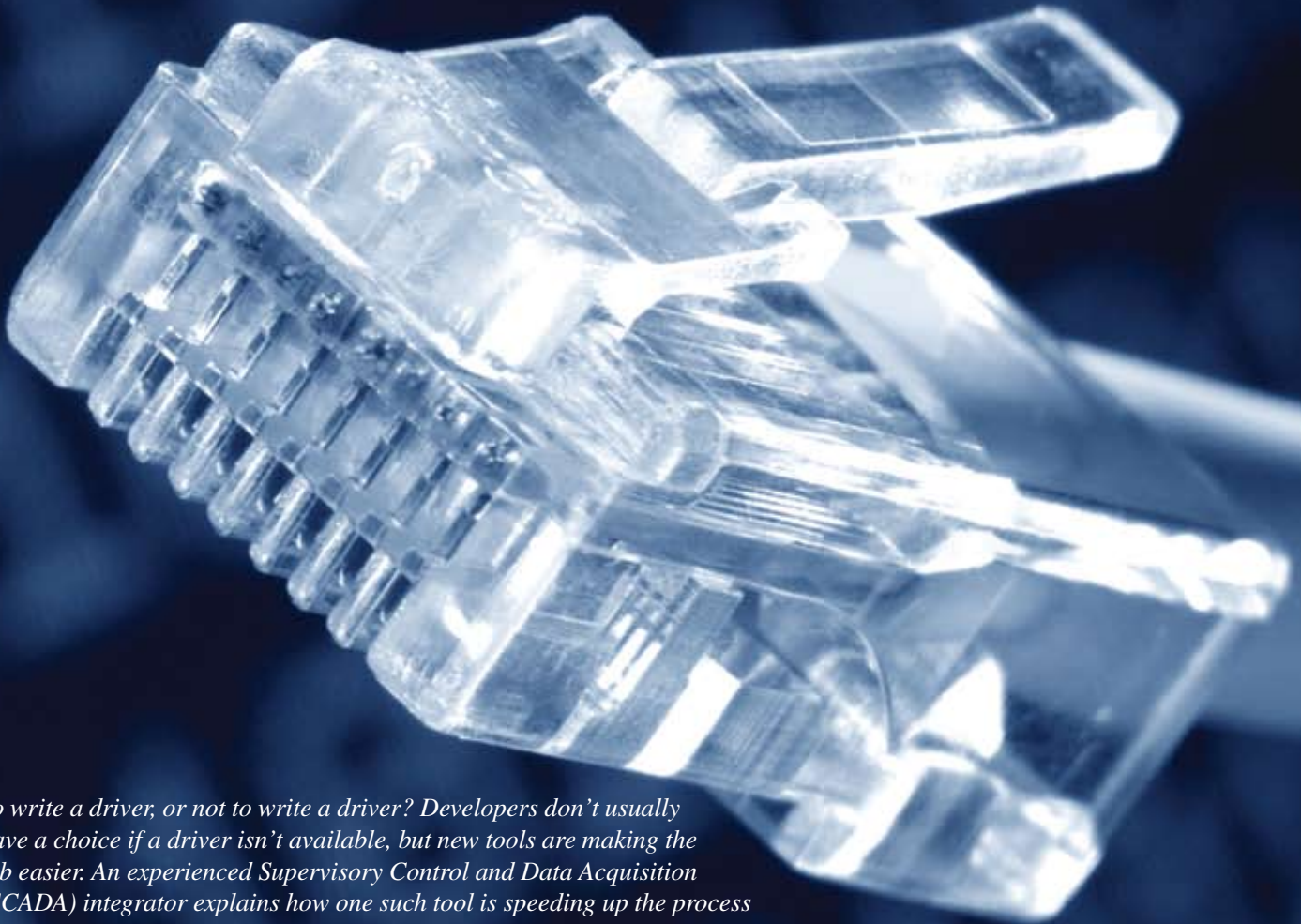


Integrating proprietary serial and Ethernet devices with SCADA/HMI systems

By Stephen Friedenthal



To write a driver, or not to write a driver? Developers don't usually have a choice if a driver isn't available, but new tools are making the job easier. An experienced Supervisory Control and Data Acquisition (SCADA) integrator explains how one such tool is speeding up the process of integrating OLE for Process Control (OPC) hardware.

When starting a SCADA project, developers must first identify the installed devices and controllers to determine how they will interface with the SCADA system. Most of the time, a programmable logic controller with one or more I/O drivers will be available. With OPC's growing popularity, hardware connectivity is becoming less difficult.

However, connecting to devices that lack an available driver often presents the greatest challenge. This is especially a problem with devices that have proprietary serial and Ethernet protocols such as electronic scales, particle counters, controllers, and so forth. Developers traditionally resolved this issue by writing an I/O driver from scratch.

Consequently, developers would do well to heed this advice: Don't bid fixed price. Creating a robust and reliable driver is not trivial, as it requires a keen understanding of hardware and software interfaces and error modes. Additionally, as OPC becomes an industry standard, it only makes sense to undertake the effort if the driver supports the OPC protocol.

Creating drivers without writing them

If developers don't have the time or expertise to write their own drivers, they should consider using the Kepware User-Configurable (U-CON) driver, which has proven useful in two recent development projects. Figure 1 shows the U-CON transaction editor.

U-CON is a universal translator between the proprietary serial world and OPC. With the U-CON driver, developers can easily create OPC data item tags that represent almost any value from the serial device. As with other OPC devices, tags can be read-only or read-write and consist of any data type.

The U-CON transaction editor makes this possible by allowing developers to create the command structure for a tag as a series of simple steps. For example, the screen shot in Figure 1 shows the command structure for reading a temperature probe. The transaction steps are generated using a menu system to create an easy-to-read state machine, as shown in Figure 2.

One of the transaction editor's powerful features is its ability to create global functions that all the tags can reuse. This reduces development time and allows changes to be made in one location for all the tags.

The U-CON driver also enables developers to write the transactions using a device ID variable as opposed to hard-coding the ID for each new device.

U-CON in action

EVSystems developers used the U-CON driver in two recent projects that required integrating GE Fanuc's iFIX SCADA product with legacy devices using a proprietary RS-485 protocol.

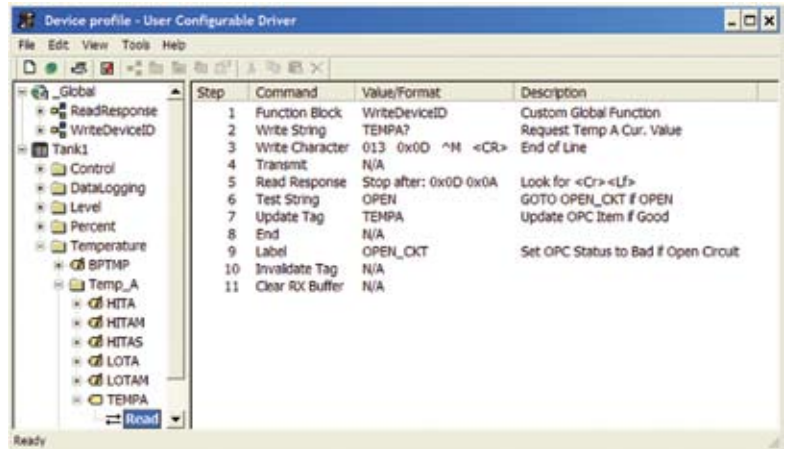


Figure 1

A feature common to all Kepware drivers is that all the tags are available via OPC as well as the native *nio* interface in GE iFIX. The latter approach is particularly useful because data can be read directly in iFIX without configuring any OPC items, significantly reducing configuration and validation efforts.

The first application involved 50 cryogenic tank controllers, each with an RS-485 connection and its own ID. Adding a new tank to the U-CON driver was as simple as duplicating an existing tank and changing the ID property. In total, the cryogenic tank driver took developers no more than a week to write, with half of that time spent learning the protocol and gaining familiarity with the product. (Famous last words: "We don't need to read the manual." Sometimes, it helps.)

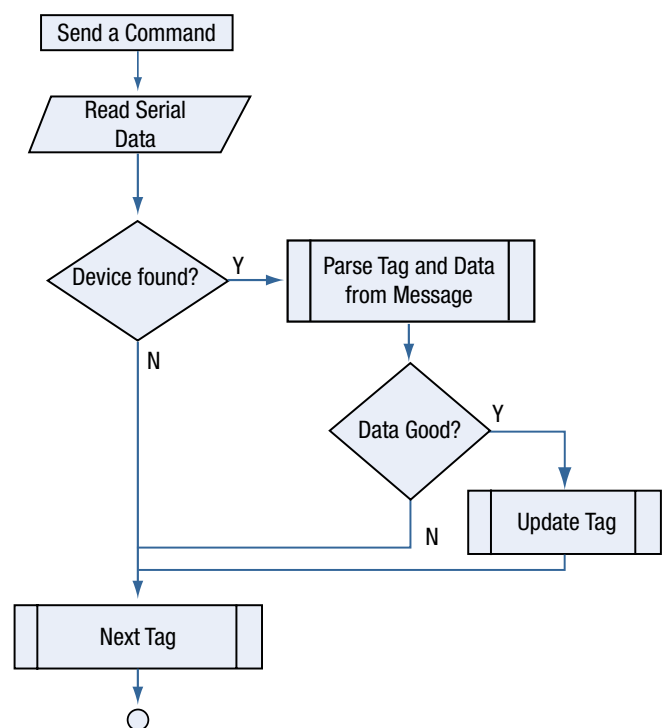


Figure 2



Figure 3

Most recently, the development team used the U-CON driver to talk to a dozen Met One particle counters (see Figure 3). Though the connection remained RS-485, the protocol was entirely different, with a mix of hexadecimal and ASCII components. Nonetheless, total driver development time took about three days.

When developers visited the site to install and commission the system, they discovered that they could not use the internal sample timers in the particle counters. Instead, they had to redesign the driver so that it would command the particle counters to stop, return a record, and start a new count every 10 minutes. Although this was a radical departure from the original design, it took the development team less than a day to implement. A custom driver with code would have required much more effort.

Weighing trade-offs

While using the U-CON driver can be effective in certain applications, writing an I/O driver may be worth the effort in other applications. When approaching each project, developers should consider a few critical questions:

- Is high-speed data processing a requirement?
- How complex is the protocol?
- Is it important to have a branded or proprietary interface?
- Which HMI/historian or client applications will need access to the data, and should they access the data via open standards or a proprietary API?
- Does the driver need to be distributed via licensing or royalty-free?
- Do you love writing code (and reading manuals)?

The answers to these questions can help guide developers to the best solution. Using U-CON can prove advantageous for most proprietary serial and Ethernet protocols.

Cutting integration time

OPC's growth has enabled system integrators to easily integrate most programmable logic control and distributed control systems with HMI/SCADA systems. However, many legacy and proprietary systems still do not have an off-the-shelf driver. Previously, the system integrator's only option in this situation was to write a custom driver at a significant cost.

The Kepware U-CON driver introduces a new option that can significantly reduce the time spent and expense paid to integrate a serial device. Using this driver, development time can drop from an estimated six-plus weeks per project to a few days. **IES**



Stephen Friedenthal is president of EVSystems Data Solutions in Newton Centre, Massachusetts. He has more than 20 years of experience working with instrumentation and control systems and developing software for industrial applications. Prior to founding EVSystems, he worked as a product manager for GE Infrastructure and managed the GE Proficy Historian. Stephen has a BS in Nuclear Engineering and an MS in Engineering Management from MIT.

EVSystems Data Solutions

617-916-5101

sfriedenthal@evsystems.net

www.evsystems.net