齐 齐 哈 尔 大 学

毕业设计（论文）

题 目 基于ARM的马路机械车库系统设计

学 院 通信与电子工程学院

专业班级 通信工程152班

学生姓名 张奥涵

指导教师 朱恒军

成 绩

2019年4月15日

# 摘 要

自此改革开放以来，我国的经济有了长足的发展，私家车也是越来越多，具保守估计，截止2018年底我国私家车已经达到了近2.5亿辆，伴随着私家车数量的飙升而出现的问题就是各个城市的停车难问题，因为城市新建停车场容纳的汽车数量远远跟不上私家车数量的增长，并且繁华区域寸土寸金，也无法分出大片的区域用来建停车场，所以传统意义上的停车场面对目前停车难的问题根本无能为力。

传统停车场最大的弊端在于空间利用率不足，并且计费手段过于原始，虽然目前停车场也发现了计费手段落后的问题，纷纷引入摄像头或者其他手段采取计时收费，但是空间利用率不足的问题根本无法解决，基于此，本系统以STM32单片机为核心设计一款空间利用率高并且可以计时收费的停车场系统。

本系统核心传感器为光学指纹识别模块，用该模块采集用户信息，本系统使用时，先是扫描指纹，如果用户之前已经将自己的指纹信息录入，那么用户可以自己进行停车和取车的操作，如果用户没有录过指纹信息，那么需要管理员来帮助其录入指纹信息，然后改用户才能正常使用本系统，另外用户停完车后开始计时收费，用户取车时只要点击TFT屏上的“停止使用本停车场”按键就可以进入取车操作了。并且，管理员登录时，可以自动识别管理员的指纹信息，然后直接进入管理员界面，管理员界面可以查看停车场的使用情况、帮新用户录指纹以及查看停车场的环境信息。

本系统在满足传统停车场的基本需求外，并且还更加自动化，大大减少了人力资源的浪费，并且随着停车行业的快速发展，这种能高效利用空间的机械车库必将成为停车场的未来主要发展方向。

随着人类社会的不断进步和科学技术的发展，生产、生活方式趋于集中，城市的规模越来越大，人们在城市里的生存空间却越来越小，于是出现了要利用空间的理念，城市中开始建设立体建筑、立体交通和立体停车。

**关键字**：停车场；STM32；停车产业；指纹识别；TFT屏

# **Abstract**

Since the reform and opening up, China's economy has made great progress, and there are more and more private cars. According to conservative estimates, the number of private cars in China has reached nearly 250 million by the end of 2018. The problem accompanying the soaring number of private cars is the problem of parking difficulties in various cities. Because the number of cars in newly built parking lots in cities is far from keeping up with the growth of the number of private cars, and the prosperous areas cannot be divided into large areas to build parking lots, the traditional parking lots can do nothing to deal with the current parking difficulties.

The biggest disadvantage of the traditional parking lot is that the space utilization rate is insufficient and the charging method is too primitive. Although the parking lot has also found the problem of backward charging method, cameras or other means are introduced one after another to adopt timing charging, but the problem of insufficient space utilization rate cannot be solved at all. Based on this, this system designs a parking lot system with high space utilization rate and timing charging with STM32 microcontroller as the core.

The core sensor of the system is an optical fingerprint identification module, which is used to collect user information. When the system is used, the fingerprint is scanned first. If the user has previously entered his fingerprint information, the user can park and pick up the car himself. If the user has not recorded fingerprint information, the administrator is required to help him to enter the fingerprint information, and then the user can normally use the system. In addition, the user starts timing and charging after stopping the car. When picking up the car, the user can enter the car picking up operation by clicking the "Stop using the parking lot" button on the TFT screen. In addition, the administrator can automatically identify the fingerprint information of the administrator when logging in, and then directly enter the administrator interface, which can view the usage of the parking lot, record fingerprints for new users and view the environmental information of the parking lot.

This system not only meets the basic needs of traditional parking lots, but also is more automated, greatly reducing the waste of human resources. With the rapid development of the parking industry, this kind of mechanical garage that can efficiently use space will become the main development direction of parking lots in the future.

With the continuous progress of human society and the development of science and technology, the production and life style tend to be centralized, the city scale is getting larger and larger, but people's living space in the city is getting smaller and smaller, so the concept of using space has emerged, and three-dimensional buildings, three-dimensional traffic and three-dimensional parking have begun to be built in the city.

**Keywords** **Urban Scale：**Parking lot;STM32;Parking industry; Fingerprint identification; TFT screen

目 录

[摘 要 I](#_Toc10663067)

[**Abstract** II](#_Toc10663068)

[第1章 绪 论 2](#_Toc10663069)

[1.1 课题研究背景 2](#_Toc10663070)

[1.2 国内外研究现状 3](#_Toc10663071)

[1.2.1 国外研究现状 3](#_Toc10663072)

[1.2.2 国内研究现状 4](#_Toc10663073)

[1.3 课题研究内容 5](#_Toc10663074)

[1.4 文本结构 5](#_Toc10663075)

[第2章 总体设计方案 6](#_Toc10663076)

[2.1 总体结构设计 6](#_Toc10663077)

[2.2 系统硬件选择 6](#_Toc10663078)

[2.2.1 主控选择 6](#_Toc10663079)

[2.2.2 显示屏选择 8](#_Toc10663080)

[2.2.3 环境监测传感器的选择 9](#_Toc10663081)

[2.2.4 指纹模块选择 10](#_Toc10663082)

[2.2.5 机械结构方案设计 12](#_Toc10663083)

[2.3 指纹识别算法 13](#_Toc10663084)

[2.4 本章小结 14](#_Toc10663085)

[第3章 系统硬件设计 15](#_Toc10663086)

[3.1 STM32F103ZET6最小系统电路设计 15](#_Toc10663087)

[3.1.1 STM32F103ZET6的晶振电路设计 15](#_Toc10663088)

[3.1.2 STM32F103ZET6的复位电路设计 16](#_Toc10663089)

[3.2 供电电路设计 16](#_Toc10663090)

[3.3 显示电路设计 18](#_Toc10663091)

[3.4 停车场环境监控电路设计 20](#_Toc10663092)

[3.4.1 停车场环境温湿度监测电路设计 20](#_Toc10663093)

[3.4.2 停车场环境光照强度监控电路设计 20](#_Toc10663094)

[3.5 电机驱动电路设计 22](#_Toc10663095)

[3.6 指纹模块电路设计 23](#_Toc10663096)

[第4章 系统软件设计 25](#_Toc10663097)

[4.1 主控程序设计 25](#_Toc10663098)

[4.1.1 主控程序设计流程图 25](#_Toc10663099)

[4.1.2 部分主控代码以及介绍 26](#_Toc10663100)

[4.2 传感器程序设计 29](#_Toc10663101)

[4.2.1 传感器程序设计流程图 29](#_Toc10663102)

[4.2.2 传感器部分代码及介绍 30](#_Toc10663103)

[4.3 电机及驱动程序设计 33](#_Toc10663104)

[4.3.1 电机及驱动控制流程图 34](#_Toc10663105)

[4.3.2 电机及驱动控制部分代码及介绍 36](#_Toc10663106)

[4.4 显示屏程序设计 37](#_Toc10663107)

[4.4.1 显示屏控制流程图 38](#_Toc10663108)

[4.4.2 人机交互界面部分代码及介绍 39](#_Toc10663109)

[4.5 指纹模块程序设计 39](#_Toc10663110)

[4.5.1 指纹模块控制流程图 40](#_Toc10663111)

[4.5.2 指纹模块控制部分代码及介绍 41](#_Toc10663112)

[第5章 系统测试 43](#_Toc10663113)

[5.1 已录指纹的用户自主操作系统测试 43](#_Toc10663114)

[5.1.1 系统开机测试 43](#_Toc10663115)

[5.1.2 用户自主停车操作测试 45](#_Toc10663116)

[5.1.3 用户查看信息以及取车操作测试 48](#_Toc10663117)

[5.2 管理员操作系统测试 49](#_Toc10663118)

[5.2.1 管理员基本信息查看系统测试 49](#_Toc10663119)

[5.2.2 用户信息管理操作系统测试 50](#_Toc10663120)

[5.3 自动照明灯系统测试 51](#_Toc10663121)

[结 论 52](#_Toc10663122)

[参考文献 53](#_Toc10663123)

[致 谢 55](#_Toc10663124)

[附录 1 指纹模块主要程序 56](#_Toc10663125)

[附录 2 主控程序 77](#_Toc10663126)

# 第1章 绪 论

## 1.1 课题研究背景

中国第一辆私家车可以追溯到清末，新中国改革开放以后，中国的私家车随着我国经济的快速增长也在已更加迅猛的速度增长，在我国国民的小康愿景中私家车占据了不可或缺的地位，拥有私家车成为了进入小康俱乐部的邀请函，但是从2010年开始，停车难和城市道路拥堵的问题也凸显出来，而单纯的增加传统停车场并不能很好的解决停车难的问题，所以越来越多关于停车场建设的新方案就被提了出来，其中中小型的机械车库就是其中的一种主流方案。

机械车库最先是美国工程师最先提出并最先建设的，机械车库又被称为立体车库，最大的优点就是空间利用率高，因为虽然城市越来越大，但是闲置地皮却越来越小，特别是城市的市中心和繁华地段，寸土寸金，车辆流动极大，但是却不能在繁华地段大规模的修建停车场，而非固定式的、空间利用率高并且占地面积不大的机械车库就可以很好的解决这个问题，机械车库可以根据需要在任何地方修建，并且结构一般为组合式，即机械车库的层数和大小可以根据实际情况进行组合，当不需要时或者其它情况需要拆除时，仅需1到2天就可以分块拆走，非常方便。所以，机械车库在未来必将成为停车行业中不可或缺的一部分。

另外机械车库作为一种机械设备，其相比传统的地下停车场或地面停车场有一些独有的优势，比如，地下停车场在建设时就必须考虑停车场的通风问题以及采暖问题，另外下水道的设计也要根据不同城市的情况具体设计，成本极高，但是对于机械车库来说这些问题就不存在，机械车库的主体采用铁架构成，本身的通风性就极好，所以就单单这个方面，机械车库的耗能就比传统的底下停车场要低得多。另外机械车库相比较地上停车场也具有一些独有的优势，就比如汽车安全问题，传统的地上停车场一般都有专门的保安来保护车主的汽车安全，防止小偷在车主不在时对私家车内的财务进行盗窃，而机械车库本身的高度就足够高，可以很好的避免这个问题。另外无论是底下停车场还是地上停车场，为了尽可能的增加停车位，往往车道仅能通过一辆汽车，这样当停车和取车就无法同时进行，并且因为车位之间的距离不够大，一些不太熟练的驾驶员很容易造成汽车的剐蹭等一些安全问题，给人们的日常生活造成不必要的麻烦，最重要的是，传统停车场的汽车停放往往没有规律，这样车主在停车场外面很难得知该停车场还有多少个停车位，所以当车流高峰期，一个私家车主想停个车既需要很好的驾驶技术更需要很好的眼神，无形当中增加了驾驶员的压力，但是这些问题对于机械车库来说，也是根本不存在，机械车库往往都有专门的显示设备，驾驶员远远的就可以得知机械车库内还有没有空余的停车位，另外停车和取车的一大部分过程都由机械设备完成，这样可以很好地减轻对驾驶员技术的考量，最重要的是，机械车库做的了完全的人车分流，不会再机械车库附近造成大范围的堵车。此外机械车库作为一个机械设备，还可以对其进行很好地装饰，在可以正常行使其停车场功能的同时，还可以作为城市的装饰物存在于城市的各个角落，所以其美观性和实用性都是传统停车位无法比拟的。

另外因为机械车库本身的特性，在城市规划中，可以大规模的在特定区域建设，以大量增加停车位，还可以见缝插针式的建设，在城市繁华区域的角落里建设，以缓解停车压力，还可以和传统停车场进行搭配使用，进行停车效率的最大化，所以机械车库无论在功能上还是发展趋势上，都会成为未来城市停车场建设过程中的一个不可替代的选择，所以对机械车库设计的研究很有必要同时也很符合市场的需求。

## 1.2 国内外研究现状

本小节主要介绍我国自身以及西方发达国家关于机械车库方面的研究过程和目前最新的理论，详细介绍机械车库被提出的目的、产品的特性以及未来的发展趋势。

### 1.2.1 国外研究现状

机械车库最早叫做自动立体停车场[2]，机械车库的概念最早由美国工程师提出，由德国最先大规模应用，目前德国的机械车库技术最好，德国目前主流的机械车库一般为3层以上，结构为H型，也就是机械车库在横跨一个车道，该车道为停车以及取车的车道，并且该车道就是机械结构的底板，可以将汽车抬高和放下，用户要停车时直接驶入该车道，然后进行停车操作，操作完成后，底板将汽车抬高然后送人指定停车位。除了H型外，主流的机械车库结构还有U型和V型。

再比如日本，日本是世界上人口密度最大的国家，并且日本本身经济比较发达，私家车普及率和数量都很高，但是日本国土面积小，在城市规划中根本不可能大量建造传统停车场，所以日本从1959年就开始引进德国的设备，并积极的进行自主研究和设计，并于1983年开始在日本本土大规模建设机械车库，到了2000年，日本关于机械车库方面的研究以及很成熟了，并且实践活动也很多，据统计，截止到2005年，日本全国一共建造了近50000座不同型号的机械车库，这些车库大大缓解了日本国内因国土面积不足而引起的停车难问题，最重要的是，这些机械车库大部分体积都不大，可以建设在城市的各个角落，比如酒店附近的空地上，再比如公园里的空地上，既不影响这些区域的基本人员流动又能很好的为城市提供大量的停车位。

到了2010年，随着无线通信技术的发展，英国的部分机械车库开始进行设备升级，在原有机械车库的基础上增加了无线数据传输功能，当某一部分的机械车库出现问题后，系统自检无法解决时，可以立即通知工程师，在用户还未拨打服务电话的时候就已经有工程师前往了，另外该项技术还可以将城市各个机械车库的使用情况发给城市的政府，让政府知道城市的那部分区域对停车位的需求大，从而施行相应的措施来使城市的运行更加流畅。

到了2016年，随着材料学的突破，建材公司开始向市场提供一种强度大质量轻的建材结构，随后美国的机械车库建设就采用了该类型的建材，新式的机械车库重量更轻，建造更加方便，随之也可以建造更多的层数，可以大幅提高车库容量，相信随之科技的发展，未来的机械车库技术会更加成熟，机械车库未来必将为城市建设提供更大的便利。

### 1.2.2 国内研究现状

我国关于机械车库方面的研究起步较晚，并且城市的停车场规划管理技术也没有那么成熟，所以现在基本上所有的城市都面临着停车难的问题，所以机械车库在我国有着很好的研究和应用前景。

我国的机械车库最早出现于上世纪80年代，最早出现在香港，但是由于当时只是引进国外的先进设备，设备的维护和管理手段没有相应的跟进，所以直到2000年，我国机械车库在解决城市交通问题的优势还没有被最大化的体现出来，随着我国地府政府相关管理水平的提升，机械车库才慢慢在大城市开始普及，到了2016年就单单北京的机械车库就可以提供近3000个停车位。

并且从2010年开始，我国的机械车库就开始了自主化的研究，到了2011年，我国自主设计的机械车库实现了无人化的管理，仅仅考一些标识，就可以使用户轻松的使用设备进行停车和取车操作，大大减少了人力资源的浪费。

到了2015年，随着云计算方面的研究趋于成熟，我国的机械车库的收费方式通过云计算来进行设置，先是通过大数据对城市各个区域对停车位的需求进行预计算，然后根据需求的高低进行费用的设定，就比如上海陆家嘴区域的机械车库的费率就比上海闵兴区的机械车库费率高上不少，这不仅仅是因为陆家嘴区域的机械车库建设成本比较高，而且还与该区域对车位的需求较高有很大的关系。

到了2018年，我国关于下一代机械车库的概念主要基于物联网，将机械车库的车位当成一种商品放到互联网的平台上，用户可以根据自己的行程规划提前数天预定某机械车库的一个车位，并且费用的收取也与时俱进采用微信或者支付宝，此外为了打击黄牛党和增加车位的流动性，当用户长时间占据一个停车位时，将会大幅提高该车位的费率。

另外随着我国AI技术的快速发展，机械车库的管理和运行会越来越智能化，未来我国的机械车库无论在数量上还是技术上都会有长足的发展，机械车库在我国的城市规划中肯定会占据越来越重要的地位。

## 1.3 课题研究内容

本系统基于停车场的作用和传统停车场的弊端，设计一款机械车库系统，该款机械车库主要以单片机组成的嵌入式系统为核心，采用光学指纹识别模块[1]作为登录识别的主要手段，并结合电机、舵机、TFT屏以及一些传感器来模拟完成机械车库的主要运行流程。

本系统主要技术为指纹识别算法，利用图像处理识别出管理员和用户，然后通过TFT屏的触摸功能来进行相应的操作，其中用户可以使用本系统进行停车操作、查看信息操作以及取车操作，而管理员可以使用本系统采集新用户的指纹信息、查看停车场的基本情况以及查看停车场环境信息等。

## 1.4 文本结构

本论文共分为6章，第一章主要介绍机械车库的基本概念、国内外研究现状以及未来发展的前景和趋势。第二章主要介绍系统的总体设计方案，并详细介绍了各个传感器模块的主要工作参数数据以及指纹模块的图像处理算法。第三章主要为硬件电路的设计，详细介绍了各个部分的电路设计以及各个部分的电路原理图等。第四章主要介绍系统的程序设计流程图，并给出了部分代码来详细阐述。第五章为系统测试，主要介绍系统的主要测试功能以及运行结果和分析。第六章为结论、个人总结以及参考文献等等。

# 第2章 总体设计方案

## 2.1 总体结构设计

本系统以STM32F103ZET6单片机为核心，以AS608光学指纹模块为主要传感器，来设计机械车库，使用电机、丝杆和电机驱动来设计机械车库升降平台，使用指纹模块的指纹识别算法来设计用户识别功能，使用TFT\_LCD来设计人机交互界面，使用STM32内部的定时器和其他资源来设计计费功能，使用触碰按键传感器来设计用户操作功能设计，使用DHT11温湿度检测传感器、光照强度检测传感器以及多色彩显示LED灯来设计机械车库的环境监测以及其他功能设计。

本系统的基本设计框架如下图：

主

控

指纹识别系统

传感器检测系统

机械结构系统

TFT屏显示系统

登录管理

环境检测

人机交互

机械控制

图2-1 基于ARM的机械车库系统设计基本框架

## 2.2 系统硬件选择

### 2.2.1 主控选择

本系统采用的单片机为STM32F103ZET6，该单片机为意法半导体公司生产的一款主控芯片，该单片机相比较市面上其他流行的单片机具有很高的性价比，首先，STM32单片机的机器字长为32位，也就是STM32可以同时接受、处理以及输出长达32的数据，所以STM32单片机的运算效率很高。其次，STM32单片机的内核为ARM，处理数据的能力也非常出色，并且STM32的闪存和SRAM分别为256KB和64KB，内存也有64KB，非常适合嵌入式的学习和开发。另外，STM32的内置资源相当丰富，用于多达8个定时器以及多种通信接口，比如STM32就配置有3路硬件SPI通信接口、21路ADC接口以及2路DAC接口，使用STM32进行嵌入式系统开发时，不需要再设计专门的AD转换电路和DA转换电路，大大简化了系统硬件设计的复杂性。

STM32F103ZET6单片机的基本参数和基本内置资源接口如下表所示：

|  |  |
| --- | --- |
| 名称 | 功能 |
| 芯片型号 | STM32ZET6 |
| 主频 | 72MHz |
| 硬件SPI接口 | 3个 |
| 工作电压 | 2.7V-3.6V |
| 定时器 | 8个 |
| SRAM | 64KB |
| 内部flash | 512KB |
| 硬件12位ADC | 21路 |
| 硬件12路DAC  硬件IIC接口  通用IO口 | 2路  2个  114个 |

表2-2 STM32基本工作参数表

STM32除了拥有丰富的内置资源外，其外部高频晶振为8MHz，系统使用时，可以通过PLL倍频到72MHz，这样是STM32的程序运行速率远快与其他单片机，可以处理相当复杂的逻辑运算。并且STM32的功耗也很低，给其3.3V的电就可以使其正常工作，最重要的还是价格也不贵，性价比极高，非常适合作为嵌入式系统的主控芯片。

此外STM32的引脚资源也极其丰富，就比如STM32F103ZET6单片机的引脚高达144个，在设计开发板时，出去必要的电源引脚和一些无法正常使用的IO口，剩下被引出来的引脚也高达114个，并且这144个引脚，每个引脚都可以根据实际被配置为8种工作模式，来满足一些特殊功能的需求，比如：如果使用PA1引脚作为AD输入引脚，首先需要将PA1引脚配置为模拟输入模式，然后再开启ADC以及配置ADC的相关寄存器，将ADC通道映射到PA1引脚上，这样就可以将输出模拟值的传感器直接连到STM32单片机的PA1引脚上。另外，被引出来的114个引脚每个引脚都可以作为外部中断输入引脚，这是因为STM32单片机在设计的时候中断线是悬空的，在使用本功能时，首先需要先通过配置相关寄存器，将外部中断与设定的引脚线连接到一起，然后对该外部中断线进行相应的触发配置即可使用。

本系统使用的开发板实物图如下图：

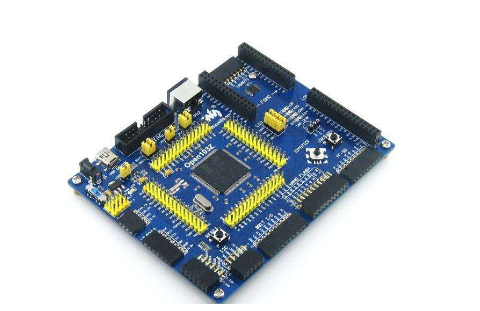


图2-3 本系统采用的STM32ZET6最小系统开发板

### 2.2.2 显示屏选择

本系统非常重要的一个功能就是人机交互的设计，既然是人机交互，那么必然伴随着信息的显示和用户的个人选择，所以本系统的显示屏需要集显示和触碰输入功能于一身，而TFT屏就是一个很好的选项，TFT屏属于LCD显示屏家族，并且是LCD显示屏中的最高端的产品之一，其比一般的LCD屏在功能上更强，显示上也具有独特的优势，并且其具体触摸输入功能，很好的满足了本系统的需求，降低了本系统的硬件设计复杂度，并且TFT\_LCD比Oled屏更加便宜，虽然Oled屏在性能和显示色彩方面比TFT屏更加强大，但是其价格也比TFT屏贵得多，相比之下，TFT屏的性价比更高。

本系统采用的是2.8寸的TFT屏，该屏幕的分辨率为240×320，显示的信息数量上也足够多。并且可以显示16位的真彩，也就是说，TFT屏上的每个像素点可以显示65536种颜色，所以其在色彩显示的还原度和丰富度上，已经达到了很高的标准。另外，本系统采用的2.8寸的TFT屏也是电阻式的触摸屏，也就是说TFT屏的每一个像素都可以作为一个输入的按键，虽然在实际操作中，我们不可能会用到那么小的按键，但是我们可以预设TFT屏中的一块作为一个按键接口，只要该区域内有足够的的像素点检测到手指按下就可以认为该区域被点击，然后就可以进行相应的控制函数，比如可以开启相关控制电路或者进入其他的显示界面，所以就显示屏的功能而言，TFT屏也能很好的满足本系统的需求。

2.8寸TFT显示屏实物图如下图所示：



图2-5 2.8寸TFT显示屏实物图

### 2.2.3 环境监测传感器的选择

根据系统预设的功能，环境监测的主要参数有：

1. 停车场温湿度监测
2. 停车场光强监测

1：停车场温湿度监测

本系统对机械车库附近环境的温湿度监测主要采用DHT11传感器[4]，该传感器能够测量温度数据和湿度数据是因为DHT11内部集成了相应的感温元件和感湿元件，将DHT11放入环境中，通完电后，感温元件和感湿元件将会根据环境的温度和湿度改变回路的电流大小，然后DHT11将该电流数据转化为一个4位的数据传送给单片机，这4位数据前两位为温度数据，后两位为湿度数据，所以单片机在得到DHT11传回的数据后，需要先将该数据进行拆分，分别得到温度数据和湿度数据。

并且DHT11还具有自动校准的功能，内部设有校准电路，每次使用时先通过单片机发送信息使其进行自动校准，校准之后单片机才读取DHT11的数据信息，这样可以使得到的数据信息更加稳定和准确。此外，DHT11还拥有很大的温度量程和湿度量程，其温度量程为0摄氏度到50摄氏度，湿度量程为20%RH到90%RH，并且温湿度数据的精度也很高，比如其温度精度为±2摄氏度，湿度精度为±5%RH，本系统对环境温湿度的量程要求和精度要求都没那么高，所以采用DHT11就可以很好地满足本系统的要求。

其实物图如下图所示：



图2-6 DHT11温湿度传感器实物图

2：停车场光强监测

本系统之所以加入光强检测的功能，是因为机械车库一般为全天24小时自动运行，而有些地方的夜晚光线不足，可能对用户在夜间进行停车及取车操作造成一定的困扰，所以本系统加入光线检测功能，当晚上光线不足时，打开照明灯，方便用户使用，也可以当成一个路灯来使用。

本系统对光照强度检测主要使用的传感器为光敏电阻模块，光敏电阻的材料特性也很特殊，即光敏电阻的阻值会根据照射到其检测面上的光照强度的变化而变化，在实际运用中，将光敏电阻嵌入到电路里，然后检测该电路中的回路电流即可，因为在电压稳定的前提下，其阻值的改变会引起回路电流的变化，另外本模块内部还集成有数据比较器，使用非常方便。

产品主要参数如下：

1. 工作电压：3.3V—5V DC
2. 型号：5516

我们在使用本传感器模块进行系统的设计时，可以直接将其模拟输出接口连接至单片机的ADC输入端，单片机可以直接获得实时的光照强度数据，另外当我们不需要那么精确的光照强度数值而只需要将其作为一个光照强度检测开关时，我们也可以使用其数字量输出引脚，通过调节该模块上的电位器，可以达到改变灵敏度的效果。

光敏电阻模块实物图如下图所示：

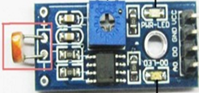


图2-7 光敏传感器实物图

### 2.2.4 指纹模块选择

本系统的重要功能就是用户登录管理和识别，而这部分功能我们主要通过AS608指纹模块来完成，该指纹模块采用SIFT算法来进行指纹图像识别，当用户将手中放到感应区域时，传感器先监测到有指纹按下，然后采集指纹图像信息，并与其内部存储器中的指纹信息库进行对比，并根据对比结果返回相应的信息。

另外该模块的晶振为12MHz，所以其有很高的运算速度，可以在很短的时间内采集足够多的图像信息，这样数据将会更加精确。该模块与STM32单片机之间的通信方式为串口通信，另外还有一个触碰感应输入端，占用主控的资源也不多，并且其内部的存储器大小为72KB，可以储存多达300个指纹信息。

AS608指纹模块的实物图如下：



图2-8 AS608指纹模块实物图

AS608的基本工作参数如下表所示：

|  |  |
| --- | --- |
| 参数 | 说明 |
| 工作电压 | 3.3V DC |
| 工作电流 | 40mA |
| 识别图像分辨率 | 256×288 |
| 图像处理时间 | 0.4 S |
| 上电延迟时间 | 0.1 S |
| 搜索时间 | 0.3 S |
| 拒真率（FRR） | 1% |
| 认假率（FAR） | 0.001% |
| 指纹信息容量 | 300 |
| 工作环境温度 | -20摄氏度-60摄氏度 |
| 工作环境湿度 | 0%RH-90%RH |

表2-9 AS608指纹模块基本工作参数

### 2.2.5 机械结构方案设计

本系统的机械结构主要是为升降平台，目的是将目标车辆运送到指导层数，所以机械结构主要由：舵机、电机、触碰按键以及电机驱动组成。

（1）其中舵机选用SG90模拟舵机，来实现门杆的智能开启和关闭功能。

其中模拟舵机的主要工作参数如下表：

|  |  |
| --- | --- |
| 参数 | 指标 |
| 工作电压 | 5V DC |
| 工作电流 | 3mA |
| 转动角度 | 180° |
| 工作环境温度 | -30℃-60℃ |
| 最大转速 | 0.5秒/60° |
| 扭矩 | 1.6Kg/cm |
| 响应延迟时间 | 5us |

表2-10 舵机基本工作参数

（2）电机采用直流电机，驱动采用TB6612FNG。

具体结构中，采用联轴器将直流电机与丝杆连接起来，然后通过控制直流电机的转动来使升降台到达指定高度，其中高度的检测通过碰撞开关来实现，碰撞开关的感应区在没有东西挤压时，该模块输出高电平，当升降台升高的指定层数时会挤压碰撞开关，此时，碰撞开关输出低电平，以此来检测升降台的具体位置并进行相应的控制。

直流电机需要的电流较大，无法使用单片机直接进行控制，一般都需要驱动，本系统控制电机采用的驱动为TB6612，该驱动的输入电压为12V直流，其最大输出电流为1.2A，足以满足电机的最大转速工作所需的电流，控制时，可以直接通过STM32单片机的任一定时器的输出通道，输出PWM波即可控制直流电机的转速。

TB6612实物图如下：



图2-11 TB6612的实物图

## 2.3 指纹识别算法

本系统采用的指纹模块采集图像的方法主要为光学信息采集，采用这种方法得到的指纹图像信息更加准确，图像也更加清晰。其中指纹识别的主要步骤一般为三步，分别为：指纹图像采集及预处理、指纹图像特征值的读取以及根据指纹图像特征值与指纹库进行比较。

（1）指纹图像采集及预处理

图像的预处理是为了更好地提前图像中的特征值，所以需要先对图像进行一些基本的数字图像处理操作，比如在指纹识别算法中对采集到的图像一般先进行图像的分割，该步主要是为了简化图像，将指纹图像与背景色分割开来，以免背景色对提前特征值产生影响，然后就说图像的滤波，该步骤主要是为了进行降噪，也是预处理过程中最重要的一步，降噪处理的效果直接影响了能否从该图像中正确的提取相关特征值，最后一步便是图像的二值化，这是为了使图像特征值更加明显，方便提取。

其中，在对指纹图像进行通信分割步骤时利用的是归一化处理，具体公式如下：

其中和为期望的均值和方差，和为采集到的指纹图像的均值和方差。

另外在对指纹图像进行二值化处理时，我们采用的思想是阈值二值化，具体公式如下：

其中T为程序里设定的阈值，x为灰度值。

（2）指纹图像的特征值提取主要是从我们上一步二值化后的图像中进行提取特征值，也就是从指纹图像中的一些特殊点出发，进行数据的读取，比如提取某指纹图像的端点个数和分叉点个数等信息。

具体算法流程如下：

先是从指纹图像的端点出发，如果某一个端点的附近区域内有一个有效点，那么就认为改点为可跟踪的下一个点，然后通过对脊线中间点进行分析，如果中间的附近有一个黑点，那么就可以忘掉上一个被跟踪的点，设定下一个被跟踪的点为待跟踪点，最后通过对有效点坐标数据进行比较和运算，来得出一些特征值。具体公式如下：

其中cn表示该像素点的灰度值，sn为跟踪点的步长。

（3）指纹图像的对比是指纹识别的最后一步，主要运用的是指纹匹配算法，因为在使用指纹模块的时候，每次手指放的位置和方向都会有所不同，所以需要用指纹匹配算法来进行快速准确的判断，指纹匹配算法的好坏直接决定了指纹识别的响应速度以及准确性。

## 2.4 本章小结

本章的主要工作是介绍本系统的总体结构、各个模块的选择和各个模块的基本工作参数，另外本章还介绍了机械结构的设计方案，具体的机械结构工作流程图将会在第三章详细讲解，另外因为本系统使用的指纹模块涉及指纹图像识别技术，所以在本章最后还简单的介绍了一下指纹模块使用的指纹识别算法。

# 第3章 系统硬件设计

## 3.1 STM32F103ZET6最小系统电路设计

任何一个由单片机组成的嵌入式系统，核心都是单片机的最小系统，而无论是什么单片机，其最小系统都是由晶振电路以及复位电路组成，下面我们将分2个小节，分别介绍单片机的晶振电路和复位电路，详细的阐述各部分的作用、原理以及具体实现过程，并给出相应的电路原理图。

### 3.1.1 STM32F103ZET6的晶振电路设计

晶振电路对应单片机而言就相当于心脏对应我们人而言，这是因为单片机工作是按照时钟周期一步一步运行的，而单片机的时钟周期就是由晶振电路产生的，所以没有晶振电路，单片机无法工作，并且晶振的频率还决定了单片机的运行速度。

STM32的晶振电路主要分为2块，一块是外部高频晶振，这一块主要是用于正常的单片机数据处理用的，一般采用8MHz的晶振，经过PLL倍频处理后，可以达到72MHz，那么，单片机正常工作时，其时钟周期就为，使用STM32的运算速度很快，可以处理相当复杂的逻辑运算。STM32单片机的另一块晶振为外部低频晶振，这一块晶振的频率一般采用36KHz，主要在STM32处于待机或者低功耗状态下发挥作用。

其中STM32的外部高频晶振电路原理图如下：

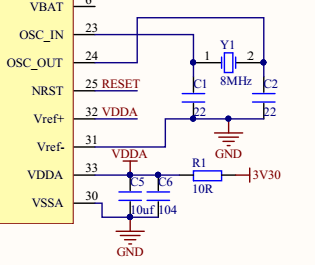


图3-1 STM32的外部高频晶振原理图

其中STM32的外部低频晶振电路原理图如下：

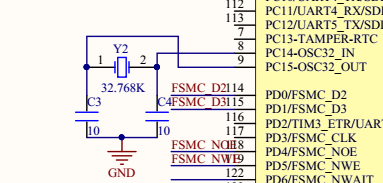


图3-2 STM32的外部低频晶振原理图

### 3.1.2 STM32F103ZET6的复位电路设计

每个单片机的最小系统都离不开晶振电路，同样的，作为和晶振电路同样主要的还有复位电路，复位电路顾名思义就是使系统进行复位，程序从头再来，一般复位电路都是由一个电解电容和一个复位开关组成，主要原理为，每个单片机都有一个复位引脚，当连续给该引脚3个以上时钟周期的低电平或者高电平时，将会触发复位操作，即将STM32几乎所有的寄存器都清零，程序从头再来。

直接对单片机系统断电也能达到复位的效果，那为什么还要再设计复位电路呢，这是因为对单片机断电和上电操作会在一定程度上损害单片机，一般STM32的可上电次数为10万次，而使用复位按键进行复位操作，则大大减小了对单片机的损伤。

STM32复位电路原理图：

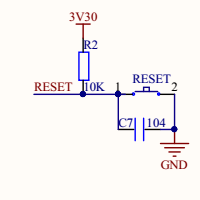


图3-3 STM32的复位电路原理图

## 3.2 供电电路设计

每个系统都需要根据实际需要设计稳定供电电路，本系统中TB6612需要12V的电压，传感器需要5V的电压，主控板需要3.3V的电压，使用本系统需要输入12V的电压，然后将12V的电压使用降压模块分别降到5V和3.3V供系统使用，其中采用LM2596降压模块来将12V的直流电降到5V，然后再使用LM1117-3.3降压模块将5V的电降到3.3V。

其中LM2596降压模块的基本工作参数如下：

|  |  |
| --- | --- |
| 参数 | 指标 |
| 输入电压 | 3.2V-40V |
| 输出电压 | 1.25V-35V |
| 输出电流 | <3A |
| 转换效率 | <92% |
| 输出纹波 | <30Ma |
| 频率 | 65KHz |
| 工作环境温度 | -45℃-85℃ |

表3-4 LM2596基本工作参数

需要注意的是LM2596在使用时其输出电压不能大于其输入电压。

LM2596电路原理图如下：

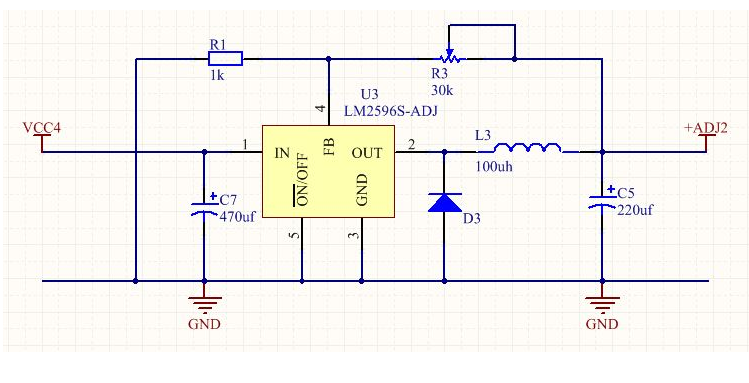


图3-5 LM2596原理图

LM1117电路原理图如下：

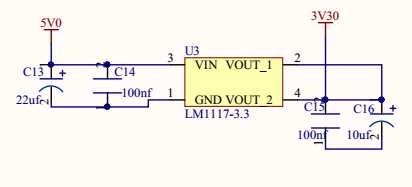


图3-6 LM1117原理图

## 3.3 显示电路设计

本系统用于实现人机交互功能的主要器件为TFT屏，本系统采用的TFT显示屏尺寸为2.8寸，分辨率为320×240，另外TFT屏自带触摸屏，可以用来作为输入控制。

其中TFT屏主要特性如下所示：

1. 2.4’ /2.8’ /3.5’ /4.3’ /7’ 5 种大小的屏幕可选。
2. 320×240 的分辨率（3.5’ 分辨率为:320\*480， 4.3’ 和 7’ 分辨率为： 800\*480）
3. 16 位真彩显示
4. 自带触摸屏，可以用来作为控制输入

并且TFT屏有多达34个引脚，具体引脚名称如下图所示：：

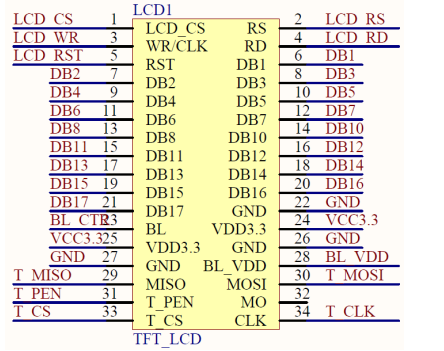


图3-7 TFT屏接口电路图

TFT屏为大功率器件，无法使用单片机的IO口直接进行驱动，所以需要驱动芯片，而本系统采用的驱动芯片为ILI9341型，该驱动芯片功能强大，比如:其自带172800kb的显存。另外其主要工作模式为16位数据读写及缓存模式，该模式下，ILI9341驱动会将所有像素点的颜色数据自动转化为RGB565格式，即进行数据的压缩，没转化之前每个像素点的RGB分别为8位的，即RGB888格式，也就是每个像素点的颜色信息需要3字节的存储空间。由于我们主要使用的颜色以红色为主，并且不需要那么多数据位，但是也要保证基本常用颜色数据的准确性，所以可以将RGB888数据格式转换为RGB565格式，这样在颜色数据不出错的前提下，就把3字节的数据压缩到2字节。2.8寸TFT屏的分辨率为320×240，也就是该TFT屏共有320×240=76800个像素点，每个像素点的颜色数据减少一字节，那就一共减少了76800字节的数据信息，大大提示了系统工作的效率。

在RGB565格式下 ILI9341的 16 位数据线与 MCU 的 16 位数据线以及 LCD GRAM 的对应关系如表所示：

|  |  |  |
| --- | --- | --- |
| 9341总线 | MCU数据（16位） | LCD GRAM（16位） |
| D17 | D15 | R4 |
| D16 | D14 | R3 |
| D15 | D13 | R2 |
| D14 | D12 | R1 |
| D13 | D11 | R0 |
| D12 | NC | NC |
| D11 | D10 | G5 |
| D10 | D9 | G4 |
| D9 | D8 | G3 |
| D8 | D7 | G2 |
| D7 | D6 | G1 |
| D6 | D5 | G0 |
| D5 | D4 | B4 |
| D4 | D3 | B3 |
| D3 | D2 | B2 |
| D2 | D1 | B1 |
| D1 | D0 | B0 |
| D0 | NC | NC |

表3-8 16位数据与显存对应关系图

## 3.4 停车场环境监控电路设计

本系统主要的环境数据监测为环境温湿度监测以及光照强度监测，监测信息主要供管理员登录后查看，方便管理员能够了解停车场的环境信息。

### 3.4.1 停车场环境温湿度监测电路设计

本系统用于监测环境温湿度数据信息的主要是DHT11温湿度传感器，DHT11主要输出数字量数据，可以连接至单片机的任一IO口，并且只需要将该IO口配置为上拉输入即可正常使用。并DHT11主要有3个引脚，分别为：VCC、GND以及DO。

DHT11与STM32连接的原理图如下：

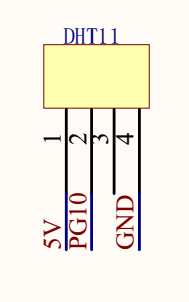


图3-9 DHT11与STM32单片机连接的原理图

### 3.4.2 停车场环境光照强度监控电路设计

本系统用于监控停车场环境光照强度数据信息的传感器为光照强度传感器，然后自动控制电路为LED照明灯电路，主要为夜间进行停车或取车操作的用户提供便利。

另外，光敏电阻因光照强度不同导致的回路电流变化，这个回路电流不能直接被单片机进行读取，需要先将该电流数据通过LM393比较器进行比较后，才能传输给单片机。

光敏电阻与LM393比较器的原理图如下：



图3-10 光敏电阻与LM393比较器连接的电路原理图

经过比较器对光敏电阻回路电流数据处理后，就可以直接发送给单片机，其中我们使用的上图中的AO输出，将其与单片机的ADC引脚相连。电路原理图如下：

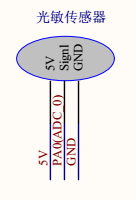


图3-11 光敏电阻模块与STM32连接的电路原理图

相应的照明灯控制电路主要采用高亮LED灯，因为其工作电流和工作电压均不大，所以可以直接使用STM32的任一引脚对其进行控制，本系统采用低电平触发，也就是说，将LED灯的正极与电源正极相连，将LED灯的负极连接至单片机的IO口，如果需要点亮LED灯时，通过该IO口输出低电平即可。具体原理图如下：

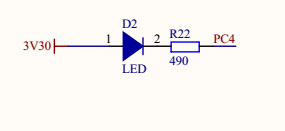


图3-12 照明灯电路原理图

## 3.5 电机驱动电路设计

本系统的机械部分主要由直流电机来进行控制升降台的升降，之所以采用直流电机时因为直流电机的控制很简单，只有控制正反转和控制输出的PWM波来调速就可以实现系统的升降平台的正常工作。

电机与STM32的接口部分原理图如下：

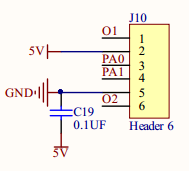


图3-13 直流电机原理图

因为电机正常工作需要的电路远远超过STM32单片机引脚可以通过的最大电流所以我们需要采用TB6612来对直流电机进行驱动。具体使用时，将电机的正负极接入TB6612的输入端（AIN1和AIN2 或者 BIN1和BIN2），这样电机和驱动部分的电路就算完成，然后是驱动与STM32之间的接口电路，TB6612与单片机之间主要由3个端口进行通信，两个是用来控制正反转的，即当单片机设置AO1=1,AO2=0时，电机正转，反之则反之，还有一个端口是输出PWM的，STM32共有8个定时器，每个定时器都有4个输出通道，这4个通道都可以被配置为PWM输出通道，所以可以将TB6612的PWM端口与STM32单片机的任一定时器中的任一输出通道相连。

TB6612的电路原理图如下：

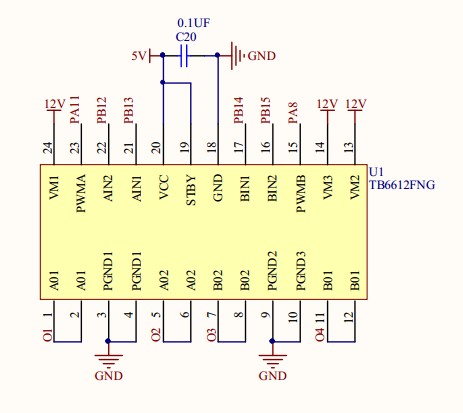


图3-14 TB6612电路原理图

## 3.6 指纹模块电路设计

本系统的指纹信息采集工作主要由AS608指纹传感器完成，该模块主要工作原理为光学图像信息采集，并且在该模块内部还内置了DSP处理芯片，可以高效的读取以及对比指纹图像信息，并配有光学触碰检测，也就是说，该模块先监测感应区有没有手指，然后才开始采集指纹图像信息，这样可以使资源浪费大大降低，另外该模块与单片机的通信采用串口通信，也就是通信接口仅占用2根线，还有一根是触碰感应输入端，所以使用AS608指纹模块进行嵌入式开发不需要连接大量的的数据线，这样使该模块的使用有了很高的容错率，并且系统的工作也更加稳定。

AS608的共有8个输出端口，这8个端口的具体功能如下表所示：

|  |  |  |
| --- | --- | --- |
| 序号 | 名称 | 介绍 |
| 引脚1 | VCC | 电源正极 |
| 引脚2 | TX | 串口输出 |
| 引脚3 | RX | 串口输入 |
| 引脚4 | GND | 电源负极 |
| 引脚5 | TCH | 触摸输入 |
| 引脚6 | VCC | 触摸感应正极 |
| 引脚7 | USB\_VCC | USB正极 |
| 引脚8 | USB\_GND | USB负极 |

表3-15 AS608主要引脚功能说明

指纹识别模块电路原理图如下：

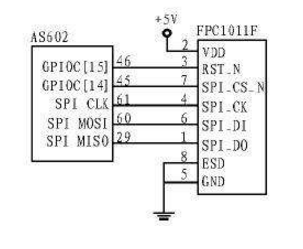


图3-16 AS608电路原理图

# 

# 第4章 系统软件设计

## 4.1 主控程序设计

本小节主要介绍系统运行的总体流程图，主控程序主要功能是对STM32单片机的基本功能进行初始化，比如串口初始化，时钟周期进行PLL倍频设置以及导入基本的头文件“sys.h和delay.h”等，另外主控函数还负责各个子函数的初始化过程，以及定义一些需要的全局变量。

### 4.1.1 主控程序设计流程图

指纹模块扫描

主控数据处理

机械部分控制

TFT屏显示

传感器数据监测

照明灯控制

得到用户指纹信息

得到环境数据信息

取车或停车

人机交互

夜间照明

初始化

图4-1 主控程序运行流程图

主控程序作为系统运行的核心，就像一个国家的元首一样，需要管理整个系统的所有事情，但在实际的程序编写过程中，一般都把所有的逻辑函数和各个部分的驱动函数都写入各个的子函数之中，主控程序只负责调用即可，这样就将一个系统的程序清晰的分成一个个小块，在进行程序的调试时就非常方便，另外这样写出来的程序也非常美观。

### 4.1.2 部分主控代码以及介绍

本部分将通过部分程序代码来详细阐述主控程序的具体功能。

#define usart2\_baund 57600//串口2波特率，根据指纹模块波特率更改

u8 tmp\_buf[33];

u8 page=0;

char \*str1;

u16 t1,t2,t3,t4;

u8 use1=0,use2=0,use3=0,use4=0;

u8 use\_start=0;

u8 ste;

u8 use\_error=0; //用户出错 警告信息显示时间变量

//u8 car[4]={0,0,0,0,}; //共4个停车位可供选择

u16 car\_id[4]={0,0,0,0}; //使用停车位的用户ID

u8 dl\_success=0; //开始进行停车操作的标志量

u8 people\_car=0;

u16 people\_id;

u16 gly\_id=0;

u8 gly\_page=0;

u8 dessce=0;

int main(void)

{

Stm32\_Clock\_Init(9); //系统时钟设置

delay\_init(72); //延时初始化

uart\_init(72,115200); //串口初始化为115200

LCD\_Init();

W25QXX\_Init(); //初始化W25Q128 和SPI2

tp\_dev.init(); //触摸屏初始化

usart2\_init(36,usart2\_baund); //初始化串口2

LED\_Init();

Key\_init();

PWM\_Init(999,71); //pwm满值=7200，频率=10K

DJ\_PWM\_Init(200,7199); //控制舵机

TIM3\_Int\_Init(9999,7199);

while(font\_init()) //检查字库

{;}

while(PS\_HandShake(&AS608Addr))//与AS608模块握手

{

delay\_ms(400);

LCD\_Fill(0,40,240,80,WHITE);

Show\_Str\_Mid(0,40,"未检测到模块!!!",16,240);

delay\_ms(800);

LCD\_Fill(0,40,240,80,WHITE);

Show\_Str\_Mid(0,40,"尝试连接模块...",16,240);

}

AS\_Init();

gly\_id\_save();

while(motor\_rest()) //电梯复位

{

Show\_Str(30,140,120,24,"电梯复位中",24,0);

}

LCD\_Clear(WHITE);

delay\_ms(100);

while(1)

{

use\_control();

}

}

void use\_control(void) //所有页面最终都会先回归主页面，再从主页面进入其他页面

{

if(page==0) //显示主页面 登录界面 一直扫描指纹模块，看是否有人登录，并且得出用户是否来取车

{

DL\_Display();

use\_as608\_scan();

}

else if(page==1) //来存车

{

save\_car\_display();

}

else if(page==2) //来取车或者查看信息

{

qu\_car\_display();

}

else if(page==3) //管理员登录界面

{

if(gly\_page==0)

{

Gly\_page\_0\_display();

}

else if(gly\_page==1)

{

Gly\_page\_1\_display();

}

else if(gly\_page==2)

{

Gly\_page\_2\_display();

}

gly\_scan();

}

}

通过以上代码可以看出，主控程序结构非常清晰，先是定义各种全局变量，然后配置STM32的内置基本功能，比如配置系统的时钟周期、系统的延时初始化已经系统的串口初始化，这些基本功能配置完后就开始运行各个模块的初始化，比如TFT屏的初始化、触摸初始化、电机初始化以及指纹模块初始化等，这些都运行完后就可以正常使用系统中的各个模块了，然后开始进行电梯复位，当电梯复位完成后就进入死循环，正式开始系统的功能实现了。

## 4.2 传感器程序设计

本系统的主要使用的传感器为DHT11温湿度监测传感器以及光照强度监测传感器，DHT11监测的温湿度主要是为了给管理员关于机械车库附近的基本物理信息数据，因为机械车库的机械设备对环境要求比较高，就比如如果城市的某区域湿度比较大，这样就需要管理员考虑设备的老化速度快等问题以及特殊电子设备的防潮问题。而监测机械车库附近的光照强度是为了在夜间光线比较昏暗时为用户提供基本的照明，防止因光线不足引起一些人为事故。

### 4.2.1 传感器程序设计流程图

图4-2 传感器程序运行流程图

该部分为传感器监测及照明灯控制部分，其中光照强度传感器先进行环境光照强度的监测，监测完成后直接进行判断，如果光照强度过低，则直接打开照明灯，然后再将数据交给单片机用作显示，而DHT11环境温湿度监测传感器则是实时的监测环境的温湿度数据，然后不停的讲数据发送给主控用作显示，而TFT屏显示环境温湿度数据和环境光照强度数据需要先进行管理员的登录操作，只有管理员才能查看具体的数据信息，普通用户无法查看相关数据信息。

环境温湿度监测

环境光照强度监测

光照强度过低

打开照明灯

将数据传回主控

是

否

### 4.2.2 传感器部分代码及介绍

该部分代码主要步骤为：先是对传感器与STM32单片机的接口引脚进行相关配置，因为就比如DHT11给单片机输入数字量，所以DHT11可以连接至单片机的任一引脚，将其进行简单的输入配置即可，但是对于光敏电阻而言，光敏电阻给主控输入的是模拟量，虽然STM32有多达21路ADC输入端口，但是这些端口是特定的引脚才具有的功能，并不是所有的引脚都可以作为ADC输入端，并且将ADC输入引脚配置成模拟输入模式之后，还需要对相应的ADC寄存器进行配置，最后在读取光敏电阻的模拟量时，是先连续读取10次的数据，然后再求均值，这是采用均值滤波算法，对数据进行处理，这样可以使数据更加稳定也更加精确。

具体代码如下：

void Adc\_Init(void)

{

//先初始化IO口

RCC->APB2ENR|=1<<2; //使能PORTA口时钟

GPIOA->CRL&=0XFFFFFF0F;//PA1 anolog输入

RCC->APB2ENR|=1<<9; //ADC1时钟使能

RCC->APB2RSTR|=1<<9; //ADC1复位

RCC->APB2RSTR&=~(1<<9);//复位结束

RCC->CFGR&=~(3<<14); //分频因子清零

//SYSCLK/DIV2=12M ADC时钟设置为12M,ADC最大时钟不能超过14M!

//否则将导致ADC准确度下降!

RCC->CFGR|=2<<14;

ADC1->CR1&=0XF0FFFF; //工作模式清零

ADC1->CR1|=0<<16; //独立工作模式

ADC1->CR1&=~(1<<8); //非扫描模式

ADC1->CR2&=~(1<<1); //单次转换模式

ADC1->CR2&=~(7<<17);

ADC1->CR2|=7<<17; //软件控制转换

ADC1->CR2|=1<<20; //使用用外部触发(SWSTART)!!! 必须使用一个事件来触发

ADC1->CR2&=~(1<<11); //右对齐

ADC1->SQR1&=~(0XF<<20);

ADC1->SQR1|=0<<20; //1个转换在规则序列中 也就是只转换规则序列1

//设置通道1的采样时间

ADC1->SMPR2&=~(3\*1); //通道1采样时间清空

ADC1->SMPR2|=7<<(3\*1); //通道1 239.5周期,提高采样时间可以提高精确度

ADC1->CR2|=1<<0; //开启AD转换器

ADC1->CR2|=1<<3; //使能复位校准

while(ADC1->CR2&1<<3); //等待校准结束

//该位由软件设置并由硬件清除。在校准寄存器被初始化后该位将被清除。

ADC1->CR2|=1<<2; //开启AD校准

while(ADC1->CR2&1<<2); //等待校准结束

//该位由软件设置以开始校准，并在校准结束时由硬件清除

}

以上代码主要是ADC寄存器的配置以及ADC输入引脚的配置，配置完成后，就可以将光敏电阻或连接至本系统之中，光敏电阻就可以正常的给主控发送数据了，然后再进行AD数据的读取，具体函数如下：

//获得ADC1某个通道的值

//ch:通道值 0~16

//返回值:转换结果

u16 Get\_Adc(u8 ch)

{

//设置转换序列

ADC1->SQR3&=0XFFFFFFE0;//规则序列1 通道ch

ADC1->SQR3|=ch;

ADC1->CR2|=1<<22; //启动规则转换通道

while(!(ADC1->SR&1<<1));//等待转换结束

return ADC1->DR; //返回adc值

}

//获取通道ch的转换值，取times次,然后平均

//ch:通道编号

//times:获取次数

//返回值:通道ch的times次转换结果平均值

u16 Get\_Adc\_Average(u8 ch,u8 times)

{

u32 temp\_val=0;

u8 t;

for(t=0;t<times;t++)

{

temp\_val+=Get\_Adc(ch);

delay\_ms(5);

}

return temp\_val/times;

}

以上部分为光敏电阻部分的配置程序，主控在读取光敏电阻的数据时仅需要直接调用最后的那个读取函数即可。

然后再介绍DHT11的配置函数，DHT11因为是数字量输出，所以配置就相应的简单的多，具体代码如下：

//初始化DHT11的IO口 DQ 同时检测DHT11的存在

//返回1:不存在

//返回0:存在

u8 DHT11\_Init(void)

{

RCC->APB2ENR|=1<<4; //使能PORTG口时钟

GPIOC->CRH&=0X0FFFFFFF;//PORTG.11 推挽输出

GPIOC->CRH|=0X30000000;

GPIOC->ODR|=1<<15; //输出1

DHT11\_Rst();

return DHT11\_Check();

}

## 4.3 电机及驱动程序设计

本小节主要介绍单片机是如何通过TB6612直流电机驱动来间接的控制直流电机的，实际控制工程中也很简单，TB6612与STM32单片机的连接端口仅有3个，分别为电源正负极以及PWM控制端口，STM32输出PWM波形仅仅依靠STM32的定时器就可以完成，因为STM32单片机内部任一一个定时器都有多达4个通道，这4个通道可以作为输入捕获也可以作为PWM输出，具体使用过程如下：

比如采用TIM3\_CH1作为PWM输出端，在对TIM3进行配置时就需要将TIM3的预分频寄存器及预装载值寄存器内的数据分别配置为7199和1999，即psc=7199,arr=1999,具体的运算公式如下：

1：先计算时钟频率f0，即1秒钟内可以计多少个数，预设psc=7199。

2:然后根据舵机需要20Ms的脉冲，求预装载寄存器的预装载值arr。

配置完TIM3定时器后，其就可以稳定的输出20Ms的脉冲，然后在对TIM3-CCR1寄存器进行赋值来达到输出PWM波形的目的，比较令TIM3-CCR1=1000，那么该定时器就将输出频率为50Hz的、占空比为50%的波形。

最后通过使用，输出不同占空比的波形，观察舵机转动的角度，来最终确定该输出多大占空比的PWM波。

### 4.3.1 电机及驱动控制流程图

用户车位选择

机械控制开启

是否开始启动

是

否

等待用户自主启动

图4-3 电机驱动程序运行流程图

以上流程图仅仅为机械控制的基本操作流程图，主要介绍用户存车时的必要操作，而用户取车时基本操作和存车是一样的，就是进入的界面不同，具体的操作流程图将在下面详细阐述。下面的流程图为用户存车和取车的详细流程图：

用户刷指纹

是否已存过车

否

存车操作界面

信息查看界面

是

选择停车位位

是否开始启动

否

等待

是

是否取车

否

返回主页面

机械系统运行

是否开始返回

机械系统运行

否

等待

是

图4-4 电机驱动程序运行具体流程图

该部分流程图详细的解释了用户操作本系统的具体流程，该流程图中判断是否开始启动和返回都是通过触摸按键实现的，而机械系统运行部分主要通过碰撞开关来获取升降台的具体位置信息，实际实践中，是通过死循环来运行机械系统，当碰到了触碰开关就关闭电机并跳出死循环，进行下一轮程序的运行。

### 4.3.2 电机及驱动控制部分代码及介绍

本小节主要通过给出具体的程序来详细的介绍本系统是如何通过C语言代码来使系统运行的，主要有电机的初始化配置以及如果通过程序输出PWM和电机正反转的控制。

void Motor\_Init(void)

{

RCC->APB2ENR|=1<<7; //使能PORTa时钟

GPIOF->CRL&=0XFFFFFF00;

GPIOF->CRL|=0X00000022;//PA.5推挽输出

}

void PWM\_Init(u16 arr,u16 psc)

{

Motor\_Init(); //初始化电机控制所需IO

RCC->APB2ENR|=1<<11; //TIM1时钟使能

GPIOA->CRH&=0XFFFFFFF0; //PA8清除之前的设置

GPIOA->CRH|=0X0000000B; //复用功能输出

TIM1->ARR=arr; //设定计数器自动重装值

TIM1->PSC=psc; //预分频器设置

TIM1->CCMR1|=7<<4; //CH1 PWM2模式

TIM1->CCMR1|=1<<3; //CH1预装载使能

TIM1->CCER|=1<<0; //OC1 输出使能

TIM1->BDTR|=1<<15; //MOE 主输出使能

TIM1->CR1=0x0080; //ARPE使能

TIM1->CR1|=0x01; //使能定时器1

}

以上为电机初始化配置，下面给出的程序是如何具体的控制电机正反转和输出PWM。

u8 motor\_rest(void) //电梯复位

{

if(KEY\_DOWN==1)

{

motor\_down();

return 1;

}

else

{

motor\_stop();

PWM=18;

return 0;

}

}

void motor\_up(void)

{

PWMA=500;

AIN1=1;

AIN2=0;

}

以上给出了系统初始化之后进行的升降台复位程序已经升降台上升的程序，其中PWMA为控制电机转速的PWM输出，而PWM为控制模拟门杆的舵机PWM输出。

## 4.4 显示屏程序设计

本小节主要介绍TFT屏的基本操作流程，并给出了在系统运行时，TFT屏显示函数具体被插入到哪个部分，并在最后给出了部分界面显示函数的代码，来详细阐述本系统的人机交互界面的设计。

以下为TFT屏的基本控制流程图：

硬复位

初始化序列

设置坐标

写GRAM指令

读GRAM指令

写颜色数据

读颜色数据

LCD显示

单片机处理

图4-5 TFT屏操作流程图

### 4.4.1 显示屏控制流程图

本次主要介绍在系统中，什么时候调用TFT屏、使用TFT屏显示什么信息以及TFT屏之间页面之间的关系流程图。

主页面（刷指纹）

用户登录界面

是否登录成功

管理员登录界面

返回

取车

查看信息

操作运行界面

存车

查看信息

用户管理

返回

否

警告信息

是

录指纹

删指纹

图4-6 TFT屏具体操作流程图

本处通过流程图详细的介绍了系统各个页面在实际操作过程中是如何相互转化的，各个页面之间的调转都是通过TFT屏的触摸功能实现的，即在TFT屏上设计各种虚拟按键接口，当手指点击该区域后，就进入相应的操作界面。

### 4.4.2 人机交互界面部分代码及介绍

本次通过部分程序代码来具体的阐述系统各页面是如何通过程序设计的。

void Success\_0\_display(void) //用户存车等待界面 显示5秒

{

POINT\_COLOR=BLACK;

Show\_Str\_Mid(0,50,"正在进行存车操作",16,240);

}

void DL\_Display(void) //登录界面 page==0

{

POINT\_COLOR=BLUE;

Show\_Str\_Mid(0,50,"机械车库系统",16,240);

POINT\_COLOR=BLACK;

Show\_Str(48,100,144,24,"请刷指纹登录",24,0);

}

void Error\_0\_display(void) //用户没有录过指纹，警告界面 显示5秒

{

POINT\_COLOR=BLUE;

Show\_Str\_Mid(0,50,"没有您的指纹信息",24,240);

POINT\_COLOR=BLACK;

Show\_Str\_Mid(0,100,"请联系管理员录指纹信息",16,240);

}

本次只是给出了3个显示界面的程序设计函数，其他显示界面的设计过程与之类似。

## 4.5 指纹模块程序设计

本小节主要介绍指纹模块的程序设计，主要包括录指纹删指纹流程图以及刷指纹流程图，指纹模块的程序设计最复杂的就是数字图像处理技术以及指纹识别算法技术，但是选用的AS608指纹模块内部集成的高性能DSP处理芯片可以很好地进行图像识别，我们只需要通过串口来使主控与AS608建立通信，通过对指纹模块进行数据的读写操作，就可以使指纹模块给主控返回我们需要的数据信息。大大简化了系统的复杂程度。

### 4.5.1 指纹模块控制流程图

本部分主要分为两块，分别为录指纹和删指纹流程图以及刷指纹流程图，以下为录指纹和删指纹流程图：

开始读取图像

生成特征值

发送特征值

输出存储ID

录指纹结束

输入要删除的ID

删指纹结束

返回

图4-7 指纹模块删指纹和录指纹流程图

由以上原理图可以看出，录指纹和删指纹的操作流程并不复杂，步骤也不多，该部分的主要难点在于人机交互界面的设计，因为要输入ID，所以要设计虚拟键盘，并且根据触碰点击还有判断出点击的ID是多少，另外还较为复杂的算法运算和图像处理部分也有指纹模块内部自己完成，所以本部分的流程图看起来并不复杂。

下面将给出刷指纹的操作流程图：

是否触发触摸

读取图像信息

数据库比对

输出

是

否

图4-8 指纹模块刷指纹流程图

### 4.5.2 指纹模块控制部分代码及介绍

本次通过给出指纹模块的部分初始化代码来具体阐述实践使用中，指纹模块是如何工作的。

u32 AS608Addr = 0XFFFFFFFF; //默认

//初始化PA6为下拉输入

//读摸出感应状态(触摸感应时输出高电平信号)

void PS\_StaGPIO\_Init(void)

{

RCC->APB1ENR |= 1<<3;//使能PORTc时钟

GPIOC->CRL &=0XFF0FFFFF;//复位PA6

GPIOC->CRL |=0X00800000;//输入模式，默认下拉

GPIOC->BRR |=1<<5;//下拉

}

//串口发送一个字节

static void MYUSART\_SendData(u8 data)

{

while((USART2->SR&0X40)==0);

USART2->DR = data;

}

//发送指令码

static void Sendcmd(u8 cmd)

{

MYUSART\_SendData(cmd);

}

//发送校验和

static void SendCheck(u16 check)

{

MYUSART\_SendData(check>>8);

MYUSART\_SendData(check);

}

本次给出的是指纹模块部分关于触摸感应输入引脚的初始化函数以及指纹模块和STM32单片机之间的串口通信函数。

# 第5章 系统测试

本系统为机械车库系统，主要功能是能够为城市提供停车位，并且主要采用指纹模块作为登录识别的主要手段，并使用TFT屏作为人机交互的主要设备，另外还配有传感器用来检测车库环境的主要信息。

本系统测试的主要功能如下：

1. 已录过指纹的用户自主进行停车操作、取车操作以及信息查看操作。
2. 管理员操作界面，主要包括用户信息管理、停车场信息查看以及环境信息查看等。

## 5.1 已录指纹的用户自主操作系统测试

要想自主使用机械车库，必要先找管理员录取个人指纹信息，如果已经录过指纹，那么可以直接在指纹模块上刷自己的指纹即可，如果用户之前没有在该停车场停过车，那么系统将自动转到停车位选择界面，如果用户之前已经停过车，那么系统将会转入用户信息查看界面，在该界面用户可以查看自己的使用时间并且可以进行停止使用停车位的操作。

### 5.1.1 系统开机测试

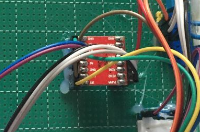
本部分测试主要为各个模块的初始化检测及控制，比如先是主控与指纹模块建立信息连接，唤醒指纹模块，然后检测电梯是否在一楼，否则将进行自动复位等。

系统通电后，各个模块的电源指示灯都正常亮起，着说明系统的电源连接没有问题，如图5-1所示：

**启动开关**

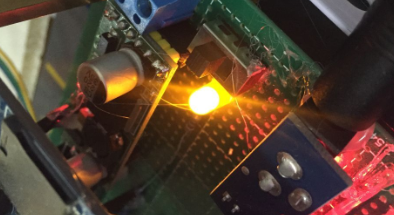
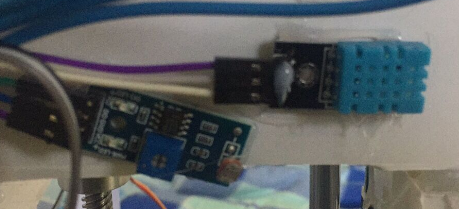
**电机**

**TB6612**



**模拟门杆**

**指纹模块**

图5-1 系统供电测试图

**碰撞开关**

**TFT屏**

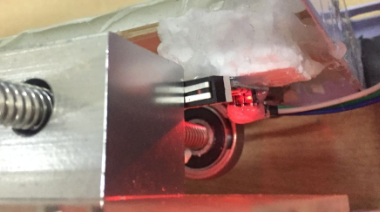
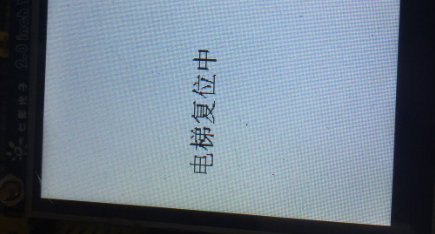
**供电系统**

**光敏电阻模块**

**DHT11**

**模拟路灯**

由测试图可以看出，当系统通电后，系统各部分都能正常工作，这也是下面测试的基础，往往因为供电的电流不足而导致一些大功率用电器无法正常工作，所以本系统的供电电源采用12V5A的电源适配器，足以满足系统的所有用电需求。

下面将电梯手动往上拧，使下端的碰撞开关处于无碰撞状态，这样是模拟一开始电梯没有处于起始位置的情况，具体如图5-2所示：

手动拧丝杆，使滑块与碰撞开关分离

图5-2 系统电梯自动复位测试图

由测试图可以看出，当电梯在通电后没有处于初始位置时，在系统通电后，会先进行复位操作，并在TFT显示屏上进行相应的提示。这样是为了防止在操作过程中突然断电，而电梯操作没有完成，如果没有复位操作，那么再次上电时，由于电梯没有处于可控制的状态而导致系统卡顿或者卡死的情况。

### 5.1.2 用户自主停车操作测试

该部分测试主要为已录有指纹信息的用户进行自主的停车操作，用户刷完指纹后，将会提示用户之前存储的信息ID以及本次刷指纹的匹配得分，匹配得分是本次刷的指纹与数据库里指纹信息的对比相似度，当相似度过小时也视为刷卡失败。同样的，如果用户之前没有在系统里录过指纹信息，那么用户刷完指纹后将会提示没有搜索到指纹信息的提示。具体测试如图5-3所示：

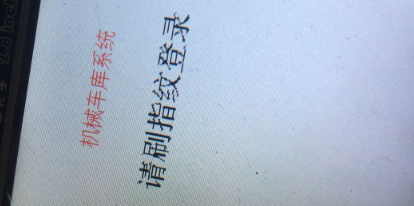
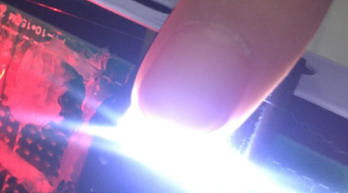


图5-3 用户刷指纹测试图

如果指纹模块里没有用户的指纹信息或者匹配得分过小则显示刷指纹失败，具体如图5-4所示：

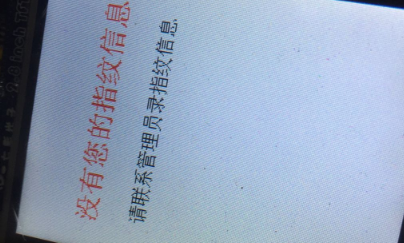
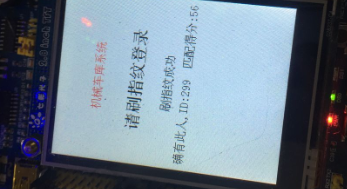
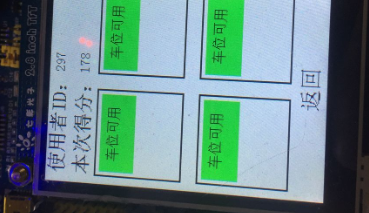


图5-4 刷指纹失败测试题

如果刷指纹成功，并且用户之前没有进行停车操作，则会先给出刷指纹成功等信息并自动进入选择停车位的界面，具体测试题如图5-5所示：

图5-5 用户刷卡成功测试图

用户刷卡成功后，进入停车位选择界面，这里将会显示目前停车场的车位选择虚拟按键，如果车位没有被占用，则用绿色色块表示，并且中间还会用汉字显示 车位可用，用户可以直接通过点击进行车位选择，同样的，如果某个车位被占，则相应的位置会用红色色块显示，并且用汉字现在车位被占，具体效果下面的步骤会有。

用户选择完停车位后会启动机械控制系统，根据用户选择的车位把用户拉到相应的楼层，比如用户选择停车位1和停车位2（第一行的停车位），则表示1楼，用户选择第二行的停车位，则表示二楼，会触发电机进行不同的操作，本次先测速1楼的停车位，具体测试题如图5-6所示：

启动开关

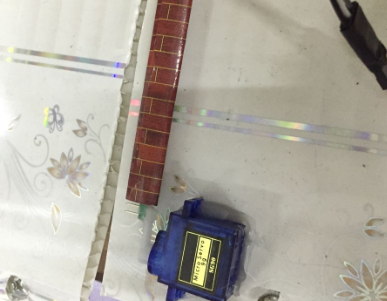


图5-6 停车操作测试图

测试时，显示屏显示进行操作显示界面，该界面不能被人为退出，这样是为了提醒下一位用户，避免误操作，并且按下启动开关后，舵机控制的模拟门杆抬起，用户可以进行停车操作，然后完成后再点击一下触碰开关，表示操作完成，然后TFT显示屏也会返回主页面。

下面再测试第二位用户的停车操作，具体效果如图5-7所示：

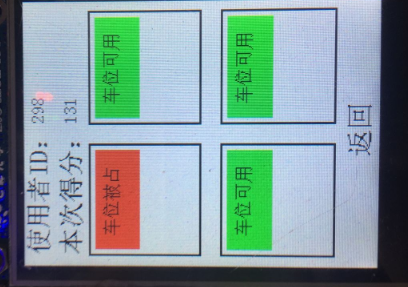


图5-7 第二个用户的停车操作测试图（1）

第二个用户进行停车操作时，具体操作流程和第一个用户一样，唯一不同的是停车位选择界面，因为第一个用户已经选择了一个1楼的停车位，所以对应位置用红色色块显示，并且用汉字显示车位被占，现在第二个用户测试选择一个2楼的停车位，具体测试题如图5-8所示：



图5-8 第二个用户的停车操作测试图（2）

剩下的操作也与第一个用户操作流程一样，就是在按下启动按键后，因为选择的是2楼的车位，所以电梯首先会抬升，抬升到2楼后会自动打开门杆，用户停完车后再按一下触摸开关，然后门杆落下，并且电梯开始下降，到1楼后，操作完成，TFT屏的显示界面也会相应的返回主界面。至此，用户的自主停车操作测试已经基本完成，各项功能均正常。

### 5.1.3 用户查看信息以及取车操作测试

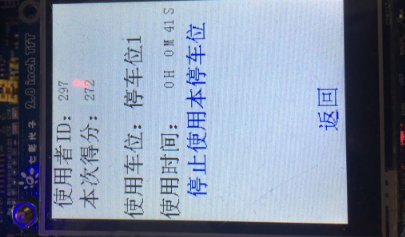
已经存过车的用户再次刷卡后，系统会自动进入信息查看界面，该界面主要显示了用户的指纹信息、存车时间、存车位以及相应的虚拟按键，具体界面如图5-9所示：

图5-9 信息查看界面测试图

该界面主要显示了用户的相关信息，并且还有返回虚拟按键以及停止使用停车位的虚拟按键，用户点击返回虚拟按键就会直接返回主页面，相当于用户本次仅仅查看一下停车时间信息，还继续使用本停车位，下次刷指纹时依旧会进入信息显示界面。如果用户点击停止使用本停车位，那么将会进入取车操作，具体测试图如图5-10所示：

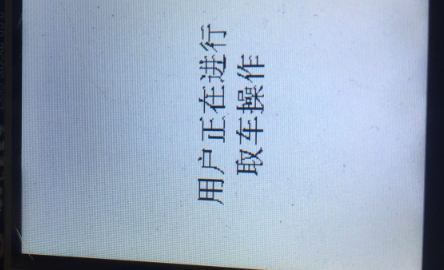


图5-10 用户取车操作测试图

用户进入取车操作后，操作流程和存车流程是一样，如果该用户选择的是1楼，先按下启动按键，然后门杆抬起，车辆开出后再按一次触摸开关，然后门杆落下取车操作完成，如果该用户选择的是2楼，按下启动开关后，电机会先抬升至2楼，然后等下一次触摸按键输入信号后再返回1楼。

## 5.2 管理员操作系统测试

本部分主要是测试管理员登录系统，管理员的指纹信息ID被存储到固定的位置（299），管理员登录与普通用户一样，也是先进行刷指纹,刷成功后直接进入管理员界面，具体界面如图5-11所示：

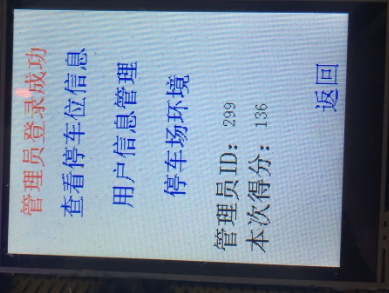
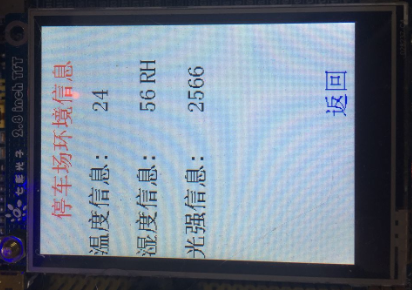


图5-11 管理员界面测试图（1）

### 5.2.1 管理员基本信息查看系统测试

由上图易知管理员界面主要有四个虚拟按键（蓝色字体）以及管理员的ID信息和本次刷指纹匹配得分情况，管理员如果点击返回按键，则直接返回主页面，如果点击其他几个虚拟按键则会进入相应的显示界面，本处先测试进入停车场环境信息界面以及查看停车场车位信息界面具体测试效果如他5-12所示：

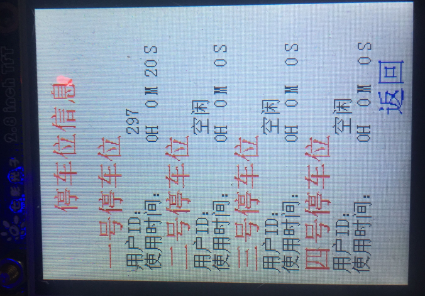


图5-12 管理员界面测试（2）

由测试图可知，进入停车场环境信息界面后，会显示停车场的温湿度信息，并且还可以查看系统环境的光照强度信息。同样的进入停车位信息界面后，会显示所有车位的情况，当车位没人使用时会显示空闲，否则会显示占用此停车位用户的ID，并且还会显示该用户使用停车场的时间。这两个界面都有相应的返回按键，点击返回按键后会返回上一步。

### 5.2.2 用户信息管理操作系统测试

刚刚对管理员的基本信息查看界面进行了测试，下面再对用户信息管理界面进行测试，管理员点击相应的按键后，会直接进入用户信息管理界面，具体如图5-13所示：

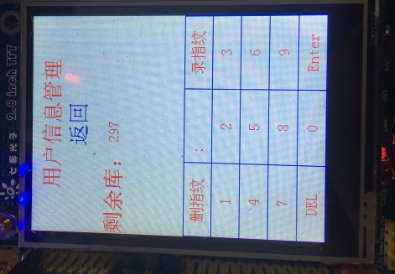


图5-13 用户信息管理界面

该界面主要是为管理员进行新用户的指纹信息添加以及相应指纹信息的删除操作，本界面除了有数量众多的虚拟按键外，还显示了剩余库的信息，一般指纹模块因为内部存储空间是固定的，并且每个指纹信息经过处理后占用的存储空间是固定的，所以一般一个指纹模块最多能存储300个指纹图像信息，剩余库就是显示还有多少剩余空间数量。

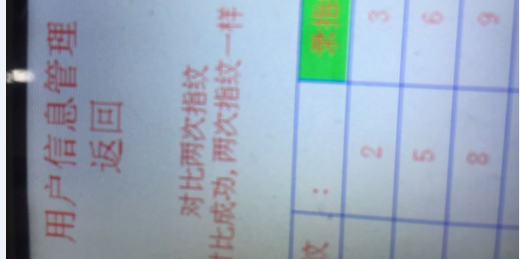
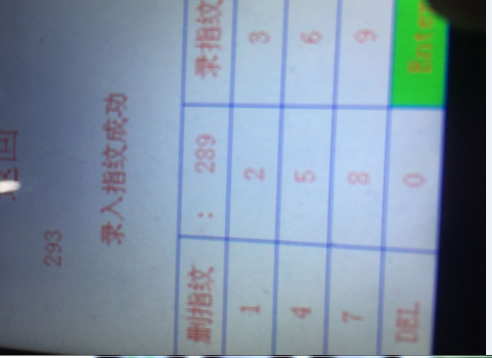
 下面先测试录指纹操作，具体测试图如图5-14所示：

图5-14 管理员录指纹操作测试图

具体操作时，先是点击录指纹按键，然后屏幕会提示请刷指纹，这时用户将自己的手指放到指纹模块的刷指纹区域，录指纹一共需要录2次，只有两次指纹信息数据相似度都超过80%时才成功，此时将提示要输入该指纹信息的存储ID，直接用下面的虚拟按键输入即可，输入完点击Enter按键，自此该用户的录指纹操作基本完成。

下面再测试删指纹操作，具体如图5-15所示：

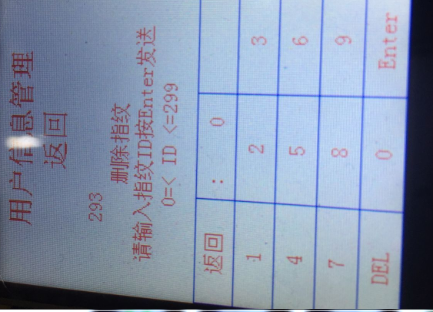


图5-15 管理员删指纹操作测试图

进入删指纹操作后，先通过按键输入要删除的指纹ID，然后点Enter按键即可。自此，管理员所有界面的基本功能也测试完毕，经测试，各项功能正常。

## 5.3 自动照明灯系统测试

本部分主要测试系统的照明灯，照明灯是在夜间光线昏暗时为用户提供基本的照明设施，该部分采用高亮LED灯组和继电器搭配，当光敏电阻返回的光强过低时，单片机将控制继电器来打开照明灯，具体测试效果如图5-16所示：

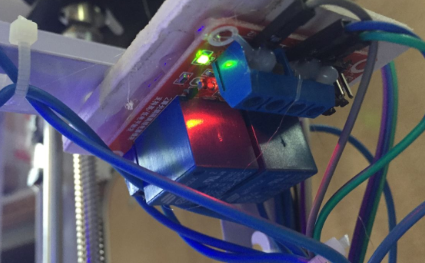


图5-16 自动照明系统测试图

经测试，当光线比较暗时（用手指捂住光敏电阻探头）继电器打开，将高亮LED灯组的正极与电源正极接通，照明系统工作，当光线比较强时（用手机手电筒照射光敏电阻探头），继电器关闭，照明灯熄灭。自此，系统的所有功能均测试完成，各项功能均正常，系统供电也稳定，系统测试完成。

# 结 论

经过一学期的学习和工作，基于ARM的马路机械车库系统设计基本完成，一开始选题目时，根据网上的一则关于未来城市的新闻想到的这个题目，并且通过查阅相关资料得知机械车库的应用前景十分广泛，并且随着我国的快速发展，社会的机械车库的需求会越来越高，所以，我就把自己毕业设计研究方向定为研究机械车库。

又因为自己对STM32单片机进行过一段时间的学习，所以我就采用STM32为核心主控开始进行系统的设计。

先是进行系统的功能预设，明确基本功能，然后开始根据功能来采购元件，再然后是进行系统程序的编写和设计，最后是实物的搭建。

# 参考文献

1. 郑康亮. 基于IP核复用的指纹识别系统的研究与应用[D]. 广东工业大学, 2008.
2. 陈杰. 巷道堆垛式及塔式停车系统中的优化问题研究[D]. 青岛科技大学, 2017.
3. KIPBY M，SIROVICH L.Application of the Kar- hunen-Loeve procedure for the characterization of hu- man faces[J].IEEE Transactions on Pattern Analy- sis and Machine Intelligence,2006,12(1):103-108．
4. 孙英瑜. 不同所有制工业企业环境效应研究[D]. 东北财经大学, 2011.
5. 李胜善. 公交车立体车库[D]. 城市车辆, 2002.
6. 周智慧. 立体停车库刚架结构CAE分析及其优化设计[D]. 苏州大学, 2005.
7. BARTLETT M S,MOVELLAN JR,SEJNOWSKI T J.Face recognition by independent componentanalysis[J].IEEETransactionsonNeuralNetworks,2002,13(6):1450-1464．
8. Xie Z，Fang Z．Face Recognition Based on Combination of Human Per- ception and Local Binary Pattern[J].Lecture Notes in Computer Sci- ence:2012,72(2):365-373.
9. Upton E,Halfacree G.Raspberry Pi User Guide[M].New Jersey:John,2013.
10. 张冰洋. 全自动固相萃取仪控制系统的设计与实现[D]. 中南民族大学, 2013.
11. 蒋明明. 基于STM32的低端智能家居控制系统研制[D]. 西南交通大学, 2013.
12. 冯美艳. 基于STM32的参数在线配置数字滤波器[D]. 西安工程大学, 2017.
13. 龚海霞. 矿用移动式救生舱温湿度智能监控系统的研究与设计[D]. 中国矿业大学, 2014.
14. HUANGR，LIU Q，LU H，et al． Solving the small sample size problem of LDA[C].IEEE Proceedings of International Conference on pattern Recognition,USA, 2002:29-32．
15. 张浩伟. 基于智能控制和云平台技术的远程植物工厂系统研究[D]. 天津工业大学, 2017.
16. 杨志江. 小型化微带天线及雷达探测系统的设计[D]. 杭州电子科技大学, 2017.
17. 刘骁枭. 基于32位微控制器成品油混合度监测技术实验平台研究[D]. 中国石油大学（华东）, 2013.
18. 梁爽. 基于双直流电机控制的电脑鼠系统研究[D]. 吉林大学, 2015.
19. 刘定发. 基于指纹自动识别技术的出入境旅客身份验证系统研制[D]. 贵州大学, 2009.
20. 李传宝. 基于激光光谱技术的低浓度CO检测系统设计[D]. 内蒙古科技大学, 2015.
21. 张峰. 小功率直流电动机数字控制系统的开发[D]. 西安石油大学, 2013.
22. 刘滨. 基于STM32的指纹识别系统的研究[D]. 燕山大学, 2014.
23. 赵玮 代军 郭雷. 一种基于74HC245D的管道机器人的控制系统[D]. 承德石油高等专科学校, 2018.
24. 种嘉亮. 智能分层试采系统的研究[D]. 西安石油大学, 2017.
25. 邵焕杰. 便携式发射动力学参数测试系统设计方法研究[D]. 南京理工大学, 2017.

# 致 谢

借此毕业设计完成之际，我首先要感谢我的指导教师朱恒军教授，从毕业设计选题到设计完成，给予了我耐心指导与细心关怀，有了朱恒军老师耐心指导与细心关怀我才会在设计的过程中精准定位，把握需求。朱恒军老师有严肃的科学态度，严谨的治学精神和精益求精的工作作风，使我醍醐灌顶，感谢朱恒军老师给予了我这样一个珍贵的学习机会，对我在大学四年学习和实验室的研究中所给予的大力支持和悉心关怀。感谢朱恒军老师在大学四年来对我的实验指导和对我两年来参加全国大学生电子设计竞赛的鼎力相助。

感谢我大学四年来的所有老师对我在软件工程方向上的启蒙，使我对软件开发的认识更加深刻，而且在李老师的指导下，我的软件开发技术水平得到质的飞跃。还要感谢房汉雄老师对我的一贯帮助和考研期间的解惑与支持。向通信工程系全体教师致敬，对我们授业解惑、开拓创新、精心点拨、热忱鼓励。

同时，感谢与我并肩作战的309实验室全体同学，电子设计大赛的战友们，感谢关心我支持我的朋友们，感谢你们给予我的帮助，感谢电工电子实验中心为我提供的优良的实验室环境和先进的实验设备与仪器。感谢齐齐哈尔大学通信与电子工程学院为我提供的良好学习环境。

# 附录 1 指纹模块主要程序

#include <string.h>

#include "delay.h"

#include "usart2.h"

#include "as608.h"

u32 AS608Addr = 0XFFFFFFFF; //默认

//初始化PA6为下拉输入

//读摸出感应状态(触摸感应时输出高电平信号)

void PS\_StaGPIO\_Init(void)

{

RCC->APB1ENR |= 1<<2;//使能PORTA时钟

GPIOA->CRL &=0XF0FFFFFF;//复位PA6

GPIOA->CRL |=0X08000000;//输入模式，默认下拉

GPIOA->BRR |=1<<6;//下拉

}

//串口发送一个字节

static void MYUSART\_SendData(u8 data)

{

while((USART2->SR&0X40)==0);

USART2->DR = data;

}

//发送包头

static void SendHead(void)

{

MYUSART\_SendData(0xEF);

MYUSART\_SendData(0x01);

}

//发送地址

static void SendAddr(void)

{

MYUSART\_SendData(AS608Addr>>24);

MYUSART\_SendData(AS608Addr>>16);

MYUSART\_SendData(AS608Addr>>8);

MYUSART\_SendData(AS608Addr);

}

//发送包标识,

static void SendFlag(u8 flag)

{

MYUSART\_SendData(flag);

}

//写系统寄存器 PS\_WriteReg

//功能: 写模块寄存器

//参数: 寄存器序号RegNum:4\5\6

//说明: 模块返回确认字

u8 PS\_WriteReg(u8 RegNum,u8 DATA)

{

u16 temp;

u8 ensure;

u8 \*data;

SendHead();

SendAddr();

SendFlag(0x01);//命令包标识

SendLength(0x05);

Sendcmd(0x0E);

MYUSART\_SendData(RegNum);

MYUSART\_SendData(DATA);

temp = RegNum+DATA+0x01+0x05+0x0E;

SendCheck(temp);

data=JudgeStr(2000);

if(data)

ensure=data[9];

else

ensure=0xff;

if(ensure==0)

printf("\r\n设置参数成功！");

else

printf("\r\n%s",EnsureMessage(ensure));

return ensure;

}

u8 PS\_SetAddr(u32 PS\_addr)

{

u16 temp;

u8 ensure;

u8 \*data;

SendHead();

SendAddr();

SendFlag(0x01);//命令包标识

SendLength(0x07);

Sendcmd(0x15);

MYUSART\_SendData(PS\_addr>>24);

MYUSART\_SendData(PS\_addr>>16);

MYUSART\_SendData(PS\_addr>>8);

MYUSART\_SendData(PS\_addr);

temp = 0x01+0x07+0x15

+(u8)(PS\_addr>>24)+(u8)(PS\_addr>>16)

+(u8)(PS\_addr>>8) +(u8)PS\_addr;

SendCheck(temp);

AS608Addr=PS\_addr;//发送完指令，更换地址

data=JudgeStr(2000);

if(data)

ensure=data[9];

else

ensure=0xff;

AS608Addr = PS\_addr;

if(ensure==0x00)

printf("\r\n设置地址成功！");

else

printf("\r\n%s",EnsureMessage(ensure));

return ensure;

}

//功能： 模块内部为用户开辟了256bytes的FLASH空间用于存用户记事本,

// 该记事本逻辑上被分成 16 个页。

//参数: NotePageNum(0~15),Byte32(要写入内容，32个字节)

//说明: 模块返回确认字

u8 PS\_WriteNotepad(u8 NotePageNum,u8 \*Byte32)

{

u16 temp;

u8 ensure,i;

u8 \*data;

SendHead();

SendAddr();

SendFlag(0x01);//命令包标识

SendLength(36);

Sendcmd(0x18);

MYUSART\_SendData(NotePageNum);

for(i=0;i<32;i++)

{

MYUSART\_SendData(Byte32[i]);

temp += Byte32[i];

}

temp =0x01+36+0x18+NotePageNum+temp;

SendCheck(temp);

data=JudgeStr(2000);

if(data)

ensure=data[9];

else

ensure=0xff;

return ensure;

}

//读记事PS\_ReadNotepad

//功能： 读取FLASH用户区的128bytes数据

//参数: NotePageNum(0~15)

//说明: 模块返回确认字+用户信息

u8 PS\_ReadNotepad(u8 NotePageNum,u8 \*Byte32)

{

u16 temp;

u8 ensure,i;

u8 \*data;

SendHead();

SendAddr();

SendFlag(0x01);//命令包标识

SendLength(0x04);

Sendcmd(0x19);

MYUSART\_SendData(NotePageNum);

temp = 0x01+0x04+0x19+NotePageNum;

SendCheck(temp);

data=JudgeStr(2000);

if(data)

{

ensure=data[9];

for(i=0;i<32;i++)

{

Byte32[i]=data[10+i];

}

}

else

ensure=0xff;

return ensure;

}

//高速搜索PS\_HighSpeedSearch

//功能：以 CharBuffer1或CharBuffer2中的特征文件高速搜索整个或部分指纹库。

// 若搜索到，则返回页码,该指令对于的确存在于指纹库中 ，且登录时质量

// 很好的指纹，会很快给出搜索结果。

//参数: BufferID， StartPage(起始页)，PageNum（页数）

//说明: 模块返回确认字+页码（相配指纹模板）

u8 PS\_HighSpeedSearch(u8 BufferID,u16 StartPage,u16 PageNum,SearchResult \*p)

{

u16 temp;

u8 ensure;

u8 \*data;

SendHead();

SendAddr();

SendFlag(0x01);//命令包标识

SendLength(0x08);

Sendcmd(0x1b);

MYUSART\_SendData(BufferID);

MYUSART\_SendData(StartPage>>8);

MYUSART\_SendData(StartPage);

MYUSART\_SendData(PageNum>>8);

MYUSART\_SendData(PageNum);

temp = 0x01+0x08+0x1b+BufferID

+(StartPage>>8)+(u8)StartPage

+(PageNum>>8)+(u8)PageNum;

SendCheck(temp);

data=JudgeStr(2000);

if(data)

{

ensure=data[9];

p->pageID =(data[10]<<8) +data[11];

p->mathscore=(data[12]<<8) +data[13];

}

else

ensure=0xff;

return ensure;

}

if(USART2\_RX\_STA&0X8000)//接收到数据

{

if(//判断是不是模块返回的应答包

USART2\_RX\_BUF[0]==0XEF

&&USART2\_RX\_BUF[1]==0X01

&&USART2\_RX\_BUF[6]==0X07

)

{

\*PS\_Addr=(USART2\_RX\_BUF[2]<<24) + (USART2\_RX\_BUF[3]<<16)

+(USART2\_RX\_BUF[4]<<8) + (USART2\_RX\_BUF[5]);

USART2\_RX\_STA=0;

return 0;

}

USART2\_RX\_STA=0;

}

return 1;

}

//模块应答包确认码信息解析

//功能：解析确认码错误信息返回信息

//参数: ensure

const char \*EnsureMessage(u8 ensure)

{

const char \*p;

switch(ensure)

{

case 0x00:

p="OK";break;

case 0x01:

p="数据包接收错误";break;

case 0x02:

p="传感器上没有手指";break;

case 0x03:

p="录入指纹图像失败";break;

case 0x04:

p="指纹图像太干、太淡而生不成特征";break;

case 0x05:

p="指纹图像太湿、太糊而生不成特征";break;

case 0x06:

p="指纹图像太乱而生不成特征";break;

case 0x07:

p="指纹图像正常，但特征点太少（或面积太小）而生不成特征";break;

case 0x08:

p="指纹不匹配";break;

case 0x09:

p="没搜索到指纹";break;

case 0x0a:

p="特征合并失败";break;

case 0x0b:

p="访问指纹库时地址序号超出指纹库范围";

case 0x10:

p="删除模板失败";break;

case 0x11:

p="清空指纹库失败";break;

case 0x15:

p="缓冲区内没有有效原始图而生不成图像";break;

case 0x18:

p="读写 FLASH 出错";break;

case 0x19:

p="未定义错误";break;

case 0x1a:

p="无效寄存器号";break;

case 0x1b:

p="寄存器设定内容错误";break;

case 0x1c:

p="记事本页码指定错误";break;

case 0x1f:

p="指纹库满";break;

case 0x20:

p="地址错误";break;

default :

p="模块返回确认码有误";break;

}

return p;

}

#include "sys.h"

#include "usart2.h"

#include "stdarg.h"

#include "stdio.h"

#include "string.h"

#include "timer.h"

//串口接收缓存区

u8 USART2\_RX\_BUF[USART2\_MAX\_RECV\_LEN]; //接收缓冲,最大USART2\_MAX\_RECV\_LEN个字节.

u8 USART2\_TX\_BUF[USART2\_MAX\_SEND\_LEN];

//通过判断接收连续2个字符之间的时间差不大于10ms来决定是不是一次连续的数据.

//如果2个字符接收间隔超过10ms,则认为不是1次连续数据.也就是超过10ms没有接收到

//任何数据,则表示此次接收完毕.

//接收到的数据状态

//[15]:0,没有接收到数据;1,接收到了一批数据.

//[14:0]:接收到的数据长度

vu16 USART2\_RX\_STA=0;

void USART2\_IRQHandler(void)

{

u8 res;

if(USART2->SR&(1<<5))//接收到数据

{

res=USART2->DR;

if((USART2\_RX\_STA&(1<<15))==0)//接收完的一批数据,还没有被处理,则不再接收其他数据

{

if(USART2\_RX\_STA<USART2\_MAX\_RECV\_LEN) //还可以接收数据

{

TIM7->CNT=0; //计数器清空

if(USART2\_RX\_STA==0) //使能定时器7的中断

{

TIM7->CR1|=1<<0; //使能定时器7

}

USART2\_RX\_BUF[USART2\_RX\_STA++]=res; //记录接收到的值

}else

{

USART2\_RX\_STA|=1<<15; //强制标记接收完成

}

}

}

}

//初始化IO 串口2

//pclk1:PCLK1时钟频率(Mhz)

//bound:波特率

void usart2\_init(u32 pclk1,u32 bound)

{

float temp;

u16 mantissa;

u16 fraction;

temp=(float)(pclk1\*1000000)/(bound\*16);//得到USARTDIV

mantissa=temp; //得到整数部分

fraction=(temp-mantissa)\*16; //得到小数部分

mantissa<<=4;

mantissa+=fraction;

RCC->APB2ENR|=1<<2; //使能PORTA口时钟

RCC->APB1ENR|=1<<17; //使能串口2时钟

GPIOA->CRL&=0XFFFF00FF; //PA2\PA3IO状态设置

GPIOA->CRL|=0X00008B00; //IO状态设置

RCC->APB1RSTR|=1<<17; //复位串口2

RCC->APB1RSTR&=~(1<<17); //停止复位

//波特率设置

// USART2->BRR=(pclk1\*1000000)/(bound);// 波特率设置

USART2->BRR=mantissa;

USART2->CR1|=0X200C; //1位停止,无校验位.

//使能接收中断

USART2->CR1|=1<<5; //接收缓冲区非空中断使能

MY\_NVIC\_Init(0,1,USART2\_IRQn,2);//组2

TIM7\_Int\_Init(99,7199); //10ms中断

TIM7->CR1&=~(1<<0); //关闭定时器7

USART2\_RX\_STA=0; //清零

}

//串口2,printf 函数

//确保一次发送数据不超过USART2\_MAX\_SEND\_LEN字节

void u2\_printf(char\* fmt,...)

{

u16 i,j;

va\_list ap;

va\_start(ap,fmt);

vsprintf((char\*)USART2\_TX\_BUF,fmt,ap);

va\_end(ap);

i=strlen((const char\*)USART2\_TX\_BUF); //此次发送数据的长度

for(j=0;j<i;j++) //循环发送数据

{

while((USART2->SR&0X40)==0); //循环发送,直到发送完毕

USART2->DR=USART2\_TX\_BUF[j];

}

}