

第一单元

案例驱动的物联网系统开发

本章以智能鱼缸系统为案例,基于 CC3200 和 FreeRTOS 介绍物联网系统中物端的系统开发流程及其架构。需要注意的是,出于系统设计的完整性考虑,本章还将涉及一些移动端应用开发和物联网云平台的介绍。

1.1 系统概述

1.1.1 系统架构

智能化设备在人们日常生活中的应用已是随处可见,如智能家电、智能导航和智能健康检测等。借助 Wi-Fi 或者移动互联网,智能家居系统能够让家庭中的各种智能家具在同一个网内互联,实现各个设备间的通信,打破用户对各种家具进行操作的时间和空间限制,能够随时随地进行控制,给生活带来了极大的便利。

当前市面上的智能鱼缸大多采用蓝牙连接,用户能通过手机来控制鱼缸,能够进行温度调节和光照调节等操作。但受限于蓝牙的覆盖范围,用户只有在家中时才能用手机查看鱼缸的状态,而距离如此之近并没有创造打破时间、空间限制的便利,不能体现“智能”之智慧。因此,解决方案就是——将鱼缸接入物联网云计算平台,让数据通过云平台进行传输。这样,用户就可以不再受距离的影响,随时随地查看鱼缸的情况。

智能鱼缸总体分为三部分,分别为智能鱼缸硬件部分、物联网云计算平台以及移动终端应用程序,如图 1-1 所示。

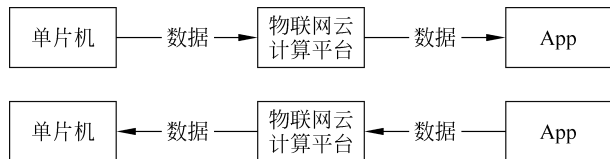


图 1-1 智能鱼缸的系统架构

不同部分各司其职,相互配合。智能鱼缸的硬件部分在图中对应“单片机”,这一部分主要负责对鱼缸的状态监控,将光照、温度和水位值等状态信息上传并智能化处理。物联网云计算平台作为智能鱼缸管理平台,主要提供用户管理、消息推送、P2P 授权通信以及 P2P 通信支持等功能。移动终端应用程序对应图 1-1 中的 App,作为智能鱼缸的手机客户端,主要为用户提供设备绑定、设备信息和设备状态查询等功能。如图 1-1 所展示的,当用户查询鱼缸状态时,鱼缸状态信息,如水位值、温度值、光照值等数据会先通过光照传感器、温度传感器和水位传感器等传感器传输至物联网云计算平台,而后再由物联云计算平台将数据传输

给 App,显示在设备详情的 App 页面上。当用户控制智能鱼缸时,数据会先从 App 传递至物联网云计算平台,再从云平台传递至相应的控制器。

1.1.2 软硬件开发环境

智能鱼缸的设计包括两部分,分别为硬件设计和云平台设计,开发环境如下。

- (1) 软件环境: Windows 7、CentOS 7、FreeRTOS(Version 8.5.2)。
- (2) 硬件环境: Intel Core(TM)、2 核 1.8GHz、8GB 内存、1TB 硬盘。
- (3) 开发工具: VS 2020、Eclipse、MySQL、MongoDB、RabbitMQ、Maven。

CC3200 SDK 包含基于 CC3200 开发所需的用于 CC3200 可编程 MCU(微控制单元)的驱动程序、闪存编辑器、示例程序和说明文档等。闪存编辑器可用于配置网络 and 软件参数 (SSID(服务集标识)、接入点通道、网络配置文件等)、系统文件和用户文件。示例程序中包 含多种应用示例,例如,片上互联网应用示例:使用 Wi-Fi 解决方案发送电子邮件、从互 联网上获取时间和天气信息等; Wi-Fi 应用示例: 简易 Wi-Fi 配置、TCP(传输控制协议)/ UDP(用户数据报协议)连接等; MCU 外设应用示例: 并联摄像机、I²S 音频、SDMMC(安 全数字多媒体卡)、ADC(模数转换)等。

CC3200 SDK 安装好之后的文件夹包含如表 1-1 所示的内容。

表 1-1 CC3200 SDK 文件夹包含的内容

文 件 夹	包 含 内 容
docs	说明文档,主要是对 SDK 中的应用示例进行讲解
driverlib	包含 CC3200 所有的底层驱动,如 UART、I ² C 等底层配置代码
example	提供一些基本功能演示案例,可供初学者快速学习及掌握
inc	寄存器地址宏定义
netapps	实现常用的网络应用层协议,包括 HTTP、MQTT(消息队列遥测传输)等
oslib	操作系统 API(应用程序接口)
simplelink	SimpleLink 框架,提供从基本的设备管理到无线网络配置等功能
simplelink_extlib	实现在线升级和对 Flash 的读写
third_party	第三方开发工具
tools	工具集

SimpleLink 框架提供了一套广泛的功能,从基本的设备管理到无线网络配置、BSD 插座服 务等。为了获得更好的设计粒度,这些功能被隔离到各个模块中,每个模块都代表了 SimpleLink 的不同功能,SimpleLink 模块组成如图 1-2 所示,其各个模块的功能如表 1-2 所示。

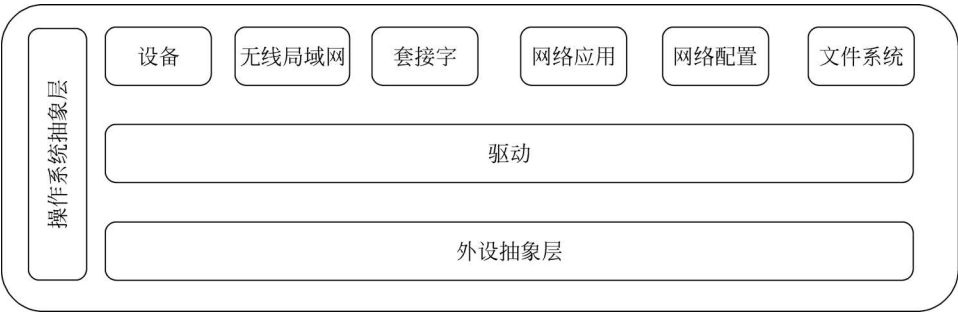


图 1-2 SimpleLink 模块组成

表 1-2 SimpleLink 各模块功能

模 块	功 能
Device	初始化主机,控制与网络处理器的通信
WLAN	连接到 AC,扫描接入点,添加/删除接入点配置文件,WLAN 安全
Socket	UDP/TCP 客户端 Socket,UDP/TCP 服务器 Socket,UDP/TCP Rx/Tx
Netapp	DNS 解析,Ping 远程设备,地址解析协议
Netcfg	IP/MAC 地址配置
FS	文件系统读写

1.1.3 需求分析

如上所述,智能鱼缸案例由物端、云端、移动端三部分组成。本节先介绍云端和移动端的主要功能。再在此基础上,分别从软件需求的四个层次,即业务需求、用户需求、功能需求以及非功能需求,对物端系统进行需求分析。

1. 云端

本系统云端的主要功能包括以下几个。

(1) 用户管理。

用户的注册、登录、个人信息修改以及密码恢复等。

(2) 设备管理。

用户可以通过物联网云平台绑定设备、添加设备,并能够对添加的设备进行逻辑组网从而对组中的信息进行管理和设置。同时还支持用户对供应商、用户、第三方开发者等进行相关的安全管理。

(3) 设备控制。

云平台还为用户提供获取设备信息、修改设备信息、设置设备控制状态、更改设备状态以及设置设备控制范围等功能支持。另外,云平台能提供设备数据的持久存储能力,用户可以查看历史数据并获取最新数据。

2. 移动端

本系统的移动端 App 是让用户通过移动端来控制鱼缸,主要功能包括以下几个。

(1) 用户的登录和注册。

(2) 获取设备列表,绑定想要控制的设备。

(3) 对设备进行相应的控制操作。

用户通过 App 使用智能鱼缸的流程如图 1-3 所示。

3. 物端

针对智能鱼缸系统的物端开发需求,本书将从以下 4 方面进行详细说明。在实际的实验过程中,实验人员将先定性地分析不同的需求,并根据提出的需求进行定量的测试。本节先简要描述智能鱼缸系统的各类需求的具体含义,而在 1.2.4 节中将以吞吐量和响应时间为例,展示智能鱼缸系统具体的性能需求并进行相关测试,以此来说明量化指标在需求开发和软件测试活动中的具体应用。

(1) 业务需求。

业务需求指的是开发者需要为用户解决的问题。当前市面上大部分智能鱼缸大多采用

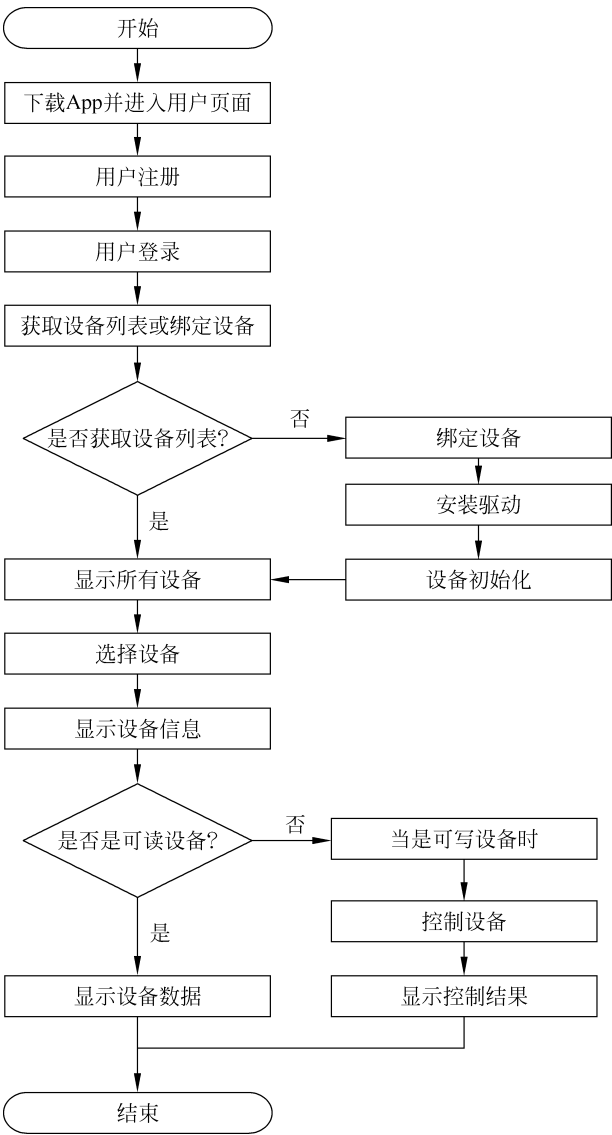


图 1-3 用户通过 App 使用智能鱼缸的流程

蓝牙技术进行鱼缸与用户终端的连接,存在的问题是其覆盖范围有限,一旦超出蓝牙的距离,用户便无法控制鱼缸,而借助移动互联网云平台可以解决这样的距离限制问题。具体业务需求如下。

- ① 创建设备列表,其中可以添加物联网设备并查看设备详情。
 - ② 查看智能鱼缸设备的当前以及历史状态数据,将光照强度、水位、温度等数据以一定形式反馈给用户。
 - ③ 用户可以在移动端远程控制鱼缸,例如设置参数来改变设备温度等。
- (2) 用户需求。

用户需求指的是用户想要系统实现的功能。以下从用户角度出发,充分考虑用户的使用习惯,设计用例并借助用例描述需求。用例图如图 1-4 所示。

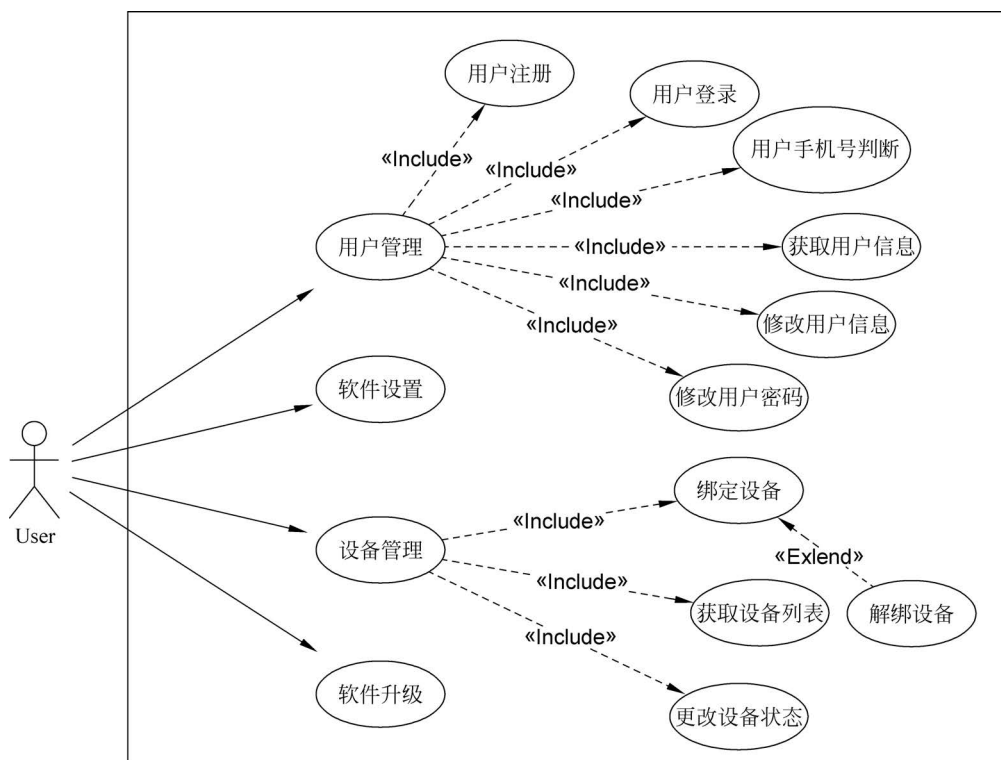


图 1-4 用户通过 App 使用智能鱼缸的流程用例图

从图 1-4 可以看出,系统的用户需求包括以下用例。

① 用户管理。包括用户注册登录、用户手机号判断、用户设置、获取用户信息以及修改用户密码等功能。

② 软件设置。主要指智能鱼缸应用程序中各种参数的设置,如界面的大小等。

③ 设备管理。主要包括绑定设备、解绑设备、获取设备列表等功能。

④ 软件升级。实现软件自动或手动更新升级功能。

(3) 功能需求。

功能需求指的是开发人员必须在产品中实现的功能,用户通过使用这些功能解决问题,从而满足业务需求。下面对智能鱼缸物端系统的功能需求进行分析。

智能鱼缸硬件部分的设计主要包括 CC3200、Wi-Fi 模块、用于显示当前状态的 LED 以及检测鱼缸状态的一系列传感器:水位传感器、温度传感器和光照传感器。各模块功能如下。

① 传感器模块。

3 个传感器负责从环境中获取相应的数据——水位传感器、温度传感器以及光照传感器分别监测鱼缸当前的水位值、温度值和光照值,并将数据发送至物联网云计算平台。

② Wi-Fi 模块。

数据传输过程中,Wi-Fi 模块提供数据传输的网络服务,通过 TCP/UDP 协议来收发相关数据。

③ CC3200 模块。

在系统整体运转过程中,CC3200 模块作为核心模块,主要负责数据运算、控制引脚以及不同模块间的通信,进而实现对水位、光照强度和温度的监控。智能鱼缸系统物端的硬件组成如图 1-5 所示。

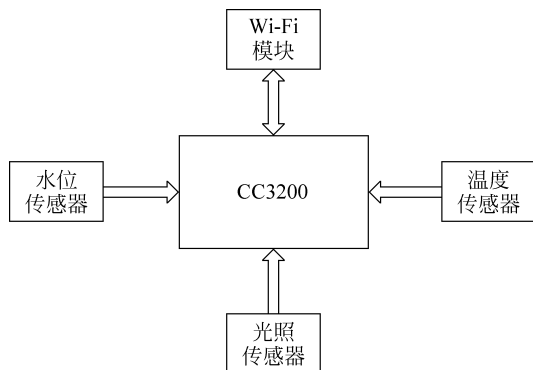


图 1-5 智能鱼缸系统物端的硬件组成

(4) 非功能需求。

为了全面分析系统的需求,对业务、用户以及功能需求进行分析之后,还需要补充对非功能需求的分析,非功能性需求指的是系统除功能需求以外,为满足业务需求而必须具有的特性,如安全性、可靠性等。本节主要从以下 5 方面进行系统非功能需求的分析:性能需求、安全需求、适用性需求、可靠性需求,以及 UI 界面友好性需求。

① 性能需求。

系统的性能指标一般包括吞吐量、响应时间和并发量等。

吞吐量: 指的是物联网云平台的网络传输流量,由于本系统中有大量的数据需要实时传输,因此物联网云平台应该具有较大的传输流量。

响应时间: 指的是用户在发出命令时,系统响应的延迟时间。当系统接收到用户发出的控制鱼缸状态的命令时,应该控制延迟,在短时间内作出响应。

并发量: 指的是在不影响用户使用的前提下,云平台可同时接入的最大设备数量以及用户量。系统应该提高最大并发量,避免出现拥堵,并保证正常运行下的良好体验。

② 安全需求。

由于本智能鱼缸系统由单片机、物联网云计算平台和 App 三部分组成。因此安全性需求也需要从这三方面进行考虑。

单片机: 即物端的安全性,在鱼缸内的单片机与物联网云计算平台进行数据传输时,系统通过设置安全措施来防范外部攻击,确保不会发生数据被篡改的情况,另外,也要保障物理方面的安全,避免出现因发生物理破坏而威胁人身安全的情况。

物联网云计算平台: 为了确保用户的信息安全,平台应对用户进行分级,并根据等级设置不同的权限,对数据进行加密并设置部分公开。

App: 为了保障移动端的安全登录,应对用户的合法性和准确性进行验证,如增加指纹、面部识别等功能,避免被不法分子利用用户账户进行违规操作。

③ 适用性需求。

为了提高系统的适用性,需要充分考虑单片机、物联网云计算平台和 App 各个部分的适配:系统物端的各部分模块应能够进行拆卸更改,便于适应不同的机械和硬件设置;App 要尽可能适应当前市面上流行的不同型号的手机以及常用的操作系统;物联网云计算平台则需要与物端和 App 适配。

④ 可靠性需求。

除了满足用户最基本的功能需求,系统也应具备较高的稳健性,避免发生时延高、页面加载慢或操作卡顿等情况从而影响用户的正常使用,最重要的是要避免出现系统部分或整体运行崩溃的情况,因此需要在设计时遵循“高内聚,低耦合”等原则,提高系统的可靠性。

⑤ UI 界面友好性需求。

为了方便用户操作,提高用户体验,UI 页面布局应简洁、大方,操作设置不应复杂,要迎合大部分用户的习惯。同时可以支持个性化定制,让用户能按照个人喜好自主更改页面设置。

1.2 系统的物端设计与实现

基于上文的需求分析对智能鱼缸系统进行设计,本节主要介绍智能鱼缸系统物端 3 大功能模块的设计:光照模块、温度模块以及水位模块,并进行相关功能模块测试。

1.2.1 光照模块

光照模块的主要功能是自动检测光照强度并实现灯光的自主开关。硬件结构采用了 CC3200 嵌入式开发板和光照传感器,光照模块的组织结构如图 1-6 所示。

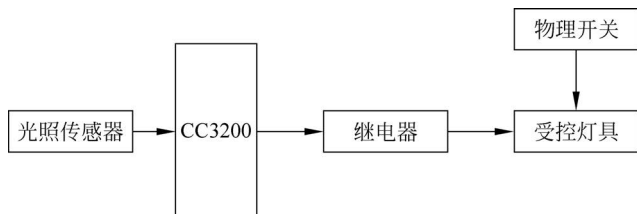


图 1-6 光照模块的组织结构

光照模块的工作流程如图 1-7 所示,主要分为以下几步。

(1) 初始化系统并进行相应的网络配置,将开发板接入网络。

(2) CC3200 获取光照传感器寄存器中的数据,并进行数据处理,而后通过 Wi-Fi 模块发送给服务器。

(3) 根据用户的实际光照需求动态调节光照强度,检测是否存在超出阈值的情况,控制光强于目标范围之内。其中光照的调整主要通过继电器设备动态地调整电流强度,从而让光照强度发生变化。

(4) 等待在经过一定时延后再读取下一个数据,重复步骤(2)、(3)。

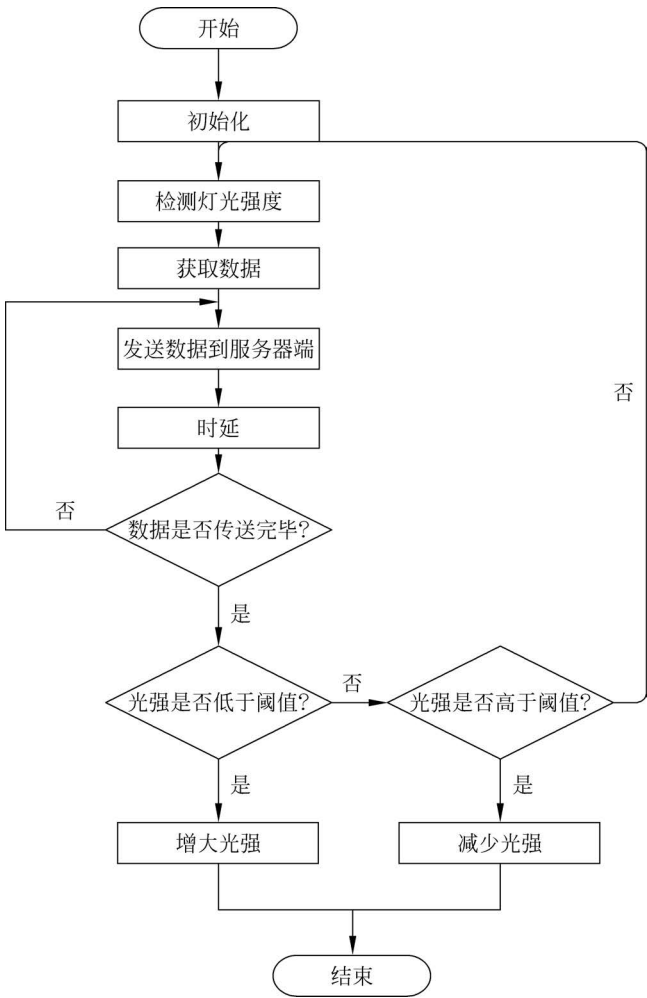


图 1-7 光照模块的工作流程

1.2.2 温度模块

温度模块的主要功能是对当前环境的温度数据进行采集并实时传输,硬件结构采用了CC3200 嵌入式开发板和温度传感器等,温度模块的组织结构如图 1-8 所示。

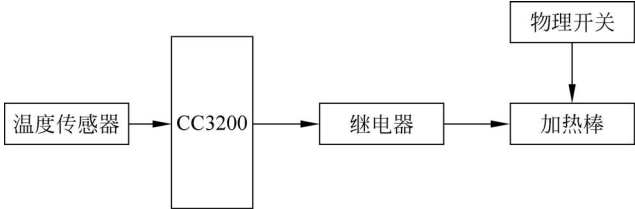


图 1-8 温度模块的组织结构

温度模块的工作流程如图 1-9 所示,主要步骤如下。

- (1) 首先初始化系统并进行相应的网络配置,将开发板接入网络。
- (2) CC3200 获取温度传感器寄存器中的数据,并进行数据处理,而后通过 Wi-Fi 模块

发送给服务器。

(3) 根据用户的需求调整鱼缸内的温度,当温度低于阈值时,鱼缸会通过继电器打开加热棒直至达到用户设定的温度区间。

(4) 等待,经过一定时延后再读取下一个数据,重复步骤(2)、(3)。

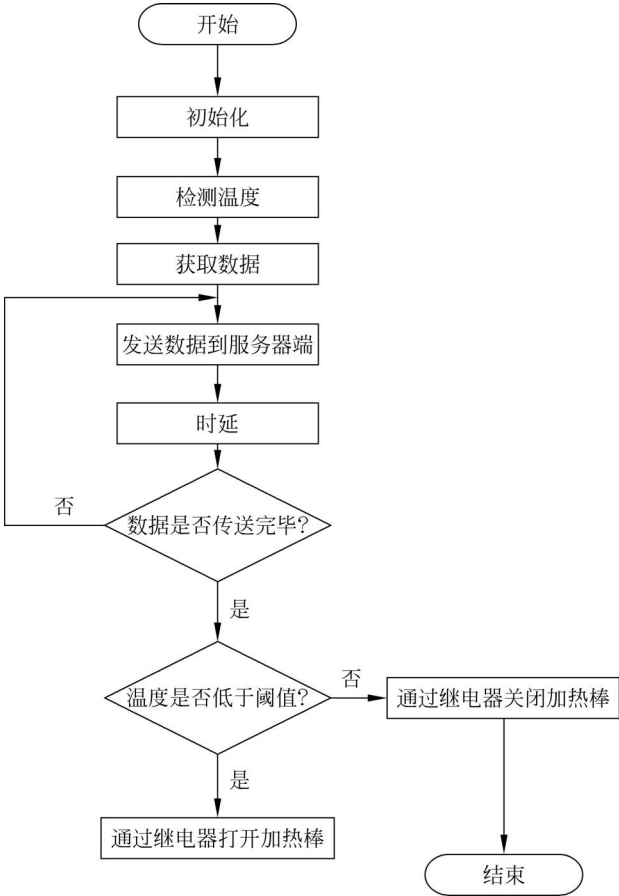


图 1-9 温度模块的工作流程

1.2.3 水位模块

水位模块的主要功能是对当前环境的水位定时监控,并进行数据传输,同时实现自动调整。水位模块的硬件结构采用了 CC3200 嵌入式开发板和水位传感器等,水位模块的组织结构如图 1-10 所示。

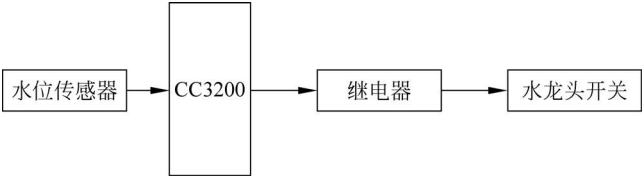


图 1-10 水位模块的组织结构

水位模块的工作流程如图 1-11 所示,主要步骤如下。

(1) 首先初始化系统并进行相应的网络配置,将开发板接入网络。

(2) CC3200 获取水位传感器寄存器中的数据,并进行数据处理,而后通过 Wi-Fi 模块发送给服务器。

(3) 根据用户的需求调整水位,当水位低于阈值时,鱼缸会通过继电器打开水龙头,向鱼缸加水到阈值,然后关闭水龙头。

(4) 等待,经过一定时延后再读取下一个数据,重复步骤(2)、(3)。

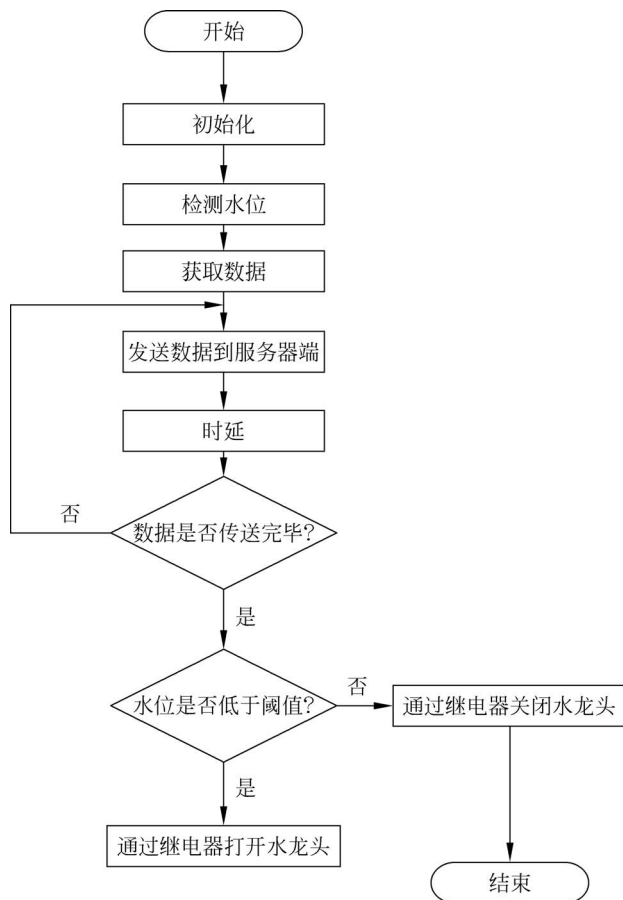


图 1-11 水位模块的工作流程

1.2.4 系统测试

1. 功能测试

为了验证功能模块设计的正确性,本节使用白盒测试的方法对物端功能进行测试,主要对光照模块、温度模块和水位模块进行测试。

1) 光照模块功能测试

表 1-3 是针对光照模块的光照控制功能进行测试的数据,测试时将鱼缸放置于暗房中,将光强计置于鱼缸底部,用于实时显示实际光强值,通过观察其是否达到目标光强值来验证

模块功能。从表中的测试数据可以看出,所有的预期输出和实际输出结果是一致的,测试结果全部正确。

表 1-3 光照模块功能测试用例

测试序号	A			
测试目的	测试光照模块能否正确控制光照强度			
测试功能	调节鱼缸内的光强			
测试条件	在主功能页面开启自动光照控制模式,并设置不同的光照强度目标值(光强单位: Lux)			
子测试序号	测试条件	理论结果	实际结果	正确或错误
A-1	设置目标光强为 50	光强升至 50	光强为 50	正确
A-2	设置目标光强为 100	光强升至 100	光强为 100	正确
A-3	设置目标光强为 150	光强升至 150	光强为 150	正确
A-4	设置目标光强为 200	光强升至 200	光强为 200	正确
A-5	设置目标光强为 300	光强升至 300	光强为 300	正确
A-6	设置目标光强为 200	光强降至 200	光强为 200	正确
A-7	设置目标光强为 150	光强降至 150	光强为 150	正确
A-8	单击采集图像为 100	光强降至 100	光强为 100	正确
A-9	单击采集图像为 50	光强降至 50	光强为 50	正确

2) 温度模块功能测试

表 1-4 是针对温度模块的温度控制功能进行测试的数据,测试时将鱼缸置于空调房中(室温约为 16℃),将温度计置于鱼缸中,玻璃泡悬空于水中,用于显示实际温度值,通过观察其是否达到目标温度值来验证模块功能。从表 1-4 中的测试数据可以看出所有的预期输出和实际输出结果是一致的,测试结果全部正确。

表 1-4 温度模块功能测试用例

测试序号	B			
测试目的	测试温度模块能否正确控制鱼缸水温			
测试功能	对鱼缸内水温的升降进行控制			
测试条件	在主功能页面开启自动温度控制模式,并设置不同的温度目标值(温度单位: ℃)			
子测试序号	测试条件	理论结果	实际结果	正确或错误
B-1	设置目标温度为 18	水温至 18	水温为 18	正确
B-2	设置目标温度为 20	水温至 20	水温为 20	正确
B-3	设置目标温度为 23	水温至 23	水温为 23	正确
B-4	设置目标温度为 25	水温至 25	水温为 25	正确
B-5	设置目标温度为 30	水温至 30	水温为 30	正确
B-6	设置目标温度为 25	水温至 25	水温为 25	正确
B-7	设置目标温度为 23	水温至 23	水温为 23	正确
B-8	设置目标温度为 20	水温至 20	水温为 20	正确
B-9	设置目标温度为 18	水温至 18	水温为 18	正确

3) 水位模块功能测试

表 1-5 是针对水位模块的温度控制功能进行测试的数据,测试时选用高为 850mm、起

初水位为 100mm 的鱼缸。用刻度尺测量实际水位,通过观察其是否达到目标水位值来验证模块功能。从表 1-5 中的测试数据可以看出,所有的预期输出和实际输出结果是一致的,测试结果全部正确。

表 1-5 水位模块功能测试用例

测试序号	C			
测试目的	测试水位模块能否正确控制鱼缸水位			
测试功能	对鱼缸内水位的控制			
测试条件	在主功能页面开启自动水位控制模式,并设置不同的水位目标值(水位单位: mm)			
子测试序号	测试条件	理论结果	实际结果	正确或错误
C-1	设置目标水位为 200	水位至 200	水位至 200	正确
C-2	设置目标水位为 300	水位至 300	水位至 300	正确
C-3	设置目标水位为 400	水位至 400	水位至 400	正确
C-4	设置目标水位为 500	水位至 500	水位至 500	正确
C-5	设置目标水位为 600	水位至 600	水位至 600	正确
C-6	排水为 200,设置目标水位为 500	水位至 500	水位至 500	正确
C-7	排水为 200,设置目标水位为 400	水位至 400	水位至 400	正确
C-8	排水为 200,设置目标水位为 300	水位至 300	水位至 300	正确
C-9	排水为 200,设置目标水位为 200	水位至 200	水位至 200	正确

2. 性能测试

除了对 3 大模块的功能测试外,针对智能鱼缸系统的物联网云计算平台进行了性能测试,从压力测试、稳定性测试和安全性测试 3 个方面展开,最后根据测试结果进行总结评价。

测试目标: 在智能鱼缸系统的物联网云计算平台的登录模块中模拟实现 2000 个用户并发访问本系统。

1) 测试方案

本次实验从 8000 个用户测试数据中抽取 25%左右的用户,即约 2000 个数据进行测试。采用自动化测试工具,配置相关参数以及创建测试脚本,当准备工作完成后就选取 2000 个用户同时对该系统进行访问。打开服务器对该系统进行性能监视,时间间隔为 1 小时,连续观察 96 小时,系统分析各性能指标。

单独场景压力测试: 指在特定场景下对具体的一些功能进行压力测试,当外部要求满足用户的基本需求时,分析测试的结果能否达到最低标准。

混合场景压力测试: 指在不同的场景下对具体的功能进行压力测试,当各种临界指标不断提升时,分析测试的结果能否达到最低标准。

稳定性测试: 指系统在极端条件或者苛刻情况下时,测试系统资源是否能够保证该系统的正常运行,根据监控的结果情况分析系统的稳定性。

2) 测试用例

在同一个时间段,随机抽取 2000 个用户对该系统进行访问。

用例名称: 随机抽取 2000 个用户同时访问系统。

3) 测试场景

(1) 随机抽取 2000 个用户进行测试。

(2) 加压方案: 先后 3 轮, 分别以 1 秒、1 分钟、10 分钟为周期按照斐波那契数列不断递增的方式增加用户, 直到用户数量达到 2000 个时停止增长, 3 轮循环后分析最终的结果。

(3) 减压方案: 每一轮采用 3 个不同周期和比例的等比数列不断递减用户, 直到全部停止——第一次每 1 秒按照公比为 $1/2$ 的方式递减; 第二次每 1 分钟按照公比为 $1/3$ 的方式递减; 第三次每 10 分钟按照公比为 $1/5$ 的方式递减。

4) 测试步骤

(1) 配置测试环境。

(2) 用性能测试工具录制测试脚本。

(3) 选取特定场景进行实验, 执行时间: 5 小时。

5) 预期结果

(1) 系统的响应时间在 5s 以内。

(2) 系统的吞吐量不小于 100GB/s。

(3) 系统的 CPU 平均使用率不高于 70%。

6) 测试结果

在该性能测试实验中, 在同一时间段随机抽取 2000 个用户对同一网页进行访问, 分 3 轮进行加压与减压测试, 性能测试的时间跨度为 5 小时。本次测试中, 用户共提交了 600 多万次请求, 只有不到 1% 的请求无响应, 其余请求全部正常响应。

从响应的时间进行分析, 响应时间曲线平滑, 从整体情况来看比较平稳。事务平均响应时间为 1.1s, 在 5s 的接受范围内。

从单击率分析, 测试中单击率在 130~170 次/秒区间内, 整体比较稳定, 波动幅度不大, 平均单击率达 165 次/秒, 单击数也属于常规数值。

从吞吐量分析, 波动幅度在 200~230GB/s 范围内, 但不同时间段内的吞吐量峰值不断变化, 因此在不同的阶段吞吐量呈波浪式变化, 需要重点关注这些振幅大的区间段。

从性能测试数据分析, 系统的服务器 CPU 的使用率保持在 43% 左右, 较为稳定。内存使用率整体波动较大, 在 80~332GB 区间内基本满足预期。

3. 系统测试小结

通过对智能鱼缸系统三大功能模块的功能测试以及物联网云计算平台的性能测试分析可知, 智能鱼缸系统的设计基本满足业务需求, 并能达到预期目标。

1.3 系统联网任务的实现

智能鱼缸系统的网络任务主要包括一系列处理设备网络模块的操作, 如初始化网络、获取 IP 地址、建立站点(Station, STA)设备的连接、开启并监听内部 HTTP 服务器端等。

1.3.1 网络模块的开启和配置

首先将网络设备配置为默认状态, 在默认状态下的 STA 模式会从网络接入点(accesspoint, AP)获得 IP, 然后用获得的 IP 切换到 AP 模式, 这样其他智能设备就能够借助 HTTP 服务通过 IP 访问到设备内的文件数据。最后, 将设备连接到接入点后的展示信

息、对用户动作进行反馈的控制脚本以及其他需要的可执行二进制文件烧录在 CC3200 的外部 SFLASH 中即可。烧录的主要信息如表 1-6 所示。

表 1-6 烧录的主要信息

名 称	功 能
main. html	登录到接入点后展示的主页
test. jpg	背景图片
jquery-1. 8. 3. min. js	用于处理用户操作的 JS 脚本

1. 3. 2 HTTP Server 服务

CC3200 可以与网页文件系统进行数据传输,因为其中内置了支持 HTTP 协议的服务器。但由于网页文件是保存在串行闪存中的,并且在访问网页文件时需要提供整个路径,但在根目录中服务器只支持 www/safe 与 www/这两种路径。因此,服务器端为解析和返回数据分别设计了不同的函数,并分别存放在不同的文件中。其中,Http Core. c 文件中存放的函数主要负责对客户端的请求进行解析和处理,Http Dynamic. c 文件中存放的函数主要负责将系统采集的数据组合成 Response 并将其返回给客户端,而 Http String. c 文件中存放的函数主要负责操作所有的字符串和数字。下面定义了一个结构体,其中的变量值用于标识 HTTP 服务器端建立连接的状态。

```
1. struct Http Global State          //全局状态
2. {
3.     int listenSocket;              //监听套接字
4.     UINT16 uOpenConnections; //实际连接个数
5.     //最大连接数
6.     Struct HttpConnection Data connections [HTTP_CORE_MAX_CONNECTIONS];
7.     //HTTP 发送数据包个数
8.     UINT8 packetSend[HTTP_CORE_MAX_PACKET_SIZE_SEND];
9.     //HTTP 发送数据包大小
10.    UINT16 packetSendSize;
11.    //HTTP 接收数据包个数
12.    UINT8 packetReceive[HTTP_CORE_MAX_PACKET_SIZE_RECEIVED];
13.    //HTTP 接收数据包大小
14.    int packetReceiveSize;
15. };
```

HTTP 的可靠数据连接是使用 TCP 协议来实现的。首先,HTTP 服务器端需要建立 TCP 套接字(socket),然后等待客户端发送连接请求。接下来将路由配置为 AP 模式,获取 IP 地址,并将 socket 描述符、获取到的 IP 地址以及 80 端口进行绑定。之后服务器端和客户端便可以通过套接字描述、IP 地址、端口地址进行通信。受限于本系统中嵌入式软硬件资源的不足,将宏定义 HTTP_CORE_MAX_CONNECTIONS 设置为 10。服务器端将以轮询的方式遍历检查每个客户端是否存在数据请求。如果有客户端存在数据请求,还要对其数据请求的类型进行检查。服务器端将首先对客户端发来的数据请求中的统一资源名(uniform resource name,URN)进行检查,并先后在监督闪存(supervisory flash,SFLASH)以及 CC3200 的 ROM 中查询这个资源。若服务器端在 SFLASH 中找到了该资源,服务器端会将该资源封装到 Response 中,并将 Response 返回给客户端。若该资源不在 SFLASH

中,则服务器端会在 CC3200 的 ROM 中继续寻找,找到则返回该资源,未找到则返回 404 错误码。

1.3.3 嵌入式 HTTP 服务器

为方便用户进行嵌入式网络应用开发,CC3200 的 SimpleLink 驱动库中已经集成了 HTTP 服务器,并提供了相关的软件接口。并且 SimpleLink 中有相应的事件捕捉位,可以通过触发这个事件来执行所需要的指令。因此,在进行 HTML 网页交互时,客户端可以通过触发服务事件来调用相关的函数,从而使 HTTP 服务器向网页发送请求。下面展示了 HTTP 服务器的调用函数示例。

```
1. void SimpleLinkHTTPServerCallback (SlHTTPServerEvent_t * pSlHTTPServerEvent, SlHTTPServerResponse_t * pSlHTTPServerResponse) {
2.     if ((pSlHTTPServerEvent == NULL) || (pSlHTTPServerResponse == NULL)) { //参数检验
3.         printf ("Null Pointer\n\r");
4.         while(1) :
5.             switch (pSlHTTPServerEvent -> Event) { //判断 HTTP 请求类型
6.                 case SL_NETAPP_HTTPPOSTTOKENVALUE_EVENT { //POST 请求
7.                     //表示图像采集的 POST Token
8.                     if (0 == memcmp (pSlHTTPServerEvent ->
9.                         EventDataHTTPPostData.token_name.data, "SL_P_U.C",
10.                    pSlHTTPServerEvent -> EventData.HTTPPostData.token_name.len)) {
11.                         if (0 == memcmp (pSlHTTPServerEvent ->
12.                            EventData.HTTPPostData.token_value.data, "start",
13.                            pSlHTTPServerEvent -> EventData.HTTPPostData.token_value.len)) {
14.                             //POST Token 值为"start",则置位图像采集事件代码为
15.                             _lwevent -- set(&lwevent_group1, Event_CaptureImage):
16.                         }
17.                     }
18.                 }
19.                 break :
20.             default:
21.                 break
22.         }
23. }
```

1.4 本章小结

在介绍了 FreeRTOS 和 CC3200 的基础知识之后,本章以智能鱼缸系统为案例,介绍了物联网系统软件的设计方法和主要步骤。通过对智能鱼缸系统的案例分析,首先明确了系统需求,包括业务需求、用户需求、功能需求以及非功能需求四部分;接着对物端的三大模块,即光照模块、温度模块和水位模块进行了详细设计,同时介绍了该系统的软件测试;最后对物联网系统的网络任务进行了介绍。

第 2 章

基于 Android 的物联网 系统移动端开发

本章立足于物联网的实战场景,介绍基于 C/S 网络架构的智能鱼缸系统,聚焦于移动端的应用开发。在该案例中,智能鱼缸系统的移动端应用采用 Android 平台开发,本章着重分析其应用场景和全生命周期的开发过程。

2.1 需求分析

随着物联网技术的成熟,智能家居(smart home)设备逐渐走入人们的生活。智能鱼缸作为智能家居设备中的一种,为用户提供了只需连入互联网即可随时随地查看、管理家中鱼缸的能力。

按照物联网设备的类型进行分类,智能鱼缸属于复合的可读写型设备,智能鱼缸设备内部包括单片机、温度传感器、光照传感器、水位传感器,Wi-Fi 模块等。其中,单片机主要进行数据运算和引脚控制,实现鱼缸的水位控制、温度监控和光照强度监控功能;Wi-Fi 模块具有功耗低、集成度高、性能稳定等特点,其通过各引脚连接传感器设备实时收集数据,同时将数据通过互联网传输到云计算平台。

对于智能鱼缸系统,其用户角色包括了企业(company)、管理员(manager)、驱动开发者(driver developer)以及基础用户(user)。前三者主要通过浏览器访问云平台完成账号、设备、驱动数据的各种管理操作,此处不予赘述。而移动端应用的面向角色为基础用户。在接入云计算平台后,用户的智能鱼缸设备可由用户通过某种唯一性标识在 Android 移动端上进行绑定,用户可以通过移动端应用实时获取鱼缸状态。

2.1.1 功能需求分析

功能需求是对开发人员必须实现的软件功能的定义。通过这些软件功能,用户得以完成各项操作或任务,从而满足业务需求。由于习惯上经常用“应该”对功能需求进行描述,其有时也被称作行为需求。简单来说,功能需求描述了开发人员需要实现什么。

智能鱼缸移动端应用为用户提供以下功能。

- (1) 登录功能,用户登录后才能访问自己绑定的设备。
- (2) 注册功能,新用户可申请新账号以获取服务。
- (3) 用户能够查看已绑定设备列表。

- (4) 用户能够绑定设备,将新设备添加至设备列表。
- (5) 用户能够查看智能鱼缸的当前状态(包括水温、水位值、光照强度)以及相应的历史记录。
- (6) 用户能够操控智能鱼缸,如给鱼缸升温、加水以及照明。
- (7) 用户能够从设备列表解绑移除设备。
- (8) 鱼缸各指标(水温、水位值、光照强度)超过阈值时,系统发出警告通知用户。
- (9) 用户能够通过移动端应用查阅养鱼知识的相关资料。

2.1.2 非功能需求分析

非功能需求是系统除了业务需求、用户需求、功能需求外所需具有的特性,与系统特性相关,可以完善用户体验,为用户带来更好的系统使用感受。本节描述的系统非功能需求包括系统的性能需求、安全需求、可靠性需求、UI 界面友好性需求和可适应性需求。

1. 性能需求

系统性能指标一般包括响应时间、吞吐量、并发用户等。响应时间是指用户在发出命令时,鱼缸应尽快响应用户,缩短系统延迟时间;吞吐量是指物联网云计算平台应具有较大的网络传输流量;并发用户是指当云计算平台接入更多的设备、有多个用户同时与系统交互时,不影响用户的使用体验。

2. 安全需求

智能鱼缸系统由三部分组成,分别为单片机组成的鱼缸、物联网云计算平台和移动端应用,因此系统的安全性需求也从这三个方面出发来考虑。首先是鱼缸的安全性,鱼缸内的单片机向物联网云计算平台发送数据,同时接收物联网云计算平台发送的命令,系统应确保单片机不受攻击,数据不被篡改。其次是物联网云计算平台的安全性,不同角色的用户有着不同权限,不能使数据全部公开。最后是移动端应用,应用软件基于 Android 操作系统开发,对于用户合法性应谨慎验证。

3. 可靠性需求

智能鱼缸应满足用户最基本的需求,不能随意崩溃致使用户无法使用,同时也要避免页面响应慢、出现卡顿等情况的发生。

4. UI 界面友好性需求

UI 页面布局应符合大部分用户的操作习惯,使用户可以尽快熟练操作。

5. 可适应性需求

本终端应用软件基于 Android 操作系统开发,Android 手机的品牌众多,屏幕大小不尽相同,开发应用软件时应尽可能考虑手机适配问题。

2.2 用例建模

在用例建模中,功能性需求是通过参与者(系统的用户)与用例来描述的。实际上每个用例定义了一个或若干参与者与系统间的交互序列。

2.2.1 用例的参与者

用例的参与者代表应用领域内的一类角色,在系统的外部与系统进行交互。一般地,用例参与者为人类用户,但在特定的应用环境中,系统外部也会有其他类型的参与者作为对人类参与者的补充。

在智能鱼缸案例中,对于移动端系统,考虑其用例的外部参与者包括以下两类。

- (1) 人类参与者: 用户。
- (2) 外部系统: 智能鱼缸系统的物联网云计算平台。

2.2.2 用例图

用例图(use case diagram)展现了用户和与其相关的用例之间的关系,是用于表示用户与系统之间交互的最简形式。用例图使得系统中不同类型的用户和用例都变得清晰明了。对于图 2-1 中的用例,用户均为主要参与者,智能鱼缸系统的物联网云计算平台作为次要参与者为用例提供数据支持。

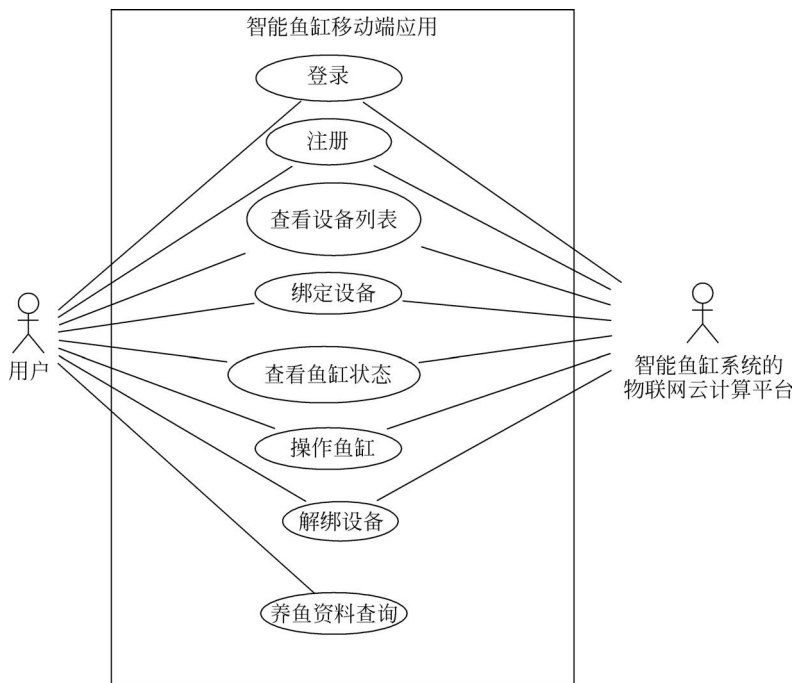


图 2-1 智能鱼缸移动端用例图

2.2.3 用例模型

对于每个用例,一般采用文档化的方式来进行描述,每个用例包含了下列关键内容:

- ①用例名称;
 - ②参与者名称;
 - ③用例的概要描述;
 - ④对用例中事件的主交互序列的描述;
 - ⑤对主序列的替代情况的描述。
- 结合智能鱼缸移动端应用的案例如下。