

RTDX 在弹载 DSP 软件代码调试中的应用^①

张 飞¹, 敖志伟¹, 陈 勇¹, 张 航²

¹(中国空空导弹研究院 第六研究所, 洛阳 471009)

²(黎明化工研究院 分析与测试中心, 洛阳 471001)

摘 要: 本文在分析现有实时嵌入式软件调试技术的基础上, 对利用 RTDX 技术开展红外导引头软件代码调试的可行性进行了分析, 并以 TMS320F2812 为硬件平台将 DSP 程序的调试数据传输至主机. 通过对数据传输的正确性和实时性进行分析, 认为 RTDX 能够用于弹载软件代码调试, 论文的研究成果为后续基于 RTDX 开展嵌入式系统软件故障注入和软件测试验证积累了经验.

关键词: 弹载数字信号处理器; 数据交换; 代码调试; 实时

Applying RTDX in Missile-Borne DSP Software Debug

ZHANG Fei¹, AO Zhi-Wei¹, CHEN Yong¹, ZHANG Hang²

¹(China Airborne Missile Academy, Luoyang 471009, China)

²(Liming Research Institute of Chemical Industry, Luoyang 471000, China)

Abstract: Based on the study of existing technique, this paper analyses the feasibility of using RTDX in Embedded Software debugging. In order to validate it, experiment of using RTDX to transmit debug information from TMS320F2812 DSP to host is carried out. Experiment results prove that the RTDX can correctly transmit debug information from DSP to host in real-time, also provide basis for further researching on fault injection and verification.

Key words: missile-borne DSP; RTDX; debug; real-time

数字信号处理器(Digital Signal Processing, DSP)是对信号和图像实现实时处理的一类高性能处理器^[1]. 在以 DSP 为硬件平台的嵌入式软件开发过程中, 代码调试必不可少, 断点调试是最常用的技术之一. 该技术采用在程序中设置断点暂停软件运行的方式来观测变量、寄存器等信息, 以此为基础对程序的执行过程、数据处理进行分析. 然而对实时系统来说, 嵌入式软件与外部设备之间的数据交互是实时的, 要求软件的执行过程不被暂停. 因此, 设置断点的方式在实时嵌入式软件的代码调试中有局限性.

RTDX(Real-Time Data Exchange, 实时数据交换)是 TI 公司用于实现 DSP 与主机之间实时数据交换的解决方案. RTDX 提供了实时、连续了解 DSP 程序运行情况的手段, 允许在不干扰系统程序运行的情况下, 实现主机和 DSP 之间的实时数据传递^[2]. RTDX 技术在

诸多嵌入式系统中得到了应用, 如: 运动数字铭牌识别^[3]、实时语音分析^[4]、图像处理^[5]、仿真验证^[6]等.

在制导武器中, 弹载软件与探测器等硬件设备通过数据交互构成闭环的反馈系统实现目标识别、跟踪和抗干扰等功能, 弹载软件的正常工作对实时性具有较高的要求. 由于 RTDX 可以在不停止应用程序执行的情况下实现 DSP 与主机之间的数据传递, 因此可以利用 RTDX 从 DSP 向主机发送代码调试所需的内存和寄存器等数据, 在主机端通过分析数据来掌握程序的运行情况, 这样可以有效地避免因设置断点而暂停程序运行的缺陷, 为导引头软件代码调试提供了新的技术途径.

本文以 TMS320C2812 为硬件平台, 通过实验验证了 RTDX 技术应用于弹载软件代码调试的可行性, 并在此基础上测试了 RTDX 在 TMS320C2812 上的数

^① 收稿时间:2013-07-10;收到修改稿时间:2013-08-19

据传输速率,为其在实时嵌入式软件中更广泛地应用提供了依据。

1 RTDX原理

RTDX 技术的原理如图 1 所述。

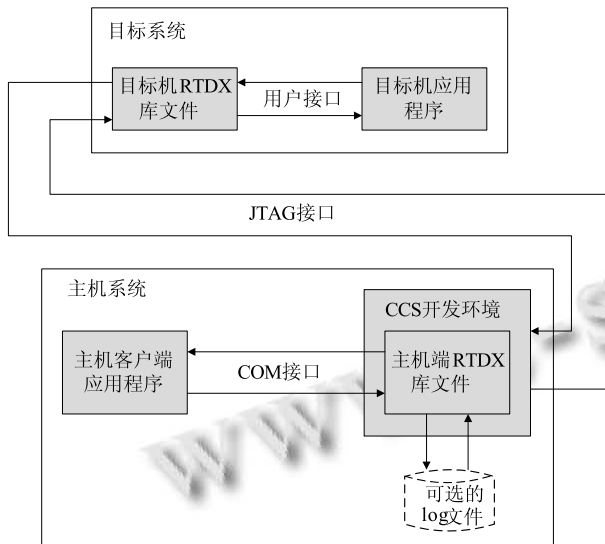


图 1 RTDX 原理图

目标系统上的软件由目标机应用程序和目标机

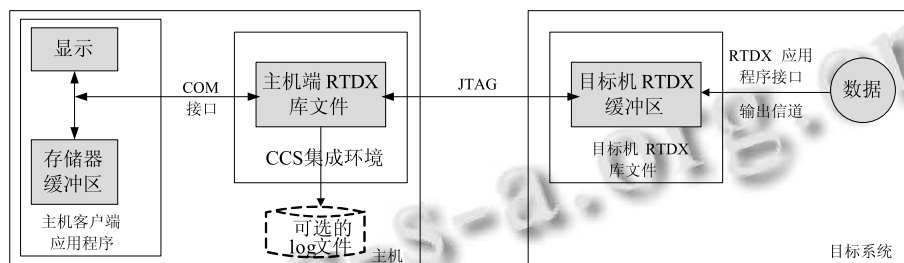


图 2 目标系统发送数据到主机^[1]

当目标系统从主机接收数据的时候(如图 3 所示),在目标机应用程序中要创建输入信道.目标机应用程序通过 RTDX 程序接口从输入信道请求数据,目标机 RTDX 缓冲区内记录多个信道的数据请求,RTDX 库文件中的请求查询函数通过 JTAG 接口查询主机端 RTDX 缓冲区中的数据是否准备好.当主机应用程序通过 COM 接口将数据发送到主机 RTDX 缓冲区之后,主机端 RTDX 数据传输函数根据接收到的请求将相应信道的数据直接发送到目标机应用程序的缓冲区中^[1].

目标机与主机的数据传递过程并未暂停目标机应

用程序的执行流程,RTDX 的这个特点使得从目标机应用程序接收到的数据真实地反映了程序的控制流和数据处理情况,更有利于对软件的调试和分析。

RTDX 库文件组成.目标机应用程序是用户编写的 DSP 软件,它调用目标机 RTDX 库文件所提供的接口函数,完成数据的发送和接收操作。

主机系统上的应用程序由主机客户端应用程序、CCS 开发环境及可选的 log 文件组成.客户端应用程序是运行在主机上的应用程序,该程序利用 COM 接口调用主机端 RTDX 库文件实现从目标系统接收数据或向目标系统发送数据,并以自己需要的形式展示和解释数据^[1].目前,Visual C++、Visual Basic、Excel、Matlab 等应用程序都支持 COM 接口调用^[7,8].

当目标系统发送数据给主机的时候(如图 2 所示),在目标机应用程序中要创建输出信道.通过 RTDX 用户接口中定义的接口函数将数据写入输出信道,此数据立即被记录到 RTDX 库文件所定义的缓冲区中,之后由库文件提供的数据传输函数将缓冲区中的数据通过 JTAG 接口发送到主机.主机端 RTDX 库文件接收到这个数据后,存储在数据缓冲区或 log 文件中^[1].主机客户端应用程序通过 COM 接口访问指定输出信道的数据,并以某种有意义的方式显示出来供调试分析。

用程序的执行流程,RTDX 的这个特点使得从目标机应用程序接收到的数据真实地反映了程序的控制流和数据处理情况,更有利于对软件的调试和分析。

2 RTDX在代码调试中的应用

本节以 TMS320C2812 的 DSP 为硬件平台,在 DSP 应用程序中利用 RTDX 技术将程序运行时的变量、寄存器值实时地传递给主机.在主机端利用 CCS3.1 和 RTDX 数据接收终端接收 DSP 程序实时发送过来的调试数据并加以分析。

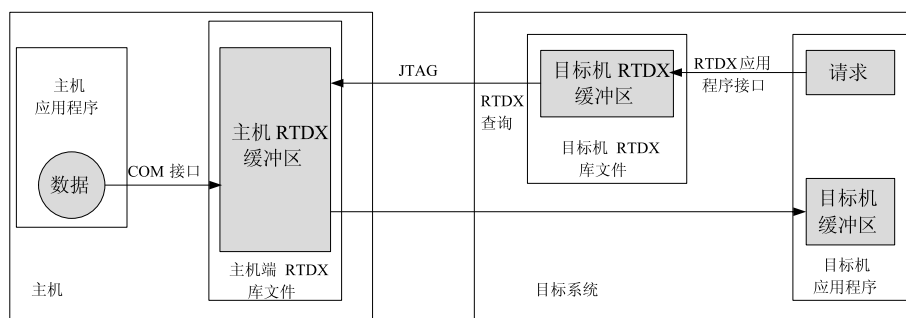


图 3 目标系统从主机接收数据^[1]

2.1 目标机程序设置

目标机程序的设置主要包括以下内容:

① 添加目标机 RTDX 库文件、头文件

目标机 RTDX 库文件中封装了应用程序可使用的创建信道、读数据、写数据、查询信道状态等接口函数. 在 CCS 中根据目标 DSP 的型号将相应的库文件 rtdx.lib 添加到应用程序的项目工程之中.

此外, 还须添加以下目标机 RTDX 头文件:

rtdx.h: 包含 RTDX 库文件中接口函数的声明以及全局变量的定义;

RTDX_access.h: 包含对 RTDX_FAR_MODE 宏的定义;

rtdxpoll.h: 包含对数据传输方式的定义;

target.h: 包含硬件初始化、禁用看门狗的汇编代码.

② 定义目标机 RTDX 数据传输方式

目标机 RTDX 数据传输分为两种方式: 主动传输 (poll-driven) 和中断传输 (interrupt-driven).

主动传输方式是指目标机应用程序要显式地调用数据传输函数 RTDX_Poll(). 如果不调用该函数, 那么待传输的数据将会永远停留在缓冲区内, 不会被传输. 为了提高传输效率, 可以在待传输的数据准备好时就调用 RTDX_Poll() 函数. 在数据准备好的前提下, 调用 RTDX_Poll() 的频率越高, 传输效率越高.

中断传输方式是指目标机不必显式地调用 RTDX_Poll(), 只须将待传输的数据放入缓冲区即可. 真实的传输过程由事件中断 (DLOGIV 或 MSGINT) 的中断服务子程序完成. 当主机端准备好接收数据时, 会通过 JTAG 触发 DSP 的事件中断, 在中断服务子程序完成真实的数据传输.

两种方式相比, 主动方式一旦有数据准备好就传送, 而中断方式是有数据传递需求才传送. 因此, 中

断方式更适用于对实时性要求较高的应用程序.

C28x 系列 DSP 采用中断传输方式, 因此需将 RTDX_POLLING_IMPLEMENTATION 宏置 0.

③ 设置中断屏蔽标记

仅当采用中断传输方式时, 才需设置中断屏蔽标记. 通过设置中断屏蔽标记用来在 RTDX 数据传输的过程中禁止或启用其他中断源.

④ 配置 CMD 文件

在内存分配文件中, 要为 rtdx 代码段 (rtdx_text) 和 rtdx 数据段 (rtdx_data) 分配存储空间. DSP 上 rtdx 的缓冲区大小与 rtdx 数据空间的设置有关.

2.2 发送调试数据

目标机利用 RTDX 发送数据时, 需要按顺序在 DSP 应用程序中添加如下代码:

① 定义输出信道

RTDX_CreateOutputChannel(ochan);

② 初始化目标库

TARGET_INITIALIZE();

③ 使能输出信道

RTDX_enableOutput(&ochan);

④ 收集调试数据, 并将待发送的调试数据放入缓冲区中

RTDX_write(&ochan,&data,sizeof(data));

⑤ 关闭输出信道

RTDX_disableOutput(&ochan);

2.3 主机端接收调试数据

在主机端, 利用数据接收程序将接收到的调试数据展示出来以供分析, 如图 4 所示.

通过分析对比, 接收到的数据如实地反映了 DSP 程序运行期间所发送的调试信息, 验证了利用 RTDX 技术实现代码调试分析的可行性.

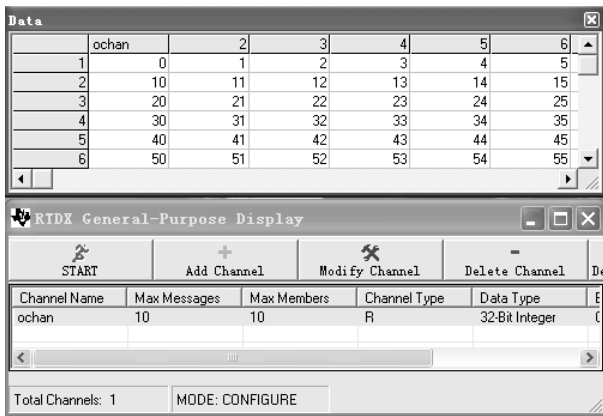


图 4 主机端数据接收程序

3 传输速率测试

本节针对利用 RTDX 发送 DSP 程序调试数据的应用场景, 对 RTDX 数据传输速率和库函数的时间开销进行了测试. 测试环境如表 1 所述.

表 1 测试环境

目标系统	DSP 型号	TMS320F2812
	CPU 频率	150MHz
	RTDX 数据传输方式	中断传输
主机系统	集成开发环境	CCS3.1
	数据接收程序	Gpdprog
	仿真器	Blackhawk USB560m
	示波器	Tektronic TDS2024

3.1 测试方法

本文利用了 TMS320F2812 的 GPIO 功能辅助完成测试的整个过程.

在 DSP 应用程序中, 通过 F2812 的 GPIO 的相关寄存器将指定的 IO 管脚配置为输出管脚. 在程序初始化时将该管脚输出置 0, 在待测函数运行之前将测试管脚输出置 1, 在待测函数运行完之后将测试管脚输出置 0. 程序运行期间, 利用示波器观测测试管脚的输出波形, 并记录波形高电平的方波时间, 该时间即为待测函数的运行时间.

3.2 测试结果

测试程序中, 分别将调试所关心的 16 位整型数据和 32 位浮点型数据通过 JTAG 发送给了主机, 发送的数据放置在数组中, 分别测试了长度为 8、16、32、64、128、256 的数组的传输时间(如图 5 所示)和传输速率(如图 6 所示).

由于浮点数是 32 位, 整型是 16 位, 因此传输同样长

度的数组, 浮点数的传输时间要比整型的传输时间长;

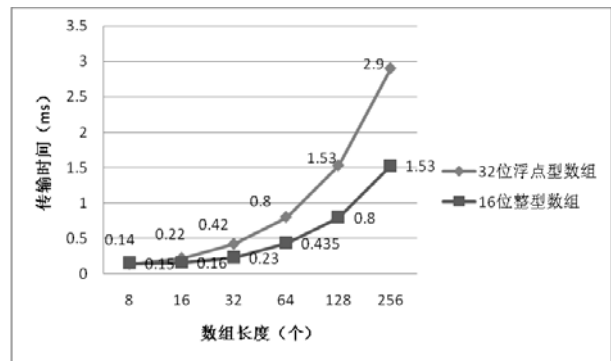


图 5 RTDX 传输时间(ms)

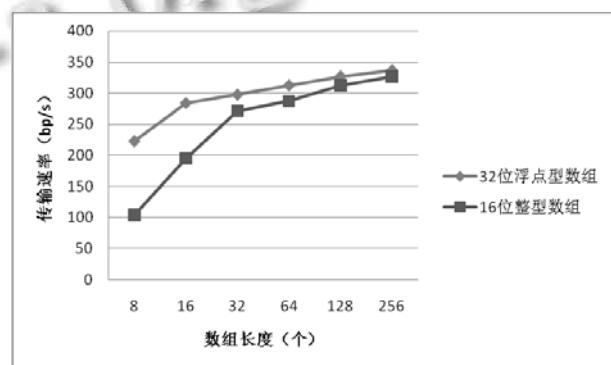


图 6 RTDX 传输速率(kb/s)

由于长度为 n 的浮点型数组与长度为 $2n$ 的整型数组所包含的位数相同, 因此其传输时间基本相同, 这也从图 5 的测试结果中得到了验证.

从图 5 的测试结果中可以看出, 数据传输时间的量级为毫秒级(ms), 从 DSP 传输 128 个整型到主机的时间为 0.8ms. 从已有的数据看, 利用 RTDX 技术进行代码调试所花费的时间开销不会影响弹载信息处理软件的实时性.

从图 6 可以看出, 数据传输速率随着传输的数据量的增大而增大. 这是因为利用 RTDX 从 DSP 传输数据至主机, RTDX 本身所带来的时间开销是一定的, 并不随着数据传输量的增加而增加. 这也与 RTDX 技术手册“通过增加传输的数据量来提高传输速率”的建议相符合.

通过对上述测试结果的分析, 本文认为利用 RTDX 实现 DSP 与主机之间的数据传输有以下优点:

- ① 与设置断点的方式相比, 该技术能够在不中断 DSP 软件运行的前提下实现调试数据的实时传递;
- ② 具有实时性, 能够满足弹载 DSP 软件对仿真

调试技术的实时性要求;

③ 简单实用, 主机端能够在 Matlab、Excel 中使用 COM 接口对接受到的 DSP 数据进行分析处理。

4 结语

本文针对设置断点不能有效地对实时嵌入式系统软件进行代码调试的问题, 从分析 RTDX 技术的原理入手, 提出了利用 RTDX 开展了弹载软件代码调试的技术途径。实验结果表明, 利用 RTDX 技术能够正确地将调试所需要的数据从 DSP 传输至主机进行分析, 其传输时间和传输速率也能够满足当前导引头信息处理软件的实时性要求, 为开展实时嵌入式软件代码调试提供了新的技术途径。

此外, RTDX 技术支持主机客户端发送数据至 DSP 软件, 这为下一阶段利用 RTDX 开展弹载 DSP 软件故障注入、建立闭环的弹载 DSP 仿真验证环境等工作提供了研究方向。

参考文献

1 Texas Instruments Incorporated. 彭启琮, 张诗雅, 常冉等译。

(上接第 182 页)

- 4 Horn BKP, Schunck BG. Determining optical flow. *Artificial Intelligence*, 1981, 17(1): 185–203.
- 5 Swain M, Ballard D. Color indexing. *International Journal of Computer Vision*, 1991, 7(1): 11–32.
- 6 杨康叶, 邬春学. 基于 RGB 模型颜色相似性的彩色图像分割. *计算机系统应用*, 2013, 22(3): 128–131.
- 7 Lee J, Lee W, Jeong W. Object tracking method using back-projection of multiple color histogram models. *Proc. of ISCAS' 03*, 2003, 2(2): 668–671.
- 8 Adelson EH, Anderson CH, Bergen JR, et al. Pyramid methods in image processing. *RCA engineer*, 1984,

TI DSP 集成化开发环境使用手册. 北京: 清华大学出版, 2005: 184–185.

- 2 彭启琮, 管庆等. DSP 集成开发环境. 北京: 电子工业出版社, 2004: 163–164.
- 3 刘红刚, 尹岗. RTDX 在运动数字铭牌识别中的应用. *自动化应用*, 2011, (2): 3–4.
- 4 顾长怡, 黄佩伟. 利用 RTDX 技术实现计算机实时语音分析. *计算机工程*, 2001, (2): 155–156.
- 5 胡敏露, 徐刚锋, 李飏. RTDX 在图像处理中的应用. *国外电子元器件*, 2003, (5): 21–24.
- 6 倪骏, 徐向东. 运用 RTDX 技术的 TMS320C6201 的软件验证方法. *空军雷达学院学报*, 2002, (4): 40–42.
- 7 许海丽, 张茂青, 严震宇, 沈芳. 基于 MATLAB 的 DSP2812 RTDX 实时绘制电机转速曲线. *苏州大学学报(工科版)*, 2013, (5): 47–50.
- 8 张传清, 安钢, 樊新海. RTDX 与 Matlab 实现基于 DSP 的 FIR 滤波器设计. *装甲兵工程学院学报*, 2001, (3): 64–69.

29(6): 33–41.

- 9 熊哲源, 樊晓平, 黎燕等. 基于数学形态学边缘检测的车牌字符分割算法. *计算机系统应用*, 2010, 19(9): 155–158.
- 10 毛若羽, 陈相宁. 改进的 HSI 空间形态学有噪彩色图像边缘检测. *计算机应用研究*, 2013, 30(2): 635–637.
- 11 Gonzalez RC, Woods RE. *Digital image processing (International ed.)*. 2008.
- 12 Bradski GR, Kaehler A. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Incorporated, 2008: 115–118.