The Case for Intelligent Remote I/O

Wise Delegation for Your Automation System

Introduction

Are you having to do more with less? Are limited staff and budget causing you to look more closely at how you spend dollars and time?

Even in the automation industry, where we may jokingly refer to ourselves as "control freaks," delegating wisely is increasingly more important.

Just as a good supervisor strives to hire competent staff he can trust to do their assigned tasks with minimal instructions, automation engineers can use wise delegation and division of labor in automation systems so they run more smoothly and efficiently.

As automation engineers we're used to finding ways to automate something a person used to do by hand—freeing that person to do something more intelligent and less mundane—but it's easy to overlook opportunities for further automation and even greater time and money savings.

Maybe it's time to take another look at our systems.

Distributed Computing

As computers have become smaller and far more widespread, the concept of distributed computing has been employed in increasing numbers of applications.

From volunteer computing projects like Stanford University's Folding@home[1] disease research, to real-time control systems like those used in aircraft control[2], to the assortment of networked intelligent sensors and systems in your late-model automobile, more and more applications are using what amounts to a grid of networked computers, each one contributing its processing power to the same overall purpose.

On some level you've already embraced the distributed computing concept, as have all users of the Internet—the largest distributed computing application on the planet.

1. http://folding.stanford.edu/home/
2. http://www.laas.fr/TSF-EN/44-29985-Future-flight-control-systems.php

Users of the Internet, whether or not they're aware of it, benefit from three of the main advantages of distributed computing:

- **Minimizing single points of failure.** This is one of the biggest advantages to cloud computing applications. If one of Google's servers goes down, others pick up the slack. If many of the servers and switches in your region—including those that sent your request to Google—lose power at once, your answers can be routed from Google to your PC via others.
- **Spreading the load.** Multiple servers distributed across the network mean that a single, hugely powerful computer is not needed. Instead, these smaller servers are used for everything: searching for whatever keyword you just googled, routing the search results to your PC, and displaying the results in some meaningful manner on the monitor in front of you.
- **Maximizing scalability.** Distributed computing makes it easy to scale the system. Every day the Internet grows. Internet World Stats says the number of Internet users has almost quadrupled in 10 years: from 757 million in May 2004 to an estimated 2,937 million in March 2014[3]. Had it somehow run on one giant computer, major performance hits would have constantly knocked the system out.

Your industrial control system, and those who build and maintain it, can also reap the benefits of distributed computing. But what's the best way to maximize the advantages of a distributed system?

Minimizing single points of failure, spreading the load, and maximizing scalability are three key benefits of distributed computing.

3. http://www.internetworldstats.com/emarketing.htm

Proprietary distributed control systems (DCSs) have been available for many years. Typically used in the process industries, a DCS was traditionally complex and expensive.

Programmable logic controller (PLC) systems, meanwhile, added analog control to their original discrete control capabilities. But their system architecture remained centralized: a central controller scans all I/O, solves logic, and sets outputs.

Now, however, thanks to newer technologies such as Ethernet networking and the availability of smaller, less expensive processors, we can utilize the best of both the DCS and PLC worlds. We can leverage the advantages of distributed computing in a variety of automation applications, quickly, easily, and far more cheaply.

Instead of one PLC taking care of all control, multiple networked smaller processors—intelligent remote I/O—handle individual areas and communicate with a central controller. These smaller processors have automatic functions built into them that do not require programming, only simple configuration.

Why do you need multiple processors?

Minimizing Single Points of Failure

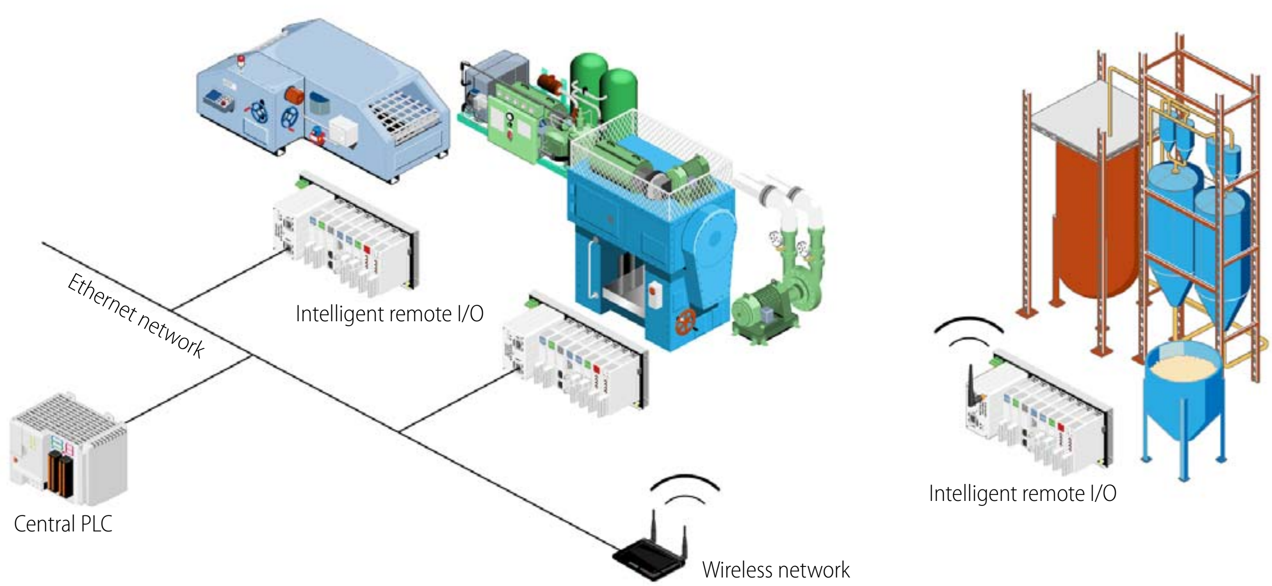
Given the availability of Ethernet cables and network infrastructures, it's tempting to just connect sensors and switches in a remote part of a building to an existing central controller. Why not?

When you're designing an automation system, it's important to look for the weakest link and make it stronger. Often the network is the weakest link.

Let's take the example of a temperature regulating application. Suppose you've installed a new refrigeration unit. You could hook the temperature sensor and the on/off switch for the chiller directly to your central controller. However:

- If the Ethernet cable connecting the refrigeration unit to the rest of the network is disconnected by accident, the temperature could become unstable and you could lose the contents of the unit.
- If the central controller is already busy doing many other tasks, the addition of the refrigeration unit could slow down processing or reduce network throughput, affecting all the other equipment and processes the controller is managing.

In other words, this architecture increases the odds of failure both at the new control area and at all the other areas controlled by that same, now busier, controller. It's like adding one more thing to the list of many things your favorite employee needs to do—other priorities might start to slip.



An example of a system utilizing intelligent remote I/O on both wired and wireless networks

Instead, we want to minimize single points of failure and bypass the weakest link by utilizing local control.

What could go wrong?

When looking at possible points of failure with remote I/O, think about how often and reliably that I/O is connected to the rest of the network.

For example, consider a wireless I/O system tasked with turning on a pump to fill a tank. If the wireless network goes down (for example, knocked out by weather), you might have a situation like this:

> *The central controller sends a command to the wireless remote I/O unit to turn on the pump to fill the tank. Fifty minutes later the controller sends a command to turn off the pump. If during those 50 minutes the connection to that wireless remote has gone down, the water will continue to flow, and overflow.*

For potentially unreliable remote connections like this one, you could look for a simple I/O *watchdog* feature on the remote I/O. If communication is lost, a watchdog sets outputs to a safe value—in this case, by turning off the pump.

Even better, instead of programming all the logic in the central controller, consider a built-in function at the I/O level that replaces programming in the controller—such as a pulse feature. Let the central controller delegate this task.

In this case you could have the central controller send a *Start On-Pulse* command, with a 50-minute pulse duration. With the pulse control occurring at the I/O level, the remote I/O will shut off the pump even if the I/O is cut off from the network during those 50 minutes. That's distributed control.

Spreading the Load

After looking at possible points of failure, take a look at spreading the load.

See what tasks you can offload from the main controller to intelligent remote I/O. Pay special attention to smaller tasks that haven't been completely automated yet, even though they're already part of your automation system.

Spreading the load lets the central controller use its processing power on supervisory tasks and overall task management, while local processors handle time-critical tasks locally and report back.

Counting each time a digital input goes on, keeping track of the maximum value of an analog signal coming into your system, and controlling a temperature loop (PID) are all tasks that can easily be done by a local processor that requires no programming.

It's easy to see the opportunity to automate a human task like an employee hand-counting the number of customers coming through the door. But consider how much programming may be required to capture that type of information in your automation system. While simple I/O can get the job done, intelligent I/O may be available to do it without requiring programming time.

You wouldn't hire extra staff to hand-crank a mimeograph machine rather than buy a photocopier. Similarly, you shouldn't hire someone to program a remote controller for simple tasks if you can reasonably buy intelligent remote I/O that already has those functions built in.

It makes sense to look for I/O that's intelligent enough to save time and expense on the programming side.

Built-in Remote I/O Functions

For applications involving analog control, look for the following built-in analog functions at the I/O level:

- PID loop control
- Minimum and maximum values
- Analog scaling
- Calibration: offset and gain
- Ramping
- Totalizing
- Engineering unit conversion
- Thermocouple linearization
- Temperature conversion
- Watchdog timeout
- Output clamping
- Filter weight

For digital control, look for these built-in functions:

- Input latching
- Quadrature counting
- High-speed counting
- Watchdog timeout
- Pulse generation and measurement
- Digital totalizing
- Time-proportional output (TPO)
- Frequency and period measurement

The ability to add all these functions in remote I/O with no programming and little impact on the overall system may well be the wisest delegation you've ever made.

Maximizing Scalability

Few systems are static. Most grow over time as I/O points and new functions are required or become desirable. New tanks or pumps may be needed for increased volume, for example. Or higher energy bills may make it desirable to add I/O for monitoring and controlling variable-speed motors.

And systems change. A production line for an obsolete product may be dismantled or modified for a new product. Replacement equipment may require additional sensors or controls.

With intelligent remote I/O, your system is much more scalable. Additions and changes are much easier.

- Remote I/O is located closer to the equipment it manages. Bus speed is faster; time-critical tasks are more easily and reliably accomplished.
- PLC scan times remain similar, rather than increasing with every I/O point added, because the remote I/O processor becomes the only item to be scanned.

With minimal disturbance to the central controller, intelligent remote I/O means you can add and change I/O as necessary and scale your system as required.

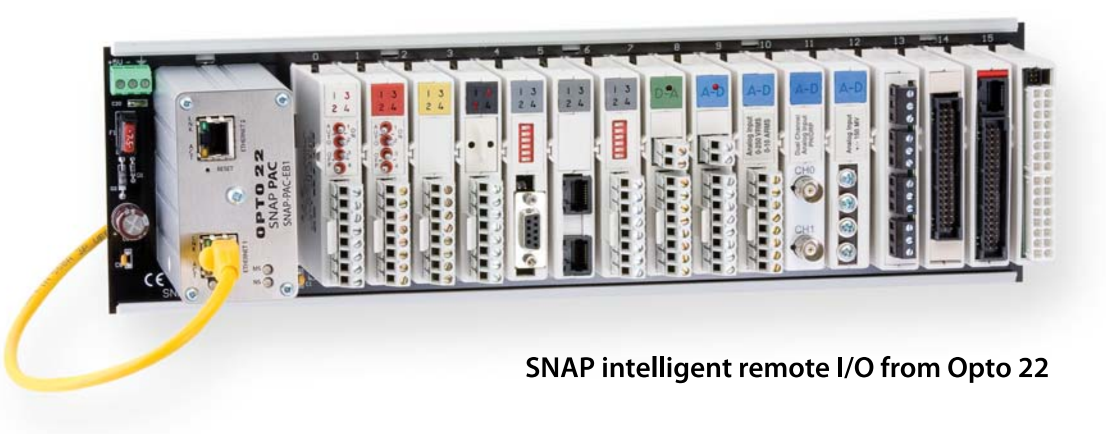
Smart I/O Options

We've touched on the tasks of controlling temperature and a simple turn on/off with a pulse at a remote location. The pulsing, watchdog, and counting features mentioned above are available at the I/O level without requiring programming. Many additional functions are available as well. Let's look at one of them: proportional-integral-derivative (PID) loop control.

In our remote refrigeration application described earlier, built-in PID loop control offers three advantages:

- **Reduces system load.** The remote I/O processor increases the overall computing power of the system, offloading a function that would otherwise have to happen on the central controller. Whether you have one PID loop or 90 PID loops in your system, each one's setpoint will be maintained, because they're all automatically calculated in remote processors.
- **Maintains key processes.** If someone accidentally severs a communication cable or a wireless connection goes down, PID loops on the remote I/O continue to control the temperature.
- **Saves programming time and expense.** Since the function is built in, there's no programming involved, just configuration of I/O points.

Functions that take the most processing power typically involve the heavy analog signal processing required by most process control applications. If remote I/O processors include built-in capabilities like ramping, thermocouple linearization, filtering a noisy signal, analog scaling, and minimum/maximum values, the main controller can focus on supervision and overall control. And your employees can tackle more interesting projects.



SNAP intelligent remote I/O from Opto 22

Example: Using intelligent remote I/O

So suppose you have an Allen-Bradley PLC system, for example, and you need to add additional I/O for new equipment. How would you utilize intelligent remote I/O?

First, locate reliable intelligent I/O that has the functions you need built in, for example tracking minimum and maximum values for analog temperature inputs and counting on/off states for digital inputs. Make sure this I/O is compatible with your automation system; for this example, it should be able to communicate using EtherNet/IP™, so it can talk to your A-B PLC without integration worries.

Next, install and wire the inputs to your sensors and actuators. Using a software tool provided by the I/O manufacturer, configure I/O as needed—for example, to turn on counting, to use an input signal that's different from the default, or to set a watchdog on the output.

Once the I/O is configured, the intelligent remote I/O will automatically track analog minimum and maximum values, and automatically count digital input states. All the PLC needs to do is to read the minimum and maximum values and the counts from the I/O.

Locating intelligent remote I/O

When you're looking for intelligent remote I/O, one good option is SNAP I/O™ and SNAP PAC brains from experienced automation manufacturer Opto 22. All Opto 22 products are manufactured and supported in the U.S.A., and product support is free.

- SNAP I/O comes in a wide variety of signal types and offers densities from two to 32 points per module. Up to 16 modules can be placed on one mounting rack.
- Each SNAP I/O module is tested twice. This I/O has proven so reliable that all solid-state modules are guaranteed for life.
- SNAP PAC brains are remote I/O processors that include all of the built-in analog and digital functions we've discussed. These brains mount on the same DIN-rail or panel-mounted rack as the I/O. Brains are available for serial, Ethernet, and wireless networks.
- Ethernet and wireless brains communicate natively with Modbus®/TCP and EtherNet/IP systems for easier system integration.

Contact Opto 22 with any questions you may have about using intelligent remote I/O in your application. Pre-sales engineering is free.

About Opto 22

Opto 22 was started in 1974 by one of the co-inventors of the solid state relay (SSR), who discovered a way to make SSRs more reliable. The company developed the red-white-yellow-black color-coding system for input/output (I/O) modules and the open Optomux® protocol, and pioneered Ethernet-based I/O.

Opto 22 is probably best known for its high-quality I/O and SSRs, both of which are manufactured and supported in the U.S.A. and guaranteed for life. In a time of financial uncertainty and increasing budget restraints, the company is especially attractive for its continuing policy of providing free product support, free training, free documentation, and free pre-sales engineering.