PC 机通信模块测试

这部分我们主要依靠串口调试助手来观察修改串口配置是否成功，依靠返回值来测试读写器串口部分基本配置是否成功。帧格式测试依靠奥地利微电子提供的上位机软件，通过其后续的配置是否成功，即可证明帧格式以及串口传输是否成功，其截图如图 6.3 所示。由图中可以看到，我们首先修改左边框的串口基本配置数据，然后观察有边框返回的数据，测试数据证明串口配置数据通过测试。

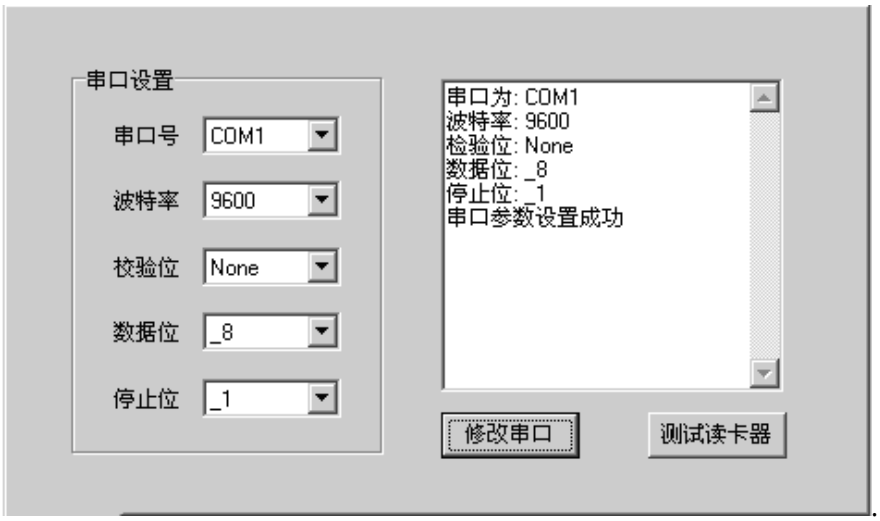

图 6.3 串口配置检测界面

6.2.2 基本配置模块测试

上位机配置界面如图 6.4 所示，包括反向链路速率、Q 初值、编码方式、读写器与标签通信中会话值等。各种参数的测试方法各不相同。

Q 值的修改关系到标签一次可以读写多少张标签，当我们放四张标签的时候，Q 值设置成 0，由上位机可以看到标签只可以读到一张标签，当 Q 值设置成 2 的时候，读写器可以顺利的读到四张标签，其测试图如 6.6 所示。

反向链路速率的测量需要借助示波器的帮助，将示波器的探头接入读写器后向通路，然后利用示波器测试反向散射波形的脉宽，来实现对此参数的测量，由图 6.5 可以看到，当我们修改为 80KHz 的时候，脉宽正好 12.5μs，因为单个波形测试增加示波器的误差，所以我们测量了多个数据的脉宽来测量，从图中可以看出测试通过。

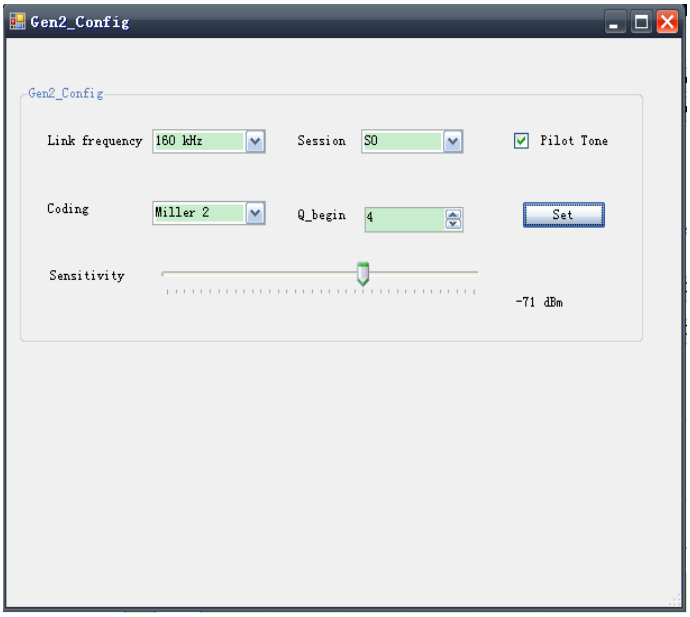

图 6.4 Gen-2 协议通信基本配置

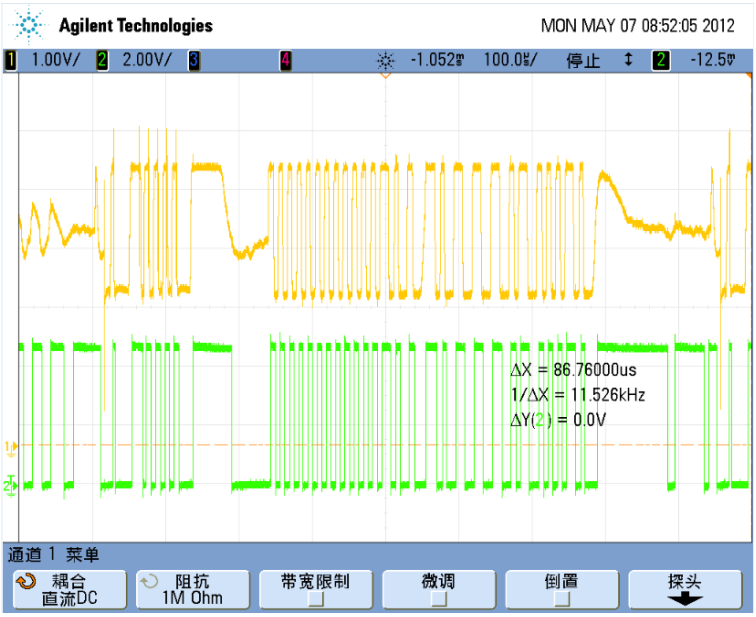

图 6.5 反向脉宽测试图

6.2.3 协议命令执行模块测试

(1) 通过Antenna on来完成天线开关，然后利用示波器来测试反向链路是否有回复波形，测试通过；

(2) 点击图6.6中List Tag ID来测试盘存命令是否成功，可以看到当Q值设置成2的时候，标签可以同时读到4个标签，标签ID各不相同；

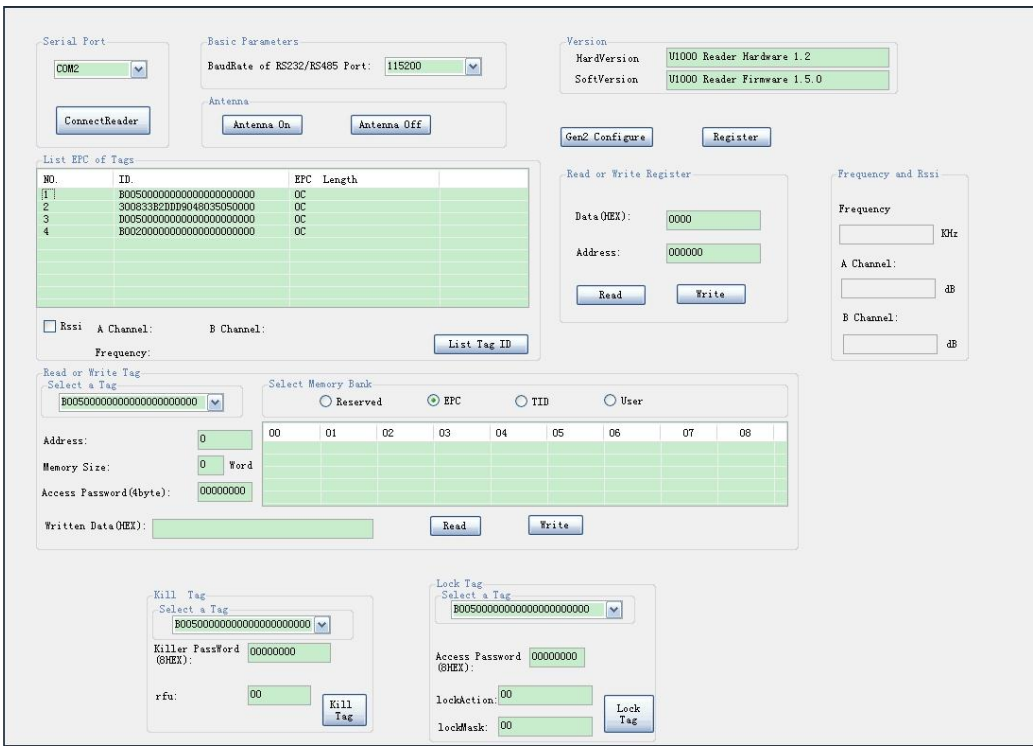

图 6.6 读写器配置界面

(3) 写标签过程，填写内存地址和数据，然后读同一地址的数据来证明写命令的成功。如图6.6所示；

(4) 读命令可以读写EPC部分的值，来证明读命令代码部分的成功，测试通过；

(5) Kill和lock命令，通过示波器观察标签回复波形来测试读写器这两条命令部分；

(6) 反向链路中，正交的两路信号波形差别较大，我们通过单步调制，断点设置到波形选择之后，来观察波形选择是否成功。然后看读写器能够成功的读到标签，证明解码部分实现成功，如图6.7所示。

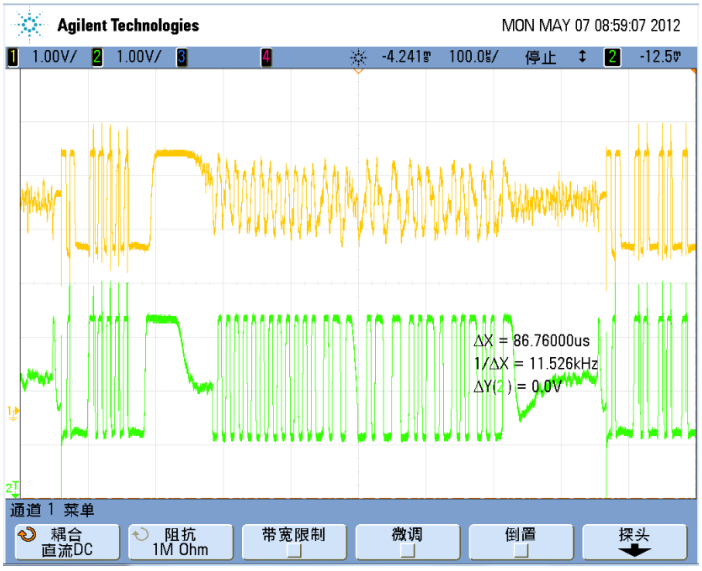

图 6.7 正交双通道波形

6.2.4 防冲突碰撞测试

将四张标签放到天线上面，然后调整 Q 值，通过不同的方式观察读到标签的数量来实现标签防冲突测试。

其中硬件环境为将标签置于读写器天线 1 米范围内，同时读写器发射功率设置最大，载波频率为 920MHz，读写四个不同厂家的商用标签进行测试。

(1) 将Q值设置成3，这样读写器读到标签的最大数量为8，通过点击图6.4中Q值，完成通信参数配置，然后点击图6.6的List Tag ID，可以看到读写器读到四张标签，黑匣子测试证明读写器同时能够读到多张标签，达到预期效果。

(2) 修改图6.4中Q值，调整为2，然后依靠示波器的辅助，来实现标签防冲突算法测试。从图6.9可以看出，其中两个比较长的幅度标签回复的信号的波形，幅度比较大的脉冲为读写器发送命令的波形，由于Gen-2规定编码方式的特殊性，因此可以用肉眼观察发送接收信号的波形，通过观察对照各条命令波形格式，可以看到中间两段脉宽比较长的脉冲即为回复的标签ID信息，可以直观的证明读写器程序的防冲突算法实现成功。

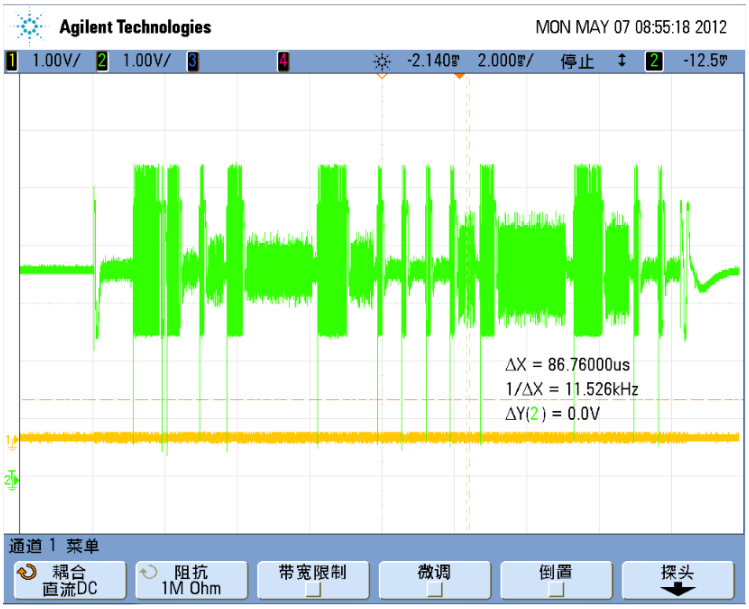

图 6.9 读写器读写多张标签的通信波形图

6.3 稳定性测试

通过将读写器与韦根门控系统连接，然后给员工配发上下班门禁卡，经过多天的测试，基本能够满足市场要求，达到了预期目标。

6.4 本章小结

本章主要对标签通信模块和读写器配置模块中部分测试，重点展示了协议处理、通信配置、防冲突算法等部分。测试了多标签读写和单标签读写，通过观察分析测试数据，证明软件系统的可靠。

第七章 总结与展望

7.1 总结

射频识别（Radio Frequency Identification，简称 RFID）技术当前非接触射频识别技术的前沿，在工业界和学术界得到了迅速的发展。供应链管理、电子支付、RFID 护照、环境监控、办公室访问控制、智能标签、目标探测和跟踪、港口管理、食品工业监控以及动物鉴定等方面都将应用到这一技术；同时 RFID 也是实现“物联网”的基础。其中，UHF RFID 又是未来 RFID 发展的驱动力。但是由于 RFID 系统中最重要的一环读写器，成本一直保持很高的价位，对于市场应用是一个极大的阻碍，因此降低读写器的价格成为了解决 UHF RFID 发展的核心问题。本文提出的低成本读写器设计是一款性价比很好的读写器，由于读写器硬件设计的特殊性，因此软件系统的成功对于这款读写器的完成起着很关键的部分。

此款读写器的提出是在做了充足的市场调研和考察之后提出的，是一款基于产品级的读写器。由于主要面向国内市场，所以其工作频率主要支持 902~928MHz。符合 EPC Class1 Gen-2 标准，射频部分采用收发芯片，并且使用非常先进的微带线技术作为发送接收隔离器件，实现了在硬件成本上的大跨越。

(1) 软件部分基于硬件电路的设计，完成了Gen-2协议处理流程。

(2) 协议处理中采用了动态ALOHA算法，对于解决多标签冲突问题起到了关键的作用。

(3) 编解码模块利用有限MCU资源实现了高效的代码，不仅能够很好的完成协议规定的Gen-2编解码，而且为协议处理节省了充足的时间。

(4) 与上位机接口部分采用了自己定义的帧格式传输协议，保证了数据的稳定传输。

(5) 与MCU连接器件的驱动力求模块化，对于后期开发升级具有重要意义。

经过不断了调试和测试，最终实现了读标签距离 8 米（天线增益 5dBi）写标签 3 米的性能，最高单标签读取速度每秒 200 次和抗冲突的性能，成本约为市场读写器的三分之一，达到了设计指标，有较高的实用价值。

7.2 展望

由于我国 UHF RFID 才刚刚起步，有许多关键技术还有待提高，在本文的设计实现中也遇到了许多问题，在未来的工作中待续提高：

(1) 首先需要将与上位机接口中网口部分的程序完成，还有POS端口的完成。

(2) 解码部分需要更加高效全面，在工作中有同事提出采用匹配滤波器，在MATLAB实现过程中具有很好的解码效果。

(3) 由于我们读写器主要应用于安装于固定位置的模式，对于手持机还软件系统需要进一步改版，以适应市场应用中的各种场合。

致　　谢

本文是在李文钧教授和王彬教授的共同指导，各位同学和同事的帮助下完成的，在论文完成之际，我要向他们表示我诚挚的谢意。

首先，我要感谢我的导师李文钧老师，在我读研究生的两年多时间里，导师给了我悉心的指导和关怀，给了我充分的信任。在他的关心和指导下，我的科研能力和分析动手能力得到了极大的提高，通过和李老师共事的过程中，我还从他身上学习到了很多优秀的品格、为人处事的道理、对事业的努力追求、对家庭的责任、对项目管理的经验、对时间的控制能力等等，这些，都将成为我以后学习和生活的宝贵的财富。在此，再次向李老师表达我最诚挚的谢意和深深的敬意。

我要感谢扬州稻源微电子有限公司的王彬总经理、杨作兴副总经理和张宏副总经理。王彬总经理作为我的校外导师，在我实习的一年里，为我的学习工作提供了良好的平台，能够严格要求我，并在我工作中给予足够的帮助，在我一年的实习中给了我最好的资源，也在毕业设计中给了我很大的帮助。我要感谢项目组的同事江汉、卞荣辉和江成，在他们与我的共同努力下快速完成了读写器的开发。同时也要感谢同事高阳和袁传奇在我生活学习中不断鼓励和给我提供必要的硬件知识，同时我要感谢扬州稻源微电子有限公司的所有员工，和他们在半年多的时间里一起工作学到了许多宝贵的知识，这个温暖的集体给我留下了许多珍贵的记忆。

最后，衷心感谢我的父母在生活等各方面对我的极大关怀，他们是我坚实的后盾，正是由于他们无私的爱护关心才使我能够安心学习，在学业上给予我极大的支持。他们的恩情永铭我心。

参 考 文 献

[1] 黄玉兰. 物联网射频识别（RFID）核心技术详解[M]. 北京: 人民邮电出版社,2010:3-4.

[2] 周晓光,王晓华. 射频识别（RFID）技术原理与应用实例[M]. 北京: 人民邮电出版社,2006.

[3] Yan Zhang, Laurence T.Yang,Jiming Chen. RFID 与传感器网络：架构、协议、安全与集成[M]. 浙江: 机械工业出版社,2012:1-2.

[4] Carl J. Weisman. Essential Guide to RF and Wireless[M]. 2nd Edition. NJ,US:Prentice Hall,2002:35-48

[5] 张智文. 射频识别技术理论与实践[M]. 北京: 中国科学技术出版社, 2008:18-33.

[6] 刘禹,曾隽芳. RFID：让物 S 与物交流[N]. 计算机世界报, 2007(16): 19-20.

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

[8] 伊应增. 微波射频识别技术[D]. 西安: 西安电子科技大学, 2002.

[9] 谭民,刘禹,曾隽芳. RFID 技术系统工程及应用指南[M]. 北京: 机械工业出版社, 2007.4

[10] JU-YEN HUNG. Improving reader performance of an UHF RFID system sing frequency hopping techniques[D]. Taipei: National Open University, 2006.

[11] Mark Brown,Sam Patadia,Sanjiv Dua. Mike Meyers Comptia RFID+Certification Passport[D]. USA：McGraw-Hill,2007:36-39.

[12] D M．Dobkin. The RF in RFID: Passive UHF RFID in Practice[M]．USA：Newnes, 2007: 33-189.

[14] Jingchao Wang,Baoyong Chi,Xuguang Sun. System Design Considerations of Highly Integrated UHF RFID Reader Transceiver RF Front—End[C]. 9th International Conference on Solid-State and Integrated-Circuit Technology, 2008: Page(s)：1560-1563.

[15] Rob Glidden,Cameron Boekoriek,et a1.Design of Ultra-Low-Cost UHF RFID for Supply Chain.Applieations[J]．IEEE Cpmmu Mag,August 2004.

[16] 李进东,范琴秀. 射频识别技术的发展与应用[J]. 科技信息(学术版), 2006,01(01): 46-47.

[17] 中华人民共和国科学技术部等十五部委. 中国射频识别(RFD)技术政策白皮书, 2006.06.09.

[18] 康东,石喜勤,李勇鹏. 射频识别(RFID)核心技术与典型应用开发案例. 第一版[M]. 北京: 人民邮电出版社, 2008: 39-56.

[19] Klaus Finkenzeller. 射频识别技术（第三版）[M]. 北京: 电子工业出版社, 2006.

[20] 刘修伦. 基于 STM32 微控制器的数据通信接口设计[D]. 青岛: 中国石油大学, 2006.22-29.

[21] 温耀军. 基于 STM32F107VC+CS495313 的数字调音台的设计与开发[D]. 湖南: 湖南大学, 2006:4.

[22] A.Goldsmith. Wireless Communication[M]. 北京: 人民邮电出版社, 2007.

[23] Daniel W. Engels Sanjay E. Sharma, Stephen A. Weis. R_d systems. Security and privacy implications[N]. Technical report, MIT Auto-ID Center, February 2002.

[24] Junfang Zeng, Yu Liu, Chong Liu. Research on Test based RFID Deployment Simulator[C]. Third 2008 International Conference on Convergence and Hybrid Information Technology, 2008:Volume 1,Page(s)：1 142-1146 11-13.

[26] 胡乃英. UHF 频段 RFID 空中接口协议的研究浅谈[D]. 西安: 西北大学硕士论文, 2008.

[27] ISO/IEC 18000-6: Information technology automatic identification and data capture techniques-Radio frequency identification for item management air interface. Part6: Parameters for air interface communications at 860-960MHz.

[28] EPC™ Radio Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860MHz-960MHz Version1.1.0.

[29] 刘锡杰, UHF 频段 RFID 读写器的设计[D]. 大连: 大连理工大学,2007.

[30] 田丹. 基于嵌入式 Linux 的 UHF 智能 RFID 读写器的研究与实现[D]. 成都: 电子科技大学, 2006.

[31] 吴春华,陈军. 动态 ALOHA 法在解决 RFID 反碰撞问题中的应用[J]. 电子器件, 2003,26(2):173-176.

[32] J R Cha, J H Kim. Dynamic framed slotted aloha algorithms using fast tag estimation method for RFID systems [D]. IEEE Communications Society, 2006, (2):768-772.

[34] Kin Seong Leong, Mun Leng Ng,Cole P H.The reader collision problem in RFID systems. Microwave [J], Antenna, Progagation and EMC Technologies for Wireless Communications, 2005, 1(1): 658-661.

[35] Maozu Guo, Yang Liu, Malec J. A new Q-learning algorithm based on the metropolis criterion[C]. Systems, Man, and Cybemetics, Part B, IEEE Transactions on, 2004, 34(4): 2140-2143.

[36] 周祥. RFID 技术在物联网中应用的关键技术探讨[D]. 镇江: 江苏大学, 2005: 60-65.

[37] 王伟. 射频识别(RFID)技术及其应用的研究[N]. 安徽师范大学学报, 2008,31(2):139-141, 149.

[38] 姜丽芬,卢桂章,辛运帏. 射频识别系统中的防碰撞算法研究[J]. 计算机工程与应用, 2007, 43(15):29-32.

[39] 刘佳,张有光. 基于时隙的 RFID 防碰撞算法分析[J]. 电子技术应用, 2007, 33(5): 94-96, 100.

[40] 权冀川,周满珍. ALOHA 系统信息到达流的概率分析[N]. 解放军理工大学学报, 2001,2(3):67-70.

[41] 王玉宝,史亮,徐宝文. 基于 Q 学习的复杂程序动态依赖性分析[D]. 计算机与数字工程, 2005,33(2):9-11.

[42] Joshi G P, Abdulla Mamun K M, Sung Won Kim. A Reader Anti-collision Protocol for Dense Reader RFID System[C]. In: Communications and Mobile Computing, 2009.CMC'09. WRI International Conference on.Yunnan, 2009:313-316.

[43] 胡益, 苏娟. 一种 ISO18000-6B 的回波解码方法[J]. 微计算机信息. 2008:24, 9-2, 163-164.

[44] International organization for standardization. ISO/IEC18000-6 Information Technology AIDC Techniques RFID of Iterm Management[S]. Switzerland:ISO/IEC 2004.

[45] 韩益峰. 射频识别读写器的研究与设计[D]. 上海: 复旦大学, 2005.

[46] 刘君军,刘陈. 基于单片机的 UHF RFID 读写器基带编解码模块的设计[J]. 计算机工程应用技术,

2010, 6(36).

[47] 乌云高娃,鲁俊,高荣华. RFID 读写器接收机基带数字信号处理研究[J]. 电子学报, 2009,6.

[48] 李萌. UHF RFID 读写器基带关键模块的设计研究[D]. 上海: 华东师范大学,2008:12-36.

[49] 张远海,翁佩纯. UHF RFID 读写器基带信号 FM0 解码研究[J]. 电子设计工程,2011,6(19).

[50] 梁飞,张红雨,陈友平. UHF 读写器设计中的 FM0 解码技术[J]. 电子设计工程,2010,18(11).

[51] 段璞. UHF 频段 RFID 读写器基带分析研究[D]. 西安: 西北大学,2010:23-25.

[52] 阳璞琼. 超高频 RFID 编解码系统研究与设计[J]. 南华大学学报(自然科学版),2010,24(3).

[53] 肖菊兰,张红雨. 超高频 RFID 读写器设计[J]. 电子工程设计,2010,18(11).

附录

作者在读期间发表的学术论文

发表论文：

[1] 王成,李文钧,王彬. 基于STM32的UHF RFID读写器基带编解码的实现[J]. 硅谷,2012,19 :65.