

# 智慧机械-设备模块化的趋势与方案

WHITE PAPER

  
**KINGSTAR**

# 简介

随着客制化产品需求日益增加，以及产品生命周期日渐缩短，从事自动化生产的工业控制器与自动化设备势必也要走向模块化架构。为了生产线的弹性需求，自动化设备必须改为模块化而设计，而工业控制器也必须能支持低阶到高阶不同生产需求的设备。用于开发控制器核心的实时控制平台必须具备让开发工程师能够针对不同客户或市场增加新的功能；对设备制造商而言，控制器本身必须可以新增或移除模块，或选择不同等级的硬件；生产端的系统集成商也能与设备制造商同样受益于模块化。最后，设备的架构设计要能与工厂网络系统集成，而且操作员必须能够加载与建立不同的工作。KINGSTAR 软件控制平台是一个专为模块化所设计的设备自动化软件平台。在这份技术简介中，我们将列出模块化的主要类型、建立模块化设备的最佳实务、并介绍 KINGSTAR 平台及其独特功能如何促成改变。

## 模块化类型

为了准确描述工业设备的不同模块化类型，以下范例为我们在客户应用上最常见到的种类。为符合许多不同的使用情况，愈新、愈复杂的设备使用下列诸多乃至全部的模块化类型。

## 可插拔装置

可插拔装置为无法独立运作的设备类型。以往，这些设备会有自己的控制器，且透过 I/O 与其他设备同步。例如：该模块可装载一个零件、将一片 PCB 翻面、或标识一个零件。这类设备可以依生产需求从一台设备移动至另一台。随着科技演进，这类设备现在可作为从站 (slave) 装置，由主要设备透过现场总线直接控制。

## 选用硬件

现今多数设备都有选用功能，例如雕刻、效能测试或特殊运作等，这些功能大多需要额外的硬件作业，因此设备是由许多独立工作站组合而成，而各个工作站皆有其特定功用，因此设备控制器必须能够辨识并且控制每个工作站。

## 硬件版本

许多公司针对不同的酬载大小及精密程度需求推出不同版本的设备。这些不同版本在驱动器与致动器上使用不同的硬件型号，但具有几乎相同的控制逻辑，因此单一控制器类型必须能支持所有不同的硬件版本。

## 选用软件

现今设备通常有「仅限软件」的选用组件，如工作编辑器、CNC 中的 CAD/CAM 软件，或是藉此优化产制速度与预估维护需求等的分析软件。



### 人机界面

由于自动化设备很少独立作业，需要与其他设备及工厂 SCADA 和 MES 系统做集成。客户端（非设备制造商）负责决定与其他设备和工厂信息交换用的现场总线及数据格式，这代表机器内客户使用的接口必须为一个独立的模块，有不同的选项以兼容于所有客户的工厂。

### 工作接口

考虑设备在部署的生命周期中通常会生产不同的产品，就需要支持由操作者所定义的不同工作。各个产业与工厂可能在定义这些工作时各有各的工具与形式，这也表示工作直译器必须具备多种版本的独立模块，让设备可以使用在不同的环境中。

## 模块化架构

这里我们将描述有关硬件及软件架构的最佳实例，让工业设备可完全模块化。

### 从站控制器或硬件组件

从站控制器与硬件组件之间的不同在于，从站控制器即内建独立的控制器。

如前述，为了让客户选择所需要的功能，自动化产线应该设计为执行各别行程的独立工作站。这些工作站每个都是硬件组件，上面有连接到主设备的简易接口，

通常为现场总线与电源的输入和输出连接器。组装设备时只需要将不同组件连接在一起即可。

依照执行的工作而定，部分组件或许不会被使用到，但这也表示组件可于其他设备再利用。如前面所提被称为「可插拔装置」。若这些装置仅适用于供货商的特定设备，或逻辑操作相当简单，则无须内建控制器，因为插入的设备内即有一个控制器。

较为复杂且可用于不同设备的特定组件，如上料手臂，应内建从站控制器。从站控制器内含控制其组件的操作逻辑，但不会执行任何工作或用户程序，仅接收主设备的工作指令。

### 硬件抽象层

控制逻辑必须能在不同的硬件版本上执行，相同的逻辑才能被用于不同的负载设备，也能让控制逻辑不被硬件更新所影响。一般设备会持续生产许多年，硬件的变更在所难免，所以必须确保控制逻辑独立，并且可安装于不同硬件的设备。

因为更新通常包含了对设备与工厂接口的安全性修正，这点非常重要。为每个硬件版本建立各自的更新会导致不必要的负担，且较旧的设备在缺乏抽象层的状况下，非常有可能无法收到安全性更新。

这也赋予设备制造商更多的采购安全性，因为他们可以在潜在供应问题出现时更换硬件品牌。近期半导体



市场的缺货状况就使得设备制造商将此列为重点采购要求。

### 模块化软件

与硬件一样，软件功能也必须具备模块化开发。软件平台必须与内部现场总线及不同的应用层互连。平台内的硬件抽象层控制内部现场总线，提供应用程序可用的硬件组件信息。

不同的应用程序时常由不同的程序语言开发且可能需要实时性，因此软件平台也应该在实时环境中运行。同时，平台的接口也需要尽可能支持多种的开发环境。

另外，软件平台需要与不同的应用程序维持松散的链接，这样就算其中一个应用程序故障，也能维持系统正常运作。这点至关重要，因为部分应用程序可能内建安全措施。如果让单一应用程序影响到整个系统，所有应用程序都会被要求依循安全开发规则。

由于模块与模块之间时常需要分享信息，因此软件平台还要能容许不同应用程序间的通讯。通常会保存每个组件目前的行程状态，并且让所有模块更新。

为使所有模块运作顺畅，会有一个主应用程序，依照配置文件及联机硬件执行开启、初始化平台并启动各个模块。也会控制整个开机及关机程序。

### 设备接口

设备与整个工厂间进行通讯的能力为目前工业 4.0 的重要要求。

设备接口必须可远程访问硬件以进行维护与诊断、利用设备整体信息进行行程管理及分析，并且让不同的功能与设备相互配合。

为了减少对设备的潜在攻击和避免生产过程中有意外产生，不应该直接存取硬件，而应该透过设备控制进行维护与诊断。如此一来可确保设备处于维护模式，而且硬件与控制器之间正常联机。另外为了更多的安全性、远程控制，非变量及 API 存取，此接口还需要使用与其他设备接口不同的协议与软件模块。

行程管理及分析接口会定期搜集数据，通常相当简单；但是为了达成实质的分析及管理，还要能够分析结果及行动，修改设备的优先级、排程与校准。除了定期存取情境化的数据以外，此接口还应该具备能更新设备的 API。

设备共同运作的接口复杂度视设备对彼此依赖程度而不同。与其为每个项目建立不同的接口，近来的标准提倡与管理分析接口融合，并提供一个完整的设备功能（包含状态）描述，还可以修改参数及呼叫功能。另外为了安全性，不同的功能与参数可能有不同的存取限制。工厂内的各个系统，如 SCADA、MES、分析及其他设备，会使用不同的凭证存取所需的不同功能。



这样一来只须修改不同凭证的定义，不再需要修改设备接口即可达成集成。

### 工作档案

现代设备可以在不同工作之间变换，甚至可能同时执行不同项目的不同工作。如果正处于不同的工作站，工作内容就没办法由操作员执行手动定义。

如同有 G/M 代码档案的 CNC，工作定义必须依照工厂排程系统，储存为易于转换、修改和由服务器下载的档案。在某些情况下，与其使用排程器指派工作给每台机器，更好的是使用 ID 标签指派工作。设备在读取下个工作站的 ID 后便会自动下载其工作档案。

### 技术要求

为达到模块化设备所需要的架构，设备使用的硬件、软件与协议都需要具备几个重要功能。

### 具热插拔功能的现场总线、总线扫描及自动配置

第一个关键要素为用于连接控制器与不同硬件的现场总线。鉴于模块化设备通常有许多选项，现场总线应容许超过 100 个装置连接，且因部分组件需要高取样率，每个循环应低于 1ms；还要能链接与标记硬件，让硬件组件可相互连接并辨认彼此，即使多次使用相同硬件亦然。由于可能有许多种组合，现场总线要能

让链接的硬件在开机时被扫描到，使控制器自动对应所连接上的硬件。

如上所述，维护与诊断应可透过控制器远程联机进行，所以总线可容许装置配置、诊断及更新。如果设备需要在运作中进行维护或修改（例如部分行程耗时过长），现场总线还要能支持热插拔，在运作中移除或新增硬件。

因为会使用到各种不同的硬件，当中的通讯协议就必须是被许多硬件供货商所采用的，这也让设备制造商在供应出现问题时还能拥有许多其他硬件的选项。

### 标准工控机

下个重要组件为控制器硬件。现代控制器内含实时与非实时组件，意指硬件必须有多核心 CPU 和网络孔。考虑所需数量大，控制器通常建议使用特制 SoC。但是有时可能会适得其反，例如控制器要使用在许多不同的设备上。在这种情况下，要就是 SoC 能支持到最复杂版本的设备，不然就需要许多不同的 SoC，如此将显著减少每款 SoC 的需求数量，并增加每台新设备的开发时间。

对模块化设备而言，标准工控机是最合适的解决方案，因为可以在各个设备版本上使用不同的 CPU 及内存，无须额外的工作或设计成本。另外标准工控机可使用标准 PCIe 扩充卡，这对工厂通讯而言相当重要。



### 多重接口的子系统

建立模块化软件的关键是一个链接所有组件的中央子系统。另一个选项是为不同模块建立可相互呼叫的金字塔，但是这种架构不如新型模块般有弹性，而且无法轻易插入至现有的数据堆栈中。

软件设计通常建议接口只处理模块间的互相连接，但是因为几乎所有应用程序模块都会使用到这些功能，导致增加开发的复杂性。因此，我们建议开发一套子系统作为应用程序平台，在应用程序互连之外还包括现场总线、I/O 接点和标准运动控制。因为其中会有实时应用程序模块，这个子系统就必须在控制器中有实时性的部分里运作，这也表示子系统不应该包含在控制器非实时部分里运作的主应用程序。软件架构会有一个启动控制子系统、包含现场总线的主应用程序，现场总线启动并扫描，接着主应用程序依扫描到的硬件与配置档案启动附加的软件模块。

其他软件模块种类繁多，可能是一个实时行程、一个可视化工具或数据分析，也可能由不同程序语言和环境所开发，所以子系统必须要有同时给 C、C++ 与 PLC 等工业实时环境，像是 .NET、Java 或 Python 等 IT 环境的接口。

### 标准通讯及控制

当部署设备时，最常碰到的状况就是将设备加入工厂现有的产线中，要能连结现有设备和工厂自动化，这表示广泛的支持工业标准与专用协议。因此要能新增自选协议板，通常为一张 PCIe 扩充卡，与工厂及其他

设备介接。为避免修改控制器软件，设备接口还要使用开放与普遍的工业标准，即可轻易找到将标准协议转换为工厂协议的组件。

此外，客户们通常会要求对控制逻辑稍作修改，或新增客户自有的软件逻辑。此逻辑在多数案例中用于通讯接口与其他设备间的控制和同步。为了避免每个项目都需派遣工程师前往调整控制器，控制器本身应该内建标准控制环境，让客户自行新增操作逻辑，此环境一般为软件 PLC。

### 脚本工作

由于设备会同时执行许多不同的工作，这些工作必须在易于修改与更新的档案中定义。在部分案例中，工作还需要在运行中同时进行编辑。比起使用编译应用程序，利用脚本程序语言 (scripting language) 并在控制器内建编辑器会更有弹性。

### 远程控制与维护

最后，远程访问的功能越来越重要，应用于设备控制与诊断、为设备维护做准备，以及明确定义设备将执行的工作，如此就能让未经特定训练的当地工程师进行维护。

### KINGSTAR

在检视不同的模块化类型，及讨论最佳实务与建立模块化设备的技术要求后，让我们来探讨 KINGSTAR 的解决方案。



如前述，KINGSTAR 是设备自动化的软件平台，主要客群为控制器与设备制造商的厂商。KINGSTAR 为美商英特蒙旗下部门，公司总部位于美国，并在全球设有办公室，多年来的专长为实时与嵌入式系统。

长久以来英特蒙专注于 Windows 作业环境的设备控制器。英特蒙开发与维护实时系统 RTX 产品线超过 30 年，提供 Windows 实时扩充组件。RTX 使用控制与通讯板来建立设备控制器。随着处理器效能提升，以及新以太网现场总线协议的出现，客户持续地要求软件协议和软件控制逻辑。因此，英特蒙开发出 KINGSTAR 自动化软件平台，以建构智能设备控制器。

KINGSTAR 包含以下五个元素：

- **KINGSTAR Fieldbus** (实时 EtherCAT®主站)
- **KINGSTAR Motion** (运动控制)
- **KINGSTAR PLC** (软件可程序化逻辑控制器)
- **KINGSTAR Vision** (实时视觉解决方案)
- **KINGSTAR IoT** (物联网平台)

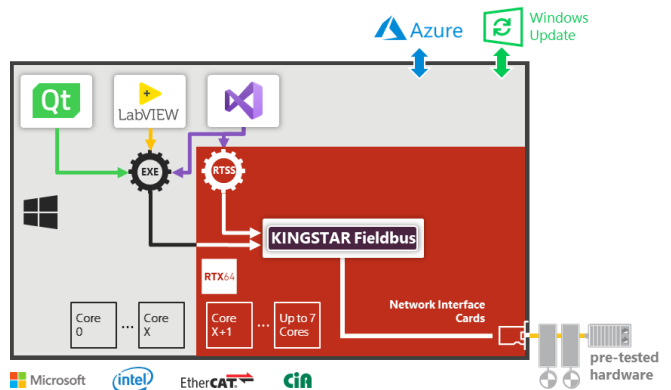


## 电脑实时控制

第一个重要层面为硬件平台。如前述，安装世界级操作系统的标准工控机带来许多优势。

KINGSTAR 于 64 位的实时扩充组件 RTX64 上运作，这是 KINGSTAR 平台的关键组件，将 Windows 转变成实时操作系统。

以 64 位 Windows 10 为运作平台。RTX64 支持在 Visual Studio 中以 C/C++ 开发实时应用程序，可广泛适用各种通用电脑，并部署于多种不同的产业，例如自动化设备、机器人、医疗、国防与仿真器。



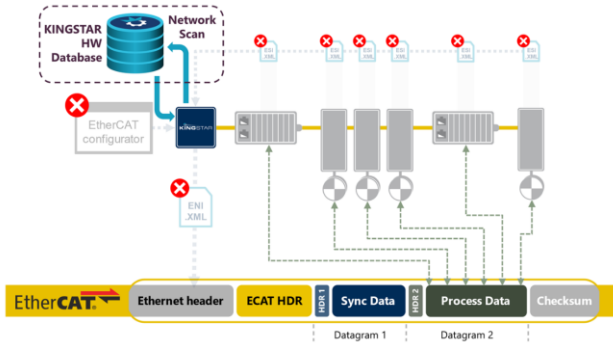
KINGSTAR Fieldbus 架构

## EtherCAT 自动扫描与设定

KINGSTAR 现场总线于 RTX64 上建置「即插即用」的 EtherCAT® 通讯协议。如我们的白皮书所述，在比较市面上最重要的 5 个现场总线后，英特蒙相信 EtherCAT® 为最佳的设备自动化协议，所以 KINGSTAR 以此为基础。为了赋予应用程序更多弹性，



英特蒙利用 EtherCAT®的总线扫描能力建立自动设定功能，相同的应用程序即可在不同的硬件配置上运作。自动扫描与设定的主要优势在于支持所有 EtherCAT 伺服驱动及 I/O 模块的硬件品牌。此外，现场总线层可让存取变量如本地存取般直接，从应用程序端完全隐藏。

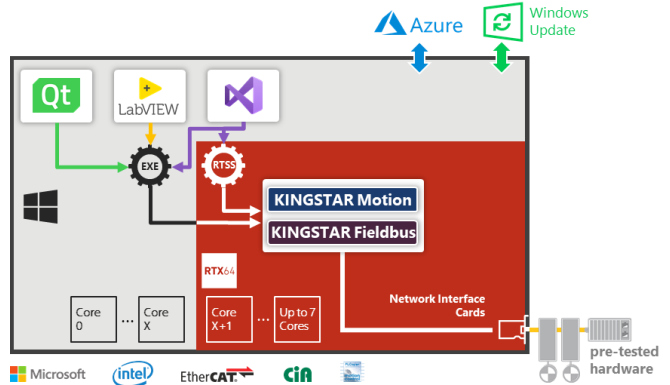


KINGSTAR Fieldbus 自动配置功能

## 工业标准：EtherCAT、CAN DS402、PLCopen、OPC UA

为了使智能设备控制平台更加完善，KINGSTAR 也提供软件运动控制函数。KINGSTAR Motion 依循 PLCopen 所定义的运动控制标准规范，包含点对点、同步、群组动作、插补以及运动学。藉由现今处理器及 KINGSTAR 中经过优化的运动方程式，可控制大量快速循环的轴。例如，应用程序可使用 20 轴每循环 125µs，或 60 轴每循环 500µs。每个轴可使用不同品牌的硬件，并且拥有自己的控制模式。与驱动器的通讯以循环同步模式为基础，在控制器里进行插补，但是 PID 可置于控制器或驱动器中。运动算法可在轴动作时修改配置文件。同步可支持电子凸轮、齿轮、线

性、圆形与螺旋型移动的群组动作。因为 CAM 或齿轮主轴可能有多个从站装置，或本身为虚拟轴甚至为其他轴的从站装置，这些功能都让 KINGSTAR Motion 功能具有高度弹性。

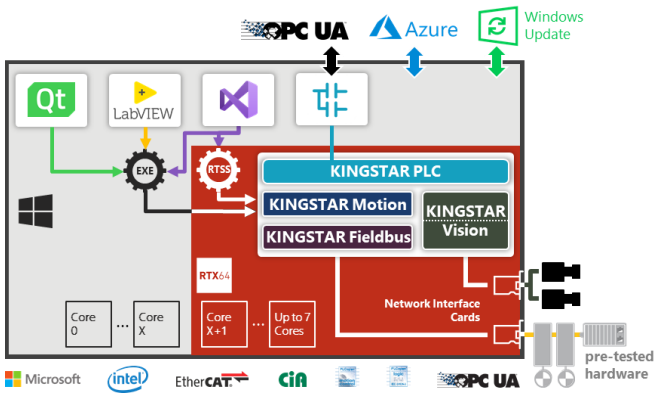


KINGSTAR Motion 架构

第三个组成组件为 KINGSTAR PLC，开放且易于使用的实时操作系统 RTOS – 英特蒙 RTX64，提供一个功能完善并集成的软件 PLC。KINGSTAR PLC 也包含运动控制和机器视觉的扩充或第三方组件，提供 C++ 程序设计师及非开发者功能丰富的用户接口以进行管理。

KINGSTAR Vision 为一套实时 GigE Vision®图像处理，让客户在 Windows 工控机上透过 OpenCV（开放函式库）开发以视觉为导向的运动控制。KINGSTAR Vision 是一套综合的软件工具，目的是为了在 GigE Vision®与其他摄影机接口上开发机器视觉、影像分析及医疗成像软件应用程序。包含行程中各步骤所使用的工具，从评估应用程序的可行性开始、到原型制作，再到开发以及最终部署。





包含 KINGSTAR Vision 的 KINGSTAR PLC 架构

最后同样重要的是，Windows 工控机的 KINGSTAR IoT 将物联网功能透过最开放的设备自动化软件平台添加到设备控制上。更多有关此主题的信息、请参阅 KINGSTAR 官网 <https://kingstar.com/cn> 上的数份白皮书，尤其是**前进工业 4.0：智能机器自动化四要素**，特别针对本议题进行深度探讨。

## 客制化平台

除了模块化以外，另一个重要的市场需求为客制化控制器或设备的能力。我们已在另一份白皮书与**线上研讨会**探究这个议题。部分技术需求与模块化相当类似，设备自动化软件平台必须具备开放且支持工业标准。设备与控制器愈发复杂，导致时常需要多个不同技能的团队参与功能开发。为了开发一个在控制器核心运作的特定运动算法，时常需要实时技术与 C/C++ 程序语言能力；为了操作员客制图形用户界面时，必须在 PLC 程序环境运用 HMI 技术。又或者其他程序语言和环境也许是更好的选项，例如以 .NET 建立系统集成商的用户接口。

## 结论

KINGSTAR 设计为考虑开放工业标准及市场需要，模块化将会是控制器与设备制造商最重要的需求。其他如客制化、维护、远程协助、或安全性等需求同等强劲，并且已在前述其他白皮书或在线研讨会探讨，又或者将被纳入未来白皮书。KINGSTAR 软件平台另一项值得关注的特点为其迎接工业 4.0 时代的能力。多数设备及控制器制造商需要在他们的设计中考虑工业 4.0，或将其纳入规画。就像先前强调的，考虑下一代系统可能为内建 IoT 服务的控制器或设备，支持以工控机为基础的控制器、Windows 10 操作系统、OPC UA、Azure 等标准，是为次世代系统做准备的**最佳方法**。