Fully Automated

Process control with PLCs

R.A. Hulsebos (The Netherlands)

Mass production is an important part of our modern society. Every production process involves actions that are repeated innumerable times, and these actions are often controlled by PLCs. What exactly are PLCs, and how are they used?

Programmable logic controllers (PLCs) are the workhorses of industrial automation. Originally developed as software simulations of relay control circuits, PLCs have developed into a platform that forms the basis for control applications generated using structured programming languages (IEC 61131), including high-speed motion, machine vision, networking, and integration with databases and logistics systems.

From an electronics perspective, a PLC is simply a processor with memory, I/O channels (digital, analogue, and/or serial), some counters and logic circuitry, and a network interface. What transforms this into a PLC is the PLC operating system (OS). Using a programming package running on a PC, programmers generate PLC application software to control machines and production lines.

The electronic configuration of a PLC varies from one supplier to the next. A wide variety of processor types can be used for the processing function, such as ARM, X86, NIOS, and so on. Although PLCs are industrial equipment instead of consumer products, there is considerable price pressure, so suppliers devote a lot of attention to cost-efficient development.

Some suppliers take a different approach. They start with a PC, install a PLC OS, add some I/O, and the result is a PLC. This is called a 'software PLC'. The difference between this and a 'real' PLC is that it also has all the capabilities of a normal PC. However, from a hardware perspective a PC is not entirely the same as a 'real' PLC. Some of the specific features are discussed below.

Fast up and running

A PLC must operate all the time, 24/7. If a PLC fails, the result is usually dramatic: production grinds to a halt. In most companies, the maintenance department has only one priority when this happens: getting production up and running again. If power cycling (off/on) doesn't help, the PLC is replaced by a spare unit. It must be possible to install and

connect this unit quickly.
Software installation is often a tricky problem in this situation. It is increasingly common practice for controllers to fetch their software from a central server. However, central servers are not always available, so some suppliers use memory sticks instead. Memory sticks resemble USB sticks, but they are different. An example of a memory stick is the C-Plug (**Figure 1**) for the Siemens S7 family of PLCs. A memory sticks holds all the required software and configuration data, and it can simply be unplugged from the old PLC and plugged into the new one. This technique is also being used more and more often with peripheral equipment.
A problem with Ethernet is that the new controller has a new MAC address, since the Ethernet standard requires every device in the world to have a unique MAC address. This creates difficulties in an industrial environment, because the Ethernet network address depends on the MAC address. In a network environment, this means that the new controller is invisible until the other network devices have been configured to use the new MAC address. Naturally, this is very inconvenient in an industrial environment. For this reason, many suppliers allow customers to configure the controllers with their own MAC addresses. This increases flexibility, but it also creates responsibility: since every MAC address must remain unique worldwide, the old controller must never again be used with its original MAC address.

No more battery backup

PLCs rarely have hard-disk drives. In the first place, hard disk drives are far too expensive, and the vast majority of their storage capacity would remain unused. The moving parts of hard disk drives also make them too vulnerable, and they are very sensitive to hard shocks and strong vibrations. As a result of rapid technological evolution, hard disks are quickly replaced by newer models with even more capacity.

Most PLCs need only a few megabytes of memory, and until a few years ago many of them used static RAM (SRAM) for this. However, static RAM requires a battery, and batteries have a limited operating life. The situation changed with the advent of flash memory, which is now more or less standard for program code storage. However, it is not always practical for data storage because it does not support write transactions for individual bytes. Ferromagnetic RAM (FRAM) is much more suitable for this purpose.

Watchdog

It's always possible for an application program to hang or the underlying core routines to stop working. In order to prevent the entire system or machine from coming to a halt when this happens, a watchdog is always present. A watchdog is nothing more than a simple timer that resets the processor if it times out, which causes the controller to execute a restart. The watchdog is reset periodically by the application code or a core routine, so it never times out under normal conditions.

Dual processors are used in applications where controller failure is intolerable. One of the processors is the active or 'hot' processor, while the other one is the standby processor. Both processors execute the same code and receive the same input data, but only the hot processor drives the outputs. If the hot processor fails, the standby processor can seamlessly take over control.
Naturally, dual processors cannot eliminate the risk of problems due to software errors. If a controller fails due to a bug in its application software, there's little point in switching to another one, since the other processor has the same software with the same bug. There is only one way to avoid this problem, which is to develop all software redundantly using teams that are not allowed to communicate with each other. The idea here is that the two versions of the software will have completely different designs, so it is unlikely that the same conceptual error, and thus the same bug in the same routine, will be present both versions. This approach is used in the control systems of Airbus aircraft, among other examples.

Figure 1.
A Siemens C-Plug with a capacity of 32 MB. Source: Siemens

Windows

The enormous popularity of Windows makes it a natural candidate for use as the internal operating system of controllers. This is entirely hidden from users because the PLC's control application completely conceals the OS. Microsoft provides two products for this purpose: Windows CE6 and Windows XP Embedded (XPe). Windows CE is primarily intended for embedded applications, such as small PLCs. It has a very low licence fee (a few pounds) and can run on all sorts of processors. XP Embedded is more suitable for rel-

Figure 2.
A DiskOnChip flash disk with wear levelling, which can be plugged directly into an IDE connector. Source: Coresolid Storage

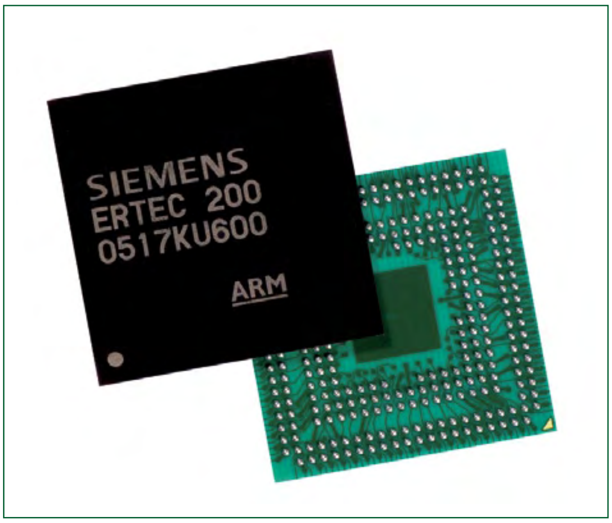

Figure 3. A Siemens ERTEC 200 processor with a built-in ARM processor and two Ethernet interfaces. Source: Siemens

atively large PLCs with networking capability, video, image processing and data handling capability, but it requires a more powerful processor (500 MHz minimum) and more memory (at least 256 MB of RAM), and it is much more expensive (around £100 / $145 per licence). On the other hand, with XPe you can do everything you can do with a normal PC, because XPe is essentially the same as XP Professional. The only difference is that you can choose which parts to leave out, which yields major savings in storage space: an XPe installation can fit into 100 MB. Up until a few years ago, this was ideal for working with flash disks, although their capacity has increased enormously in recent years to the point that a few megabytes more or less are no longer a major issue.

However, flash memory suffers from the limitation of a finite number of write cycles per sector. For example, many flash chips can only be written 1 million times. This is too little for Windows XP, especially in areas of the file system that are written very often, such as folders and the Registry. The NTFS file system is also very active source of write transactions. For this reason, Microsoft has an option for XPe called 'Embedded Write Filter' (EWF), which stores disk transactions in RAM. The disk appears entirely normal to the application software, but all changes are lost when XPe restarts. For many applications, this does not pose a serious problem. In fact, it can be an advantage to always start with a clean slate.

For applications where EWF is too restrictive, flash disks that spread write transactions over the entire disk must be used. This is called 'wear levelling', and it is available from a variety of suppliers. One example is DiskOnModule from Coresolid Storage, which can be fitted directly to an IDE connector (**Figure 2**). Even if an application constantly writes data to a disk of this sort, wear levelling provides an acceptable service life. Windows sees a DiskOnModule as a normal hard disk, and no special driver is necessary.
If you wish, you can experiment with XP Embedded free of charge. All of the software can be downloaded from the Microsoft website, and applications will run for 90 days without a licence.

Copy exactly

Some industries demand 'copy exactly' from their suppliers. This means that a controller that they buy today must be exactly the same as one they bought several years ago. In this way, customers are not repeatedly confronted with new controllers that require the installation of different software or make it necessary to revise drawings, modify connecting cables, rewrite documentation, and so on. There are also companies that put together a production line in one country and then want to copy it in other countries, so that the same product can be made everywhere in the world in the same way.
Naturally, 'copy exactly' places heavy demands on logistics chains. It prevents the use of components with short market lives, which is a frequent phenomenon in the PC world. Industrial PCs also suffer from this phenomenon. If you want to avoid it, you must explicitly look for a PC supplier that can guarantee long-term deliverability. As this is requested relatively often by industrial customers, there are

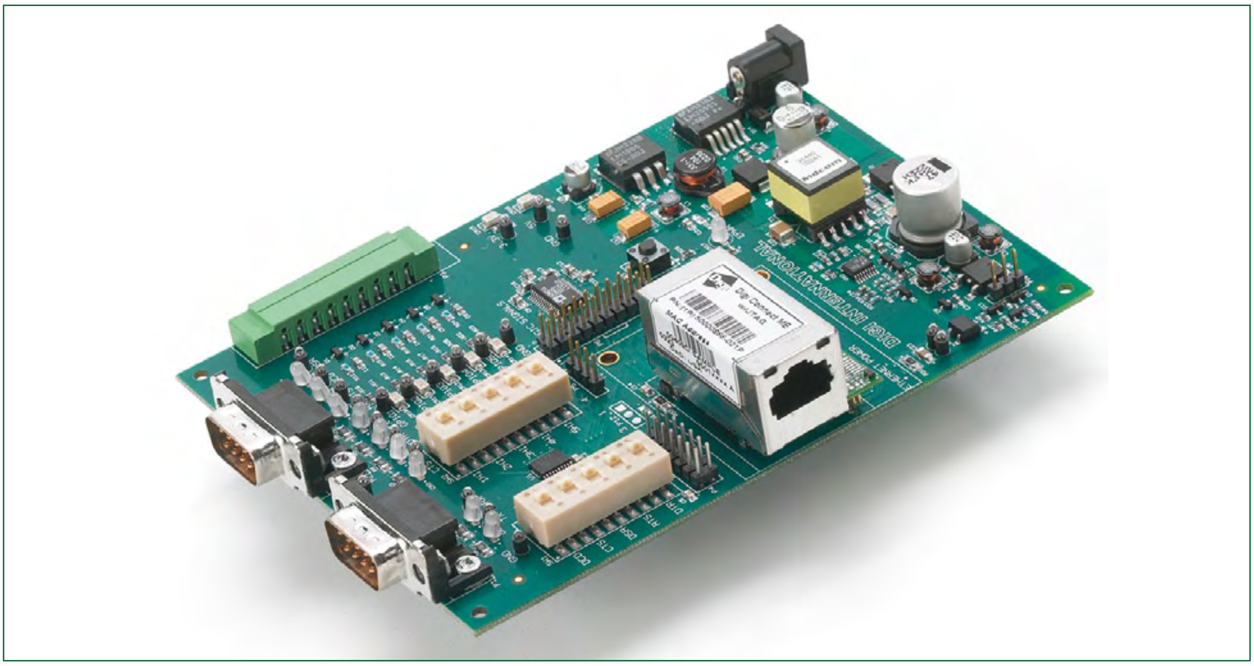

Figure 4. Developer's kit for the Digi/ME controller. The controller is fitted in the middle of the PCB. Source: Digi

in fact suppliers (such as Advantech) that can do so. Intel can also provide long-term delivery guarantees for some processors, including Celeron.
If you want long-term deliverability, you're looking in the wrong place if you constantly focus on the leading edge of industrial PCs. Instead, you should be thinking in terms of a 500-MHz Intel Celeron processor. Although this may sound rather archaic, it isn't. Many industrial applications do not require especially high processing power, and 500 MHz is already more than enough for such applications. Another helpful factor is that industrial IT programmers are often used to working with controllers that have limited resources. This is quite different from the situation in business IT, where nobody thinks twice about another gigahertz or an extra gigabyte.

Networks

A modern PLC is equipped with suitable interfaces, including network interfaces. This naturally includes Ethernet, which makes it very easy to use a PC to download program code and correct bugs in the software. Ethernet can also be used for quick, inexpensive linking to a supervisory control and data acquisition (SCADA) system. Operators can use a SCADA system to run the machine or system, check the current status, enter production orders, collect statistical data, and analyse error messages.

Many PLCs also have an RS232 interface, although this interface is being used less and less often. RS422 and RS485 are also quite common because they can be used over much longer distances (up to 1200 metres). In top-end units, these interfaces always have galvanic isolation with short-circuit protection. This is often missing in inexpensive controllers, or they may only be able to withstand a one-second short circuit. As it is very easy to cause a short circuit, especially with 9-way D-Sub connectors, inexpensive controllers often turn out to be a costly choice. On top of this, you will be faced with a failed controller, and thus possibly a production shutdown. if the short circuit causes the transceiver to fail.

In addition to the RSxxx interfaces, there are all sorts of fieldbus interfaces available. The fieldbus market is highly fragmented, with more than 500 different bus systems. A few of them are very well known, such as Profibus, CANbus and AS Interface, but there are also many systems with only a small market share. As each system has not only its own cabling and connectors but also its own range of products, including I/O modules, motor controllers, dampers, valves, serial interfaces, repeaters (amplifiers) and so on, the market can be regarded as highly fragmented. Consequently, before you buy a PLC you must carefully consider which fieldbus you want to use and whether all of the necessary functionality is actually available for it.

Ethernet

A trend toward using only Ethernet instead of the hundreds of different fieldbus systems has developed in recent years. As certain adjustments are made to make this interface suitable for control applications, it is also referred to as 'industrial Ethernet'.
Two different philosophies can be seen here. Some companies use the TCP/IP protocol as much as possible with standard Ethernet, while other companies want to use Ethernet in high-speed motion control systems to drive as many motors (servos) as possible and run them as fast as possible. This requires special Ethernet interfaces. The idea here is to execute the entire Ethernet protocol in hardware so that it is very fast and real-time (deterministic).
Siemens supplies an ARM-based ERTEC-400 controller (**Figure 3**) for its ProfiNet protocol, and Beckhoff supplies special ASICs for its Ethernet protocol. This makes it possible to drive Ethernet I/O directly without processor involvement. The author managed to do this at a frequency of 30 kHz using a standard desktop PC. There is ongoing development activity in the Ethernet world, and gigabit industrial Ethernet is expected to be available fairly soon.

USB

Due to the popularity of USB on PCs, this interface is also being used more and more in industrial applications. Much longer distances than the usual 5 metres can be bridged by using USB extenders. This means that keyboards and mice can be located much further away than usual, which makes it considerably easier to install a PC (or an industrial PC) in a system. USB is hardly used for I/O. With its short cable lengths and star topology, it is not a good fit with typical industrial applications. However, it is used in instrumentation systems, such as with Matlab or Labview.

Figure 5.
A Lantronix XPort/AR controller, which is scarcely larger than an RJ45 plug. Source: Lantronix.

No PLC

Despite the popularity of PLCs, there are a fair number of automation specialists that prefer not to not use them. Limited memory capacity, the low-level programming languages specified by IEC 61131 (which among other things do not support object-oriented programming), and strict encapsulation in typical application architectures make them unsuitable for use as high-end machine controllers. Instead of PLCs, such applications employ several small embedded controllers linked to an industrial PC by a network. In this arrangement, the non-real-time portions of the application are executed on the PC. Two examples of small embedded controllers are the XPort from Lantronix and the Digi/ME from Digi (**Figures 4** and **5**). Actually, these devices are small only with respect to their dimensions (the size of an RJ45 connector) and their prices (a few dozen pounds); they both have a powerful processor, lots of memory, ample I/O, and networking capabilities. They also come with a (hard) real-time kernel. Program code can be downloaded via the Ethernet interface or the JTAG port.