

嵌入式系统开发面临的问题与集成开发环境的应用

1. 嵌入式系统开发所面临的问题

嵌入式软件开发有别于桌面软件系统开发的一个显著的特点,是它一般需要一个交叉编译和调试环境,即编辑和编译软件在主机上进行(如在 PC 机的 Windows 操作系统下),编译好的软件需要下载到目标机上运行(如在一个 PPC 的目标机上的 VxWorks 操作系统下),主机和目标机建立起通讯连接,并传输调试命令和数据。由于主机和目标机往往运行着不同的操作系统,而且处理器的体系结构也彼此不同,这就提高了嵌入式开发的复杂性。

总的来说,嵌入式开发所面临的问题主要表现在以下几个方面:

1.1 涉及多种 CPU 及多种 OS

嵌入式的 CPU 或处理器可谓多种多样,这包括了 Pentium、MIPS、PPC、ARM、XScale 等,而且应用都很广,在其上运行的操作系统也有不少,如 VxWorks、Linux、Nucleus、WinCE 等等,即使在一个公司之内,也会同时使用好几种处理器,甚至几种嵌入式操作系统。如果需要同时调试多种类型的板子,每个板子上又运行着多个任务或进程,那复杂性是可想而知的。

1.2 开发工具种类繁多

不仅各种操作系统有各自的开发工具,在同一系统下开发的不同阶段也有不同的开发工具。如在用户的目标板开发初期,需要硬件仿真器来调试硬件系统和基本的驱动程序,在调试应用程序阶段可以使用交互式的开发环境进行软件调试,在测试阶段需要一些专门的测试工具软件进行功能和性能的测试,在生产阶段需要固化程序及出厂检测等等。一般每一种工具都要从不同的供应商处购买,都要单独去学习和掌握,这无疑增加了整个公司的支出和管理的难度。

1.3 对目标系统的观察和控制

由于嵌入式硬件系统千差万别,软件模块和系统资源也多种多样,要使系统能正常工作,软件开发者必须要对目标系统具有完全的观察和控制能力,例如硬件的各种寄存器、内存空间,操作系统的信号量、消息队列、任务、堆栈等。

此外,嵌入式系统变化更新比较快,对开发时间要求比较紧,尤其是消费类产品更是如此,如果有一套功能强大的嵌入式软件集成开发工具可以满足嵌入式软件开发各个阶段的需求,同时又使用方便,界面友好,那是最理想不过了。美国风河系统公司新近推出的“Workbench”嵌入式软件开发平台正是这样一个“全能选手”。

2. Workbench 嵌入式开发平台的特点和优势

风河公司一直致力于嵌入式软件领域的研究,其 Tornado 开发环境和 VxWorks 嵌入式操作系统产品在嵌入式领域中应用非常广泛。风河公司的新一代 Workbench 开发平台继承了其原有的 Tornado 集成开发平台的一贯优势,并且功能更加强大,由于新采用了先进的 Eclipse 软件框架结构,从而使整个系统更加开放和易于扩展。它的主要优势表现在:

2.1 以开放的 Eclipse 平台为框架，调试环境可充分进行客户化定制

Eclipse 软件框架结构是一个完整和开放的基础平台，它能够将图形工具以及任何必须的功能通过标准的接口集成到同一个开发环境中。目前 IBM，HP 及 Borland 等商业开发工具提供商均支持 Eclipse 开发平台，网上也存在着大量的为 Eclipse 平台开发插件的社区。由于 Workbench 符合 Eclipse 框架，所以这些商用的和免费的符合 Eclipse 平台接口的插件均可以集成到 Workbench 开发环境中，这极大扩展了 Workbench 的功能。例如，如果用户想使用自己熟悉的配置管理工具或者编辑器，就可以找到这样一个插件并集成进来，这样用户就会感到开发过程非常适合自己的需要。

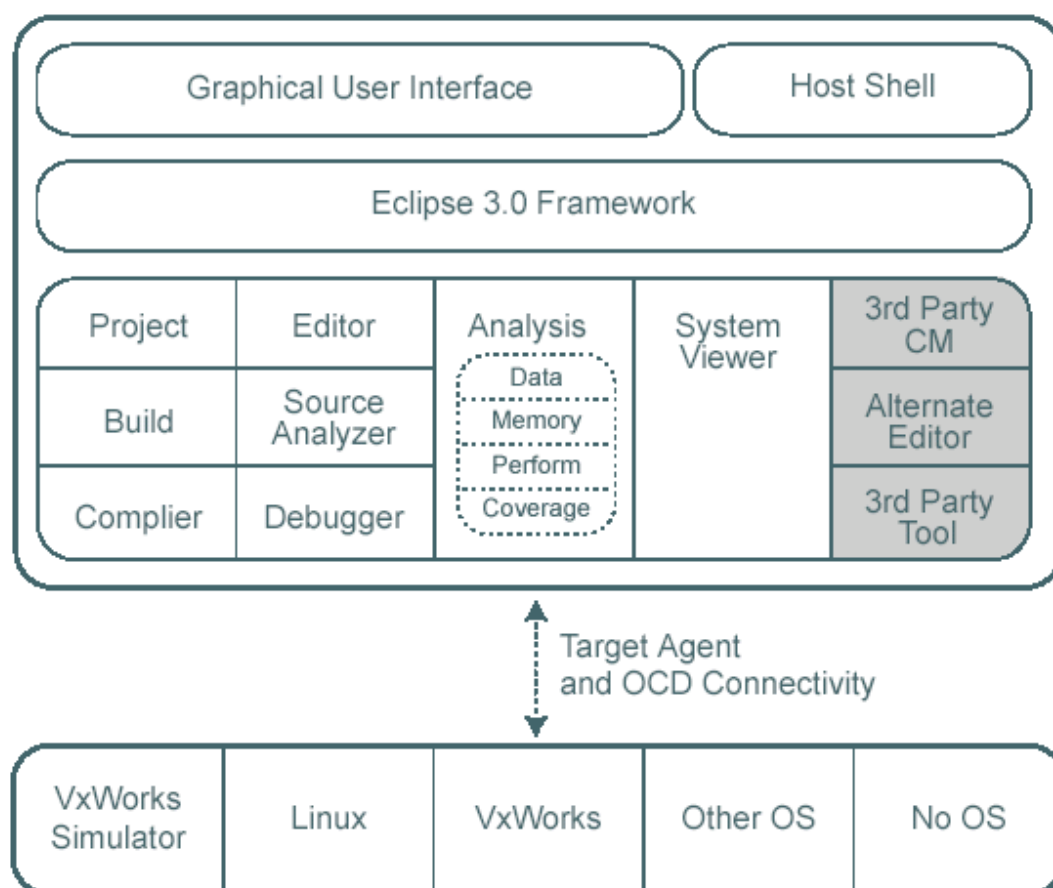


图 1 风河 Workbench 组件图

不仅如此，风河公司还对 Eclipse 做了多方面的增强，从而显著提高了系统的性能。例如 Workbench 实现了异步接口功能，需要长时间运行的任务可以在后台运行，与此同时还能继续接收开发人员的调试请求，这样不至于使开发人员做长时间的等待；另外，Workbench 还优化了对目标机数据的请求机制，即系统只检索在集成开发环境的当前视图中所需要的少量数据，看不见的数据则不向目标机索取，这样就大大提高了交叉调试时系统的响应速度并减少了系统负荷。

2.2 单一的全功能平台，涉及到产品的整个开发周期

一个嵌入式产品的软件开发，其“实质性”的过程包括了从开始的硬件启动，BSP(板级支持包)及驱动的开发，到应用程序的开发，再到后来的测试、验证，直至最后的生产阶段。一般来说，每个阶段都可以找到不同的工具来使用(实际情况也大都如此)。而风河公司的 Workbench 开发平台改变了这一现状，它以一个单一的 Workbench 平台，提供了上述开发阶段所需要的几乎所有功能。具体来说，这包括了硬件系统仿真功能(有配套的硬件仿真器)，

工程管理和构建系统，编辑器，版本管理，命令解释器，调试工具，系统分析工具，系统观察工具，FLASH 编程工具等等，再加上风河公司自己开发的和第三方厂家提供的各种软件功能模块和开发工具的插件，所有这一切，都集成在 Workbench 这个统一的平台之下，这必将大大改善嵌入式软件的开发环境。对一个企业来讲，更可以提高效率，减少投资，简化管理。嵌入式软件开发的各个阶段以及 Workbench 在各个阶段提供的功能如图 2 所示。

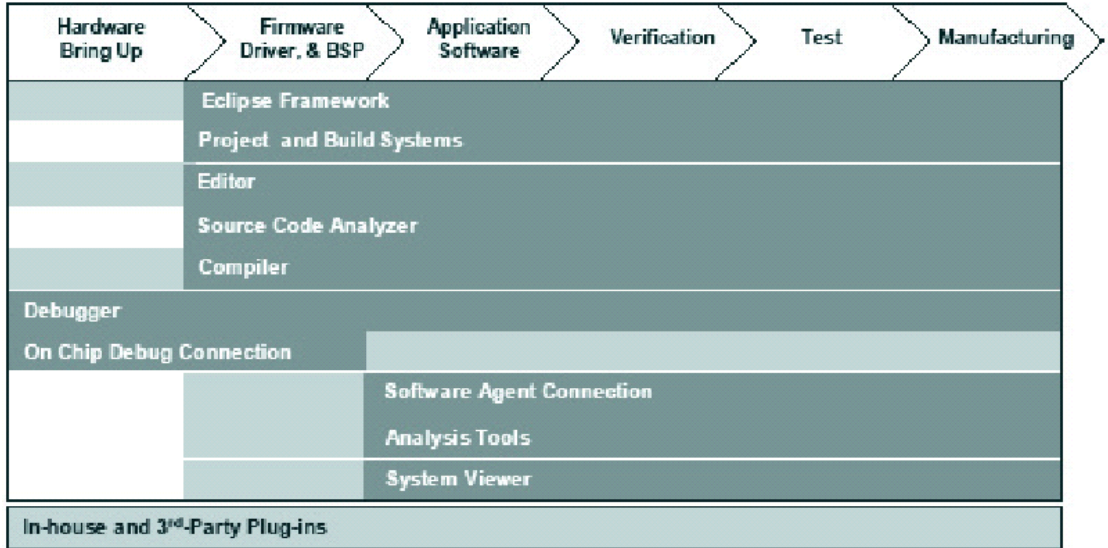


图 2 Workbench 涉及的嵌入式软件开发阶段

2.3 广泛的适用性，特别适合复杂的目标系统

Workbench 平台的广泛适用性主要体现在七“多”上，即多任务、多目标、多模式、多 OS、多 CPU、多连接形式、多主机环境。

多任务：在 Workbench 调试环境下，可以同时连接目标系统上的多个不同的任务(或者进程)，每个任务都可以单独设置断点，进行单步调试。开发者再也不用象以前那样，为了调试多个任务，先断开一个任务的连接，再连接到另外一个任务上去。这就给具有多任务的目标系统开发提供了便利，尤其是需要查看多个任务之间的配合关系时体现的优势更加明显；

多 CPU：我们知道，风河公司的 Tornado 开发环境可以说已经支持了全系列的主流 CPU(或处理器)，而 Workbench 也将继承这个传统，目前推出的 Workbench 2.2 版本已经可以支持 PENTIUM、PPC、MIPS 系列的所有主流 CPU。对 ARM、XSCALE、SH 等系列的支持正在移植和测试中，不久即将发布正式版本(Workbench 2.3)。

多 OS：目前的 Workbench 2.2 不仅支持 VxWorks 嵌入式操作系统(6.0 以上版本)，而且支持 Linux 操作系统，这确实给广大的 Linux 嵌入式开发人员带来了福音。大家知道，由于 Linux 操作系统的“免费”性质，使得其上的软件开发系统一直比较分散和孤立，用起来不是特别方便，有了 Workbench 以后，嵌入式 Linux 的开发就同 VxWorks 的开发一样方便和快捷了。并且以后，如果有必要的话，还可以增加对其它类型嵌入式操作系统的支持，因为 Workbench 本身是开放和可扩展的。

多目标：Workbench 对复杂目标系统的强有力的支持还表现在它可以支持同时连接多块目标板进行调试开发，更为可贵的是这些目标板上处理器可以各不相同，并且在目标板上运行的操作系统也可任意(既可以运行 VxWorks，也可以运行 Linux)，而调试工作都可以在同一个 Workbench 界面中同时进行，这无疑给一些复杂系统的调试带来了

极大的便利。

多连接：所谓多连接，是指 Workbench 所在的主机和目标机之间可以有多种连接方式进行通讯，如果目标系统中存在以太网接口，那就应该首选以太网连接，因为这种方式速度比较快；如果目标系统中不存在以太网接口，那也可以选用串行口进行通讯连接，除了速度较慢以外，功能上与以太网连接没有什么区别。另外，在目标板初始调试阶段，还可通过风河的硬件仿真器(WindRiver ICE / WindRiver Probe)进行连接，观察硬件状态，以迅速定位硬件问题。

多模式：在 Workbench 中调试程序，即可以采用任务模式，也可以采用系统模式。所谓任务模式，是指各个任务之间独立运行，暂停其中一个任务的运行不影响其他任务的运行(主要用于调试各个任务)；而系统模式则是指只要暂停了系统的任何一处，整个系统都停止下来(主要用于调试中断程序)。

多主机：这是指 Workbench 可以在 Windows、Linux、Solaris 这三大流行的主机操作系统下运行，这不仅可以适合不同开发者的使用习惯，而且在一定程度上有利于某些目标系统的开发，如使用 Linux 主机环境调试运行 Linux 系统的目标板(但这并不是必须的)。

2.4 丰富易用的调试手段，大大加快调试进度

动态链接：用过 Tornado 开发环境的人都对其独特的分模块单独下载、动态链接到目标系统的功能深有体会，而 Workbench 也完全继承了这个特性。有了这个功能，就使开发者不必将所有应用模块都编写完成再行调试，也不用每次将应用模块和操作系统编译到一起然后下载调试，而是编好一个小模块，如果想进行验证，只需将这个小模块(哪怕只是一个函数)编译然后下载到目标板中就行了，这大大加快了调试验证的速度，尤其是在目标板和主机之间是低速连接的情况下更能体现出其优势所在。

目标可视：对目标系统的可视化是每一个嵌入式软件开发人员都希望得到的功能，为达到这一目的，Workbench 除了提供一般的查看内存、寄存器、变量、调用树等的传统视图工具外，还提供了几种增强的系统查看工具：目标浏览器、系统观察器和 SCOPETOOLS 工具集。目标浏览器可以用来查看目标系统每一个系统资源的使用情况和当前状态，包括系统各个模块、任务、信号量、消息队列、内存、堆栈等。系统观察器可对系统事件提供详细的分析和图形化的可视效果，同时还可展示在目标机上执行的应用程序的相关任务、中断以及系统对象之间复杂的交互作用。它能够清楚地显示上下文变化以及信号量、消息队列、信号、任务、用户事件和计时器一类的系统事件，就象一个软件的“示波器”。应用这些系统查看工具，开发人员可以很容易地去诊断和解决嵌入式系统中存在的调度问题(如死锁，饥饿和竞争)，性能问题(如优先级设置，资源的竞争与互斥)，定时问题等。而 RTI SCOPETOOLS 则是专门为设备类软件开发提供的一组强大的可视化工具集，用户可以动态观察整个系统平台，包括应用代码、第三方的库，甚至操作系统本身。Workbench 2.2 中集成进来了用于观察函数执行效率的 ProfileScope 工具，用于观察内存使用效率和检测内存泄露的 MemScope 工具，和用于检测变量或指定内存位置处数值的动态变化情况的 StethoScope 工具。另外，还有可选的用于实时跟踪代码执行情况的 TraceScope 工具和用于检测代码执行覆盖率的 CoverageScope 工具。

仿真环境：在目标系统的硬件开发完成之前，往往软件系统的开发也要同步进行，Workbench 提供的模拟仿真环境(只适用于针对 VxWorks 操作系统的应用软件开发)就提供了这

种同步开发的途径。VxWorks 软件仿真器运行于主机操作系统环境中，是一种从 VxWorks6.0 操作系统移植过来的本地应用程序，它能够精确地实施 VxWorks 6.0 的高级复杂特性，包括实时进程、内存保护等，它还具有完整的文件系统和网络功能，并能调用主机系统 API。这些特性使得在 VxWorks 软件仿真器中开发复杂的系统成为可能。例如，在 VxWorks 软件仿真器中可以组建多个网络，这些网络不但互相之间可以通讯，还可以通过主机系统的网口与外界相连，实现一个完全真实的网络环境；再例如，可以利用主机系统的显示屏幕来开发具有图形用户界面的应用程序，这是通过调用主机系统与图形相关的 API 来实现的。一般来说，用户可以为主机系统的任何资源编写 API，而在 VxWorks 软件仿真环境中进行调用，这极大地扩展了 VxWorks 软件仿真器的功能，也更便于应用程序的同步开发。

除了以上所提到的需要特别强调的功能外，Workbench 的强大调试功能还表现在支持各种类型的断点设置、方便灵活的操作系统内核配置工具，具有主机 Shell 和目标机 Shell，具有 4 种命令解释器，以及丰富的视图界面等等，详细情况请参考 Workbench 的说明及手册，这里就不多做介绍。总之，使用 Workbench 给开发人员的感受就是方便灵活，功能强大，几乎无所不能。

3. Workbench 与 Tornado 的功能对比

Workbench 平台相对于 Tornado 平台来说，各项功能均有所增强，也有不少新加入的功能，主要的功能对比如下表所列：

表 1 Tornado 与 Workbench 的功能对比

项目	Tornado	Workbench
工程管理	基本的工程管理功能，对分级的工程支持有限，并需要手工修改 makefile 等，比较麻烦	支持任意工程的分级控制，采用简单的“拖拉”方式就可完成工程项目的分级构造，非常方便
工程项目	支持可下载的工程项目和客户化的 VxWorks 工程项目	增加了对系统引导工程项目、实时进程工程项目、文件系统工程项目、用户定制工程项目等的支持
编辑器	功能比较简单，只能完成一般的编辑功能	具有语法分析功能，可以进行语法着色显示、符号定义查找、符号补全、函数参数提示等功能，具有强大的符号查找和浏览能力
断点设置	支持任务级断点和全局断点	增加支持对处理器硬件断点等的支持，并且对各种断点还提供多种控制手段
仿真环境	基本的 VxWorks 仿真环境，可以操作主机系统中的文件和网络访问	增强了仿真环境中的复杂组网功能，提供了调用主机系统 API 的能力，并可模拟中断信号的产生
目标连接	支持网络、串口等连接方式	增加了对硬件仿真器的连接
下载的代码量	下载的目标代码一般在 10MByte 之内，超过 10MByte 会出现下载速度慢，系统不稳定的问题	设计能力为可下载 750MByte 的目标代码，充分满足开发和调试需要
目标操作系统	只支持 VxWorks 操作系统	增加了对 Linux 操作系统的支持，

		并且还可以扩展到其它目标操作系统
目标系统数目	单一的目标系统,同一个开发环境中同时只能建立一个连接,调试一种类型的单板	可以同时进行多目标板、多 CPU 体系结构的调试
多任务调试	不支持,每次只能挂接一个任务,或者进行系统级调试	支持,同时可挂接目标系统的多个任务或进程进行调试
第三方工具	结构比较封闭,集成第三方工具比较困难	采用 Eclipse 的开放架构,第三方工具容易集成,且资源丰富

Workbench 是对 Tornado 的一次脱胎换骨的升级,但目前并不能说 Workbench 就可以完全取代 Tornado。这是因为 Workbench 只对 VxWorks6.0 以上的版本(具有“进程”的概念)进行支持,如果用户想使用 VxWorks6.0 以下的版本(扁平地址空间,应用程序均在内核中运行),那只能用 Tornado 进行开发。

4 . Workbench 的当前应用状况

如上所述,Workbench 无疑是当前嵌入式软件开发领域中功能非常强大的一个集成开发环境,它最适合应用于复杂系统的开发或多个开发团队的合作开发,比如一个复杂的系统需要用到多种 CPU 或多种目标操作系统,或者应用软件本身非常复杂,具有多个任务,并且相互之间关联紧密,或者多个项目组之间需要进行协同开发和软件模块共享,或者企业涉及到了从硬件开发,到软件开发,再到生产测试的全过程。在这些情况下,考虑使用 Workbench 平台则非常合适,因为这样不仅能快速有效地进行系统开发,并且能够有效地进行项目的组织和管理,最终从整体上降低企业的成本。

Workbench 目前正式发布的版本是 2.2 版,即将推出的 2.3 版本则提供了对更多种类处理器和目标操作系统的支持,其功能和适用范围也将进一步增强。在国内,已经有不少大型企业都正在积极地评估 Workbench 平台的使用,并体验其强大的开发能力和方便灵活的操作过程。在国外,已经有一些企业用 Workbench 开发出来了自己的产品。相信不久以后,风河公司的 Workbench 开发平台将会遍及到嵌入式软件开发的各个领域,并使嵌入式软件开发成为一个令人愉悦的过程。

作者:康宇峰,风河公司应用工程师