

基于GNSS驱动的嵌入式时间同步系统实现

郑枫, 张开星, 汪华章*

(西南民族大学电气工程学院, 四川 成都 610041)

摘要:针对分布式嵌入式系统中时间基准不一致引起的数据融合偏差和控制失稳问题,设计并实现了一种基于全球导航卫星系统(Global Navigation Satellite System, GNSS)驱动的嵌入式时间同步系统。该系统通过集成卫星授时功能的网络时间协议(Network Time Protocol, NTP)服务器提供时间源,利用串口转以太网模块卸载硬件协议栈,并在模块内解析NTP报文并提取时间戳,将精确时间数据传输至STM32控制器。STM32通过本地定时器维持时钟并定期同步以修正累积误差。与传统基于软件协议栈的实现方式相比,通过报文透明映射与数据重构机制,将微控制器(Microcontroller Unit, MCU)处理数据量降低约80%,优化了数据处理流程。实验结果表明:在无需运行TCP/IP协议栈的条件下,系统仅通过串口通信即可实现时间同步,误差小于20 ms,运行稳定可靠。该系统适用于资源受限的嵌入式设备,为高精度时间同步提供了一种低复杂度、低成本的工程实现方案。

关键词:GNSS授时;嵌入式系统;时间同步;NTP协议

中图分类号:TB939;P228.4

文献标志码:A

文章编号:2095-4271(2026)05-0569-10

Implementation of an embedded time synchronization system driven by GNSS

Zheng Feng, Zhang Kaixing, Wang Huazhang*

(College of Electrical Engineering, Southwest Minzu University, Chengdu 610041, China)

Abstract: To address data fusion deviations and control instability caused by inconsistent time references in distributed embedded systems, this paper designed and implemented a GNSS-driven embedded time synchronization system. The system provided a time source through an NTP server integrated with satellite timing functionality. A serial-to-Ethernet module was used to offload the hardware protocol stack, and NTP packets were parsed within the module to extract timestamps, which were then transmitted as precise time data to the STM32 controller. The STM32 maintained the system clock using a local timer and performed periodic synchronization to correct accumulated errors. Compared with traditional software protocol stack-based implementations, a transparent packet mapping and data reconstruction mechanism was introduced, reducing the MCU processing load by approximately 80% and optimizing the data processing workflow. Experimental results demonstrated that, without running a TCP/IP protocol stack, the system achieved time synchronization solely via serial communication, with an error of less than 20 ms, ensuring stable and reliable operation. The proposed system was suitable for resource-constrained embedded devices and provided a low-complexity, low-cost engineering solution for high-precision time synchronization.

Keywords: GNSS timing; embedded system; time synchronization; NTP protocol

在工业控制、智能电网^[1]、物联网^[2]等领域,各个节点间时间基准的统一是确保系统行为可预测和数据融

收稿日期:2026-03-23

*通信作者, E-mail: wanghuazhang@swun.edu.cn

基金项目:西南民族大学2025年研究生创新型科研项目

引用格式:郑枫, 张开星, 汪华章. 基于GNSS驱动的嵌入式时间同步系统实现[J]. 西南民族大学学报(自然科学版), 2026, 52(5): 569-578.

Zheng Feng, Zhang Kaixing, Wang Huazhang. Implementation of an embedded time synchronization system driven by GNSS[J]. Journal of Southwest Minzu University (Natural Science Edition), 2026, 52(5): 569-578.

合可靠的基础条件. 随着系统规模扩大和节点数量增加, 分布在不同位置的采样数据、控制命令和事件记录必须基于统一的时钟参考进行时间对齐. 一旦缺少有效的同步机制, 系统便可能出现时序混乱乃至逻辑矛盾. 在多机协同控制与状态估计这类对精度要求严苛的场合, 即使是微秒量级的同步误差, 经过系统级联传递后也可能显著降低整体运行精度. 因此, 建立稳定、可溯源且高度一致的时间体系^[3], 已经成为分布式嵌入式系统设计中必须解决的关键问题.

在实际的工程应用中, 现有方案大多基于 NTP 协议实现时钟同步. 主要是由上游高精度时钟源获取协调世界时 (Coordinated Universal Time, UTC) 后向终端设备分发. 但是对于资源有限的嵌入式设备来说, 本地运行完整的 TCP/IP 协议栈和 NTP 客户端会带来明显的系统负担. 主要是因为全功能协议栈需要占用较多的 RAM 和 Flash 空间, 而且复杂的报文解析过程也会消耗 CPU 的算力. 在网络协议处理过程中存在的不确定性容易引发时间抖动^[4], 这就会使得本地时钟校准出现随机延迟, 最终影响同步精度. 另外, 当嵌入式节点同时承担高频采样或控制任务时, 频繁的网络中断可能干扰原有的实时任务调度, 造成资源竞争, 降低系统运行的确定性^[5].

针对上述问题, 本文提出一种分层同步架构, 由专业 NTP 服务器与硬件协议转换网关协同实现. 该架构以具备卫星授时功能的 NTP 服务器作为所有设备的时间基准, 在网关侧集中处理复杂的协议交互和时间戳计算, 通过通信模块完成协议解析与时间信息提取^[6], 最终将时间数据通过串口发送至 STM32 端. 与现有侧重协议优化或高精度算法设计的研究不同, 本系统主要面向资源受限嵌入式场景, 重点关注时间同步系统的工程实现方法与系统结构优化. 该方案无需在 MCU 侧部署完整协议栈, 终端只需通过串口读取经过预处理的时间报文即可. 从整体架构来看, 该设计不仅有效减轻了 CPU 负担, 同时抑制了因中断处理带来的不确定性. 本文将从该方案的系统架构、理论模型和实验验证方面进行阐述.

1 GNSS 授时与守时机理分析

GNSS 授时与源头守时机制是整个时间同步体系的物理基础, 也直接决定了系统能够达到的时间精度

上限. 本文选用专业 NTP 时间服务器 TF-NTP-PRO 型作为源头时间基准设备. 该服务器内部集成了多模 GNSS 接收单元, 以卫星时间为绝对参考, 通过卫星时间解算配合本地高稳定振荡器的驯服控制^[7], 完成从“绝对时间获取”到“连续稳定输出”的闭环时间控制过程. 其核心机理可分为两个层面: 一是 GNSS 信号接受与时间解算过程; 二是本地温补晶振 (TCXO) 的频率驯服与守时机制.

在 GNSS 授时阶段, 服务器通过卫星天线接收来自北斗 (BDS) 与 GPS 卫星的 L 波段射频信号. 卫星导航电文中包含了 UTC 时间偏移参数、卫星钟差修正参数以及轨道星历数据等关键信息. 接收机对射频信号完成前端放大、下变频与模数转换后, 通过伪随机码捕获与相关运算实现卫星信号锁定, 随后进行码跟踪和载波跟踪, 从而获得稳定的伪距观测量^[8]. 与常规的定位模式不同, 授时应用中的天线位置为已知固定值, 因此空间坐标变量可视为常量, 原本多维的问题简化为仅关于时间偏差的求解问题, 其关系表达式为:

$$t_{\text{local}} = t_{\text{GNSS}} + \Delta t_{\text{clock}} + \varepsilon. \quad (1)$$

其中 t_{GNSS} 为经 UTC 转换的卫星系统时间, Δt_{clock} 为接收机本地时钟相对于卫星的偏差, ε 表示信号传播延迟以及测量噪声等误差项. 通过最小二乘或卡尔曼滤波等方法可实时求解本地钟差, 实现对 UTC 时间的精确校准. 当可见卫星数量充足且信号质量好时, GNSS 授时精度可达到纳秒至百纳秒量级, 从而为后续网络时间分发提供稳定可靠的时间基准.

但在实际情况中, 卫星信号可能受到建筑物遮挡、电磁干扰或天气变化等因素影响而出现短时失锁, 因此仅依赖实时卫星解算难以保证时间输出的连续性^[9]. 为此, 服务器内部配置高稳定度的温补晶振作为本地频率参考. 当卫星信号正常时, 系统将 GNSS 解算得到的 UTC 时间与本地振荡器计数值进行比较, 计算相位误差与频率误差, 并通过闭环控制算法调节振荡器控制电压, 对其输出频率进行细微校正, 使本地振荡器频率逐步逼近 GNSS 基准:

$$f_{\text{TCXO}} \rightarrow f_{\text{GNSS}}, \varphi(t) \rightarrow 0. \quad (2)$$

式 (2) 中, $\varphi(t)$ 为本地振荡器相对于 GNSS 基准的相位误差. 这一过程本质上可视为具有较长时间常数的相位锁定过程, 其作用在于消除振荡器的长期频率偏差. 当卫星信号出现短时中断时, 系统自动进入

Holdover模式,此时停止来自GNSS的外部校正,仅依靠已完成驯服的TCXO维持时间输出.由于锁定阶段已经对频率误差进行了补偿,守时阶段的时间漂移主要由晶振短期稳定度与老化误差决定,其漂移速率明显低于未驯服状态,因此即使在数分钟甚至数小时内仍能维持较小的误差.

2 NTP时间同步机制与误差模型

网络时间协议(NTP)是计算机广泛采用的时间同步协议,其核心作用是确保网络中各设备的系统时钟保持一致,为时间戳记录、数据处理以及各类依赖时间顺序的计算任务提供统一的时间参考^[10].通过网络时间同步机制,不同节点能够在统一时间尺度下进行信息交换和任务调度,这也是构建分布式控制系统与测量网络的重要基础.

NTP的时间同步过程本质上是客户端与服务器的双向报文交互过程,在这个过程中记录并利用四个关键时间戳来估计客户端与服务器之间的时钟偏差.如图1所示,客户端会向NTP服务器发送请求报文.当报文从客户端发出时,客户端会记录当前本地时钟值 T_1 ,该时间被称为始发时间戳,并写入请求报文中.而请求报文经过网络传输后到达服务器端,服务器在接收瞬间读取本地时钟值并记录为 T_2 ,这个时间称为接收时间戳.随后服务器对请求报文进行处理,并在发送应答报文时记录本地时间为 T_3 ,该时间为服务器发送时间戳.应答报文返回客户端后,客户端在接收瞬间记录当前本地时钟读数 T_4 ,即目的时间戳.当整个时间交换过程结束后,四个关键时间戳 T_1, T_2, T_3 和 T_4 均已获得.

在理想条件下,若假设网络路径在两个方向上的

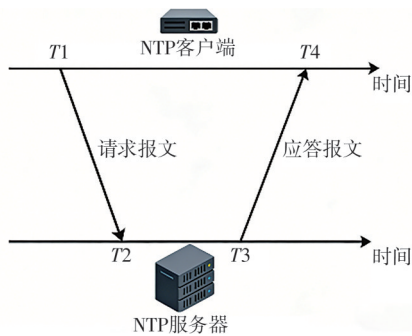


图1 NTP时间同步原理示意图

Fig.1 Schematic diagram of NTP time synchronization principle

传播时延近似相同,则可利用上述时间戳计算网络往返时延以及客户端相对于服务器的时钟偏差.其中,往返时延 δ 可表示为:

$$\delta = (T_4 - T_1) - (T_3 - T_2). \quad (3)$$

客户端相对于服务器的时钟偏差 θ 可表示为^[12]:

$$\theta = \frac{(T_2 - T_1) + (T_3 - T_4)}{2}. \quad (4)$$

客户端根据计算得到的时钟偏差对本地系统时钟进行校正,使其逐渐逼近服务器时间,从而实现网络节点之间的时间同步.

但在实际网络环境中,上述计算模型往往受到多种因素影响.其中,网络链路的非对称传播特性是影响同步精度的重要因素之一^[13].若客户端到服务器的传播时延为 d_1 ,服务器到客户端的传播时延为 d_2 ,当两者不相等时,偏差估计结果将受到影响.由于实际网络中报文可能经过不同的路由路径或交换设备排队过程,导致两个方向上的传播时间存在差异,从而在同步结果中引入额外误差^[14].此时的偏移估计误差 ε 为:

$$\varepsilon = \frac{d_1 - d_2}{2}. \quad (5)$$

此外,时间戳获取精度同样会对同步效果产生影响.NTP算法依赖四个时间戳 T_1, T_2, T_3 和 T_4 的准确记录,如果时间戳是在操作系统或应用层完成标记,则其记录时刻可能受到中断响应延迟、协议栈处理时间以及任务调度等因素影响,使时间记录与报文真实发送或接收时刻之间产生偏差.当这些不确定延迟与网络传播时延处于同一量级时,将明显影响往返时延和时钟偏差的计算结果.故可将打标抖动表示为随机变量 η ,则实际偏移估计模型应修正为^[15]:

$$\hat{\theta} = \theta + \frac{d_1 - d_2}{2} + \eta. \quad (6)$$

因此,减少软件协议处理路径并提高时间获取的确定性,是提升嵌入式系统时间同步精度的重要途径.

3 硬件级协议栈卸载机制设计

在传统的嵌入式网络时间同步方案中,MCU通常需运行完整的TCP/IP协议栈及NTP客户端程序,实现UDP报文构造、时间戳解析与偏移计算等.这种方案不仅占用较大存储与算力资源,还容易产生协议处理过程的中断抖动与调度不确定性,导致系统的实时性与确定性降低^[16].针对上述问题,本文提出一种硬件协议

栈卸载(Hardware Protocol Stack Offloading)机制,将复杂网络协议处理从主控 MCU 转移至专用通信模块,实现网络处理与时间控制逻辑的结构解耦. 本节从体系结构与报文透明映射机制两方面进行理论阐述.

3.1 系统架构设计

本系统采用的通信模块为串口转以太网模块,型号为 ZLAN3003S,其内部核心芯片为 ZLAN1003,负责 TCP/IP 协议栈处理及串口与以太网数据转换. 该模块具备以下功能:

- ①以太网 MAC/PHY 接口;
- ②IP、UDP 协议解析与封装;
- ③Socket 管理;
- ④数据缓冲与转发控制.

传统软件架构与硬件卸载架构的处理流程对比如图 2 所示. 在传统实现方式中,网络数据到达后触发网络中断,随后依次在 MCU 内部完成 MAC 层帧解析、IP 层头部校验与合法性检查、UDP 层端口匹配与数据解封装、NTP 报文字段解析、时间戳提取与同步计算. 在该架构下,MCU 既要承担控制算法的任务,又要承担网络层与传输层之间的协议处理任务. 协议栈执行过程中涉及头部校验、报文完整性验证、端口匹配、Socket 状态维护以及可能的报文重组等操作,这些都会增加处理器的计算资源并引入不可预测的执行延迟. 而在硬件卸载架构中,网络协议处理功能被转移至独立通信模块内部. 网络中断仍然存在,但其后续处理不再由 MCU 完成,而是在通信模块内部完成协议栈解析. 随后,模块仅将已解析完成的有效时间数据通过串口(UART)传输给 MCU. MCU 只需执行串口数据读取和时间计算,不再参与 IP 头校验、UDP 解封装、报文重组或 Socket 状态维护等网络层与传输层功能.

从系统理论角度分析,该结构等效于将网络层与传输层功能从 MCU 控制闭环中移除,使系统的输入形式由具有随机性和突发性的网络中断事件转变为相对确定的串口数据流. 假设 MCU 总计算资源为 R_{cpu} , 软件协议栈占用资源为 R_{net} , 则传统架构下可用于控制与时间计算的有效资源为 $R_{available} = R_{cpu} - R_{net}$, 而在硬件卸载架构中可近似认为 $R_{available} \approx R_{cpu}$. 因此系统实时裕量显著提高,同时中断源由高频网络中断转变为低频串口接收中断,使时间采样抖动明显减小,从而为高精度时间同步提供了更加稳定的运行环境.

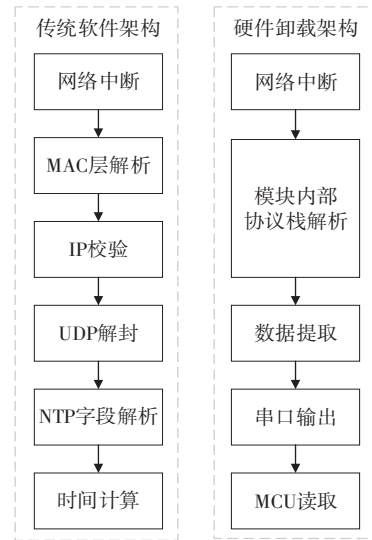


图2 传统软件与硬件卸载架构处理流程对比

Fig.2 Comparison of processing flows between traditional software and hardware offload architectures

3.2 报文透明映射与数据重构方法

在本系统中,ZLAN3003S 模块不仅承担 TCP/IP 协议栈处理任务,还在应用层实现报文透明映射机制,以降低 STM32 端的解析复杂度并增强系统时间获取的确定性.

NTP 标准报文长度为 48 字节,其结构如表 1 所示. 在表中时间戳字段为 64 位格式(32 位秒+32 位小数秒). 在客户端时间校准过程中,最关键的数据为服务器发送时刻的 T_3 . 若 MCU 直接解析完整 NTP 报文,则需要完成 48 字节数据缓存、字段偏移定位、大端字节序转换、时间单位换算等步骤,对于资源受限的 MCU,这一流程不仅增加代码复杂度,还引入协议解析延迟抖动.

表 1 NTP 标准报文

Table 1 NTP standard message

字节偏移	字段	长度/bit	功能说明
0	LI, VN, Mode	8	闰秒指示、版本号、工作模式
1	Stratum	8	时间源层级
2	Poll	8	轮询间隔(2 n 秒)
3	Precision	8	本地时钟精度(2 n 秒)
4-7	Root Delay	32	到一级时间源的总往返延迟
8-11	Root Dispersion	32	累计最大误差
12-15	Reference ID	32	上级时间源标识
16-23	Reference Timestamp	64	最近一次同步时间
24-31	Originate Timestamp	64	客户端发送时间 T_1
32-39	Receive Timestamp	64	服务器接收时间 T_2
40-47	Transmit Timestamp	64	服务器发送时间 T_3

因此,为进一步降低MCU解析复杂度,模块内部实现如下数据重构流程:

- ①模块接收完整48 Byte NTP UDP 报文;
- ②依据固定偏移直接访问 T_3 字段;
- ③读取64位时间戳数据;
- ④完成字节重组与时间基准转换;
- ⑤按预定义结构生成精简时间帧;
- ⑥通过串口以16进制数据流输出至STM32.

经模块完成 T_3 字段提取与时间基准转换后,原始48 Byte NTP 报文被重构为轻量化时间数据帧,其结构如表2所示. 其中秒计数采用48 bit 保证长期运行能力,毫秒计数提供子秒分辨率,帧头实现帧同步,校验位用于误码检测以提升系统可靠性.

表2 映射后的时间帧内容

Table 2 Time frame content after mapping

字节偏移	字段	长度/bit	功能说明
0	帧头	8	帧起始标识
1-6	秒计数	48	Unix 时间秒计数
7-8	毫秒计数	16	子秒时间(0~999 ms)
9	校验位	8	数据完整性校验

映射后的时间帧结构如图3所示. 因为原始数据长度 $L_{NTP} = 48$ Byte, 映射后长度 $L_{map} = 10$ Byte, 则MCU实际处理数据量比例可计算出:

$$\frac{L_{map}}{L_{NTP}} = \frac{10}{48} \approx 20.8\%.$$

由上述计算可知MCU仅需处理原始数据量的五分之一,数据处理规模降低约80%,同时完全消除协议字段解析逻辑.



图3 映射后的时间帧结构

Fig.3 Time frame structure after mapping

通过硬件卸载机制,网络中的随机因素被限制在ZLAN3003S模块内部,MCU仅处理稳定且格式化后的时间数据,而串口传输延迟的确定性保证了时间数据传输的高精度和低抖动. 因此,整个系统的实时性、准确性和稳定性得到显著增强,尤其是在面对不确定网络环境时,表现出较高的抗干扰能力和可靠性. 该机制实现了从完整NTP协议解析向轻量化时

间数据的转换,是本文系统降低复杂度与提高时间精度的关键技术.

4 系统硬件结构设计

本系统采用基于硬件协议栈的NTP网络时钟方案,系统核心由以太网接口、网络协议转换模块、主控制微处理器及显示驱动模块组成.

系统物理层链路采用集成变压器的RJ45连接器HR911105A作为核心接口. 如图4所示,该器件内部集成的隔离变压器抗浪涌电压在1.5 kV以上,实现了系统逻辑地与外部网络环境的电气隔离,通过共模扼流圈抑制了传导干扰. 在差分信号传输路径上,NET_TX±与NET_RX±链路严格遵循阻抗匹配原则,通过串接49.9 Ω (1%精度)的精密端接电阻,将特征阻抗稳定在100 Ω 附近,最大限度地消除了高频信号在传输线上的反射损耗,确保以太网帧在物理层的高可靠透传.

网络协议处理层由硬件协议栈芯片ZLAN1003及其外围配置电路组成,如图5所示. 该模块承担了底层MAC/PHY信号与应用层异步串口数据之间的协议封装与拆解任务. 为保障芯片在高速处理过程中的热稳定性和时序鲁棒性,系统构建了多级电压调节架构,通过LDO稳压器AMS1117-3.3与AMS1117-1.8分别为I/O环路与逻辑内核提供静态偏置,并采用 π 型滤波电容阵列抑制电源轨道的纹波噪声. 硬件配置方面,利用电阻排RN1实施硬接线编程,预设了SPEED与PULL_UP等关键配置位的电平状态,结合25 MHz外部有源晶振提供的低抖动时钟参考,确保了NTP报文在硬件层面的透明映射与快速响应.

系统主控单元采用基于ARM Cortex-M3内核的STM32F103C8T6微处理器,负责全局资源调度NTP报文的应用层解析,如图6所示. 在时钟体系设计中,系统配置了8 MHz主时钟源以驱动内部PLL倍频至72 MHz高速运行,并独立部署32.768 kHz外部低速补偿晶振为内置RTC模块提供恒定的频率基准,降低本地时钟的累积漂移误差. 主控芯片通过USART接口与协议转换模块进行全双工串行交互,实时捕捉经过解包后的NTP精确时间戳. 通过中断驱动的报文解析算法,系统能够动态校准内部寄存器数值,并经由总线驱动显示模块,最终实现毫秒级授时精度的时间可视化输出.

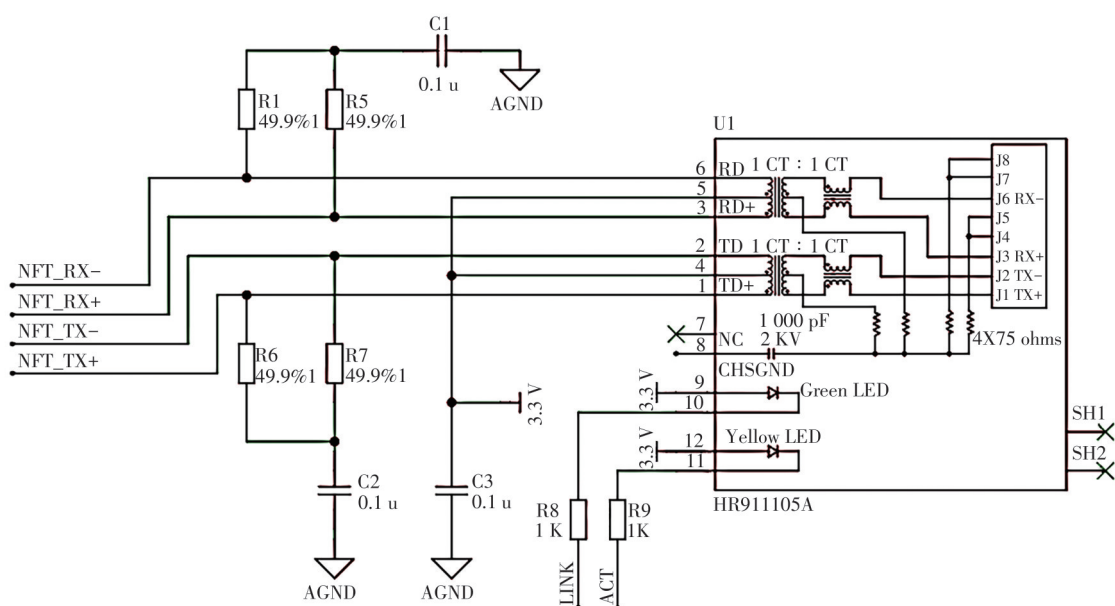


图4 HR911105A网口原理图设计

Fig.4 Schematic design of the HR911105A ethernet interface

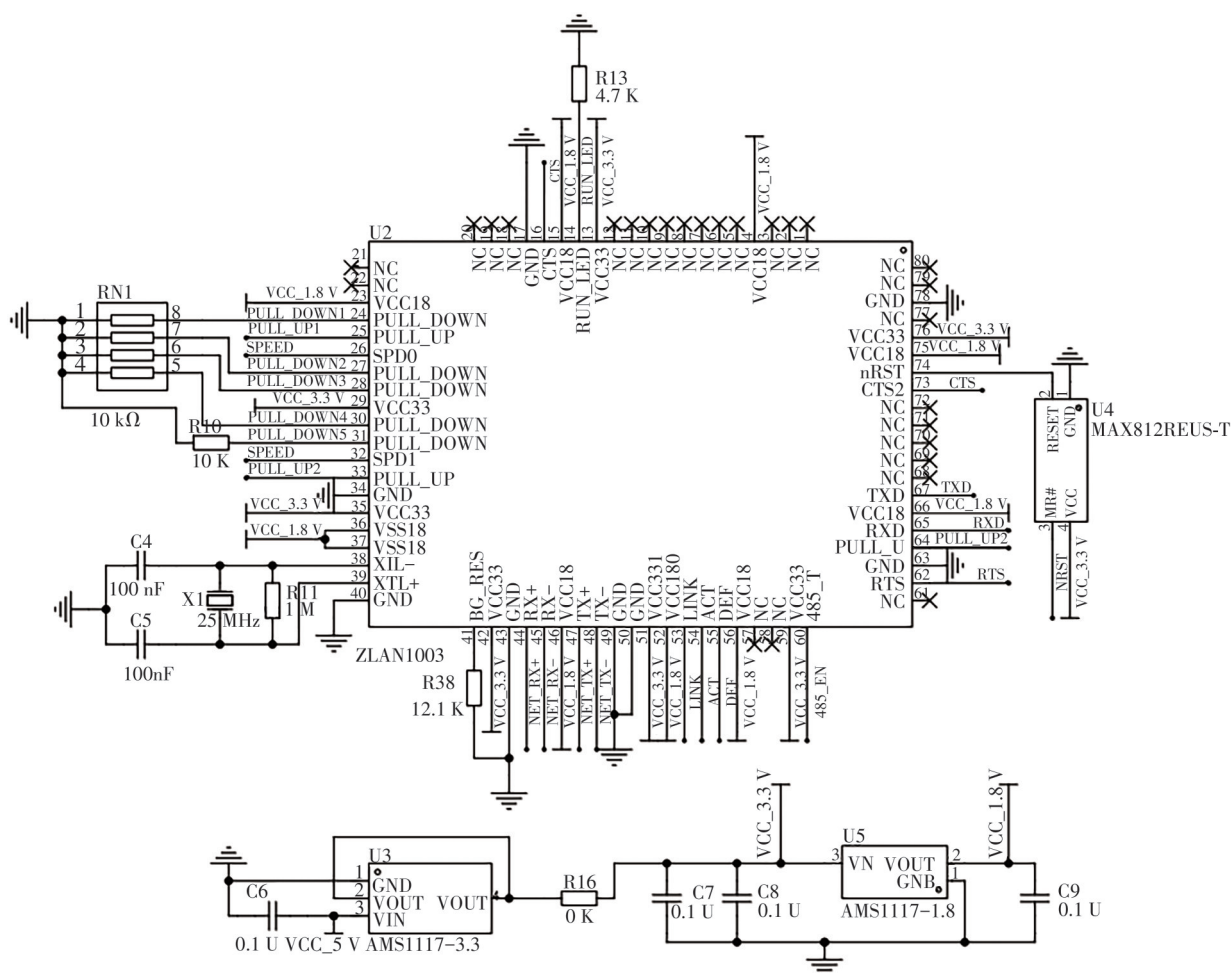


图5 串口转以太网芯片 ZLAN1003 原理图设计

Fig.5 Schematic design of the ZLAN1003 serial-to-ethernet chip

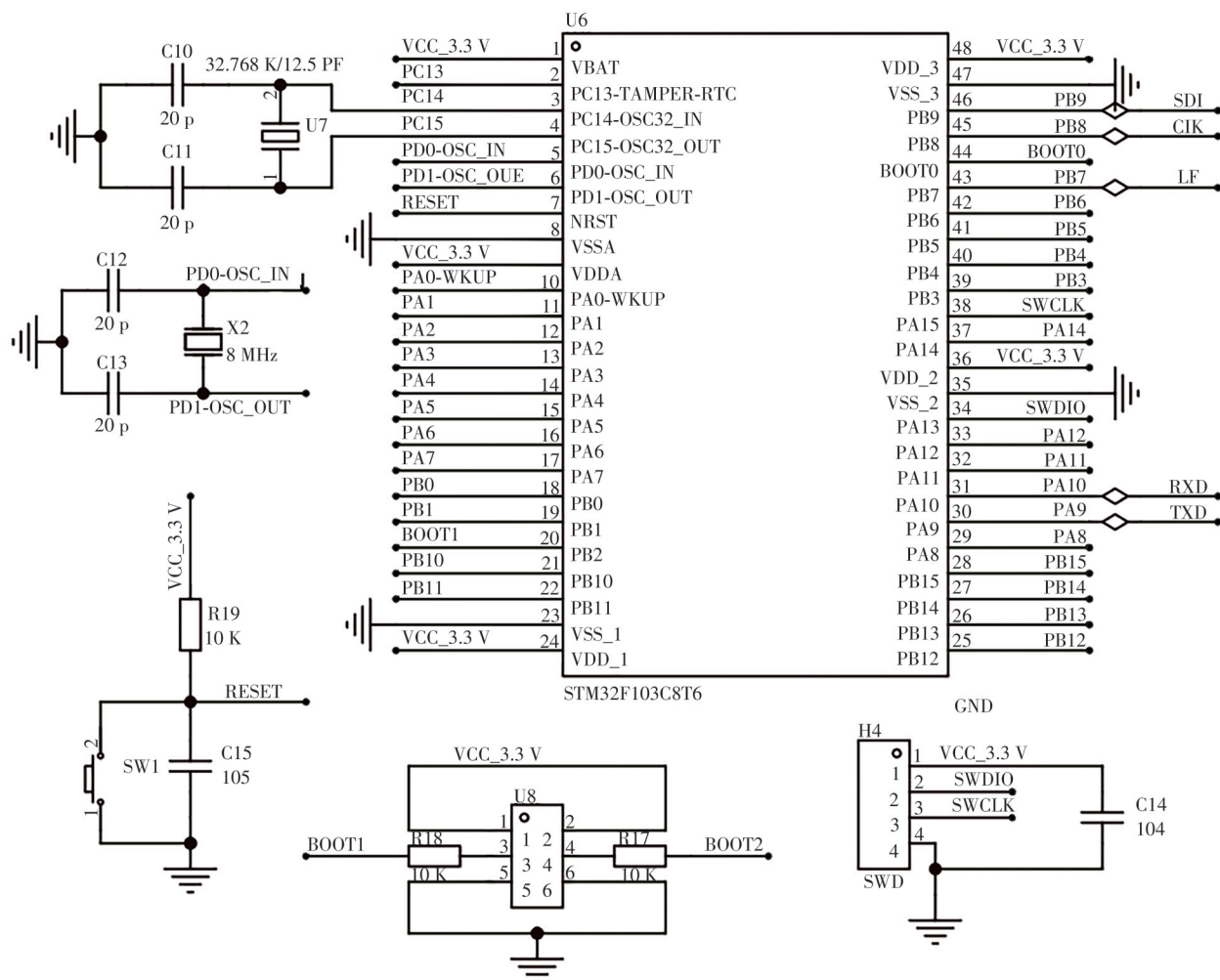


图6 STM32F103C8T6原理图设计

Fig.6 Schematic design of the STM32F103C8T6

5 时间同步系统软件设计

系统程序主要由网络参数配置、NTP请求发送、报文解析、本地时钟同步及数码管显示等部分构成,程序总流程如图7所示.系统启动后首先完成初始化,包括数码管GPIO配置、NVIC中断分组设置、定时器中断开启及115 200波特率串口初始化等;随后通过卓岚配置软件将IP地址、网关及子网掩码写入ZLAN3003S,以建立与NTP服务器的通信连接.系统初始赋予本地时钟一个基准时间,并依托STM32内部晶振维持走时,同时通过10 ms定时中断周期性发送NTP请求.若请求失败且未超过设定重试次数,则继续发送请求;否则判定网络异常,数码管显示故障代码,状态机标记为同步失败.若请求成功,则将串

口接收数据存入缓冲区并提取关键时间字段,当返回的分钟或秒与本地时钟存在差异时,触发时间更新流程;更新前对数据进行逻辑校验以保证有效性,校验通过后完成NTP报文解析,获取时、分、秒信息并同步本地时钟,最终更新数码管显示,状态机标记为同步成功.

6 实验设计与性能评估

实验系统由NTP服务器、嵌入式时钟设备、交换机及上位机组成.如图8所示,将嵌入式时钟连接至服务器,时钟显示模块的时间信息与NTP服务器的时间一致,图中右下角为时钟内部硬件实物图.实验环境中,NTP服务器与嵌入式节点通过局域网连接,网络环境稳定,未引入额外网络负载.

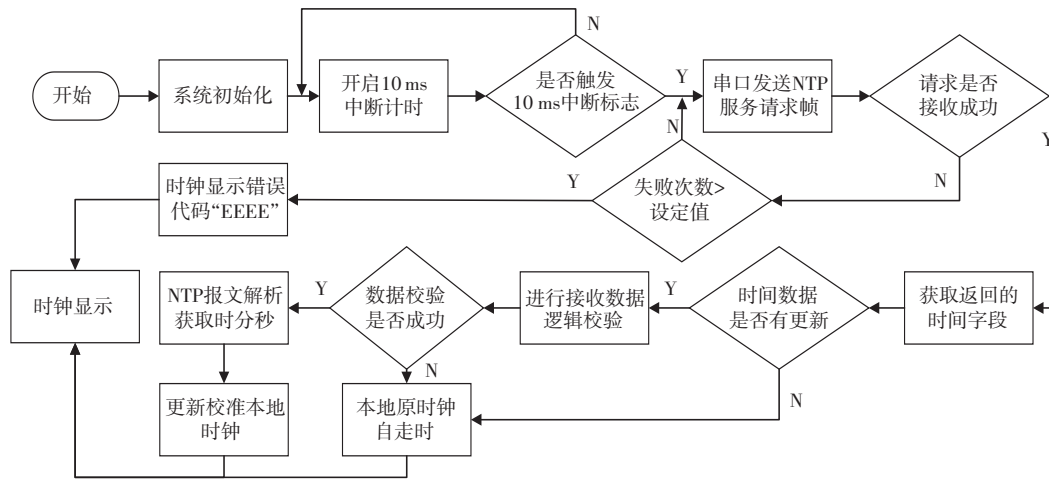


图 7 时间同步系统程序流程

Fig. 7 Program flow of the time synchronization system



图 8 实物连接测试

Fig. 8 Physical connection test

为评估时间同步性能,本文开展了基于嵌入式节点与标准时间源之间的误差测量实验. 实验中以服务器时间作为参考基准,将系统同步误差定义为标准时间与 MCU 本地时间之差,即 $Error = T_{NTP} - T_{MCU}$. 在系统运行过程中,嵌入式节点以 1 s 为周期执行时间同步更新,并通过串口实时输出当前误差值及对应时间信息. 上位机利用串口调试工具对数据进行连续采集与记录,形成时间序列日志数据,日志内容如图 9 所示,用于后续分析与评估.

在整个过程中,共采集约 2 000 组数据,本文采用等间隔抽样的方式选取 500 组样本进行误差曲线绘制. 由图 10 可见,误差整体在约 4 ms 附近小幅波动,未出现明显的累积漂移现象,表明系统具有良好的稳定性. 统计结果并计算出平均误差为 5.38 ms,标准差约为 2.88 ms,最大误差为 14 ms,表明系统误差由一

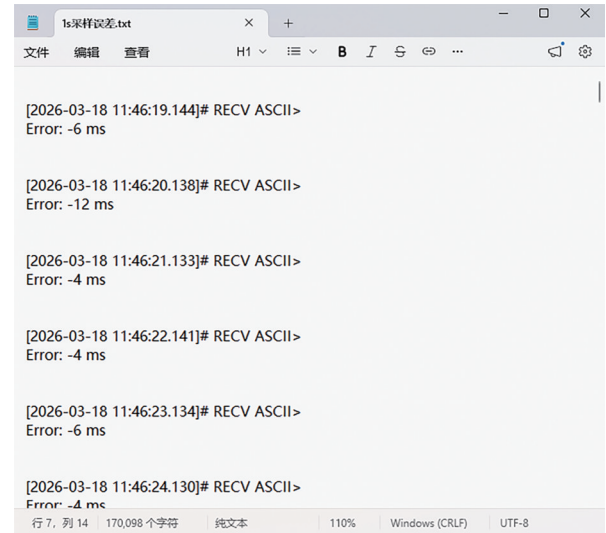


图 9 误差采样日志

Fig. 9 Error sampling log

定的固定偏差与随机波动组成. 该误差主要来源于网络传输时延不确定性、串口通信延迟以及 MCU 任务调度引起的时间偏移.

同样地,在 10 ms 同步周期条件下,系统共采集约 6 000 组数据. 由于高频同步会产生更大量数据,为兼顾数据处理与存储开销,适当缩短了采集时长. 并且为保证对比一致性,同样采用等间隔抽样的方式选取 500 组样本进行误差曲线绘制. 结果如图 11 所示,误差分布更加集中,波动幅度明显减小,未出现显著突发偏移或累积漂移. 统计结果显示,平均误差约为 4.34 ms,标准差约为 2.08 ms,相较于 1 s 同步周期,误差水平及离散程度均有所降低.

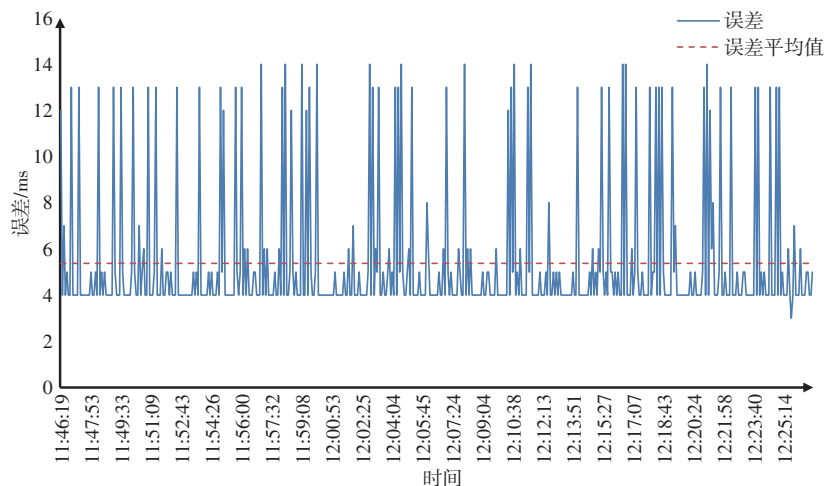


图 10 同步周期为 1 s 的误差曲线

Fig.10 Error curve with 1 s synchronization cycle

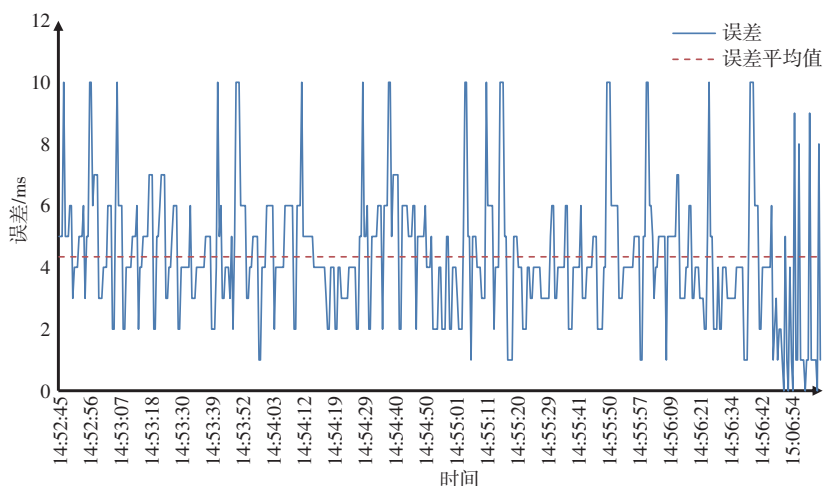


图 11 同步周期为 10 ms 的误差曲线

Fig.11 Error curve with 10 ms synchronization cycle

综合上述实验结果可知,在基于 GNSS 驱动的嵌入式时间同步系统中,系统在不同同步周期条件下均能够稳定运行并实现小于 20 ms 的时间同步精度.相比传统基于软件协议栈的 NTP 实现,本文系统在保证同步精度的同时显著降低了实现复杂度,更适用于资源受限嵌入式场景.

7 结语

本文设计并实现了一种基于 GNSS 驱动的嵌入式时间同步系统.系统采用 NTP 服务器作为时间基准,通过硬件协议栈卸载与报文透明映射机制,实现了时间信息的高效获取与轻量化处理,避免了在嵌入式设备中构建完整 TCP/IP 协议栈,降低了系统复杂度与

资源占用.实验结果表明,系统时间同步误差控制在 20 ms 以内,在不同同步周期下均能稳定运行.该方案在保证同步性能的同时简化了系统结构,适用于校园、医院等多节点嵌入式应用场景.本研究为资源受限环境下的时间同步提供了一种通用的工程实现思路,具有一定的应用价值和推广前景.但系统同步精度仍受网络时延不确定性影响,后续将进一步优化.

参考文献

- [1]Liu Y, Sun B, Wu Y R, et al. Time synchronization techniques in the modern smart grid: A comprehensive survey[J]. Energies, 2025, 18(5): 1163.
- [2]李宗辉, 杨思琪, 喻敬海, 等. 时间敏感网络中确定性传输技术综述[J]. 软件学报, 2022, 33(11): 4334-4355.

- [3] Han K, Meng K T, Masouros C. Over-the-air time-frequency synchronization in distributed ISAC systems [PP/OL]. arXiv (2025-03-11) [2026-03-03]. <https://doi.org/10.48550/arXiv.2503.08920>.
- [4] Bui T T, Ngo H Q T. Enhancing time synchronization between host and node stations using a fuzzy proportional-integral approach for EtherCAT-based motion control systems[J]. PLoS One, 2025, 20(6): e0324939.
- [5] Bode V, Shen W, Gujarati A. EtherTime: Cross-vendor evaluation of PTP/NTP on Ethernet-based COTS embedded platforms[C]//2025 IEEE 31st International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA). Singapore: IEEE, 2025: 169-182.
- [6] Deng J H, Du D J, Zhu M G, et al. A protocol conversion and clock adaptation method between 5G and TSN[C]//IECON 2024-50th Annual Conference of the IEEE Industrial Electronics Society. Chicago: IEEE, 2024: 1-6.
- [7] 张晗, 黄艳, 张磊, 等. 基于 GNSS 的高可靠授时方法研究[J]. 计量科学与技术, 2025, 69(2): 37-44.
- [8] 朱晓章. 卫星授时技术综述: 原理与纳秒级精度实现方法[EB/OL]. [2026-03-23]. https://faculty.uestc.edu.cn/zhuxiaozhang/zh_CN/article/394483/content/4282.htm.
- [9] Pham K. Time scale in alternative positioning, navigation, and timing: New dynamic radio resource assignments and clock steering strategies[J]. Information, 2025, 16(3): 210.
- [10] 葛芙蓉. 基于北斗的时间同步系统设计与实现[D]. 太原: 中北大学, 2022.
- [11] 王震, 满令伍, 张道伟. 基于 NTP 的轻量化时间同步设计[J]. 指挥信息系统与技术, 2022, 13(2): 91-94.
- [12] 赵瑞康, 王素珍, 苗其军, 等. 基于 STM32 的硬件协议 NTP 网络时钟系统实现[J]. 电子器件, 2024, 47(3): 641-648.
- [13] Tang Z W, Yan G F. Research on enhancing the time synchronization accuracy of IEEE 1588 based on asymmetric delay prediction[C]//Proceedings of the 2025 2nd International Conference on Computer Network and Cloud Computing. Nanchang: ACM, 2025: 52-56.
- [14] 王子鉴, 梁坤, 赵俊良, 等. 铁路时间同步网 NTP 授时服务监测技术[J]. 中国铁路, 2025(2): 119-127.
- [15] Shim H, Joo H, Kim K, et al. Provisioning high-precision clock synchronization between UAVs for low latency networks[J]. IEEE Access, 2024, 12: 190025-190038.
- [16] Li Y F, Estebsari A, Hoptroff R, et al. Advancing power system time synchronization: A PTP-based solution with hardware-in-the-loop[J]. IEEE Transactions on Industry Applications, 2026, 62(2): 2062-2073.
- (责任编辑: 张阳, 殷锋, 付强, 和力新, 肖丽; 英文编辑: 郑玉才, 周序林)