

基于物联网的自助洗车系统设计与实现

孙开鑫, 樊启润, 唐彪, 万美琳, 卢仕

(湖北大学 物理与电子科学学院, 武汉 430062)

摘要: 针对传统洗车低效率、高消耗等问题, 设计实现了一套基于物联网技术的新型自助洗车机系统; 系统分为洗车机终端和云服务器端两个部分; 洗车机终端以 STM32F103VET6 作为主控芯片, 并利用霍尔流量传感器获取水流量, 实现流量计费; GPRS-SIM868 模块可实现与远程服务器的无线数据通信, 并通过搭载各类传感器采集洗车机的状态数据; 同时, 云服务器端与洗车机终端建立 UDP 通信, 实现对洗车机终端数据的获取, 并通过以 Tomcat 作为 Web 服务器、MySQL 为数据库建立 B/S 架构的数据管理系统; 用户可以根据洗车机终端 LCD 显示屏提供的信息, 自主选择支付方式、洗车模式以及计费模式; 同时, 后台管理人员可以通过网络及时了解自助洗车机的相关信息, 对洗车机进行实时监控与管理。

关键词: 物联网; 自助洗车机; UDP 通信; JavaWeb; MySQL

Design and Implementation of Self-service Car Wash System Based on Internet of Things

Sun Kaixin, Fan Qirun, Tang Biao, Wan Meilin, Lu Shi

(Faculty of Physics and Electronic Science, Hubei University, Wuhan 430062, China)

Abstract: Aiming at the low efficiency and high consumption of traditional car washing, a new self-service car washing machine system based on Internet of things technology is designed and implemented. The system is divided into two parts: car washing machine terminal and cloud server. STM32F103VET6 is used as the main control chip in the car washing machine terminal, and the Hall flow sensor is used to obtain the water flow and realize the flow billing. The GPRS-SIM868 module can realize wireless data communication with the remote server, and collect the state data of the car washing machine with various sensors. At the same time, the cloud server side established UDP communication with the car washing machine terminal to achieve the data acquisition of the car washing machine terminal, and established the B/S data management system with Tomcat as the Web server and MySQL as the database. Users can independently choose payment mode, car washing mode and billing mode according to the information provided by the LCD display of the car washing machine terminal. At the same time, background managers can timely understand the relevant information of the self-service car washing machine through the network, real-time monitoring and management of the car washing machine.

Keywords: IoT; self-service car wash; UDP communication; JavaWeb; MySQL

0 引言

随着社会经济的快速发展, 私家车的拥有量在快速增长, 人们对汽车出行的依赖感逐渐增加, 汽车的清洁和保养也越来越受到重视^[1]。洗车作为汽车清洁的主要组成部分, 越来越为人们所需。然而传统洗车行业却存在着较大的弊端, 首先传统洗车无法灵活的进入社区、停车场等车流量高的场所, 居民洗车往往需要去专门的店面, 所以无法实现灵活洗车; 其次, 传统的洗车方式会浪费大量的水资源, 不利于建设资源节约型社会。

针对上述传统洗车行业所出现的问题, 本文设计了一套自助洗车系统, 能根据用户的需求实现自助洗车, 并对洗车用水按流量或者按时间计费, 真正做到即来即用、即

用即走。同时, 本系统可以将洗车机的相关信息通过网络发送至远程服务器端, 实现洗车机联网, 从而利于管理人员对洗车机进行管理, 以及对洗车行业的大数据统计与分析。

1 系统设计

自助洗车系统由终端洗车机和云端管理系统组成, 其整体框图如图 1 所示。终端洗车机由单片机主控板、LCD 显示屏、GPRS 网络通信模块、支付模块、传感器组和水泵组成。主控芯片通过水流量计反馈的信息来控制水泵开关和计费, 从而实现按流量计费的洗车方式, 并通过 GPRS 模块将洗车机信息打包发送到远程服务器端。云端管理系统主要由 UDP 服务器、MySQL 数据库和 Web 服务器组成。UDP 服务器接收从洗车机终端发送的数据信息, 并将数据信息存入数据库, Web 服务器通过网页将数据库中的信息展现出来。

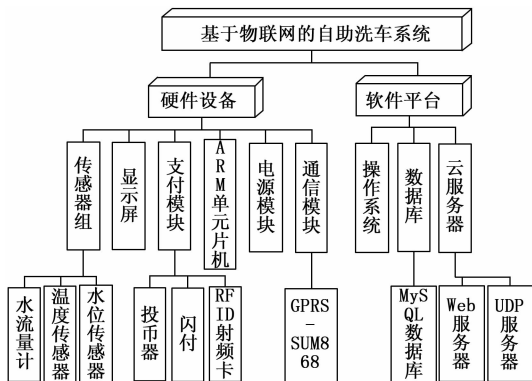
2 洗车机设计

洗车机设计分为洗车机硬件设计和洗车机软件设计。

收稿日期: 2019-04-24; 修回日期: 2019-05-15。

基金项目: 国家自然科学基金(61704050)。

作者简介: 孙开鑫(1998-), 男, 湖北天门人, 大学本科, 主要从事嵌入式系统方向的研究。



洗车机硬件由洗车机外壳、主控板、LCD 显示屏、水桶、泡沫桶、水泵和水流量计等组成。洗车机硬件结构示意图如图 2 所示。

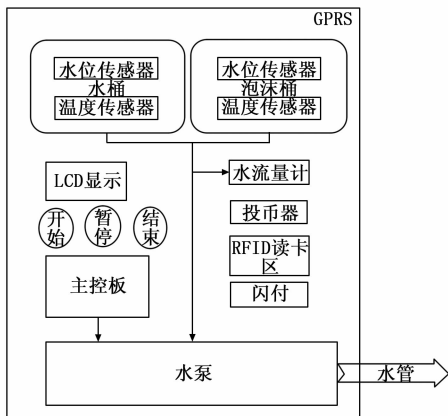


图 2 洗车机结构示意图

洗车机软件程序设计主要为 GPRS 网络通信模块和支付模块的配置和使用^[2]。洗车机终端运行程序流程图如图 3 所示。

2.1 硬件设计

2.1.1 主控板设计

本系统的主控板选用基于 Cortex - M3 内核的 STM32F103VET6 作为主控芯片，并且集成了 USB 电源输出接口、温度和液位传感器接口、水流量计的 OUT 输出接口、投币器接口、闪付串口接口、以及 11 个继电器接口和按键若干。如图 4 所示，为主控板的三维 PCB 图。

2.1.2 主要模块的应用

1) 水流量计：本系统选用霍尔流量传感器检测水流量^[3]。水流量传感器主要由塑料阀体、水流转子组件和霍尔传感器组成。当水通过水流转子组件时，磁性转子转动并且转速随流量变化而变化。随着磁铁和霍尔开关的距离改变，通过霍尔开关的磁通量也发生变化。当电流垂直于外磁场通过导体时，磁通量最大，在导体的垂直于磁场和电流方向的两个端面之间会出现电势差，进而霍尔开关就

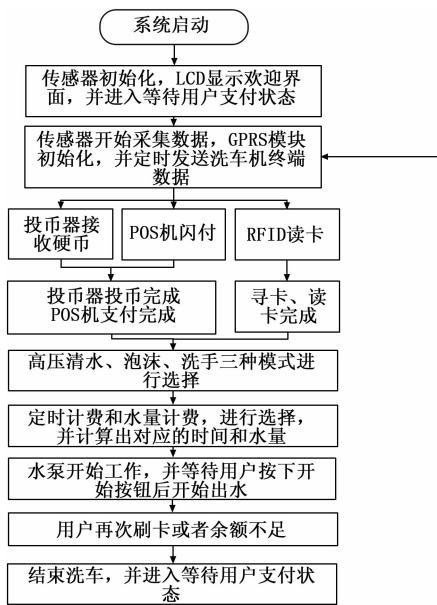


图 3 主控程序流程图

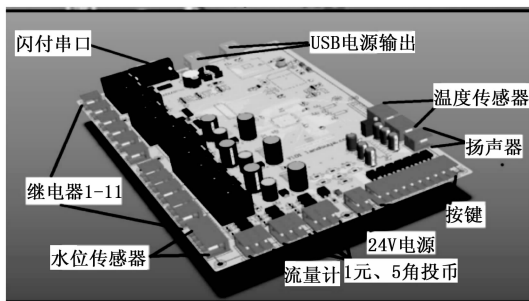


图 4 三维 PCD 图

会产生差动脉冲信号并输出。通过检测脉冲信号的数量，并根据脉冲数量和水流量之间的对应关系就可以得出脉冲数量对应的水流量。因此，水流量的测量主要是依靠对传感器输出脉冲的频率进行处理，由流量计在一段时间下产生的高电平数决定，即 Q (流量) = F (频率) / R (商家设定值)，对于不同的流量计其 R (商家设定值) 不同。本系统选用直径为 20 mm，流完 1 L 水输出 2400 个脉冲的水流量计。

2) LCD 显示屏：本系统选用 2.8 寸的 ALIENTEK TFT-LCD 模块作为显示屏。该模块支持 65 K 色显示，显示分辨率为 320 × 240，接口为 16 位的 80 并口。LCD 的驱动程序步骤为：设置 STM32 与 TETLCD 模块的管脚连接，初始化 TFTLCD 模块以及通过函数将字符和自定义汉字显示到 TFTLCD 模块中。

3) 温度传感器：本系统选用 DS18B20 数字温度计模块测量水的温度^[4]。该模块将周围环境的温度信号转换为数字信号，并将转换结果存放到 Scratchpad 中。Scratchpad 一共有 9 个字节 (BYTE0 - BYTE8)，其中转换的数字信息就存放在 BYTE0 和 BYTE1 中。所以，DS18B20 的使用程序

就是读取 Scratchpad 中 BYTE0 和 BYTE1 的值, 并通过温度—数据对应表得出相应的温度数据。温度传感器开发使用的程序流程如下:

- 1) Initial (DS18B20 初始化命令)
- 2) ROM Command (与功能命令相匹配的 ROM 命令)
- 3) Convert T (功能命令: 让 DS18B20 开始工作, 采集周围温度)
- 4) 等待 DS18B20 采集完成
- 5) Initial (DS18B20 初始化命令)
- 6) ROM Command (与功能命令相匹配的 ROM 命令)
- 7) Read Scratchpad (功能命令: 读取 Scratchpad 的 BYTE0 和 BYTE1 字节)
- 8) 对照温度—数据表计算出当前周围环境温度值, 并将数据结果反馈给主控芯片。

温度—数据对照如表 1 所示。

表 1 温度—数据对照表

temperature /°C	digital output /BINARY	digital output /HEX
+125	0000 0111 1101 0000	07D0h
+85 *	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0191h
+10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1000	FFF8h
-10.125	1111 1111 0101 1110	FF5Eh
-25.0625	1111 1110 0110 1111	FE6Fh
-55	1111 1100 1001 0000	FC90h

4) 水位传感器: 本系统选用基于红外光学原理的液位传感器套件测量水位。传感器套件实物如图 5 所示。该套件利用红外光学原理, 将检测的液位信号通过光学传递转换成电压信号, 从而判别液位情况。该套件的输出方式有两种, 方式 1: A 管脚模拟输出 0~4.5 V, 单片机通过对其模拟输出量进行 AD 转换后可以得出具体的液位数据; 方式 2: D 管脚电平信号输出, 单片机通过检测输出端的高低电平可以判断液位是否低于所设定阈值。本系统采用第二种输出方式来测量当前水位, 当水位低于设定阈值时, 传感器套件 D 管脚输出低电平。单片机采集水位不足信号后, 通过通信模块发送信息到远程服务器端, 通知后台工作人员加水。

2.2 软件设计

2.2.1 GPRS 通信模块

自助洗车系统最重要的部分就是数据的远程通信。本系统通过单片机采集各个传感器的数据信息, 并利用 GPRS 模块将数据发送到远程服务器。GPRS 模块的通信流程为 GPRS 模块通过基站将本地信息发送到移动公司的服务器, 移动公司服务器再通过互联网将信息转发到具有公网 IP 的



图 5 水位传感器实物图

服务器中。基站、移动服务器都由通信公司来提供, 所以本系统只需要对 GPRS 模块进行编程^[5]。

GPRS 模块有多种型号可供选择, 本系统选用 SIM868 来实现面向无连接的 UDP 通信, SIM868 作为 socket 客户端, 远程数据接收服务器作为 socket 服务器端。SIM868 模块编程主要步骤为: 时钟初始化、串口初始化、串口接收和发送函数编写, 以及通过串口通信对 SIM868 进行初始化。单片机和 SIM868 通信波特率数值设置为 9600, 并通过串口发送“AT+指令”对 SIM868 实现模块上电、检测 SIM 卡是否在位、查看是否注册 GPRS 网络等初始化设置。如表 2 所示, 为 SIM868 主要指令信息表。其中指令“AT+CIPSTART”后需要写入传输协议类型、服务器 IP 地址和相应的端口号。本系统选用 UDP 面向无连接协议进行网络通信, 所以传输协议类型应为 UDP; 服务器 IP 地址为可访问的公网 IP; 端口号绑定为 19999。当数据发送完成后, 通过指令查询服务端是否将信息接收完成, 并等待下一次数据的写入。

表 2 SIM868 初始化命令表

指令	主要作用
AT+QGNSSC=1\R\N	对 GPS 模块上电
AT+CPIN? \R\N	检查 SIM 卡是否在位
AT+CGREG? \R\N	查看是否注册 GPRS 网络
AT+CIPSTART="UDP", \ "220.180.239.212",8532\R\N	选择传输层协议和目的主机的 IP 地址和端口号
AT+CIPSEND\R\N	等待数据进入数据缓冲区, 并发送
AT+CIPSACK\R\N	查询服务端是否将信息接收完成

2.2.2 支付系统模块

支付系统模块主要分为 3 种支付方式: 投币器支付、RFID 射频卡支付和银联闪付。

1) 投币器支付: 本系统选用“港都 007”款投币器模块, 该投币器接口主要有 9 个接线管脚, 分别标注为“COUNTER, GND, COIN, DC+12 V, FAST, MIDDLE, SLOW, NO, NC”。其中“GND、DC+12 V”分别接地和直流电 12 V 电源; “COIN”为信号输出线, 其本身是没有电压输出的, 需要在外部上拉一个电阻才能输出对投币完成的识别脉冲; “FAST, MIDDLE, SLOW”3 个接线引脚

用于控制输出信号线的高低电平脉冲时间, 为了防止误差, 让单片机好捕捉到脉冲信号, 所以本系统选择的是 SLOW-60 ms 的脉冲时间; 最后“NC, NO”分别表示正常情况下投币器信号端为低电平, 有投币时输出为高电平和正常情况下投币器信号端为高电平, 有投币时输出为低电平(前提是信号输出端已经有了上拉电阻), 本系统选择 NC 常开, NO 常闭。根据以上的接口信息, 我们可以利用单片机对投币器进行控制。当投币器正常工作时 COIN 信号输出口为低电平; 当有硬币进入时, COIN 口输出 60 ms 的高电平信号, 单片机接收信号后将硬件计数加 1, 完成一次投币。当投币数量满足洗车要求时, 进入洗车模式选择, 并根据用户的选择计算出洗车的时间或者洗车的用水量

2) RFID 卡支付: 一套完整的 RFID 硬件系统由 Reader 和 Transponder 两部分组成, 其动作原理为由 Reader 发射特定频率的无线电波能量给 Transponder, 用以驱动 Transponder 电路将内部的 ID CODE 送出^[6]。本系统选用 MF-RC522 射频模块作为 Reader, 标准 S50 空白卡作为 Transponder。MF-RC522 射频模块向 S50 卡发一组固定频率的电磁波, 卡片内有一个 LC 串联谐振电路, 其频率与读写器的发射频率相同, 在电磁波的激励下, LC 谐振电路产生共振, 向电容充电, 当电荷达到 2 V 时, 此电容可作为电源为其电路提供工作电压, 将片内的数据发射出去或读取读卡器的数据。

对于本系统而言, MF-RC522 射频模块主要用于对射频卡的充值操作和扣费操作。其中充值操作由后台人员在固定操作点进行充值操作, 扣费操作由洗车终端程序进行自动扣费。本系统由单片机利用 SPI 接口对 RC522 模块进行编程, 编程流程为模块初始化、寻卡、防冲突、选卡、操作卡(读卡或者写卡)。MF-RC522 射频模块的初始化命令如图 7 所示, 寻卡操作主要由单片机向射频卡发送命令进行寻卡, 完成寻卡操作并成功寻到卡后程序将进入防冲突操作, 在防碰撞正确读到卡的序列号后, 使用 select 命令与卡建立通信, 选卡成功后进入认证操作, 认证完成之后就能进入读写操作了。洗车机终端主要根据水流量计或者定时器来计算消费金额, 消费完成后停止出水, 并将余额信息写回射频卡中。

Table 149: Command overview

Command	Command code	Action
Idle	0000	No action; cancels current command execution.
Mem	0001	Stores 25 byte into the internal buffer
Generate RandomID	0010	Generates a 10 byte random ID number
CalcCRC	0011	Activates the CRC co-processor or performs a selftest.
Transmit	0100	Transmits data from the FIFO buffer.
NoCmd Change	0111	No command change. This command can be used to modify different bits in the command register without touching the command. E.g. Power-down.
Receive	1000	Activates the receiver circuitry.
Transceive	1100	Transmits data from FIFO buffer to the antenna and activates automatically the receiver after transmission.
-	1101	Reserved for further use
MFAuthent	1110	Performs the MIFARE® standard authentication as a reader
Soft Reset	1111	Resets the MFRC522

图 6 MF-RC522 初始化命令表

模块功能的 POS 机, 然后设定好所需要支付的金额, 并等待用户完成付款。闪付完成后, 由闪付串口向单片机发送支付完成信息, 单片机接收信息后开始进行洗车操作。

3 云管理系统设计

云管理系统设计主要分为 UDP 服务器设计、数据库设计和 Web 服务器设计, 其中 UDP 服务器通过 Python 语言调用 socket 库进行网络编程, 负责将洗车机硬件端发送过来的信息进行切片处理并连接数据库, 将信息导入数据库^[7]。数据库采用 MySQL 关系型数据库, 存放洗车机终端的相关信息和运行状态。Web 服务器采用 Tomcat 服务器, 通过连接数据库将洗车机相关信息显示到前端。UDP 服务器和 Web 服务器为同一具有公网 IP 地址的主机, 但是两个服务器端口号不同, UDP 服务器端口号与洗车机终端 GPRS 模块绑定端口号一致为 19 999, 而 Web 服务器端口号为 8080。两个服务器共享一个数据库, 通过数据库将两个服务器联系起来。

3.1 UDP 服务器设计

网络通信传输层协议主要有面向连接的传输控制协议 TCP 和面向无连接的用户数据报协议 UDP。本系统选用 UDP 协议进行远程通信, UDP 服务器端采用循环的方式处理来自多个客户端的请求。由于采用的是 UDP 方式来通信, 使得没有一个客户端可以总是占据着服务器, 因此服务器在设计、编程、排错和修改等工作都比较简单^[7]。

3.1.1 UDP 数据包格式

基于 UDP 协议的数据包由 UDP 首部和数据两部分组成^[8]。其中 UDP 的首部格式固定, 所以本系统需要对数据部分的格式进行规定, 数据部分的格式可以理解为应用层协议^[9]。具体数据的通信协议如表 3 所示。通信协议中具体的数据内容如表 4 所示。

洗车机启动后首先向服务器发送开机成功包通知服务器, 之后每隔 1 分钟向服务器发送心跳包保持在线, 并定期向服务器报告各个传感器的数值。由于温度传感器读取数据的取值范围为 $-55 \sim +125$ °C, 所以协议中温度数据内容从 0x00 到 0xB4 依次表示 $-55 \sim +125$ °C。在交易状态中, 由 0x00 和 0x01 分别表示当前是否有交易; 由 0x00 和 0x01 分别表示客户选择的交易类型为按照时间扣费还是按照流量扣费; 由 0x00、0x01 和 0x02 分别表示客户的付款方式为水卡支付、POS 闪付还是投币支付; 另外使用 3 个字节表示预付费金额, 其中整数部分两个字节, 小数部分一个字节, 表示范围为 0.0~999.9。

3.1.2 服务端程序设计

UDP 服务器端由 Python 语言进行开发, 需要导入的第三方库为 socket 库和 MySQLdb 库。通过 socket 库函数进行网络编程, 接收洗车机终端发送的信息, 并通过 MySQLdb 库函数将数据导入数据库。在将数据导入数据库之前, 需要完成数据库创建, 并根据洗车机运行状态等数据信息, 完成了相关表的创建。服务器程序流程如图 7 所示。

3) 银联闪付: 银联闪付需要在银行办理具有闪付支付

表 3 UDP 数据包数据部分格式

名称	包头	硬件 ID	类型	数据长度	数据内容	日期	校验	包尾
长度	2 字节	4 字节	1 字节	4 字节	N 字节	7 字节	1 字节	2 字节
内容	0x55, 0xaa	硬件编号	运行状态 传感器数值 交易状态	数据长度	① ② ③	年份占用 2 个字节, 月、日、时、分、秒各自占用 1 字节, 共 7 字节——一节	除包头包尾校验外所有数据字节的异或值(BCC 校验)	0x11, 0xbb

表 4 具体数据内容

编号	类型	类型字节	数据内容	长度	数值	解释
①	运行状态	0X01	设备是否开启	1 字节	0xab: 开机 0xac: 心跳	报告系统的运行状态, 包括开机和心跳包
②	传感器数值	0x02	温度 1	1 字节	0x00-0xB4	报告系统的各个传感器数值, 参数等信息。其中有无水、水位低、水位中、水位高都是开关量, 只有 0 和 1 两种状态。
			温度 2	1 字节	0x00-0xB4	
			流量计	1 字节		
			有无水	1 字节	0x00-0x01	
			水位低	1 字节	0x00-0x01	
			水位中	1 字节	0x00-0x01	
			水位高	1 字节	0x00-0x01	
③	交易状态	0x03	GSM 信号强度	1 字节		报告系统当前的交易状态, 当前是否有客户, 余额多少等信息
			当前是否有交易	1 字节	0x00, 0x01	
			交易类型	1 字节	0x00, 0x01	
			付费方式	1 字节	0x00-0x02	
			水卡 ID	2 字节	ID 号	
			预付费金额	3 字节		
剩余金额	3 字节					

cvfrom () 方法就可以获取客户端的 IP 地址和端口号以及洗车机的相关信息。接收到信息以后, 需要对信息进行一个简单的校验来判断数据的准确性, 如果数据准确就对信息进行切片处理, 然后将信息通过调用 MySQLdb 库 cursor.execute (sql) 方法执行相应的 SQL 语句, 最后通过调用 db.commit () 方法将数据提交到数据库, 完成一次数据的处理。

3.2 洗车机信息管理平台设计

洗车机信息管理平台需要实时显示洗车机终端的相关数据, 而且对于不同地区的管理人员需要负责的洗车机终端也不尽相同, 所以本系统的 Web 前端需要使用动态页面。本系统选用 Java 语言进行 Web 开发, 并选用 Tomcat 作为 Servlet 容器, 采用 JSP+Servlet+MySQL 体系开发。通过 Servlet 连接并访问数据库 MySQL, 最后将数据库信息回传给 JSP, 通过 JSP 显示洗车机相关信息的页面^[10]。

3.2.1 后台程序设计

服务器后台程序主要是 Servlet 对数据库的连接和信息处理。其主要步骤为, 通过加载 JDBC 驱动实现对 MySQL 数据库的连接, 数据库连接完成之后创建一个对象用来执行 SQL 语句。由于本系统只需要对数据库信息进行查询, 所以仅需要通过 SQL 语句查询所需的数据, 同时利用 JavaBean 定义数组对象接收查询结果, 最后将结果返回给前端。

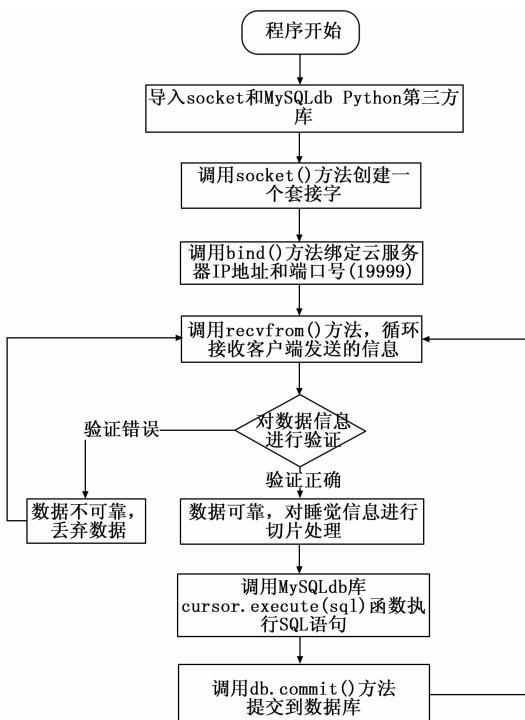


图 7 UDP 服务器程序流程图

首先创建一个被动套接字, 绑定好 IP 地址和端口号, 与 TCP 服务器不同, UDP 服务器不需要与客户端进行三次握手连接, 可以直接进入循环接收。通过调用 socket 库 re-



图 8 信息管理网站页面

本功能列表。网站页面如图 8 所示。

4 实验结果和分析

完成对系统的设计和实现之后，进行系统测试工作。测试内容包括洗车机的正常运作状态、各项数据传感器的采集精度、GPRS 通信模块的数据传输准确性、以及后台管理系统的运作情况。测试中，我们在学校附近设置了自助洗车点，测试时间为一周。各项传感器采集的数据通过 GPRS 模块发送到实验室租借的云服务器上，并存储到 MySQL 数据库中，实现了对洗车机的远程监控，图 9 为信息管理系统 Web 服务器连接到 MySQL 数据库对洗车机终端发送的数据进行显示的界面。

5 结束语

基于物联网技术的自助洗车系统中，通过 STM32 对多种传感器的实时信息采集，可以更加准确的获取洗车机终端的详细运行信息，并利用霍尔流量传感器实现可以按照流量计费的新型计费方式。同时，将网络编程技术、数据库技术和 Web 开发技术相结合，能有效的搭建一个针对智能终端的信息管理系统。相比于传统洗车方式，自助洗车机能具有更高的效率，更少的时间以及更少的水资源浪费，相信在物联网技术快速发展的今天，基于物联网的自助洗车系统能得到一个更好的应用。

ID	UDP数据	IP地址	端口	设备编号	数据类型	数据长度	数据内容	日期时间	录入日期	是否删除	是否启用
1	55 AA 05 D9 FF 30 02 00 00 08 37 37 00 00 01 117.136.19.35	1894	05D9FF30	02	11	37 37 00 00 00 00 01 2000-01-04 18:28:45	2000-01-04 18:28:45	2015-07-07 11:42:01	0	1	
2	55 AA 05 D9 FF 30 03 00 00 0C 00 01 D1 DF: 117.136.19.35	22600	05D9FF30	03	12	00 01 D1 DF 57 04 00 0 2000-01-04 18:02:04	2000-01-04 18:02:04	2015-07-07 11:36:00	0	1	
3	55 AA 05 D9 FF 30 02 00 00 08 37 37 00 00 01 117.136.19.35	13071	05D9FF30	02	11	37 37 00 00 00 00 01 2000-01-04 18:22:41	2000-01-04 18:22:41	2015-07-07 11:35:57	0	1	
4	55 AA 05 D9 FF 30 03 00 00 0C 00 01 D1 DF: 117.136.19.35	31905	05D9FF30	03	12	00 01 D1 DF 57 04 00 0 2000-01-04 18:22:33	2000-01-04 18:22:33	2015-07-07 11:35:49	0	1	
5	55 AA 05 D9 FF 30 02 00 00 08 37 37 00 00 01 117.136.19.35	21594	05D9FF30	02	11	37 37 00 00 00 00 01 2000-01-04 18:22:09	2000-01-04 18:22:09	2015-07-07 11:35:19	0	1	
6	55 AA 05 D9 FF 30 03 00 00 0C 00 01 D1 DF: 117.136.19.35	5756	05D9FF30	03	12	00 01 D1 DF 57 04 00 0 2000-01-04 18:21:55	2000-01-04 18:21:55	2015-07-07 11:35:11	0	1	
7	55 AA 05 D9 FF 30 03 00 00 0C 00 01 D1 DF: 117.136.19.35	22839	05D9FF30	02	11	37 37 00 00 00 00 01 2000-01-04 18:05:23	2000-01-04 18:05:23	2015-07-07 11:38:39	0	1	
8	55 AA 05 D9 FF 30 03 00 00 0C 00 01 D1 DF: 117.136.19.35	5783	05D9FF30	03	12	00 01 D1 DF 57 04 00 0 2000-01-04 17:59:19	2000-01-04 17:59:19	2015-07-07 11:32:35	0	1	
9	55 AA 05 D9 FF 30 02 00 00 08 37 37 00 00 01 117.136.19.35	14500	05D9FF30	02	11	37 37 00 00 00 00 01 2000-01-04 17:52:04	2000-01-04 17:52:04	2015-07-07 11:06:00	0	1	
10	55 AA 05 D9 FF 30 03 00 00 0C 00 01 D1 DF: 117.136.19.35	16002	05D9FF30	03	12	00 01 D1 DF 57 04 00 0 2000-01-04 17:35:51	2000-01-04 17:35:51	2015-07-07 10:49:07	0	1	
11	55 AA 05 D9 FF 30 02 00 00 08 37 37 00 00 01 117.136.19.35	18660	05D9FF30	02	11	37 37 00 00 00 00 01 2000-01-04 17:35:21	2000-01-04 17:35:21	2015-07-07 10:48:37	0	1	
12	55 AA 05 D9 FF 30 03 00 00 0C 00 01 D1 DF: 117.136.19.35	2575	05D9FF30	03	12	00 01 61 22 50 04 00 0 2000-01-04 16:46:51	2000-01-04 16:46:51	2015-07-07 10:00:06	0	1	
13	55 AA 05 D9 FF 30 02 00 00 08 37 37 00 00 01 117.136.19.35	3519	05D9FF30	02	11	37 37 00 00 00 00 01 2000-01-04 16:46:31	2000-01-04 16:46:31	2015-07-07 09:59:47	0	1	
14	55 AA 05 D9 FF 30 03 00 00 0C 00 01 D1 DF: 117.136.19.35	8802	05D9FF30	03	12	00 01 61 22 50 04 00 0 2000-01-04 16:44:42	2000-01-04 16:44:42	2015-07-07 09:57:58	0	1	
15	55 AA 05 D9 FF 30 02 00 00 08 37 37 00 00 01 117.136.19.35	15493	05D9FF30	02	11	37 37 00 00 00 00 01 2000-01-04 16:44:32	2000-01-04 16:44:32	2015-07-07 09:57:47	0	1	

图 9 测试结果

3.2.2 数据库设计

本系统选用 MySQL 关系型数据库实现对数据的存取^[11]。根据 UDP 协议数据部分，以及后台管理人员的相关信息，所以数据库需要建立多张表。主要是设备信息、交易记录、统计查询、管理员注册信息和 UDP 监控。设备信息中包含设备编号、设备名称、设备放置地点、负责人姓名、设备是否处于工作状态以及是否有水等信息；交易记录中包含设备编号、名称型号和付款方式等信息；统计查询中包含收益情况等；管理员信息包括账号身份信息和创建日期等信息；UDP 监控中包含目的地址、设备编号和数据类型等信息。

3.2.3 前端设计

前端页面主要由传统网页 HTML 中插入 Java 程序段形成 JSP 文件，并通过 JSP 技术进行动态页面显示。信息管理系统网站包含设备信息管理、UDP 数据管理和系统管理。其中设备信息管理包含设备信息（用于实时显示设备编号、名称、型号等信息）、交易记录（用于实时显示客户交易方式、消费金额、交易时间等信息）和统计查询（用于统计归纳并实时显示每台设备月季收入）三项基本功能。UDP 数据管理包含接收数据（显示设备目的地址、设备编号、数据类型）和 UDP 监控（用于显示设备 IP 地址、目标端口号等信息）两个基本功能。系统管理包含系统菜单、初始化、数据备份、后台管理、信息提醒、系统配置、操作日志、计划任务等基

参考文献:

[1] 钟石根, 张良杰, 朱丽敏, 等. 基于物联网技术的自助洗车系统设计 [J]. 现代电子技术, 2017, 40 (20): 1-4, 9.

[2] 刘火良, 杨 森. STM32 库开发实战指南基于 STM32F103 [M]. 第二版. 北京: 机械工业出版社, 2017.

[3] 乔 波. 多功能智能水流量测控系统设计 [D]. 成都: 电子科技大学, 2017: 23-26.

[4] 张仲明, 郭东伟, 吕 巍, 等. 基于 DS18B20 温度传感器的温度测量系统设计 [J]. 实验技术与管理, 2018, 35 (5): 76-79, 88.

[5] 张瑞增. 基于智能车锁位的共享停车位管理系统研究与设计 [D]. 济南: 山东大学, 2017: 37-44.

[6] 单承赣, 单玉峰, 姚 磊. 射频识别 (RFID) 原理与应用 [M]. 第 2 版. 北京: 电子工业出版社, 2015.

[7] Rhodes B, Goerzen J. Python 网络编程 [M]. 诸豪文, 译. 第 3 版. 北京: 人民邮电出版社, 2016.

[8] 谢希仁. 计算机网络 [M]. 第 7 版. 北京: 电子工业出版社, 2008.

[9] 朱望纯, 曹 健, 宋 辉. 基于物联网的农场信息接收与发布平台设计与实现 [J]. 计算机测量与控制, 2015, 23 (7): 2549-2552.

[10] 林光源, 张国平, 高雪莲, 等. 基于 JavaWeb 的远程库房环境监控系统 [J]. 信息技术, 2017 (5): 50-55.

[11] 黄文娟. 基于 Java 和 MySQL 的图书馆信息化管理系统设计 [J]. 电子设计工程, 2019, 27 (2): 20-24.