[7 Differentiating Characteristics of PLC / PC]

Seven Major Areas to Consider when Selecting PC or PLC Based Architecture

Operation

When analyzing system operation, focus on how the system will run and how instructions and tasks are processed. The standard PLC has an embedded real-time operating system (RTOS) with a dedicated processor that ensures a high degree of control system reliability. Furthermore, since the PLC handles only automation and/or a process, it does not need to run other utilities, such as antivirus programs or system updates.

A PC with a real-time kernel or real-time operating system (OS) can accomplish the same reliability of control as a PLC. From earlier experiences with PCs at home and in the office, users might be wary of lock-ups of the PC (the so-called "blue screens"). However, these lock-ups could occur on any OS, including a PLC, if the OS software is not handling priorities properly. For industrial use, the software running the PC is dedicated to automation and will therefore have a minimal chance of lock-ups. Even if a lock-up would occur, the real-time kernel is not affected and continues operation.

Real-time operation is a relative concept that means any task is guaranteed to be handled within a certain time. Synchronized motion and/or advanced PID control requires a high level of real-time determinism, while noncritical supervisory controller operations, such as monitoring error messages or sending noncritical controller commands or queries, would not.

Robustness

Robustness of the controller refers to its durability in various environments. The standard off-the-shelf PLC has no moving parts, so it can withstand harsh environments for millions of cycles. A standard PC contains moving parts, such as fans or hard disk drives, and is less suitable for environments with high vibration levels.

However, industrial PCs (IPCs) offer options such as solid-state drives, fan-less cooling, and in-cabinet mounting. These options make a PC just as durable as a PLC, able to withstand the toughest industrial or environmental conditions. PLCs and PCs have converged in this area, but the PC requires additional options to equal a standard PLC.

Serviceability

Another factor to consider is the ease and cost of serviceability, which can be measured by the repair and replacement costs over the life of the controller. For a PLC, external devices can be replaced with ease while the system is in operation. Moreover, the PLC's compact modular design makes replacing the actual controller an easy job. This saves cost because it reduces machine downtime. It's also possible to perform a hot swap with a PC, but only for USB or other external peripheral devices. If the PC has a more modular design, such as a rack or panel-mount system, replacement time is closer to that of a PLC.

It is helpful to be able to effortlessly change out a system or its components and have a resource pool of replacement parts with long-term availability. In some industries, "copy exact" policies require this long-term availability. It is easier to implement copy-exact with a PLC since the hardware and firmware don't change as rapidly as for a PC. Trying to find parts for a PC (even after a year or two) can be more challenging than for a PLC. (Ask the vendor about such policies, availabilities, and costs.)

Hardware integration

Every engineer appreciates a wide choice of options when selecting control system hardware, since there is always a need for items such as additional peripherals, memory, and a user interface. Both the PLC and PC have the ability to control a multitude of devices using industrial communication networks. Some of the well-known networks are SERCOS, Profibus, DeviceNet, and CANbus, as well as their Ethernet-based counterparts such as SERCOS III, Profinet, EtherNet/IP, and EtherCAT.

Although the PLC and PC can offer an array of fieldbus options, the PLC has many of these options built in, whereas PCs and some IPCs need additional cards and drivers to provide a comparable offering. However, besides the typical fieldbus networks, the PC is equipped with a more open and flexible array of interfaces, such as USB, FireWire, serial, wireless Ethernet, etc.

This gives the user access to more off-the-shelf devices to handle tasks that a PLC usually could not handle, such as an advanced high-resolution imaging system, where the images would be stored, analyzed, compared, and possibly archived. A PC would be well suited for this advanced task because of the amount of memory required; a PLC would have limited storage and processing capabilities.

For some applications, the user interface is a crucial machine component. The PC has a built-in user interface, while the PLC would need one or all of the following for primary operation: switches, operator panel(s), or an industrial PC. So while the PLC can interface with devices over fieldbus and perform complex operations, it still needs a PC to handle memory-demanding tasks and make connections to other devices in the system.

Security

Security mainly deals with protecting the file system and application. This has two aspects: preventing unauthorized access from the outside world (such as virus attacks, malware, etc.) and limiting user access (such as restricting user rights, hiding files, etc.).

Traditionally, a PLC is less exposed to unauthorized access from the outside world. Because of its dedicated OS, there are very few known instances of virus attacks on PLCs. However, this does not mean that PLCs are immune to viruses. Since PLCs have enjoyed a virus-free status for years, there are no standard ways of detecting and eliminating a virus if it were to occur. Although the PC is far more susceptible to virus attacks than a PLC, strict security measures can be invoked to significantly reduce potential threats, and standard software is available to detect and remove viruses. PLCs and PCs offer different levels of user access to keep the contents as secure or as open as desired.

Safety

Depending on the operating environment, safety can be a major concern for the user. It is vital in cases where human interaction with the machine could be potentially dangerous and when safety standards in a semiconductor fabrication facility or other factory are crucial to machine acceptance.

The PLC has a long history in machine automation, designated communication channels to slave devices to tightly monitor operation, and optional integrated circuits for redundancies. Integrated safety has only recently become available on some PC-based platforms.

Programming

Device functionality is only as good as the program running on it. Therefore, the programming environment and language are crucial to optimal machine performance. One main difference between PLC- and PC-based solutions is how the code is executed. A PLC mixes scan-based and event-driven program execution, whereas PC software is typically event-driven. The scan-based execution of a PLC program might take longer because the system needs to complete the higher priority actions in the cycle first. The difference in execution style requires a different programming philosophy, and often users are committed to either one or the other.

The same holds for the programming language: PLCs are programmed using languages specified in IEC 61131-3 (ladder logic, instruction list, etc.), or proprietary vendor languages. PC-based controls can use programming languages such as C/C++/.NET.