

# 西安邮电大学

## 毕业设计（论文）

题目： 基于嵌入式的门禁系统

学院： 自动化

专业： 自动化

班级： 自动 1403

学生姓名： 李智文

学号： 06141106

导师姓名： 马翔 职称： 助理工程师

起止时间： 2017 年 12 月 5 日至 2018 年 6 月 10 日

## 毕业设计（论文）声明书

本人所提交的毕业论文《基于嵌入式的门禁系统》是本人在指导教师指导下独立研究、写作的成果，论文中所引用他人的文献、数据、图件、资料均已明确标注；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注明并表示感谢。

本人完全理解《西安邮电大学本科毕业设计（论文）管理办法》的各项规定并自愿遵守。

本人深知本声明书的法律责任，违规后果由本人承担。

论文作者签名：

日期：      年    月    日

## 西安邮电大学本科毕业设计(论文)选题审批表

申报人	张永健/马翔	职称	高级工程师/助理工程师			学院	自动化	
题目名称	基于嵌入式的门禁系统							
题目来源	科研				教学		其它	是
题目类型	硬件设计	是	软件设计		论文		艺术作品	
题目性质	应用研究		是		理论研究			
题目简述	<p>随着智能建筑的发展及智能化小区的出现,人们已接受了智能建筑及智能化小区所带来的高效、舒适的工作及生活环境,同时对智能化系统的可靠、便利也提出了更高的要求。作为智能建筑的重要组成部分之一,门禁系统的功能早已超越了单纯的门道及钥匙管理,它已经逐渐发展成为一套完整的出入管理系统,它在工作环境安全 and 人事考勤管理等管理工作中发挥着巨大的作用。</p>							
对学生知识与能力要求	<p>掌握微机原理与接口技术及单片机原理及应用,理解嵌入式系统的基本组织结构与工作原理,具有一定的模拟电路技术基础及网络通信原理,熟悉常用传感器原理及使用方法,具有嵌入式软件设计能力及解决实际问题的动手能力。</p>							
具体任务以及预期目标	<p>设计基于嵌入式的门禁系统,用户可通过密码,门禁卡等方式进行门禁电磁锁的开关,并能够记录用户进出记录。</p>							
时间进度	<p>2017 年 12 月 05 日—2017 年 12 月 10 日 选取毕设题目  2017 年 12 月 11 日—2018 年 01 月 06 日 查阅资料,撰写提交开题报告  2018 年 01 月 07 日—2018 年 03 月 04 日 确定系统架构设计方案  2018 年 03 月 05 日—2018 年 03 月 31 日 系统软件程序设计  2018 年 04 月 01 日—2018 年 04 月 15 日 软硬件的联合调试  2018 年 04 月 16 日—2018 年 04 月 31 日 系统功能完善  2018 年 05 月 01 日—2018 年 05 月 25 日 撰写毕业设计论文  2018 年 05 月 26 日—2018 年 06 月 01 日 修改、装订论文  2018 年 06 月 02 日—2018 年 06 月 10 日 准备毕业答辩</p>							

系（教研室）主任 签字	年 月 日	主管院长 签字	年 月 日
----------------	-------	------------	-------

## 西安邮电大学本科毕业设计（论文）开题报告

学生姓名	李智文	学号	06141106	专业班级	自动 1403
指导教师	张永健/马翔	题目	基于嵌入式的门禁系统		
<p>选题目的（为什么选该课题）</p> <p>随着智能建筑的发展以及智能化小区的出现，人们已经接受了智能建筑及智能化小区所带来的高效、舒适的工作及生活环境。同时对智能化系统的可靠、便利也提出了更高的要求。门禁系统就是为了满足人们对现代办公和生活场所的更高层次安全管理的需要应运而生的。作为智能建筑的重要组成部分之一，门禁系统的功能早已超越了单纯的门道及钥匙管理，它已经逐渐发展成为一套完整的出入管理系统。早期的门禁系统都是基于单片机理念设计的，虽然可实现基本的门禁控制功能，但系统结构和功能相对单一。而嵌入式门禁系统集成嵌入式系统实时、稳定、可靠，具有网络支持的特点和感应卡识别方式安全、方便、快捷的优点，可以实现对门禁管制区域的安全、高效、可靠的智能化管理。使得它能在工作环境安全和人事考勤管理等管理工作中发挥巨大作用。</p>					
<p>前期基础（已学课程、掌握的工具，资料积累、软硬件条件等）</p> <p>已学课程：单片机原理及应用、微机原理与接口技术、模拟电子技术基础、传感器原理及应用、高级语言程序设计（C 语言）</p> <p>掌握的工具：keil</p> <p>软硬件条件：相关的编程软件、单片机、磁卡和读卡器、显示屏、蜂鸣器、继电器等。</p>					
<p>要研究和解决的问题（做什么）</p> <ol style="list-style-type: none"> <li>1、读卡器要能正确识别磁卡信号，进行判断是放行还是报警。</li> <li>2、密码能通过键盘进行输入，并且单片机能够判断密码的正确与否。</li> <li>3、为了增强保密性，要能够直接在产品中修改门禁的密码。</li> </ol>					
<p>工作思路和方案（怎么做）</p> <p>4 月上旬，查阅并了解读卡器和磁卡的相关资料，了解门禁基本工作原理，选择电磁锁及可编程控制器作为主要元器件，实现设计方案。</p> <p>4 月中下旬，对“如何记录用户的进入记录”这个问题进行深入的了解和学习，并且设计出硬件线路和计算机服务器编码。</p> <p>5 月初，先做出一个基本满足设计要求的样品。</p> <p>5 月底之前，对样品的不足之处进行修改和优化，直到完全满足设计要求为止。</p> <p>5 月底，完成满足设计要求的作品，同时撰写毕业设计论文。</p> <p>6 月初，对毕业论文进行查重、修改，完成毕业设计答辩。</p>					

指导教师意见

签字：

年 月 日

## 西安邮电大学毕业设计（论文）成绩评定表

学生姓名	李智文	性别	男	学号	06141106	专业 班级	自动 1403
课题名称	基于嵌入式的门禁系统						
指导教师 意见	<p style="text-align: center;">（从开题论证、论文内容、撰写规范性、学习态度、创新等方面进行考核）</p> <p style="text-align: right;">评分（百分制）：      指导教师(签字)： _____ 年    月    日</p>						
评阅 教师 意见	<p style="text-align: center;">（从选题、开题论证、论文内容、撰写规范性、创新和预期成果等方面进行考核）</p> <p style="text-align: right;">评分（百分制）：      评阅教师(签字)： _____ 年    月    日</p>						
验收 小组 意见	<p style="text-align: center;">（从毕业设计质量、准备、操作情况等方面进行考核）</p> <p style="text-align: right;">评分（百分制）：      验收教师(签字)： _____ 年    月    日</p>						
答辩 小组 意见	<p style="text-align: center;">（从准备、陈述、回答、仪表等方面进行考核）</p> <p style="text-align: right;">评分（百分制）：      答辩小组组长(签字)： _____ 年    月    日</p>						
评分比例	指导教师评分（20%） 评阅教师评分（30%） 验收小组评分（30%） 答辩小组评分（20%）						
学生总评 成绩	百分制成绩				等级制成绩		
答辩委员 会意见	<p style="text-align: center;">毕业论文(设计)最终成绩(等级)：</p> <p style="text-align: right;">学院答辩委员会主任(签字、学院盖章)： _____ 年    月    日</p>						

# 目 录

摘 要 .....	I
ABSTRACT .....	II
第一章 引 言 .....	1
1.1 研究背景 .....	1
第二章 系统总体方案设计 .....	2
2.1 系统设计原理 .....	2
2.2 硬件方案的选择 .....	2
2.2.1 方案选择 .....	2
2.2.2 主控芯片的选择 .....	3
2.2.3 显示模块的选择 .....	4
2.2.4 键盘输入电路的选择 .....	4
第三章 硬件电路设计 .....	6
3.1 电路原理 .....	6
3.1.1 门禁系统主要组成 .....	6
3.1.2 门禁系统工作原理 .....	6
3.2 资源分配 .....	8
3.3 最小系统设计 .....	9
3.3.1 STC89C52 的简述 .....	9
3.3.2 最小系统电路设计 .....	9
3.4 门禁控制电路 .....	10
3.4.1 继电器介绍 .....	10
3.4.2 继电器驱动电路设计 .....	10
3.5 蜂鸣器电路 .....	11
3.6 控制指示灯 .....	11
3.7 4×4 矩阵键盘 .....	11
3.8 读卡模块 .....	12
第四章 软件电路设计 .....	14



4.1 系统主程序分析 .....	14
4.2 子程序分析 .....	15
4.2.1 键盘程序 .....	15
4.2.2 操作模块 .....	19
结束语 .....	30
致  谢 .....	31
参考文献 .....	32
附录 A.....	1

## 摘 要

本次设计中的门禁系统采用的是 STC89C52 单片机作为主控芯片，外围有 4×4 矩阵键盘和 RFID 射频 IC 卡感应模块作为输入模块，有蜂鸣器控制电路、开门指示灯电路、继电器控制电路以及 LCD 液晶显示电路等作为输出模块。

该门禁系统可以通过 RFID 刷卡和密码输入进门，其工作原理为：当刷卡时，若卡获得进入权限，则 LCD 液晶显示模块显示刷卡成功，并显示卡的 ID 号和持卡人姓名，同时继电器电位变化打开门禁，指示灯点亮，延时一段时间后继电器电位再次变化将门锁关闭，同时指示灯熄灭。若卡未获得权限，则显示失败，失败 3 次后，会启动蜂鸣器报警。若选择通过输入密码打开门禁，当输入密码正确的时候，显示模块显示密码正确，继电器工作将门锁打开，指示灯点亮，延时一段时间后，继电器再次工作将门锁关闭，指示灯熄灭。若密码输入不正确，显示模块显示密码不正确，三次之后，启动蜂鸣器报警。

**关键词：**RFID，射频 IC 卡，门禁，STC89C52，嵌入式

## ABSTRACT

The design of the entrance guard system USES is STC89C52 single-chip microcomputer as the master control chip, periphery have 4 \* 4 matrix keyboard and RFID radio frequency IC card induction module as input module, a buzzer control circuit, the door light circuit, relay control circuit and LCD liquid crystal display circuit and so on as the output module.

The entrance guard system can through the RFID card and password input into the door, its working principle is: when a credit card, if the card to get access to, the LCD liquid crystal display module display card is successful, and display card ID number and the name of the cardholder, relay potential changes to open the door at the same time, the light is lit, delay after a period of time relay potential change will lock closed again, at the same time light is put out. If the card does not get permission, the display fails, and after three failures, the buzzer alarm will be activated. If choice by entering a password to open the door, when the input password correctly display module display the password correctly, relay work will open the door, light is lit, the delay after a period of time, the relay work will lock closed again, light is put out. If the password is incorrect, the display module shows the incorrect password, and after three times, the buzzer alarm is started.

**Key words:** RFID radio frequency IC card, access control, STC89C52, embedded.

## 第一章 引言

### 1.1 研究背景

在现在这个社会，人们越来越追求高智能的东西，因为智能产品能为我们提供很多的便利，比起以往，现在社会上也到处都充斥着智能产品，这是人们的追求，也是时代的发展。尤其一些智能化的建筑及智能化小区所带来的高效、安静、舒服的环境，也是越来越受到人民的喜爱。而不论是在智能化的建筑还是智能化的小区，有一个必要的产品就是智能门禁系统<sup>[1]</sup>。它便是为了满足人们对当代办公和生活场所的更高级别的安全治理应运而生的。并且它早就逐步成长为一套完全的进出管理体系。眼下，门禁控制体系已作为安防系统中最为重要的构成部分之一。究其原因，一是因为这是一个智能化的产品。二便是因为这是一款对于安全进行管理的产品，人们对自己的人身安全有了不一样的追求了。三便是因为门禁体系转变了以往的安防类产品。而可以适用于门禁系统的 IC 卡也越来越普及，而且比起以前一把钥匙只能打开一把锁，在一栋办公楼或者住宅楼里，可能一个人身上要携带很多的钥匙才能满足日常的要求的局面。现在的 IC 卡在公司可能一个人的卡，可以打开写字楼出入口的门禁，打开自己办公室的门禁，打开会议室的门禁等多个门锁。可以做到一卡打开多个门禁的情况，方便的人们的携带，不再需要每天携带大量的钥匙，效率既低又容易出错。而且以前的钥匙如果丢失，尤其一些重要地方的门锁的钥匙，可能丢失之后立马就需要换锁。首先可能因为发现不及时而丢失一些重要东西，其次，握有这个门锁钥匙的人都需要将原钥匙丢弃。这无疑是一种对资源的浪费。而选择了 IC 卡，若卡丢失，只需要将这一张卡从系统里面去除就可以。完全不会影响到其他人的使用。既方便也是对资源的一种高效利用，所以本次设计会在传统的密码门禁上添加 IC 卡门禁控制功能。

## 第二章 系统总体方案设计

### 2.1 系统设计原理

本次设计是基于 STC89C52 单片机设计在传统密码门禁的基础上添加 IC 卡门禁系统的设计。

硬件电路此次一共设计了 6 个模块，除了必要的 52 单片机最小系统之外，此次设计还拓展了一些关于此次设计需要的一些拓展模块，如：射频 IC 卡模块、液晶显示模块、蜂鸣器报警提醒模块、4 乘 4 矩阵键盘模块、继电器控制电磁锁模块<sup>[2]</sup>。

设计的硬件电路布局如图 2.1 所示。

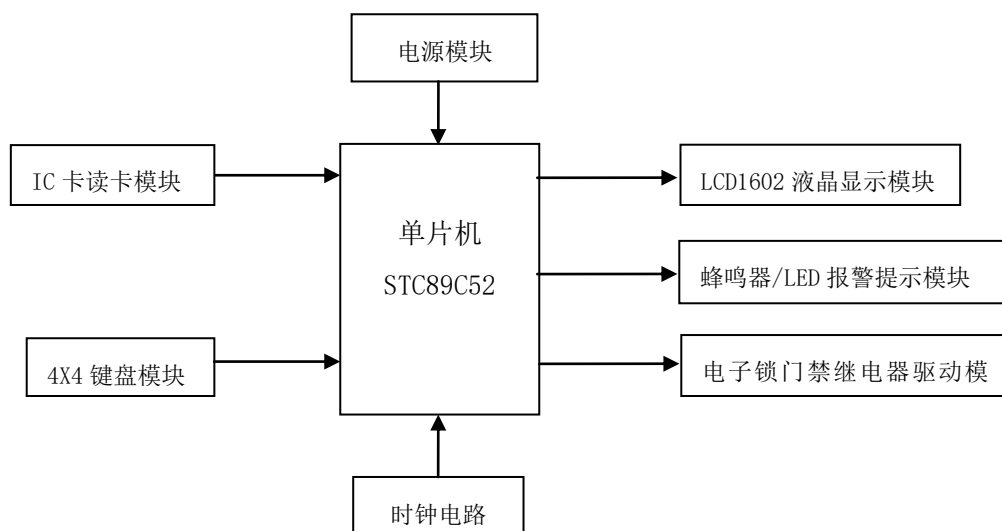


图 2.1 系统总体框图

### 2.2 硬件方案的选择

#### 2.2.1 方案选择

门禁系统有许多种方案：大体上主要可以将其分为两种，联网型和不联网型。

联网型：结构图如图 2.2 示：

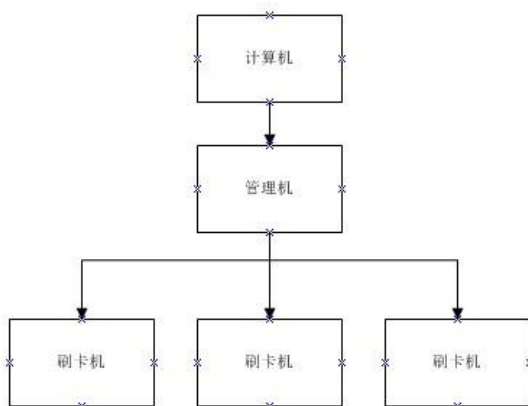


图 2.2 联网型结构图

表 2.1 联网型和不联网型优缺点分析表

	联网型：	不联网型：
优点：	1、实时联网； 2、刷卡后立即把数据传输到主机上； 3、便于查询并能生成报表；	1、单台刷卡机就能工作； 2、不用布线； 3、操作简单方便。
缺点：	1、计算机要求 24 小时开机； 2、管理机如果出现故障，受影响面积比起不联网型会相当大。	不便于查询。

联网型和不联网型的优缺点如表 2.1 所示。

此次设计选择不联网型。究其原因有二，其一由于选用该型，设计相比来说比较简单，而且其中的问题也比较容易发现。联网型需要的数据量会比较庞大，一些问题可能不容易发现，导致设计出的产品可能只能适用于小部分人群，但是用于小部分人群的话，显然不联网型会更适合，所以就没必要选择联网型，这是其一。其二，联网型所需要的设备有所增加，所以也会增加设计的成本，而且在检验的时候，因为没有那么庞大的数据量，所以可能存在一些设计问题因为数据量较小的原因，所以无法检查出其设计问题，然而当真正要投入使用的时候，可能会因为没有发现的问题而是矛盾增加。

## 2.2.2 主控芯片的选择

方案一：采用 ARM9 微处理器

ARM9 系列处理器是英国 ARM 公司设计的主流嵌入式处理器。

方案二：采用 STC89C52 单片机

STC89C52 是 STC 公司生产的一种微控制器。

此次设计选择了 STC89C52，究其原因有三。其一由于其成本较低。其二因为其功能可以满足设计要求。其三因为其发展也比较成熟。而 ARM 系统固然可以在功能各个方面上远优于 52 单片机系统，但是也因此可能在资源上会有所浪费。并且其价格比起 52 单片机昂贵太多，用一个昂贵的价格去购买一个有很多功能并不会在设计中使用即会浪费相当大资源的芯片，是一个完全没必要的行为。而且作为设计者也没有什么使用 ARM 的经验，相比于在大学期间，曾经多次使用 52 单片机的人来说，选择 STC89C52 作为主控芯片是一个好的选择。

### 2.2.3 显示模块的选择

方案一：采用数码管显示

数码管显示亮度高、成本低廉，可靠性高。但是在此次设计当中，希望能够显示出足够多的汉字来做一个声明或者提醒之类的文字，所以在此次设计中若要达成上述目标，选择采用数码管的话，可能需要更高的成本来实现这一目标，而且使用起来也可能不够美观，如果出现问题解决起来也较为复杂。

方案二：采用 LCD 液晶显示

一：选择 LCD1602

1602 液晶是一种专门用来显示字母、数字、符号等的点阵型液晶模块。

二：选择 LCD12864

LCD12864 是一种内部含有国标一级、二级简体中文字库的点阵图形液晶显示模块。

此次设计选择采用 LCD12864 液晶显示，首先 LCD12864 具有中文字库，其次使用简单。其价格相比 LCD1602 虽然略高，但相差不大，完全在可以接受的范围之内，而且 LCD12864 相比于 LCD1602，还有以下两个优点，其一 LCD12864 的显示窗口更大。其二，因为具有中文字库，所以 LCD12864 在此次设计中更能为使用者提供足够的文字说明，便于使用者进行操作。

### 2.2.4 键盘输入电路的选择

方案一：采用独立按键电路输入

此方法使用起来简单，只需要将 I/O 口和按键相连，在连接到地即可，当按下下一个按键的时候，其中一位 I/O 口电压被拉低，单片机就可以检测到是哪个按键被按下，只需要一次就可以确认是否有按键被按下，并且可以确认是哪一位按键被按下。但是此方法只适用于按键数目较少的情况，这是它的局限性。若是需

要使用的按键数目量过大的话，不仅会占用相当大量的 I/O 口，而且其效率使用率还并不高，

方案二：采用 4 乘 4 矩阵键盘模块

此模块使用起来比独立按键电路要复杂，不可能通过一次检测就可以确认具体是哪一位按键被按下，但是可以大量减少 I/O 口的使用。对比一下采用独立按键电路输入和采用 4 乘 4 矩阵键盘模块输入的效率。可以很清楚的看到同样是选用了 8 个 I/O 口，但是采用 4 乘 4 矩阵键盘模块比采用独立按键电路的按键数目多出一倍，而且随着 I/O 数目的增多，这个差距会越来越大。

虽然此次设计中的按键数目并不是太多，但是需要的按键数量依旧要达到十几个的数量，若是选择独立按键电路的话，就要提供十几个 I/O 口，虽然要提供的话还是可以提供出来的，但是如此以来，在其他模块的设计上就会遇到一些更大的困难。但是如果采用 4 乘 4 矩阵键盘，只须要 8 个 I/O 口就能实现的设计，而且还会有一些按键剩余，并且使用的 I/O 口还是远远少于独立按键电路的<sup>[3]</sup>。所以选用 4 乘 4 矩阵键盘模块作为按键输入。



### 第三章 硬件电路设计

#### 3.1 电路原理

##### 3.1.1 门禁系统主要组成

此次设计中，选择了 52 单片机，那么最小系统必然是需要的。其次就需要拓展的功能了，包括刷卡时需要使用的 IC 卡读卡模块，输入密码时需要使用的 4 乘 4 矩阵键盘模块，可以让直观看到的 LCD12864 液晶显示模块，还有打开门禁的继电器驱动电路，最后还需要使用一个小小的 LED 灯来对进行打开门禁是进行一种显著的提示，并且还需要使用蜂鸣器的震动发生来警示错误情况的发生，如刷卡错误或密码输入错误等都包含在内。

构成如图 3.1 示：

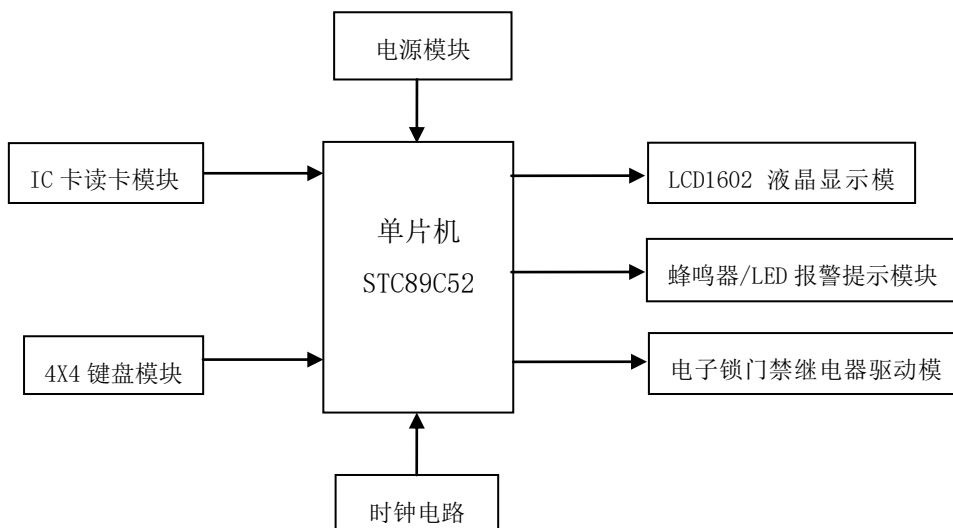


图 3.1 系统总体框图

##### 3.1.2 门禁系统工作原理

此次设计中一共设计了 5 个部分，第一部分是通过 RFID 和 IC 卡进行刷卡来打开门禁的开门模式，第二部分是通过输入密码，验证密码正确后的开门模块，第三部门是添加新卡模块，第四部分是注销旧卡模块，第五部分是修改密码模块。

当选择刷卡模块的时候，需要将卡放置到 IC 卡读卡模块，若卡已经获得权限，则 LCD 液晶显示屏会显示“刷卡成功”，并呈示卡的 ID 和持卡人姓名。同时门禁开启的指示灯点亮，继电器电位转变打开门禁的电磁锁，此时人们就可以安全通过门禁。延时大概几秒后继电器关闭电磁锁，门禁控制灯熄灭。此时人们就不能再通过了，若再想通过，就需要再次刷卡了。若卡未获得权限或者已经被

注销，在刷卡的时候 LCD 液晶显示屏会仍会显示卡的 ID，但是不在显示“刷卡成功”的字样，而是显示“失败！未注册！”的提示。并且错误次数会加 1，此时 LCD 显示屏上也会显示出数字“1”，这是提醒刷卡失败的次数为 1 次了。此时若你继续刷卡失败的话，则错误次数会再加 1，即现在的错误次数已达两次，并且 LCD 显示屏上也会显示出数字“2”这是提醒刷卡失败的次数为 2 次了。若再次失败，也就是当错误次数加到 3 的时候，蜂鸣器就会启动进行报警。

当选择密码开门模块的时候，直接输入 8 位数字密码，若密码准确，则液晶显示屏会显示提示，同时指示灯点亮，继电器打开电磁锁，此时人们就可以安全通过门禁。延时大概几秒后，继电器关闭电磁锁，指示灯熄灭。此时人们就不能再通过了，若再想通过，就需要再次通过刷卡或者重新输入密码的方式进入了。若密码错误，显示屏首先会显示“密码错误！”的字样，接着显示“请重新输入！”，并且错误次数会加 1，错误次数计数和启动蜂鸣器报警的原理和刷卡一致。

当选择添加新卡模块时，首先需要在 IC 刷卡模块先放置管理员的卡，相当于是管理员先刷卡，表示一个授权的行为，表示接下来的卡是确定要赋予它打开门禁的权限的。这样方便管理，而且也给管理做了一些安全性的保障。管理员刷卡成功后，会显示出“请放置新卡！”的字样，此时将新卡放置到 IC 刷卡模块时，若卡是未授权的，则会显示卡的 ID 并显示“卡已注册成功！”的字样。此时，新卡已经获得授权，可以在刷卡模块成功打开门禁。若卡是已经授权过的，则不会再次赋予它权限，而是会显示卡的 ID 并显示出“此卡已注册！”的字样，以此来告诉或者说提醒注册人该卡已经是注册过的，即该卡本来就具有打开门禁的权限，已经不需要再次注册了。

当选择注销旧卡模块时，首先仍需要将管理员的卡放置到 IC 刷卡模块，和添加新卡时设计相同，也是为了表示一个授权的行为，表示接下来的卡是确定要注销掉的。这样方便管理，而且也给的管理做了一些安全性的保障。待管理员刷卡后，并显示出了“请放置注销卡！”后。此时将需要注销的卡放置到 IC 刷卡模块，若卡是已经获得权限的，则会显示卡的 ID 并显示“卡已注销成功！”的字样。此时，该卡已经没有了打开门禁的权限，若再在刷卡模块进行刷卡的话，就会显示“失败！未注册！”的字样。若注销的时候放置到 RFID 刷卡模块的是本来就没有获得权限的话，则无法成功注销。会显示“不是管理员”的字样来提醒管理员该卡是已经注销过的，无须再次注销了。

当使用者觉得密码不安全或者其他一些原因需要修改密码的时候，可以通过修改密码模块，改变密码。若选择了修改密码模块，首先需要做的就是输入 8 位原密码，这里和密码开门模块还有一点设置不一样，密码开门模块，只要你 8 位密码输入正确，在输入之后，系统自己会作出判断，判断你的密码是否正确，是否要开门。但是这里为了做一些安全性的保障，在输入原密码完成之后需要自

已按下 4\*4 矩阵键盘上的 “#” 键，这是设置的一个确认键，也是做一些安全性的保障，避免在别人随意操作下或尝试下，误进入了设置新密码的模块。这里只有在你真正确认密码无误并且知道这一方法的人才能真正修改密码成功。在输入密码之后，自己已经确定无误对的时候，再按下 “#” 键，单片机会将输入的密码与保存的密码一位一位进行比对，若完全一致，则会显示“请输入新密码”的字样，此时需要输入 8 位的密码，作为新的密码，输入完成后按下 “#” 键表示确认新的密码已经输入成功了，此时单片机会将新的密码记录下来，用以在之后的密码开门模块进密码的一一比对<sup>[4]</sup>。此时会显示“密码已设置成功！”的字样。此为第一种情况，第二种情况则是当让输入原密码的时候，将输入的密码不正确，此时显示屏会显示提示告知。

### 3.2 资源分配

对于此次设计，将资源分配如图 3.2 和表 3.1 所示：

晶振采用 11.0592MHZ；

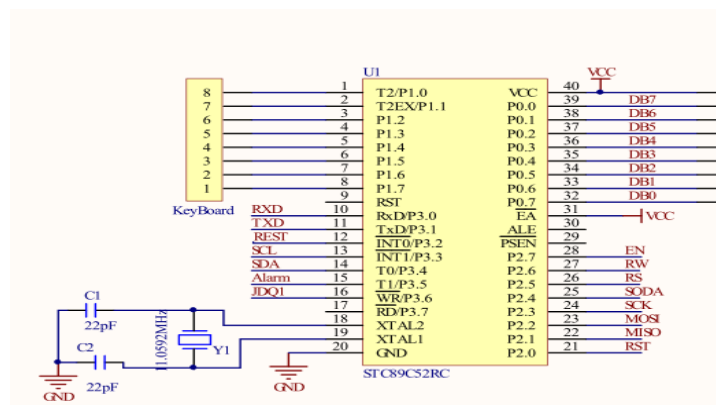


图 3.2 资源分配图

表 3.1 资源分配表

P0.0-P0.7	P1.0-P1.7	P2.0	P2.1-P2.4	P2.5
LCD 显示	矩阵键盘	RST	RC522	RS
P2.6	P2.7	P3.0	P3.1	P3.2
RW	EN	RXD	TXD	REST
P3.3	P3.4	P3.5		
SCL	SDA	蜂鸣器		
0 开	0 响	0 亮		
1 不开	1 不响	1 不亮		

### 3.3 最小系统设计

#### 3.3.1 STC89C52 的简述

##### 1. STC89C52 的简介

STC89C52RC 是 STC 公司生产的一种低功耗、高性能 CMOS8 位微控制器。

STC89C52 引脚图如图 3.3 示：

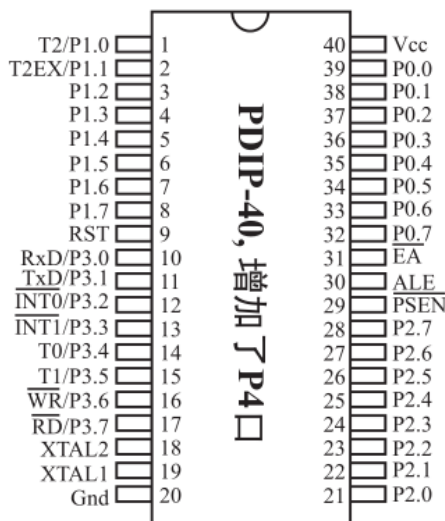


图 3.3 STC89C52 的引脚图

在此次设计中，除最小系统之外，还需要在外围增加一些所需要的其他控制电路，如：刷卡时需要使用的 IC 卡读卡模块，输入密码时需要使用的 4 乘 4 矩阵键盘模块，可以让直观看到的 LCD12864 液晶显示模块，还有打开门禁的继电器驱动电路，最后还需要的蜂鸣器/LED 报警提示模块<sup>[5]</sup>。

#### 3.3.2 最小系统电路设计

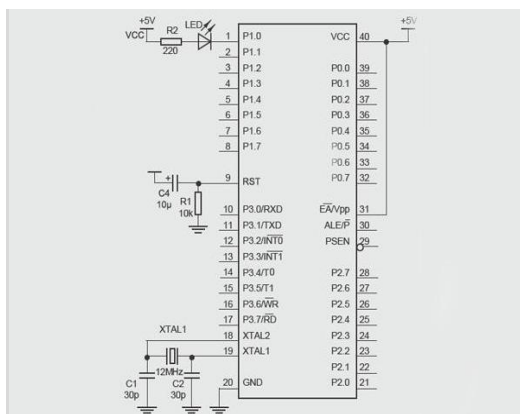


图 3.4 STC89C52 最小系统

### （1）复位电路设计

单片机的复位分为上电自动复位和按钮手动复位两种<sup>[6]</sup>。

这里采用按键复位来实现, 如图 3.5 所示

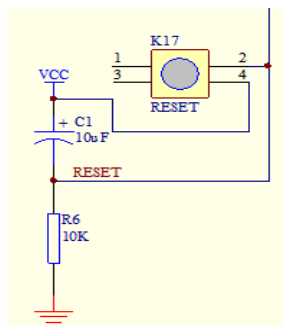


图 3.5 复位电路

### （2）时钟电路设计

时钟电路用于产生单片机所需要的时钟信号。

电路设计如图 3.6 所示

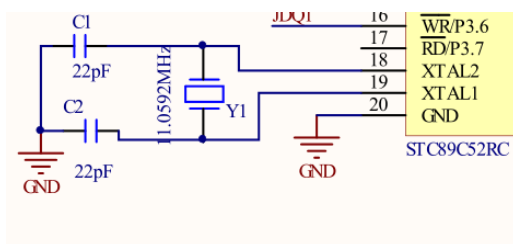


图 3.6 系统时钟电路

## 3.4 门禁控制电路

### 3.4.1 继电器介绍

继电器是具有隔离功能的自动开关元件, 广泛应用于遥控、遥测等电力电子设备中, 是最重要的控制元件之一。

### 3.4.2 继电器驱动电路设计

本次设计中, 使用 PNP 三极管来驱动电路, 目的是为了增大驱动电流。因为单片机 I/O 口驱动电流太小<sup>[7]</sup>。

控制锁电路如图 3.7 所示

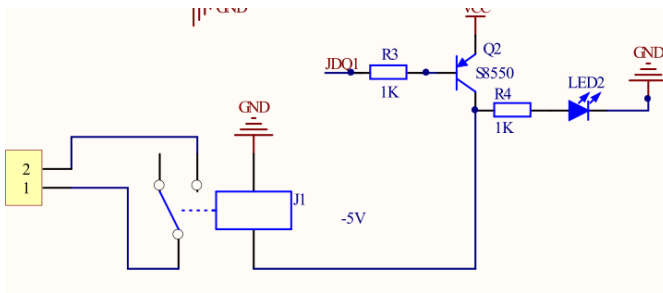


图 3.7 控制锁电路图

### 3.5 蜂鸣器电路

蜂鸣器电路设计如图 3.8 所示

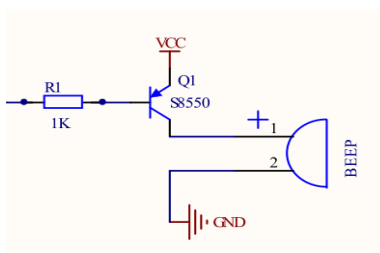


图 3.8 蜂鸣器电路图

### 3.6 控制指示灯

工作指示灯电路设计如图 3.9 所示

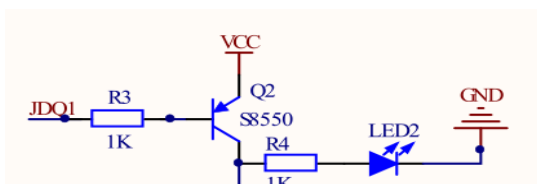


图 3.9 工作指示灯电路图

### 3.7 4×4 矩阵键盘

矩阵键盘设计如图 3.10 所示

各键位赋值表如表 3.2 所示

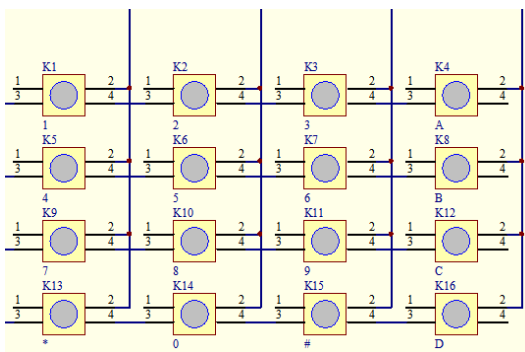


图 3.10 4×4 矩阵键盘电路图

表 3.2 各按键赋值表

1	2	3	上一项
4	5	6	下一项
7	8	9	
	0	确认	

### 3.8 读卡模块

设计上采用 IC 射频卡读卡器。

IC 射频卡读卡器优缺点如表 3.3 所示

表 3.3 IC 射频卡读卡器优缺点

优点：	缺点：
（1）独立性高	电脑需要保证一定和固定开机的时间，便于数据之间的交互。
（2）射频卡不分正反面，而且比起接触 IC 卡，磨损程度大大降低。	
（3）安全性高，RFID 不需要瞄准读取，只要被置于设备形成的电磁场内就可以准确读到 <sup>[8]</sup> 。	
（4）可选择数据输出格式	
（5）RFID 的读取高效且准确	

读卡器示意图如图 3.11 所示

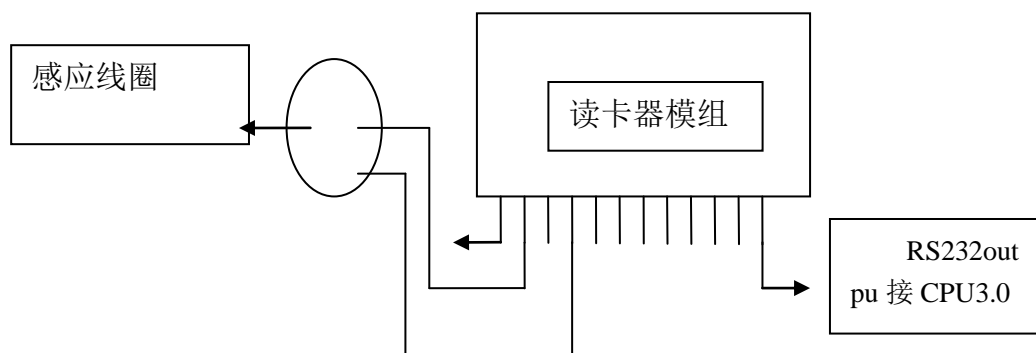


图 3.11 读卡器示意图

读卡器引脚连接图如 3.12 示:

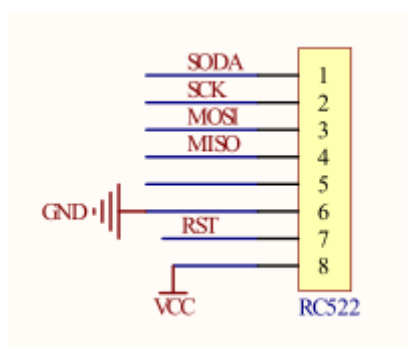


图 3.12 读卡器引脚连接示意图



## 第四章 软件电路设计

### 4.1 系统主程序分析

#### 1. 主程序流程图

主程序流程图如图 4.1 示：

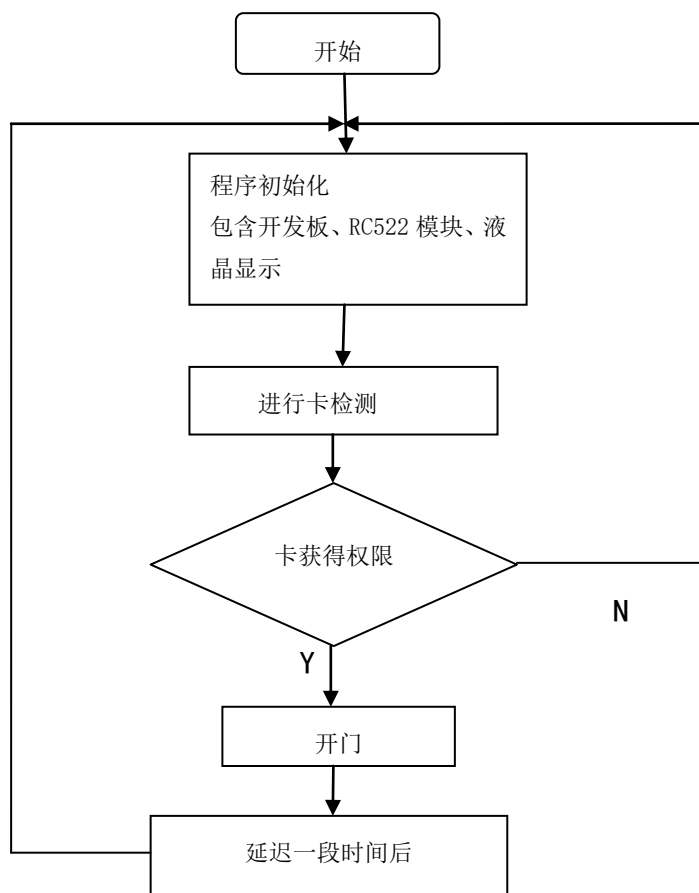


图 4.1 主程序流程图

#### 2. 主程序清单：

主函数：

在程序的一开始，首先需要将开发板、rc522 模块、液晶显示模块等都进行初始化，以方便后面的使用。如图 4.2 所示。

```
void init_all()          //总初始化程序
{
    init_mcu();          //开发板初始化
    init_rc522();        //rc522初始化
    lcd_init();          //液晶显示初始化

    EA = 0;              //关总中断
    init_timer();        //定时器初始化
    init_uart();         //串口初始化

    EEPROM_READ(0, Card_SN_BUF, 4); //EEPROM初始化,取出记录的卡号,第一扇区的存储内容
    EEPROM_READ(1, NewCard1, 4);    //EEPROM初始化,取出记录的卡号,第二扇区的存储内容
    EEPROM_READ(2, NewCard2, 4);    //EEPROM初始化,取出记录的卡号,第二扇区的存储内容
    EEPROM_READ(7, KEY_BUF, 8);    //EEPROM初始化,取出记录的密码,第八扇区存储密码

    send_bytes(NewCard1, 4);
    send_bytes(NewCard2, 4);
    send_bytes(Card_SN_BUF, 4);

    EA = 1;              //开总中断
}
```

图 4.2 初始化程序图

## 4.2 子程序分析

### 4.2.1 键盘程序

1. 键盘程序流程图如图 4.3 示：

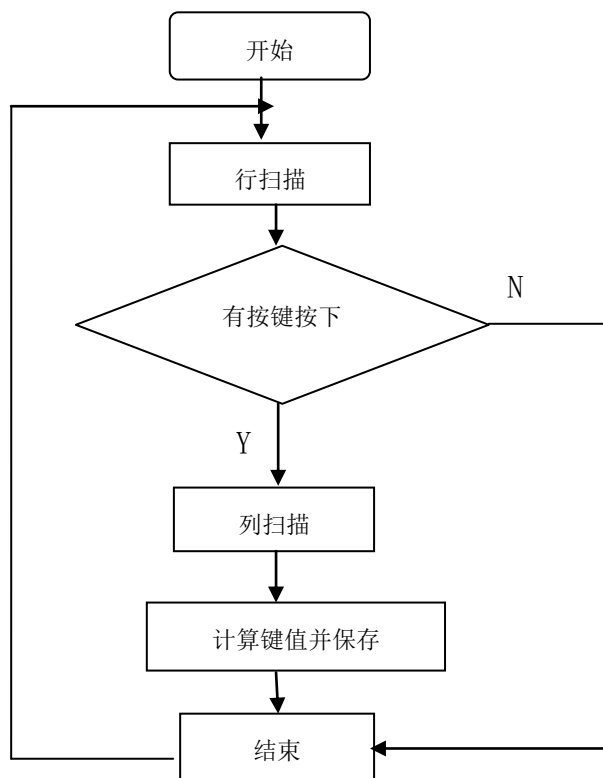


图 4.3 键盘程序流程图

## 2. 键盘原理分析

对于矩阵键盘，想确认是哪一个按键被按下比起独立键盘来说，不容易确认，因为不可能只通过一个端口的数据，就得出结论。需要有更高级的算法，才能得出结论。这里是采用一行一行分析的方法。

首先高四位给成高电位，低四位给成低电位。此时只要高四位的电位不发生变化，就表示没有按键被按下。如果有按键被按下的话，则高四位中必然会有高电平被拉低。依次做依据进行判断。

此时，为了确认是哪一个按键被按下，就需要通过一种特殊的算法，可以称之为“行列逐级扫描法”来确认。首先，要确定已经有按键被按下，当确认之后，就能去找寻具体是哪一个按键被按下。用图 4.4 来做一个具体的说明的话，PA0~PA3 分别连接的是第一、二、三、四行。同理可知 PA4~PA7。首先先把连接第一行的全部按键的这一位的电位先置成低电平，即把 PA0 置成低电平，其余全部置成高电平，即 PA1~PA7 全部置为高电平。此时检测 PA4~PA7 的值，若 PA4 电位被拉低的话，因为 PA4 连接的是第一列，加上一开始置成低电平的是第一行，所以就可以得知被按下的按键图中的 S1 按键。由此可以得知当 S2、S3、S4 按键分别被按下的时候，PA5、PA6、PA7 分别会由高电位变为低电位。若当赋予 PA0 的值为 0 的时候，而赋予 PA1 至 PA7 的值为 1 的时候，若检测 PA4~PA7 的值和赋予的值一样的时候，即 PA4~PA7 的值没有发生变化，就可以得知按下的按键不属于第一行，由此就排除了 S1、S2、S3、S4 四个按键。这个时候，赋予 I/O 口的值就可以修改为 1111 1101，即 0xfd。同理，可知这是用来检测第二行的，已经可以确定 PA1 为低电平，只需要检测 PA4~PA7 的电位就可以知道 S5 至 S8 的按键是否被按下，又是哪一个按键被按下。第三行和第四行同理。

矩阵键盘接法如图 4.4 所示

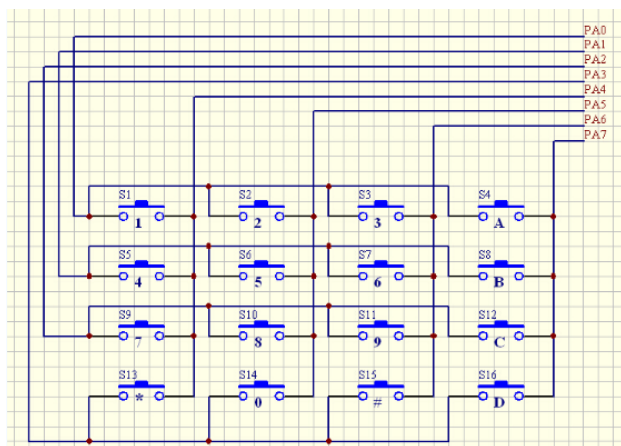


图 4.4 4 乘 4 矩阵键盘



图 4.5 矩阵键盘实物图

矩阵键盘实物图如图 4.5 所示

将图 4.4 和图 4.5 进行比较，设定实物图中的数字按键“1”对应图 4.4 中的 S1 按键，数字按键“2”对应图 4.4 中 S2 按键，数字按键“3”对应图 4.4 中 S3 按键，字母按键“A”对应图 4.4 中 S4 按键，这是第一行，代码如图 4.6 所示，其他行同理可得。

```

P1=0xfe;           //P1的八位前四位控制行，后四位控制列，都为低电平有效    此处为：第一行
temp=P1;
temp=temp&0xf0;
if(temp!=0xf0)
{
    delay(100);
    temp=P1;
    temp=temp&0xf0;
    if(temp!=0xf0)
    {
        temp=P1; //按键传送的值赋给temp，用来给下面switch做判断
        switch(temp)
        {
            case 0xee:key=1;break;
            case 0xde:key=2;break;
            case 0xbe:key=3;break;
            case 0x7e:key=12;break;
        }
    }
}

```

图 4.6 第一行的按键赋值

这是检测第一行是否有按键按下并且确定具体是哪一个按键的思路，给 P1 赋值 0xfe，即前 7 位都为 1，只有最后一位为 0，表示检查的是第一行。此时把 P1 的值赋予给 temp，再将 temp 和 f0 相与，因为 f0 高四位都为 1，无论和谁相与都是原来的数值，不会发生改变。又因为 temp 一开始赋予的值为 0xfe，所以高四位置成的电位全部为高电位，与 0xf0 相与，高四位的电位应该仍全为高电位，不会发生变化，由此可以得知，当检测到 temp 的高四位电位发生了变化的时候，必然可以排除检测到的电位变化是因为“与”运算产生的。所以当检测高四位的电位发生了变化的时候，一定是因为有按键被按下的原因。

所以当没有按键按下时，temp 和 f0 相与，应该得到的仍是 f0。而 temp 和 f0 相与得到的结果不是 f0 的时候，就表示 P1 的高四位中有的位置上的数值不是 1，那么就可以证明是有按键被按下了，而且因为一开始赋予 P1 的值是 fe，所以可以知道是第一行的按键被按下了，那么剩下的就是要知道是哪一列的就可以确定是那个按键了。此时只需要去检测高四位的电位变化就可以知道了。当检测到 P1 的数值变为 0xee 的时候，即变为 1110 1110，即第一列的数值发生了变化，由此可以知道是第一行和第一列共同连接的按键，即 S1 按键被按下。

由程序可知设定当 S1 按键被按下的时候, key 的值为 1, 同理可知, 当 S2、S3、S4 按键分别被按下的时候, 分别令 key = 2、3、12。

同理, 可以得知当按下按键 S5、S6、S7、S8 的时候, 分别令 key=4、5、6、13。第三行和第四行的 S9 至 S12 和 S13 至 S16 按键被按下的时候, key 分别为 7、8、9、14 和 10、0、11、15。如图 4.7、4.8、4.9 所示。

```
P1=0xfd;    //P1的八位前四位控制行, 后四位控制列, 都为低电平有效 此处为: 第二行
temp=P1;
temp=temp&0xf0;
if (temp!=0xf0)
{
    delay(100);
    temp=P1;
    temp=temp&0xf0;
    if (temp!=0xf0)
    {
        temp=P1; //按键传送的值赋给temp, 用来给下面switch做判断
        switch (temp)
        {
            case 0xed:key=4;break;
            case 0xdd:key=5;break;
            case 0xbd:key=6;break;    //
            case 0x7d:key=13;break;
        }
    }
}
```

图 4.7 第二行的按键赋值

```
P1=0xfb;    //P1的八位前四位控制行, 后四位控制列, 都为低电平有效 此处为: 第三行
temp=P1;
temp=temp&0xf0;
if (temp!=0xf0)
{
    delay(100);
    temp=P1;
    temp=temp&0xf0;
    if (temp!=0xf0)
    {
        temp=P1; //按键传送的值赋给temp, 用来给下面switch做判断
        switch (temp)
        {
            case 0xeb:key=7;break;
            case 0xdb:key=8;break;
            case 0xbb:key=9;break;
            case 0x7b:key=14;break;
        }
    }
}
```

图 4.8 第三行的按键赋值

```
P1=0xf7;    //扫描第四行
temp=P1;
temp=temp&0xf0;
if (temp!=0xf0)
{
    delay(100);
    temp=P1;
    temp=temp&0xf0;
    if (temp!=0xf0)
    {
        temp=P1;
        switch (temp)
        {
            case 0xe7:key=10;break;
            case 0xd7:key=0;break;
            case 0xb7:key=11;break;
            case 0x77:key=15;break;
        }
    }
}
```

图 4.9 第三行的按键赋值

其中 key 的值从 0 至 9 共 10 个按键分别表示 0 至 9 这 10 个数字, 用来在输入密码模块里面输入密码时使用。而 key 值是 12 和 13 的 A 和 B 两个按键, 则是在硬件上切换不同的模块时使用, A 切换上一个功能模块, B 切换下一个功能模块, 代码如图 4.10 所示。

```
if(keys==13)          // B : 表示下一功能界面
{
    states++;
    if(states==5)
        states=0;
}

if(keys==12)// A : 表示上一功能界面
if(states>0)
    states--;
else if(states==0)
    states=4; //上一功能
else
    states=0;
```

图 4.10 切换上下功能键设置

#### 4.2.2 操作模块

设计的时候提到过，此次设计中一共设计了 5 个部分，一部分是通过 RFID 和 IC 卡进行刷卡来打开门禁的开门模式<sup>[9]</sup>，第二部分是通过输入密码，验证密码正确后的开门模块，第三部分是添加新卡模块，第四部分是注销旧卡模块，第五部分是修改密码模块。

首先在键盘上设置的有功能选择按键，分别为按键 A 和 B，其中 A 表示的是切换上一项功能界面，而 B 表示的是切换下一项功能界面，因为每一个界面的显示会不一样，所以每当有按键按下，即表示要切换功能界面的时候，首先需要清屏，然后根据选择的功能显示不同的界面和提供不同的功能。

##### 1 刷卡模块

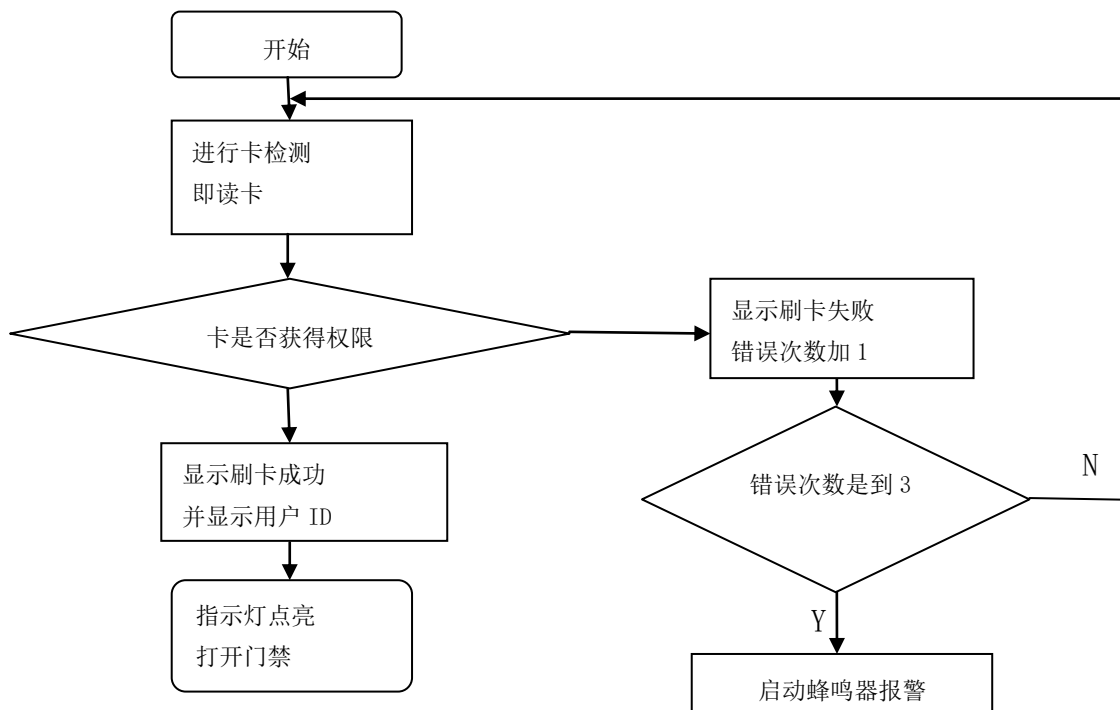


图 4.11 刷卡模块流程图

刷卡模块流程图如图 4.11 所示

程序：

管理员刷卡时显示“刷卡成功 管理员”，流程图如图 4.12 所示。

```

case 0: //IC卡读卡输入（默认状态）
    lcd_mesg("***RFID门禁系统***",0x80,16);
    lcd_mesg("1.请刷卡:",0x90,16);
    if(IC_READ()) //读卡程序
    {
        if //判断卡号和存储的卡号是否相同 读取的卡号和EEPROM里的数值比较
        {
            (Card_SN[0]==Card_SN_BUF[0])&&(Card_SN[1]==Card_SN_BUF[1])
            &&(Card_SN[2]==Card_SN_BUF[2]) &&(Card_SN[3]==Card_SN_BUF[3])
        )
        {
            card_correct1(); //显示刷卡成功 管理员
            count_card=0;
        }
        else if
        {
            (Card_SN[0]==NewCard1[0])&&(Card_SN[1]==NewCard1[1])
            &&(Card_SN[2]==NewCard1[2]) &&(Card_SN[3]==NewCard1[3])
        )
        {
            card_correct2(); //显示刷卡成功 用户1
        }
    }

```

图 4.12 管理员刷卡程序图

用户刷卡成功时会显示用户 ID，程序图如图 4.13 所示

```

        count_card=0;
    }
    else if
    {
        (Card_SN[0]==NewCard2[0])&&(Card_SN[1]==NewCard2[1])
        &&(Card_SN[2]==NewCard2[2]) &&(Card_SN[3]==NewCard2[3])
    }
    {
        card_correct3(); //显示刷卡成功 用户2
        count_card=0;
    }
    else
    {
        if(count_card!=1 && count_card!=2 && count_card!=3 )
        {
            count_card=0;
        }
        count_card++; //错误次数
        card_fault(); //显示刷卡失败
        display_num(0x96,(48+count_card));
    }
}
break;

```

图 4.13 用户刷卡成功程序图

每当刷卡不成功时，错误次数就会加 1，当加到 3 的时候，就会启动报警系统，首先蜂鸣器响进行报警，同时将错误次数清零。程序如图 4.14 所示。

```

        if(count_card>=3)
        {
            lcd_mesg("报警系统已启动!",0x98,16);
            beepss(); //防盗报警系统启动
            display_clear(3);
            count_card=0;
        }
    } //break;
}
break;

```

图 4.14 启动报警器程序图

射频识别，RFID（Radio Frequency Identification）技术，又称无线射频识别，是一种通信技术，可通过无线电信号识别特定目标并读写相关数据，而无需识别系统与特定目标之间建立机械或光学接触<sup>[10]</sup>。

首先，在设置的时候因为有要添加新卡和注销卡的功能，为了保险起见，所以设置的卡的级别有管理员和普通用户两个级别，在添加新卡和注销卡的时候都需要管理员的卡先给权限才能继续操作，这个后面会说到。现在先说明刷卡进门的这个部分。

首先，要先判断刷的卡是不是具有能打开门禁的权利，因为在添加新卡的时候，会有所记录，所以这里的判断方法就是，将刷卡的卡号与保存的卡号进行比较，若相同，则表示是具有权限的，否则不具有权限。具有权限的又分两种情况，在将卡号进行比较的时候，若是与管理员的卡号一致，则表明是管理员，否则就是一般用户。若比较卡号结果不相同，那么首先错误次数会加1，待错误次数加到3则会启动蜂鸣器报警。

## 2、密码输入模块

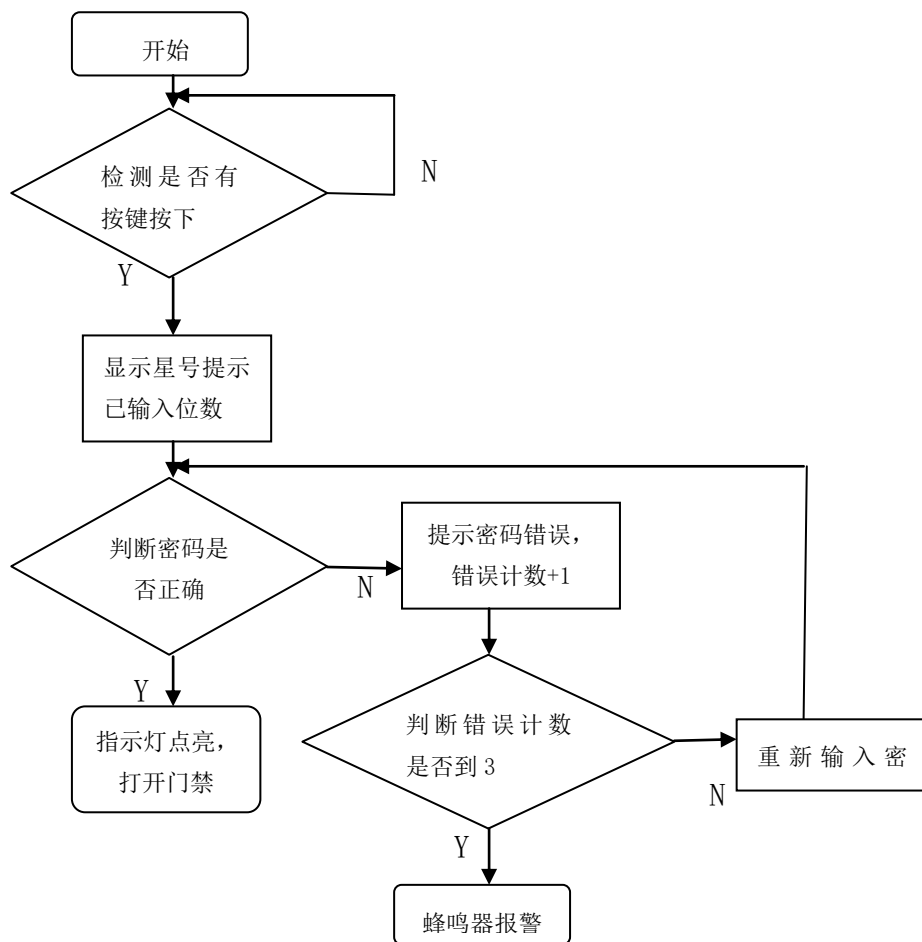


图 4.15 密码输入模块流程图



密码输入模块流程图如图 4.15 所示。

程序：

要使用密码模块开门，首先要切换到密码模块。上文曾提到过，我们将 A、B 两键分别设定为切换上一功能和切换下一功能。所以从刷卡模块到密码模块我们就需要用到这两个按键。程序图如图 4.16 所示

```
case 1:
    lcd_msg("**RFID门禁系统**",0x80,16);
    lcd_msg("2.请输入密码:",0x90,13);           //第二种开门方式，用密码开门
    lcd_msg("-----",0x98,16);
    key_count=0; //按键次数
    while(1)
    {
        key_value=keyscan();           //进入下一菜单，任然可以选择其他功能

        if(key_value==13)
        {
            states++;
            if(states==5)
            {
                states=0;
                return;
            }
        }
        if(key_value==12)
        {
            states--;
            if(states < 0)
            {
                states=4;
                return;
            }
        }
        if(key_value>=0 && key_value <=9) //有按键输入
        {
            table[key_count++]=key_value+'0'; //转换成ASCII码值，显示数字，存储起来
        }
    }
}
```

图 4.16 切换到密码模块程序图

当输入完密码之后，就需要进行判断密码是否正确了，以便用来判断是打开门禁还是进行报警。判断密码正确与否的程序图如图 4.17 所示。

```
/****** 有按键输入，显示星号，提示已输入的位数 *****
    if(key_count<=8)
        lcd_msg("*****",0x98+0x02,key_count);

}
if(key_count==8)
{
    if( table[0]==KEY_BUF[0] &&
        table[1]==KEY_BUF[1] &&
        table[2]==KEY_BUF[2] &&
        table[3]==KEY_BUF[3] &&
        table[4]==KEY_BUF[4] &&
        table[5]==KEY_BUF[5] &&
        table[6]==KEY_BUF[6] &&
        table[7]==KEY_BUF[7] ) //密码正确
    {
        beeps();
        card_correct1();
        count_code=0;
    }
}
```

图 4.17 判断密码正确与否程序图

当判断密码正确的时候打开门禁，当密码不正确的时候，错误次数加 1，当错误次数加到 3 的时候就和刷卡模块一样进行报警。通过蜂鸣器响进行报警，同时将错误计数清零。程序设计如图 4.18 所示。

```

else if
( table[0]!=KEY_BUF[0] ||
  table[1]!=KEY_BUF[1] ||
  table[2]!=KEY_BUF[2] ||
  table[3]!=KEY_BUF[3] ||
  table[4]!=KEY_BUF[4] ||
  table[5]!=KEY_BUF[5] ||
  table[6]!=KEY_BUF[6] ||
  table[7]!=KEY_BUF[7] )
{
    //密码错误
    if(count_code!=1 && count_code!=2 && count_code!=3 )
    {
        count_code=0;
    }
    count_code++; //错误计数
    beeps();
    display_num(0x97,(48+count_code));
    lcd_msg("密码输入错误!",0x98,14);
    delay(100);
    display_clear(3);
    lcd_msg("请重新输入!",0x98,12);
    delay(100);
    if(count_code>=3)
    {
        lcd_msg("报警系统已启动!",0x98,16);
        beeps(); //防盗报警系统启动
        display_clear(3);
        count_code=0;
    }
    break;
}

```

4.18 密码错误和报警程序图

这是第二种模块。设置密码必须为 8 位，所以只有满足 if(key\_count==8) 的一个判断依据，你的密码才有可能是正确的，所以是一个先决条件。之后，才会根据你输入的密码是否和记录的密码一致进行判断，如果 if( table[0]==KEY\_BUF[0])就是在判断输入的第一位和记录的密码第一位是否相同，只有 8 位都正确，才会显示密码正确，打开门禁。否则，就会显示密码错误，并且错误计数会加 1，当错误计数加到 3 的时候，就回启动蜂鸣器报警。

### 3、卡注册模块

卡注册模块流程图如图 4.19 所示

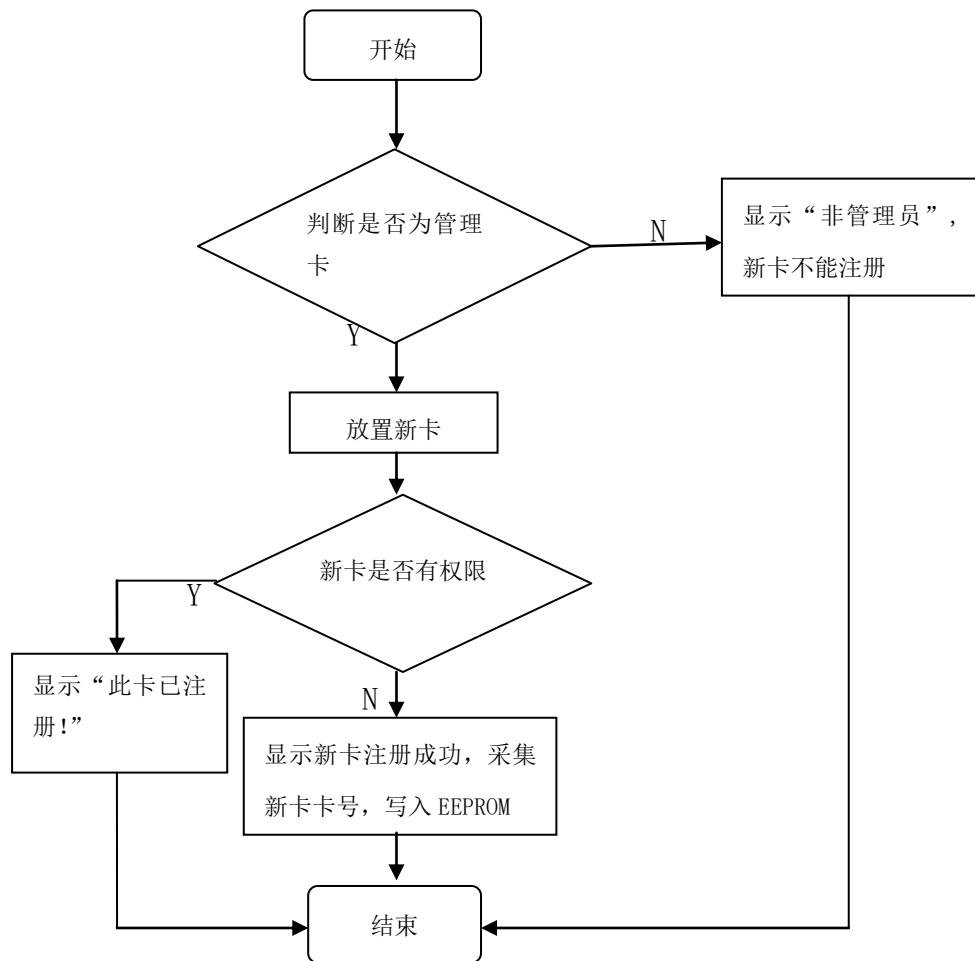


图 4.19 卡注册模块流程图

程序：

在注册卡号的时候，为了保证安全，所以我们需要在添加新卡之前，先使用管理卡进行授权，以此保证注册的新卡的合法性。程序设计如图 4.20 所示。

```

                                卡注册
*****
case 2:
    lcd_mesg("***RFID门禁系统**",0x80,16);
    lcd_mesg("3.IC登记注册",0x90,16);           //注册卡号
    if(IC_READ())
    {

        if                                //判断卡号是否为主卡
        {
            (Card_SN[0]==Card_SN_BUF[0])&&(Card_SN[1]==Card_SN_BUF[1])
            &&(Card_SN[2]==Card_SN_BUF[2]) &&(Card_SN[3]==Card_SN_BUF[3])
        }
        {
            beeps();
            lcd_mesg("请放置新卡!",0x98,12);
            delay(500);
        while(1)
        {
            key_value=keyscan();           //进入下一菜单，任然可以选择其他功能

            if(key_value==13)
            {
                states++;
                if(states==5)
                states=0;
                return;
            }
        }
    }
}

```

图 4.20 注册卡号先使用管理卡授权程序图

当在注册的时候，先使用管理卡进行授权之后，就可以添加新卡了，将新卡放置到刷卡模块上时，就可以看到显示屏显示出“卡已注册成功”的字样。程序设计如图 4.21 所示

```

    if
    {
        NewCard1[0]==0xff
    }
    {
        for(i=0;i<4;i++)
        {
            NewCard1[i] = Card_SN[i]; //新卡卡号采集
            EEPROM_WRITE(1,Card_SN,4); //写入EEPROM第二个扇区
        }

        {
            lcd_mesg("卡已注册成功!",0x98,14);
            delay(200);
            beeps();
            display_clear(2);
            display_clear(3);
        }
        break;
    }

    if
    {
        NewCard2[0]==0xff
    }
    {
        for(i=0;i<4;i++)
        {
            NewCard2[i] = Card_SN[i]; //新卡卡号采集
            EEPROM_WRITE(2,Card_SN,4); //写入EEPROM第二个扇区
        }
    }
}

```

图 4.21 新卡注册成功的程序图

这是注册新卡会用到的程序，首先第一点，上面也提到过，在注册新卡的时候，首先需要使用管理卡来授权，只有管理卡授予了权限，才能在这个模块添加新卡，所以在程序里第一个判断条件为“判断卡号是否为管理卡”。如果刷的不是管理卡，会提示“非管理员”，而且也无法进行新卡的注册。如果管理卡授权无误，那么接下来就可以添加新卡了，这里又可以分为两种情况，一种就是该卡已经有了开门的权限，而另一种情况就是该卡还没有开门的权限。为了区分这两种情况，所以首先对该卡进行判断，若存的记录的卡中含有这张卡，表示它是已经被赋予了开门的权限，那么就不会再次给这张卡注册，而是会在显示屏上提示，显示出“此卡已注册！”的字样。如果这张卡没有被记录，就代表它还是没有被授权的，此时，就会采集新卡的卡号并写入 EEPROM，并在显示屏上显示“新卡已注册成功”的字样，来告诉新卡已经注册完毕。

#### 4、卡注销模块

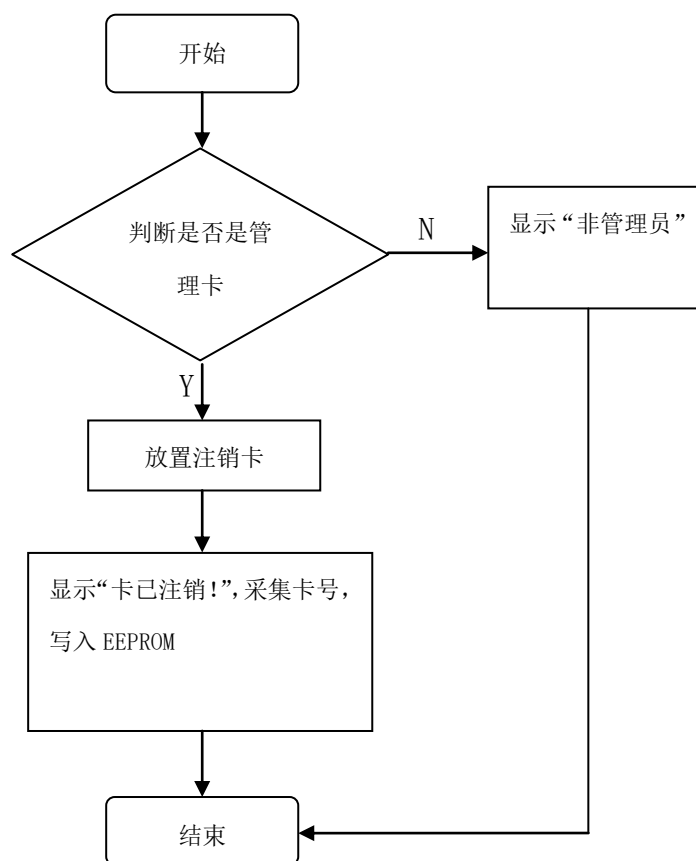


图 4.22 卡注销模块流程图

卡注销模块流程图如图 4.22 所示  
程序：

当要将卡注销时，和注册卡类似，我们同样要先使用管理卡授权，接着才能将卡进行注销。程序设计图如图 4.23 所示。

```

卡注销
*****
case 3:
    lcd_mesg("***RFID门禁系统**",0x80,16);
    lcd_mesg("4.IC卡注销",0x90,16);          //注册卡号
    if(IC_READ())
    {
        if
        //判断卡号是否为主卡
        (
            (Card_SN[0]==Card_SN_BUF[0])&&(Card_SN[1]==Card_SN_BUF[1])
            &&(Card_SN[2]==Card_SN_BUF[2]) &&(Card_SN[3]==Card_SN_BUF[3])
        )
        {
            beeps();
            lcd_mesg("请放置注销卡!",0x98,16);
            delay(500);
        }
        while(1)
        {
            key_value=keyscan();          //进入下一菜单，任然可以选择其他功能

            if(key_value==13)
            {
                states++;
                if(states==5)
                states=0;
                return;
            }
        }
    }
}

```

图 4.23 注销卡时先使用管理卡授权程序图

当使用管理卡授权之后，将需要注销的卡放置到读卡模块进行注销。程序设计如图 4.24 所示。

```

        if(key_value==12)
        {
            states--;
            if(states < 0)
            states=4;
            return;
        }

        if(IC_READ())
        {
            if( (Card_SN[0]==NewCard1[0])&&(Card_SN[1]==NewCard1[1])
            &&(Card_SN[2]==NewCard1[2]) &&(Card_SN[3]==NewCard1[3]) )
            {
                for(i=0;i<4;i++)
                {
                    NewCard1[i]=0xff; //新卡卡号采集
                    EEPROM_WRITE(1,NewCard1,4); //写入EEPROM第二个扇区
                }
                lcd_mesg("卡已注销!",0x98,16);
                delay(800);
                beeps();
                display_clear(2);
                display_clear(3);
            }

            if( (Card_SN[0]==NewCard2[0])&&(Card_SN[1]==NewCard2[1])
            &&(Card_SN[2]==NewCard2[2]) &&(Card_SN[3]==NewCard2[3]) )
            {
                for(i=0;i<4;i++)
                {
                    NewCard2[i]=0xff; //新卡卡号采集
                    EEPROM_WRITE(1,NewCard2,4); //写入EEPROM第二个扇区
                }
                lcd_mesg("卡已注销!",0x98,16);
            }
        }
    }
}

```

图 4.24 注销卡成功程序图

注销模块和注册模块一样，在注销卡之前首先需要管理员授权，只有在管理员授权了之后，才能成功将卡注销掉。所以和注册模块一样，需要在程序里将“判断卡号是否为管理卡”设为第一个判断条件。如果刷的不是管理卡，会提示“非

管理员”，而且也无法进行卡的注销。如果管理卡授权无误，那么接下来就可以进行注销了。只要将之前授权过的卡放置在刷卡模块上就可以完成卡的注销了。

#### 4、密码修改模块

密码修改模块流程图如图 4.25 所示

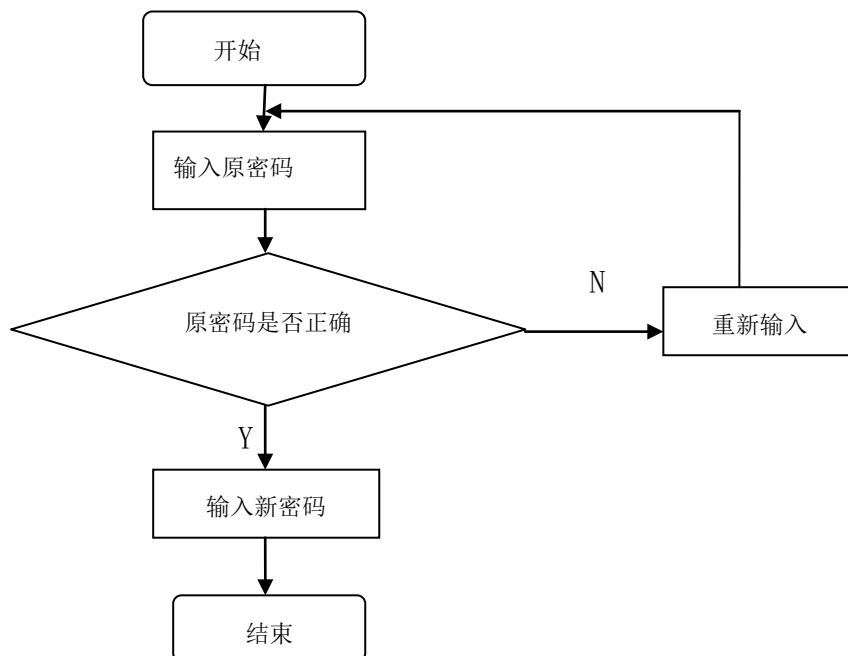


图 4.25 密码修改模块流程图

程序：

当需要修改密码的时候，需要先验证原密码，避免密码被恶意修改的情况发生。所以在修改密码之前，我们要先确认原密码的正确性。程序设计图如图 4.26 所示。

```

case 4:
    lcd_msg("**RFID门禁系统**",0x80,16);
    lcd_msg("5.密码设置:",0x90,10);
    lcd_msg("请输入原密码",0x88+0x01,12);
    lcd_msg("-----",0x98,16);
    key_count=0;
    while(1)
    {
        key_value=keyscan();

        if(key_value==13)
        {
            states++;
            if(states==5)
            {
                states=0;
                return;
            }
        }
        if(key_value==12)
        {
            states--;
            if(states < 0)
            {
                states=4;
                return;
            }
        }

        if(key_value>=0 && key_value <=9)//有按键输入
        {
            table[key_count++]=key_value+'0';

            if(key_count<=8)
            {
                lcd_msg("*****",0x98+0x02,key_count);
            }
        }
    }
  
```

图 4.26 修改密码前验证原密码的程序图

当原密码确定正确后，我们将新密码进行输入便可成功修改密码了。程序设计如图 4.27 所示。

```

if(key_count>=8 && key_value==11 ) //按下确定键
{
    if (
        table[0]==KEY_BUF[0]&&
        table[1]==KEY_BUF[1]&&
        table[2]==KEY_BUF[2]&&
        table[3]==KEY_BUF[3]&&
        table[4]==KEY_BUF[4]&&
        table[5]==KEY_BUF[5]&&
        table[6]==KEY_BUF[6]&&
        table[7]==KEY_BUF[7]
    )
    {
        lcd_mesg("请输入新密码",0x88+0x01,12);
        lcd_mesg("-----",0x98,16);
        key_count=0;
        while(1)
        {
            key_value=keyscan();
            if(key_value>=0 && key_value <=9) //有按键输入
            {
                table[key_count++]=key_value+'0';
                if(key_count<=8)
                    lcd_mesg("*****",0x98+0x02,key_count);
            }
        }
        if(key_count>=8 && key_value==11 ) //按下确定键
        {
            for(i=0;i<8;i++)
                KEY_BUF[i]=table[i];
            EEPROM_WRITE(7,KEY_BUF,8); //写入EEPROM
            lcd_mesg("密码已设置成功!",0x98,16); //表示密码设置完成
            beeps();
            break;
        }
    }
}

```

图 4.27 密码修改成功程序图

修改密码的模块这里，虽然不再需要管理卡授权，但是为了保险，一开始需要输入原密码，将此密码与原密码进行比对，无误的话，就可以进行密码的修改了。但是这里多做了一个设定，就是在输入完原密码后要自己按下确定按钮才可以，也是为了多一重防护吧。在输入完原密码之后，按下确认按钮，设置的是键盘上的“#”键，若密码正确，则可以进入输入新密码的界面，否则会显示屏上出现“请重新输入”的提示。在输入新密码的界面，在输入完新的密码后，仍然需要按下确认按钮，即键盘上的“#”键，确认过后，密码才算修改完成。



## 结束语

此次设计，首先制作了一个简单的可以通过密码就打开的门禁系统，在此基础上，又添加了 IC 卡刷卡的方式。因为密码门禁毕竟相对来说简单了一点，它的安全性可能已经不足以满足对当下对安全的严格要求了。这个技术毕竟使用时间已经不短了，它的破解方法也已经不少了。密码门禁一开始只有单纯的数字密码，数字密码虽说组合可以相当的多，但是毕竟还是只有 10 个数字。比起一开始的单纯的数字密码，虽然随着后来的发展，在密码当中也加入了英文字符，比起一开始确实复杂了不少，英文有 26 个字母，加上字母又区分大写和小写两种，再加上 10 位数字，其实它的随机组合的数量已经相当庞大了。因为其复杂程度，虽然安全性能有所提高，但是同样的，也因此致使一些用户在使用的时候可能频繁忘记自己设置的密码，导致一些尴尬的局面。

所以希望可以通过 IC 卡来进行记录，这个既不需要人们人为去记录什么密码，而且比起以前的钥匙来说，又方便了很多。首先，在以前的日常生活中，人们身上总会有相当数量的钥匙，尤其对于一些公司的管理钥匙的人来说，可能自身要携带的钥匙数量相当庞大。但是有了 IC 门禁卡之后情况就会有所变化，首先的 IC 卡一张卡可以打开多个门禁，比如一个白领的 IC 卡可以打开写字楼楼下的门禁、自己办公室的门禁、会议室的门禁等多个地方。甚至于可以根据级别的不同，赋予不同级别的人所持的卡不同的权限，比如你的级别越高，你的卡可以打开的门禁系统数量就会越多。

随着智能化的发展，相信以后的门禁技术会越来越成熟，它的保密性也会越来越好，越来越能够保护人们的安全和隐私。并且它的使用范围也会越来越增大的，现在很多小区的大门处，楼房的单元处，已经开始出现比起以往更多的门禁系统。人们不仅越来越适应这种智能的门禁系统，而且使用起来也是比起以往的门锁更加方便。相信以后智能门禁系统会出现在目之所及的各个地方，的生活中会到处充斥着它们。

## 致 谢

转眼间，已经在美丽的西安邮电大学度过了四个年头。四年，这是人生中十分重要的四年，有幸能够接触到这些不仅仅传授我知识、学问，而且从更高层次指导我的人生与价值追求的良师。他们使我坚定了人生的方向，获得了追求的动力，留下了大学生活的完美回忆。在此，我真诚地向尊敬的老师们和母校表达深深的谢意！

经过了两个多月的努力，我最后完成了论文的写作。从开始接到论文题目到系统的实现，再到论文文章的完成，每走一步对我来说都是新的尝试与挑战，这也是我在大学期间独立完成的最大的项目。我首先要感谢我的论文指导老师、西安邮电大学自动化学院的马翔老师。马老师对我论文的研究方向做出了指导性的意见和推荐，在论文撰写过程中及时对我遇到的困难和疑惑给予悉心指点，提出了许多有益的改善性意见，投入了超多的心血和精力。此外，还要感谢朋友以及同学们在论文编写中带给的大力支持和帮忙，给我带来极大的启发。也要感谢参考文献中的作者们，透过他们的研究文章，使我对研究课题有了很好的出发点。在这段时间里，我学到了很多知识也有很多感受，从一无所知，到开始了独立的学习和试验，查看相关的资料和书籍，让自己头脑中模糊的概念逐渐清晰，使自己十分稚嫩作品一步步完善起来，每一次改善都是我学习的收获，每一次试验的成功都会让我兴奋好一段时间。

我的论文作品不是很成熟，还有很多不足之处。但是这次做论文的经历使我终身受益。我感受到做论文是要真真正正用心去做的一件事情，是真正的自己学习的过程和研究的过程，没有学习就不可能有研究的潜力，没有自己的研究，就不会有所突破，那也就不叫论文了。期望这次的经历能让我在以后学习中激励我继续进步。

最后，谢谢论文评阅老师们的辛苦工作。衷心感谢我的家人、朋友，以及同学们，真是在他们的鼓励和支持下我才得以顺利完成此论文。

## 参考文献

- [1] 陆扬. 住宅小区建筑电气与智能化控制系统的设计[J]. 建筑工程技术与设计, 2016(29)
- [2] 马巧梅. 基于 STC89C52 的智能门禁系统的设计[J]. 机械制造与自动化, 2017, 46(05):235-238.
- [3] 王维建. Wi-Fi 无线加密移动硬盘的设计与实现[D]. 杭州: 杭州电子科技大学, 2016
- [4] 卢新文, 沈陆垚, 黄淼, 王华磊, 梁增辉. 基于 STC89C51 单片机的遥控智能开关控制器[J]. 电子测试, 2016(08):5-6.
- [5] Yue Jun He. Development of Key Reliability Tester[J]. Applied Mechanics and Materials, 2014, 3485(635).
- [6] 刘莹, 吴蓬勃, 李新龙. 基于单片机的精密仪器生产车间温湿度自动控制系统[J]. 决策与信息, 2014(5)
- [7] 梁佩仪, 吴晓君, 朱齐媛. 基于 STC89C52 单片机的宿舍门禁管理系统设计[J]. 佳木斯职业学院学报, 2015(02):328.
- [8] 江波. 移动图书馆架构下 RFID 的应用创新——西南政法大学图书馆 RFID 项目实践[J]. 图书馆论坛, 2015, 35(04):106-109+23
- [9] Jun Hai Jiang, Ming Hui Liu. A Study on Speed Control System Based on STC89C51[J]. Applied Mechanics and Materials, 2015, 3744(713).
- [10] Jun Hai Jiang, Zhi Guo Xiao. A Study on Liquid Level Measurement and Control System Based on Single Chip Microcomputer[J]. Applied Mechanics and Materials, 2015, 3744(713).

## 附录 A:

```
Main.c
#include "include.h"
void init_all()      //总初始化程序
{
    init_mcu(); //开发板初始化
    init_rc522(); //rc522 初始化
    lcd_init(); //液晶显示初试化
    EA = 0;      //关总中断
    init_timer(); //定时器初始化
    init_uart(); //串口初始化

    EEPROM_READ(0, Card_SN_BUF, 4); //EEPROM 初始化, 取出记录的卡号,
    第一扇区的存储内容

    EEPROM_READ(1, NewCard1, 4); //EEPROM 初始化, 取出记录的卡号, 第
    二扇区的存储内容

    EEPROM_READ(2, NewCard2, 4); //EEPROM 初始化, 取出记录的卡号, 第
    二扇区的存储内容

    EEPROM_READ(7, KEY_BUF, 8); //EEPROM 初始化, 取出记录的密码, 第八
    扇区存储密码

    send_bytes(NewCard1, 4);
    send_bytes(NewCard2, 4);
    send_bytes(Card_SN_BUF, 4);
    EA = 1;      //开总中断
}

/*****

主函数

*****/
```

```

void main()
{
    init_all(); //初始化
    while(1)
    {
/*****按键部分程序*****/
        unsigned char keys;
        keys=keyscan(); //按键操作
        if(keys!=16)
        {
            if(keys==13)          // B : 表示下一功能界面
            {
                states++;
                if(states==5)
                states=0;
            }
            if(keys==12) // A : 表示上一功能界面

            if(states>0)
                states--;
            else if(states == 0)
                states=4; //上一功能
            else
                states=0;
        }
        ctrl_process(); //进行监测卡程序
    }
}

```

Ctrl.c

```
#include "include.h"
```

```
uchar states=0; //其实状态，既默认显示刷卡状态
```

```
uchar Card_type[2],          //卡片类型
```

```

Card_SN[4]={0},          //IC 卡号

Card_SN_BUF[4]={0xff,0xff,0xff,0xff},  //卡号登记缓冲区 1
NewCard1[4]={0xff,0xff,0xff,0xff},    //卡号登记缓冲区 2
NewCard2[4]={0xff,0xff,0xff,0xff},    //卡号登记缓冲区 3

KEY_BUF[8]='8','8','8','8','8','8','8','8'; //默认状态密
码为 8 个 8: :“8888 8888”
uchar ii=0;

void uart_over( void ) //串口数据还原
{
    UartCount = 0;
    UartStart = FALSE;
    UartComp = FALSE;
}

uchar IC_READ( void )      //读卡程序
{
    uchar ID_ASC[8],i;
    if( PcdRequest( PICC_REQIDL, Card_type ) != MI_OK )//寻天线区内
未进入休眠状态的卡，返回卡片类型 2 字节
    {
        if( PcdRequest( PICC_REQIDL, Card_type ) != MI_OK )//寻天线
区内未进入休眠状态的卡，返回卡片类型 2 字节
        {

            return FALSE;

        }
    }

    if( PcdAnticoll( Card_SN ) != MI_OK ) //防冲撞，返回卡的序列号 4
字节
    {

```

```

        return FALSE;
    }

    send_bytes(Card_type, 2); //发送卡类型
    send_bytes(Card_SN, 4);   //发送卡号
    /*****
    *****/
    将 IC 卡的物理地址转换为显示在液晶屏上的 ASCII 值 卡
    号

    *****/
    *****/
    for(i=0;i<4;i++)          //卡 ID 号转化成 ASCII
    {
        if(Card_SN[i]/16>9)    ID_ASC[i*2]=Card_SN[i]/16+'7';
        else ID_ASC[i*2]=Card_SN[i]/16+'0';

        if(Card_SN[i]%16>9)    ID_ASC[i*2+1]=Card_SN[i]%16+'7';
        else ID_ASC[i*2+1]=Card_SN[i]%16+'0';
    }

    lcd_mesg("ID:", 0x88, 4);          //显示 ID:
    display_card_num(0x88+0x02, ID_ASC, 8); //显示: 当前刷卡
    的卡号

    return TRUE;
}

void ctrl_process( void )              //RFID-RC522 刷卡处理函数
{

    uchar i, key_count, key_value=16, table[8]="-----", statesbuf;
    uchar count_card, count_code;      //刷卡错误次数计数

```

```

if(states!=statesbuf)          //状态改变，既有按键选择按下，则清
屏
{

    display_clear(1);
    display_clear(2);
    display_clear(3);
    statesbuf=states;          //根据按键选择的不同，显示下面功能
}
switch(states)
{

/*****
*****
*****          刷卡开门          显示车牌号
*****
*****/

    case 0:          //IC 卡读卡输入（默认状态）
        lcd_mesg("**RFID 门禁系统**", 0x80, 16);
        lcd_mesg("1. 请刷卡:          ", 0x90, 16);
        if(IC_READ())          //读卡程序
        {
            if
                //判断卡号和存储的卡号是否相同    读取的卡号和
EEPROM 里的数值比较
                (
(Card_SN[0]==Card_SN_BUF[0])&&(Card_SN[1]==Card_SN_BUF[1])
                &&(Card_SN[2]==Card_SN_BUF[2])
&&(Card_SN[3]==Card_SN_BUF[3])
                )
            {

```



```

        card_correct1(); //显示刷卡成功  管理员
        count_card=0;
    }
    else if

(

(Card_SN[0]==NewCard1[0])&&(Card_SN[1]==NewCard1[1])
    &&(Card_SN[2]==NewCard1[2])
&&(Card_SN[3]==NewCard1[3])
)
    {
        card_correct2(); //显示刷卡成功  用户 1
        count_card=0;
    }

else if

(

(Card_SN[0]==NewCard2[0])&&(Card_SN[1]==NewCard2[1])
    &&(Card_SN[2]==NewCard2[2])
&&(Card_SN[3]==NewCard2[3])
)
    {
        card_correct3(); //显示刷卡成功  用户 2
        count_card=0;
    }

else
{
    if(count_card!=1    &&    count_card!=2    &&
count_card!=3 )

        {
            count_card=0;

```

```

    }
    count_card++;    //错误次数
    card_fault();    //显示刷卡失败
    display_num(0x96, (48+count_card));

    if(count_card>=3)
    {
        lcd_mesg("报警系统已启动!", 0x98, 16);
        beepss();    //防盗报警系统启动
        display_clear(3);
        count_card=0;
    }
} //break;
}

break;

/*****
*****

                密码输入

*****
*****/

case 1:
    lcd_mesg("**RFID 门禁系统**", 0x80, 16);
    lcd_mesg("2. 请输入密码:", 0x90, 13);    //第二种开门
方式，用密码开门
    lcd_mesg("    -----    ", 0x98, 16);
    key_count=0;    //按键次数
    while(1)
    {
        key_value=keyscan();    //进入下一菜单，任然可以
选择其他功能

        if(key_value==13)
        {

```

```

        states++;
        if(states==5)
            states=0;
        return;
    }
    if(key_value==12)
    {
        states--;
        if(states < 0)
            states=4;
        return;
    }
    if(key_value>=0 && key_value <=9)//有按键输入
    {

        table[key_count++]=key_value+'0';    //转换成
ASCII 码值，显示数字，存储起来

        /***** 有按键输入，显示星号，提示已输入的位数
        *****/
        if(key_count<=8)
            lcd_mesg("****",0x98+0x02,key_count);

    }
    if(key_count==8)
    {
        if( table[0]==KEY_BUF[0] &&
            table[1]==KEY_BUF[1] &&
            table[2]==KEY_BUF[2] &&
            table[3]==KEY_BUF[3] &&
            table[4]==KEY_BUF[4] &&
            table[5]==KEY_BUF[5] &&
            table[6]==KEY_BUF[6] &&

```

```

        table[7]==KEY_BUF[7] )           //密码正确

    {
//        beeps() ;
card_correct1() ;
        count_code=0;

        break;

    }
    else if
        ( table[0]!=KEY_BUF[0] ||
          table[1]!=KEY_BUF[1] ||
          table[2]!=KEY_BUF[2] ||
          table[3]!=KEY_BUF[3] ||
          table[4]!=KEY_BUF[4] ||
          table[5]!=KEY_BUF[5] ||
          table[6]!=KEY_BUF[6] ||
          table[7]!=KEY_BUF[7]
        )           //密码错误

    {
        if(count_code!=1   &&   count_code!=2   &&
count_code!=3 )

            {
                count_code=0;

            }

            count_code++;    //错误计数

            beeps() ;
            display_num(0x97, (48+count_code));
            lcd_mesg("密码输入错误!", 0x98, 14);
            delay(100);

```

```
display_clear(3);  
lcd_mesg("请重新输入!", 0x98, 12);  
delay(100);  
if(count_code>=3)  
{  
    lcd_mesg("报警系统已启动!", 0x98, 16);  
  
    beepss();    //防盗报警系统启动  
  
    display_clear(3);  
    count_code=0;  
  
}  
break;  
}  
  
}  
  
}break ;    //while 的 break
```