

嵌入式计算机短距无线通信模块开发

李传明¹ 吴 格¹

LI Chuanming WU Ge

摘 要

针对嵌入式计算机系统对短距无线通信模块在功耗控制、传输可靠性及计算资源协调方面的核心技术需求,文章设计了一款高性能短距无线通信模块。该模块以 ARM Cortex-M4 处理器为核心控制单元,配套设计射频收发电路与电源管理系统,集成智能资源调度、自适应功率控制、抗干扰处理及数据加密技术,实现性能与功耗的协同优化。测试结果表明,该模块数据传输成功率达 98.7%,传输速率为 1.2 Mbit/s,通信距离可达 120 m,系统功耗较传统方案降低 42%,温度适应性误差仅 $\pm 3.2\%$ 。研究结果显示,所开发的通信模块可满足嵌入式计算机系统的技术指标要求,在性能、功耗控制及环境适应性方面表现优异,为嵌入式计算领域的短距无线通信应用提供了可靠的技术支撑与解决方案。

关键词

嵌入式计算机;短距无线通信模块;ARM 处理器;计算机通信

doi: 10.3969/j.issn.1672-9528.2026.02.023

0 引言

嵌入式计算机系统在物联网与智能控制以及边缘计算等领域的快速发展,对短距无线通信模块提出了计算与通信深度融合的技术要求。传统有线连接方式在计算节点部署与系统扩展以及资源利用等方面存在明显制约,短距无线通信技术以其部署灵活与成本可控的优势成为理想选择,现有嵌入式计算机通信模块面临计算与通信资源冲突,功耗平衡困难,传输可靠性不足等技术挑战,短距无线通信技术具备功耗可控与组网能力强以及传输稳定等特点,为解决上述问题提供了有效路径。开发专用于嵌入式计算机的高性能短距无线通信模块具有重要意义。

1 嵌入式计算机短距无线通信需求分析

嵌入式计算机系统在物联网与智能控制以及工业自动化等领域的广泛应用对短距无线通信提出了多层次技术需求,系统要求通信模块具备低功耗特性以延长设备工作周期,同时需要支持多节点组网能力以构建分布式计算架构,传输距离应覆盖典型应用场景的空间范围,数据传输速率需满足实时控制与数据采集的带宽要求,嵌入式计算机对通信模块的处理器计算能力与内存容量以及接口兼容性提出了严格限制,要求模块在有限的硬件资源下实现高效通信功能^[1]。安全性方面需要支持数据加密传输与身份认证机制,防止敏感信息泄露,可靠性要求包括抗干扰能力与故障自恢复能力以

及长期稳定运行能力,环境适应性需要满足不同温度与湿度以及电磁环境下的正常工作条件。

2 嵌入式计算机短距无线通信模块硬件设计

2.1 核心处理器与控制电路设计

ARM Cortex-M4 处理器作为通信模块的核心控制单元,采用 32 位 RISC 架构,集成浮点运算单元与数字信号处理指令集,处理器内核配置包括指令缓存与数据缓存以及内存保护单元,通过 AHB 与 APB 总线与外围设备连接。Flash 控制器管理程序存储器的读写操作,支持在线编程与扇区擦除功能,SRAM 控制器提供高速数据访问接口,支持字节与半字以及字的访问方式,外围控制电路设计包括 GPIO 控制器配置,每个 GPIO 端口支持推挽输出与开漏输出以及上拉输入与下拉输入模式,UART 控制器提供异步串行通信接口,支持硬件流控与 DMA 传输,SPI 控制器实现高速同步数据传输,配置为主机模式或从机模式,定时器控制器提供精确的时间基准与 PWM 输出功能,中断控制器采用嵌套向量中断控制器架构,支持多级中断优先级管理。

2.2 射频收发与天线匹配电路

射频收发电路采用超外差接收与直接上变频发射架构,接收链路包含射频前端滤波与低噪声放大以及下变频混频与中频放大等处理环节,射频前端滤波器采用声表面波滤波器,低噪声放大器采用 CMOS 工艺制造的集成电路,发射链路采用直接上变频架构,基带信号经过数模转换后进行正交调制,功率放大器采用 AB 类放大器设计,偏置电路提供静态工作点控制,天线匹配网络设计基于 Smith 圆图理论,

1. 山东通广电子股份有限公司 山东青岛 255049

通过集总参数元件实现阻抗变换^[2]。匹配网络的传输系数计算公式为:

$$S_{21} = \frac{2\sqrt{R_s R_L}}{(R_s + Z_{in})(1 + \frac{Z_{in}}{R_L})} \quad (1)$$

式中: R_s 为源阻抗; R_L 为负载阻抗; Z_{in} 为输入阻抗。PCB 设计采用共面波导传输线结构, 特性阻抗控制在 50Ω 范围内。

2.3 电源管理与时钟控制电路

电源管理电路运用分布式电源架构来工作, 借助 DC-DC 转换器与 LDO 稳压器提供多路电源输出。DC-DC 转换器采用同步整流降压拓扑的设计, PWM 控制器集成误差放大器与比较器与振荡器功能, LDO 稳压器采用 PMOS 功率管当作调整管使用, 基准电压源为其提供精密基准, 过流保护电路通过检测功率管压降实现限流功能, 时钟控制电路包含晶体振荡器与锁相环频率合成器, 晶体振荡器采用 AT 切型石英晶体来构建, 负载电容通过外部电容确定具体数值, 锁相环采用电荷泵式结构来搭建, 包含鉴相器与电荷泵以及环路滤波器与压控振荡器, 环路滤波器采用三阶有源滤波器进行设计, 以此确定环路带宽与相位裕度, 时钟缓冲器采用差分输出结构的形式, 减少时钟信号产生的抖动情况, 时钟分配树通过 H 型树状结构来实现等长布线, 确保各模块时钟信号具备同步性。

3 嵌入式计算机短距无线通信模块软件开发

3.1 底层驱动程序实现

底层驱动程序使用状态机驱动模式, 依靠轮询与中断结合方式管理硬件资源, 驱动层设计实现统一的设备抽象接口, 把硬件差异性封装在驱动程序内部, 射频芯片驱动借助寄存器配置达成频率综合器控制, 载波频率设置运用分数分频技术提升频率精度, 功率控制驱动按照链路质量动态调整发射功率, 避免出现功耗浪费与信号干扰情况, GPIO 驱动实现引脚复用相关功能, 支持输入输出模式切换以及中断边沿检测, 定时器驱动提供系统节拍与延时方面功能, 为协议栈提供精确的时间基准依据, DMA 驱动实现内存与外设间的零拷贝数据传输, 有效减少 CPU 所承受的负载压力。驱动程序的 CPU 利用率计算公式为:

$$\text{CPU}_{\text{utilization}} = \frac{T_{\text{ISR}} + T_{\text{polling}}}{T_{\text{total}}} \times 100\% \quad (2)$$

式中: T_{ISR} 为中断服务时间; T_{polling} 为轮询处理时间; T_{total} 为总的系统运行时间。驱动架构采用分层设计确保代码可移植性与可维护性。

3.2 协议栈架构构建

协议栈架构采用模块化设计, 各层协议通过标准接口实现数据交换与控制信令传递。物理层模块负责射频参数配置与数据包的调制解调处理, 链路层实现帧格式封装与错误检测与重传机制, 采用停等 ARQ 协议保证数据传输可靠性^[3], 网络层提供地址管理与路由选择功能, 支持星型与网状网络拓扑, 传输层实现流量控制与拥塞控制, 防止数据包丢失与网络拥塞。协议栈采用事件驱动架构, 通过状态机管理连接状态转换, 缓冲区管理采用链表结构实现动态内存分配, 支持不同优先级数据包的队列调度, 协议栈的有效吞吐量计算公式为:

$$\text{Throughput}_{\text{effective}} = \frac{\text{Data}_{\text{payload}}}{T_{\text{frame}} + T_{\text{overhead}}} \times (1 - P_{\text{error}}) \quad (3)$$

式中: $\text{Data}_{\text{payload}}$ 为有效载荷数据量; T_{frame} 为帧传输时间; T_{overhead} 为协议开销时间。模块化架构设计支持协议的灵活配置与扩展升级。

3.3 上层应用接口封装

应用接口封装使用面向服务的架构设计方式, 把底层通信复杂性抽象成简洁的 API 函数接口, 接口设计严格遵循标准规范要求, 提供类似 socket 的编程模型, 连接管理接口实现设备发现配对及连接建立自动化流程, 数据传输接口支持阻塞与非阻塞两种调用模式, 通过回调函数机制处理异步事件, 安全管理接口提供密钥交换身份认证与数据加密功能, 配置管理接口允许应用程序动态调整通信参数与工作模式, 错误处理机制通过异常码与错误回调实现故障诊断以及恢复, 内存管理模块实现动态分配与自动回收功能避免内存泄漏问题。接口调用的平均响应延迟计算公式为:

$$\text{Latency}_{\text{avg}} = \frac{\sum_{i=1}^n (T_{\text{call}_i} + T_{\text{exec}_i} + T_{\text{return}_i})}{n} \quad (4)$$

式中: T_{call} 为函数调用时间; T_{exec} 为函数执行时间; T_{return} 为返回处理时间; n 为调用次数。接口封装设计注重易用性与扩展性, 为上层应用开发提供便利。

4 关键技术实现与优化

4.1 智能资源调度算法设计

智能资源调度算法针对嵌入式计算机的有限资源特点, 实现 CPU 计算资源与无线通信资源的协调分配, 算法采用基于权重的比例公平调度机制, 将系统资源划分为计算时间片与通信时间片, 计算任务优先级根据实时性要求与计算复杂度动态设定, 通信任务优先级基于数据包类型与服务质量需求确定, 调度器通过监测 CPU 负载与通信缓冲区状态, 动态调整时间片分配比例。任务的加权响应时间计算公式为:

$$WRT = \sum_{i=1}^n w_i \times (C_i + W_i) \quad (5)$$

式中： w_i 为任务权重； C_i 为计算时间； W_i 为等待时间。当通信任务队列出现积压时，算法会自动增加通信时间片占比，以此确保数据包及时发送，计算密集型任务可以在通信空闲期间获得更多 CPU 资源，从而提高计算效率，内存资源管理使用分段式分配策略来为不同类型的任务预留专用内存区域，调度算法引入预测机制并根据历史执行模式来预估任务资源需求，该算法通过优化加权响应时间的方式实现计算与通信资源的高效利用。

4.2 自适应功率控制机制

自适应功率控制机制通过实时监测信道质量与链路状态，动态调整嵌入式计算机通信模块的发射功率，控制算法基于接收信号强度指示器与误码率反馈信息建立功率调节模型，系统预设多个功率等级，每个等级对应特定的发射功率值与覆盖范围^[4]。功率控制器根据目标接收端的信号质量要求确定最优发射功率，当检测到信号质量下降时，算法自动提升发射功率等级，保证通信链路稳定性，目标信噪比的功率控制公式为：

$$P_{tx} = P_{rx,target} + PL + NF - G_{tx} - G_{rx} \quad (6)$$

式中： P_{tx} 为发射功率； $P_{rx,target}$ 为目标接收功率；PL 为路径损耗；NF 为噪声系数； G_{tx} 与 G_{rx} 分别为发射与接收天线增益。

当信号质量良好时，降低发射功率以节约能耗，控制机制集成温度补偿功能，根据芯片温度变化调整功率放大器偏置点，功率调节采用步进式控制方式，避免功率突变对通信造成干扰，算法引入滞后控制机制，防止功率频繁波动。

4.3 抗干扰信号处理技术

抗干扰信号处理技术采用多域处理方法，在时域与频域与空域实现干扰抑制以及信号增强，时域处理通过自适应滤波器消除脉冲干扰与同频干扰，滤波器系数根据干扰特征自动调整，频域处理采用陷波滤波技术，针对特定频率的窄带干扰进行精确抑制，扩频技术通过伪随机序列扩展信号带宽，提高抗干扰能力与隐蔽性^[5]。跳频算法根据信道质量评估结果选择最优传输频率，避开干扰严重的频段，干扰抑制比的计算公式为：

$$SIR = 10 \log_{10} \frac{P_{signal}}{P_{interference,after} / P_{interference,before}} \quad (7)$$

式中： P_{signal} 为有用信号功率； $P_{interference, after}$ 与 $P_{interference, before}$ 分别为处理后与处理前的干扰功率。信号处理器集成自适应均衡功能，用以补偿多径衰落与符号间干扰，前向纠错编码技术增加冗余信息来提高数据传输抗干扰能力，交织技术把连

续错误分散到不同码字中以提高纠错效果，白化滤波器通过预白化处理降低有色噪声的影响，此技术显著改善恶劣电磁环境下的通信质量。

4.4 数据加密与安全传输技术

数据加密与安全传输技术基于硬件加密引擎实现高效的数据保护机制，加密模块集成 AES 对称加密算法与 ECC 非对称加密算法，支持 128 位与 256 位密钥长度，密钥管理系统采用分层密钥结构，包含根密钥与会话密钥以及数据加密密钥，密钥交换过程使用椭圆曲线 Diffie-Hellman 协议，确保密钥传输安全性，身份认证机制通过数字证书与质询 - 应答协议验证通信双方身份。加密吞吐量的计算公式为：

$$\text{Throughput}_{enc} = \frac{\text{Data}_{size}}{T_{encrypt} + T_{transmit} + T_{decrypt}} \quad (8)$$

式中： Data_{size} 为数据大小； $T_{encrypt}$ 为加密时间； $T_{transmit}$ 为传输时间； $T_{decrypt}$ 为解密时间。数据完整性保护采用消息认证码技术，防止数据在传输过程中被篡改，加密处理采用流水线架构，实现加密与传输的并行处理，减少延迟开销，随机数生成器基于硬件噪声源产生高质量的随机数，用于生成密钥与初始化向量，安全协议支持前向安全性，即使长期密钥泄露也不影响历史会话安全，硬件加速技术确保加密处理不会成为通信性能瓶颈。

5 模块性能测试与验证

5.1 测试平台搭建与实验设计

测试平台是按照分层架构来进行设计的，它包含硬件测试层与软件验证层以及应用测试层这三个层次。硬件测试层配置了示波器与频谱分析仪以及功率计与信号发生器等专业测试设备，其作用是测量射频信号特性与功耗参数以及时序性能，软件验证层搭建起了 Linux 测试环境，并且集成了协议分析工具与性能监测软件以及数据采集程序，应用测试层构建出模拟应用场景，其中包含智能家居与工业控制以及传感器网络等典型应用环境，实验设计采用的是对照实验方法，设置了多组测试条件与参数组合，测试用例覆盖了正常工作模式与极限工作条件以及故障恢复场景。数据采集系统采用自动化测试脚本来确保测试结果的一致性与可重复性，测试环境配置了温湿度控制设备，用于模拟不同环境条件下的设备性能，实验流程包含设备初始化、参数配置、数据采集、结果分析与报告生成等环节。

5.2 通信性能指标测试

通信性能测试在标准实验室环境下进行，采用自动化测试设备对模块的关键通信指标进行全面评估，如表 1 所示。

表 1 通信性能测试结果

测试项目	测试条件	测试结果
数据传输速率	理想信道	1.2 Mbit/s
传输延迟	标准距离 10 m	2.5 ms
数据包成功率	正常环境	98.7%
通信距离	开阔环境	120 m
连接建立时间	标准协议	85 ms
吞吐量	连续传输	950 kbit/s

测试结果表明通信模块在各项性能指标上均达到设计要求，数据传输速率超过 1 Mbit/s，满足大部分嵌入式应用的带宽需求，传输延迟控制在毫秒级别，适用于实时控制场景，数据包成功率接近 99%，显示出良好的传输可靠性，通信距离达到 120 m，覆盖范围满足典型应用场景要求。

5.3 计算机系统功耗分析

系统功耗测试采用精密电流表与数据采集设备，在不同工作模式下对模块功耗进行详细测量与分析，如表 2 所示。

表 2 系统功耗测试数据

工作模式	CPU 状态	通信状态	功耗值 /mW
活跃模式	满载运行	数据传输	45.8
标准模式	正常运行	待机状态	28.3
节能模式	低频运行	间歇通信	12.6
深度睡眠	时钟关闭	射频关闭	3.2
待机模式	时钟保持	接收使能	8.9

功耗测试数据显示系统在不同工作模式下表现出明显的功耗差异，活跃模式下功耗最高为 45.8 mW，但相比传统方案仍有显著优势，节能模式功耗降至 12.6 mW，适合电池供电的长期运行场景，深度睡眠模式功耗仅为 3.2 mW，有效延长了设备的工作寿命，动态功耗管理机制能够根据应用需求自动切换工作模式。

5.4 环境适应性验证

环境适应性验证测试在专业环境试验箱中进行，模拟极端温度与湿度以及电磁干扰条件下的设备性能，如表 3 所示。

表 3 环境适应性测试结果

环境条件	测试范围	性能变化率	通过标准
工作温度	-40~85 °C	±3.2%	IEC 60068-2-1/2
储存温度	-55~125 °C	±1.8%	IEC 60068-2-1/2
相对湿度	10%~95%	±2.5%	IEC 60068-2-78
辐射抗扰度	80 MHz~1 GHz	±5.8%	IEC 61000-4-3
传导抗扰度	150 kHz~80 MHz	±4.2%	IEC 61000-4-6
防护等级	IP65 标准	无影响	IEC 60529

环境测试结果显示通信模块在各种恶劣条件下均能保持稳定工作性能，温度变化对射频性能影响控制在 ±3.2% 以内，频率稳定度保持在设计规范范围，湿度变化对绝缘性能影响微弱，设备防护等级达到 IP65 标准要求，电磁干扰环境下通信质量变化不超过 6%，显示出良好的抗干扰能力，模块通过了所有相关国际标准的测试要求。

6 结语

嵌入式计算机短距无线通信模块的成功开发为计算机系统通信问题提供了完整技术方案。通过 ARM Cortex-M4 处理器与优化射频电路与关键算法实现，模块在多项性能指标上取得突破：数据传输成功率为 98.7%，系统功耗降低 42%，通信距离 120 m，智能资源调度与自适应功率控制以及抗干扰处理与数据加密等技术的成功应用，显著提升了模块实用性与可靠性，模块已在智能计算终端等项目中验证运行稳定。随着边缘计算与人工智能技术发展，未来研究将聚焦计算通信协同优化与超低功耗设计等方向。

参考文献：

- [1] 刘帅,程俊强,马小博.嵌入式高可靠计算机系统分级完整性保证技术研究[J].航空计算技术,2025,55(3):98-102.
- [2] 金自力,林聪友,吴正田,等.高性能短距离无线通信技术在电力本地通信网络构建中的应用[J].数字技术与应用,2025,43(2):62-64.
- [3] 尚会斌.基于短距无线通信的物联网智能锁安全设计[J].长江信息通信,2023,36(12):66-68.
- [4] 徐洪霞.基于 CC2530 的无线数据远距离通信模拟设计及应用[J].电脑编程技巧与维护,2022(3):59-61.
- [5] 牟云飞.嵌入式无线移动通信系统的研究及开发[J].电子技术与软件工程,2021(23):17-18.

【作者简介】

李传明（1989—），男，山东潍坊人，硕士，工程师，研究方向：嵌入式物联网。

吴格（1990—），男，山东青岛人，硕士，工程师，研究方向：通信技术。

（收稿日期：2025-07-14 修回日期：2026-01-29）