

基于云平台的STM32的智能家居监测系统设计

雷庆东

上海工商职业技术学院, 中国·上海 201806

摘要: 传统的智能家居系统中难以兼顾用户对实时性、可靠性以及智能化的需求。本文通过云-边协同的方式设计并实现了一种智能家居监测系统。在边缘端, 通过使用 STM32 微控制器作为处理单元, 负责多种环境传感器的采集、滤波以及本地数据融合处理, 通过控制算法处理传感器采集的数据, 具备本地自动控制与异常判断能力, 降低了云端负载与通信延迟。云端平台负责海量监测数据的存储, 深度分析以及可视化, 并通过定制的规则引擎还能实现智能告警与多设备联动策略。经过测试, 这套系统能够对数据高效实时处理, 结合规则引擎还具备多种智能化需求。

关键词: 云-边协同; STM32; 智能家居; 监测系统; 物联网

Design of STM32 Smart Home Monitoring System Based on Cloud Platform

Lei Qingdong

Shanghai Industrial and Commercial Polytechnic, China Shanghai 201806

Abstract: In traditional smart home systems, it is difficult to balance users' demands for real-time performance, reliability, and intelligence. This paper designs and implements a smart home monitoring system through cloud-edge collaboration. At the edge, an STM32 microcontroller is used as the processing unit to handle the collection, filtering, and local data fusion of various environmental sensors. By processing the sensor data through control algorithms, it has local automatic control and anomaly detection capabilities, reducing the load on the cloud and communication latency. The cloud platform is responsible for the storage, in-depth analysis, and visualization of massive monitoring data. Through a customized rule engine, it can also achieve intelligent alerts and multi-device linkage strategies. After testing, this system can efficiently and promptly process data and, combined with the rule engine, meet various intelligent demands.

Keywords: Cloud-edge collaboration; STM32; Smart home; Monitoring system; Internet of things (IoT)

0 引言

伴随着物联网技术的进步以及生活品质的日益提升, 人们对家居环境的舒适度监测、安全方面的预警功能, 还有不同场景下的联动效果, 都提出了更为高要求。现有的智能家居存在两种模式, 一种是云端集中式架构, 所有的数据经上传至服务器进行处理, 不仅带宽高, 延迟大甚至会产生服务中断的问题; 另一种方案是纯本地化的控制, 虽然可以做到快速响应, 但是对数据的存储和深度的分析能力有限, 难以远程的管理与智能的策略的实施等。基于上述问题, 本文设计一种基于云-边的协同的智能家居的监测系统, 在硬件配置方面, 我们选用了具备高性能与低功耗特性的 STM32 微控制器, 并搭配 FreeRTOS 操作系统, 共同搭建起边缘计算节点。在云端层面集成了规则引擎、数据存储功能以及 Web 可视化展示模块, 旨在为构建高效、稳定的同时又具有智能家居的监测方案提供更

好的技术的参考依据。

1 系统架构设计

本系统总体架构如图 1 所示, 分为边缘感知层, 网络传输层以及云端应用层三个逻辑模块。

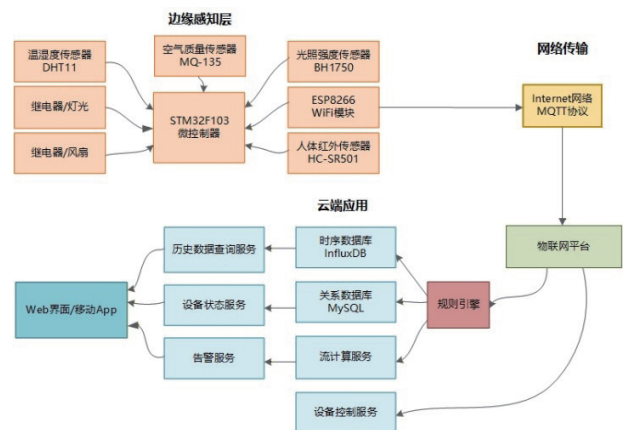


图 1 系统总体架构

1.1 边缘感知层

边缘感知层的核心是采用 ARM Cortex-M3 内核的 STM32F103 微控制器，该控制器集成了多种外设接口设备包括 GPIO、UART、SPI、I2C、ADC 等，能够同时外接多种传感器采集工作，具有较强的数据处理能力，高达 72MHz 的主频，能为智能家居监测系统提供强大和高效的边缘计算能力^[2]。在本系统设计中集成的传感器分别为：用于监测环境舒适度的温湿度传感器（DHT11）、评估空气质量的烟雾及有害气体传感器（MQ-135 与 MQ-2）、感知环境光照强度的数字光照传感器（BH1750）、以及用于安防与节能联动的人体红外传感器（HC-SR501）。根据传感器手册编写相应的驱动程序进行周期性数据采集，同时利用其强大的计算能力完成数据的初步滤波、校验和融合任务，形成结构化的环境状态信息。最后，通过 ESP8266 连接 Wi-Fi，将数据经过打包处理后，上传至云端应用层。

1.2 网络传输层

网络传输层采用搭载 TCP/IP 协议栈的 WIFI 模块。本文选用的是乐鑫公司生产的 ESP8266 模块，它支持 802.11 b/g/n 标准，内置 TCP/IP 协议栈，能够实现运行中最少地占用系统资源，同时采用轻量级的 MQTT 协议作为核心通信协议。MQTT 协议是基于发布 / 订阅模式的网络协议，能够在低带宽、高延迟以及不可靠的网络环境下提供实时可靠的安全消息传输^[3]。在本系统设计中边缘感知层将传感器数据处理成 JSON 格式再封装成 MQTT 消息包经 ESP8266 上传至云端的订阅控制器，云端控制器将接受的指令、配置或数据经解析后交给对应的规则引擎处理。边缘层只负责数据的处理、缓存和打包，网络层负责数据的传输，云端服务负责数据的存储、深度检测和可视化，这样模块之间相互独立工作降低了耦合，增强了系统的鲁棒性设计。

1.3 云端应用层

云端应用层是系统的“智能大脑”，部署在物联网云平台上，主要功能如下：

(1) 设备管理功能：提供设备的身份认证、生命周期管理（注册、激活、注销）、在线状态监控与设备影子（Device Shadow）服务，即使设备离线，也能保存其最新状态和待下发指令。

(2) 数据枢纽功能：实时接收并处理通过 MQTT 协议传来的原始数据。先解析、分类，再用对应的服务模块存储和分析。

(3) 数据存储与分析功能：处理后的数据会分开存储。带时间戳的传感器数据存到 InfluxDB，方便进行时间序列和做趋势分析。设备信息、用户数据、告警记录等结构化数据则存到 MySQL 数据库。

(4) 智能应用与服务功能：核心是提供一个 Web 界面，实时查看家居环境数据和历史数据，远程控制设备。规则引擎也支持自动化操作和智能告警功能。

2 功能模块设计与实现

2.1 传感器采集模块

传感器数据采集模块的核心任务在于实现对多源异构传感器的统一、可靠管理和数据处理。基于上述的思路，采用硬件抽象与接口统一的设计原则，提升系统的可扩展性与可维护性。再结合智能家居检测系统的所需的监测维度、精度、成本及功耗，选配了如表 1 所示的传感器阵列。

这些传感器涵盖了物理环境的多个关键参数的采集，并通过相应的通信接口与 STM32 微控制器连接。

2.2 网络传输模块

网络传输模块的核心任务保证 STM32 微控制器和云端应用层之间建立稳定、安全、高效的连接，并实现数据的可靠传输。基于上述需求，硬件模块采用搭载 STM32 主控器的 ESP8266 连接 WI-FI，通过配置 MQTT 消息协议发送消息，连接云端平台进行高效的异步通信。

2.3 云端应用层模块

云端应用层是系统的“智能大脑”与指挥中枢，负责对海量边缘终端数据进行聚合、持久化存储与智能分析，并向用户提供可视化交互界面。基于云原生架构思想，模块采用分层微服务模式，确保了系统的高可用性、可扩展

表 1 传感器阵列图

监测参数	传感器型号	通信接口	测量范围	精度	备注
温度/湿度	DHT11	单总线	温度：0-50° C 湿度：20-90%RH	温度：±2° C 湿度：±5%RH	集成式，成本低
空气质量	MQ-135	模拟电压	10-1000 ppm	-	需校准，检测 VOC / CO ₂
光照强度	BH1750	I2C	1-65535 lx	1 lx	数字输出，精度高
人体感应	HC-SR501	数字电平	探测距离 ≤7m	-	感应输出高电平
烟雾浓度	MQ-2	模拟电压	100-10000 ppm	-	可检测 LPG、烟雾

性与安全性。总体架构如图 2 所示。

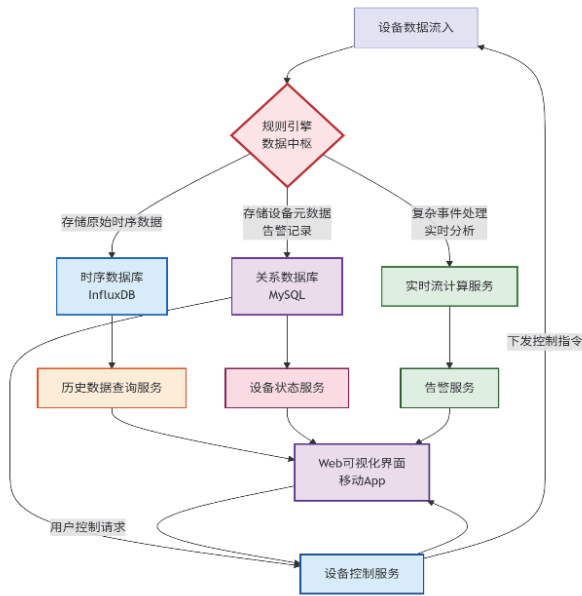


图2 云端系统架构图

规则引擎与数据流转模块负责数据的实时处理与分发是云端的数据“中枢神经系统”。首先云端应用层模块会监听物联网平台的消息总线的数据，当检测设备数据上报后，规则引擎会根据匹配的规则触发对应预配置的 SQL 式数据处理脚本，从而对数据进行过滤、转换和富化，然后再根据规则，将处理后的数据转发至不同下游服务或数据库包括：(1) 存储原始的、带时间戳的传感器数据时序数据库 InfluxDB。(2) 存储设备元数据、用户信息、告警记录等结构化数据关系型数据库 MySQL。(3) 处理复杂事件（如判断连续 1 分钟温度超标），实时流计算服务。(4) 提供设备在线 / 离线状态查询、最新数据获取的设备状态服务。(5) 提供按时间范围、设备 ID 等条件查询的历史数据查询服务。(6) 接收来自 Web 前端或 App 的控制请求、API 或更新设备固件、边缘设备指令下发的设备控制服务。(7) 基于规则引擎触发的告警事件，生成告警记录的告警服务。

3 实现

本章节将详细阐述系统的软件实现过程，包括边缘侧的软件流程、数据通信协议的应用以及云端核心服务的配置，旨在将第二章的模块化设计转化为可运行的软件系统。

3.1 硬件电路设计实现

本系统整个硬件平台主要由 STM32 最小系统电路、电源管理电路、多传感器接口电路以及 Wi-Fi 通信接口电路四部分组成。

3.1.1 传感器接口电路设计

硬件主控采用 STM32F103 微控制器，其中温湿度

采集 DHT11 采用单总线通信模式，其数据引脚连接至 STM32 的 GPIO 口，由于 DHT11 为开漏输出，硬件上需在数据线上接入一只 4.7kΩ 的上拉电阻，以确保总线空闲时处于高电平状态，保证数据传输的稳定性。根据手册空气质量监测传感器 MQ-135 和 MQ-2 内部加热丝需要 5V 供电，而 STM32 的 ADC 参考电压为 3.3V，因此传感器输出的模拟电压信号需经过电平转换电路处理后，接入给 STM32 的 PA1 和 PA2 通道输入，以防止过压损坏 MCU，同时确保采样精度。光照强度监测 BH1750 是通过 I2C 总线与 MCU 连接，其中 SCL 和 SDA 分别连接至 STM32 的 I2C 接口 PB6 和 PB7 引脚，并在两条总线上各接一个 4.7kΩ 的上拉电阻，以满足 I2C 协议的电气要求。人体红外感应 HC-SR501 输出为数字 TTL 电平，因此直接将其输出引脚连接至 STM32 的外部中断 IO 口，当检测到人体移动时输出高电平，触发 MCU 中断服务程序，实现低延迟的安防响应。

3.1.2 网络通信与电源模块设计

ESP8266 模块通过 UART 串口与 STM32 进行全双工通信。考虑到 ESP8266 在发送数据时瞬间电流较大，在模块的 VCC 电源端我们并联了一颗 470 μF 的大容量钽电容，防止因电压跌落导致模块复位或掉线。同时，模块的 CH_PD 引脚通过电阻上拉至高电平以使能芯片。

为了给 STM32、ESP8266 及其他数字传感器供电，电路中选择 AMS1117-3.3 线性稳压芯片，将 5V 转换为稳定的 3.3V 输出，并配有 LED 电源指示灯，方便直观判断电路供电状态。

3.2 边缘终端软件实现

考虑到搭载 STM32 微控制器的边缘终端需要对多个传感器采集处理，为避免单一采集任务阻塞导致系统延迟或逻辑死循环，主程序选择搭载 FreeRTOS 实时操作系统，通过多任务与消息队列机制来实现传感器数据采集、数据处理、网络通信等关键功能的并发执行与解耦。这种设计确保了即使在高负载或复杂工况下，关键任务（如异常报警）也能得到及时响应。核心任务流程与关键技术细节如下：

(1) 系统任务划分与调度：基于对系统精细的任务划分和对 FreeRTOS 的统一调度，使得系统的资源得到更好的利用。首先本文把云端接收到的控制指令作为高优先级任务。即云端应用层一旦接收到控制指令，立即中断当前低优先级任务，待控制任务完成后再继续执行未完成的低优先级任务。其次，将所有的传感器周期性采集任务作为

中优先级任务，保证边缘终端会以固定 2s 的时间间隔采集实时数据。最后低优先级任务负责数据预处理、细加工和上传，它们的执行时机取决于其他任务的状态。

(2) 传感器数据采集任务的精细化设计：该任务以 2 秒为周期定时执行，其内部流程遵循严格的顺序和错误处理机制。在共享硬件接口的场景下，各个任务需要通过竞争机制来获取信号量，这样做能够保证在同一个时间点上，只有一个任务可以获得总线的使用权，从而有效防止数据访问过程中出现冲突。在驱动层的具体实现里，针对每个传感器都配置了通信超时机制，同时还加入了 CRC 校验的重试功能，这样做的目的是为了进一步提升数据传输的可靠性。

(3) 数据预处理与融合任务的核心算法：该任务是边缘计算价值的集中体现，负责将原始数据转化为可靠、可用的环境状态信息。数据滤波降噪：为消除传感器读数的瞬时抖动和随机噪声，对连续采集的数据采用滑动平均滤波算法。以温度为例，算法实现如公式（1）所示：

$$T_{\text{filtered}} = (T_1 + T_2 + \dots + T_n) / n \quad (1)$$

式中， n 表示滤波窗口的大小。经过本系统测试，取 $n=5$ 。

(4) 状态判断与本地智能控制：STM32 会对传感器采集的数据滤波后与预设的安全阈值。若超出阈值上限，则立即置位相应的异常标志位同时执行本地控制任务。

(5) 数据封装与格式化：JSON 数据格式具有可读和易解析性，本文采用 JSON 格式将传感器数据、设备状态信息、电池电量、WiFi 信号强度以及时间戳数据统一封装，用于网络传输和数据处理。

3.3 云平台核心功能实现

云端功能的实现依赖于物联网平台提供的核心服务，关键实现如下：

(1) 规则引擎数据流转配置：规则引擎是数据流的中枢。通过在云平台控制台创建处理规则后，监听所有上报数据的主题包括 device、data、upload、add 并编写 SQL 语句提取有效字段，例如 device_id、timestamp、temperature、humidity，然后配置多个并行处理转发调度器一方面将实时数据全量存储到时序数据库 InfluxDB，用于长期趋势分析与报表生成，另一方面复杂事件转发至实时计算服务，用于复杂的智能告警判断。

(2) 智能告警实现：通过数据通过实时计算服务与规

则引擎处理的校验和判断后生成告警信息。具体判断逻辑是当规则引擎匹配到边缘端上报的温湿度、空气质量、光照强度、人体感应等结构化数据时，会执行事先配置好的 SQL 脚本进行多维度条件联合判断。当传感器检测到温湿度、气体浓度等参数超出预设阈值时，系统将同步触发设备联动控制与告警通知。

3.4 Web 应用层实现

为保证应用层的模块化设计思路，采用了前后端分离架构。后端采用 Spring Boot 框架提供配合 RESTful API 实现对数据库和物联网平台数据获取。前端使用 Vue.js 框架构建单页面应用，完成数据分析、规则引擎配置、可视化分析、实时流计算以及告警查看等功能。后台界面如图 3。

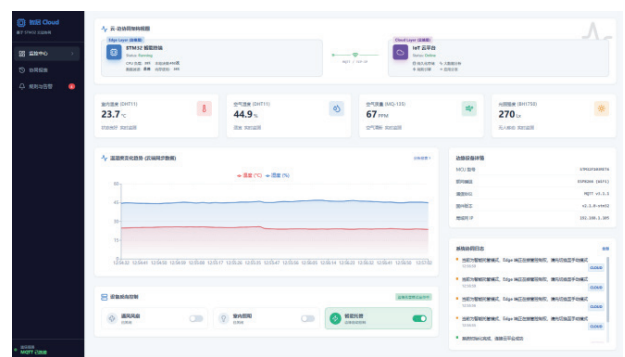


图 3 Web 应用端架构图

4 结语

凭借对传统的智能家居检测的局限性深入的分析，我们构建了一套一体化的硬件模块与云服务的智能家居的监测平台。该平台采用“云-边”协同的策略，利用 STM32 微控制器强大的边缘处理能力来应对实时采集与控制任务，同时借助云端强大的计算能力处理数据汇聚与分析。实验表明，这种架构不仅功能完善、响应迅速，而且在环境数据实时监测、远程操控及复杂规则报警等方面都非常可靠。相比单一的云端或本地处理模式，本系统显著降低了延迟，减轻了云端带宽负担，并且在网络不稳定的情况下依然能保持较高的自主运行能力。这种架构设计不仅提升了当前的系统性能，更为后续集成机器学习等智能化应用提供了坚实的基础设施^[5]。

参考文献：

[1] 宋妮璐. 基于 STM51 的智能家居控制系统开发与研究[J]. 软件, 2025,46(11):50-52.
 [2] 张紫燕, 洪陈旭, 彭元等. 基于物联网技术的智能

家居系统设计与实现[J]. 物联网技术, 2025,15(19):86-89.

DOI:10.16667/j.issn.2095-1302.2025.19.018.

[3] 王沈皓, 戴文君, 陈思哲等. 一种基于 STM32 的云端服务器设计[J]. 常州工学院学报, 2020,33(01):32-36.

[4] 胡彩刚. 基于物联网的智能灯光与环境自适应室内

装饰系统设计[J]. 居舍, 2025,(30):18-20.

[5] 林帅志. 面向物联网的智能家居安全监控系统设计[J]. 智能物联技术, 2025,57(05):46-49.DOI:10.26921/j.cnki.2096-6059.2025.05.009.