

## STRS 系统异构处理器通信中间件的设计与实现\*

钱宏文<sup>1</sup>,朱江伟<sup>1</sup>,吴翼虎<sup>1</sup>,杨旭<sup>2</sup>,方海<sup>2</sup>,陈显舟<sup>2</sup>

(1.中国电子科技集团公司第五十八研究所,江苏无锡 214072;2.中国空间技术研究院西安分院,陕西西安 710100)

**摘要:** 针对空间通信无线电系统(STRS)异构处理器间通信存在的实时性不高、冗余度较大、无法故障恢复等问题,将分布式数据分发服务(DDS)中间件技术引入到STRS架构中,实现STRS异构处理器波形应用组件之间基于发布/订阅模式的通信中间件。在完全兼容STRS标准规范的前提下,有效地提高了基于STRS的通信系统消息传递的实时性,降低了系统的复杂度和冗余度,提高了开发效率,节省了系统的开发和维护成本,实现了全局和局部模块的动态重构。

**关键词:** 分布式数据分发服务;空间通信无线电系统;发布/订阅

**中图分类号:** TN402

**文献标志码:** A

**DOI:** 10.16157/j.issn.0258-7998.222757

**中文引用格式:** 钱宏文,朱江伟,吴翼虎,等. STRS系统异构处理器通信中间件的设计与实现[J]. 电子技术应用, 2023, 49(2):92-96.

**英文引用格式:** Qian Hongwen, Zhu Jiangwei, Wu Yihu, et al. Design and implementation of communication middleware of heterogeneous processors in STRS system[J]. Application of Electronic Technique, 2023, 49(2):92-96.

## Design and implementation of communication middleware of heterogeneous processors in STRS system

Qian Hongwen<sup>1</sup>, Zhu Jiangwei<sup>1</sup>, Wu Yihu<sup>1</sup>, Yang Xu<sup>2</sup>, Fang Hai<sup>2</sup>, Chen Xianzhou<sup>2</sup>

(1.China Electronics Technology Group Corporation No.58 Research Institute, Wuxi 214072, China;

2.Xi'an Branch, China Academy of Space Technology, Xi'an 710100, China)

**Abstract:** Aiming at the problems of low real-time performance, large redundancy, and inability to recover from faults in the communication between heterogeneous processors in the Space Telecommunication Radio System (STRS), the distributed data distribution service (DDS) middleware technology is introduced into the STRS architecture to realize the publish/subscribe mode-based communication middleware between the STRS heterogeneous processor waveform application components. Under the premise of being fully compatible with the STRS standard specification, it effectively improves the real-time performance of the communication system based on STRS, reduces the complexity and redundancy of the system, improves the development efficiency, saves the development and maintenance costs of the system, and realizes the dynamic refactoring of global and local modules.

**Key words:** data distribution service; space telecommunication radio system; publish/subscribe

### 0 引言

随着卫星业务需求的不断增长,对卫星的灵活性提出了越来越高的要求,空间通信无线电系统(Space Telecommunication Radio System, STRS)架构标准采用开放式架构,软件和硬件体系结构分离设计<sup>[1]</sup>。基于STRS的卫星任务系统能够在卫星发射后改变卫星任务功能,使卫星功能具备软件定义能力,可以满足卫星多样性的业务需求。

STRS波形组件使用STRS消息传递方法建立消息

传递通道。STRS基础架构提供了两种消息传递模式,一种是STRS队列,另一种是发布/订阅模式<sup>[1]</sup>。分布式数据分发服务(Data Distribution Service, DDS)中间件遵循发布/订阅设计模式<sup>[2]</sup>。本文利用DDS的特点,将DDS应用到STRS架构中,实现STRS系统通信中间件,满足STRS系统强实时性的要求,实现STRS波形组件的分布式部署,很好地弥补了STRS架构标准存在的不足。

### 1 STRS概述

#### 1.1 STRS架构标准

STRS架构标准采用模块化和分层的设计思想,目

\* 基金项目:国家重点研发计划项目(2020YFB1808003)

的是提高波形应用设计的通用性和可移植性,分为硬件和软件两大体系结构<sup>[1]</sup>。

STRS 硬件架构如图 1 所示,由三大模块组成,分别为通用处理模块(General-purpose Processing Module, GPM)、信号处理模块(Signal Processing Module, SPM)和射频模块(Radio Frequency Module, RFM)<sup>[1]</sup>。通用处理模块是主要的控制模块,信号处理模块主要用于将接收到的数字信号转换成数据分组或将数据分组转换为数字信号并进行发送的信号处理,射频模块主要功能包括频率转换和增益控制、模数和数模转换、辐射跟踪等<sup>[3]</sup>。GPM、SPM 和 RFM 3 个模块共同组成了包括 CPU、FPGA、DSP 等处理器的异构处理平台。

STRS 软件体系结构定义了运行时不同的波形应用程序软件和操作环境之间的软件接口层以及操作环境和硬件平台之间的接口,由应用层、核心层和物理层组成<sup>[4]</sup>。

## 1.2 不采用通信中间件 STRS 系统的不足之处

STRS 定义了统一的开发标准架构,使得软硬件可分离,目的是:在不进行硬件平台重新设计的情况下,在同一硬件平台上加载不同的波形组件以实现不同的功能,也可根据特定的需求对系统进行局部或全局的动态重构,从而最大化地利用系统资源。从工程实践来看,不采用通信中间件的 STRS 存在以下不足:

(1)耦合度高:STRS 系统未明确说明如何实现消息传递,软件开发人员大多数会首选手动编写代码的方式来实现波形组件应用程序间的数据交互,这样每一个连接都需要开发人员来维护。每增加一个节点,系统的耦合性呈指数级增长。

(2)灵活性较差:STRS 要求所有的波形组件作为一

个整体创建、运行,如果波形的某一个组件损坏,整个应用将不能正常运行,存在单点失败的风险,必须要由管理员进行整套系统的卸载和重新部署。

(3)组件间同步耗时较长:为了建立组件之间的同步连接,当组件较多时,过程会更复杂,开销更大,导致波形组件启动较慢<sup>[5]</sup>。

## 2 基于 DDS 的 STRS 系统通信中间件的设计与实现

将 DDS 引入 STRS 系统,实现 STRS 系统发布/订阅方式的通信中间件。STRS 系统通信中间件屏蔽了底层传输控制协议/网际协议(Transmission Control Protocol/Internet Protocol, TCP/IP),通过简单的编程就可以构建一个以数据为中心、可扩展、与位置无关的分布式网络平台<sup>[5]</sup>,为 STRS 系统各个波形组件提供了简单、高效的消息传递服务。

### 2.1 基于 STRS 硬件架构的 DDS 设计实现

DDS 使用操作系统适配层和网络适配层来实现多个平台的支持,操作系统适配层为 DDS 提供运行时所需的内存申请、定时器、多线程等功能接口。不同的操作系统实现 DDS 操作系统适配层的接口均不相同,如 Linux、SYS/BIOS。DDS 操作系统适配层默认支持可移植操作系统接口(Portable Operating System Interface, POSIX),在支持 POSIX 接口的环境中,只需要做细小的修改就可以实现 DDS 操作系统适配层<sup>[6]</sup>。DDS 网络适配层基于不同的操作系统提供的网络编程接口实现<sup>[7]</sup>,如 Linux 操作系统下的 Socket 接口、SYS/BIOS 操作系统下的网络开发工具(Network Developer's Kit, NDK)等。

#### 2.1.1 基于 GPM 通用处理模块的 DDS 设计实现

GPM 模块采用 Xilinx 公司的系统级芯片(System

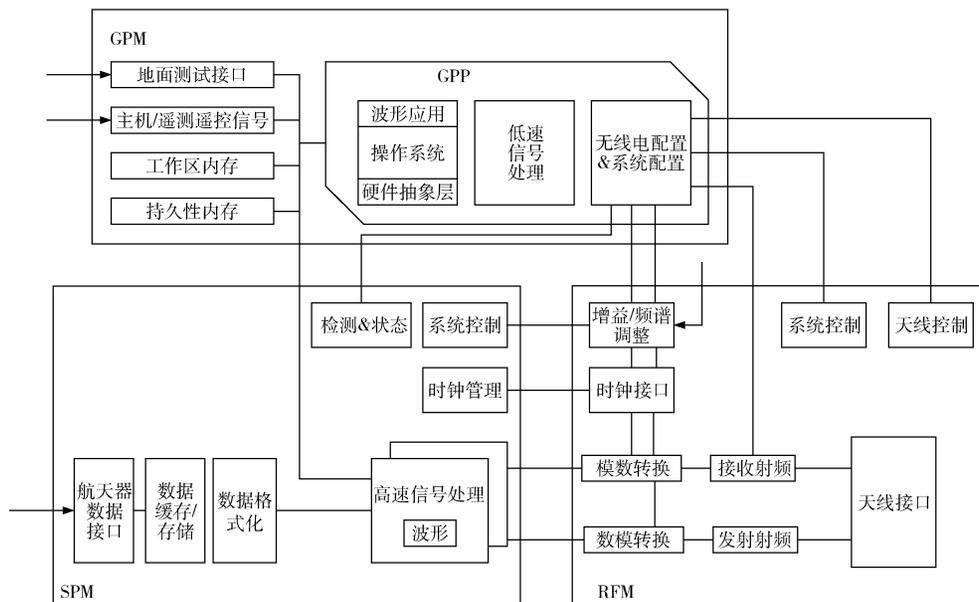


图 1 STRS 硬件架构

on Chip, SoC)作为主处理器。在GPM模块运行PetaLinux<sup>[8]</sup>,提供了DDS运行所需的POSIX接口和网络套接字Socket接口,DDS基于POSIX和Socket接口实现操作系统适配层和网络适配层,完成DDS在GPM PetaLinux环境中的实现。基于GPM通用处理模块的DDS软件基础结构如图2所示。

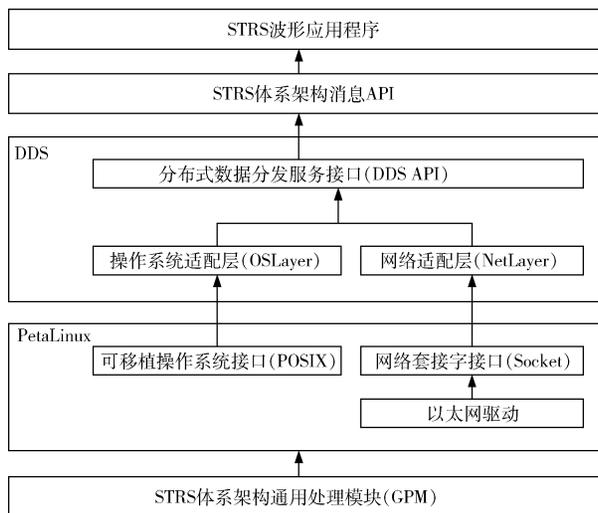


图2 基于GPM模块的DDS软件基础结构

### 2.1.2 基于SPM信号处理模块的DDS设计实现

SPM模块主处理器采用TI公司多核DSP,在SPM模块上运行多核SYS/BIOS。SYS/BIOS提供了DDS运行所需的POSIX接口<sup>[9]</sup>和NDK<sup>[10]</sup>,DDS基于POSIX接口和NDK组件实现操作系统适配层和网络适配层,完成DDS在SPM SYS/BIOS中的实现<sup>[11]</sup>。基于SPM模块的DDS软件基础结构如图3所示。

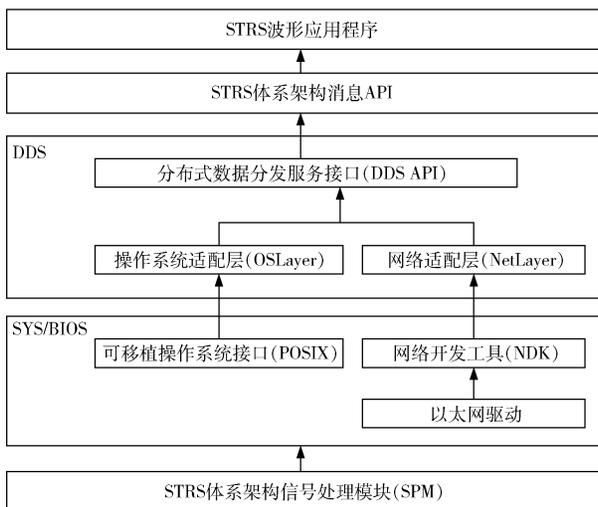


图3 基于SPM模块的DDS软件基础结构

### 2.1.3 基于RFM射频模块的DDS设计实现

RFM模块主要器件为Xilinx公司的高端FPGA芯

片。基于RFM模块,使用FPGA资源搭建MicroBlaze<sup>[12]</sup>软核,PetaLinux支持MicroBlaze,提供了对应的BSP<sup>[13]</sup>。PetaLinux提供了DDS运行所需的POSIX接口和网络套接字Socket接口,DDS基于POSIX和Socket接口实现操作系统适配层和网络适配层,完成DDS在RFM模块PetaLinux中的实现。基于RFM模块的DDS软件基础结构如图4所示。

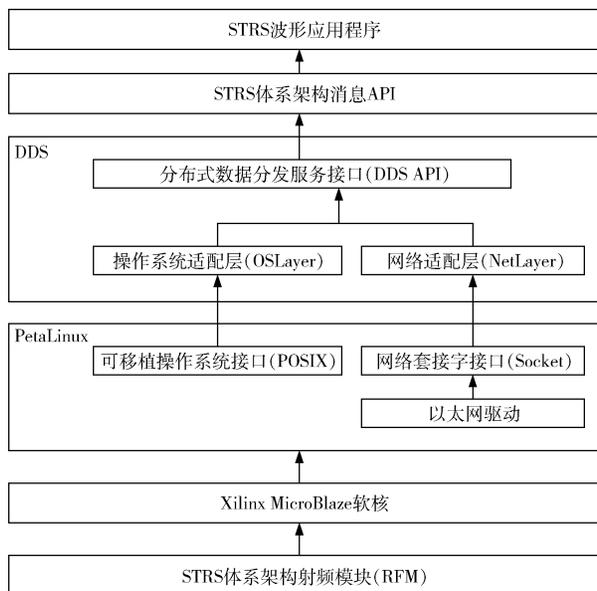


图4 基于RFM模块的DDS软件基础结构

## 2.2 基于DDS的STRS系统通信中间件的软件设计

基于STRS的波形组件应用程序的软件设计包括STRS核心框架设计和基于DDS的STRS系统通信中间件的接口设计两部分。

### 2.2.1 STRS核心框架设计

基于STRS系统通信中间件,STRS核心框架加载波形应用程序时只加载并运行各个波形组件,不需要建立波形之间的通信链路,也不需要处理波形之间的数据同步,包括消息传递在内的工作都交由DDS来完成。基于DDS的STRS软件架构如图5所示。

### 2.2.2 STRS系统通信中间件接口设计

DDS接口设计对于实现STRS系统中的发布/订阅通信方式来说是一项至关重要的工作。STRS核心框架只负责波形组件的加载、部署和运行,因此在STRS消息接口设计中,需要完成STRS消息接口定义、波形组件间的通信链路建立和波形组件间的数据同步这三方面的工作。

#### (1)STRS系统通信中间件接口定义

DDS使用接口定义语言(Interface Definition Language, IDL)<sup>[2]</sup>进行数据结构的定义。根据波组件需要传输的数据类型以及业务,使用IDL定义数据结构和主题,并使用DDS代码自动生成工具加载IDL文件生成处

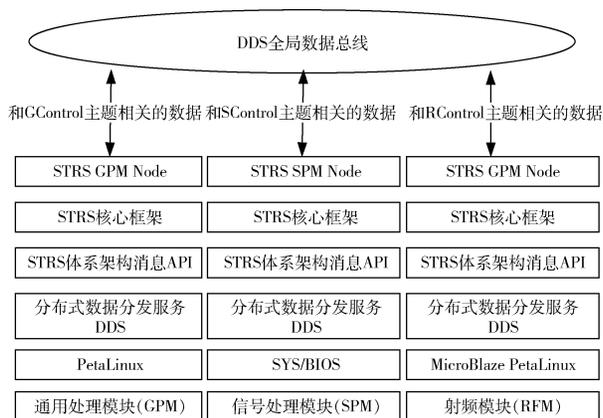


图5 基于DDS的STRS软件架构

理数据结构相关的代码,通过对生成代码的封装来实现STRS消息接口,通过这种方式实现STRS中所有用来进行数据传输的接口。由于基于DDS定义的STRS系统通信中间件接口只是实现了系统内部的消息传输,并且都在各个组件的代码中实现,遵循STRS架构标准,并不会影响到STRS架构,完全兼容。

### (2) 波形组件间的通信链路建立

基于STRS系统通信中间件实现的STRS波形组件是位置无关的,波形组件之间的互联互通通过“主题”+“数据结构”作为唯一标识。波形组件间的通信链路建立通过“发现”机制<sup>[14]</sup>来实现,使用周期性的“心跳”来保证参与者的存活性,参与者在网络上通过“心跳”周期性地宣告他们的存在。一旦两个参与者相互发现后,一个可靠的通信链路建立。在STRS系统的开发过程中无需考虑组件间的连接建立问题。

### (3) 波形组件间的数据同步

在进行波形组件的开发过程中,会遇到这样的问题,先启动订阅端,再启动发布端,往往订阅端无法收到第一包数据,这时就无法保证数据的完整性。针对上述问题,STRS系统通信中间件提供了解决方案,发布端提供了一个接口on\_publication\_matched,通过这个接口可以获得当前与发布端建立通信链路的订阅端的个数,软件开发者可以通过对与发布端建立通信链路的订阅端的个数的判断来决定是否开始发布数据,当订阅端个数大于0时开始发布数据,这样订阅端就会收到所有数据。这样很容易地解决了波形组件间的数据同步问题,保证了数据的完整性。

## 2.2.3 STRS系统通信中间件的性能测试

基于可重构综合处理器平台,使用STRS系统通信中间件接口,在GPM模块和RFM模块之间进行STRS系统通信中间件接口的性能测试,包括:波形组件间的通信链路建立时间、丢包率和传输延时<sup>[15]</sup>,测试结果如下。

### (1) 波形组件间的通信链路建立时间测试

构建一对一发布-订阅场景,发布端mainPublisher一直循环发送数据,订阅端mainSubscriber记录时间 $T_1$ 后马上使能,接收到第一包数据记录时间 $T_2$ ,波形组件间的通信链路建立时间为 $T_2-T_1$ ,测试结果显示通信链路建立时间为420 ms。

### (2) 丢包率测试

丢包率表示在Best-effort QoS<sup>[16]</sup>的配置下,STRS系统通信中间件接口的丢包情况。构建一对一发布-订阅场景,发布端循环 $n$ 次发布数据,订阅端每收到一包数据记录一次,根据收到的数据情况统计丢包率。实验中分别测试了单包大小分别为1 KB、4 KB、16 KB、32 KB、64 KB、1 MB的丢包率,均未出现丢包情况。

### (3) 传输延时测试

测试传输延时是为了证明STRS体系架构对消息接口实时性的要求。基于STRS系统通信中间件接口构建一对一请求-回复场景,回环测试,每次发布和订阅数据使用信号量进行同步。实验中分别测试了单包大小分别为1 KB、4 KB、16 KB、32 KB、64 KB、1 MB的传输延时,测试结果如表1所示。

表1 STRS消息接口传输延时测试结果( $\mu\text{s}$ )

单包数据大小	传输延时
1 KB	612
4 KB	841
16 KB	1 487
32 KB	2 310
64 KB	3 828
1 MB	18 348

## 2.3 STRS系统通信中间件的优点

STRS系统通信中间件有如下优点:

(1) 提高通信效率,降低网络延迟:STRS系统中的所有波形组件之间是一种对等网的关系,不需要中心服务器的连接,不存在单点失败风险,而且底层协议栈并不复杂,不需要太大的软件开销,这大大提高了通信的效率和实时性。

(2) 降低系统的冗余度,提高了灵活性:各个波形组件之间是松耦合的关系,参与通信的节点并不知道对方的信息,节点之间的通信仅仅需要一个主题就能完成。

(3) 单个波形组件可重构:在STRS系统中,每个波形组件的加入和退出均不会影响整个系统的功能和业务。波形组件启动后,组件间只需通过“主题”就可以完成数据交互,当某一个波形组件故障时,可以通过显控软件完成单个波形组件的重启,进而实现单个组件模块的重构。

(4) 减少了组件间同步时间:STRS系统通信中间件接口的性能测试结果表明STRS系统中的各个波形组件

启动后,在极短的时间内就可完成相互发现和同步,所以,使用STRS系统通信中间件极大地减少了组件间的同步时间。

### 3 结论

本文首先介绍了STRS架构标准,并根据工程实践经验总结出了目前不采用通信中间件STRS系统中存在的问题,即耦合度高、实时性不强、灵活性较差、组件间同步耗时较长。重点介绍了基于DDS的STRS系统通信中间件的设计与实现,最后对STRS系统通信中间件接口进行了性能测试,证明了STRS系统通信中间件的优点。本文的创新点在于:基于STRS硬件体系架构,在GPM、SPM和RFM 3种异构处理器平台上基于DDS完成了STRS系统通信中间件的设计与实现,并采用STRS系统通信中间件来实现波形组件间的通信,降低了系统的冗余度和复杂度,提高了系统的实时性、灵活性和可靠性。

### 参考文献

- [1] National Aeronautics and Space Administration. Space Telecommunications Radio System (STRS) architecture standard[S/OL]. (2018-03-14)[2022-03-17]. <https://standards.nasa.gov/sites/default/files/standards/NASA/w/Change-1/1/Historical/nasa-std-4009a.pdf>.
- [2] Object Management Group. Data distribution service for real-time systems. Version1.4 [EB/OL]. (2015-03)[2022-03-17]. <https://www.omg.org/spec/DDS/1.4>.
- [3] 胡行毅.NASA STRS开放架构硬件模块[J].卫星与网络,2008(9):62-66.
- [4] 周力,魏急波,唐麒,等.空间软件无线电标准和体系架构研究[J].卫星与网络,2018(5):38-40,42-43,45.
- [5] 李欣宇.DDS在SCA系统中的应用[J].计算机工程与设计,2013,34(6):1931-1935.
- [6] RTI. RTI connext micro 3.0.3 users manual[EB/OL]. [2022-03-17]. <https://community.rti.com/static/documentation/connext-micro/3.0.3/doc/html/usersmanual/index.html>.
- [7] EPROSIMA. eProsim micro XRCE-DDS user manual [EB/OL]. (2020-05-XX) [2022-03-17]. <https://micro-xrce-dds.docs.eprosima.com/en/latest/#eprosima-micro-xrce-dds>.
- [8] Xilinx. PetaLinux tools documentation reference guide [EB/OL]. (2021-02-XX) [2022-03-17]. [https://china.xilinx.com/content/dam/xilinx/support/documentation/sw\\_manuals/xilinx2021\\_2/ug1144-petalinux-tools-reference-guide.pdf](https://china.xilinx.com/content/dam/xilinx/support/documentation/sw_manuals/xilinx2021_2/ug1144-petalinux-tools-reference-guide.pdf).
- [9] TI. TI-POSIX user's guide [EB/OL]. [2022-03-17]. [https://software-dl.ti.com/ecs/SIMPLELINK\\_CC32XX\\_SDK/4\\_40\\_00\\_07/exports/docs/tiposix/Users\\_Guide.html](https://software-dl.ti.com/ecs/SIMPLELINK_CC32XX_SDK/4_40_00_07/exports/docs/tiposix/Users_Guide.html).
- [10] TI. Running NDK examples for KeyStone devices[EB/OL]. (2016-11-XX)[2022-03-17]. <https://www.ti.com.cn/cn/lit/an/sprac57/sprac57.pdf>.
- [11] 吴翼虎,钱宏文,朱江伟.嵌入式异构平台DDS中间件设计[J].电子与封装,2021,21(8):56-61.
- [12] Xilinx. MicroBlaze processor quick start guide [EB/OL]. (2018-01-03).[2022-03-17]. [https://china.xilinx.com/content/dam/xilinx/support/documentation/quick\\_start/microblaze-quick-start-guide.pdf](https://china.xilinx.com/content/dam/xilinx/support/documentation/quick_start/microblaze-quick-start-guide.pdf).
- [13] 薛慧敏,武传华,路后兵,等.基于MicroBlaze的PetaLinux嵌入式操作系统移植[J].微计算机信息,2011,27(8):108-110.
- [14] RTI. RTI connext DDS core libraries user's manual version 6.1.0[EB/OL]. (2020-XX-XX) [2022-03-17]. [https://community.rti.com/static/documentation/connext-dds/6.1.0/doc/manuals/connext-dds\\_professional/users\\_manual/index.htm](https://community.rti.com/static/documentation/connext-dds/6.1.0/doc/manuals/connext-dds_professional/users_manual/index.htm).
- [15] RTI. RTI PerfTest user's manual[EB/OL]. [2022-03-17]. <https://community.rti.com/static/documentation/perftest/current/introduction.html>.
- [16] OpenDDS. OpenDDS developer's guide. Version 3.14 [EB/OL]. (2021-01-27)[2022-03-17]. <http://download.objectcomputing.com/OpenDDS/previous-releases/OpenDDS-3.16.pdf>.

(收稿日期:2022-03-17)

### 作者简介:

钱宏文(1975-),男,本科,研究员级高级工程师,主要研究方向:集成电路应用、微系统。

朱江伟(1989-),通信作者,男,本科,主要研究方向:嵌入式计算、分布式网络中间件, E-mail: zhujiw1125@163.com。

吴翼虎(1986-),男,硕士,高级工程师,主要研究方向:嵌入式计算、微系统。



扫码下载电子文档

## 版权声明

经作者授权，本论文版权和信息网络传播权归属于《电子技术应用》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、DOAJ、美国《乌利希期刊指南》、JST日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《电子技术应用》编辑部

中国电子信息产业集团有限公司第六研究所