GETTING STARTED with Industrial Edge Computing

- The Essential Evolution of Industrial Process Control
- Should I Use an Edge Controller, PLC, or PAC?
- From Edge Controller to General-Purpose Computing Platform

AUTOMATION.COM | ISA A subsidiary of the International Society of Automation

EMERSON™

Table of Contents

PROCESS CONTROL EVOLVES TOWARD THE EDGE

Introduction

THE EVOLUTION OF PROCESS CONTROL

FROM PLCs AND PACs TO EDGE AUTOMATION

To successfully implement Edge technology, start at the beginning. From PLCs and PACs to edge automation, beginning the digital transformation journey begins with knowing the best technologies to apply. As edge computing evolves, adopting these basic tools and strategies can future-proof your industrial control system.

Subject matter experts from Emerson's Machine Automation Solutions can provide a range of resources to help end users understand and implement big data solutions that use IIoT and edge computing technologies to deliver operational improvements to industrial manufacturing applications. This ebook represents a fraction of the resources available. Find out more at:

Industrial Edge Computing & Control Solutions

https://www.emerson.com/en-us/automation/control-and-safety-systems/programmable-automation-control-systems/industrial-edge-software-solutions

Edge/IIoT for Manufacturing

https://www.emerson.com/en-us/automation/control-and-safety-systems/edge-iiot-manufacturing

The Essential Evolution of Industrial Process Control

As IIoT capabilities become more common, get ready to embrace edge automation.

BY JACK SMITH, AUTOMATION.COM

Not long after relay-based control gave way to programmable logic controllers (PLCs), the PLC began sharing the industrial process control (IPC) landscape with programmable automation controllers (PACs). Now edge controllers are on the scene, combining industrial-grade operational technology (OT) control with information technology (IT)-type computing power in one compact form factor tough enough to survive industrial locations. Due to this innovation, IPC is evolving again. Technology advances and Industry 4.0 initiatives, including the Industrial Internet of Things (IIoT), are making edge automation a reality.

The Essential Evolution of Industrial Process Control

Should I Use an Edge Controller, PLC, or PAC?

From Edge Controller to General-Purpose Computing Platform

Programmable logic controllers

PLC is the original ruggedized industrial computer adapted for controlling manufacturing processes such as assembly lines, machines, robotic devices, or any activity that requires high reliability, ease of programming, and process fault diagnostics. Dick Morley, considered to be the father of the PLC, invented the Modicon 084 for General Motors in 1968.

PLCs were first developed to provide automobile manufacturers with flexible, rugged, and easily programmable controllers to replace hard-wired relay logic systems. Since then, they have been widely adopted as high-reliability automation controllers suitable for harsh environments. A PLC is an example of a hard real-time system, since output results must be produced in response to input conditions within a limited time, otherwise unintended operation will result.

PLCs can range from small modular devices with tens of inputs and outputs (I/O) in an enclosure integral with the processor to large rack-mounted modular devices with thousands of I/O points, and which are often networked to other PLCs and/or supervisory control and data acquisition (SCADA) systems. They can be designed for many arrangements of digital and analog I/O, extended temperature ranges, immunity to electrical noise, and resistance to vibration and impact. Programs to control machine operation are typically stored in battery-backed-up memory or non-volatile memory.

A PLC is an example of a hard real-time system, since output results must be produced in response to input conditions within a limited time.

PLC hardware architectures consist of a central processing unit (CPU), operating memory, program memory, analog to digital (A/D) conversion, digital to analog (D/A) conversion, internal data bus, and a ruggedized form factor. The CPU of the PLC performs control calculations. The A/D and D/A converters interface with process values, making analog device information digital for the CPU and I/O to process, and digital information readable for analog output devices. The internal data bus allows data exchange between the sections of the PLC. The ruggedized package protects all of this from harsh environments.

While traditional PLCs are far from obsolete, only a few are part of a larger portfolio because the hardware and software are not easily extendable. Manufacturers today want scalable solutions they can use regardless of machine or application size, so they can proceed with minimal learning and effort to create robust automation projects. In contrast, edge-ready compact PLC controllers (figure 1) have enough I/O to handle common needs. They can be programmed for local control, monitoring, or both. They can perform data pre-processing such as incrementing a running counter of how many parts are produced based on a discrete input. Using an onboard Ethernet port, these compact PLCs can transmit information to higher-level edge-enabled devices.

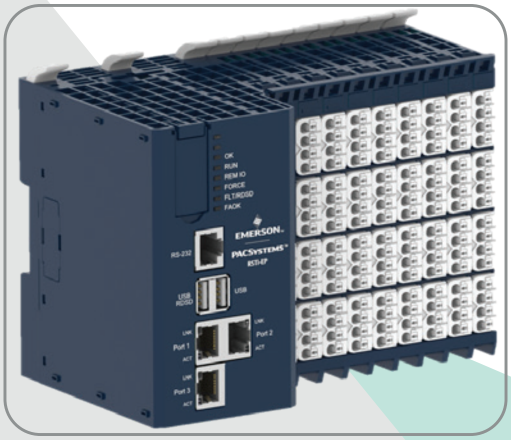


Figure 1. Compact PLCs with large PLC features like secure OPC UA communications over 1 Gb Ethernet ensure the entire portfolio is ready for IIoT connectivity. Courtesy: Emerson

The Essential Evolution of Industrial Process Control

Should I Use an Edge Controller, PLC, or PAC?

From Edge Controller to General-Purpose Computing Platform

Programmable automation controllers

PACs are used in motion control, machine control, machine vision, and general-purpose industrial control applications. They are functionally like PLCs but provide advanced software features found in PCs. PACs are used in logic, motion, process control, and vision systems that require data logging, signal processing, and other advanced software capabilities. According to ARC Advisory Group, the company that coined the PAC acronym (attributed to Craig Resnick), PACs have five main features:

1. Multi-domain functionality: logic, motion, and continuous control systems can be designed within the same hardware platform.

2. A single, multi-discipline development platform with data tags in a common database.

3. Flexible software tools that maximize process flow across machines or process units, which means one tool for every programming task—control logic, motion control, human-machine interface (HMI) design for more than one machine in a process.

4. Open, modular architecture that lets engineers use only the equipment they need.

5. Compatibility with enterprise networks, and use of many communication standards and programming languages (IEC 61131-3 and higher-level languages).

According to ARC, PACs comprise nearly two-thirds of the $12 billion PLC market, and this segment is growing faster than traditional PLCs.

One reason the PAC is popular: when a vendor launches a new product, it is more likely to be a PAC than a traditional PLC. Machine builders and end users prefer to buy integrated solutions instead of spending time configuring dissimilar systems. Most large and many small controllers qualify as PACs. Only low-cost micro and nano controllers are still considered to be PLCs.

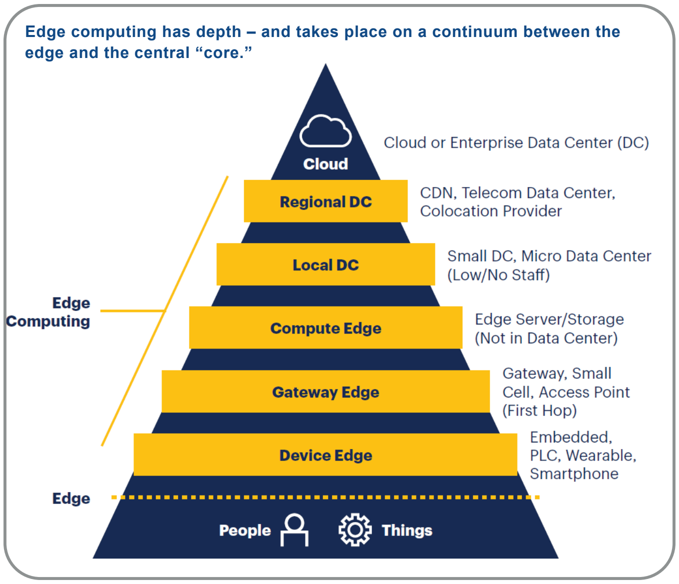


Figure 2. This graphic defines edge computing, identifies the drivers for it, and reveals how it is being used. Source: Gartner

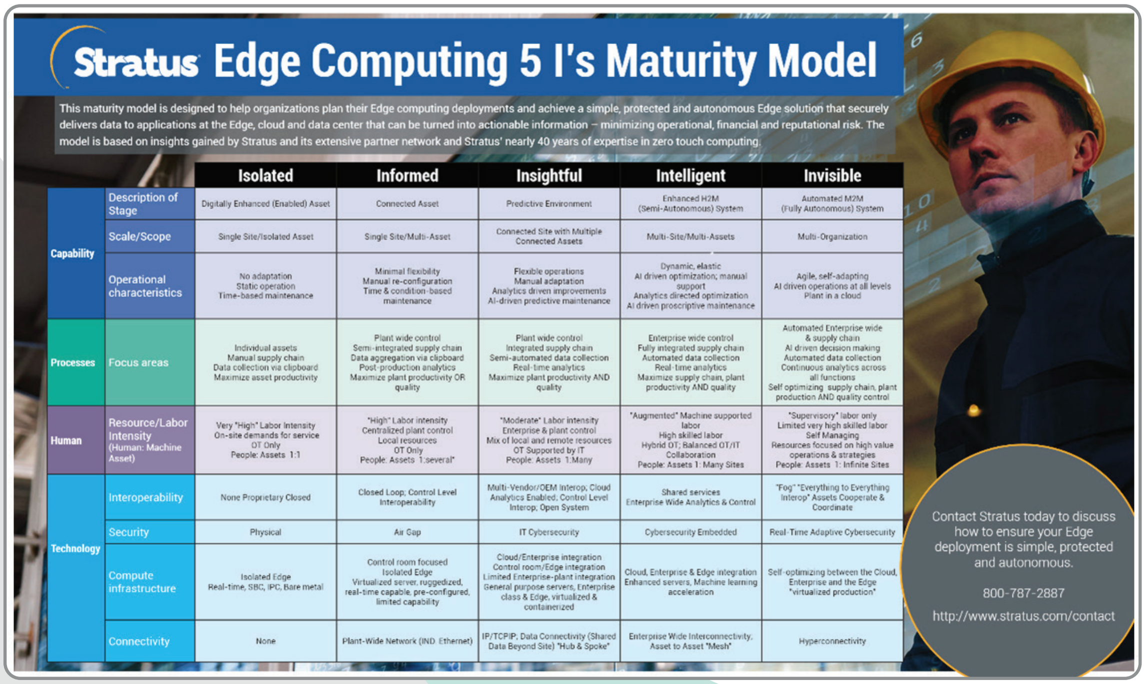
Edge controllers

There is a lot of talk about edge computing today. Says Gartner, "One of the problems in the market is having a sound understanding [of] what edge computing really is. Today, there are many emerging use cases for edge computing and not one, or even a few, predominant use cases and/or patterns for edge yet."

To help create a foundational understanding of edge computing, Figure 2 from Gartner defines edge computing, identifies the drivers for it, and reveals how it is being used. According to Gartner, this information is essential to build a foundation for leveraging what promises to be a growing style of computing that will have a major impact on how infrastructure and applications are deployed in an ever-evolving digital world.

According to Deloitte, edge computing is often confused with cloud computing, but they are very different and do very different jobs. “The specific features of edge computing make ‘edge-ops’ solutions unique to each situation.” And according to Stratus, edge computing solutions in industrial settings are poised for rapid growth and innovation, driven by the need for real-time insights and localized action (figure 3).

Edge computing combines a machine’s control and computing hardware into one platform, either with a controller that has a built-in computer or with a computing module that sits on the same rack as the controller. This two-in-one approach allows users to combine digital content (custom code, the controller’s human-machine interface [HMI] application, and third-party software programs) where the controller resides instead of another location. This creates inherent benefits for end users including space savings and access to data at its source. It also creates new opportunities for users to build new solutions for production applications.


Stratus Edge Computing 5 I's Maturity Model

This maturity model is designed to help organizations plan their Edge computing deployments and achieve a simple, protected and autonomous Edge solution that securely delivers data to applications at the Edge, cloud and data center that can be turned into actionable information – minimizing operational, financial and reputational risk. The model is based on insights gained by Stratus and its extensive partner network and Stratus' nearly 40 years of expertise in zero touch computing.

		Isolated	Informed	Insightful	Intelligent	Invisible
Capability	Description of Stage	Digitally Enhanced (Enabled) Asset	Connected Asset	Predictive Environment	Enhanced H2M (Semi-Autonomous) System	Automated M2M (Fully Autonomous) System
	Scale/Scope	Single Site/Isolated Asset	Single Site/Multi-Asset	Connected Site with Multiple Connected Assets	Multi-Site/Multi-Assets	Multi-Organization
	Operational characteristics	No adaptation Static operation Time-based maintenance	Minimal flexibility Manual re-configuration Time & condition-based maintenance	Flexible operations Manual adaptation Analytics driven improvements AI-driven predictive maintenance	Dynamic, elastic AI driven optimization; manual support Analytics directed optimization AI driven proscriptive maintenance	Agile, self-adapting AI driven operations at all levels Plant in a cloud
Processes	Focus areas	Individual assets Manual supply chain Data collection via clipboard Maximize asset productivity	Plant wide control Semi-integrated supply chain Data aggregation via clipboard Post-production analytics Maximize plant productivity OR quality	Plant wide control Integrated supply chain Semi-automated data collection Real-time analytics Maximize plant productivity AND quality	Enterprise wide control Fully integrated supply chain Automated data collection Real-time analytics Maximize supply chain, plant productivity AND quality	Automated Enterprise wide & supply chain AI driven decision making Automated data collection Continuous analytics across all functions Self optimizing supply chain, plant production AND quality control
Human	Resource/Labor Intensity (Human: Machine Asset)	Very "High" Labor Intensity On-site demands for service OT Only People: Assets 1:1	"High" Labor intensity Centralized plant control Local resources OT Only People: Assets 1:several"	"Moderate" Labor intensity Enterprise & plant control Mix of local and remote resources OT Supported by IT People: Assets 1:Many	"Augmented" Machine supported labor High skilled labor Hybrid OT; Balanced OT/IT Collaboration People: Assets 1: Many Sites	"Supervisory" labor only Limited very high skilled labor Self Managing Resources focused on high value operations & strategies People: Assets 1: Infinite Sites
Technology	Interoperability	None Proprietary Closed	Closed Loop; Control Level Interoperability	Multi-Vendor/OEM Interop; Cloud Analytics Enabled; Control Level Interop; Open System	Shared services Enterprise Wide Analytics & Control	"Fog" "Everything to Everything Interop" Assets Cooperate & Coordinate
	Security	Physical	Air Gap	IT Cybersecurity	Cybersecurity Embedded	Real-Time Adaptive Cybersecurity
	Compute infrastructure	Isolated Edge Real-time, SBC, IPC, Bare metal	Control room focused Isolated Edge Virtualized server, ruggedized, real-time capable, pre-configured, limited capability	Cloud/Enterprise integration Control room/Edge integration Limited Enterprise-plant integration General purpose servers, Enterprise class & Edge, virtualized & containerized	Cloud, Enterprise & Edge integration Enhanced servers, Machine learning acceleration	Self-optimizing between the Cloud, Enterprise and the Edge "virtualized production"
	Connectivity	None	Plant-Wide Network (IND Ethernet)	IP/TCPIP; Data Connectivity (Shared Data Beyond Site) "Hub & Spoke"	Enterprise Wide Interconnectivity, Asset to Asset "Mesh"	Hyperconnectivity

Contact Stratus today to discuss how to ensure your Edge deployment is simple, protected and autonomous.

800-787-2887

http://www.stratus.com/contact

Figure 3. The Stratus Edge Computing Maturity Model identifies edge computing characteristics. Source: Stratus

The Essential Evolution of Industrial Process Control
Should I Use an Edge Controller, PLC, or PAC?
From Edge Controller to General-Purpose Computing Platform

Said another way, true edge controllers combine PLC and PAC functionality with general-purpose computing to deliver responsive edge analytics and real-time reaction to insights. Edge technologies can include hardware and software located near machines and sensors where data is sourced. They can perform computation or control optimization tasks and transmit data to higher level and cloud-based systems. Accessing available data and acting on the resulting information is necessary for improving operations.

Achieving industrial-grade robustness while delivering advanced computational capabilities is a challenge. In addition to enabling streamlined data flow to the cloud, edge controller capabilities involve obtaining IT and computing capacity and moving it into the OT space where actual control is performed. Edge controllers also have the ability to generate analytics at the edge.

Achieving industrial-grade robustness while delivering advanced computational capabilities is a challenge.

Most IPC scenarios demand something more than general-purpose computing solutions geared toward non-control applications. IPC projects have used PLCs—and more recently PACs—to provide deterministic control. Many of these deployments have used both PLCs and PACs for more than 15 years.

Edge controller alternatives

PLCs and PACs lack the processing power, memory, and storage needed to process modern analytics or visualization applications and are therefore limited when it comes to providing general-purpose edge computing. Industrial PCs also can provide the general-purpose functionality and performance aspect.

Combining these functionalities would be ideal. However, hardware-independent designs can't guarantee the necessary performance across the deterministic and non-deterministic applications. Only software-defined designs implemented on validated hardware can provide the performance required for mission critical operations, while enabling analytics and machine learning to work in parallel.

"Hardware independent" edge implementations involve software intended to run on any industrial hardware platform, which can be convenient but typically presents risks and sacrifices. Downtime is usually necessary for certain upgrades. Hardware-independence is typically a consumer-grade version, although it's sometimes offered for industrial applications. "Software-defined" edge implementations are designed to deliver the deterministic performance necessary for reliable, repeatable, and safe control/computing, which is necessary for industrial control applications, but often requires tailored hardware.

Edge controllers feature multiple cores managed through carefully designed virtual machines (VMs) (figure 4). A real-time operating system (RTOS) supports highly reliable, deterministic runtime control, like traditional PLCs or PACs. The addition of a second onboard general-purpose operating system (OS) running a variation of Linux is the notable upgrade.

Figure 4. Edge controllers feature multiple cores managed through carefully designed virtual machines (VMs) and provide a unique combination of deterministic control like a PLC or PAC, and independent general-purpose processing like a PC.

For performance and security reasons, the most capable edge controllers use VMs to ensure the deterministic controller runs independent from the edge computing tasks. Segregation of OT and IT environments is provided at hardware and software levels.

Best of all worlds

An edge controller can be used as a PLC or PAC—even to the extent of using the same logic libraries developed for PLC/PAC members of the same product family. If the edge computing part of the edge controllers uses an open-standard OS like Linux, future users are free to add almost any type of application. The immediate benefits are more compelling if the edge controller is available with a carefully chosen set of open-source and commercial applications congruent to common industrial computing needs (figure 5).

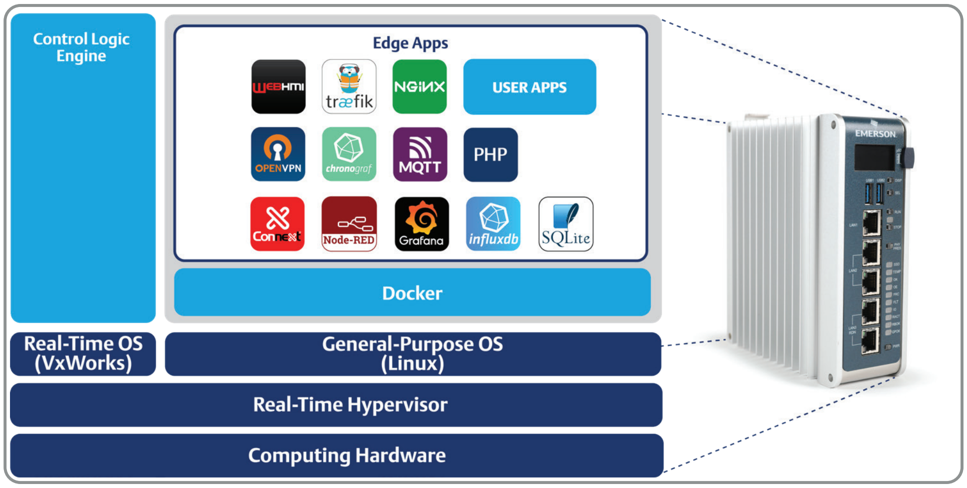


Figure 5. Some edge controllers use Linux for the edge computing platform, so they can run a wide variety of user applications. Users can take advantage of preloaded industrial edge platform software, providing the apps needed to help manufacturers accelerate digital transformation projects.

No additional hardware is needed to run visualization, data connectivity, and analytical applications when the user is ready to take advantage of these functions. Modern languages can be used where they are best suited, offering end users the flexibility to extend their application with edge capabilities.

End users may want to test the IIoT waters but can't jeopardize production. Using an edge controller for IT computing capabilities and OT connectivity is a practical approach. Edge controllers can be installed anywhere on the factory floor, and one edge controller can be networked to many PLCs and/or PACs, sensors, or other instruments to act as a gateway and information concentrator.

Users can configure an edge controller to gather data, provide visualization hosting, and preprocess the data as needed (filtering, averaging, or even executing local analytics), and to forward the information to higher level systems. They can scale this IIoT functionality at their own pace, without impacting their existing automation systems.

Maximum advantage is realized by combining both control and computing in support of one application. A deterministic "inner loop" accesses field-sourced data and controls devices while an edge compute "outer loop" combines deterministic data and real-time information from outside sources or analytics. The outer loop functions advise the inner loop on operating parameters and tuning to achieve optimum efficiency.

Looking ahead

It is now possible to update or replace PLCs and PACs with edge controllers to address current challenges. Edge controllers can offer real advantages over traditional technology by implementing carefully designed OSs for addressing OT and IT needs regarding control, communications, and application development.

ABOUT THE AUTHOR

Jack Smith (jsmith@automation.com) is a contributing editor for Automation.com and ISA's *InTech* magazine. He spent more than 20 years working in industry—from electrical power generation to instrumentation and control, to automation and electronic communications—and has been a trade journalist for 22 years.

Should I Use an Edge Controller, PLC, or PAC?

Edge control technologies excel with familiar PLC and PAC applications while delivering advanced computing and communication options.

BY DARRELL HALTERMAN

For commercial and industrial computing products, software and hardware development progress proceeds in tandem fashion, with the lead alternating at times. Sometimes the software complexity and features increase in a way that bumps into processing limitations, while other times hardware advances unleash newfound capacity for more sophisticated software.

EMERSON

It is easy to look at today's traditional operational technology (OT) industrial controller options, most often represented by traditional programmable logic controllers (PLCs) and process automation controllers (PACs) and see them as mature technologies with capable software and sufficiently fast hardware. The challenge lies in identifying what comes next.

A few industry trends are pointing the way. Modern consumer and commercial computing experiences are ripe for merging into industrial products. Surging numbers of Internet of Things (IoT) devices are becoming commonplace, and many users are looking at how to incorporate Industrial IoT (IIoT) devices into their automation systems. Digital transformation requires connecting with many data sources, collecting and storing that data, visualizing and analyzing it, enabling optimized operations.

The journey to realize value in these trends calls for something new: an edge controller that is more than just a PLC or a PAC. Edge controllers are making it possible for end users to continue implementing robust automation systems as they always have, but now also seamlessly take advantage of the latest communication and application development options.

Edge controllers can upgrade existing systems, create new designs, improve productivity, address skill gaps, and enhance security.

Edge controller hardware and software

On the surface, observers and implementers of industrial technology would be excused for believing the industrial controller world has plateaued in processing power and physical footprint. This is somewhat true regarding simple processor clock speed.

However, cutting-edge commercial and industrial processors have realized substantial processing performance gains due to multi-core designs and increasing numbers of processor cores in a single device. At the same time, hardware-level virtualization provides a convenient way to manage multiple cores and assign them to virtual machines (VMs).

PCs and servers used in conjunction with industrial controllers already benefit from this improved computing performance coming from the fast-paced commercial world.

Deterministic industrial control systems, on the other hand, require high speed, repeatable, and rigorously time-based processor execution. Unlike many commercial applications, any industrial application of VMs and multi-core management must be free from jitter or any other deviations from reliable timely operation.

Realizing equivalent performance gains for industrial applications requires a new class of controller, and the edge controller was born (figure 1). These controllers feature multiple cores managed through carefully crafted VMs. A real-time operating system (RTOS) supports highly reliable, deterministic runtime control, similar to traditional PLC or PAC. The real upgrade is the addition of a second on-board general-purpose operating system (OS) running a variant of Linux.

Figure 1. Edge controllers, such as those offered by Emerson, provide a unique combination of deterministic control like a PLC or PAC, and independent general-purpose processing like a PC.

The two OSs are independent at the hardware level, but able to securely communicate with each other via OPC UA. The "open" general-purpose OS can even be rebooted independently without affecting the deterministic runtime.

Combining a deterministic runtime with a general-purpose OS is an advancement that builds on traditional control methods by enabling closely integrated deterministic computing and secure real-time data access technologies.

What makes edge controllers different?

Conventional PLCs and PACs will continue to be available for many years, although today, many users find an overlap in these product descriptions. The term PAC has been used to describe controllers with more advanced control strategies and communication capabilities than a PLC, but in fact, many PLCs have been creeping into PAC capability territory. PLCs and PACs are a good technology developed during a different era, but it is now possible to update or replace them with edge controllers to address current challenges. Edge controllers can offer real advantages over traditional technology by implementing carefully designed OSs for addressing OT and IT needs with regards to control, communications, and application development.

As dedicated control devices with a specific functional scope, basic PLCs are often “bare metal” designs or use an extremely limited and proprietary OS. PACs gained more advanced communication and functional services, while preserving a deterministic runtime using an RTOS. For creating an edge controller that adds at least one general-purpose OS in addition to the deterministic runtime, designers must address new considerations. Hosting all these functions in a single OS puts deterministic performance at risk. Incorporating a bare-metal virtualization hypervisor, however, enables deploying a rock-solid deterministic runtime securely and independently in parallel with a general-purpose OS on the same platform (figure 2). This foundation is crucial.

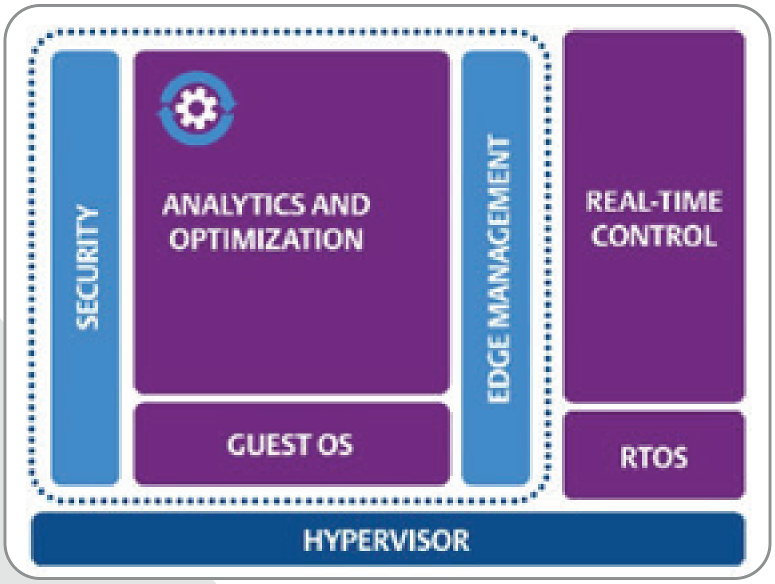


Figure 2. A bare metal virtualization hypervisor design as used with Emerson edge controllers delivers rock-solid deterministic control in parallel with an independent general-purpose OS.

Original PLCs were largely isolated, or they communicated with slow and cumbersome serial links. PACs included better industrial protocol implementation and Ethernet for improved connectivity and interoperability. Edge controllers deliver these advantages but must take things to a whole new level to perform in an increasingly IT-connected environment. They should incorporate a management umbrella adding security and defenses suitable for prevalent IT-like issues like network storms and Denial of Service (DoS) attacks. In conjunction, IT-oriented protocols with built-in security such as OPC UA, MQTT, and secure sockets (HTTPS, SSL, FTPS) can provide appropriately secure communications performance.

For application development, PLCs used mostly proprietary ladder logic and relatively rudimentary tools. Taking on some concepts from the greater software industry, PACs may offer a level of support for standard IEC-61131 languages, custom user code blocks, and some basic capabilities for code reuse and object-oriented design. A well-designed edge controller preserves the deterministic runtime, and adds a computing environment for performing analytics, data aggregation, and other advanced tasks. The general-purpose OS allows application development in IT-oriented languages like C and C++, Python, and Java. The most flexible architectures can allow OT personnel to focus on the deterministic system, IT personnel can work on the general-purpose system, and the two groups can coordinate and crossover as needed or desired in a clearly defined manner.

Edge controllers are making it possible for end users to continue implementing robust automation systems, and also take advantage of the latest communication and application-development options.

How to move to edge control

When users only need a PLC/PAC feature set, they can still safely and economically specify an edge controller to fill that role. Standard IEC-61131-compliant programming methods can be used, or even commercial languages like C. These users never need to access the general-purpose OS features if they choose not to do so.

Those who are interested, and perhaps more tech-savvy, should look for edge controllers offering a native toolbox of IT-friendly features such as Linux OS, a Python interpreter, secure sockets, an embedded database, and OPC UA support. Part of the reason IT-friendliness is so important is because much today's IIoT technology is built on Linux. Edge controllers provide an environment where users can safely and securely explore their options while preserving reliable deterministic operation.

With the deterministic and the general-purpose OS on one hardware platform, it is easier to securely coordinate the two, compared to deploying separated devices. The general-purpose OS has access to wider variety of information sources and high-fidelity data, both from the deterministic system and IIoT devices, and it can perform local analytics and advanced algorithm execution. The results can be used immediately to advise the deterministic system for optimal operation. In addition, the general-purpose OS is ideal for transforming raw data into pre-processed information for securely sharing with higher level systems.

Controller usability

Modern edge controller technology can benefit those specifying, programming, and maintaining automation systems. At design time, users can select from product families sized to meet the need from small to large based on a performance and price standpoint, all using the same deterministic runtime environments, the same toolchains, and configured with the same integrated development environment. Users only need to learn one common environment. Their code and configurations can readily be reused.

Another consideration is how end user demographics are changing. Experienced users already in the workforce are familiar with PLC and PACs. Their depth of experience affords them a more instinctive approach to programming, tuning, and troubleshooting based on the products they have used for years.

The newest software users entering the workforce have grown up in the smartphone age. They are used to modern programming languages and working in a developer kit environment, rather than developing everything from scratch. They are probably most comfortable assembling code elements from a library or repository to create a solution (figure 3).

Figure 3. Edge controllers implemented into automation systems provide a good platform for those entering the workforce to apply their coding skills and mobile computing experiences.

The latest edge control automation platforms address skill gaps in two ways. The first is that younger workers will be more comfortable and perform better in a development environment that uses tools and languages familiar from their schooling or training. The second is that edge control capabilities are a good way to create advisory algorithms to assist newer teams with operating and maintaining systems. Companies can then rely on a smaller set of experts and improve their productivity by leveraging that expertise in the form of analytics and advising algorithms to empower the rest of the team.

Cybersecurity at the edge

Edge control solutions, with their inherent connectivity and new analytic capabilities, may raise security concerns for traditional controls developers. Developers accustomed to air-gapped networks as a standard security strategy may not be comfortable dealing with the unknowns of more open networks. Even more concerning, many older automation platforms were developed before cybersecurity was considered a priority, and when security was addressed as more of an add-on afterthought than a built-in feature.

For edge controllers and the most modern PACs, security-by-design is a fundamental requirement. Suppliers of edge controller and PAC hardware and software products should:

- Validate the supply chain
- Build their devices using only trusted components and suppliers
- Incorporate hardware-based security technologies, such as Trusted Platform Module (TPM) technology
- Use digitally signed and encrypted product firmware to guarantee software integrity.

Some vendors already offer software suites to help users navigate and manage security certificates, enabling simple, secure and encrypted communications right down to the application control level.

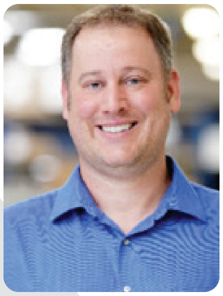
The latest automation product portfolios should provide the right tools to make strong IT-type security easy for operation technology (OT) users to apply.

Edge controller performance criteria

For industrial automation end users, the technology discussion is less about product acronyms like PLCs and PACs, and more about delivering the right performance. Some key performance criteria are:

- Achieving improved design, operating, and maintenance productivity
- Addressing human factors associated with new workers and skill gaps
- Applying the best security in an increasingly connected world.

Edge controllers are the answer for helping end users win in each of these areas by delivering a combination of the latest technology coupled with ease of use.



ABOUT THE AUTHOR

Darrell Halterman is a senior product manager of PACSystems controllers at Emerson's machine automation solutions business, and he is also responsible for the portfolio's control solutions modernization strategy. He enjoys working with customers to find the right modernization strategy to enhance their existing control solutions with the latest advances in automation.

All images courtesy of Emerson

From Edge Controller to General-Purpose Computing Platform

Future-proof controls programming by adopting these basic tools and strategies as edge computing evolves.

BY DARRELL HALTERMAN

Edge computing, new communication methods, and advanced analytics are among the hottest topics related to industrial automation, a field more often associated with stoic reliability than cutting-edge technology. Yet there remains a fundamental need for creating traditional deterministic control solutions suitable for new and retrofit automation projects.

EMERSON

Designers are often concerned about managing interactions with industrial automation devices, especially as hardware and communications capabilities rapidly evolve. Fortunately, by adopting some basic tools and strategies for abstracting application data, interactions with modern edge controllers can be nearly as easy as traditional programmable logic controllers (PLCs).

As hardwired relays yielded to PLCs, which in turn were surpassed by programmable automation controllers (PACs), the means and methods of programming control systems have adapted. The same evolution is occurring as edge controllers and other edge devices begin entering service. Controls programming during the edge evolution does not have to mean wholesale change but can instead offer more choices.

The edge is here

Most users have heard terms like “edge computing” and “digital transformation,” but many may feel unsure how edge computing can solve problems in their applications. At least, users stand to gain improved visibility into equipment and process operations, even via mobile devices. Streamlined data gathering capabilities support analytics at the edge and in the cloud, which provide novel insights for improving availability, utilization, and efficiency. In the most advanced scenarios, edge controllers can autonomously leverage analytical results to advise deterministic control applications for near-real-time responsiveness.

Even as these Industrial Internet of Things (IIoT) capabilities become more commonplace, not every developer or project is ready to commit to wholesale edge automation. By architecting a control solution properly, end users can implement classic functionality today, while future-proofing their designs to take advantage of the edge moving forward.

Developers unfamiliar with edge controllers may feel compelled to stick with tried-and-true control technologies and products. But the immediate impacts of ongoing labor and component shortages demand modernized systems that counter these challenges by taking advantage of new ways of accessing and using data. Developers and end users can embrace making their automation designs IIoT ready, and rest assured the required programming will build on past concepts.

Edge controller drone application

Exemplifying the new potential made available by edge controllers, an end user recently explored using commercial drones for inspecting remote pipeline. The user was able to quickly incorporate the standard Linux software development kit (SDK) for a popular commercial drone into the edge compute portion of the edge controller. Using standard OPC-UA communications, the company activated the drone to fly predetermined inspection routes either on default time intervals or on alarm conditions as commanded by the deterministic control portion of the edge controller. The edge compute portion captured video and telemetry data and then posted it to a secure web dashboard for remote operators to assess in real time.

Programming evolution

When discrete automation operational technology (OT) originally shifted from hardwired systems created with electromechanical relay and timer devices to electronically programmable devices like PLCs, the Ladder Diagram (LD) language was created to help transition developers and technicians to the new solution. Visually, LD programming looks like the wiring diagrams used with hardwired relay and timer systems—but it greatly simplified programming, debugging, and modifying sequencing logic with a PLC.

As PLCs gained power and began to perform more complex functions like mathematics, analog loop control, and motion control, the associated languages had to evolve for performing complex expression processing, advanced process control, and motion. Languages like Structured Text (ST) arose to meet the new programming and debugging demands, yet LD was still available.

The development of PACs represented an expansion of PLC functionality to incorporate even more advanced application processing and communication functionality. PACs began to take on more tasks we would associate with the edge today, although their dedicated real-time operating systems (RTOS) imposed some limitations. In many cases, users found it necessary to create complex algorithms, like machine learning strategies, using modern information technology (IT)–type languages like C++ and Python, running on PCs and industrial PCs (IPCs) working in conjunction with PACs.

Now edge controllers are on the scene, combining industrial-grade OT control with IT-type computing in one compact form factor tough enough to survive edge locations. Programming is evolving again.

Edge controllers provide real-time machine control using a deterministic RTOS running the kernel, integrated with a flexible general-purpose Linux-based IIoT computing environment designed to operate without interfering with the real-time controls.

Edge programming concepts

Some end users may opt to create edge solutions using a variety of hardware platforms and programming methods. This ad hoc approach introduces risk and requires time-consuming testing and integration efforts that generate late-stage changes and potential cost increases unacceptable in today's industrial projects.

Selecting hardware and software offered as part of a wide portfolio of coordinated industrial automation products from a single provider ensures compatibility. A compatible portfolio of PLCs, PACs, edge controllers, and the software they run makes it possible for users to learn just a few programming tools and apply them to many products. Harmonized library objects lead to greater consistency and rapid development; proven code is readily reused on projects; and data is more readily transferred among systems.

To ensure future-proof solutions, users should look for product portfolios that embrace open programming, software, and communication standards wherever possible. Even open standards may benefit from industrial-centric extensions where appropriate. For an edge controller, this requires:

- deterministic control engines that support well-known and stable IEC-standard and C languages
- variable programming, with rich standard and user-defined data types
- extensive standard and user-defined function blocks
- support for object-oriented communications protocols like OPC-UA that can seamlessly transmit data in context between deterministic and analytic applications
- support for OT-centric communication protocols like Modbus, Profibus, and Profinet
- Linux-based OS for the general-purpose computing part of the edge controller—bootable and upgradable independently of the deterministic control portion. Linux is lightweight and delivers high performance while using fewer CPU and RAM resources than other alternatives
- ability to run open-source apps like Node-RED and Grafana, as well as commercial and custom apps
- programmable using C/C++, Python, and other modern languages suitable for applications like machine learning and artificial intelligence
- support for IT-centric communication protocols like MQTT and HTTPS.

A true edge controller with these features is the best choice for implementing traditional deterministic control today and taking advantage of the edge computing evolution now and in the future.

Edge control and computing

Edge controllers are touted by many suppliers, but these devices vary significantly in performance and internal operation. For performance and security reasons, the most capable implementations use hardware virtualization to ensure the deterministic controller runs independently of the edge compute activities. Rigorous segregation of OT and IT environments is provided at hardware and software levels.

Today's controls designers may be hesitant to apply new edge controllers to equipment and systems usually automated with PLCs or PACs. They may be concerned about the skillset required to make an edge control solution work reliably. They may feel unsure of the benefits gained from trying a new architecture.

These reservations disappear if the deterministic portion of the edge controller runs the same kernel as PLCs and PACs from the vendor and uses the same integrated development environment toolchain and toolchest of library functions. An edge controller can be used as a PLC or PAC today, even to the extent of using the same logic libraries developed for PLC/PAC members of the same product family. There is great power in using familiar methods tied to new capabilities to obtain new results.

In the most advanced scenarios, edge controllers can autonomously leverage analytical results to advise deterministic control applications for near-real-time responsiveness.

If the edge compute portion of the edge controllers uses an open standard OS like Linux, users are free, in the future, to add almost any type of application needed. The immediate benefits are more compelling if the edge controller is available with a curated set of open source and commercial applications suiting the most common industrial computing needs.

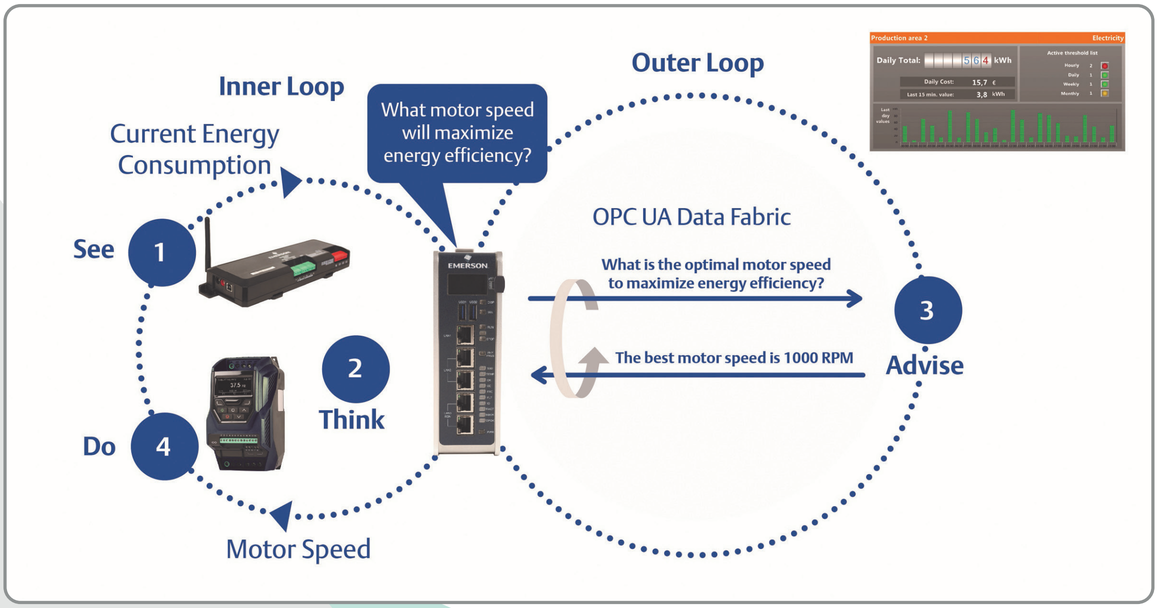
Control. The value of using an edge controller as a supercharged PLC/PAC is that the edge controller is natively ready to run visualization, data connectivity, analytical applications, and much more whenever the user is ready to take advantage of these functions. No additional hardware is needed; no cutovers or physical tie-ins are needed; and there is no requirement to interrupt execution of the automation application. Modern languages can be used where and when they are best suited for an application. End users gain the flexibility to extend their applications with edge capabilities when they are ready.

Computing. On the other hand, it is common for manufacturing facilities to have significant quantities of legacy automated equipment already in service. End users at these facilities may want to test the IIoT waters, but simply cannot jeopardize production.

In this case, using an edge controller only for the IT computing capabilities and OT connectivity, without even one line of control logic, is a practical approach. The edge controller can be installed anywhere on the factory floor, just like a PLC/PAC, and one edge controller can be networked to many PLCs/PACs, sensors, or other instruments to act as a gateway and information concentrator.

Operating in this capacity, users configure the edge controller to gather data, provide visualization hosting, and preprocess the data—such as filtering, averaging, or even executing local analytics—as needed, and to forward the information to higher level systems. Users can add and scale up this IIoT functionality at their own pace, without affecting any underlying automation systems.

Control and computing. Although the individual control and computing capabilities of edge controllers are important, users gain maximum advantage by combining them in support of one application. The deterministic "inner loop" directly accesses field-sourced data and controls devices. The edge compute "outer loop" combines deterministic data and real-time information from outside sources or analytics to advise the inner loop on operating parameters and tuning to achieve optimal efficiency (see below).



Edge controllers combine OT and IT operations, using programming methods optimized for integrating each role.

The Essential Evolution of Industrial Process Control
Should I Use an Edge Controller, PLC, or PAC?
From Edge Controller to General-Purpose Computing Platform

Looking ahead

Edge programming is more evolution than revolution. Originally, PLCs performed specific jobs and were programmed using correspondingly specialized methods. As PLCs improved and progressed to become PACs, the programming means and methods preserved the best aspects of the original Ladder Logic roots, but gained new languages, function blocks, and organizational methods as needed in support of added capabilities.

Edge controllers represent an exponential capability increase over traditional PLCs and PACs, so it makes sense that even more languages and applications are being added. This does not mean end users need to discard their experience or embrace all the added capabilities at once.

Instead, well-designed edge controllers remain highly compatible with existing PLC and PAC ecosystems, so users can continue to build on their OT knowledge base. When they are ready to extend applications into the IT realm, the same edge controller provides a general-purpose computing platform, effectively future-proofing their application.

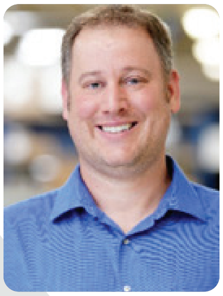
Edge controllers represent an exponential capability increase over traditional PLCs and PACs

RESOURCES

"Programming standards improve automation and controls"
https://www.isa.org/intech-home/2016/september-october/features/programming-standards-improve-automation-controls

"Better performance begins at the edge"
https://www.isa.org/intech-home/2020/january-february/features/better-performance-begins-at-the-edge

"Solving big data problems with a little data approach"
https://www.isa.org/intech-home/2020/march-april/columns/solving-big-data-problems-with-a-little-data-appro



ABOUT THE AUTHOR

Darrell Halterman is a senior product manager of PACSystems controllers at Emerson's machine automation solutions business, and he is also responsible for the portfolio's control solutions modernization strategy. He enjoys working with customers to find the right modernization strategy to enhance their existing control solutions with the latest advances in automation.