

基于Hi3863的多模星闪网关系统设计

安皓楠, 白浩洁

北京华清远见科技发展有限公司, 研发中心, 北京

收稿日期: 2025年11月12日; 录用日期: 2026年1月16日; 发布日期: 2026年1月29日

摘要

随着物联网技术发展, 智能家居、工业互联等场景对无线通信提出低延迟、高可靠、多设备连接和低功耗的要求。星闪(SLE)作为我国自主研发的短距无线技术, 为物联网碎片化连接提供了解决方案。本文基于海思Hi3863芯片设计多模星闪网关系统, 实现星闪与Wi-Fi异构网络的互联, 构建边缘数据枢纽。本文首先分析了多模网关的需求, 提出了以Hi3863为主控、集成了星闪与Wi-Fi通信模组的硬件架构。系统采用分层软件设计, 底层基于LiteOS操作系统抽象层进行任务调度与资源管理; 中间层分别实现了星闪协议栈的设备发现、连接管理、数据传输功能, 以及基于LWIP的Wi-Fi TCP/IP通信功能; 应用层核心设计了一个智能数据转发引擎, 通过解析数据包中的网络与设备标识, 实现星闪网络与Wi-Fi网络间数据的无缝转换与可靠转发。测试结果表明, 本系统能够稳定维护多个星闪设备与Wi-Fi客户端的并发连接, 成功实现了跨异构网络的数据透传, 有效验证了基于Hi3863构建低成本、高性能多模星闪网关系统的可行性。

关键词

Hi3863, 星闪, 多模网关, 物联网, 异构网络

Design of a Multi-Mode SparkLink Gateway System Based on Hi3863

Haonan An, Haojie Bai

Research and Development Center, Beijing Huaqingyuanjian Technology Development Co., Ltd., Beijing

Received: November 12, 2025; accepted: January 16, 2026; published: January 29, 2026

Abstract

With the advancement of IoT technology, applications such as smart homes and industrial interconnection require wireless communications to meet demands for low latency, high reliability, multi-device connectivity, and low power consumption. As a short-range wireless technology independently developed in China, SparkLink provides an effective solution to fragmented connectivity in IoT. This

paper designs a multi-mode SparkLink gateway system based on HiSilicon's Hi3863 chip, enabling interconnection between SparkLink and Wi-Fi heterogeneous networks to establish an edge data hub. The study begins by analyzing the requirements of multi-mode gateways and proposes a hardware architecture with Hi3863 as the main controller, integrating both SparkLink and Wi-Fi communication modules. The system adopts a layered software design: the underlying layer utilizes the LiteOS operating system abstraction layer for task scheduling and resource management; the middle layer implements SparkLink protocol stack functions including device discovery, connection management, and data transmission, as well as Wi-Fi TCP/IP communication based on LWIP; the core application layer features an intelligent data forwarding engine that enables seamless and reliable data transmission between SparkLink and Wi-Fi networks by parsing network and device identifiers in data packets. Test results demonstrate that the system can stably maintain concurrent connections with multiple SparkLink devices and Wi-Fi clients, successfully achieving cross-heterogeneous-network data transparent transmission. This effectively validates the feasibility of constructing a low-cost, high-performance multi-mode SparkLink gateway system based on Hi3863.

Keywords

Hi3863, SparkLink, Multi-Mode Gateway, Internet of Things (IoT), Heterogeneous Network

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着物联网技术在智能家居、工业互联等领域的深化应用，场景对无线通信技术提出了低延迟、高可靠、多设备连接与低功耗的更高要求[1]。星闪作为我国自主研发的新一代短距无线通信技术，通过融合创新，在关键性能指标上超越了传统蓝牙与 Wi-Fi，为破解物联网碎片化连接难题提供了优质的底层技术方案。然而，一项新技术的规模化部署有赖于成熟的生态系统，其中，能够桥接新兴网络与现有网络的网关设备至关重要。

在此背景下，本研究基于海思 Hi3863 芯片设计多模星闪网关系统。Hi3863 芯片集成了星闪与 Wi-Fi 通信能力，具备高性能处理核心与丰富外设接口，为构建紧凑型、低成本的边缘数据枢纽提供了理想的硬件平台。本研究的核心意义在于，通过设计并实现一个星闪与 Wi-Fi 异构网络融合的网关，旨在打通两种网络的通信壁垒，构建高效的边缘数据枢纽。这能够满足智能家居场景下对海量传感器数据本地汇聚与响应的需求，对推动星闪技术的产业应用与构建自主可控的物联网体系具有重要的实践价值。

本研究的主要内容包括基于 Hi3863 的星闪-Wi-Fi 双模网关的软硬件一体化设计与实现，其核心创新点体现在异构网络数据转发机制上。

本研究首先完成了以 Hi3863 为主控的硬件平台设计，并采用了分层的软件架构。底层基于操作系统抽象层进行任务与资源调度；中间层并行实现了完整的星闪协议栈(涵盖设备发现、连接管理与数据传输)与基于 LWIP 的 Wi-Fi TCP/IP 通信功能；应用层则集成了智能数据转发引擎等核心逻辑。

本研究提出并实现了一套高效的异构网络数据转发机制。该机制首先定义了网关内部的统一数据模型，实现了星闪与 Wi-Fi 两种异构网络协议间的无缝转换。进而，设计了一种基于标识解析的智能路由算法：数据转发引擎通过解析数据包中预置的网络类型标识与设备类型标识，动态查询设备连接状态表，从而精准地将数据路由至目标星闪连接或 Wi-Fi Socket。该机制有效解决了异构网络间的通信壁垒，确保了数据在双模网络间可靠、低延迟的透传，为多模网关的核心功能实现了关键技术支撑。

2. 系统架构设计

2.1. 网络架构

本系统旨在设计并实现一个以海思 Hi3863 为核心的多模无线网关。该网关作为异构网络融合的枢纽, 主要承担两大功能: 一是实现星闪 SLE 与 Wi-Fi 两种异构无线网络间的协议转换与数据转发; 二是作为边缘节点, 通过串口与上层主控 MCU 进行数据交互, 实现更复杂的业务逻辑或接入更广阔的网络。系统采用分层与模块化的设计思想, 确保了功能的清晰性、稳定性与可扩展性。

本系统采用以 Hi3863 网关节点为中心的星型网络拓扑, 如图 1 所示。该拓扑结构简单、可靠, 易于管理, 非常适合物联网中设备与网关的通信模型。

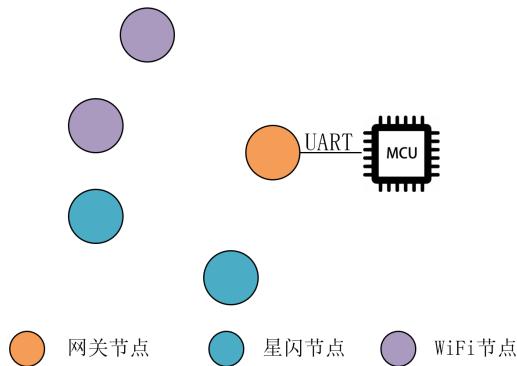


Figure 1. Network architecture

图 1. 网络架构

Hi3863 多模网关节点是整个网络的唯一核心。

星闪 SLE 节点为各类基于星闪技术的传感器(如温湿度、门磁)或执行器(如智能开关)。它们直接与 Hi3863 网关建立星闪 SLE 连接, 形成一个独立的星闪 SLE 子网。

Wi-Fi 节点为各类基于 Wi-Fi 技术的传感器或执行器。它们直接与 Hi3863 网关建立的 Wi-Fi Socket 连接进行通信, 形成一个 Wi-Fi 子网。

Hi3863 网关通过 UART 串口与一个功能更强大的上位主控 MCU 相连。此设计将复杂的多模无线通信功能与高级应用处理解耦, 网关专注于高效、低延迟的数据汇聚与转发, 而主控 MCU 则负责数据处理、决策、显示或通过无线网络/有线网络等方式上传至云端。

该星型拓扑实现了所有通信都必须经由网关中心节点, 为集中式的设备管理、数据路由和网络监控提供了天然的便利。

2.2. 硬件构成

本系统的硬件围绕 Hi3863 主控芯片构建。Hi3863 系列是 2.4GHz Wi-Fi 6 星闪多模 IoT SoC 芯片[2], 采用高度集成的设计理念, 实现了星闪与 Wi-Fi 双模通信的硬件基础。其集成 IEEE 802.11 b/g/n/ax 基带和 RF 电路, 包括功率放大器 PA、低噪声放大器 LNA、RF balun、天线开关以及电源管理模块等; 支持 20 MHz 频宽, 提供最大 114.7 Mbps 物理层速率, 支持更大的发射功率和更远的覆盖距离; 支持星闪 SLE 1 MHz/2 MHz/4 MHz 频宽、SLE1.0 协议、支持 SLE 网关功能, 最大空口速率 12 Mbps。

该芯片集成了高性能的 32 位 RISC-V 架构处理器核心, 运行频率最高可达 260 MHz, 内置 606 KB SRAM 和 300 KB ROM, 为复杂的协议栈运行和数据转发处理提供了充足的计算资源和存储空间。

该芯片包含丰富的外设接口, 包含 SPI、I2C、I2S、UART、GPIO 等, 方便与传感器和其他芯片进

行交互。

在实际的部署过程中, 网关节点放置在华清远见的网关底板上面。该底板包含无线模块接口和一个 STM32F103, 如图 2 所示。



Figure 2. Gateway hardware
图 2. 网关硬件

星闪节点和 Wi-Fi 节点放置在节点底板上面。该节点底板可插接多种通信模组和多种传感器。

2.3. 软件架构

本系统的软件架构采用分层设计理念, 构建了一个模块化、可扩展的软件体系。如图 3 所示, 整个软件栈自底向上分为驱动层、操作系统抽象层、协议栈层和应用层, 各层之间通过清晰的接口进行通信, 确保了系统的稳定性和可维护性。

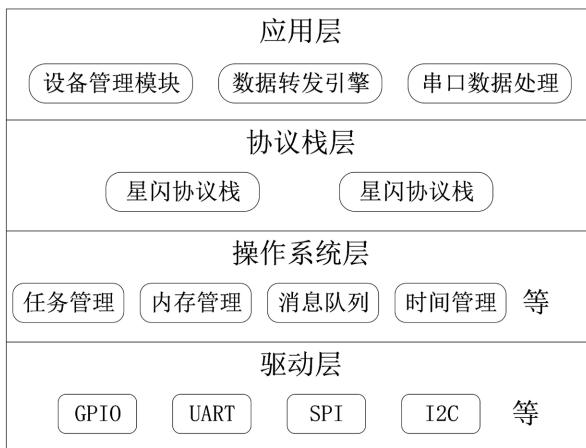


Figure 3. Network architecture
图 3. 软件架构

驱动层直接与硬件交互, 为上层提供统一的硬件操作接口。该层包含了多个关键外设的接口, 如 GPIO、UART、SPI、I2C 等, 可获取传感器数据、控制执行器。

操作系统层内部采用 OpenHarmony 架构下的 LiteOS 操作系统, 并将其封装成 OSAL 接口。该层通过封装实时操作系统的功能, 为上层应用提供了与具体操作系统内核无关的统一编程接口, 有效增强了软件的可移植性与可维护性。

协议栈层是双模通信能力的实现核心, 并行运行着星闪与 Wi-Fi 两套独立的协议栈。星闪协议栈完整实现了星闪技术的设备发现、连接建立与数据收发功能[3]。Wi-Fi 协议栈基于轻量级 IP 协议栈 LWIP [4], 实现了完整的 IP、TCP、UDP 等网络协议, 使网关能够接入本地局域网。网关作为 TCP 服务器, 创建监

听 Socket 并接受 Wi-Fi 客户端的连接请求。为每个连接的客户端维护一个 Socket 描述符，并通过标准的 Socket API (如 send、recv) 进行数据收发。

应用层是系统业务逻辑的核心，它整合下层服务，实现了网关的智能数据路由与设备管理功能。设备管理模块负责维护两个核心数据结构——星闪设备连接表(sle_conn_device)和 Wi-Fi 设备连接表(wifi_conn_device)。数据转发引擎作为网关的智能中枢，通过解析数据包中的网络与设备标识，动态查询连接表并调用相应协议栈 API，实现了星闪与 Wi-Fi 异构网络间的无缝数据透传。串口命令处理模块定义并实现了一套与上位主控 MCU 的通信协议。该模块解析来自串口的控制指令，并将其转换为对网关内部模块的操作；同时，也将网关的状态信息(如连接设备数、网络异常等)封装成数据帧，上报给上位 MCU。

3. 软件系统设计与实现

3.1. Wi-Fi AP 初始化及数据接收任务

Wi-Fi AP 初始化及数据接收任务负责启动设备的无线热点功能并建立 TCP 服务器。该任务首先等待 Wi-Fi 硬件初始化完成，然后配置并启用 SoftAP 模式，设置热点的 SSID、密码、IP 地址等网络参数，同时创建 TCP Socket 进行端口绑定和监听。在任务运行期间，它会创建专门的 TCP 连接接受任务来处理客户端连接请求，并持续轮询所有已连接的 Wi-Fi 客户端，接收它们发送的数据并存入缓冲区，为后续的数据转发处理提供支持。

主要流程：

```

while (wifi_is_wifi_initiated() == 0)      // 等待Wi-Fi初始化完成
example_softap_function()                  // 启动SoftAP和TCP服务器
创建 TCP_accept_Task 任务
while (1) {
    for(所有Wi-Fi客户端) {
        wifi_device_recv()                  // 轮询接收数据
        osal_mdelay(100)
    }
}

```

3.2. TCP 连接接受任务

TCP_accept_Task 作为 TCP 连接接受任务，在 Wi-Fi AP 网络环境中承担着客户端连接管理的核心职责。该任务持续监听服务器端口，当检测到新的连接请求时，通过系统调用接受连接并为每个客户端动态分配设备管理结构体，记录其套接字描述符和网络地址信息。同时，该任务维护着全局连接计数和设备状态跟踪，确保系统能够同时处理多个客户端的并发访问，为后续的数据通信建立稳定的网络链路基础。

主要流程：

```

while (1) {
    dev_id = FindFreeDeviceIndex()      // 查找空闲设备槽位
    if (dev_id != -1) {
        new_fd = accept(sock_fd, ...)  // 接受新连接
        if (new_fd != -1) {
            初始化 wifi_conn_device[dev_id] // 记录连接信息
            g_wifi_conn_num++           // 连接计数增加
        }
    }
    osDelay(100)                      // 延时等待
}

```

3.3. 星闪设备管理任务

星闪设备管理任务在异构网络架构中负责实现星闪协议的设备发现、连接建立与数据路由功能。该任务在系统启动后首先完成硬件初始化与协议栈配置，随后进入持续监听状态，通过消息队列机制接收来自串口的外设数据。在任务核心处理逻辑中，它解析数据包中的网络标识与设备类型信息，动态匹配已连接的星闪终端设备，并基于 SSAPC 协议将数据准确转发至目标设备。同时，该任务与设备扫描模块协同工作，维护着星闪设备的连接状态表，确保数据传输的可靠性与实时性。

主要流程：

```

osal_msleep(5000)           // 等待Wi-Fi初始化完成
app_uart_init_config()      // 初始化UART
bsp_init()                  // 初始化硬件(LCD, GPIO等)
sle_client_init()           // 初始化星闪协议栈

while (1) {
    osal_msg_queue_read_copy() // 从消息队列读取数据
    sle_wifi_send_data(msg_data) // 转发数据到SLE/Wi-Fi设备
}

```

`sle_client_init()`函数是星闪通信系统的核心初始化模块，主要完成星闪协议栈的完整启动与回调机制注册。该函数内部涉及的接口如图 4 所示。

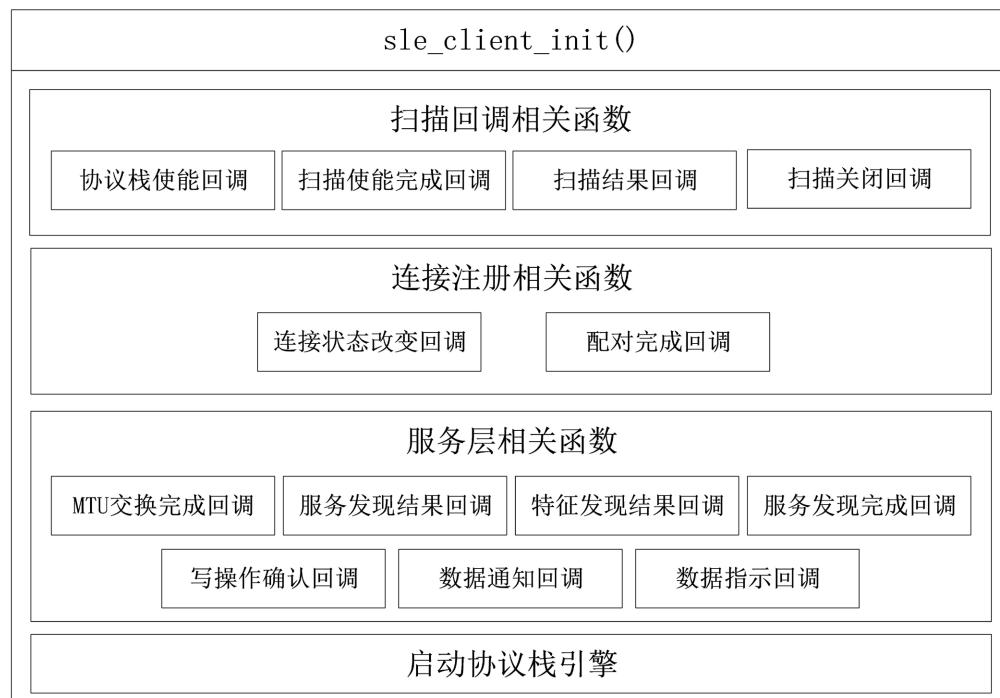


Figure 4. SLE protocol stack related callback functions

图 4. SLE 协议栈相关回调函数

该函数首先注册设备扫描相关的回调函数，包括协议栈使能通知、扫描启停状态反馈及设备发现结果处理，为后续的设备发现提供事件响应基础。接着建立连接管理回调，实时监控设备连接状态变化与配对完成事件，动态维护设备连接表。最后配置服务层通信回调，实现 MTU 协商、服务发现、特征值读

写及数据通知等高级通信功能的异步处理。在完成所有回调注册后，函数调用 `enable_sle()` 正式启动星闪协议栈，并将网络状态指示灯设置为就绪状态，标志着星闪通信系统已全面进入可操作状态。

数据通知回调函数作为星闪协议栈的数据接收通知回调，在系统中负责处理来自星闪设备的异步数据上报。当远程设备通过通知机制发送数据时，该函数将接收到的数据安全拷贝至专有的星闪串口缓冲区，并更新缓冲区的写入位置索引。最后，它会调整缓冲区管理指针并为后续数据包预留存储空间，从而确保连续数据流的可靠接收与处理，为上层应用提供了完整的数据帧。

3.4. 数据发送处理任务

数据发送处理任务在通信系统中承担着数据汇聚与转发的关键职能。该任务以固定周期轮询检查星闪与 Wi-Fi 两个网络通道的接收缓冲区，当检测到有效数据包时，按照预定义的通信协议进行格式解析与校验，并通过串口接口将整合后的数据发送至外部处理单元。其设计采用双缓冲区管理机制，能够并行处理多路数据流，确保来自不同网络协议的传感器数据能够实时、有序地传输，为上层应用提供统一的数据接入服务。

主要流程：

```
osal_msleep(5000) // 等待Wi-Fi初始化完成
while (1) {
    DataHandler(&Usart_SLE_Control, 0) // 处理SLE接收数据
    DataHandler(&Usart_WIFI_Control, 0) // 处理Wi-Fi接收数据
    osal_mdelay(50) // 50ms周期处理
}
```

3.5. 串口数据接收回调函数

由于网关需要通过 UART 接收来自外部处理单元，因此需要使用串口接收回调函数接收来自上位机的数据。其负责将接收到的原始数据安全地封装并传递到消息队列中。

3.6. 数据缓存机制

为高效管理来自星闪 SLE 与 Wi-Fi 异构网络的多路异步数据流，并应对数据接收与处理速度不匹配的问题，本系统设计并实现了一套基于环形指针队列的双缓冲区数据缓存机制。该机制确保了在高并发数据流入时的系统稳定性，并提供了高效、可靠的数据存取路径。

3.6.1. 缓存结构设计

系统为星闪 SLE 和 Wi-Fi 分别独立定义了完全相同的缓存管理结构，实现了数据隔离与并行处理。核心数据结构包括：

数据缓冲队列：一个静态分配的字节数组，作为存储原始数据的物理内存区域。

指针缓冲队列：一个 USART_Control 类型的结构体，用于管理缓冲区的读写状态。

```
typedef struct
{
    uint32_t RxBufferCounter; // 指示下一个字节应存入数据缓冲队列的哪个位置
    LocationPtr RxLocationPtr[LocationPtrNum]; // 管理指针缓冲队列的结构体数组
    LocationPtr *RxDataInPtr; // In指针：标记指针缓冲队列插入数据的位置
    LocationPtr *RxDataOutPrt; // Out指针：标记指针缓冲队列提取数据的位置
    LocationPtr *RxDataEndPrt; // End指针：标记指针缓冲队列的末尾，防止数组越界
} USART_Control;
```

其中 LocationPtr 类型如下所示。

```
typedef struct
{
    uint8_t *StartPtr; // 帧起始指针
    uint8_t *EndPtr; // 帧结束指针
} LocationPtr;
```

该缓存结构如图 5 所示。

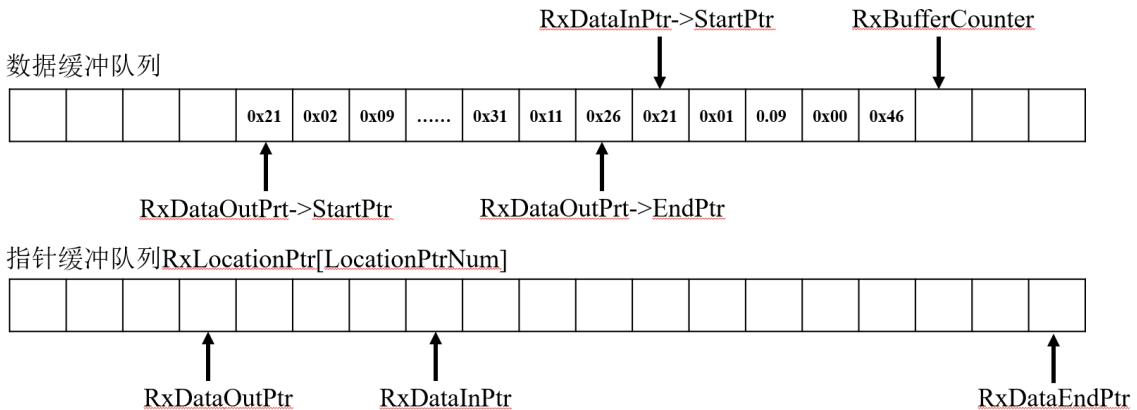


Figure 5. Buffer structure

图 5. 缓存结构

在系统初始化阶段，对上述结构体进行初始化，建立缓冲区与指针队列之间的映射关系，并将读写指针均指向数据缓冲队列起始位置。

3.6.2. 数据接收与缓存流程

当串口空闲中断触发，表明一个完整数据包接收完毕时，系统会调用相应的接收完成回调函数。

封包入队：回调函数首先将当前数据包在指针缓冲队列的结束地址写入 RxDataInPtr 的 EndPtr 指针所指向的位置，从而完成一个数据包的封包操作。然后让 RxDataInPtr 指针往后移动一个 LocationPtr 数据结构长度。

移动写入指针：随后，RxDataInPtr 的 StartPtr 指针移动到之前数据包结束位置 EndPtr 指针的后面，指向下一个空闲数据元素。

队列循环处理：

情况一：若移动后的 RxDataInPtr 已到达队列末尾，则将其重置指向队列头部，实现环形队列的循环使用。

情况二：函数检查物理缓冲区的剩余空间。若剩余空间足以容纳下一个最大数据包，则设置下一个数据包的起始地址为当前写入偏移；若剩余空间不足，则将写入偏移复位至缓冲区头部，从头开始循环利用缓冲区，防止数据覆盖紊乱。

3.6.3. 数据处理与转发流程

数据处理任务 DataHandler 被周期性地调度执行，其负责检查并处理缓存中的数据包。

数据包检查：任务通过比较 RxDataInPtr 和 RxDataOutPrt 判断是否有未处理的数据包。通过辅助函数 get_untreated_datapacket 可精确计算未处理数据包的数量。

拥塞处理模式: 当未处理数据包数量超过设定的阈值时, 系统进入拥塞处理模式。在此模式下, 任务将快速连续地处理队列中的大量积压数据, 直至清空队列, 以尽快恢复系统正常响应。

常规处理模式: 在正常负载下, 任务每次执行仅处理一个数据包。它首先验证数据包的协议头标识, 若为有效数据, 则通过相应的串口发送出去; 若为无效数据, 则直接丢弃。

移动读取指针: 无论数据有效与否, 在处理完一个数据包后, RxDataOutPrt 指针都会向后移动一位。同样, 当指针到达队列末尾时, 会重置到队列头部。

该双缓冲区机制将数据的高速接收与相对低速的处理解耦开来, 通过异步方式避免了数据丢失, 确保了在多模网关复杂通信环境下数据转发的完整性与可靠性。

此流程实现了数据包在逻辑队列中的有序存入, 并通过物理缓冲区的循环使用, 最大限度地利用了有限的内存资源。

3.7. 协议格式

为确保异构网络(星闪与 Wi-Fi)中数据帧的统一处理与可靠传输, 本系统设计了一套精简且高效的应用层通信协议。该协议定义了数据帧的统一格式, 使得网关能够正确解析来自不同网络节点的数据, 并根据协议中的字段进行路由和决策。节点与网关之间数据传输的格式如表 1 所示。

Table 1. Protocol format

表 1. 协议格式

标识	数据 长度	数据 偏移	数据 类型	网络 类型	设备 地址	传感器 类型	数据域	CRC8 校验和
1byte	1byte	1byte	1byte	1byte	2byte	1byte	...	1~255byte	1byte

各字段的详细定义与功能如下:

标识(Header): 固定为 0x21, 用于标识一个数据帧的开始。

数据长度(Data Length): 指示数据域的实际字节数。例如, 当数据域包含 1 字节数据时, 该字段值为 0x01。

数据偏移(Data Offset): 定义了从数据包起始位置(包头)到数据域开始位置的字节偏移量。该设计为协议头部保留了可扩展性, 未来若需增加新的固定字段, 只需调整偏移量即可保持兼容。

数据类型(Data Type): 用于区分数据包的功能属性。例如, 0x00 代表承载传感器读数的普通数据包, 而 0x01 代表用于设备配置、命令下发的控制命令包。

网络类型(Network Type): 标识数据包来源或目标节点的网络类型。该字段是网关实现双模数据路由的核心依据, 网关根据此值决定将数据通过星闪(SLE)或 Wi-Fi 网络转发。

设备地址(Device Address): 一个 2 字节的唯一标识符, 用于区分同一网络内的不同终端设备, 遵循传感器类型表中的预定义地址, 传输顺序为高位在前, 低位在后(Big-Endian)。

传感器类型(Sensor Type): 参照传感器类型表, 标识设备所搭载的传感器种类(如温湿度、光照强度等), 便于网关对数据进行分类处理与应用。

数据域(Data Field): 承载有效载荷的核心部分, 长度可变(1~255 字节), 内容可为传感器采集的实时数据或控制命令的具体参数。

CRC8 校验和(CRC8 Checksum): 用于保障数据传输的完整性。校验范围覆盖从包头至数据域结尾的所有字节, 采用生成多项式 $x^8 + x^2 + x + 1$ (CRC-8/MAXIM)进行计算。接收方通过验证校验和来判断数据包在传输过程中是否出错。

4. 实验部署

为验证本系统的功能与性能, 我们搭建了完整的实验环境, 对多模网关的通信能力、数据转发效率及系统稳定性进行了测试。

实验环境包括:

硬件平台: Hi3863 网关、星闪传感器节点(温湿度、光照)、Wi-Fi 传感器节点、STM32F103 上位主控板。

软件环境: LiteOS 操作系统、星闪协议栈 V1.0、LWIP 2.1.2。

网络拓扑: 采用星型结构, 网关同时接入 8 个星闪设备与 4 个 Wi-Fi 客户端, 主控 MCU 通过 UART 与网关交互。

通过实验部署验证了以下内容:

双模接入能力: 在 30 米室内距离内, 网关成功同时建立星闪与 Wi-Fi 连接, 设备识别率 100%, 星闪的连接建立时间小于 2 秒。

数据透传测试: 在 30 米室内距离内, 星闪传感器数据经网关转发至外部处理单元, 反向控制指令亦可准确送达星闪执行器, 双向传输成功率 100%。

协议解析与路由: 网关正确解析自定义协议中的网络类型与设备地址字段, 实现跨网络数据路由, 未出现路由错误或数据丢失。

实验结果表明, 本系统在软硬件协同设计下, 实现了星闪与 Wi-Fi 异构网络的高效融合。网关具备良好的实时性、可靠性与扩展性, 可满足智能家居等场景中对多协议接入与边缘数据汇聚的需求。基于 Hi3863 的解决方案在成本与性能之间取得了良好平衡, 具备规模化部署的潜力。

5. 未来工作

尽管本研究成功设计并实现了一个基于 Hi3863 的多模星闪网关原型系统, 验证了星闪与 Wi-Fi 异构网络融合的基本可行性, 但无论从系统本身的功能完善还是从技术发展的前沿趋势来看, 仍有诸多值得深入研究和改进的方向。未来的工作将主要围绕以下几个方面展开:

(1) 协议栈增强与云平台深度融合

当前系统实现了边缘侧的数据透传, 未来将重点增强其与云平台的交互能力。计划移植轻量级的 JSON 解析库(如 cJSON), 对传输数据进行标准化封装, 实现 MQTT/HTTPs 等物联网标准协议与云端服务的无缝对接。这将使网关不仅能进行本地数据路由, 还能作为真正的边缘代理, 实现云端指令的下发与设备状态的远程监控, 构建“端 - 边 - 云”一体化的完整解决方案[5]。

(2) 低功耗优化策略研究

低功耗是许多物联网场景的核心诉求。未来工作将深入研究 Hi3863 芯片及星闪协议栈的低功耗特性。通过软件层面优化任务调度策略, 引入动态电源管理, 在无数据传输时使系统进入休眠模式; 在硬件层面, 可设计专门的电源管理电路。同时, 也将评估在低功耗 Wi-Fi 模式下运行 MQTT 等长连接协议的可行性与性能表现, 以期大幅提升网关及终端节点在电池供电下的续航能力。

参考文献

- [1] Vanani, K., Patoliya, J. and Patel, H. (2016) A Survey: Embedded World around MQTT Protocol for IoT Application. *International Journal for Scientific Research & Development*, 4, 26-29.
- [2] 海思官网. Hi3863V100 产品简介[EB/OL]. <https://www.hisilicon.com/cn/products/connectivity/short-range-IoT/wifi-nearlink-ble/Hi3863V100>, 2025-11-01.

-
- [3] Gao, M., Wan, L., Shen, R., *et al.* (2023) SparkLink: A Short-Range Wireless Communication Protocol with Ultra-Low Latency and Ultra-High Reliability. *The Innovation*, 4, Article 100386. <https://doi.org/10.1016/j.xinn.2023.100386>
 - [4] Dunkels, A. (2001) Design and Implementation of the LWIP TCP/IP Stack. Swedish Institute of Computer Science.
 - [5] 柴天佑, 程思宇, 李平, 贾瑶, 郑锐. 端边云协同的复杂工业过程运行控制智能系统[J]. 控制与决策, 2023, 38(8): 2051-2062.