
将Windows转变为满足
物联网与工业4.0的纯软件
实时效能平台



IntervalZero

新一代工业、机器视觉、医疗和其他仰赖于确定性的系统已试图将高阶绘图、丰富的用户界面、IoT 云端联机与硬实时效能、排序和精准度相结合。而现今执行 64 位 Windows 的工业电脑，再加上多核处理器上单独的确定性排程器辅助，就可以在软件定义的周边装置提供如此精确的实时效能。

在智能边缘 IoT 或云端联机等日益丰富的应用需求下，硅集成电路和效能的进展脚步有增无减。然而，这进展一直朝着不同的方向前进，特别是将功能集成到单一硅芯片上的部分。嵌入式原始设备制造商（OEM）的任务是要决定如何将这革命性的发展转化为能满足客户需求的实际产品，包括高分辨率影音、机器视觉、实时性的工业产品（如六轴运动控制）、实时联机能力和丰富的用户界面，通常用户界面还必须能实时显示链接应用程序的复杂实时图形数据。

今日的硬件已经具备这样的能力，但要实现的话就必须借助软件。只要处理器内核足够强大，就不再需要依赖客制化硬件执行专门功能，因为这现在都可以由软件来完成。与硬件相比，软件更容易更新与改善，而这正是 OEM 厂家能展现真正价值的地方。

要顺利执行复杂的实时软件应用程序，最好的方法就是从正确的硬件环境着手。如上所述，硬件集成一直朝着不同的方向发展。一方面，有一种趋势是将实时系统使用的各种装置集成到单一芯片，这些装置可能包括多核处理器、DSP、FPGA 或高阶图形处理器，例如：我们最近看到集成处理器内核和 FPGA 的装置，或者能进行密集数字运算的高阶图形处理器内核，像是 DSP。

另一方面，我们还有机会利用现今的多核中央处理器（CPU）。这些 CPU 的强大效能主要来自多核架构，而逼近 3 GHz 的频率速度也在其中发挥了一定作用。这些标准市售现成工业电脑藉由一些额外的指令和排程器，就能使提供的平台具有 DSP 等级的处理、效能、排序和精度。使用现代的 CPU，这种处理即可在浮点中完成，以进行比通常在 DSP 使用的定点更多样化的计算。这样的效能将重点从尝试优化每条指令的使用，转变为真正充分利用多核 IPC 的实际功率。

这趋势已经产生能超越传统 DSP 处理器的装置。另一项重大发展是 64 位架构的转变，不但能与前身 32 位向下兼容，同时又提供大幅增强的效能。这样有几个优点，因为即使是集成不同功能，以及各自不同的指令集和协议的高集成芯片，也会对统一的软件环境造成障碍；既增加硬件困难，也加深了开发团队想避开这些困难的负担。

英特蒙 RTX64 的建置采用后者，将 Windows 转变为全功能的实时操作系统（RTOS），完全运行在 x64 多核硬件上，此外这也能依照实际映像的实体 RAM 大小来存取 128 GB 非分页内存。总体而言，Windows 512 GB 物理内存胜过 32 位 Windows 的 4 GB 物理内存限制，这样一来大量的可用内存就能分配给以前无法使用的 MRI 医学成像和高阶影片编辑等应用程序。

最重要的是，RTX64 以多核 x64 装置的形式提供单一的商用硬件环境，使单一软件环境能够容纳 Windows 及其丰富的用户界面、可用的应用程序和开发环境。Windows 还能无缝连接到可从 1 个核心扩展到 63 个核心的全功能实时对称多处理（SMP）RTX64 环境。



应用程序可编译成单一基准代码，不需要 FPGA 或 DSP 针对那些必须分别编译并链接主应用程序的不同代码而执行逻辑。单一硬件，单一操作系统环境，单一工具，单一基准代码，这表示一个团队可以沟通与合作，并制作高效能、可扩展的应用程序，同时大幅缩短上市时间。

解放的集成架构

摆脱实时系统以及像是用户界面等其他功能的孤立运作后，设备制造商就能以更低的风险与负担，探索更多创新的解决方案。以用户界面为例，现今先进的应用程序与其用户，要求的是功能丰富、互动性高、支持触控的图形化界面。先进的嵌入式系统更早已超越过去的无头（headless）系统。此外，藉由这个界面提供明确且一致的使用者体验也变得日益重要，因为这能有效巩固产品的品牌形象。

过去开发嵌入式用户界面的方式，是由一个团队先在桌面电脑使用 Photoshop、Illustrator 等软件，或后续针对嵌入式系统的工具进行设计。成果通常是一个搭配仿真数据和界面的 UI 模型，接着交给开发实时应用程序的团队，由他们在自己的实时操作系统环境将 UI 设计实现出来。这个过程无可避免地会产生许多修改与妥协，而且测试工作大都拖到开发后期才会进行，对产品上市时程的影响也可想而知。

使用 RTX64，开发团队可选择任何 Windows-based 的工具来建立 UI，并确信界面中的对象能直接与 RTX64 API 沟通，也可和嵌入式应用程序交换指令与数据。任何对 UI 或嵌入式功能的变动，都能在两者间快速追踪并更新。由于 RTX64 并不会改变 Windows，而是作为 Windows 的一个实时扩展功能，所以能为硬实时应用

程序提供同样无缝的网络通信、数据库和存储存取。也因此 RTX64 和 Windows 的通信是无缝集成的，不依赖像是在其他操作系统中常见的远程过程调用（RPC）、虚拟化或 hypervisor 等机制。藉由第二个排程器和 RTOS 基础架构，UI 功能可在 Windows 执行，而实时功能则在 RTX64 排程器上执行。将适当的任务安排在正确的排程器上，带来最佳的整体效果。

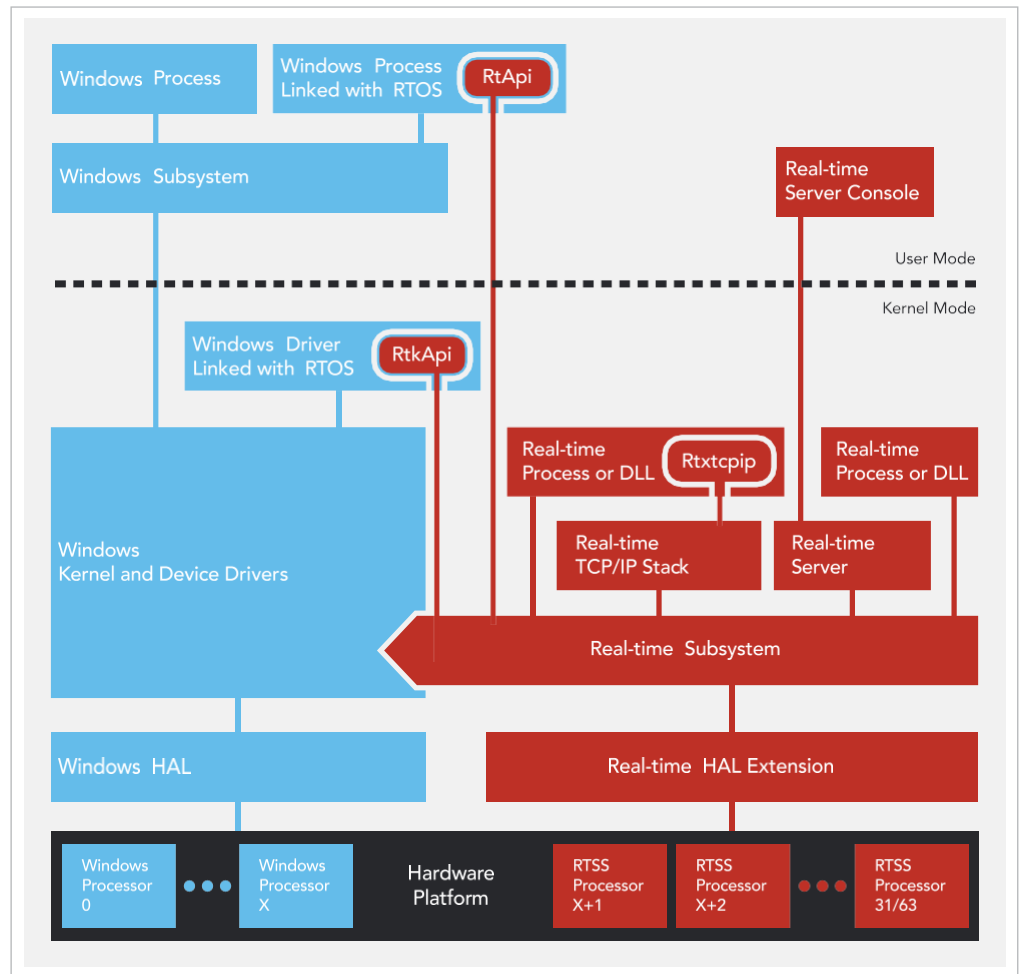


RTX64—进入 64 位的全新开始

英特蒙团队特别重新打造 RTX64，藉以开启 64 位实时运算的世界，而不是从 32 位产品移植。像是专业影音、高阶医疗设备以及包含机器视觉和丰富用户界面的高阶工业控制系统，这些应用所提出的需求只有内含丰富用户界面的高阶 64 位系统才能满足。

图一

RTX64 提供一种利用先进技术—特别是高速多核 x64 的架构—效能及规模胜过依赖 DSP、FPGA 和微控制器的传统嵌入式环境。



RTX64 提供一种利用先进技术—特别是高速多核 x64 的架构—效能及规模胜过依赖 DSP、FPGA 和微控制器的传统嵌入式环境（图一）。通过在单一硬件环境以更高的效能执行功能来实现上述这点，并能与 Windows 结合使用，提供丰富的使用者环境和存取大量可利用及支持实时操作的应用程序。

首先，RTX64 的硬件抽象层（HAL）和 Windows HAL 不同，并会与 Windows HAL 一起运行。因此，从一开始就不需要对 Windows 进行任何修改。这两个操作系统—GPOS 和 RTOS—并列执行，且通过各种机制进行通信。RTX64 HAL 在时序降至1微秒（视硬件支持而定）的期间，可从 1 个核心扩展到 63 个核心以提供确定性的实时效能。常驻在 RTX64 实时子系统（RTSS）中的排程器可将执行序分配给不同内核以实现 SMP，无须依赖虚拟化或会增加延迟的复杂进程间通信。

这种独立运行的能力来自于所有内核都可使用不须分割内存的极大内存空间。整个系统能存取高达 128 GB 的非分页内存和 512 GB 的物理内存。这为越来越依赖可视化的医疗应用，例如目前新开发的视网膜光学断层扫描（OCT），或需要准确器官成像处理（如跳动的心脏）的实时手术机器人提供显著优势。对于那些不只是向用户呈现视觉数据，还必须实时处理数据以驱动运动控制的先进工业控制系统，以及生产过程中的零件检验来说也是至关重要。

像这样的内存空间可提供给高性能通用硬件平台使用，使得设备制造商能够开发专门的软件，以执行需要使用专门硬件组件才能执行的极端专用功能。经验显示，结合不同的硬件会牵涉到使用不同语法（如 C++ 与 Verilog）和内容大相径庭的软件套件，这不仅会大幅拉长开发时间，也会限制效能和扩展性。扩展这类系统只会增加复杂度，因为每个不同的附加硬件都有自己的界面和独特的软件需求。

而 RTX64 RTSS 包括实时排程器，是完全独立于 Windows 内核与 Windows 排程器的。Windows 和实时线程之间没有既定的交互影响或干扰。开发人员只能使用实时 API 进行线程之间的特意通信。系统也提供实时 API 与用户模式的 Windows 应用程序一起使用，或提供实时内核 API 与 Windows 内核驱动程序一起使用。

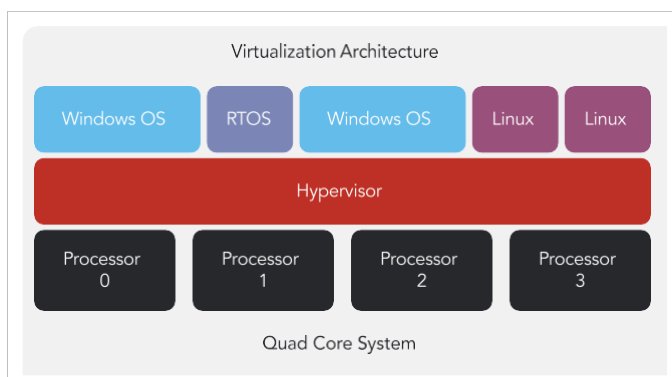
在 RTX64 启用的情境中，应用程序可以像常见的 Windows 应用程序一样呈现给用户，但在用户界面的背后，许多功能其实都会利用到 RTX64 实时处理进程，例如：设备控制程序可能会显示正在加工的零件视图，用户也可借助触控屏幕访问控制和设定。然而，实际的应用程序包含了两个部分—Windows 程序能经由实时 API 在内核和使用者两种层级上与实时控制程序进行通信。

在核心层级，Windows 驱动程序会将数据送到 RTX64 端，即可控制设备的进程，并接收目前的位置数据然后传送至用户界面，或藉由实时核心 API（RtkApi）进行某种处理。在使用者的层级，操作人员可以在触控屏幕上设定数值或开关位置等，并与 Windows 进程进行通信，该进程则会使用实时 API（RtkApi）与 RTSS 进行通信。这两种 API 会直接与实时控制程序常驻的 RTSS 进行通信。



随着对实时和嵌入式系统的丰富用户界面需求不断增长，开发人员面临到的困境是如何将这些界面与传统不支持复杂用户界面的 RTOS 环境相互链接。藉由将 RTX64 扩展到 Windows，开发人员就可直接使用自己喜欢的图形工具来设计用户界面，并以 RTX64 API 将用户界面直接链接到底层的实时应用程序。对某些人来说，更具吸引力的可能是直接购买现成的软件控制和数据摄取（SCADA）工具，包含许多预先设计但可客制化的仪表、滑杆、开关和泵、水箱、驱动器的图标等，并从中使用相同的 RTX64 API 来连接系统以进行开发。

影像数据也适用于此，有多种工具和应用程序能表现像是热分布、流体动力学、应力等物理现象，同时都运行在 Windows 上。现有的图像处理应用程序已经能进行边缘侦测和零件检测所需的其他操作，而这份清单还在持续增加中。OEM 厂家在这类 Windows 用户界面等级上提供丰富的“自建或购买”选项让开发人员有自信地使用和（或）尝试，因为他们知道这些底层实时应用程序的界面定义明确且能开箱即用。



RSMP 为效能和扩展性铺路

对于如何利用多核处理器的议题存在着不同的思想流派，这些流派基本上分为非对称多处理（AMP）或虚拟化，和对称多处理（SMP）。达成 AMP 的其中一种方式是要求操作系统的副本运行在每个内核，此时就要将内存分配给各个内核，随之而来的是进程间通信的需求，而这会增加负担和延迟，也就是确定性的敌人。如果尝试使用 Windows 执行用户界面，同样会出现效率低下的情况，加上还要在 Windows 与 RTOS 和分割内存的多个实例之间进行进程间通信，并且在系统处理（在 RTOS 上）和显示（在 Windows 上）实时视讯数据，就会需要更多的工业电脑，接着很快就会阻塞。将系统扩展到更多核心需要更多的 RTOS 副本、更多的分割内存和应用程序的重新配置。

另一种达成 AMP 的方式是使用虚拟机管理程序

（hypervisor）实施虚拟化，hypervisor 是直接运行在硬件的独立软件层，可在操作系统之间划分硬件（图二）。有些多核处理器甚至具备用于虚拟化的内建硬件辅助，能为每个操作系统提供一个虚拟“主板”。虚拟化通常用于支持“分离式内核”，这些内核与系统的其余部分隔离，只经由严格控管的机制和协议进行通信，这对于某些情况是有帮助的，但主要目的在于隔离，而 SMP 的目的是集成。

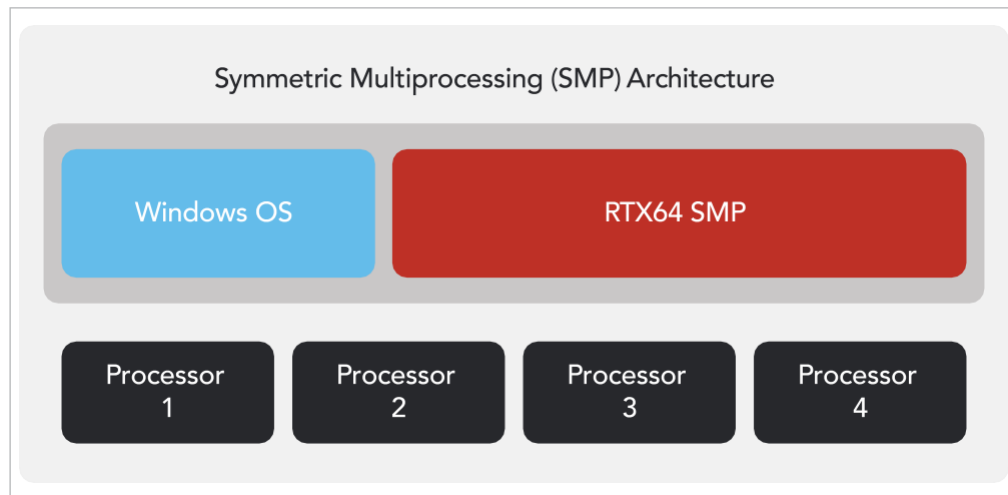
图二

另一种达成 AMP 的方式是使用虚拟机管理程序（hypervisor）实施虚拟化，hypervisor 是直接运行在硬件的独立软件层，可在操作系统之间划分硬件。

比起在 RTOS 和内核之间出现障碍的 AMP 解决方案来说，RTX64 代表 Windows 的 RTOS 扩展，并与 Windows 一起运作，形成了使用 SMP 将多处理器硬件视为单一共享资源的单一操作系统环境。整个操作系统环境只要单一副本（包括实时子系统与其实时排程器）就可以存取所有分配给该子系统的内核（图三）。与 AMP 不同，程序代码只需编写一次，之后便能静态重新分配线程，或在新增内核和重新分割等新增功能的时候进行扩展。由于所有内核以及线程都能直接存取共享数据，所有资源都可提供给实时进程，因此不须额外的副本或使用复杂的进程间通信方案或远程过程调用。

图三

RTX64 代表 Windows 的 RTOS 扩展，并与 Windows 一起运作，形成了使用 SMP 将多处理器硬件视为单一共享资源的单一操作系统环境。



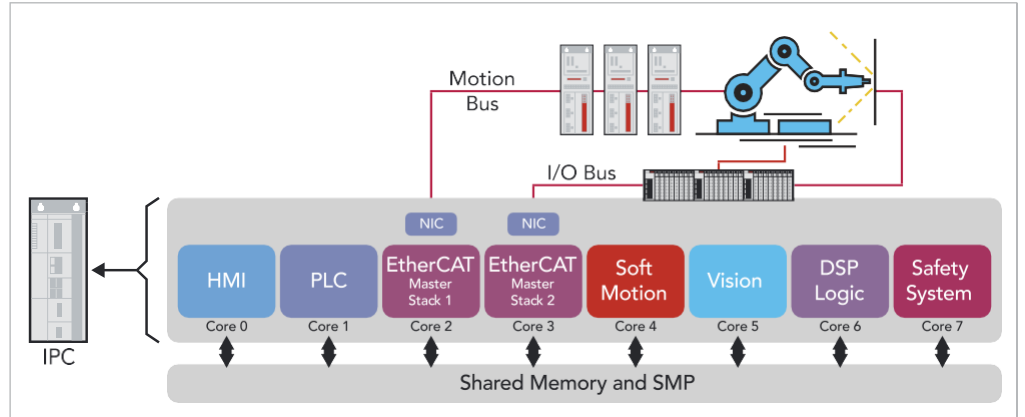
在同质的硬件平台上使用单一扩展操作系统环境，能减少设备制造商做出“是否有足够的内核来做我需要的东西？”或“要新增多少内核才能为这个应用扩展到所需的附加功能？”等主要硬件决策，不须在不同的硬件组件（如 FPGA 和 DSP）之间桥接界面，也不用为了效能更高但程序编写要求不同的零件去调整程序代码，更不需要请各种不同的硬件专家来建立或升级产品。团队会根据单一程序语言（如 C++）来定义效能。

这形成了另一种优势，就是针对整个项目使用例如 Windows Visual Studio 的单一开发工具。Windows

可以当作整个系统，也就是 Windows 功能和实时编码的开发环境，其他基于 Windows 的工具也能结合使用，例如需求分析、版本控制或静态分析工具等等。实时子系统的用户模式还包括连接到 RTSS 的 RTX64 服务器控制台，开发团队可以使用他们最喜欢的实时调试器、效能测量器和分析器来调整实时子系统，也能用相同的术语沟通和讨论，不须学习 Verilog 或 DSP 程序语言。

图四

EtherCAT 提供网关，用来集成现有的总线组件，例如 CANopen 或 Profibus。EtherCAT 在 RTX64 上以软件的方式运行，而不需要插入系统总线的专用 EtherCAT 卡。



网络和实时联机能力

在 IoT 时代，连网能力已成为一项必备条件，先将装置连接到局域网络，再连到服务器，最终连接到云端。Windows 天生就具备网络联机功能，且能客制化以便与实时处理进程交换数据与指令，同时也能提供远程用户界面，让用户可以在几乎任何地方与系统互动。然而，Windows 本身的网络联机并非实时。

不过，当相同的系统同时运行 Windows 和 RTX64 实时扩展环境时，是可以通过 EtherCAT，一种用于控制自动化技术 (CAT) 的以太网总线系统，如图四所示，来新增实时以太网联机功能。EtherCAT 也提供网关，用来集成现有的总线组件，例如 CANopen 或 Profibus。EtherCAT 在 RTX64 上以软件的方式运行，而不需要插入系统总线的专用 EtherCAT 卡。

EtherCAT 还能在一个或多个处理器内核上运行，直接与系统中使用的任何网卡 (NIC) 通信。个别装置则可在 EtherCAT 的配置过程中进行选择。

EtherCAT 为工业控制系统中常见复杂且昂贵的布线方案，提供了一个具吸引力的替代选择。单一电缆即可

承载多个控制通道以及具有安全完整性等级 (SIL) 3 认证的安全信号。此外，还能将多个处理器内核专用于 EtherCAT 功能，藉此实现真正丰富的控制联机能力，同时省去额外专用硬件所带来的费用与功耗。

英特蒙的 RTX64 为 Windows-based 实时系统开启了新局面，带来高阶机器视觉、可视化以及丰富的用户界面。藉由发挥底层多核处理硬件的效能，开发人员能将硬件组件的功能转化为软件组件。这对 OEM 厂家来说，代表没有库存压力，零件还能无限复制。对软件团队而言，也不需要具备像是 DSP 或 FPGA 这些硬件的专业知识，所有程序代码都集中在统一的基准代码中，管理起来也更为方便。

RTX64 可无缝集成至 Microsoft Visual Studio 集成开发环境，并部署到单一集成的 Windows 系统。在扩展 Windows 的同时提供具有限制延迟的硬实时精准度，并以多核处理器作为可扩展的原生 SMP 解决方案。对成本、上市时间、库存、用户体验和原始系统效能上所带来的正面影响是极具革命性的。