

InTech®

OFFICIAL PUBLICATION OF THE INTERNATIONAL SOCIETY OF AUTOMATION

FEBRUARY 2022



www.isa.org/intech

TOP INDUSTRIAL COMPANIES FIND PATHWAYS TO TRANSFORM



Digitalization Drives
Sustainability

How OPC UA Makes IIoT Possible

Integrated Data Defines the
Future of Machinery Health

What You May Not Know About
SoftPLCs

What You May Not Know About SoftPLCs



There are misconceptions about what SoftPLCs can and can't do.

By Steve Mustard, PE, Eur Ing, CAP, and Pascal Girerd

A SoftPLC is a software-based version of a programmable logic controller (PLC). The term SoftPLC has been used for many years, and like many technical terms, it means different things to different people.

IEC 61131-3 standard

For many years, the control systems market saw the proliferation of a variety of programming languages and development environments—different for each manufacturer. IEC 61131, the international standard for programmable controllers, was first published in 1992 (as IEC 1131). IEC 61131 standardizes programmable controller technology and covers equipment requirements, programming languages, user guidelines, communications, and functional safety.

The third part of IEC 61131 (IEC 61131-3) deals with the programming languages used in programmable controllers. The standard originally defined

five programming languages:

- Ladder Diagram, based on the most common form of PLC programming language.
- Structured Text, similar to high-level information technology (IT) programming languages such as C or Pascal.
- Function Block Diagram, a graphical language where users connect blocks of functionality together to create a program.
- Sequential Function Diagram, another graphical language that allows finite state machine execution.
- Instruction List, another text-based language similar in style to assembly language. This is now deprecated in the standard.

The IEC 61131-3 standard defines a series of functions and data types that must be supported by all compliant

programmable controllers. The functions are the basic building blocks of all programs and include arithmetic opera-

tions, Boolean logic, and programming structures such as loops, comparisons, and decisions.

SoftPLCs in Action

The growth of software’s role in automation has increased the spread of SoftPLC solutions, according to a Seneca Automation Interfaces blog post (<https://blog.seneca.it/en/the-softplc-concept>). These software emulations of typical PLC functionality, which are based on PC-type architectures, are equipped with real-time operating systems, like Windows CE, RTAI, RTLinux, and QNX, and implement the IEC 61131-3 standard. The Sequential Function Chart (SFC) language is used as a tool for the design and description of control logic and the four other standard programming languages: Ladder Diagram (LD), Function Block Diagram (FBD), Structured Text (ST), and Instruction List (IL).

Associations such as PLCopen (www.plcopen.org) promote standards related to SoftPLC technology, their reuse, and interoperability with other systems—not only IEC 61131-3, but also libraries for motion control, remote control, integration with safety features, and interface management. Although the IEC 61131 standard is common to SoftPLC and PLCs, manufacturers of “traditional” PLCs still tend to interpret hardware constraints and programming environments according to “proprietary” visions. The portability of applications between different PLCs or SoftPLCs never has zero cost: Accurate I/O reconfiguration and machine cycle verification activities are required, according to the Seneca post.

Conversely, SoftPLC systems are characterized by a compact architecture with decentralized I/O, a high level

of connectivity and interoperability, an ability to manage and store large amounts of data, an ability to operate in noncritical operating conditions, distributed architecture based on open hardware and common to other applications such as supervisory control and data acquisition and cloud, and easy software application transfer between different PCs (figure).

In an Automation.com article titled “RelaxC Controller Improves and Maintains Process Control,” author John Masse, founder of APPEDGE, writes about a new type of controller, “The industrial controller RelaxC is an innovative and unique method of stochastic gradient descent on time evolving systems.”

According to Masse, the “controller has been successfully tested on complex industrial plants. By its performance, RelaxC is a great alternative to advanced or classical control methods, such as predictive functional control, model predictive control, internal model control, LQG, LQR, and proportional-integral-derivative, which struggle or fail in the control of complex systems of the following types: nonlinear, with delay, with nonminimal phase, and with control constraints.

“With its small memory footprint, RelaxC can be easily encapsulated in engineering software, run times, and Soft-PLC to the IEC 61131-3 standard with languages (ST, FBD, SFC, LD, and IL).” —By Jack Smith

Feature	Standard PLC	SoftPLC
CPU	Proprietary	Standard/dedicated/multicore
I/O	Centralized/distributed	Distributed on fieldbus
Data memory	MB	GB
Operating system	Proprietary	Standard/HRTOS: Unix, POSIX, VxWorks, Linux, QNX, RTXDOS, VRTX, Windows Embedded, .NET
Programming language	IEC 61131 (SFC, LD, FBD, ST, IL), C, C++	IEC 61131 (SFC, LD, FBD, ST, IL), C, C++, C#, VB, Java Text Editor, JSON-RPC, Node-Red
LAN protocols	Proprietary, TCP/IP	TCP/IP, OPC, CoAP, MQTT, Rest
Sensors/actuators protocols	Proprietary, fieldbus	Fieldbus, Ethernet, OPC

One view of how hardware-based PLCs and SoftPLCs compare

Courtesy of Seneca Automation Interfaces

Putting IEC 61131-3 into Perspective

Industry forums and presentations focused on digitalization usually mention that companies need to look at new business models. This is also the case for traditional industrial automation vendors. With open multivendor interoperable standards, automation companies still need to be able to compete on traditional dimensions including quality, reliability, customer service, and value.

Bill Lydon, *InTech* and Automation.com contributing editor and director, North America, for the PLCopen organization, moderated an insightful roundtable discussion on “Automation Industry Expert Insights: CODESYS Tech Talk Fall 2021” (www.automation.com/en-US/Articles/October-2021/Automation-Industry-Expert-Insights-CODESYS-2021). It covered industrial automation future trends and challenges with industry leaders and innovators. Roundtable participant Don Bartusiak, president of Collaborative Systems Integration, said, “A challenge for end users is to be bold enough to go after step change improvements in productivity and step change reductions in total cost of ownership of systems by pursuing open architecture system concepts and open process automation ideas.” Bartusiak is a champion of open automation systems, instrumental in the development of the Open Process Automation System standards.

On Automation.com, Lydon also wrote “The Missing ‘Industry 4.0/Digitalization’ Link—Open Programming Standard Conformance & Certification” (www.automation.com/en-US/Articles/May-2020/Missing-Digitalization-Link-Open-Program-Standards). The fundamentals of IEC 61131-3

have been adopted by many kinds of automation vendors throughout the world. IEC 61131-3 is supported by the PLCopen organization that extends the standard with special interest groups, standards, and certifications. These standards and certifications include motion control, safety, OPC UA, XML interchange, and reusability. Due to the task structure of a full IEC 61131 implementation, both event-driven and cyclical approaches can be accomplished. There have been significant enhancements to IEC 61131 by the PLCopen organization, including OPC UA for enterprise communications, remote procedure calls, and controller-to-controller standardized data communication models.

Compliance and conformance are essential

Regardless of the programming standard, according to Lydon, there must be strong vendor compliance and certification of products to effectively ensure program portability and multivendor interoperability. This has been an ongoing issue in the industrial and process automation industry. The lack of vendor full conformance and certification to open programming standards creates tremendous inefficiencies and architectural challenges that lead to isolated “islands” of control. Integrating these islands into an entire plant architecture requires a great deal of application engineering, extra hardware, and more software to build a coordinated plant automation and control system. The added layers of interfaces cause lower reliability, increased production downtime, and potentially higher costs. —By Jack Smith

One of the outcomes of IEC 61131 adoption was the recognition that the software and hardware elements of programmable controllers could be considered separately. This in turn created the SoftPLC concept.

SoftPLC characteristics

A SoftPLC combines the functions of conventional PLCs with those of data loggers, communications gateways, and other elements such as human-machine interfaces (HMIs) and web servers. In the early days of SoftPLCs, it was common to use industrial PC hardware as the platform. This approach is one reason there are many misconceptions about what SoftPLCs can and cannot do, including:

- **Nondeterministic operation:** Early industrial PC hardware tended to run

the Microsoft Windows operating systems. Because these are nondeterministic, many people decided a SoftPLC could never be used in place of a conventional PLC.

- **Reliability and ruggedness:** The experience of Windows operating system reliability, combined with the potential lesser environmental specification of the PC hardware, led many to discount this solution for industrial applications.
- **Security:** Once industrial control system security became an issue, the maintenance of obsolete operating systems, such as Windows NT on SoftPLC hardware, was a potentially limiting factor.

However, there are many flavors of PC hardware and operating systems that can

be a SoftPLC platform. Real-time operating systems such as RTLinux, QNX, RTAI, Intime, FreeRTOS, and VXWorks provide the fault-tolerant, deterministic behavior expected in a conventional PLC.

Many PLCs that users believe to be conventional are in fact built on SoftPLC technology. The principal business model for SoftPLCs is based on original equipment manufacturers (OEMs) porting the SoftPLC run time to their platforms and seamlessly integrating it into their wider configuration and monitoring ecosystems.

Reasons to use SoftPLCs

Although there are some misconceptions, there are many more potential benefits of SoftPLC technology—some that relate to the OEM and some to the end user:

- **Time to market:** OEMs can develop their applications and select or design their hardware systems in parallel, and finally port the SoftPLC run time quickly and easily.
- **Versatility:** The same SoftPLC application can be ported to different hardware and operating system combinations. This means users can potentially develop and maintain a common application (or at least a library of common functions) that can run in different environments if needed.
- **Upgradability:** The abstraction of software and hardware makes it easier for the OEM to update the hardware and operating system while maintaining backward compatibility on software and applications.
- **Standardization:** Following IEC 61131-3 simplifies user training and supports greater consistency in application development, which in turn helps reduce ongoing maintenance costs. Even if a user changes a SoftPLC solution, using IEC 61131-3-compliant application code can greatly reduce the cost of migration. This also applies to communications protocols (such as IEC 61850, DNP3), which are often integrated into SoftPLC solutions.

- **Product differentiation:** Many SoftPLC solutions have functionality besides the IEC 61131-3 standard. Examples include support for industrial protocols, built-in redundancy, and distributed operation capabilities.

Security and safety concerns

Security and safety will always be major concerns with any automation system component. It is possible to develop a resilient, secure, and safe solution using SoftPLC technology, and many OEMs have done just that. When the OEM has all the source code, everything is under its control.

An advantage of SoftPLC solutions is hardware and operating system choices can be made independently, based on security and safety requirements.

Some SoftPLC solutions have even developed run times that OEMs can integrate into systems targeted for IEC 61508 certification. They do this by reducing the feature set, removing functions that can create safety hazards, and developing the run time itself using approved methods.

Final thoughts

Much has changed in the 30 years since the inception of IEC 61131-3, and many popular “conventional” PLCs are

running on SoftPLC technology. As with any solution choice, the requirements should drive the answer, but SoftPLC-based approaches should be included in any consideration. ■



ABOUT THE AUTHORS

Steve Mustard, PE, Eur Ing, CEng, CAP, GICSP, CMCP, FIET, is president and CEO of National Automation Inc. and former ISA

president. He is a senior member of ISA and former board member of Automation Federation, former chair of Automation Federation’s Cybersecurity Committee, and member of the ISA-99 committee. Mustard is a licensed professional engineer in Texas and Kansas, and a U.K.-chartered engineer with technical development and management experience in process automation and business process reengineering across multiple sectors.



Pascal Girerd is business developer at Straton Automation, which provides a software-based PLC with many benefits to industries like utilities and embedded automation.

SCADA Needs Open Standards Too

Standards are not meant to eliminate thinking and stifle creativity. If used properly, standards create a common language and a systematic approach to solving problems using industry-wide best practices. They encourage communication and foster creativity in solving engineering problems.

A supervisory control and data acquisition (SCADA) system should be based on open standards, which promote interoperability and make integration, scalability, and collaboration much easier, writes Don Pearson, chief strategy officer at Inductive Automation, in an Automation.com ebook article titled “Must-Have SCADA Features for the Modern Era.” Open standards are vendor neutral by definition, offering an alternative to a single vendor’s proprietary technology, which often prohibits mixing and matching pieces of software or hardware.

Open standards poise a system for moving into the future by allowing organizations to choose best-of-breed technologies instead of being locked into a specific vendor’s ecosystem, according to Pearson. —By Jack Smith

RESOURCES

“Manure Spreading Goes High-Tech with IIoT”

www.automation.com/en-US/Articles/August-2021/Manure-Spreading-Goes-High-Tech-IIoT

“PLCopen and OPC Foundation release new edition of PLCopen OPC UA Client for IEC 61131-3”

www.automation.com/en-us/articles/2016-2/plcopen-and-opc-foundation-release-new-edition-of

“Tension in Industrial Automation: Open Systems and the User’s Dilemma”

www.automation.com/en-US/Articles/2019/tension-in-industrial-automation-open-systems-and