

# 基于射频识别的停车场管理系统设计与实现

## Design and Implementation of Rfid Parking Control System

胡世杰 李林森 毛佳佳 (上海交通大学 信息安全学院 上海 200030)

**摘要:** 介绍上海交大研制的以远距离无源 UHF 波段的射频识别技术为核心的 PUR3000 读写器的硬件系统, 结合基于串口通信的读写器控制模块和上层逻辑模块给出了 RFID 停车场管理系统的应用解决方案, 以 Windows 平台的 Visual Studio 2005 为开发环境, 利用串口通信的 API 实现访问 RFID 读写器的功能, 最终完成了停车场管理系统的基本功能。

**关键词:** 射频识别 远距 RFID 无源 UHF PUR3000 读写器 停车场管理系统

射频识别技术, 英文全称为 Radio Frequency Identification ( 简称为 RFID ), 是指相关的无线电技术在自动设备识别 ( AEI ) 领域中的具体应用<sup>[1]</sup>。该技术利用无线射频方式进行非接触的双向通信, 以达到识别并交换数据的目的。可用于识别高速运动物体并可同时识别多个射频识别标签, 而且操作快捷方便, 不怕油渍、灰尘污染等恶劣环境, 特别适合于实现系统的自动化且不易损坏<sup>[2]</sup>。远距离无源 UHF 波段的 RFID 技术是目前国际先进的第四代 RFID 标准。本文介绍的停车场管理系统是将该技术应用到停车场管理系统中的一次成功尝试。

## 1 引言

### 1.1 文章安排

本文第 2 节给出停车场管理系统组成原理。第 3 节介绍 PUR3000 系列读写器的硬件特性。第 4 节给出 PUR3000 系列读写器的软件接口。第 5 节介绍 RFID 停车场管理系统软件的设计目标。第 6 节给出 RFID 停车场管理系统软件的实现。第 7 节给出结论以及未来工作。

### 1.2 基本介绍

RFID 的组成包括读写器, 天线和标签。

使用 RFID 系统的停车场需要为每部车辆发放一个 RFID 标签, 也称车卡。在停车场出入口放读写器。读写器每隔固定时间发送询问信号, 如果此时正好有车辆需要出入库, 则该车辆上的标签返回它的身份代码。

读写器读取到标签身份后提交给后台控制中心, 由控制中心将标签身份代码和数据库进行对比, 决定是否开闸放行。由于大部分操作可由 RFID 系统结合

控制中心计算机自动完成, 停车场可节省不少人力资源。另外, 在停车计费 and 车辆出入库状态统计方面, 停车场管理系统也可提供优秀的解决方案。

## 2 RFID 停车场管理系统组成原理

停车场管理系统包括放置在停车场出入口的读写器, 一个控制中心, 一个数据库, 和大量车卡。

一辆车首先需要去控制中心注册并领取一张未关联其他车辆的车卡, 此时, 控制中心将该车辆登录进数据库, 成为一条主数据, 并和车卡上的身份代码建立关联。一旦这样的关联建立后, 车卡即可认为代表该车辆。

控制中心通常是一台运行了停车场管理系统软件的计算机, 除了在发卡和销卡时需要一位管理员人工干预以外, 其他时候可自动运行, 通过与之连接的读写器, 控制中心可掌握任何时刻进出停车场的车辆并在后台进行处理, 包括开关出入口的闸机, 车辆计费以及车辆状态统计和显示等。管理员可以通过管理界面方便地使用和查看这些数据。

位于出入口的读写器会以固定时间间隔扫描附近的标签并传回控制中心。读写器与控制中心之间可以通过多种方式连接。目前停车场管理系统仅使用了 RS232 串口连接方式。且由于串口设备的限制, 系统目前只支持 1 个读写器设备。

停车场管理系统使用的读写器是 PUR3000 系列读写器。

## 3 PUR3000 系列读写器的硬件特性

PUR3000 系列读写器的工作频段范围为 902 -

928MHz。读写器与电子标签之间可通过硬件、软件和固件实现全自动、高速、双向数据传输,无需任何人工干预,大大节省人工工作量。标签中保存的信息可直接传输到主机或用户的数据库中,且有蜂鸣器发出声音,可使用户轻松了解 PUR3000 系列读写器的工作状态是否正常。

读写器的主要特性如下:

ISM UHF 频段:工作频率为 902 - 928MHz,该频段为 FCC 定义的无需认证即可使用的频段

适用范围广:能读取所有符合 ISO18000 - 6 B 型协议的无源电子标签

远距离:室外稳定工作距离不低于 6 米

高可靠:适应工业工作环境(-40℃ - 85℃),独特的防水、防冲击措施

高智能:读写器与电子标签之间可实现自动、高速、双向数据传输,无需人工干预

低成本:电子标签是无源的

通用接口:RS - 232 串口、RS - 485、Wiegand 26

外形小巧,性价比高:这是由工作频段决定的

更安全:通信加密与数据校验

停车场管理系统使用 RS - 232 串口连接 PUR3000 系列读写器和控制中心。控制中心上运行的停车场管理系统软件根据 PUR3000 系列读写器的软件接口定义的格式,通过串口向读写器发送命令并接收返回信息,所有的逻辑判断功能都在控制中心完成。

#### 4 PUR3000 系列读写器的软件接口

读写器的软件接口定义了管理软件向读写器发送命令并获取返回信息的格式。这里以 RS - 232 串口为例,简单介绍一下接口规范。

RS - 232 串口通信参数如下:波特率 9600,8 数据位,1 起始位,1 停止位,无奇偶校验。

读写器以被动方式工作,即仅在接收到来自串口的控制命令后才进行各种操作。为了方便表示,从控制中心发往读写器的一组数据串称为命令包,从读写器发往控制中心的一组数据串称为返回包。

命令包由 5 个部分组成:

(1) BootCode:引导码,1 个字节,固定为 40H。

(2) Length:包有效长度,1 个字节。该长度为后 3 个部分的总字节数。

(3) Command:命令码,1 个字节。

(4) Command Param:命令参数,其长度随命令而变化。

(5) CheckSum:校验和,1 个字节,为从引导码 (BootCode) 开始到命令参数 (Command Param) 全部字节总和、丢弃进位后的字节补码。

返回包也由 5 个部分组成:

(1) BootCode:引导码,1 个字节。命令正确执行时,返回包引导码为 F0H;命令执行失败时,返回包引导码为 F4H。

(2) Length:包有效长度,1 个字节。该长度为后 3 个部分的总字节数。

(3) Command:命令码,1 个字节,与接收到命令码相同,表示该返回包是对该命令的响应。

(4) Return Data:返回数据,返回命令执行结果,其长度随命令而变化。

(5) CheckSum:校验和,1 个字节,为从引导码 (BootCode) 开始到返回数据 (ReturnData) 全部字节的总和、丢弃进位后的字节补码。

#### 5 RFID 停车场管理系统软件的设计目标

在设计停车场管理系统软件时,主要目的是通过软件控制读写器扫描标签,并在读取到标签身份信息后作相应的处理。

通过目标细分,整个软件被分成 4 个模块,每个模块的设计目标如下:

读写器操作模块。该模块是系统软件用来控制读写器的部分,它的主要功能有:通过串口向读写器发送命令,读取标签身份代码,设置读写器参数,如发送功率,工作频率等。

数据库操作模块。该模块是系统软件用来访问数据库的部分,它的主要功能有:向数据库中增加新的记录,修改或删除已有的记录,根据一个主键(比如标签身份 ID)查询数据库中的记录。其设计目标是隔离具体的数据库管理系统 API,为上层提供统一的接口。这使得我们可以轻易替换实际使用的数据库管理系统和相应的数据库操作模块,整个系统软件的其他模块不需要做任何修改。

逻辑模块。该模块是系统软件的核心部分,负责整合其余模块的功能,提供完整的解决方案,包括新标签分发,已有记录修改,标签识别,车辆出入控制,临时标签的分发和回收,计费系统,日志记录功能等。另外,考虑到停车场管理的安全性要求,还可提供异常报警机制。

#### 6 RFID 停车场管理系统软件的实现

停车场管理系统软件以 C++ 语言实现,开发环

境是 Microsoft Visual Studio 2005 [3]。下面我们将分模块介绍实现方法。

### 6.1 读写器操作模块的实现

根据设计目标,读写器模块被实现为一个 reader 类,职责是为逻辑模块提供普适的操作接口并隐藏具体的硬件操作指令。

Reader 类对外的接口,也就是访问权限为 public 的函数一共有 4 个,分别是构造函数 reader,析构函数 ~reader,标签扫描函数 get\_labels 以及设置读写器天线功率的函数 set\_output\_power。函数签名如下:

```
reader( const wstring &device );
~reader( );
vector < wstring > get_labels( void );
void set_output_power( unsigned char power );
```

Reader 类的构造函数接受一个 wstring 的 const 引用作为参数,该参数指示和读写器相连的硬件的名字。在当前的实现中意味着串口的设备名字。构造函数除了负责在系统中创建一个 reader 类的实例以外,还负责具体硬件的初始化,具体包括:根据传入的参数打开相应的串口设备,清除串口错误状态,设置串口缓冲区,设置串口超时时间,设置波特率等参数。

Reader 类的析构函数负责关闭串口设备并在系统中析构一个 reader 类的实例。

Get\_labels 函数将控制读写器对周围进行一次扫描,如果在扫描范围内存在 RFID 标签,则将它作为返回值返回。PUR3000 系列读写器目前可以一次读取一个标签,今后在技术上实现了防碰撞算法后,还可以一次读取多个标签。因此,get\_labels 函数的返回值类型是 vector < wstring >。

Set\_output\_power 函数负责设置读写器天线的输出功率。

从软件设计的角度来说,reader 类应该被设计为一个纯虚基类,仅提供上述的纯虚接口,并另外实现一个如 WindowsSerialReader 之类的派生类来继承这个基类,专门用来操作 Windows 系统下以串口连接的读写器,或者实现一个 LinuxNetReader 用来操作 Linux 系统下以网络接口连接的读写器。但是从实际应用的角度来看,预先设计的类继承体系可能存在错误的需求预判,所以尽可能等有实际需要后再进行重构。

### 6.2 车辆信息主数据 car\_info 的实现

停车场管理系统使用的车辆信息主数据在数据库中的表示是一条数据记录,每一辆车,也即每一张发行

的车卡都有一条相关的主数据记录在数据库中。而在软件中,它的表示则是一个名为 car\_info 的类的抽象数据类型。Car\_info 包含的成员变量及详细解释如下。

Id 是该车辆关联的 RFID 标签,即车卡的身份代码。Type 是车卡的类型,目前分为临时卡和普通卡。Car\_plate 是该车辆的车牌,在添加主数据时,车牌是必须提供的信息。Car\_owner 是车主姓名,car\_type 是车辆类型,car\_color 是车辆颜色,这些都不是必须提供的信息,仅作为参考。Time 是最后一次读写器读到该车卡的时间。

作为抽象数据类型,car\_info 也有自己的操作接口,这些接口可用来简化对 car\_info 的操作,接口如下:

```
car_info( );
bool is_valid( );
void record_time( );
bool can_be_create( );
bool can_be_delete( );
```

Car\_info 构造函数负责在系统中创建一个 car\_info 实例并将各成员变量初始化。

Is\_valid 函数负责检查这条记录是否合法,检查包括各变量长度是否超出预期,是否具有合法的车卡 id 和车牌信息等。

Record\_time 将在该车卡被读写器读取到时被调用,将当前的系统时间记录在 time 这个成员变量中。

Can\_be\_create 和 can\_be\_delete 这两个函数被用来检查该车卡信息是否完备,是否满足发卡或销卡条件。下图是停车场管理系统的发卡 and 销卡流程。

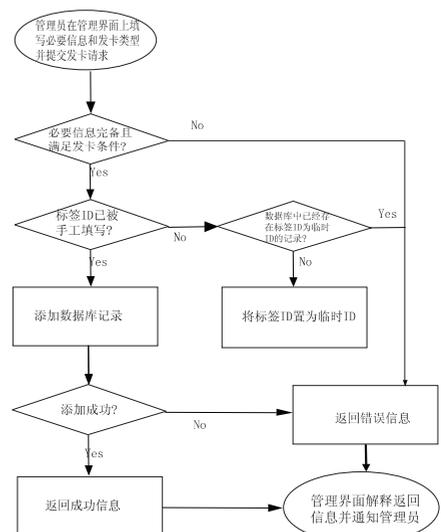


图 1 停车场管理系统发卡流程

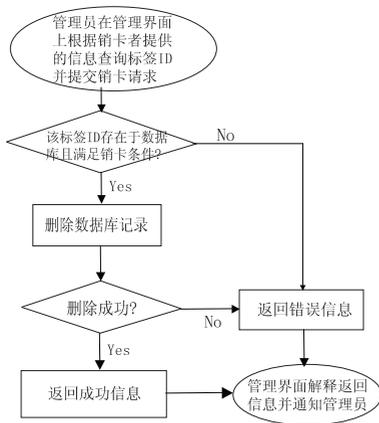


图2 停车场管理系统销卡流程

### 6.3 数据库操作模块的实现

逻辑模块希望数据库操作模块提供3个基本数据库操作功能,即添加、删除和查询数据记录的功能。所以,数据库模块拥有一个纯虚基类 database,提供3个访问权限为 public 的纯虚接口,分别是用于添加记录的 add\_info,用于删除记录的 del\_info 和用于查询记录的 get\_info。函数签名如下:

```

virtual bool add_info( car_info &info ) = 0;
virtual void del_info( wstring id ) = 0;
virtual bool get_info( wstring id, car_info &info ) = 0;
Add_info 函数将一个 car_info 中的数据添加到数据库。Car_info 在前面已经介绍过,是停车场管理系统使用的车辆信息主数据,它在软件中的表示是一个 car_info 抽象数据类型,而在数据库中的表示则是一条记录。Add_info 函数将把 car_info 中的各字段数据分别抽取并重新组合成一条符合数据库表格式的记录,插入数据库中。如果成功添加,返回 true,否则返回 false。
  
```

Del\_info 函数根据参数 id 在数据库中查询车辆信息主数据,如果查询到主数据的关联标签身份代码和 id 完全相同,则该条主数据将被删除。

Get\_info 函数根据参数 id 在数据库中查询车辆信息主数据,如果查询到主数据的关联标签身份代码和 id 完全相同,则该条主数据中各字段内容将被分别抽取并重新组合成一个 car\_info 抽象数据类型,以输出参数的形式返回调用者。如果成功查询到主数据,返回 true,否则返回 false。

目前数据库操作模块以文本文件形式存放数据。保存数据的文件被称为数据文件。所有的添加,删除和查询操作都建立在对数据文件的操作之上。每一条

车辆信息主数据在数据文件中占据一行,各字段之间用制表符分隔,各字段内容及意义和 car\_info 的成员变量一一对应。这种用文本文件形式实现的数据库在系统中被称为 simpledb。

### 6.4 逻辑模块的实现

逻辑模块是停车场管理系统的核心模块,它负责整合其余模块,提供完整的解决方案。当然,目前逻辑模块只是完成了它的基本功能,它负责调用其它模块提供的功能并完成以下流程:

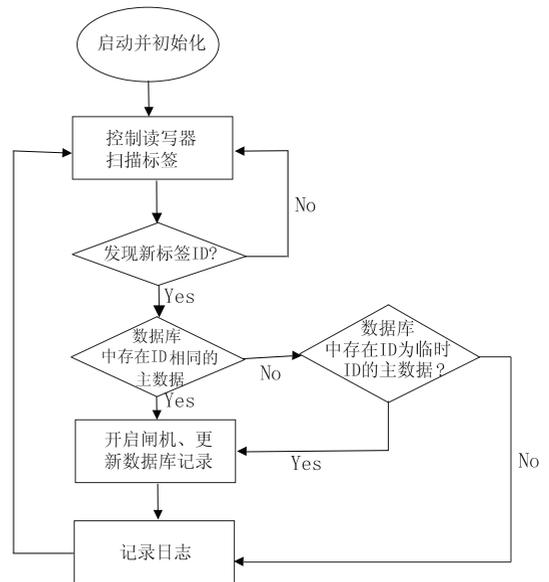


图3 停车场管理系统主流程

## 7 结束语

目前实现的停车场管理系统还处于原型开发阶段,为了使得该系统能够实际运用在各类停车场,还需要进行后续开发。后续开发包括将 PUR3000 读写器硬件接口改用 Ethernet 网络接口,支持网络接口的读写器操作模块,支持 ODBC 的数据库操作模块,支持多个读写器的逻辑模块等。而且,根据实际停车场使用的车辆出入控制功能,还需要对出入口控制进行二次开发。

### 参考文献

- 1 施俊. 远距离 RFID 在安防领域的创新应用. 中国公共安全(综合版), 2006, (5).
- 2 张益强, 等. 远距离射频识别系统及其应用前景. 中国数据通信, 2004, (1).
- 3 李现勇. 求是科技. // Visual C++ 串口通信技术与工程实践. 第二版, 北京: 人民邮电出版社, 2002.