智慧生活——智能家居系统

设计文档

**变更记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **变更说明** | **作者** |
| 2025年7月10日 | 1.0 | 正式发布 | 李学敏 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1 引言 4](#_Toc25120)

[1.1 预期的读者 4](#_Toc19278)

[1.2 背景 4](#_Toc15448)

[1.3 系统架构 4](#_Toc31916)

[1.4 目的和作用 4](#_Toc30153)

[1.5 参考资料 4](#_Toc16087)

[2 外部设计 5](#_Toc24173)

[2.1 标示符与状态 5](#_Toc1947)

[2.2 使用它的程序 5](#_Toc19379)

[2.3 命名约定 5](#_Toc31284)

[2.4 设计约定 5](#_Toc4425)

[3 结构设计 6](#_Toc27504)

[3.1 概念结构设计 6](#_Toc10548)

[1. 系统总体架构图 6](#_Toc19440)

[2. 设备类型与状态定义 6](#_Toc19863)

[3.2 系统功能模块设计 7](#_Toc20799)

[3.2.1 北向应用模块设计 7](#_Toc6625)

[3.2.2 南向设备模块设计 8](#_Toc15579)

[3.3 通信协议设计 9](#_Toc13880)

[3.3.1MQTT通信主题结构图 9](#_Toc22638)

[3.3.2 MQTT主题设计 9](#_Toc26452)

[3.3.3消息格式设计 10](#_Toc14523)

[3.3.4设备交互流程图 12](#_Toc28206)

[4 运用设计 13](#_Toc9487)

[4.1 数据模型设计 13](#_Toc15917)

[4.1.1北向应用数据模型 13](#_Toc12095)

[4.1.2南向设备数据模型 13](#_Toc1465)

[4.1.3数据模型关系图 15](#_Toc10349)

[4.2 存储设计 16](#_Toc9303)

[北向应用存储 16](#_Toc16692)

[南向设备存储 16](#_Toc17087)

[4.3 系统实施 17](#_Toc22501)

[4.3.1 北向应用实施 17](#_Toc9357)

[4.3.2 南向设备实施 18](#_Toc19963)

[5.总结 21](#_Toc18545)

## 1 引言

### 1.1 预期的读者

系统设计人员，开发人员，质量控制人员，系统确认测试人员，系统维护人员以及项目管理人员。

### 1.2 背景

随着科技的发展和生活水平的提高，智能家居作为物联网的重要应用场景，正逐渐融入人们的日常生活。智能家居系统通过集成各类智能设备和传感器，实现对家居环境的智能化监测和控制，为用户提供更加便捷、舒适、安全的生活体验。

本项目基于鸿蒙系统和润和满天星系列Pegasus OpenHarmony开发套件，构建一套完整的智能家居系统，实现环境监测、设备控制和智能交互等功能，满足用户对智慧生活的需求。

### 1.3 系统架构

本系统采用典型的物联网三层架构：

1. **感知层**：由润和满天星开发套件及各类传感器组成，负责环境数据采集和设备状态模拟
2. **传输层**：基于MQTT通信协议，实现北向应用与南向设备之间的数据交互
3. **应用层**：基于鸿蒙系统开发的北向应用，包含设备管理、环境监测、AI交互等功能模块

### 1.4 目的和作用

本设计文档是对智慧生活——智能家居系统设计的定义，包括系统架构设计、功能模块设计、通信协议设计、数据模型设计等。旨在为系统开发提供明确的技术指导，确保系统实现符合需求规格说明书的要求。

### 1.5 参考资料

《智慧生活——智能家居系统需求分析》

《HarmonyOS应用开发指南》

《OpenHarmony开发文档》

《MQTT协议规范》

《润和满天星系列Pegasus开发套件用户手册》

## 2 外部设计

### 2.1 标示符与状态

系统名称：智慧生活——智能家居系统

北向应用版本：v1.0

南向设备版本：v1.0

通信协议：MQTT 3.1.1

设备状态标识：

isBind: false-未绑定, true-已绑定

online: false-离线, true-在线

isPowerOn: false-关闭, true-开启

设备特有状态(如空调): mode, fanSpeed, targetTemperature等

### 2.2 使用它的程序

本系统包含两部分程序：

1. **北向应用程序**：运行在搭载鸿蒙系统的终端设备上
2. **南向设备程序**：运行在润和满天星Pegasus OpenHarmony开发套件上

### 2.3 命名约定

类命名：采用大驼峰命名法，如DeviceManager、HomePageViewModel

方法命名：采用小驼峰命名法，如bindDevice()、saveDevices()

变量命名：采用小驼峰命名法，如deviceId、mqttClient

常量命名：采用全大写下划线分隔法，如DEVICE\_UPDATE\_EVENT、MQTT\_URL

MQTT主题命名：采用斜杠分隔的小写形式，如/devices/unbound/request

### 2.4 设计约定

本系统采用MVVM架构模式：

Model层：包含数据模型（如DeviceBase）和数据源（如DeviceDataSource）

View层：负责用户界面展示，采用ArkTS声明式UI开发

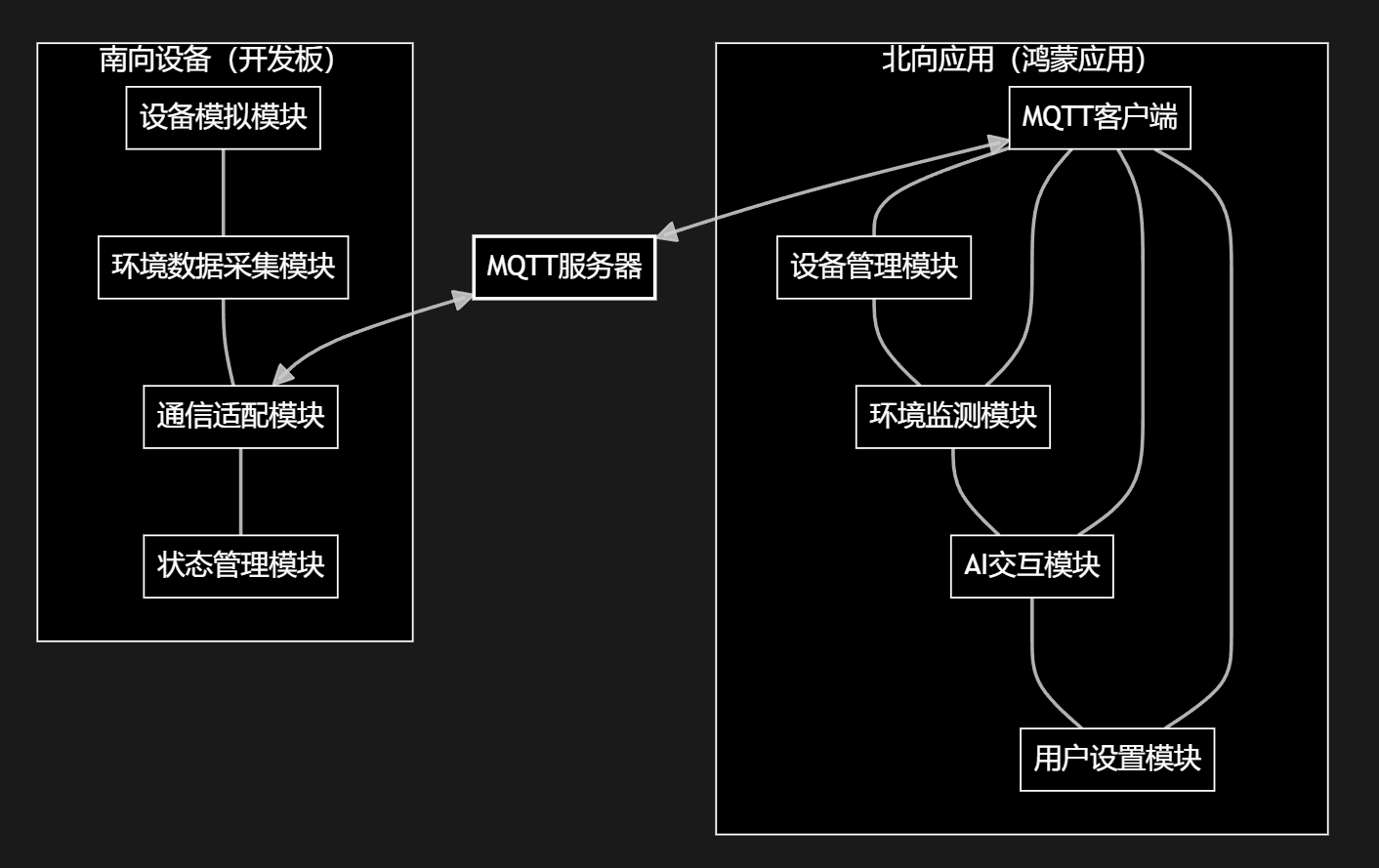
ViewModel层：负责业务逻辑处理，连接Model和View层

通信层：实现北向应用与南向设备之间的数据交互，基于MQTT协议

## 3 结构设计

### 3.1 概念结构设计

#### 1. 系统总体架构图



#### 2. 设备类型与状态定义

**设备基本属性**：

id: 设备唯一标识符

type: 设备类型（枚举值）

name: 设备名称

isBind: 绑定状态(false-未绑定, true-已绑定)

online: 在线状态(false-离线, true-在线)

isPowerOn: 开关状态(false-关闭, true-开启)

lastUpdate: 最后更新时间戳

**支持的设备类型**：

1.AIR\_CONDITIONER: 空调

currentTemperature: 当前温度

targetTemperature: 目标温度

fanSpeed: 风速（低、中、高）

mode: 模式（制冷、制热、除湿）

2.HUMIDIFIER: 加湿器

humidity: 当前湿度

targetHumidity: 目标湿度

waterLevel: 水位状态

3.GAS\_DETECTOR: 气体探测器

gasLevel: 气体浓度

alarmThreshold: 报警阈值

batteryLevel: 电池电量

4.SPEAKER: 智能音响

volume: 音量大小

isPlaying: 播放状态

currentTrack: 当前播放内容

### 3.2 系统功能模块设计

#### 3.2.1 北向应用模块设计

1.**设备管理模块**

设备发现：发送设备发现请求，获取可绑定设备列表

设备绑定：对指定设备执行绑定操作

设备控制：发送控制指令修改设备状态

设备展示：以卡片形式展示已绑定设备，支持拖拽排序

设备详情：查看设备详细信息和控制面板

2.**环境监测模块**

数据展示：展示温度、湿度、空气质量等环境数据

历史趋势：查看过去时段的环境数据变化趋势

异常告警：当环境数据超出阈值时触发提醒

3.**AI交互模块**

自然语言处理：解析用户输入，识别查询意图和关键信息

环境信息查询：根据用户问题返回当前环境数据

设备控制指令：解析并执行用户的设备控制意图

4.**用户设置模块**

个人信息：管理用户个人资料

系统设置：配置系统参数和通知选项

帮助中心：提供使用指南和常见问题解答

#### 3.2.2 南向设备模块设计

1.**设备模拟模块**

空调(AC)：模拟空调设备，支持温度控制、模式切换

音响(Speaker)：模拟智能音响，支持播放控制

环境监测系统(EMS)：采集并上报环境数据

2.**环境数据采集模块**

温湿度采集：通过AHT20传感器采集温湿度数据

气体浓度采集：通过MQ-2传感器采集可燃气体浓度

数据处理：对原始数据进行校准、转换和状态判断

3.**通信适配模块**

MQTT连接管理：维护与MQTT服务器的连接

主题订阅：订阅设备相关主题

消息发布：按照约定格式发布设备状态和环境数据

4.**状态管理模块**

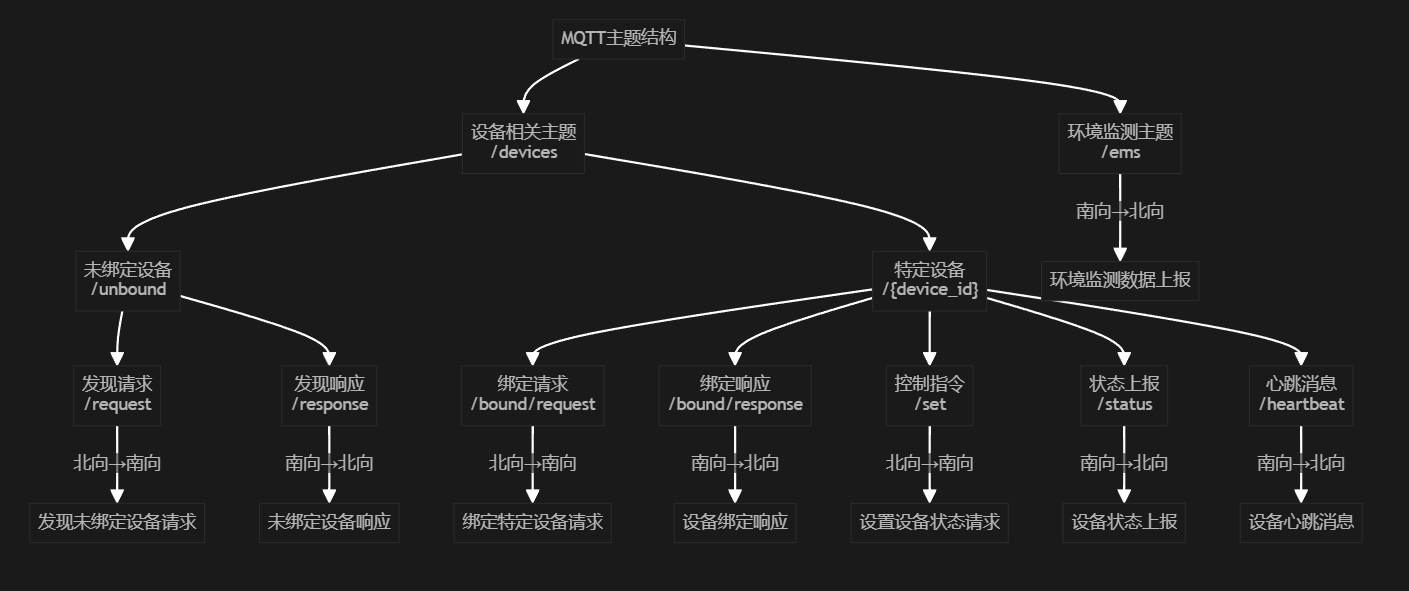
设备状态维护：管理设备在线状态、绑定状态等

心跳机制：定期发送心跳消息保持连接活跃

交互界面：通过OLED显示屏展示设备状态和操作提示

### 3.3 通信协议设计

#### 3.3.1MQTT通信主题结构图



#### 3.3.2 MQTT主题设计

1.**设备发现相关主题**

/devices/unbound/request：发现未绑定设备的请求

/devices/unbound/response：未绑定设备的响应

2.**设备绑定相关主题**

/devices/{device\_id}/bound/request：绑定特定设备的请求

/devices/{device\_id}/bound/response：设备绑定的响应

3.**设备控制相关主题**

/devices/{device\_id}/set：设置设备状态的请求

/devices/{device\_id}/status：设备状态的上报

4.**设备心跳相关主题**

/devices/{device\_id}/heartbeat：设备心跳消息

5.**环境数据相关主题**

/ems：环境监测数据的上报

#### 3.3.3消息格式设计

1.**未绑定设备响应消息格式**

{

"id": "dev101",

"type": "air\_conditioner",

"name": "Gree Air Conditioner",

"isBind": 0

}

2.**设备绑定响应消息格式**

{

"result": 0,

"message":

"success"

}

3.**环境数据消息格式**

{

"stat": 0, // 0: 正常, 1: 警告, 2: 危险

"gas": 300,

"temp": 25.5,

"hum": 60.0

}

4.**设备心跳消息格式**

"pong"

5.**设备控制消息格式（空调示例）**

{

"power": true,

"mode": 0, // 0: 制冷, 1: 制热, 2: 除湿

"temperature": 26,

"fanSpeed": 1 // 0: 低, 1: 中, 2: 高

}

#### 

#### 3.3.4设备交互流程图

## 4 运用设计

### 4.1 数据模型设计

#### 4.1.1北向应用数据模型

1.**基础设备模型（DeviceBase）**

属性：id, type, name, isBind, online, isPowerOn, lastUpdate

方法：getId(), getName(), getType(), setPowerStatus()等

2.**空调设备模型（AirConditionerDevice）**

属性：继承基础设备属性，增加currentTemperature, targetTemperature, fanSpeed, mode

方法：继承基础设备方法，增加getCurrentTemperature(), setTargetTemperature()等

3.**加湿器设备模型（HumidifierDevice）**

属性：继承基础设备属性，增加currentHumidity, targetHumidity, waterLevel

方法：继承基础设备方法，增加getCurrentHumidity(), setTargetHumidity()等

4.**气体探测器模型（GasDetectorDevice）**

属性：继承基础设备属性，增加gasLevel, alarmThreshold, batteryLevel

方法：继承基础设备方法，增加getGasLevel(), setAlarmThreshold()等

5.**环境数据模型（EnvironmentData）**

属性：temperature, humidity, gasLevel, timestamp, status

方法：getStatus(), getFormattedTimestamp()等

#### 4.1.2南向设备数据模型

1.**空调设备（AC）**

状态：isBind, isPowerOn, temperature, mode, fanSpeed

行为：响应绑定请求，控制请求，发送心跳

2.**智能音响（Speaker）**

状态：isBind, isPowerOn, isPlaying, volume

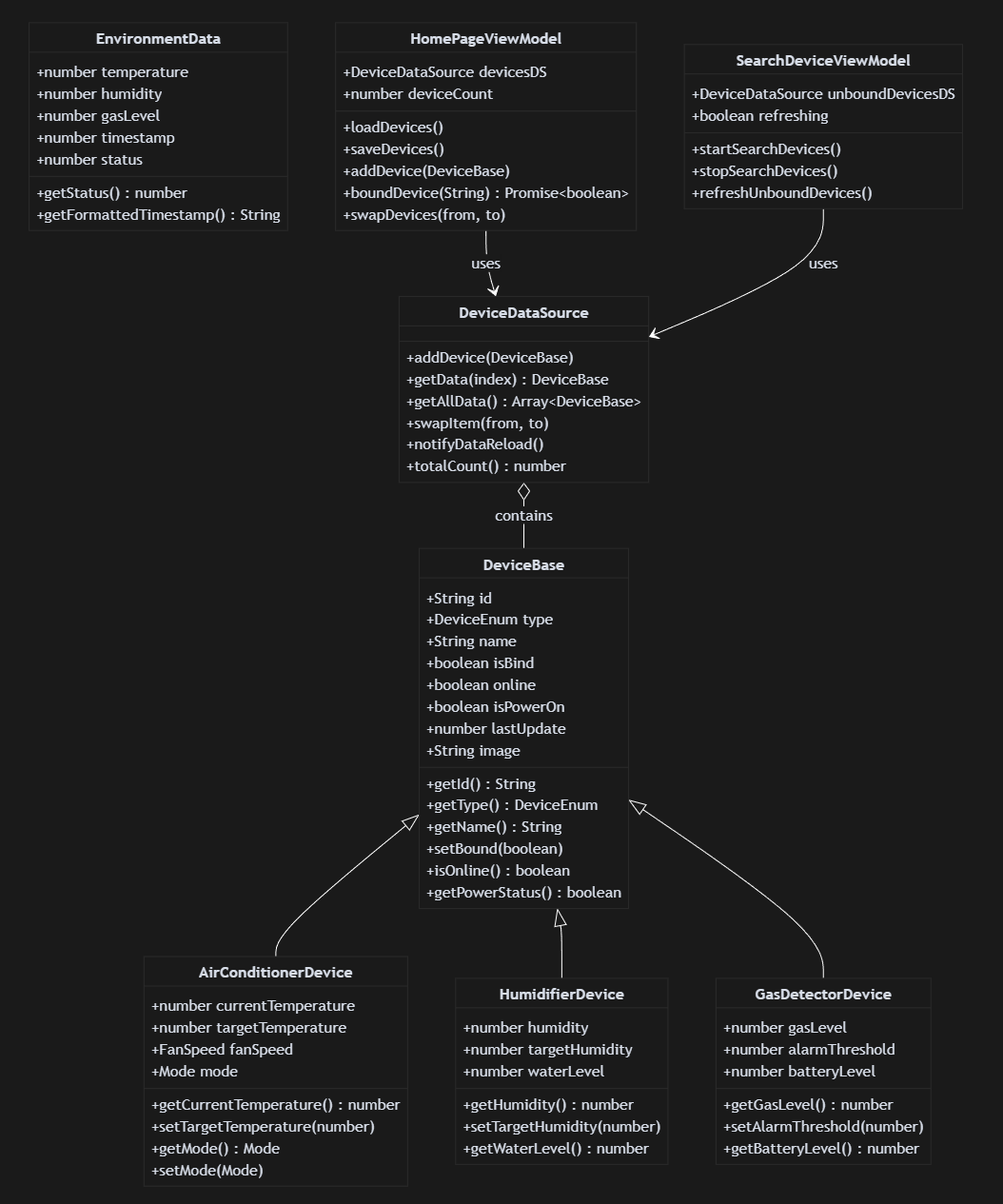
行为：响应控制请求，播放音乐，发送心跳

3.**环境监测系统（EMS）**

数据：temperature, humidity, gasLevel, status

行为：采集环境数据，判断状态，上报数据

#### 4.1.3数据模型关系图



### 4.2 存储设计

#### 北向应用存储

1.**设备数据存储**

存储内容：设备列表，设备详细配置

存储方式：首选项存储(Preferences API)

数据结构：JSON序列化的设备对象数组

操作接口：loadDevices(), saveDevices(), clearDevices()

2.**环境数据存储**

存储内容：历史环境数据记录

存储方式：关系型数据库(Relational Store)

数据表结构：timestamp, temperature, humidity, gasLevel, status

操作接口：insertRecord(), queryRecords(), deleteOldRecords()

3.**用户配置存储**

存储内容：用户偏好设置，通知选项

存储方式：首选项存储(Preferences API)

数据结构：键值对配置项

操作接口：loadPreferences(), savePreferences(), resetPreferences()

#### 南向设备存储

1.**设备配置存储**

存储内容：设备ID, 绑定状态

存储方式：Flash文件系统

数据结构：简单键值对配置文件

操作接口：load\_config(), save\_config()

2.**环境数据缓存**

存储内容：最近的环境数据

存储方式：内存缓存

数据结构：环境数据结构体数组

操作接口：update\_cache(), get\_latest\_data()

### 4.3 系统实施

#### 4.3.1 北向应用实施

1.**页面结构**

Index.ets：主页面，包含底部标签栏和导航容器

HomePage.ets：首页，展示已绑定设备列表和环境数据

SearchDevicePage.ets：设备搜索页面，展示可绑定设备

ProfilePage.ets：个人页面，展示用户信息和设置

AIPage.ets：AI交互页面，提供自然语言交互界面

2.**组件设计**

DeviceCard.ets：已绑定设备卡片，展示设备基本信息和控制界面

UnboundDeviceCard.ets：未绑定设备卡片，提供绑定功能

EnvironmentDataCard.ets：环境数据卡片，展示当前环境数据

AIConversationPanel.ets：AI对话面板，提供用户输入和反馈展示

3.**通信实现**

MQTTUtil.ets：MQTT通信工具类，处理连接、订阅、发布等操作

MqttMessageDispatcher.ets：消息分发器，根据主题分发消息到不同处理器

HttpUtil.ets：HTTP通信工具类，处理RESTful API请求

4.**视图模型**

HomePageViewModel.ets：首页视图模型，管理设备列表和数据加载

SearchDeviceViewModel.ets：设备搜索视图模型，管理设备发现和绑定

AIViewModel.ets：AI交互视图模型，处理语义理解和响应生成

EnvironmentViewModel.ets：环境监测视图模型，管理环境数据处理和展示

5.**特性实现**

设备拖拽排序：通过Grid组件的onItemDragStart和onItemDrop事件实现

消息分发机制：通过主题匹配将MQTT消息路由到对应处理函数

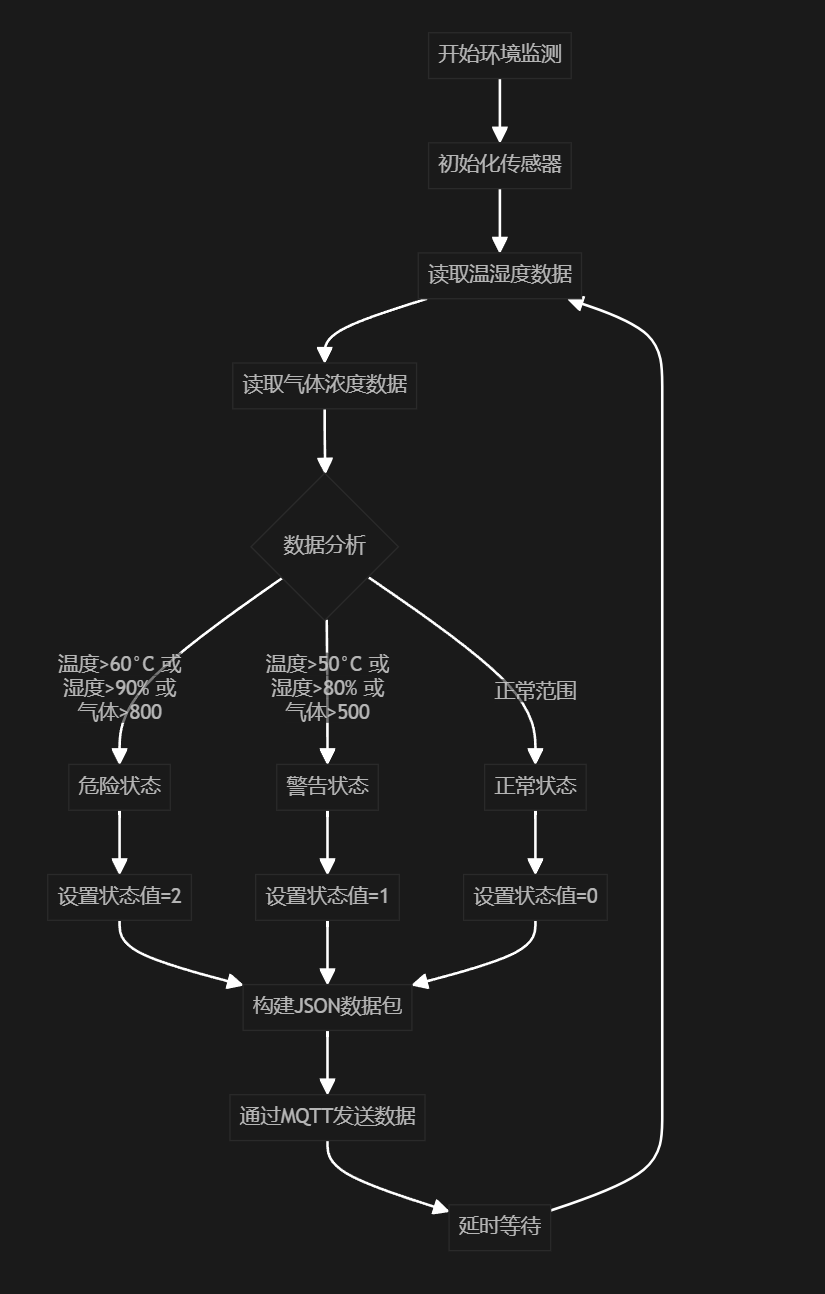
设备状态更新：通过事件总线(emitter)通知UI组件刷新

绑定流程管理：实现设备发现、确认绑定、等待确认、完成绑定的完整流程

环境数据采集：定时获取并分析环境数据，根据阈值判断环境状态

#### 4.3.2 南向设备实施

**1. 环境数据采集与处理流程图**



2.**设备模拟实现**

ac.c/ac.h：空调设备模拟，实现绑定流程和状态控制

speaker.c/speaker.h：智能音响模拟，支持音乐播放

ems.c/ems.h：环境监测系统，采集和上报环境数据

3.**硬件驱动层**

aht20.c/aht20.h：温湿度传感器驱动

mq\_2.c/mq\_2.h：气体传感器驱动

beep.c/beep.h：蜂鸣器控制，提供操作反馈

oled.c/oled.h：OLED显示屏驱动，显示设备状态和操作提示

key.c/key.h：按键输入处理，用于设备绑定确认等操作

4.**通信实现**

mqtt.c/mqtt.h：MQTT客户端实现，用于空调设备

mqtt1.c/mqtt1.h：MQTT客户端实现，用于音响和环境监测

wifi.c/wifi.h：WiFi连接管理，确保网络连接稳定

6.**任务管理**

device\_app.c：应用入口，初始化各模块并启动任务

空调任务：处理设备发现、绑定和控制请求

音响任务：处理音乐播放控制

环境监测任务：定期采集和上报环境数据

7.**安全性设计**

设备绑定确认：需要用户在设备端确认才能完成绑定

操作反馈：通过蜂鸣器和OLED显示提供操作反馈

状态校验：验证控制指令的合法性，防止非法操作

## 5.总结

本设计文档详细描述了智慧生活——智能家居系统的架构设计、功能模块设计、数据模型设计和实施方案，为系统开发提供了明确的技术指导。系统实现了设备管理、环境监测、AI交互和用户设置四大功能模块，满足了用户对智能家居系统的基本需求。通过北向应用与南向设备的协同工作，用户可以方便地管理家居设备，监测家居环境，实现智慧生活的体验。