

基于 Eclipse 嵌入式系统调试软件的设计^①

王 婷¹, 芮国俊²

¹(中国电子科技集团公司第十四研究所, 南京 210039)

²(南京国电南自电网自动化有限公司, 南京 211153)

摘 要: 基于 Eclipse 平台的嵌入式软件开发调试工具对于嵌入式系统软件开发具有重要意义, 但目前基于 Eclipse 平台的嵌入式开发调试工具实现不多. 针对这种情况, 提出了使用 Eclipse 平台及其插件技术进行嵌入式调试工具软件的开发, 实现了支持嵌入式多核系统的调试工具软件 MDSPTool, 能够与嵌入式开发平台 WorkBench 完全无缝集成. 实验结果表明, MDSPTool 调试工具提供了优良的用户界面, 便捷的并行调试方法, 高度集成的系统监测功能, 从而提高了嵌入式软件开发效率.

关键词: 嵌入式; 集成开发平台; Eclipse; 插件开发; 调试工具; 图形化

Design of Embedded System Debugging Tool Based on Eclipse

WANG Ting¹, RUI Guo-Jun²

¹(Nanjing Research Institute of Electronics Technology, Nanjing 210039, China)

²(Nanjing SAC Automation CO. Ltd, Nanjing 211153, China)

Abstract: Embedded software development and debugging tools based on Eclipse platform has important significance for embedded software development, but at present, the number of debugging tool based on Eclipse platform is few. In view of this, this paper provides a new implementation of debugging tool named MDSPTool which supports embedded multi-core system. It is based on Eclipse and plug-in development environment, and can seamlessly integrate with WorkBench IDE. Experimental results show that MDSPTool, which has an excellent user interface, provides easily used methods for parallel debugging and highly integrated functions for monitoring and measurement. Therefore, MDSPTool improves the efficiency of embedded software development.

Key words: embedded system; integrated development platform; eclipse; plug-in development; debugging tool; graphical display

嵌入式控制系统作为现代计算机技术的重要分支之一, 在人们的日常生活中越来越得到广泛的应用, 成为继 PC 机之后信息处理的另一大主要工具. 伴随着嵌入式系统在工业、军事、交通、通信等行业的广泛运用, 嵌入式开发平台正在逐步发展^[1]. 嵌入式多核系统迫切需要支持实时多任务及并发操作的软件开发调试环境. 美国 WindRiver 公司的 WorkBench3.0 采用了 Eclipse 软件框架结构, 是目前最新的嵌入式开发平台, 它的开放性和易扩展性极大降低了多核系统开

发的复杂度. Eclipse 软件框架在嵌入式开发平台方面的研究目前还不多见, 是一次新技术的运用和新方法的实践. 由于 Eclipse 针对嵌入式系统的开发工具不够丰富^[2], 基于硬件的调试方法不够灵活, 在实际应用中还存在着不足之处, 因此嵌入式软件开发调试工具一直是工具软件研究的热点^[3]. 本文介绍分析了 Eclipse 的框架, 讨论如何使用 Eclipse 和插件技术进行嵌入式平台调试软件的开发, 实现了嵌入式多核系统调试工具软件 MDSPTool.

① 收稿时间:2015-03-13;收到修改稿时间:2015-05-07

1 Eclipse平台体系结构和插件原理

1.1 Eclipse 平台体系结构

Eclipse 是著名的跨平台的自由集成开发环境,本身是一个框架平台和一组服务^[4,5],通过插件组件构建开发环境,还提供了一个用于开发插件的框架,使创建、集成和使用软件工具更容易。

Eclipse 的设计思想是:一切皆插件. Eclipse 核心很小,其它所有功能都以插件的形式附加于 Eclipse 核心之上^[6,7]. 众多插件的支持使得 Eclipse 拥有其他功能相对固定的 IDE 软件很难具有的灵活性和移植性. Eclipse 本身由多个子系统构成,都是建立在平台运行库之上. 它包含如下几个核心部分:平台运行库、工作区、工作台、图形 API (SWT/Jface), Java 开发环境插件(JDT), 插件开发环境(PDE)、帮助(Help)、调试(Debug). 如图 1 所示.

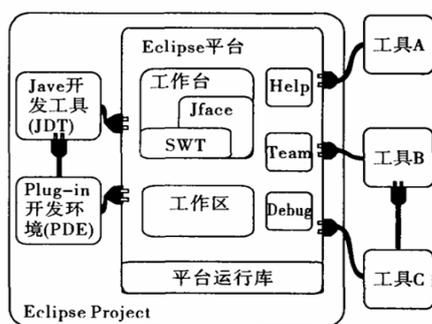


图 1 Eclipse 体系结构

1.2 插件原理

Eclipse 包含了一个插件开发环境(PDE), PDE 可以开发与 Eclipse 完全无缝集成的插件, 扩展 Eclipse 平台功能^[4]. 插件是最小的软件功能单元, 可独立开发和发布. 简单的功能可以通过一个插件实现, 而比较复杂的功能一般由若干插件共同完成. Eclipse 平台的所有功能都集中在插件中, 都是以同等的方式创建的^[6].

一个典型的插件由 JAR 类库、插件资源文件(如图像、Web 模板、消息日志)和插件清单文件等组成. 插件扩展以及它们之间的依赖关系便是通过名为 plugin.xml 的清单文件(Manifest)来完成. Manifest 是 XML 格式的文件, 含有若干扩展点. Eclipse 启动时, 平台运行库读取插件声明文件, 建立插件注册表, 查找可以使用的插件, 但是此时插件并没有加载, 只有在实际使用时才被加载, 大大降低了启动时间. 插件被加载后, 根据注册信息可以明确哪些插件要扩展它.

插件可以声明一个扩展点, 其它插件只要对这个扩展点声明扩展, 就可以利用这个插件.

1.3 开发插件的方法

插件的开发一般是基于 Eclipse 核心插件来实现^[8], Workbench 和 Workspace 是 Eclipse 平台的两个必备核心插件. Workbench 是 Eclipse 用户界面中最底层的对象, 提供了一些扩展点, 使用户界面带有菜单选择和工具栏按钮^[9]. Workspace 插件提供了可以与用户资源(包括项目和文件)交互的扩展点. 插件需要扩展点才能插入和运行, 而它们两个提供了大多数插件需要使用的扩展点.

本文的多核系统调试软件的客户端就是基于 Workbench 核心插件扩展的. 最简单的创建插件的方法是使用 PDE 开发环境. PDE 提供了一组向导以帮助创建插件, 但创建的插件模板还没有任何实际功能, 需要软件开发人员进一步设计开发.

2 调试软件MDSPTool的设计实现

传统的嵌入式开发和调试工具集成度差, 功能不够完善, 往往是针对不同开发阶段而设计应用的. 通过分析发现一些嵌入式开发系统的工具集集成度不高, 缺乏良好的可视化图形界面, 开发人员仍然需要手工操作不同工具进行开发^[10]. 另一方面受硬件条件的限制, 缺少必要的模拟调试手段, 缺乏足够的灵活性.

Eclipse 的嵌入式开发工具集仅提供了对嵌入式开发的项目管理工具, 远程调试等工具, 还不具备性能分析等实用工具. 但是 Eclipse 具有的可扩展性和可移植性使其很适合成为一个嵌入式开发平台, 并利用这些优点实现高度集成的嵌入式调试工具集.

本文设计的 MDSPTool 为嵌入式多核芯片的应用开发提供完整的开发调试套件, 完全基于 Eclipse 框架, 使用 JDT/CDT 开发工具和 PDE 开发环境开发, 可以无缝集成 WorkBench 平台, 支持 VxWorks^[11]嵌入式系统. 该软件主要完成嵌入式多核系统的软件调试和性能调控, 能够获取多核模块的软硬件信息进行图形显示, 监控模块实时工作状态, 下发调试命令, 增强软件调试功能, 加快系统的研制进度, 降低多核设备开发的跨度和复杂度, 为多核系统提供一个实时多任务、支持并发操作的软件调试环境.

2.1 MDSPTool 的设计思想

嵌入式系统软件开发都是采用交叉开发环境, 因

此 MDSPTool 的实现包括目标机端(Target)的服务器软件 MDSPServer 和宿主机端(Host)的客户端插件 MDSPClient 两个部件,两者之间使用以太网通讯协议进行通讯,从而实现多核系统的并行调试交互控制。

客户端插件 MDSPClient 为调试前端,基于 Eclipse 的集成开发环境和插件技术开发,使用 JDT 开发工具,通过以太网和目标机服务器程序通讯,提供系统硬件查询配置、应用程序的开发和调试及一套标准的数据传输函数库和数据处理函数库等,方便用户配置系统资源及并行应用程序的开发和调试。MDSPClient 通过网络可以获取目标机上传的硬件信息,事件信息,任务运行状态、系统自检信息等等,并且下发调试控制命令。

服务器软件 MDSPServer 使用 WorkBench3.0 集成开发平台开发,编程语言是 C 语言和 Mips 汇编语言,软件运行环境是 VxWorks6.7 以上操作系统,通过以太网和宿主机的 MDSPClient 通讯。MDSPServer 作为后台程序加载运行于每个目标机端,程序启动后先和 MDSPClient 软件完成握手通讯,然后监测本地节点工作状态,创建多个代理对象(监测上报 Agent、并行调试 Agent、任务下载 Agent 等)实时侦听并且响应宿主机端的查询、调试控制命令。Agent 代理对象通过调用设备驱动程序中的硬件访问和控制接口完成对多核共享设备的读写访问和配置功能,从而获取多核共享设备的实时资源信息,响应客户端的调试控制命令,上报查询结果。图 2 为 MDSPTool 的软件体系结构设计图。

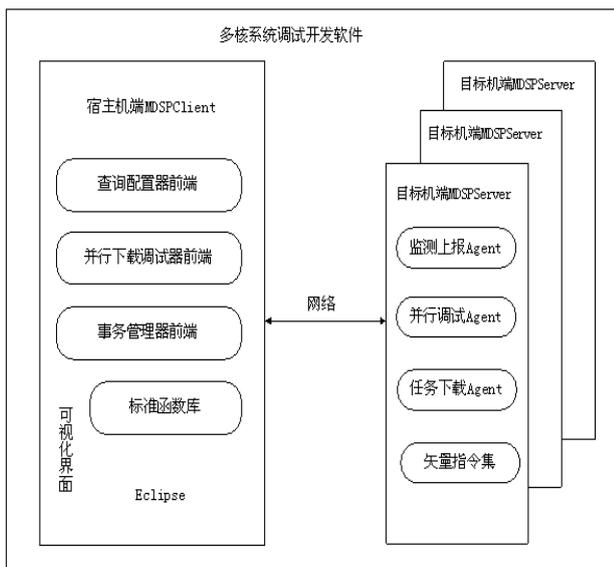


图 2 MDSPTool 软件体系结构设计图

2.2 MDSPTool 的功能设计

2.2.1 查询配置器功能

查询配置器用于查询多处理器系统的硬件资源状况并具备配置能力,可以树表和图形的方式显示系统的组成、拓扑结构及各主要部件的详细信息。建立资源动态监控机制,实时监控各节点的资源状态及各模块的负载状况,实现对服务瓶颈节点/模块的在线分析。系统上电后,开发机端向系统内各处理节点发出广播包,查询已连入系统并正常工作的目标节点,在后台接收目标机定期发来的硬件状态信息(包括工作状态和健康信息等),经过分类处理后,交由前台的人机界面程序进行图形化表述。该工具可视化显示平台的硬件信息,包括各个组件的性能参数和当前工作的实时参数,如 CPU 速度、flash 大小、当前 IP 地址、内存总线速度等等。该模块还具有侦查预警功能,当某一组件的实时工作参数超出可接受范围(如:温度、CPU 占用率、内存使用率),则以警戒色预报显示出来,并写入日志文件中。该模块还可以用户自定义预警参数以及参数范围。查询器软件程序流程图如图 3 所示。

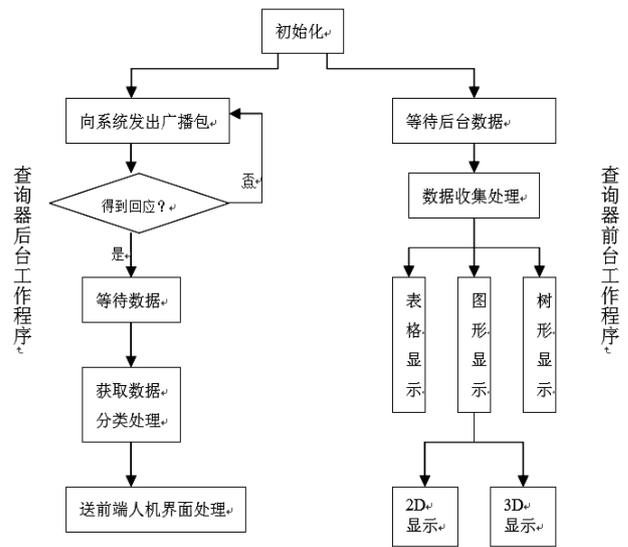


图 3 查询配置程序流程图

2.2.2 多任务远程并行下载调试模块

在本软件中,并行下载调试功能既包含传统的命令行调试方法,也开创性地实现嵌入式图形软件调试功能,最多支持 200 个节点的应用程序并行下载及运行,提供可视化界面,并能保存和加载配置信息,可以一键连接、加载、卸载、重启目标机程序。

命令行调试功能,不仅可以单目标机调试,而且可以对同一类型的目标机进行分组并行调试,有效节省了软件调试时间,提高了调试效率.每个命令行窗口通过 Telnet 连接到目标机服务器,回显目标机任务运行状态和运行参数.本软件的 Telnet 客户端功能采用第三方工具包 commons-net-3.2 来实现, MDSPClient 通过调用其 telnet client 接口函数,实现 telnet 客户端远程访问功能.目标机端的 VxWorks 操作系统包含了 telnet 服务器组件,所以可以实时响应客户端的 telnet 请求.

图形软件调试改变了传统的 Target Shell 调试方法,不需要每次重新输入调试命令和参数,配置好一次软件调试信息(如:程序入口名称、参数列表、下载节点信息等等),将命令通过网络发送至各计算节点,各计算节点接收到命令后即通过网络文件系统下载该目标代码,在目标机上建立程序的远程映像,大大缩短了软件调试周期,提高工作效率.

2.2.3 多目标机事务管理配置模块

对多目标机进行事务配置管理,为软件人员提供事件记录和分析工具,开发者可以分析软件和算法在事件驱动下的执行情况,追踪多任务多节点的事件信息,如:处理器负载情况,算法运行时间、任务上下文切换、内存占用情况.该模块支持多处理器平台环境,记录的事件作为一个特殊的会话文件保存下来,并且能够以图形化方式显示信息,可以用于事后分析,从而帮助用户改善优化算法的表现.

2.2.4 基于 SWT/JFace 的人机图形界面

本软件的客户端 MDSPClient 采用 Eclipse 平台中的 SWT/JFace^[12]图形库来绘制人机界面,并且集成了 JOGL^[13](Java Bindings for the OpenGL)图形库用于进行 2D/3D 的模型显示. SWT/JFace 是 Eclipse 的 GUI 库,为开发人员提供了独立于操作系统的 API,基于它开发出来的应用在外观上可以仿造本地操作系统的风格,运行速度非常迅速. SWT 提供了基本的图形库和小部件集,而 JFace 是基于 SWT 开发的,其 API 比 SWT 更加易于使用,添加了更多的应用服务,可用于实现复杂、高级的 GUI 交互界面.

3 应用实例

MDSPTool 软件的实验平台是由多个高性能信号处理模块组成的信号处理系统,硬件环境为四个 G4 平台信号处理模块,其运行的操作系统为 VxWorks

6.7 SMP, 宿主机调试端是 Windows XP 系统,和目标机端通过以太网通讯.

信号处理系统上电启动, MDSPServer 作为后台程序运行于每个目标机端,等待宿主机端的调试控制命令.

宿主机端使用 eclipse 自带的 Export 功能,将客户端软件 MDSPClient 导出成插件 Plugin 的形式,将该插件直接拷贝到支持 Eclipse 软件框架的开发平台(如 eclipse-java-juno、WorkBench3.0)的 plugins(或者是 tool)目录下,启动平台软件时,该插件就可以自动加载到该平台的菜单和工具栏里,如图 4 所示.



图 4 插件启动界面

进入调试之前宿主机通过网络连接目标机端,此时多目标机已启动 MDSPServer 服务程序.点击运行该插件工具,发出广播包,当前活动目标机节点就能通过网络上报显示在界面左侧的表格中,选择需要调试的节点,打开命令行 Telnet 调试界面就可以并行调试,如图 5、图 6 所示.

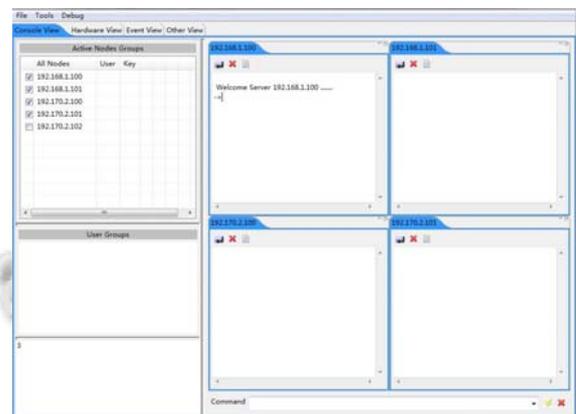


图 5 telnet 远程调试界面 1

tJobTask	1df5f0	52a190	0	PEND
tExcTask	1de744	2ef648	0	PEND
tLogTask	logTask	52d720	0	PEND
tNbioLog	1e0c44	531120	0	PEND
tShell0	shellTask	d46140	1	PEND
tShellRem1	shellTask	d5cb00	1	READY
tWdbTask	wdbTask	7fe2a0	3	PEND
tErfTask	165094	534a90	10	PEND
tNetTask	netTask	53ac60	50	PEND
tTelnetd	telnetd	7f5fc0	55	PEND
tRlogind	rlogind	7f97d0	55	PEND
tTelnetOut>	telnetOutTas	1ffe110	55	READY
tTelnetIn_>	telnetInTask	d4b920	55	PEND
mpiInt	1290dc	d484b0	60	PEND
AXISCacheM>	cache_period	d6f8d0	65	DELAY
AXISserver	db81fc	e1ff70	250	PEND
AXISclient>	AS_AXISViewC	d66b00	251	PEND+T
value = 0 = 0x0				
->				

图 6 telnet 远程调试界面 2

选取多个目标机节点, MDSPClient 和目标机 MDSPServer 程序之间通过网络进行控制命令收发和消息通讯, 可以实现目标代码下载、启动、卸载多任务并行调试, 如图 7 所示。

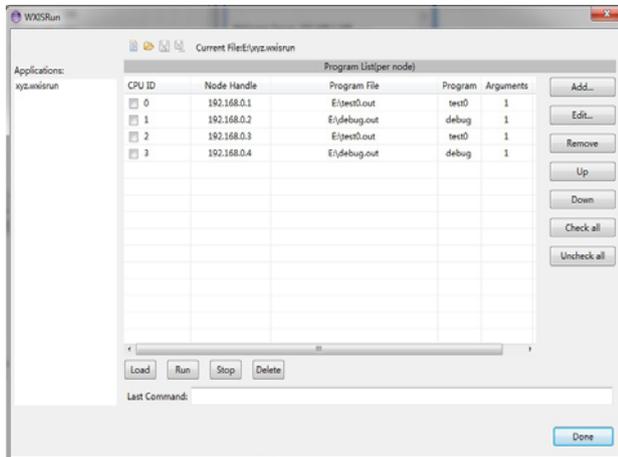


图 7 并行下载远程调试界面

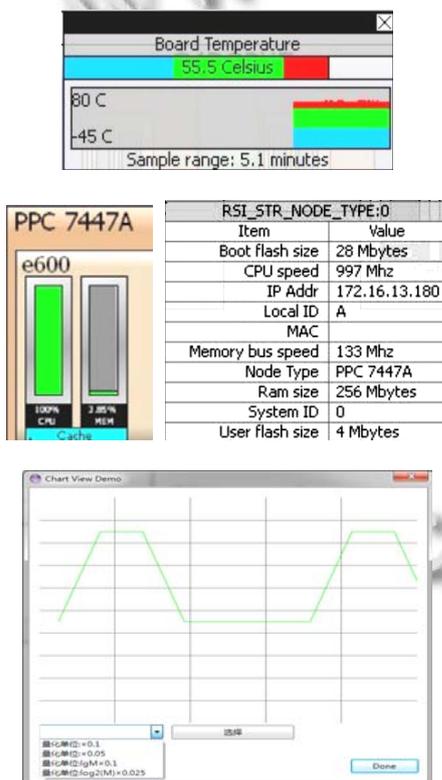


图 8 资源监测界面

如图 8 所示, 可以查看某个目标节点的资源监测状态和工作实时参数(如: 温度、CPU 占用率、内存使用率、CPU 主频、IP 地址、内存总线速度等), 可视化显示性能参数, 也能够查看某工作状态下信号量的实

时变化, 从而帮助用户改善程序设计, 优化信号处理算法。

实践证明了本调试软件支持嵌入式多核系统实时多任务及并发操作调试, 实现多核系统的资源监测功能, 极大方便了嵌入式多核系统的开发调试。

4 结语

本文介绍了嵌入式多核系统调试工具的设计思想和实现技术, 并在多核信号处理系统平台环境下得到实现。嵌入式软件开发非常需要功能强大且完善的集成开发调试环境和开发工具。本文的 MDSPTool 软件基于 Eclipse 的集成优越特性, 提供了优良的用户界面, 便捷的调试方法, 高度集成系统分析和资源监测功能, 从很大程度上提高了软件开发的效率, 缩短了开发周期。

参考文献

- 林建民. 嵌入式操作系统技术发展趋势. 计算机工程, 2001, 27(10):1-4.
- 竹居智久. Eclipse 将成为嵌入式软件主流开发环境. 电子设计应用, 2005, 9.
- 段富刚, 施展. 基于 Openocd 的嵌入式软件开发平台的研究和设计. 计算机测量与控制, 2010(2):470-472.
- Gamma E, Beck K. Contributing to Eclipse: Principles, Patterns, and Plug-ins. Addison Wesley, 2003.
- Eclipse Consortium. Eclipse Modeling Framework (Version 2.0). <http://www.eclipse.org/emf>. 2004-10-20.
- Beaton W, Rivieres J. Eclipse platform technical overview. Retrieved on November, 2006, 2: 2009.
- 冯新扬, 崔凯, 沈建京. 面向插件的应用框架研究与实现. 计算机工程与应用, 2009, 45(10):89-91.
- 高慧萍, 吕俊. 插件式开发技术研究与实现. 计算机工程与设计, 2009, 30(16):3805-3807.
- Birsan D. On plug-ins and extensible architectures. ACM Queue, 2005, 3(2): 40-46.
- 郭慎平. 基于 Eclipse 的嵌入式开发工具的研究与实现[学位论文]. 武汉理工大学, 2009.
- 罗国庆, 等. VxWorks 与嵌入式软件开发. 北京: 机械工业出版社, 2003.
- Scarpino M, Holder S, Ng S, et al. SWT/JFace in action: GUI design with Eclipse 3.0 (in action series). Manning Publications Co., 2004.
- Davison A. Notes on a JOGL Active Rendering Framework. 2007.