GE
Intelligent Platforms

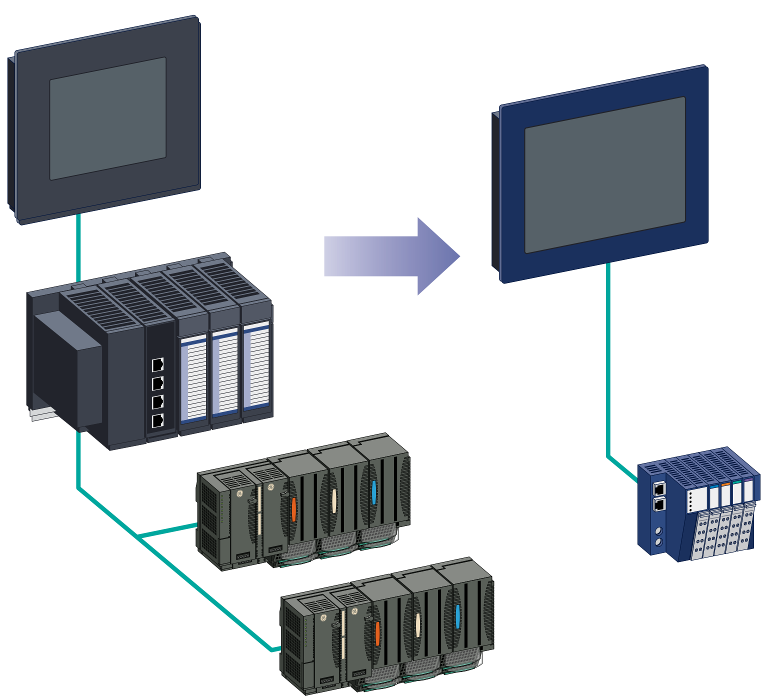
Automation Panels vs. PLCs in System Control

Automation Panels vs. PLCs in System Control

As OEMs continue to move toward leaner design, the operator interface has evolved into a replacement for the PLC in many machine applications. This white paper explores the advantages and disadvantages of consolidating HMI and control into a single operator interface panel.

In the early days of automation, original equipment manufacturers (OEMs) shipped control systems with rack-mounted PLC's, pilot lights, gauges, and push buttons. Over the years, the vast majority have simplified their systems by migrating to operator interface (OI) panels in place of other panel mount components. To reduce wiring costs and to make it easier to ship their equipment in modular sections, many have also moved to distributed IO. And, in an effort to further reduce the cost associated with reactive maintenance, many OEMs are now adding secure remote connectivity to access end user networks to perform remote program modifications and analyze equipment performance.

Typically, an OEM machine would utilize a programmable automation controller (PAC) with distributed IO, a touchscreen operator interface with data logging, and an industrial security router. The PAC, OI, and router each have their own processor, their own installation requirements, and their own unique software configuration.



Legacy OI, PLC and Distributed IO

QuickPanel+ Control System

The processing power of these individual components far exceeds the needs of most applications, so the temptation to reduce these components into a single device is overwhelming. But what are the tradeoffs?

Advantages of Automation Panels

Automation panels combine the programmable controller functionality and the operator interface into a single unit. Automation panels entered the market about 15 years ago. Many of these early units were simply operator interface panels with some local IO, ladder logic, and a flat database. Modern automation panels like the GE QuickPanel+ include the full IEC61131 programming languages (Ladder, Structured Text, Function Block Diagram, Sequential Function Chart, and Instruction List), as well as user-defined data structures and user-defined function blocks.

It may be more accurate to describe these automation panels as PAC controllers with a built-in operator interface, rather than just an operator interface that performs control. In the case of the QuickPanel+, OEMs can purchase a remote security software package from Secomea™ that will allow the OEM to securely connect to the QuickPanel+ over the internet using the customer's existing network, eliminating the need for a separate security router. The advantages of this simplified architecture include cost savings, simplified maintenance, and improved performance.

Cost Savings

Automation panels can significantly reduce software development costs. Many automation suppliers tout the benefits of a shared database between the PAC and the OI panel, but if these are separate devices, then they still have separate databases at runtime. This means that each time you add a variable, you need to download to both devices. If the controller and OI get out of sync, you end up with communication errors and possibly unexpected operation. Automation panels truly use a single database with a single development environment and a single library for reusable objects.

Hardware costs are also reduced. Combining the controller, operator interface, and remote connectivity into a single device means only one device to purchase, install, and configure. This saves money on both production time and on panel space.

Simplified Maintenance

Maintaining one device is less work than maintaining three, especially when you have shipped a system to an end user that may be hundreds or thousands of miles away. With an automation panel, you can back up the operator interface and logic program on a single memory card or USB stick. If the end user has separate files for the operator interface and controller and needs to restore one or both programs, they might load different revisions and end up with a non-working system. Having a single program to restore is easier and eliminates version compatibility issues.

A single device means a single point of connect. There is no need to connect to multiple ports to monitor or upgrade the system. This can be even more valuable when dealing with remote connectivity, especially if the PLC only has serial port programming. If using the QuickPanel+ with remote security purchased from Secomea™, the OEM can easily access any of their remote sites by logging into a single server.

Improved OI Performance

It may sound counterintuitive, but combining the PLC and OI into a single device can actually improve the update times for the operator interface in many applications. This is because one of the main CPU tasks for a traditional operator interface is communications with the controller. When today's operators press a button on the OI screen, they expect an immediate response for the equipment and immediate feedback on the graphic screen. The biggest reason for delays in that response is the communication driver between the OI panel and the PLC. With an automation panel, this communication is much faster, because it is internal to the device. There is no need to rely on serial or Ethernet communication links for updating operator screens.

It is important to note that this advantage may be overshadowed by the overall performance requirements. For example, if the control system requirements are consuming the vast majority of the CPU time, then the operator interface performance can suffer because it runs at a lower priority than the control. This will be discussed later in this document.

Perceived Disadvantages of Automation Panels – Dispelling the Myths

Programmable controllers have been an industry standard for decades. They have a strong reputation for reliability, performance, and real-time deterministic control. Anyone considering a move from traditional PLCs or PACs to automation panels should understand the tradeoffs. There are some significant misconceptions about the performance and reliability of automation panels in comparison to traditional PLCs. Before getting to the actual tradeoffs, we should examine these misconceptions.

Is it PC Control?

In the mid-'90s, many were predicting that PC control would replace PLCs in the automation control market. While PC control fits very well in certain applications, it certainly has not taken over the market. The main issues with using PC control are:

- Determinism (required for repeatable IO updates)
- Security concerns requiring virus protection and OS patches
- Long boot time
- Registry errors (often caused by powering off without shutting down Windows)
- Moving parts in hard drives and fans

While PC control applications can take steps to overcome some of these concerns, the fact remains that Windows 7/XP/NT are not designed to be real-time operating systems.

Programmable controllers use real-time embedded operating systems such as VxWorks or QNX as well as many proprietary operating systems. When people first see an automation panel with a graphic screen and a built-in control engine, they immediately think of PC control. But just like their PLC counterparts, the vast majority of automation panels run on embedded operation systems that have none of the issues listed above.

One of the most popular OS choices for automation panels is Windows CE. The Windows CE family has been used in real-time automation control for 15 years.[1] With the introduction of Embedded Compact 7 in 2011, Microsoft dropped the Windows CE name and replaced it with "Embedded Compact". This article will use the term "Windows CE" to refer to multiple vendors who use various generations of this OS including EC7.

The primary concern with traditional Windows is deterministic scan times. Windows performs numerous background tasks, so Windows users are all-to-familiar with waiting for a response while the computer is doing who-knows-what. If your control system had to wait some undetermined amount of time, the results

> *"Microsoft sells two operating systems tailored to the embedded community, confusingly named Windows Embedded Standard 7 and Windows Embedded Compact 7. Windows Embedded Standard 7 (WES), or just Standard 7, is a repackaged Windows 7 operating system described elsewhere in this magazine by my colleague Sean Liming. Windows Embedded Compact 7, or Compact 7, is a purpose-built operating system designed for mobile and embedded systems." –* ***RTC Magazine, January 2011***[2]

could be disastrous. Like other embedded real-time operating systems, Windows CE achieves deterministic scan times for the controller by using thread prioritization and scheduled interrupts. The control functionality takes the highest priority. Windows CE has a miniscule amount of background tasks compared to a PC, but even these tasks have limited impact on control updates, because they run at a lower priority. Similarly, time-consuming CPU tasks such as running a video or opening large files will have limited impact on the control scan time.

Real-Time Operating System

"There are soft real-time and hard real-time systems. A soft real-time system can miss its bounded time response once in a while and still maintain a reasonable level of acceptable performance, such as when a Voice over IP device may delay, or skip, the delivery of voice packets and still provide acceptable service to the user. A hard real-time system cannot miss any of its bounded time responses. When a hard real-time system misses a bounded time response, it can cause catastrophic system failure. Imagine what happens when an automobile's electronic brake system fails to engage in a timely manner, while the automobile travels at a high speed and needs to make an urgent stop to avoid a collision. Compact 7 is a hard real-time OS that provides reliable core services to support embedded system design that demands low-latency, deterministic real-time system performance. Compact 7 has the following features required by a real-time system. Multithreaded and preemptive Prioritized thread scheduling Priority inversion prevention using priority inheritance to dynamically adjust thread priorities Predictable thread synchronization" -PROFESSIONAL WINDOWS® EMBEDDED COMPACT 7 by Samuel Phung, David Jones, Thierry Joubert[3]

Windows PCs are much more open than embedded platforms. They allow for numerous types of custom applications that can be developed and downloaded on multiple hardware variations. This leads to more overhead, longer boot times, security concerns, and frequent updates and patches. Embedded platforms are compiled for specific hardware and software must be written and compiled for specific implementations of the operating system. Windows CE platforms carry considerably less overhead and are not susceptible to Windows viruses. A QuickPanel+ on Embedded Compact7 can boot up and be fully functional in 30 seconds. This is faster than some PLCs on the market today. Windows CE platforms can be powered down at any time without going through the OS shutdown sequence required by Windows.

Reliability

Most programmable controllers have a well-earned reputation for reliability. Operator interface products have not historically had the same reputation. Resistive touchscreens wear out over time. The display backlight eventually fades or burns out. The screen becomes damaged due to exposure to the outside of the control panel. All of these issues have led many to conclude that these types of devices cannot be relied upon for control. But is that really the case?

None of the problems described above would interrupt the controller itself from running and updating the inputs and outputs. This is demonstrated in this YouTube video (http://pages.ge-ip.com/b0000nCh5FF0HR1e010N5cV) In this video, a GE QuickPanel+ experiences catastrophic screen damage, but the QuickPanel+ is controlling the red flashing light in front of the panel, showing that the control system continues to function in spite of the damage to the screen.

If the touchscreen or display gets damaged or fails, you will need to eventually replace the automation panel, but the control program will continue to run. The unit in the video is still operational and the operator screens are fully functional through the built-in web browser.

Mean Time Between Failure (MTBF) data is often used as a benchmark for reliability. MTBF values are significantly higher on a PLC than on an automation panel because of potential failures of the touchscreen and display. These failures are included in MTBF data because they indicate the reliability of the device as a whole, not just the control functionality. This is clearly not an apples-to-apples comparison, because a failure of these components will not shut down the control system. In order for the automation panel to continue to function as a controller, it needs to have only the power supply, CPU and IO communications functional. These boards use the same types of industrial grade components that are used in PLC systems, and are ultimately just as reliable.

Control Languages

Since programmable controllers are dedicated to control, some assume that the programming languages may be more sophisticated or user-friendly. In reality, the opposite is often true. Although some automation panels still have a ladder only editor with a flat database, this is rapidly becoming the exception rather than the rule. Most automation panels support a full set of IEC languages, symbolic programming, user-defined data types, and user-defined function blocks. Of course, the same can be said of PAC controllers, but in many cases the automation panel is competing against a traditional PLC with ladder logic and referenced-based addressing schemes that date back to the '90s.

Advantages of Programmable Controllers over Automation Panels

Despite the advantages of automation panels, there are distinct, compelling advantages of using programmable controllers. These reasons include faster control system performance, modularity, and the requirements of high availability systems.

Many control applications have performance requirements that cannot be met by a single processor on an automation panel. The GE QuickPanel+ has a 1GHz CPU and up to 1G RAM. This enables the QuickPanel+ to perform as well or better than many low- to mid-range PLCs even while handling the operator interface requirements, but clearly a PAC controller with a 1GHz CPU is going to outperform an automation panel with a 1GHz CPU in terms of logic scan rate. If an application needs logic scans in the 10mS range, this can be a delicate balancing act for an automation panel to meet the need while allowing adequate CPU resources for the operator interface functions. Larger systems may exceed the needs of the single CPU because of a specific combination of logic performance, graphics, data logging, and other tasks. In these applications, a separate PAC controller is the obvious choice.

Not every control system requires a dedicated panel-mounted operator interface. Some operate as blind nodes, but many use a plant-wide SCADA system or a local PC for the operator interface requirements. Traditional PLCs and PAC controllers fit these systems very well. Many programmable controllers use separate removable modules for the CPU, power supply, local IO, and communications. If one of these components fails, it can quickly be replaced on an individual basis. Automation panels typically have the CPU, power supply, communications, and touchscreen sold as a single unit. If any of these components fail, you will have to get the entire unit replaced or repaired.

High availability systems are control systems that typically need to run 24/7 without interruption. These systems typically use hot-standby, redundant CPUs with synchronized scans to avoid a system shutdown if a single component fails. Hot-standby control is not as common in automation panels. If the touchscreen or display fails, the system will continue to run. At that point, the operator interface functionality could be operated through a remote web browser on a PC, or the system could automatically go into a controlled shutdown sequence. Either way, the automation panel would need to be replaced and would eventually require a system shutdown. With hot-standby CPUs on a PAC controller, the failed component can be replaced while the system continues to operate from the backup CPU, so typically no downtime is required.

Performance Data for GE's QuickPanel+

The following table shows scan time data for a Proficy Machine Edition control program running on the QuickPanel+. The Demo Program consisted of 22 screens, 24 scripts, 4 structure text subroutine, 3 ladder program subroutine, 1 FBD subroutine, alarm logging, data logging, 1402 variables and with Ethernet IO remote connection to RSTi node. Multiple tests were performed to determine how much impact operator interactions such as opening a video file, launching a PDF, or running a powerpoint slide, would have on the logic scan time. A further test used a script operating as part of the "View" application that looped 3600 times. The table below shows the results:

	TARGET SCAN	PEAK SCAN	AVG SCAN TIME	AVG LOGIC TIME
Controller Starting up running all programs	10	27	10	2
Launching Remote Desktop Viewer	10	36	10	3
Launching Video	10	30	10	2
Launching PDF document	10	37	10	2
Launching Excel document	10	37	10	2
Launching Power Point document	10	37	10	3
Running loop test (looping 3600 times)	10	37	10	3

"Target Scan" is a configurable value that determines how often the Quickpanel+ will attempt to scan the logic and update the IO. The "Peak Scan" is the highest observed scan for the duration of the test. Opening large files and running large scripts at the operator level do impact the logic scan, but the prioritized interrupts limited the worst case scan to 37mS, regardless of the size of the files being opened. This is analogous to the "Background Communications Window" found in traditional PLCs. Communication tasks on a PLC do impact scan time, but the amount of time spent on communication tasks is limited in order to give priority to the logic scan.

Conclusion

Automation panels offer the same deterministic real-time control as traditional programmable controllers. Programmable controllers are a better fit for extremely fast scan times, very large IO counts, or high performance redundancy. For low- to mid-range applications that require a dedicated operator interface, automation panels like the QuickPanel+ provide a simplified architecture with easy remote connectivity options and lower total cost of ownership.

References

[1] *"Windows CE Embeds Itself in Automation, Control, and Instrumentation" http://www.controleng.com/single-article/windows-ce-embeds-itself-in-automation-control-and-instrumentation/c62df27a241298339a1574974cca6590.html – 1998 Article in Control Engineering that introduces Windows CE as an Embedded OS for Automation*

[2] *"CE Goes Multicore: Microsoft Windows Embedded Compact 7" – http://www.rtcmagazine.com/articles/view/101957*

[3] *"Professional Windows Embedded Compact 7" by Samuel Phung, David Jones, Thierry Joubert – http://it-ebooks.info/book/811/*

Imagination at work

03.15 GFT-886