



韶关学院

SHAOGUAN UNIVERSITY

毕 业 设 计

题目：基于 Linux 的智能停车场收费综合系统设计

姓 名：	林锦源
学 号：	21124061034
二 级 学 院：	智能工程学院
专 业：	自动化
班 级：	2021 级自动化 01 班
指 导 教 师：	陈学华 冯宝祥
起 始 时 间：	2024 年 09 月—2025 年 05 月

学术诚信声明

本人所呈交的毕业论文(设计),是在指导教师的指导下独立完成。研究工作所取得的成果、数据、图片资料均真实可靠。除文中已注明引用的内容外,不包含任何其他人或集体已经发表或撰写过的作品成果。对本论文的研究做出重要贡献的个人和集体,均已在文中以明确的方式标明。

如有使用人工智能辅助工具(如 ChatGPT、文心一言、DeepSeek 等),仅限于文献检索、语法校对、格式优化等辅助性工作,实验数据、调研结果、论文核心论点、研究框架、及结论推导均由本人独立完成,人工智能生成内容不作为论文主体或结论依据。已向指导教师明确说明人工智能工具的具体使用场景及内容范围,并接受学校对 AI 辅助写作的审查要求,可在答辩时对 AI 使用情况作出合理解释。

本人完全意识到本声明的法律结果由本人承担。

本人签名:

林锦源

日期: 2025. 04. 13

基于 Linux 的智能停车场收费综合系统设计

摘 要：随着经济的迅猛发展，城市汽车数量不断增加，导致停车拥挤、停车效率低等一系列问题。本设计基于 Linux 嵌入式平台，构建高效稳定的智能停车场收费系统，实现车辆进出识别、信息管理与自动计费。系统通过车牌识别技术实时监测进出车辆，并借助数据库进行记录与分析，以提升停车场管理的自动化与智能化水平。软件层面采用模块化架构与多线程处理方式，提高运行效率并降低资源消耗；硬件层面选用高可靠性设备，确保系统稳定运行。通过远程监控与数据可视化功能，系统为运营者提供便捷管理模式，并优化用户停车体验，对解决我国城市停车难问题具有一定的应用价值和实践意义。

关键词：Linux；智能停车；车牌识别；自动计费；数据管理

Linux-based Intelligent Parking Lot Charging Integrated System Design

Abstract: With the rapid development of the economy, the number of cars in the city is increasing, resulting in a series of problems such as parking congestion and low parking efficiency. Based on the Linux embedded platform, this design builds an efficient and stable intelligent parking lot charging system to realize vehicle entry and exit identification, information management and automatic billing. The system monitors the incoming and outgoing vehicles in real time through license plate recognition technology, and records and analyzes them with the help of the database, so as to improve the automation and intelligence level of parking lot management. At the software level, a modular architecture and multi-threaded processing are adopted to improve operational efficiency and reduce resource consumption. High-reliability equipment is selected at the hardware level to ensure the stable operation of the system. Through remote monitoring and data

visualization functions, the system provides operators with convenient management mode and optimizes the user's parking experience, which has certain application value and practical significance for solving the problem of parking difficulties in cities in China.

Key words: Linux ; smart parking; license plate recognition; automatic billing; Data management

目 录

1 绪论	1
1.1 研究目的及意义	1
1.2 停车场收费系统的研究和发展现状	2
1.2.1 国内停车场管理系统的发展	2
1.2.2 国外停车场管理系统的发展	2
1.3 发展趋势和挑战	4
1.3.1 发展趋势	4
1.3.2 面临挑战	4
1.4 论文结构及安排	4
2 开发工具及技术简介	6
2.1 QT creator 开发工具	6
2.2 Ubuntu Linux 系统	6
2.3 C++语言	6
2.4 网络协议	7
2.4.1 OSI 和 TCP/IP 模型	7
2.4.2 HTTP 协议	9
2.5 SQL 数据库	10
3 系统分析	10
3.1 可行性分析	10
3.1.1 技术可行性分析	10
3.1.2 经济可行性分析	11
3.2 需求性分析	11
4 系统软件设计	12
4.1 UI 界面设计	12
4.2 摄像头驱动设计	14
4.2.1 V4L2 框架	14
4.2.2 设计流程	15
4.3 后台系统设计	17

4.3.1 车牌的识别	17
4.3.2 车辆的出入库	17
4.3.3 会员的查询	18
4.3.4 查询功能	18
5 系统的硬件设计	18
6 系统测试	20
6.1 管理员的账号登录注册测试	20
6.2 车牌识别测试	21
6.2.1 入库测试	22
6.2.2 出库测试	25
6.3 数据管理模块	27
6.3.1 会员查询	27
6.3.2 停车场内信息查询	29
6.3.3 历史信息信息查询	29
7 总结与展望	31
参考文献	32
致 谢	33

基于 Linux 的智能停车场收费综合系统设计

1 绪论

1.1 研究目的及意义

汽车作为现代出行中不可或缺的交通工具之一，已然成为人们日常生活中不可缺少的部分。随着经济的增长，人们的生活水平提高，我国私家车的保有量不断增长，人们享受着汽车带来的便利，但同时也被停车难、交通拥堵等问题所困扰^[1]。传统的人工收费模式效率低、易出错，且人力成本较高，难以满足当下对停车场管理的高效化与智能化需求。为应对这一挑战，智能停车系统应运而生，通过利用现代信息技术和自动化手段，实现车辆的快速进出识别、自动计费 and 数据分析管理，不仅能有效提升停车场运营效率，也可为车辆使用者提供更加便捷的停车体验^[2]。

在诸多技术路线里，基于 Linux 的嵌入式平台有着可靠性强、可定制化程度较高、资源占用较少、系统稳定性佳的特性，极为适合构建智能停车场收费体系，利用上车牌识别技术，系统可快速、精准地捕捉与记录车辆进出的数据，进而将数据存储在数据库里，供后续开展分析与查询，利用在软件层面采用模块化设计和多线程处理做法，可大幅度提高系统运行效率，同时削减对硬件资源的消耗，硬件层面选用具备高可靠性的嵌入式设备，保证系统在各种环境情形下仍可维持稳定、长期的工作能力。

鉴于前面所说的背景，本论文着重对基于 Linux 的智能停车场收费综合系统开展系统化的设计与实现研究，首先要介绍系统整体的需求以及关键的技术原理，还会对车牌识别算法、数据库管理办法以及系统架构的选择进行详细说明；接着开展软硬件的协同设计与落实，探寻系统的整体搭建路径；最后对系统的功能及性能做验证与测评，还对未来系统优化方向及应用的前景进行展望，依靠本研究，期望为智能停车领域供给具备可扩展性及实用价值的系统解决方法，同时给后续相关研究及实践打下稳固的技术底子。

1.2 停车场收费系统的研究和发展现状

随着城市化进程的加速和车辆数量的不断增加，停车难题已成为许多城市面临的主要挑战。为了提高停车资源的使用效率，现代停车场管理系统得到了迅速发展。国内外停车场管理系统正朝着智能化、自动化、无人化等方向发展，以解决停车资源管理不充分、停车收费不规范等问题。

1.2.1 国内停车场管理系统的发展

(1) 智能化管理

车牌识别技术：国内大部分停车场管理系统的内容中都采用了车牌识别系统，采用车辆牌照识别技术，系统可自行识别驶入停车场的车辆，做到车辆快速出入库和精准收费。

智能停车引导：采用传感器、红外线、摄像头等设备，实时掌握停车位的占用状况，再凭借显示屏、手机 APP，引导车主前往空车位，减少车辆停车时间，增高停车的效率。

自动化收费：以往人工收费已渐渐被自动化收费替代，车主可以依靠 ETC 卡、二维码、支付宝、微信等将停车费结好，省去了排队和找零这些麻烦事儿。

(2) 移动支付的普及

伴着移动支付的普遍采用，国内的停车场管理系统同样主动接入支付宝、微信等第三方支付平台，支持像二维码支付、NFC 支付这种付款手段，不少停车场搞出了专门针对停车业务的 APP，用户可依靠它进行停车预约、支付停车费，甚至还可远程实施停车位监控。

(3) 智慧停车平台

诸多城市跟企业开启了“智慧停车平台”建设项目，采用大数据、云计算这样的技术，将各类停车信息上传到平台上，为用户提供更精准的停车服务，城市停车信息共享平台对多个停车场的数据实施集中管理，为车主供给当下的停车资讯，消除停车面临的难题。

(4) 政策的支持和发展

国内各个地方的政府正积极推进停车场管理系统的建设与变革进程，深

圳、上海等城市正积极推行智能停车系统，还凭借政策及补贴推动停车场智能化设施的搭建，某些区域正打算使用停车位共享模式，通过线上平台来共享停车位，提高停车位的利用效果。

1.2.2 国外停车场管理系统的发展

相比国内，国外的停车场管理系统在智能化和技术应用方面起步较早，尤其是在欧洲、北美等发达国家和地区，停车场管理系统已经实现了较高水平的自动化与智能化。2001 年 Inaba K 和 Shibui M 等人开发了一种驾驶员通过互联网能预订空闲车位的智能停车系统，2015 年 Pham T N 和 Tsai M F 等人^[3]提出了一种为节约停车效率，以最低成本自动找到停车位的智能停车系统新算法，2018 年 Al-Jabi M 和 Sammaneh H 提出了一种基于增强现实(AR)的智能停车系统。

(1) 智能停车解决方案

智能车位检测：国外有的停车场已经实现了高精度车位检测体系，采用地磁、超声波、红外传感器之类的相关技术，实时测定每个车位的使用情形，然后把数据发送至云平台，从而实现了停车资源的精准调度。

动态定价：很多发达国家的停车场管理系统使用动态的收费定价机制，根据停车需求各阶段的变化，调整停车收费价钱，特定区域的停车费会依据时段、天气、事件等方面作出改变，以此实现更恰当的资源分配。

(2) 无人化和自动化

无人停车场：在众多发达国家处，部分停车场已达成无人化管理模式，车主开车驶进停车场后，车辆采用车牌识别或 RFID 技术达成自动登记，把车安顿好之后，车主凭借自动支付系统把费用付掉，整个过程一点儿人工干涉都不存在。

自动化停车设备：自动泊车空间(Automated Parking System),部分欧洲国家已开始应用 APS，车主把车子停靠到指定区域后，系统会自动把车子挪到可用的车位，节约空间且降低人为引发的差错。

(3) 移动支付与 APP 集成

移动支付已是停车场管理的主要付费形式，车主借助手机上的智能 APP，能预订车位并把停车费支付，部分 APP 另外具备车位预订、停车导航以及停

车时长测算等功能，像 ParkMobile、PayByPhone 等应用在北美和欧洲若干城市被大量采用。

（4）物联网与大数据应用

好些国外停车场管理系统采用物联网技术，把停车场里的传感器、摄像头等仪器接入网络，借助对实时数据的收集解析，提供智能化停车服务，这些数据不只是协助停车场优化车位分配，也可针对停车需求方面的变化进行分析，实施资源统筹与动态定价。

（5）共享停车

车主可借助共享平台去租用闲置的停车位，甚至在部分欧美城市，共享停车作为概念，发展相对成熟，能在自家车库处布置公共车位，供旁人借用，这种模式不仅增加了停车位的使用时长占比，还舒缓了城市停车的紧张压力。

1.3 发展趋势和挑战

1.3.1 发展趋势

智能化与自动化:停车场管理系统将进一步迈向智能化，车位检测、付款、引导、收费等各项功能将愈发自动运行，系统将变得越发高效、便利，大数据跟 AI 相关技术:依靠对停车数据的收集与分析，人工智能可较为精准地预估停车需求，调整资源的分配格局，给出贴合个人的专属服务。

新能源车的适配性:跟着电动汽车的逐渐普及，未来停车场管理系统需增设更多电动汽车充电桩，而且需在停车收费上进行合理的优化:停车场管理系统要跟城市其他的基础设施紧密连接，为车主提供更好的服务。

1.3.2 面临挑战

就算停车场管理系统正渐渐迈向智能化，然而不同系统间兼容性与标准化问题依旧未解决，尤其是不同厂商跟不同城市之间技术协同的效果。

用户隐私与数据安全问题：停车场管理系统跟大量用户数据相牵扯，含有支付信息、停车时长这类敏感数据内容，如何让数据安全跟隐私保护得到保障是当下要处理的事情。

建设成本与维护问题：虽说智能停车系统在效率跟用户体验上体现出显著优

势，然而该系统建设成本偏高，而系统后期维护及升级也是一个难题。

1.4 论文结构及安排

主要内容如下：

第1章为绪论。说明研究背景及意义，其次是对停车场收费系统的研究发展现状分析，作后明确本文所要完成的工作与章节安排。

第2章为智能停车场收费综合系统的设计方案。本章先对改系统进行实现原理的分析，接着对功能进行概括分析，最后是对总体方案的概括。

第3章为需求分析。可行性分析和系统需求分析两部分组成。

第4章为系统软件设计。ui 界面设计、摄像头驱动设计、后台系统的设计三部分组成。

第5章为系统硬件设计。主要介绍所用开发板的信息。

第6章系统测试。测试系统相关功能并进行结果分析，其中对网络传输、数据库进行测试分析。

第7章总结与展望。对本课题研究的系统进行综合评价，指出存在的缺点和需要改进的地方，并展望未来的工作方向。

2 开发工具及技术简介

2.1 QT creator 开发工具

QT Creator 是专门用来开发跨平台应用的官方工具，支持 Windows、macOS、Linux 甚至手机系统。用它写代码主要用 C++ 和 QML 这两种语言，尤其做界面特别方便——自带可视化设计器，拖拽就能搭界面，还能搞动画特效。开发时自动补全代码、调试报错这些基础功能都有，还能一键把项目打包到不同平台运行。无论是做桌面软件、手机 App，还是车载系统、工业控制之类的硬核场景都能搞定。它本身是免费的，开源和商业项目都能用，特别适合既要酷炫界面又要省心适配多端的项目！

2.2 Ubuntu Linux 系统

Linux 是一个免费使用和自由传播的类 Unix 操作系统，是由世界各地的成千上万的程序员设计和实现的，其目的是建立不受任何商品化软件的版权制约的、全世界都能自由使用的 Unix 兼容产品^[5]。Ubuntu Linux 作为自由开源的数字先锋，以用户为核心关注点的操作系统榜样，它不只是技术爱好者的得力器物，更是催化创新的通用平台，以 Debian 为根基深度优化，Ubuntu 凭借稳定性与兼容性闻名，不管是日常办公事宜、编程开发事宜，还是企业级服务器的部署，都能轻松办妥。默认搭配的 GNOME 桌面环境简洁灵秀，软件中心备有数千款即用的应用程序，新手也能迅速上手操作，其 LTS 长期支持版本（如 Ubuntu 24.04 LTS 之类）供给五年的安全更新，为关键业务的安全护航，从树莓派做物联网开发直至 Kubernetes 集群的云端管理，Ubuntu 依靠轻量高效的特点赋能万物互联，全球百万开发者社区一直奉献智慧，依靠 snap 包生态让应用实现秒级部署，选上 Ubuntu，就是选择开放、安全和未来，不管你是钻研代码的编程极客，也或是看重效率的职场打工人，它都可充当你释放创造力的良好伙伴。

2.3 C++语言

C++是工业级编程的基石，以高性能与灵活性著称的多范式语言。自 1983

年由 Bjarne Stroustrup 在 C 语言基础上革命性地引入面向对象特性以来，历经 C++11 至 C++23 的持续进化，始终稳居系统级开发的核心地位。它既能像汇编般直接操控硬件资源，又支持封装继承多态构建复杂系统，更以模板元编程实现极致代码复用。从操作系统内核到 3A 级游戏引擎，从自动驾驶算法到量子计算模拟，C++ 凭借零抽象开销的执行效率和硬件无关性，成为驱动现代科技的隐形引擎。其标准库涵盖容器算法、并发模型等工业级组件，既适合编写百万行级商业软件，也能满足嵌入式设备的严苛资源限制。对于追求性能极限的开发者而言，C++ 不仅是编程语言，更是掌控计算本质的终极工具。

2.4 网络协议

2.4.1 OSI 和 TCP/IP 模型

OSI 模型是国际标准化组织（ISO）为解决早期异构网络设备互不兼容问题而提出的七层分层参考模型，通过将网络功能划分为物理层、数据链路层、网络层、传输层、会话层、表示层、应用层，实现了模块化设计和标准化接口，确保不同厂商设备可通过统一规则通信。其核心价值在于简化网络设计、提升兼容性与可维护性，虽未全面应用，但分层思想成为 TCP/IP 协议及现代网络架构的基础，是理解计算机网络的通用框架。

网络分层模型



图 2-1 OSI 七层模型

OSI 参考模型把网络通信流程划成了七个层次,各层均承担特定的通信方面工作,并且借助标准接口为相邻层供给服务,物理层(Physical Layer)肩负着把数字信号转化为模拟信号的活儿,实现比特流在物理介质里的无干扰传送,就如同把二进制数字信号转变成电流、光信号这类物理信号。物理层的主要成效是为协议给予可靠的物理传输渠道,数据链路层(Data Link Layer)承担着对数据开展分段、差错检测及纠正等方面处理的工作,以此让数据在物理链路上实现可靠传输,数据链路层也担负着物理地址(MAC 地址)识别与管理的相关工作,让不同计算机可依靠数据链路层达成通信。网络层(Network Layer)担当起在网络里选择最优传输路径的工作,进而助力数据在不同网络之间实现传输,网络层同样承担着逻辑地址(IP 地址)的识别与管理事项,让计算机有条件在互联网的网络空间里进行通信活动,传输层(Transport Layer)承担起实现端到端通信的工作,给予数据分段、流量疏导以及差错检查等功能。传输层主要通过两种协议:传输控制协议(TCP)和用户数据报协议(UDP)来实现通信^[6]。会话层(Session Layer)负责建立、维护和终止会话在通信会话中的顺序传送和完整性,并处理会话恢复及故障处理,保障通信的持续性和可靠性。表示层(Presentation Layer)负责数据的格式转换、加密解密、数据压缩和数据格式标准化,确保数据在不同系统之间的互操作性,使得数据能够被正确解释和处理。应用层(Application Layer)则直接为用户提供各种网络应用服务,如电子邮件、文件传输、远程登录、Web 浏览等,用户通过不同的应用程序访问这些服务^[7]。

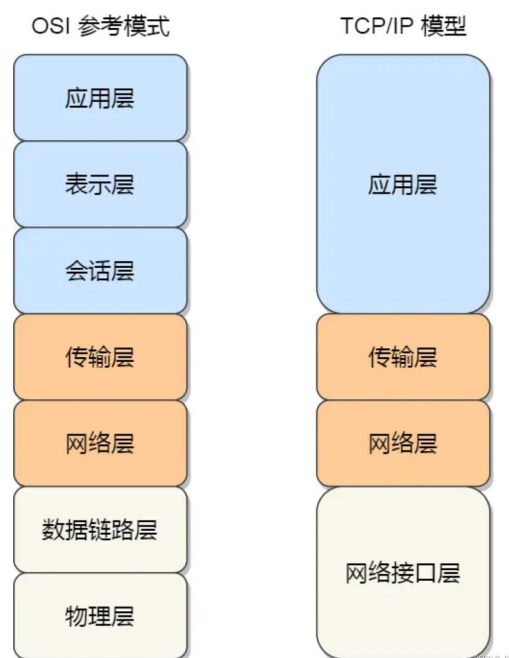


图 2-2 参考模型与 TCP/IP 模型对比图

TCP/IP 模型在层级体系上借鉴了 OSI 模型，但做了初步简化，把七层模型裁减为五层，网络接口层要把数据封装成帧，随后经物理链路进行传输，网络层的主要工作是把数据包从源地址输送到目标地址，进而选定最佳的路由，保障主机彼此间的有效通信。传输层依靠错误检查、数据分段、流量控制与拥塞控制等手段，维持数据在端到端输送过程的可靠性，应用层（Application Layer）掌管着应用程序之间的数据交换事务，虽然各个应用层协议一般针对特定的应用程序，但它们都不得不遵循 TCP/IP 协议栈规定的标准格式。

2.4.2 HTTP 协议

超文本传输协议(HTTP)是一个基于请求与响应模式的、无状态的、应用层的协议,常基于 TCP 的连接方式^[8]。其核心作用是定义浏览器（客户端）与 Web 服务器之间的通信规则，支持文本、图片、视频等资源的传输，是万维网（WWW）的基础。

原理：客户端通过 TCP 连接（默认端口 80）向服务器发送请求（如 GET 获取资源、POST 提交数据），请求包含路径、headers 等信息；服务器解析请求后返回响应，包含状态码（如 200 成功、404 未找到）、响应头和内容（如 HTML 页面）。HTTP 是无状态的，即每次请求独立，依赖 Cookie、

Session 等技术维持会话。最新版本 HTTP/2 和 HTTP/3 通过多路复用、加密（HTTPS）等优化了性能和安全性。

2.5 SQL 数据库

SQL Server 作为一款功能强大的关系型数据库管理系统，不仅提供了高效的数据存储和查询功能，还有了丰富的数据挖掘算法和工具^[9]。其作用包括：

数据存储与管理：结构化存储海量数据，支持高效的增删改查（CRUD）操作。

数据一致性：采用事务处理让操作具有原子性、一致性、隔离性和持久性；

复杂查询：凭借 SQL 语句完成过滤、排序、聚合、多表关联等复杂查询事宜。

应用支撑：为 Web 系统、企业软件、数据分析等提供底层的数据后盾，多见于用户信息、订单管理、业务流程等相关场景。

核心的优势是结构化组织、采用标准化操作（SQL）以及强数据约束，是传统数据管理的主流途径。

3 系统分析

3.1 可行性分析

3.1.1 技术可行性分析

本智能停车场收费管理系统以 Linux Ubuntu 操作系统和 QT Creator 为开发平台。借助 QT 软件强大的界面设计功能，精心搭建程序交互界面，为用户提供直观、便捷的操作体验。硬件方面采用 GEC6818 平台，确保系统具备稳定且高效的运行性能。

在数据存储与管理上，选用 SQLite3 数据库。它表现出轻量化、易集成的特点，出色地符合系统在数据存储及查询上的需求，确保数据处理既安全又高效。

系统以 C++ 编程语言和 QT 软件为核心进行设计开发。运用 C++ 编写系统后台逻辑和各个功能模块的实现接口，充分发挥其强大的性能和丰富的表达能力。同时，QT 软件的跨平台兼容性，使得系统可以在多种不同的操作系统和硬件环境下稳定运行，极大地拓展了系统的应用范围。

系统充分利用起软件自带的标准模板库。这一举措有效压缩了编码的工作分量，还把调试的难度给降低了，让开发的过程变得高效又顺畅，有益于系统各项功能迅速达成。

3.1.2 经济可行性分析

基于 Linux 开发的智能停车场收费系统在软件授权成本上天然占优，搭配摄像头、道闸等硬件一次性的投入和远程运维的低人力开支，可以让整体成本降低，从而提高车辆识别跟收费的效率，停车收入实现上升，减少了人工的相关费用，通常来说 1 - 2 年即可收回主要投入资金，同时还可增加广告、会员管理的业务，实现长期稳定的利益，所以经济上的可行性十分显著。

3.2 需求性分析

智能停车场收费综合系统的核心需求源于传统停车场效率低下、用户体验差、管理成本高的痛点，需要实现入场自动化（车牌识别）、停车引导智能化（路径规划）、缴费无感化（移动支付）及数据管理精细化（动态定价、设备监控），同时需满足跨平台兼容性、轻量级部署和扩展性设计，以适配多硬件架构与未来技术演进。该系统还承载智慧城市交通治理需求，包括缓解拥堵、降低碳排放、对接城市停车云平台，以及提升民生服务（无障碍支持、应急管理），兼具商业价值与社会价值。

4 系统软件设计

系统软件设计分为 UI 界面设计、摄像头驱动设计、后台系统的设计三大模块。

4.1 UI 界面设计

UI 设计界面是在 QT 软件上实现。Ui 设计界面分为登录注册界面，后台系统界面，会员界面，查询功能界面。界面通过使用 QLabel、QPushButton、QMessageBox 等控件进行设计，界面如图 3-1、图 3-2、图 3-3、图 3-4 所示：

注册账号:

密码:

再次输入密码:

确认 取消

图 4-1 登录注册界面

智能停车收费管理系统

停车场信息

场内停车位: 剩余停车位:

闸门状态:

停车费用: 5元/小时(非会员) 3元/小时(会员)

本次计费

进场时间:

离场时间:

停车时长: 小时

应缴费用: 元

打开闸门 关闭闸门 会员查询 车辆入库 车辆出库 查询 退出登录

图 4-2 后台系统界面



图 4-3 会员查询界面



图 4-4 查询功能界面

界面的跳转是通过信号与槽函数实现，信号和槽一起构成了一个强有力的组件编程机制，在该机制下，不仅可以多个信号与单一槽相连，也可以将单个信号与多个槽相连。例如，使用 QT 编写一个窗口，窗口中只有一个按钮。当单击按钮时，会产生关闭窗口结果。这个就是 QT 中的信号与槽机制^[10]。

4.2 摄像头驱动设计

图像采集技术是利用摄像头实时获取图像信息的手段^[11]。而摄像头驱动利用 V4L2（Video4Linux2）接口进行图像数据获取，结合百度的车牌识别接口进行车牌识别。

4.2.1 V4L2 框架

V4L2（Video4Linux2）是 Linux 操作系统下针对视频设备（如摄像头、电视接收卡等）提供的内核框架和编程接口。它是早期 V4L（Video4Linux）的升级版，旨在为各种视频采集、视频输出和图像处理设备提供统一的驱动接口与用户空间访问机制。在内核中，V4L2 驱动会向用户空间呈现类似 `/dev/video0`、`/dev/video1` 等字符设备文件，应用程序可用一系列 `ioctl` 命令（如 `VIDIOC_QUERYCAP`, `VIDIOC_S_FMT`, `VIDIOC_REQBUFS` 等）查询设备信息、设置视频格式、分配并管理缓冲区以及控制视频流的开启/停止。视频采集时，常通过 `mmap` 将内核分配的缓冲区映射到用户空间，减少数据拷贝并提升效率。V4L2 在可扩展性方面也表现优异，支持多种像素格式、分辨率及采集模式，还能与硬件编解码器协作（即 V4L2 M2M 模式）。通过与 GStreamer、FFmpeg 等多媒体框架结合，V4L2 成为 Linux 下视频采集与处理的核心基础，为摄像头应用、机顶盒、视频会议等提供了统一的开发接口。

4.2.2 设计流程

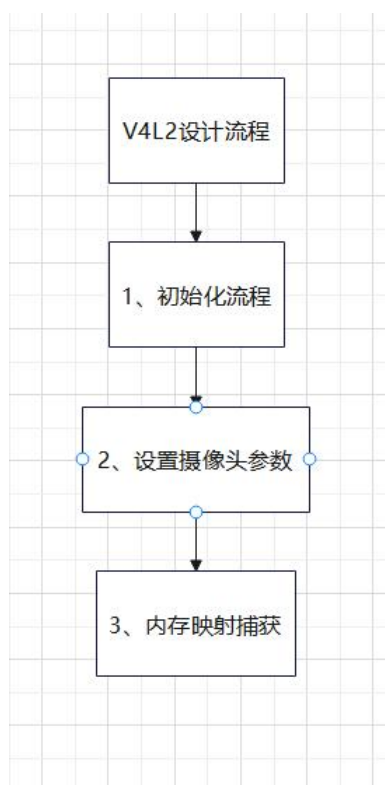


图 4-5 摄像头设计流程

一、初始化流程

首先要打开设备、后要查询设备是否支持视频捕获

```
//1、打开摄像头驱动文件
fd = open(deviceName,O_RDWR);
if(fd== -1)
{
    perror("open video error");
    return ;
}
```

图 4-6 初始化流程

二、配置参数

配置参数分为设置图像格式和设置帧率。

```

//3、设置当前摄像头的参数（拍摄画面的宽度、高度、以及拍
//#define VIDIIOC_S_FMT _IOWR('V', 5, st
struct v4l2_format vfmt; //variable has in
vfmt.type = V4L2_BUF_TYPE_VIDEO_CAPTURE; //视频
vfmt.fmt.pix.width = 640;
vfmt.fmt.pix.height = 480;
vfmt.fmt.pix.pixelformat = V4L2_PIX_FMT_YUVV;

int ret = ioctl(fd,VIDIIOC_S_FMT,&vfmt);
if(ret == -1)
{
    perror("VIDIIOC_S_FMT 失败");
    return ;
}

```

图 4-7 设置图像格式

```

m_camera = new CameraV4l2;
m_camera->start();
m_timer->start(50); //1秒钟 帧率20 1秒钟20帧画面 FPS

```

图 4-8 设置图像帧率

三、内存映射捕获

内存映射捕获分为申请缓冲区、映射缓冲区到用户空间。

```

//4、在内核空间中申请4块缓冲区存储拍摄的画面，4块缓冲区组成缓冲区队列
//#define VIDIIOC_REQBUFS _IOWR('V', 8, struct v4l2

struct v4l2_requestbuffers requestbuffer; //variable k
requestbuffer.count = 4;
requestbuffer.type = V4L2_BUF_TYPE_VIDEO_CAPTURE; //use
requestbuffer.memory = V4L2_MEMORY_MMAP; //use of unde

ret = ioctl(fd,VIDIIOC_REQBUFS,&requestbuffer); //use c
if(ret == -1)
{
    perror("VIDIIOC_REQBUFS 失败");
    return ;
}

```

图 4-9 申请缓存区

```

}
//5、将这4块缓冲区（队列）映射到用户空间
struct v4l2_buffer mapbuffer;
for(int i=0; i<4; i++)
{
    //5.1) 向内核查询你要映射哪一块缓冲区，获取该缓冲区的信息 VIDIOC_QUERYBUF
    //所谓的信息：这块缓冲区的起始地址、缓冲区的大小
    //define VIDIOC_QUERYBUF _IOWR('V', 9, struct v4l2_buffer)
    mapbuffer.index = i;
    mapbuffer.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
    ret = ioctl(fd,VIDIOC_QUERYBUF,&mapbuffer);
    if(ret == -1)
    {
        perror("VIDIOC_QUERYBUF 失败");
        return ;
    }
    //5.2) 进行内存映射 mmap
    mptr[i] = (unsigned char*)mmap(NULL, mapbuffer.length,PROT_READ|PROT_WRITE,MAP_SHARED,fd,0);
    if(mptr[i] == MAP_FAILED)
    {
        perror("mmap error");
        break;
    }

    size[i] = mapbuffer.length;

    //5.3)通知内核操作完成，下一次查询的时候会从下一块缓冲区开始 VIDIOC_QBUF
    //define VIDIOC_QBUF _IOWR('V', 15, struct v4l2_buffer)
    ret = ioctl(fd,VIDIOC_QBUF,&mapbuffer);
    if(ret == -1)
    {
        perror("VIDIOC_QBUF 失败");
        return ;
    }
}
}

```

图 4-10 映射到用户空间

4.3 后台系统设计

后台系统设计分为车牌的识别、车辆的出入库、会员查询、查询功能这四大方面来讲述。

4.3.1 车牌的识别

车牌的识别是由摄像头捕获到的照片通过 tcp 协议传到后台系统，之后通过 http 协议连接到百度 AI 的车牌识别接口进行车牌的识别。

4.3.2 车辆的出入库

车牌识别完成之后，系统会先判断停车场内是否有该车辆，如果有该车辆就会提示车辆已在库中，若没有则提示入库成功。这时相应的车牌信息会录入到数据库中。

当车辆出库时，也是调用车牌识别的接口进行车牌的识别，如果停车场

内没有该车辆就显示该车不在停车场内，反之则提示出库成功。出库完成之后会显示入库时间、出库时间、以及停车的时长和费用。

4.3.3 会员的查询

一般入库成功之后都是非会员，如果想要成为会员就可以进行会员的注册。注册会员要提供姓名、手机号、车牌号还有要充值的金额。注册完成之后就成为了会员。会员和非会员的收费是不同的，非会员是每小时 5 元，会员是每小时 3 元。当你车辆出库成功之后可以在当前账户的余额进行扣款。

4.3.4 查询功能

查询功能包括对当前停车场内的车辆信息进行查询，还有是历史记录查询。

5 系统的硬件设计

GEC6818 开发板是一款高性能嵌入式平台，搭载三星 S5P6818 八核 Cortex-A53 处理器(主频 1.4GHz)，配备 1GB DDR3 内存和 8GB eMMC 存储，支持 Android 5.1、Linux 等多系统运行。其硬件集成千兆以太网、USB、HDMI、MIPI 等接口，支持 WiFi / 蓝牙无线通信，并提供丰富的 GPIO、I2C、SPI 等外设接口，适配传感器、触摸屏、摄像头等设备。该开发板以多媒体交互为核心，整合了高清视频播放、语音控制、相册管理及游戏娱乐功能，广泛应用于智能家居（支持家电语音控制）、智能车载（集成导航与娱乐系统）、数字广告（支持图片 / 视频轮播）及工业控制等场景。开发资源方面，提供 Linux 交叉编译工具链、示例代码及开源项目，适合嵌入式开发教学、原型设计及产品化应用，兼具性能稳定性与功能扩展性。如图所示：



图 5-1GEC6818 开发板

6 系统测试

6.1 管理员的账号登录注册测试

本次测试旨在全面验证工作人员账号登录注册功能的正确性、稳定性和安全性。此功能是为了管理员能使用后台系统。登录注册流程图、管理员注册界面图、管理员登录界面图以及相关代码图如图 6-1，图 6-2，图 6-3，图 6-4 所示。

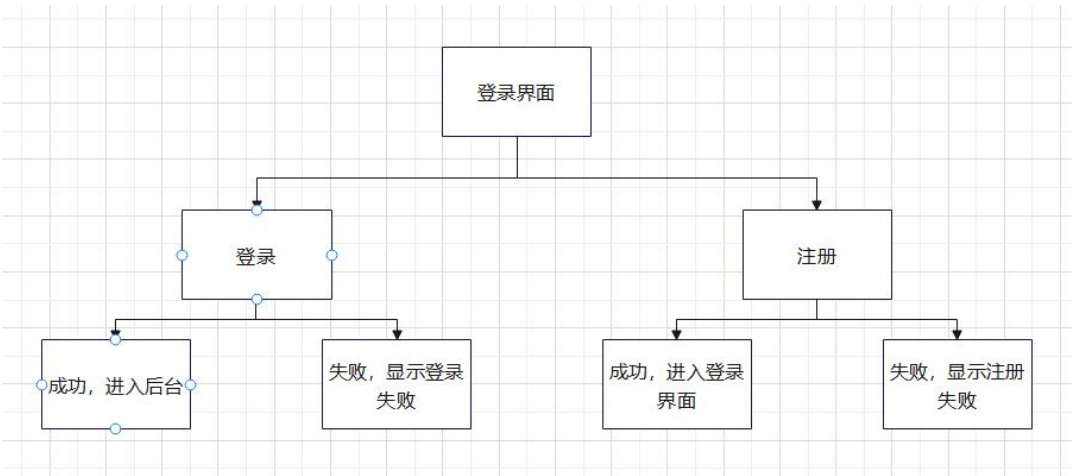


图 6-1 登录注册流程图

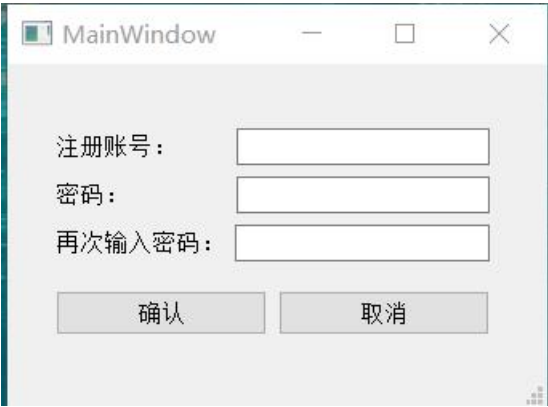


图 6-2 管理员注册界面

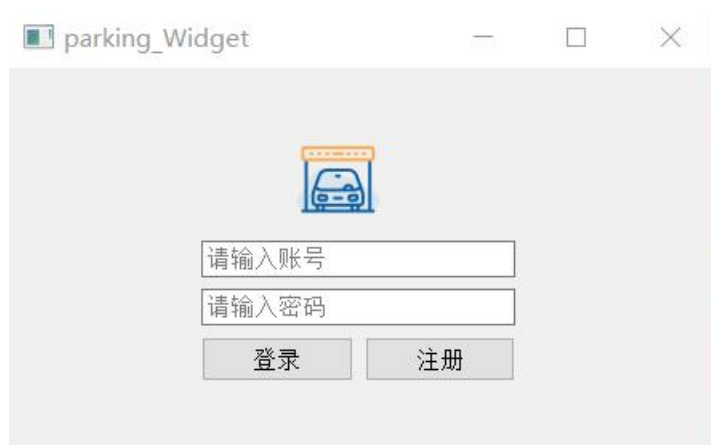


图 6-3 管理员登录界面

```
#if 1
//检查数据库中是否有相同的账号
QString query(loindb1);
query.prepare("SELECT * FROM account WHERE \"账号\" = :account");
query.bindValue(":account", ui->acEdit->text());

// 执行查询
if (!query.exec()) {
    qDebug() << "查询执行失败:" << query.lastError().text();
    return;
}

// 检查是否有匹配的记录
if (query.next()) {
    // 找到了匹配的账号
    QMessageBox::information(this, "提示", "账号已存在");
    qDebug() << "账号已存在";
    // 可以在这里添加相应的处理逻辑，如提示用户等
} else {
    // 没有找到匹配的账号
    qDebug() << "账号不存在，可以注册";
    // 可以在这里添加注册新账号的逻辑
}
```

图 6-4 相关代码图

注册登录功能主要由信号与槽函数进行登录还是注册的跳转。无论是登录还是注册都会通过 SQL 的 SELECT 语句执行比对功能，与数据库中的账号密码进行比对，当比对正确之后，就会完成登录还是注册，登录成功，之后进入后台系统。注册成功就要进行登录，在进入后台系统。

6.2 车牌识别测试

车牌识别测试是为了摄像头能正确捕获车牌，并将数据上传到后台系统，通过链接网站完成车牌识别。

6.2.1 入库测试

车牌的入库是通过 http 协议与百度 AI 的车牌识别接口进行连接，从而使用该接口进行车牌识别的功能，该功能会对车牌进行捕获，之后通过 CJSON 将车牌获取，并打印在后台系统中，车牌获取成功就会跟停车场车辆数据库中的车辆信息进行比对，如果没有的话，就会把该车牌写入数据库中，还会跟会员的数据库中进行对比，判断是否为会员，也会显示在后台系统中。如果识别成功就会弹出“入库成功”的提示框。

车牌无论是正面获取还是侧面获取都是可以的。流程及结果如图 6-5，图 6-6，图 6-7，图 6-8 所示。

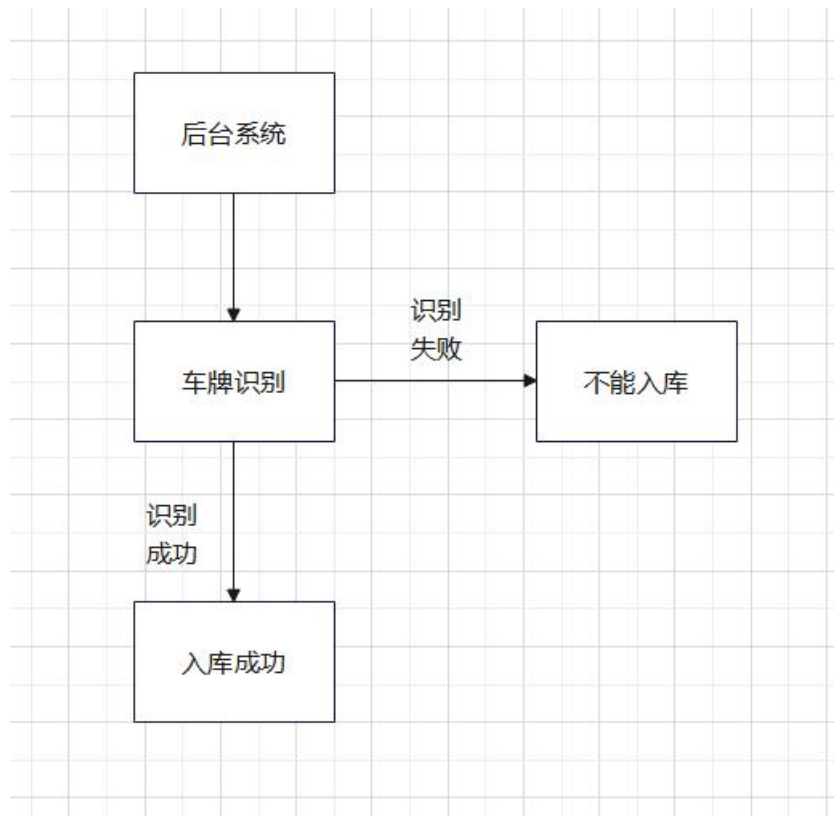


图 6-5 车辆入库流程

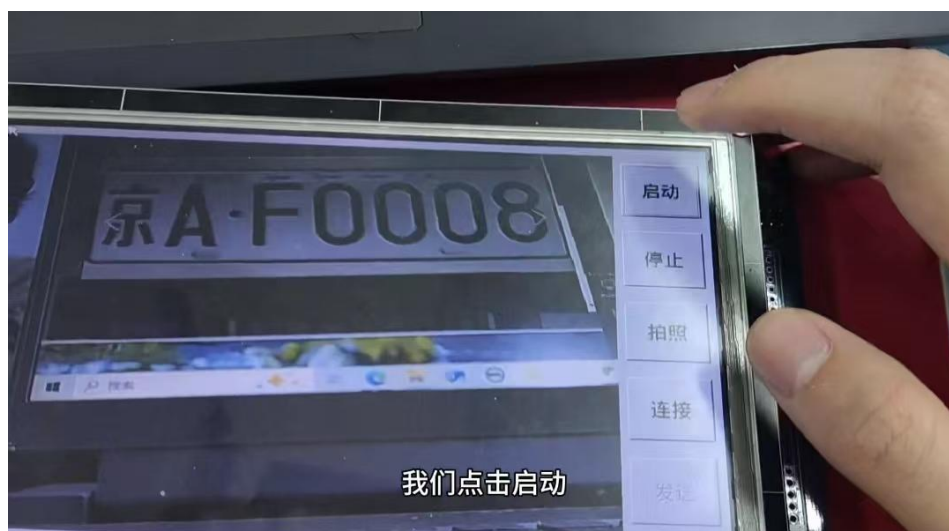


图 6-6 车牌的获取



图 6-7 车辆入库正面获取车牌



图 6-8 车辆入库侧面获取车牌

```

//1检查Car数据库中有没有该车 //负责记录停车场内有多少车
// 准备查询语句
QSqlQuery query(db);
query.prepare("SELECT * FROM Car WHERE \"车牌号\" = :num");
query.bindValue(":num", ui->label_name->text());

// 执行查询
if (!query.exec()) {
    qDebug() << "查询执行失败:" << query.lastError().text();
} else {
    if (query.next()) {
        QMessageBox::information(this, "提示", "车辆已在库中");
        qDebug() << "车牌号已存在于数据库中";
    } else {
        qDebug() << "车牌号不存在, 可进行插入操作";
        // 这里可以添加插入车牌号的代码
        QSqlQuery insertQuery(db);
        insertQuery.prepare("INSERT INTO Car (\"车牌号\", \"入场时间\", \"应缴费用\") VALUES (:num, :intime, :money)");
        insertQuery.bindValue(":num", ui->label_name->text());
        insertQuery.bindValue(":intime", ui->time_label->text());
        insertQuery.bindValue(":money", 0);
        if (insertQuery.exec()) {
            QSqlDatabase::database().commit(); // 提交事务

            QSqlQuery countQuery(db);
            countQuery.prepare("SELECT COUNT(*) FROM Car");
            if (countQuery.exec()) {
                if (countQuery.next()) {
                    resnum = countQuery.value(0).toInt(); // 使用 countQuery 获取行数
                    qDebug() << "表中的数据行数" << resnum;
                    //显示剩余停车位
                    ui->label_4->setText(QString("%1").arg(ui->label_2->text().toInt() - resnum));

                    //显示闸门的状态
                    ui->label_6->setText("已打开");
                    ui->label_6->setStyleSheet("color: green;");

                    // 使用 QTimer 在 5 秒后将 label_6 字体颜色改为红色
                    QTimer::singleShot(5000, [this]() {
                        ui->label_6->setText("已关闭");
                        ui->label_6->setStyleSheet("color: red;");
                    });
                }
            }
        }
    }
}

```

图 6-9 车辆入库部分代码

通过测试，无论车牌是侧面还是正面都是可以获取成功的，最终也会记录在数据库中。该测试是成功的。

6.2.2 出库测试

出库与入库的功能差不多。先是在停车场场内的数据库中进行查找该车辆信息，来判断该车是否在停车场内，如果查询到该车牌就会进行出库处理，并且出库完成后，会将在停车场内的数据库中删除相应的车辆信息，防止该车多次出库。之后还会更新数据库中的出库时间，并进行缴费处理。

车牌识别测试是为了摄像头能正确捕获车牌，并将数据上传到后台系统，通过链接网站完成车牌识别。车牌无论是正面获取还是侧面获取都是可以的。出库流程及结果如图 6-10，图 6-11，图 6-12，图 6-13 所示。

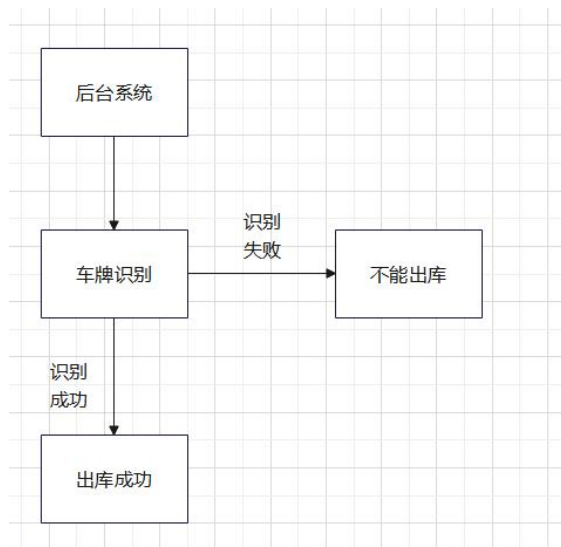


图 6-10 出库流程图



图 6-11 出库成功 1



图 6-12 出库成功 2

```

// 1. 查询 Record 表中该车牌号的最新记录，按入库时间排序
QSqlQuery queryCheck(Recorddb);
queryCheck.prepare("SELECT \"入库时间\", \"出库时间\" FROM Record WHERE \"车牌号\" = :num ORDER BY \"入库时间\" DESC LIMIT 1");
queryCheck.bindValue(":num", ui->label_name->text()); // 绑定车牌号

if (queryCheck.exec()) {
    if (queryCheck.next()) {
        QString inTime = queryCheck.value(0).toString(); // 获取入库时间
        QString outTime = queryCheck.value(1).toString(); // 获取出库时间

        // 2. 如果出库时间为空，则更新出库时间
        if (outTime.isEmpty()) {
            QSqlQuery updateQuery(Recorddb);
            updateQuery.prepare("UPDATE Record SET \"出库时间\" = :outtime, \"已缴费用\" = :money WHERE \"车牌号\" = :num AND \"入库时间\" = :intime");
            updateQuery.bindValue(":outtime", ui->time_label->text()); // 设置出库时间
            updateQuery.bindValue(":money", money); // 设置已缴费用
            updateQuery.bindValue(":num", ui->label_name->text()); // 设置车牌号
            updateQuery.bindValue(":intime", inTime); // 确保更新的是最新入库记录

            // 执行更新操作
            if (updateQuery.exec()) {
                qDebug() << "出库时间更新成功";
            } else {
                qDebug() << "更新失败: " << updateQuery.lastError().text();
            }
        } else {
            // 如果出库时间已存在，跳过更新
            qDebug() << "出库时间已存在，无需更新";
        }
    } else {
        qDebug() << "没有找到车牌号: " << ui->label_name->text() << " 的记录";
    }
} else {
    qDebug() << "查询失败: " << queryCheck.lastError().text();
}

```

图 6-13 出库部分代码展示

经过测试，该功能正常，能够正确的识别车牌并且完成出库操作和费用结算。

6.3 数据管理模块

数据管理模块分为会员查询、停车场内信息查询、历史信息查询三部分。

6.3.1 会员查询

当点击查询按钮时，此时会通过 SQL 中的 SELECT 语句遍历会员数据库中的数据，会将人员的信息打印出来，其中会显示会员或者非会员。如果想变成会员，需要填写姓名，电话号，车牌号已经要充值金额之后，就能成功会员，会员的停车价钱比非会员便宜。如果要删除对应的信息只需要填写电话号码进行删除即可。



图 6-14 会员查询

```

/*
 *
 *查询入库车辆数量
 *
 */
void find_window::on_pushButton_3_clicked()
{
    #if 1
        QSqlQuery query(Cardb);

        query.prepare("SELECT *FROM Car");

        if(!query.exec())
        {
            qDebug() << "查询执行失败:" << query.lastError().text();
            return;
        }
        // 使用 QSqlQueryModel 显示查询结果
        QSqlQueryModel *model = new QSqlQueryModel();
        model->setQuery(query);

        // 检查查询是否有错误
        if (model->lastError().isValid()) {
            qDebug() << "查询模型设置失败:" << model->lastError().text();
            return;
        }

        // 将模型设置到 tableView 中
        ui->tableView->setModel(model);
    #endif
}

```

图 6-15 会员查询代码

经过测试，该功能可以正常运行，能够实现相对应的功能。

6.3.2 停车场内信息查询

当点击查询功能时，会显示出当前停车场内的所有车辆的信息，包括车牌号、入库时间、出库时间、应缴费用。查询车辆信息，可以通过输入关键字就可查询相应包含关键字的车辆信息。



图 6-15 停车场内信息查询

6.3.3 历史信息查询

当点击历史按键时，会将历史信息的数据库中的数据全部打印在该界面中，会显示出入库时间，以及缴纳的费用。查询车牌信息，只需输入相关车牌号即可查询到相关的车辆信息。删除车辆信息，也是输入相关的车牌信息就可删除信息。

MainWindow

	车牌号	入库时间	出库时间	已缴费用
1	京AF0236	2025-03-19 20:21:02		
2	甘ADV868	2025-03-20 11:05:37	2025-03-20 11:14:43	0
3	豫U88888	2025-03-20 14:42:40	2025-03-20 14:42:43	0.00416667
4	豫U88888	2025-03-20 14:42:45	2025-03-20 14:42:52	0.00972222
5	豫U88888	2025-03-20 14:46:02	2025-03-20 14:46:19	0.0236111
6	豫U88888	2025-03-20 14:47:50	2025-03-20 14:49:26	0.08
7	京AF0008	2025-03-20 15:24:47	2025-03-20 15:26:12	0.118056
8	京AF0008	2025-03-20 15:28:21	2025-03-20 15:29:57	0.0816667
9	京AF0008	2025-03-20 18:39:08		0
10	京AF0008	2025-03-20 18:39:09		0
11	京AF0008	2025-03-20 18:39:43	2025-03-20 18:39:46	0.00416667
12	京AF0008	2025-03-20 18:41:56	2025-03-20 18:43:07	0.0583333

图 6-16 历史信息查询

7 总结与展望

本系统是基于 C++、QT 软件、网络协议以及 SQL 数据库进行开发。在系统方面分为了硬件和软件两部分。硬件方面为 GEC6818 开发板，以及摄像头一个。软件方面设计了后台系统和识别系统。

后台系统分为管理员的登录注册功能，车辆的出入库功能，会员查询以及信息查询功能。管理员登录注册本功能是为了让管理员使用后台系统做准备，注册的信息都会存入到一个叫 `account` 的 SQL 数据库中，确保账号不会重复。车辆的出入库功能，主要使用了 `Http` 协议调用百度 AI 的车牌识别接口来进行车牌的识别，每次入库都会记录入库时间，车牌号到数据库，还会进行是否为会员的检测。车辆出库，也是使用百度 AI 接口进行车牌检测，出库会记录出库时间，需要缴纳的费用，在后台系统上会显示出入库的时间还有费用，并且会将停车记录保存在 `Record` 的数据库中。会员查询的功能，包括了会员注册，充值缴费，显示是否为会员。该功能会根据手机号进行金额的充值，根据车牌号来判断是否为会员。查询功能包括对当前停车场内的车辆信息查询，以及历史信息的查询。

总体来说，该系统完成基本功能，但是仍缺乏一些实用性，对于车牌识别来说可以使用深度学习算法在车牌识别中的应用将持续深化。通过构建海量且多样化的车牌图像数据集对模型进行训练，使系统能够在诸如恶劣天气（暴雨、大雪、浓雾）、低光照、车牌污损等复杂场景下，依旧保持极高的车牌识别准确率，几乎实现零误判。

还可以将相关信息以及对应车牌图片用时间的形式来保存到数据库中，并且传输到服务器中。对于收费功能来说不够完善。后期可以与物联网结合达到通过无线通信技术接入网络。管理者可以在远程管理中心实时查看设备的运行状态，一旦设备出现故障或异常，系统立即自动报警，并推送详细故障信息，方便快速维修。还可以城市停车管理平台能够实时汇总全市停车场的车位信息，通过路边电子显示屏、手机应用等多种渠道向市民发布，引导车辆快速找到空闲车位，提高城市停车资源的整体利用率。

参考文献

- [1] 张立立,张振坤,刘明钊,沙毅.基于深度学习技术的停车场收费管理系统设计与实现[J].电子器件,2021,44(01):231-235.
- [2] 余松科.自助式智能停车场泊车系统研究[D].成都理工大学,2016.
- [3] 李月,王善勤,肖宏飞.基于物联网的智能停车系统设计[J].滁州职业技术学院学报,2024,23(04):58-62.
- [4] 贾仕俊,秦开宇,陈华伟,李志强.基于QT的数据图形化设计与分析[J].软件导刊,2011,10(07):156-158.
- [5] 周畅.Linux和Qt GUI工具包在故障诊断仪开发中的研究[D].武汉科技大学,2012.
- [6] 高毅.网络环境下嵌入式多媒体实现的研究[D]. 江南大学, 2011.
- [7] 魏煜康.嵌入式SMP环境下的TCP/IP协议栈并行优化研究与实现[D].北京邮电大学,2024.
- [8] 祝瑞,车敏.基于HTTP协议的服务器程序分析[J].现代电子技术,2012,35(04):117-119+122.
- [9] 赵跃.基于SQL Server的数据挖掘技术在电子商务中的应用与比较研究[J].电子元件与信息技术,2024,8(05):95-97+100.
- [10] 何建仓,侯泽民.嵌入式Qt环境下绘图软件的设计与实现[J].软件,2014,35(05):45-47.
- [11] 任天乐.基于机器视觉的农业机器人自主作业研究[J].南方农机,2020,51(16):26-27.

致 谢

时光荏苒，行文至此，我的大学生活也即将画上句点。值此论文完成之际，谨向所有关心、帮助过我的师长、同学和家人致以最诚挚的感谢！

首先，衷心感谢我的导师陈兴华老师。从论文选题到研究框架，从代码调试到成果完善，陈兴华老师始终以严谨的治学态度和深厚的学术素养为我指明方向。每当我在研究中陷入困境时，陈兴华老师总能以精准的点评和耐心的指导让我豁然开朗。师者风范，受益终身。

感谢智能工程学院的各位任课老师，他们在专业课程中传授的知识为我的研究奠定了坚实基础。

最后，感恩我的父母和家人。无论何时，你们的理解与支持都是我前行的动力。愿以今日之成果，回报你们长久以来的付出。

毕业在即，我将带着师长的教诲、同窗的情谊和家人的期待，继续在人生道路上探索前行。前路漫漫，初心不改。