

多节点电子控制器集群程序同步更新方法及装置研发

方正¹ 江巍² 余东海¹ 张季² 张永光¹

1.杭州迪为科技有限公司；2.浙江零跑科技股份有限公司

DOI: 10.32629/jpm.v7i2.8733

[摘要] 针对多节点电子控制器集群在程序更新过程中存在的同步性差、更新效率低、可靠性不足等问题，本文提出一种基于分层通信架构与增量同步机制的程序更新方法，并设计对应的硬件装置。该方法通过主控制器与从控制器的分级协同，结合基于CRC32校验的增量数据包传输策略，实现集群节点的高效同步更新。搭建由16个控制器节点组成的试验平台，通过两组对比试验验证方法有效性：结果表明，该方法更新延迟 $\leq 28\text{ms}$ ，同步误差控制在3ms内，数据包传输成功率达99.87%，相较于传统全量更新方法，更新效率提升42.3%，且在网络丢包率 $\leq 5\%$ 的场景下仍保持稳定更新。该研发成果为工业控制、智能终端集群等领域的程序维护提供了可靠技术支持。

[关键词] 电子控制器集群；程序同步更新；分层通信；增量同步

Research and Development of Multi-node Electronic Controller Cluster Program Synchronization Update Method and Device

Fangzheng¹ Jiang Wei² Yu Donghai¹ Zhang Ji² Zhang Yongguang¹

1. Hangzhou Diwei Technology Co., Ltd.; 2. Zhejiang Leapmotor Technology Co., Ltd.

[Abstract] To address issues such as poor synchronization, low update efficiency, and insufficient reliability in program updates for multi-node electronic controller clusters, this paper proposes a program update method based on hierarchical communication architecture and incremental synchronization mechanism, along with corresponding hardware device design. The method achieves efficient synchronization updates for cluster nodes through hierarchical collaboration between master and slave controllers, combined with incremental data packet transmission strategy based on CRC32 checksum. An experimental platform composed of 16 controller nodes was established, and the method's effectiveness was verified through two comparative experiments. Results show that the update delay is $\leq 28\text{ms}$, synchronization error is controlled within 3ms, and data packet transmission success rate reaches 99.87%. Compared to traditional full-update methods, the update efficiency improves by 42.3%, and stable updates are maintained even under network packet loss rate $\leq 5\%$ scenarios. This research provides reliable technical support for program maintenance in industrial control and intelligent terminal cluster fields.

[Key words] Electronic controller cluster; Program synchronization update; Hierarchical communication; Incremental synchronization;

在工业自动化、智能物联网等技术快速演进的背景下，多节点电子控制器集群已成为复杂系统的核心组成部分，其程序更新的同步性与可靠性直接影响系统整体运行效能。传统更新方法多采用单节点逐一更新或简单广播传输模式，存在三大突出问题：一是同步性不足，各节点程序版本差异易导致集群协同故障，尤其在实时控制场景中，节点间更新延迟可能引发设备误动作；二是传输效率低下，全量程序包传输占用大量网络带宽，在多节点场景下更新耗时过长，难以适应大规模集群的

维护需求；三是抗干扰能力弱，工业环境中的电磁干扰、网络丢包等因素易造成更新中断，且缺乏有效的容错与恢复机制。现有研究虽尝试通过时间同步协议或冗余传输优化，但未充分解决增量数据提取与集群分级协同的核心矛盾，导致实际应用中仍存在更新成功率低、维护成本高的问题。基于此，本文聚焦多节点控制器集群的同步更新需求，研发兼具高效性与可靠性的更新方法及专用装置，通过试验验证其在不同工况下的适配能力，为集群系统的柔性维护提供技术解决方案。

1 多节点电子控制器集群程序同步更新原理

1.1 集群节点通信架构设计

多节点电子控制器集群的同步更新依赖稳定的通信链路与协同机制，本文设计分层式通信架构，将集群节点划分为主控制器层与从控制器层。主控制器作为更新发起与协调核心，承担程序包解析、增量数据提取、同步指令下发及状态监控功能；从控制器层包含若干个子节点，负责接收更新指令、执行程序更新及反馈更新状态。通信链路采用“主-从双向交互+节点间辅助校验”模式，主控制器与从控制器通过 CAN 总线实现数据传输，波特率设置为 500kbps，同时预留以太网接口用于大规模集群扩展^[1]。为降低通信冲突，采用时分多址 (TDMA) 调度策略，为主控制器分配固定通信时隙 (占空比 15%)，从控制器按地址顺序分时反馈状态信息，每个节点的通信时隙 ≤ 1ms，确保集群通信的有序性。

1.2 增量同步更新机制

传统全量更新方法需传输完整程序包 (通常为 512KB~2MB)，导致传输耗时过长且带宽占用率高^[2]。本文提出的增量同步机制基于程序版本差异提取有效更新数据，具体流程如下：首先，主控制器获取各从控制器当前程序版本号与 CRC32 校验值，与目标程序进行对比，生成版本差异矩阵；其次，基于差异矩阵提取增量数据块，每个数据块大小设置为 1KB，通过 LZ77 算法进行压缩处理，降低数据传输量；最后，为每个增量数据块添加校验字段 (CRC32 值) 与地址标识，按从控制器地址顺序分批下发。该机制的核心优势在于仅传输程序变更部分，而非完整程序包，且通过校验字段确保数据完整性，避免因传输错误导致的更新失败。

2 程序同步更新装置研发

2.1 硬件装置总体设计

同步更新装置由主控制器模块、从控制器接口模块、通信适配模块、电源管理模块及状态指示模块组成，硬件架构如图 1 所示 (此处省略图，文中以参数描述替代)。主控制器采用 STM32H743VIT6 芯片，主频 480MHz，内置 2MBFlash 与 1MBRAM，支持 CAN、以太网等多种通信接口，满足多协议数据处理需求；从控制器接口模块设计 16 路独立通信接口，兼容 CAN2.0B 与 RS485 协议，可适配不同类型的电子控制器；通信适配模块集成信号隔离芯片 ADUM1400，实现通信信号的电磁隔离，增强装置抗干扰能力；电源管理模块采用宽电压输入设计 (9~36VDC)，通过 LM2596-5.0 与 AMS1117-3.3 芯片输出 5V 与 3.3V 稳定电压，为各模块供电；状态指示模块通过 LED 灯实时显示主控制器的通信状态、更新进度及故障信息，便于现场维护^[3]。

2.2 硬件参数指标

装置核心硬件参数经过多轮调试优化，关键指标如下：主控制器数据处理速率 ≥ 1.2Mbps，支持同时连接 ≤ 32 个从控制器节点；通信接口传输延迟 ≤ 5ms，信号隔离电压 ≥ 2.5kV，满

足工业电磁兼容标准；电源模块输出纹波 ≤ 50mV，负载调整率 ≤ ±2%；装置工作温度范围为 -40℃~85℃，防护等级 IP40，适配恶劣工业环境；更新过程中，装置功耗 ≤ 12W，相较于传统更新设备降低 35%，符合节能要求。

2.3 软件功能模块设计

软件系统基于 FreeRTOS 实时操作系统开发，采用模块化设计思路，主要包含版本管理模块、增量数据处理模块、通信协议栈、同步控制模块及容错处理模块。版本管理模块负责程序版本号维护与差异对比，支持版本回滚功能；增量数据处理模块实现 LZ77 压缩算法与 CRC32 校验逻辑，通过硬件加速单元提升数据处理效率；通信协议栈兼容 CANopen 与 TCP/IP 协议，针对集群更新场景优化数据帧结构，减少协议开销；同步控制模块通过定时器中断与状态机机制，实现更新流程的精准调度；容错处理模块设计三重保障机制：数据重传机制 (最多 3 次重传)、断点续传功能 (记录中断时的数据包位置)、应急恢复模式 (更新失败时自动回滚至原程序版本)，确保更新过程的可靠性^[4]。

3 试验设计与结果分析

3.1 试验平台搭建

本研究搭建由 16 个电子控制器节点 (STM32F407VET6)、研发的同步更新装置、网络干扰模拟器及数据采集仪组成的试验平台。其中，16 个控制器节点按 4×4 矩阵布局，与主控制器通过 CAN 总线连接，总线长度 12m，终端匹配电阻 120Ω；网络干扰模拟器用于模拟不同丢包率的通信环境 (丢包率 0%~10%可调)；数据采集仪 (采样频率 1kHz) 记录各节点的更新开始时间、完成时间及状态信息，同步误差通过节点间时钟校准后的时间差计算。试验分为两组：试验 1 为更新效率与同步性测试，对比本文方法与传统全量更新方法的性能；试验 2 为可靠性测试，验证不同丢包率场景下的更新成功率与稳定性。

3.2 更新效率与同步性测试

测试设定程序更新目标规模为 1.2MB，依托版本差异计算确定增量更新数据量为 428KB；集群节点数量分 8 个、12 个、16 个三组开展对照测试，通信环境配置为丢包率 0% 的无丢包状态，环境温度控制在 25℃ 且无电磁干扰，同时以基于 CAN 总线广播传输完整程序包的传统全量更新方法作为对比对象，开展相关测试分析。试验结果如表 1 所示。

在不同节点数量下，本文提出的增量同步更新方法均表现出更优的更新效率与同步性。当节点数量为 16 时，本文方法的总更新时间为 18.7s，相较于传统全量更新的 32.4s，效率提升 42.3%；同步误差最大为 2.8ms，远低于传统方法的 12.5ms。这一结果得益于增量数据传输减少了带宽占用，同时分层通信架构与时序协同策略保障了各节点的同步执行。随着节点数量增加，传统方法的更新时间呈线性增长，而本文方法

因采用分批传输与并行处理，时间增长幅度较小，体现出良好的大规模集群适配能力^[5]。

表1 不同节点数量下更新性能对比

| 节点数量 | 更新方法 | 总更新时间 (s) | 平均单节点更新时间 (s) | 同步误差最大值 (ms) | 数据包传输量 (MB) |
|------|--------|-----------|---------------|--------------|-------------|
| 8 | 本文方法 | 9.3 | 1.16 | 2.1 | 0.428 |
| 8 | 传统全量更新 | 15.7 | 1.96 | 9.8 | 9.6 |
| 12 | 本文方法 | 13.5 | 1.12 | 2.5 | 0.428 |
| 12 | 传统全量更新 | 23.4 | 1.95 | 11.3 | 14.4 |
| 16 | 本文方法 | 18.7 | 1.17 | 2.8 | 0.428 |
| 16 | 传统全量更新 | 32.4 | 2.03 | 12.5 | 19.2 |

3.3 可靠性测试

设定集群节点数量为 16 个，选取 428KB 的增量数据包作为程序更新数据载体；网络丢包率分别设置为 0%、2%、3%、5%、8%、10%并分六组开展测试，每组测试重复执行 50 次，同步记录各组的更新成功次数与平均更新时间。试验结果如表 2 所示。

由表可知，本文方法在丢包率≤5%的场景下，更新成功率保持 100%，即使丢包率达到 8%，成功率仍达 96%，而传统方法

在丢包率 3%时成功率已降至 88%。当丢包率为 5%时，本文方法的平均更新时间为 21.3s，仅比无丢包场景增加 13.9%，这得益于容错处理模块的重传与续传机制，有效弥补了数据丢失的影响；而传统方法在丢包率 5%时，平均更新时间增至 48.7s，且因缺乏有效的增量校验，部分节点出现程序损坏导致更新失败。当丢包率超过 8%时，本文方法的成功率开始显著下降。

表2 不同丢包率下更新可靠性对比

| 丢包率 (%) | 更新方法 | 更新成功次数 (/50 次) | 成功率 (%) | 平均更新时间 (s) | 程序损坏节点数 (/16 个) |
|---------|--------|----------------|---------|------------|-----------------|
| 0 | 本文方法 | 50 | 100 | 18.7 | 0 |
| 0 | 传统全量更新 | 50 | 100 | 32.4 | 0 |
| 2 | 本文方法 | 50 | 100 | 19.5 | 0 |
| 2 | 传统全量更新 | 47 | 94 | 35.8 | 1 |
| 3 | 本文方法 | 50 | 100 | 20.1 | 0 |
| 3 | 传统全量更新 | 44 | 88 | 39.2 | 2 |
| 5 | 本文方法 | 50 | 100 | 21.3 | 0 |
| 5 | 传统全量更新 | 36 | 72 | 48.7 | 3 |
| 8 | 本文方法 | 48 | 96 | 25.7 | 1 |
| 8 | 传统全量更新 | 23 | 46 | 62.3 | 5 |
| 10 | 本文方法 | 41 | 82 | 29.4 | 2 |
| 10 | 传统全量更新 | 15 | 30 | 75.6 | 7 |

4 结论

本文针对多节点电子控制器集群程序同步更新的核心需求，研发了基于分层通信架构与增量同步机制的更新方法及专用装置。通过理论设计与试验验证，得出以下结论：该方法采用主从分级协同策略，结合增量数据包传输与 CRC32 校验，有效提升了集群更新的效率与同步性，更新延迟≤28ms，同步误差≤3ms，相较于传统方法效率提升 42.3%；所设计的硬件装置集成通信适配、电源管理与容错处理功能，在网络丢包率≤5%的场景下更新成功率达 100%，具备良好的工业环境适应性。

[参考文献]

[1]张晚霞.基于集群技术的小农户智能温棚控制器的研究与开发[J].信息系统工程, 2025, (07): 36-39.

[2]袁登科, 徐延东, 李秀涛.永磁同步电动机变频调速系统及其控制[M].机械工业出版社: 2022: 12: 425.

[3]宋振龙.面向超级计算的大规模并行存储系统关键技术研究[D].国防科学技术大学, 2014.

[4]陈建.基于 FPGA 的 TTP/C 总线控制器设计及验证[D].南京航空航天大学, 2012.

[5]刘勇.基于 DSP 的燃机三余度数字电子控制器设计[D].南京航空航天大学, 2007.

作者简介：方正，出生年月：1986.04.24，男，汉族，籍贯：浙江省杭州市，学历：研究生，职称：中级工程师，研究方向：汽车电子软件开发及仿真测试技术。