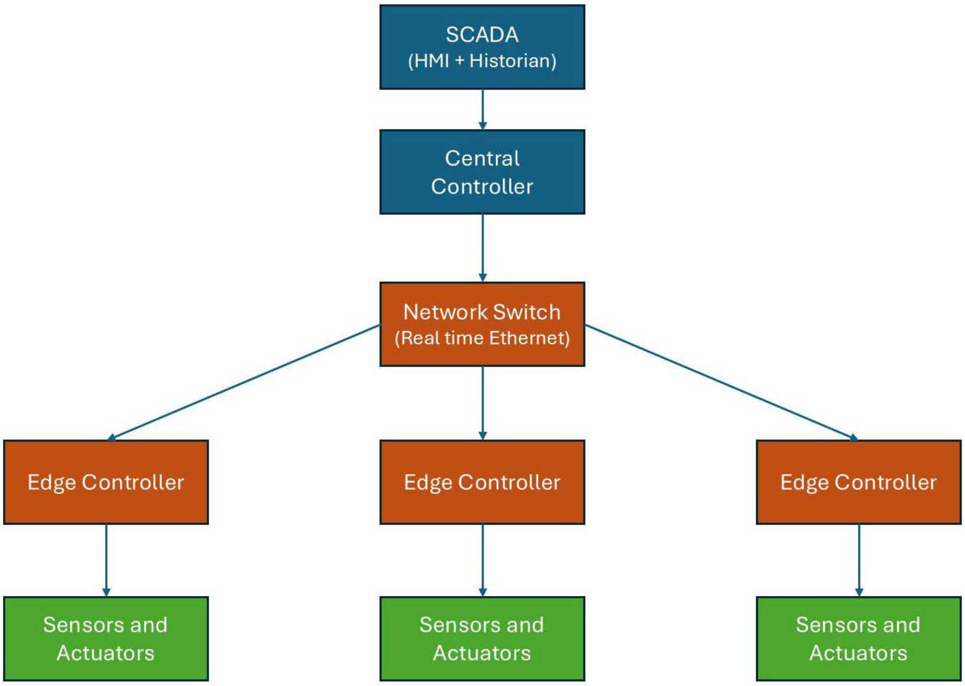
Electronic Design®

ALAN R. EARLS, Contributing Editor

Distributed Control: Finding the Right Granularity for Your Environment

Organizations are looking closely at the potential advantages of implementing distributed process control, which can provide greater flexibility and adaptability to organizations. Learn about the why and how of this approach.



In this simplified visual representation of a distributed process control system architecture, a central SCADA system interfaces with the central controller and a real-time Ethernet switch distributes control tasks to local edge controllers. Each edge controller independently manages its own sensors and actuators, enabling modular, scalable, and potentially fault-tolerant control.

Emerging electronic control challenges in process industries revolve around the increasing complexity and connectivity of systems, alongside the need for greater efficiency, reliability, and security. These challenges are driven by trends like the Internet of Things (IoT), artificial intelligence (AI), and cybersecurity threats, all of which impact how processes are controlled and managed.

One of the resulting trends is a growing emphasis on distributed process control. Process industries are moving toward more distributed and interconnected control systems, often integrating with IoT devices and cloud platforms *(see figure)*. This requires managing data from numerous sources and ensuring seamless communication between different parts of the system.

To build or convert from centralized to a distributed process control system (DPCS), electronic engineers must go beyond traditional PLC or centralized architectures. They need to dive into potentially mastering system partitioning, real-time networking, edge computing, synchronization, and fail-safe design. Below are some suggestions on how to start.

What Do I Need to Know About the Fundamentals of Distributed Control?

Particularly in complex or large-scale industrial environments, adopting distributed process control can offer significant advantages over centralized systems. Among them are improved reliability and fault tolerance as well as scalability.

Localized control reduces the risk of a single point of failure. If one controller or node fails, others continue functioning, increasing system uptime. In addition, distributed systems are often easier to expand — new units or subsystems can be added without overhauling the entire architecture. Plus, a more modular design supports phased upgrades and flexible production lines.

The benefits to production activities may include faster response and improved real-time control because proximity reduces communication latency, which can lead to faster and smarter, localized decision-making.

Distributed control also enables faster control loops for time-sensitive operations (e.g., motion control, chemical dosing). In process control, a control loop is a system that automatically regulates a process variable to maintain it at a desired setpoint. It achieves this by continuously monitoring the process variable, comparing it to the setpoint, and making adjustments to maintain the desired state. These loops are fundamental to ensuring process stability, optimizing performance, and maintaining consistent product quality.

Having distributed intelligence means that local controllers can make decisions independently using edge computing or local logic. This reduces central processing load and bandwidth requirements. Yet, distributed control nodes can still feed real-time data to cloud systems or digital twins, opening the door to predictive maintenance, process optimization, and remote monitoring.

The physical setup and maintenance of a control system can also benefit. With a more modularized approach, wiring and installation is usually easier. Also, using fieldbus or Ethernet-based networks reduces the amount of physical cabling, which can help simplify maintenance and lower installation costs. Distributed diagnostics can allow technicians to pinpoint and resolve issues faster because faults can often be quickly isolated to individual subsystems or nodes.

An additional benefit of distributed control is that it supports reconfiguration of production processes or automation sequences without major rewiring or reprogramming. This can be a plus in modern environments with demand for frequent product changes or customization.

However, distributed control also comes with some risks, namely, network latency, synchronization errors, and integration complexity.

How Can I Implement Distributed Control?

A good place to start is by analyzing the subsystems that can operate independently or semi-independently. Are valves operated in synchronization with a distant function, or could they be operated independently? Do motors respond to local signals? What triggers pump operations?

More broadly, you will need to define what logic should be local versus centralized and understand the integration and coordination challenges involved. For example, safety interlocks could be candidates for local control while optimization logic will likely stay global.

Start by breaking processes into zones or equipment modules. Standards such as ISA-88 and ISA-95 approved by the International Society for Automation can be helpful in framing both modular and hierarchical design.

ISA-88 focuses on batch process control, providing a framework for designing and implementing batch control systems such as recipe management, equipment control, and procedural control. ISA-95, which focuses on enterprise-control system integration, offers a framework for integrating enterprise and control systems, enabling seamless data exchange and coordination between different levels of a manufacturing organization.

With frameworks such as those and your own experience or organizational practices, you can then begin to map out process units and assign dedicated local controllers like PACs or industrial PCs. Work to group signals by physical and logical proximity using I/O allocation diagrams. An I/O diagram documents the connections and relationships between physical components and their associated I/O points, thereby helping to show how data flows and how devices interact.

What is a Distributed Communication Infrastructure and How Do You Implement It?

Real-time industrial networks are usually built on a familiar list of options, including:

- **EtherCAT**, a deterministic, and flexible industrial Ethernet protocol developed by Beckhoff Automation.
- **PROFINET IRT**, a specialized communication channel within the PROFINET protocol, designed for high-speed, deterministic motion control applications.
- **SERCOS III**, a real-time communication protocol for industrial automation that's also Ethernet-based.
- **TSN (Time-Sensitive Networking)**, which refers to a set of IEEE 802.1 standards that enhance Ethernet to provide deterministic, real-time communication.
- **IEEE 1588 (PTP)** for time-sensitive control loops is also often part of implementations.

Using these elements, work to design a deterministic communication backbone (ring, star, or mesh), selecting protocols based on factors such as acceptable latency, jitter tolerance, and vendor compatibility. Then incorporate time-sync hardware and firmware (especially for motion or power electronics) as needed)

How Can I Turn the Rough Concept into Reality?

With a concept of what you want to achieve and how it ought to be architected, you can begin to select and program distributed controllers. Many types of controllers are available, including compact PLCs, RTUs, or PACs.

There are also edge devices with local logic execution and distributed programming options such as ladder, ST, and FBD conforming to the IEC 61131-3 standard. They enable modular design through function blocks by providing a standardized way to structure and reuse code, which helps simplify complex control systems.

It's important to remember to ensure that each controller has sufficient CPU/memory for its local logic and diagnostic features. In addition, implement state machines or sequencers locally, with global coordination logic at the supervisory level, and ensure fail-safe behavior if communication to the central system is lost.

How Can I Ensure a Distributed Approach Supports Diagnostics and Maintenance?

Self-diagnostics and health monitoring are essential in distributed systems; fault propagation can actually be harder to trace if subsystems are isolated. To support diagnostic functions, work to build-in heartbeat/status communication from each node. Also, enable firmware updates and fault log retrieval over the network such as via OPC UA, a platform-independent, service-oriented architecture that allows for secure and reliable data exchange in industrial automation applications.

Moreover, it can be helpful to employ standard data models such as PackML or NAMUR NE 107 to achieve uniform diagnostic reporting. This distributed data may be integrated with centralized monitoring and SCADA because central coordination is still needed for visualization, optimization, and enterprise-level integration.

SCADA systems typically use the above-mentioned OPC UA or MQTT brokers to pull distributed data into unified dashboards. This can be made more accessible and useful by developing human-machine interface templates for each process module and mapping distributed tags to a centralized historian or MES platform to tie the components together.

What Should I Do About Security?

Distributed control doesn't mitigate the perennial need for security planning. While distributed process control offers major operational benefits, it also increases the cybersecurity attack surface because each distributed controller, sensor, actuator, and network switch becomes a potential entry point.

Furthermore, the use of standardized protocols makes systems more interoperable — but also more vulnerable. Meanwhile, the all-too-common lack of encryption or authentication on internal networks could open the door to attackers to intercept or manipulate data (man-in-the-middle attacks).

To minimize these cybersecurity risks, build in defense-in-depth at every layer — from physical hardware and firmware and individual devices to networking and data protocols. This, of course, is a large topic and a domain often subject to corporate policies and guidance. In other words, it tends to stay largely in the centralized category, but decentralization initiatives should make sure to incorporate needed best practices.

References

Distributed Control Systems Market Report (2025–2032): Orchestrating Automation for a Connected Future

Understanding DCS in Industrial Automation: What is a Distributed Control System