DOI:10.20033/j.1003-7241.(2026)07-0133-06

基于数字阵列雷达的 NTP 网络授时服务实现

刘钰妍, 鲍爱达, 郭　鑫, 刘昕沂
(中北大学 微纳器件与系统教育部重点实验室, 太原 030051)

摘　要:针对数字阵列雷达系统各模块间的时间同步需求以及网络时间协议(network time protocol,NTP)在不同硬件平台上的实现问题,本文设计了一种数字阵列雷达系统架构,并重点对其中的时间同步模块进行了详细阐述。为解决传统软件 NTP 同步精度不足的难题,系统引入了精确时间协议(precision timing protocol,PTP)中的硬件时间戳机制,将同步精度由 ms 级提升至 ns 级,并采用改进的最小延迟法对协议进行滤波处理,增强系统在复杂环境下的运行稳定性。在此基础上,文中给出了 NTP 协议在 Zynq-7000 全可编程片上系统(Zynq-7000 all programmable SoC,ZYNQ)平台以及 STM32 实时操作系统(real-time operating system,RTOS)中的具体实现方法,为开发人员在主流 ZYNQ 和 STM32 平台下实现 NTP 协议提供了参考。实验结果表明,该方案实现了高精度且稳定的时间同步,能够满足数字阵列雷达系统对分布式模块协同工作的时间精度要求。

关键词:NTP 协议;PTP 协议;软件调试;ZYNQ;STM32;数字阵列
中图分类号: TP311.56;TN956　　**文献标志码**: A　　**文章编号**: 1003-7241(2026)07-0133-06

Implementation of NTP network timing service based on digital array radar

Liu Yuyan, Bao Aida, Guo Xin, Liu Xinyi
(Key Laboratory of Micro-Nano Devices and Systems, Ministry of Education, North University of China, Taiyuan 030051, China)

Abstract: Aiming at the problems of time synchronization of each module under digital array radar system and the implementation of NTP protocol on different platforms, this paper designs a digital array radar system architecture, with a detailed focus on its time synchronization module. To address the issue of insufficient synchronization accuracy in traditional software-based network time protocol (NTP), the system introduces the hardware timestamping mechanism from the precision timing protocol (PTP), elevating synchronization precision from milliseconds to nanoseconds. An improved minimum delay filtering algorithm is also applied to the protocol to enhance system stability in complex environments. On this basis, the paper presents the specific implementation methods of the NTP protocol on the zynq-7000 all programmable SoC (ZYNQ) platform and within the STM32 real-time operating system (RTOS), providing developers with a reference for implementing the NTP protocol on mainstream ZYNQ and STM32 platforms. Experimental results demonstrate that the proposed scheme achieves high-precision and stable time synchronization, meeting the timing requirements for coordinated operation of distributed modules in digital array radar systems.

Keywords: NTP protocol; PTP protocol; software debugging; ZYNQ; STM32; digital array

随着时代发展,雷达的应用越来越广泛,不同场所领域的需求诞生了多种类多功能的雷达。同时在不同领域的工作和生活中,无数设备要依赖雷达系统的准确性和可靠性,这需要经常对雷达时间进行同步,这对雷达的准确性提出了极高的要求,一旦出现误差就会造成极大的危害。但在实际应用中,由于温度、电磁等各种环境条件的影响,雷达时间是会不断变化的,在极其严格的要求下,传统雷达时钟的精度会降低。因此,研究雷达时间同步是一个非常重要的问题。

时间同步是一个确保各分布系统和终端设备时钟与中心控制系统的标准时钟保持一致的过程,该过程以中心控制系统的标准时钟为基准。NTP 授时是一种专门用来对网络系统做时间同步的协议,将网络系统中各个设备的时间与世界标准时间(coordinated universal time,UTC)所同步,为各系统终端设备提供精确、可靠的时间信息,具有时间精度高、网络安全强、资源开销少、应用较广泛、成本低等特点[1]。在局域网内,由于不能直接访问互联网时间服务器,因此需要建立内部 NTP 服务器来实现时间同步,对于时间敏感的计算处理系统,如控制系统、计算平台和应用服务等,其中的数据或操作都具有严格的时序性,因此,需要同步的标准时间来记录各种事件的发生顺序,并为后续的数据分析提供准确的时间基准。近年来,NTP

收稿日期:2025-01-08;**录用日期**:2025-02-10
基金项目:山西省基础研究计划资助项目(202103021244198);国家自然科学基金资助项目(62204232)
作者简介:刘钰妍(1998—),女,硕士研究生,研究方向为无线通信。
通信作者:鲍爱达(1980—),男,博士,副教授,研究方向为电子测试仪器与系统。
引用本文:刘钰妍, 鲍爱达, 郭鑫, 等. 基于数字阵列雷达的 NTP 网络授时服务实现[J]. 自动化技术与应用, 2026,45(7):133-138. (Liu Yuyan, Bao Aida, Guo Xin, et al. Implementation of NTP network timing service based on digital array radar[J]. Techniques of Automation and Applications, 2026,45(7):133-138.)

协议在不同领域得到了广泛应用。例如,袁小迪等[2]设计了一种基于 NTP 的停车场时间同步系统,让停车场的管理更加高效准确。邓伟等[3]则提出了一种通过 GPS 高精度时间测量来验证 NTP 授时精度的方法,进一步增强了 NTP 协议在时间同步领域的可靠性和准确性。同时,ZYNQ 和 STM32 是被不同领域广泛使用的操作平台,例如,杨发发等[4]设计并实现了一种基于 STM32 控制的全自动静态箱系统,张皓等[5]以 STM32F407 微处理器为控制核心,设计了一款基于智能算法的高性能温度控制器,吕思宇等[6]设计了一种基于电源监控芯片 LTC2945 和 FPGA 的电源监控电路的 FPGA 程序,因此研究相关的平台实现具有实际意义。

本文基于以上背景,对数字阵列雷达系统同步模块进行设计,并为 NTP 在雷达系统上的应用设计了不同的软件平台实现方法,使 NTP 协议能够在不同种类的雷达平台上应用,扩大了 NTP 协议的使用范围。

1 数字阵列雷达系统

如图 1 所示,系统设计采用的数字波束形成(digital beam forming,DBF)技术,实现高效的雷达发射和接收功能。为了支持宽带宽角扫描,系统采用了实时数字延迟技术,该技术不仅适用于宽带信号,也兼容窄带信号,从而取代了传统的移相技术,极大地提高了系统的扫描速度和准确性。在水平面单脉冲测角的应用场景下,系统采用发射单波束的方法来确保信号的集中与方向性,而在接收端,系统通过 DBF 技术形成方位和波束与方位差波束两路信号,以实现对目标角度的精确测量。接收通道的信号首先经过 DBF 模块进行数字波束的合成处理,这一过程是为了增强信号的强度和清晰度,合成后的信号被送入信号处理模块进行后续处理。在信号处理模块中,依次对信号进行匹配滤波、杂波抑制、相参积累和目标检测等关键处理流程,以确保目标信息的准确提取,之后每一帧的检测结果会被打包并传送至显控终端。显控终端在接收到数据后,会进一步完成数据处理工作,包括显示目标的一次或二次轨迹,为操作人员提供直观的目标运动信息[7]。在宽带模式下,显控终端还能利用宽带信号的特性,提供目标距离成像和识别结果。此外,显控终端还具备强大的人机交互界面,操作人员可以通过这一界面方便地对雷达系统进行操作和配置,包括调整扫描参数、设置工作模式,查看系统状态、分析目标数据来满足雷达系统适应各种复杂环境和任务需求。

数字阵列雷达系统的高效运行依赖于一个统一的频率源和精确的时序信号源。这些核心信号和控制指令由主控模块实时生成,以协调雷达系统的各项功能。主控模块产生的控制信号包括详细的控制指令、参数设置、工作时序以及频率参考,确保雷达系统按照预定计划准确运行。

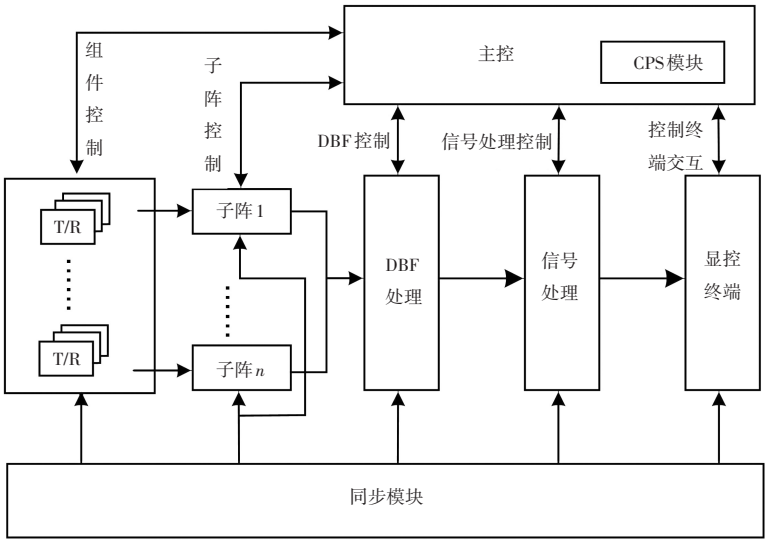


图 1 数字阵列雷达系统架构

Fig. 1 Architecture of the digital array radar system

在系统中,每个模块在接收到指令、时序和频率参考等控制信息后,需要确保在执行命令时能够同时切换,以保证系统的高效和一致性。同时,各个模块间的多路收发数据也需要进行同步处理,以避免数据之间的相对延时,保证数据的实时性和准确性。为实现这一目标,系统配备了同步模块,用于对各功能和数据传输进行同步化处理,确保雷达系统各模块在统一的时序下工作。数字雷达系统网络内的设备通过 NTP 以获取时间同步服务,以确保所有设备的时钟高度一致,以便实现整个数字阵列雷达系统网络在同一时间下的整体调度。

2 NTP 授时原理

NTP 网络授时的核心理念是通过分层分布的架构来实现的,时间信息根据 NTP 服务器的层级逐步传递,各层服务器的时间精度存在差异。层(stratum)的取值范围是 0~15,层的数值越小,优先级越高,层的值表达了节点距特定时钟源的距离,层的数值越大,时间精度越低,所以一般 NTP 的层数都很少[8]。NTP 传输原理如图 2 所示,其

中，T_1 为客户端请求报文的客户端时间，T_2 为服务器接收到客户端报文请求的服务器时间。T_3 为服务器响应客户端报文的服务器时间，T_4 为客户端接收响应报文时客户端时间。β 为客户端时间与服务器时间的偏移，δ 为本次来回通信的网络时延[9]。

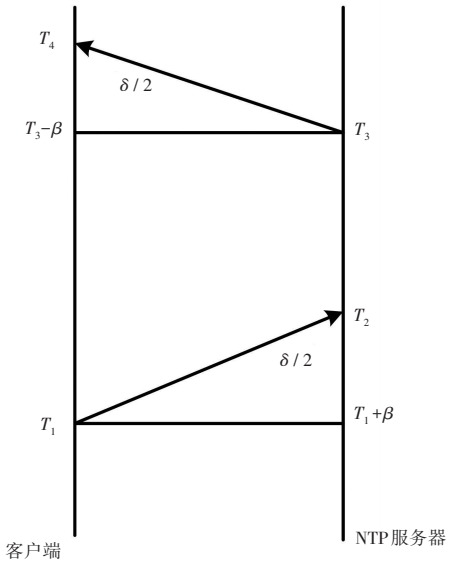


图 2 NTP 传输原理

Fig. 2 Principle of NTP transmission

两次往返时间相同：

$$T_2-(T_1+\beta)=T_4-(T_3-\beta) \tag{1}$$

客户端与服务器时间的偏移：

$$\beta=\frac{(T_2-T_1)-(T_4-T_3)}{2} \tag{2}$$

在时间同步的过程中，客户端使用 NTP 报文中指定的三个时间数值和自身保存的一个时间数，经过一系列计算来确定客户端和服务器之间的恒定时间差，这个时间差将作为时间校准的基础。一旦计算出时间补偿值，客户端会将其封装成一个数据包，并在本地时间系统中应用这个补偿值。终端会根据与服务器的时差来调整内部时钟，以达到与服务器时间的同步，在系统整体中，所有终端将以服务器时间为基准进行同步，通过持续接收服务器发送的 NTP 报文，计算时间差并应用调整值，终端能够不断保持与服务器的同步，确保整体系统时间一致，并最终实现系统时间同步的目标。

2.1 系统 NTP 协议授时

本文设计的数字阵列雷达系统时间同步模块主要由原子钟和 NTP 模块组成。原子钟可以选择 10 MHz 铷原子钟，作为系统内部频率源，FPGA 中的锁相环将该原子钟转换为高度稳定的 250 MHz 时钟，负责为 NTP 模块提供基准。NTP 模块通过内部的 ARM 芯片配置 NTP 服务，一般 ARM 的硬件时钟和系统时钟由晶振驱动，ARM 时钟的准确度由晶振的频率准确度决定，为减少因时钟漂移产生的时间偏差，系统时钟由铷原子钟提供的高精度频率基准 250 MHz 驱动，主控的 GPS 模块提供的 1PPS 是 NTP 服务的精确时间源，通过捕获 1PPS 基准信号可以定时校准 NTP 服务时间[10]。

在 ARM 服务运行时，系统优先处理来自外部的高优先级 1PPS 中断，并利用这些信号对服务器时钟进行定期同步，以提升时钟的长期计时准确度。同时，为了进一步提高性能，选择高主频的 ARM 架构平台，并借助实时任务处理机制来加快指令的处理速度。NTP 使用软件时间戳，意味着数据包的形成和时间戳发生在应用层，NTP 授时精度受 NTP 服务器与客户端间的网络状况影响，为了进一步提升 NTP 协议的精度，可以引入 PTP 协议的硬件时间戳。硬件时间戳是指时间戳的操作发生在硬件或网卡上，这种时间戳的延迟是确定的，时间戳在数据包进行传输之前完成，与系统铷原子钟一起理论上可将 NTP 授时服务精度提升至 ns 级。硬件时间戳结构如图 3 所示。

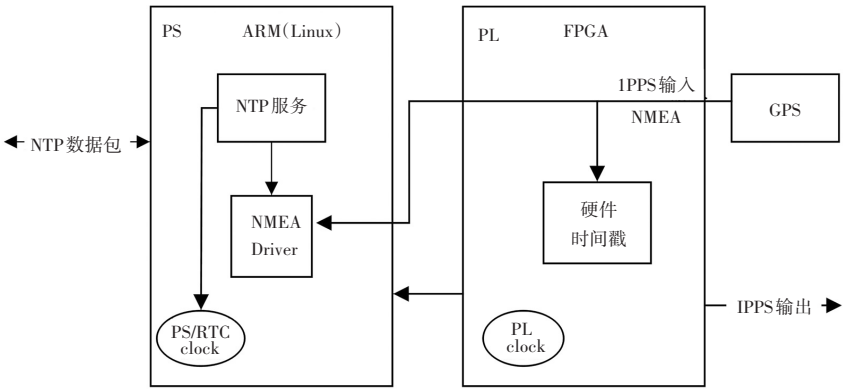


图 3 硬件时间戳结构

Fig. 3 Structure of the hardware timestamp

硬件时间戳主要原理是使用基于 ARM-SoC 的 Xilinx FPGA 和高速数据端口或以太网端口，NTP 协议数据包通过端口到达 FPGA，标记其到达时间，将数据包发送到 ARM 进行处理解析后，在 ARM 中将退出时间戳添加到数据包中加上校验位，经过 FPGA 端口离开，以这种方式为 NTP 包完成硬件时间戳。在时间戳生成部分首先检测到

1PPS 的上升沿,对上升沿进行计数。由于使用的时钟是 250 MHz,所以时间的分辨率是 4 ns 即 1 个计数,因计数器的值不能直接放在协议数据包中,在将其放入数据包之前,必须将计数器值通过乘 4 转换为实际的纳秒值,计数器在 250 MHz 计数的末端或下一个 1PPS 的上升沿增加 1 来保持计时的准确性,在 NTP 数据包中完成时间戳,计算校验并发送数据包来保证数据包的准确时间,以相同的方式使用相同的计数器,存储接收数据包的时间戳,以供参考和计算。

2.2 系统 NTP 协议滤波

当时间同步过程中网络拥塞或网络链路不对称的情况发生时,需要对授时系统进行滤波处理,来降低对时间同步精度产生的影响,NTP 协议滤波普遍采用最小延迟法,该方法是在 8 次 NTP 时间同步过程中选取最小往返延迟对应的时间偏差作为最终同步本地时间的偏差数据[2]。但经过最小延迟法计算得出的结果并不是理论最优值,在高精度系统中会产生较大误差,而过多次采集数据不利于实际系统响应和时间同步,因此对最小延迟法进行改进。

对由最小延迟法得到的结果 $\beta_i(i=1,2,\cdots,8)$ 进行最小二乘法拟合得到斜率 k,α 作为阈值并与 k 进行比较,当 $|k|\leqslant\alpha$,取拟合后的中点值,将 β_{mid} 分别与 β_i 比较,得到 $\beta_{\min}$,$\beta_{\min}$ 为 β_i 与 β_{mid} 相差最小的 β;当 $|k|>\alpha$,选取与最小往返延迟相对应的时间偏差 β,记为 $\beta_{\min}$ 并将其作为最终用于同步本地时间的偏差数据。

采集 NTP 协议的传输数据,采样间隔为 1 s,设定阈值 $\alpha=0.22$,共采集 300 组数据,用最小延迟法和改进后算法的处理计算结果如图 4 所示。

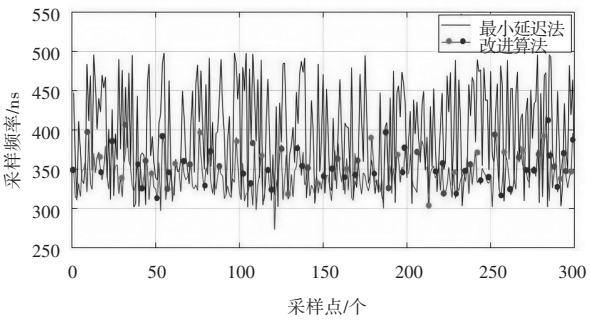


图 4 算法验证仿真

Fig. 4 Simulation for algorithm validation

对比两种算法对 NTP 网络授时精度的改善效果,采用最小延迟法处理后得到的平均时间偏差为 400.37 ns,而改进后的算法,稳定性和精度有明显提高,达到 343.72 ns。

在系统中采用余辉模式多次采样脉冲,时间偏差最大为 340 ns,如图 5 所示。

使用示波器、铷钟频标、频率计数器、时间综合分析仪等设备对系统授时指标进行测试,测试结果如下表 6 所示。

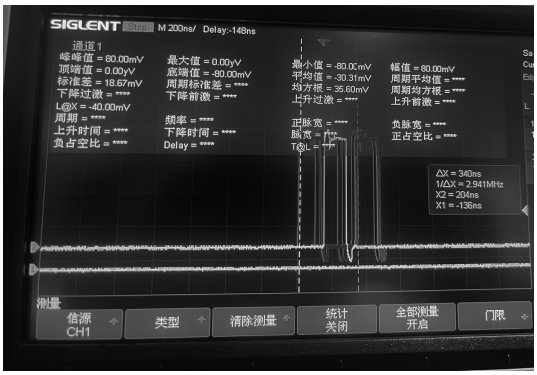


图 5 授时偏差实测

Fig. 5 Measurement of timing deviation

表 1 数据指标实测结果

Tab. 1 Measured data metrics

序号	测试指标	测试范围	实测结果
1	NTP 同步偏差/ms	1~500	47.8
2	PTP 同步偏差/ns	1~500	92.7
3	改进后 NTP 同步偏差/ns	1~500	339.1
4	1pps(TTL)时间偏差/ns	<1 000	43.5
5	1pps(TTL)上升时间/ns	≤100	35.3
6	10 MHz 稳定度(锁定 GPS 信号)/s	$\leqslant5\times10^{-11}$	3.2×10^{-11}
7	10 MHz 准确度(锁定 GPS 信号)/s	$\leqslant5\times10^{-12}$	3.1×10^{-12}

由上表可知,设计实现的授时系统指标在测试中满足需求,且系统经过长时间测试,工作状态正常,授时时间准确。

3 软件平台 NTP 设计实现

在设计完成宽带数字阵列雷达系统的 NTP 协议流程之后,对 NTP 协议进行平台程序实现。面对雷达系统的复杂多变,本文针对不同的软件平台,设计了不同的程序实现方法,列出的实现方法不仅可以应用于雷达系统,在一定条件下还可以应用于其他设备或系统。

3.1 NTP 在 ZYNQ 平台的实现

1) ZYNQ linux 的 NTP 客户端程序。首先进入 ntp 包目录,运行 configure 文件,命令如下:

```
./configure
--prefix=/home/guojr/tools/ntp/ntp-4.2.8p12/install
--exec-prefix =/home/guojr/tools/ntp/ntp-4. 2. 8p12/
```

install

--host = arm-linux-gnueabihf CC = arm-linux-gnueabihf-gcc

--with-crypto

--with-openssl-incdir = /usr/include/openssl/

--with-yielding-select = yes

其中 prefix 和 exec-prefix 都是指定的文件输出路径，With-openssl-incdir 是 openssl 的 include 路径，host 和 CC 是指定的编译器版本，然后交叉编译并执行 make & make install，并在 install/bin 文件夹中会生成 ntpd，ntpdate，将这两个文件拷贝到目标板的 bin 目录下，或直接打包到文件系统中，使用 ntpdate192. 168. 2. 100 即可同步服务器时间，IP 地址需要使用对应 NTP 服务器 IP。最后使用 date 指令查看时间是否同步成功，完成 ZYNQ 平台的客户端 NTP 引入。

2）ZYNQ 裸机 NTP 客户端程序。在 SDK 中创建 lwip echo server 的 demo 程序，将 ntp_demo. c 和 ntp_demo. h 文件加入工程，main 函数中用 lwip( ) 函数替换 start_application ( ) 函数，关闭 DHCP 功能，配置本地 IP，因为 lwip( ) 函数已经设置了 ntp 服务器 IP，正常编译下载即可看到更新的时间，本节所展示的方式在 linux 应用层和 STM32 平台都适用。图 7 为 ZYNQ 平台下的 NTP 操作流程。

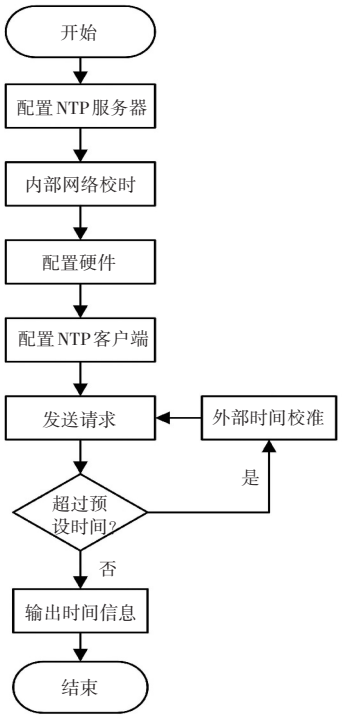


图 6　ZYNQ 平台下 NTP 操作流程

Fig. 6　NTP operational workflow on the ZYNQ platform

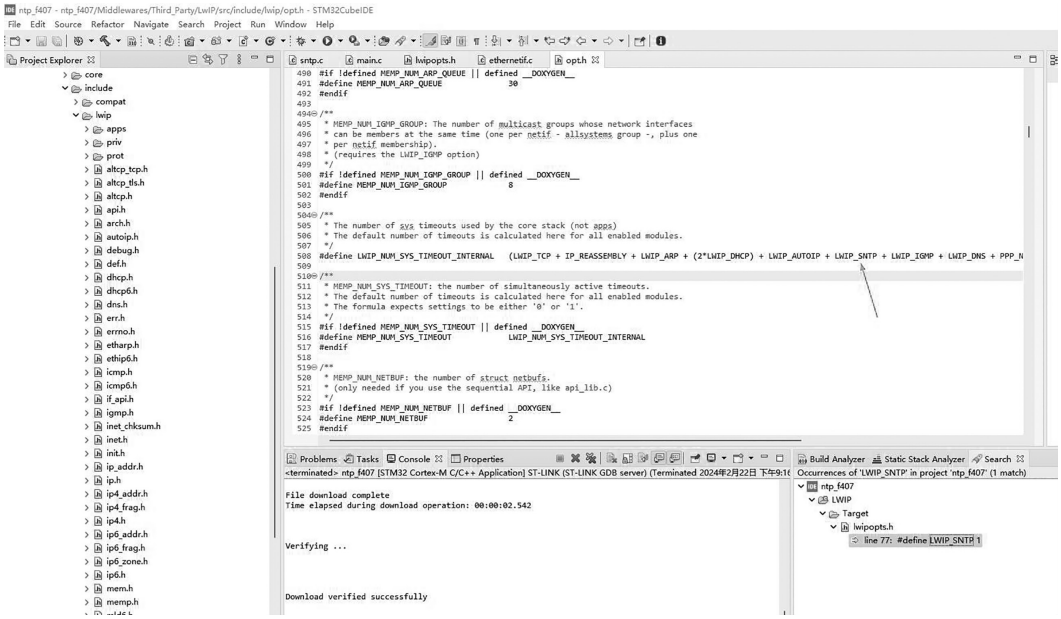


图 7　STM32 平台调试

Fig. 7　Debugging on the STM32 platform

3. 2　STM32 裸机 NTP 客户端程序

由于普遍的 NTP 协议移植是在裸机上应用，本文设计了可以在 STM32 中的系统 RTOS 中运行的方法，具体过程如下：首先使用 STM32 CubeIDE 创建工程，选择 RTOS、ETH 和 LWIP，具体引脚根据硬件配置。在 LWIP 配置中使能 SNTP，并配置好 IP 地址，其他都使用默认配置，确认生成的程序能 ping 通，在 opt. h 中的#defineLWIP_NUM_SYS_TIMEOUT_INTERNAL 中加入 LWIP_SNTP。在 lwipopts. h 中加入以下程序，分别是使能 SNTP 的 debug 模式，指令 SNTP 请求向服务器请求时间间隔，设置 SNTP 用户处理函数。将 SNTP 函数放入启动任务中的 lwip 初始化函数后面。在 RTOS 中，lwip 会通过 MDIO 访问 PHY 芯片检测 RJ45 的 link 状态。

#define SNTP_DEBUG LWIP_DBG_ON

#defineSNTP_UPDATE_DELAY20000

#define SNTP_SET_SYSTEM_TIME(sec) UpdateTime (sec)

之后在 main. c 中定义 UpdateTime 函数，如下所示：

```
void UpdateTime(time_t t)
{
struct tm *t_set;
time_t it;
t += 28800;
t_set = localtime(&t);
it = mktime(t_set);
printf("ntp time:%s\r\n",ctime(&it));
}
```

在 main 函数中调用 sntp_setserver 和 sntp_init 函数设置 NTP 服务器 IP 并初始化 NTP。在 while(1)中调用 MX_LWIP_Process(),如图 8 所示。

```
  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
// MX_USART2_UART_Init();
  MX_LWIP_Init();
  /* USER CODE BEGIN 2 */
  MX_USART2_UART_Init();

  ip_addr_t newip;
  newip.addr = (100<<24)|(2<<16)|(168<<8)|192;
  sntp_setserver(0, &newip);

  sntp_init();

  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
     MX_LWIP_Process();

  }
  /* USER CODE END 3 */
}

/**
  * @brief System Clock Configuration
  * @retval None
```

图 8 MX_LWIP_Process( ) 函数

Fig. 8 The MX_LWIP_Process( ) function

4 结论

本文以 NTP 协议为基础,对数字阵列雷达系统有关 NTP 协议的同步进行了设计,引入 PTP 协议的硬件时间戳提升 NTP 协议至纳秒级,加入改进后的最小延迟法来保证授时稳定性,验证了授时精度,并对 NTP 协议在 ZYNQ 和 STM32 平台 RTOS 系统的实现给出了操作过程。针对目前 NTP 协议在不同设备、系统和软件环境中不断增加的应用需求,本文设计的实现方法可应用于不同的场景。

参考文献

[1]卫杨, 张辉, 徐红. 基于 NTP 协议的车联网高精度时间同步系统解决方案探索[J]. 数字通信世界, 2022(1):13-15,18.

[2]袁小迪, 宋文霞, 张越. 基于 NTP 的停车场时间同步系统研究与设计[J]. 计量科学与技术, 2021, 65(5):96-100.

[3]邓伟, 翁子凡. 基于 GPS 接收机的 NTP 网络授时精度测量[J]. 软件, 2014, 35(11): 82-85.

[4]杨发发, 燕振刚, 王倩, 等. 基于 STM32 的农田 CO_2 通量测量全自动静态箱系统设计与实现[J/OL]. 自动化技术与应用, 1-10. [2025-12-24]. https://link. cnki. net/urlid/23. 1474. tp. 20250902. 1701. 045.

[5]张皓, 唐军, 唐翠微, 等. 基于 STM32 的高性能温度控制器算法研究与实现[J]. 自动化技术与应用, 2025, 44(4):1-6.

[6]吕思宇, 王传宇, 刘伟, 等. 基于 LTC2945 的电源监控电路 FPGA 开发[J]. 自动化技术与应用, 2025, 44(4):89-93.

[7]邹林. 宽带数字阵列雷达关键技术研究[D]. 成都:电子科技大学, 2013.

[8]白杉杉, 胡永辉, 徐锐. 基于 Wi-Fi 的 NTP 无线授时服务器的设计与实现[J]. 时间频率学报, 2013, 36(4):229-235.

[9]邓伟. 基于 GPS 时钟同步系统的实现与应用[D]. 北京:北京邮电大学, 2015.

[10]宋宇航, 刘敏, 吴宏硕, 等. 船用时统设备 NTP 网络授时服务的实现[J]. 导航定位与授时, 2021, 8(1):163-167.

(上接第 113 页)

参考文献

[1]赵英杰, 胡以君, 沈建, 等. 基于 SIMIT 与 NX MCD 的机器人工作站虚拟调试研究[J]. 机电工程技术, 2024, 53(10):47-51.

[2]刘敏, 张立冬, 郑婕, 等. 智慧工厂数字孪生的设计与实现[J]. 自动化技术与应用, 2024, 43(5):52-56.

[3]张泽坤, 孙渊, 王寻. 采球机设备虚拟调试设计与仿真研究[J]. 机床与液压, 2025, 53(10):222-230.

[4]乔阳, 敖冰峰, 杨宏帅, 等. 基于虚拟仿真技术的智能制造生产线自动控制系统[J]. 自动化与仪表, 2024, 39(5):45-48, 53.

[5]陈志澜, 唐昊阳. 机床上下料系统的数字化孪生设计与仿真[J]. 机械设计与研究, 2024, 40(2):7-14.

[6]邓飞, 李文昊, 蔡文龙, 等. 基于 MCD 的爆震轴类零件检测维修线虚拟调试仿真实验[J]. 实验技术与管理, 2023, 40(11): 123-130, 141.

[7]李雪铭, 洪俊杰, 王泓凯. 基于数字孪生的物料分拣教学实验平台设计[J]. 现代机械, 2024, (6):44-47.

[8]林裕程, 韩勇. 基于 NX MCD 的数控机床虚拟调试[J]. 制造技术与机床, 2021, (2):151-156.

[9]谢小正, 张旭, 孙子宽, 等. 数控机床伺服系统参数设计数字孪生实验平台[J]. 实验技术与管理, 2024, 41(5):113-120.

[10]章丽红. 基于 PLCSIM Advanced 和 TIA Portal V16 的开放式以太网通信仿真设计[J]. 工业控制计算机, 2024, 37(5):26-27, 30.

[11]王贺彬, 白锐, 吕永津, 等. 基于 PLC 与触摸屏的空压站自动监控系统设计[J]. 制造业自动化, 2023, 45(10):125-128.

[12]王晓瑜. 基于 PROFIBUS-DP 总线技术的主从站变频调速控制系统设计与实现[J]. 自动化技术与应用, 2025, 44(11):37-39.

[13]甄文斌, 谢建民, 李梓平, 等. 智能草沙障固沙机控制系统的设计与试验[J]. 农机化研究, 2025, 47(8):61-68.

[14]李浩宇, 王少锋, 刘慧, 等. 基于 PLC 的双层葵花籽烘干机控制系统设计[J]. 包装与食品机械, 2023, 41(5):71-75, 81.

[15]苏雅拉图. 农作物秸秆收获机控制系统的设计与试验研究[J]. 农机化研究, 2025, 47(2):91-95.

[16]王志远, 杨明川, 陈鑫. S7-SCL 语言在 PLC 编程中的应用研究[J]. 机电工程技术, 2024, 53(7):321-324.

[17]晏华成, 芮庆忠, 龙涛元. 基于 C#和 S7-1200 PLC 的立体仓库控制系统设计[J]. 自动化技术与应用, 2026, 45(1):6-11.