# Case Study: Using IEC 61850 Methods for RTU Replacement and Distributed Automation

Robin Jenkins and David Dolezilek, *Schweitzer Engineering Laboratories, Inc.*

*Abstract*—**This paper chronicles a recent test of replacing pad-mounted switch RTUs with IEC 61850-compatible scalable, programmable automation controllers (PAC) by a utility distribution group. New functionality was added and tested, including wireless Ethernet connections between the switchgear and remote control center as well as peer-to-peer communications between switchgear locations to support remote automation with coordinated protection and autorestoration schemes. This paper discusses the three types of switches and the variation in I/O requirements based on switch type, scalable configuration, and mounting features to support technician access and troubleshooting.**

**The switchgear are an integral part of the utility's 15 kV underground distribution expansion. The increase in the number and size of switchgear make remote monitoring and control, local automation, peer-to-peer coordination, and remote engineering access critical for system reliability. To meet all their requirements, the utility chose and tested a solution that includes I/O support for multiple CT and PT inputs, battery monitoring capabilities, a wide range of discrete and analog input types, and control outputs. Along with a variety of I/O, an easily scalable solution was necessary to support the different I/O requirements for each switch type. The final piece of the system includes integration capabilities with the utility's radio communications architecture, including serial and Ethernet connections for SCADA, remote engineering access, and peer-to-peer communications.**

**This project was the utility's first experience with IEC 61850. This paper discusses the process that was developed to engineer, configure, and diagnose IEC 61850 GOOSE messages communicating status, alarms, analog measurements, and controls.**

## I. INTRODUCTION

With over 400 pad-mounted switches in service and projections for additional installations estimated at 50 to 100 per year, this large electric utility determined that a new pad-mounted switch RTU standard was crucial to meeting their existing and future requirements for remote monitoring, control, and automation of their 15 kV underground distribution system.

Existing installations included an RTU providing basic monitoring and control capabilities. In early 2006, the utility was informed by the RTU supplier that they would no longer manufacture or support this product. This situation, along with the goal to reduce RTU configuration, startup, and maintenance costs while improving reliability, motivated the utility to develop a new standard for their pad-mounted switch RTUs. Enhancements to the standard included a scalable architecture, easily configured to match the different switch types and configuration options, Ethernet connectivity, DNP/TCP protocol support, local and remote engineering access, and logic and communications capabilities to support local and distributed protection and automation schemes.

The objective was to take advantage of new technologies to improve the underground distribution system's reliability by reducing outage duration and frequency as well as RTU development and maintenance costs.

### *A. Pad-Mounted Distribution Switch Application*

The 15 kV underground distribution system switch requirements vary depending on the application at a specific location. The number of switch positions (ways) and the type of switch (SF6, air break, or vacuum) determine the switch type used for a specific installation. One installation may require a one-way switch, while another may require up to a six-way switch. The utility chose between three different switch manufacturers, each providing different selections for ways and I/O configuration.

Each individual pad-mounted switch includes an RTU and serial radio communicating to the remote SCADA system. The RTU and radio provide the control center operator with monitoring and control capabilities via serial DNP protocol. Existing RTUs monitor analog and status inputs and respond to remote controls from the SCADA master.

With the rapid expansion of pad-mounted switch installations, the integration of multiple switch types and configurations, and the future implementation of advanced communications and automation schemes, the utility explored replacing RTUs with programmable automation controllers (PACs), which perform all RTU functions and have these additional features:

- Scalable architecture to support different I/O requirements, depending on switch type
- Sequential Events Recorder (SER), also known as Sequence of Events (SOE)
- Programmable logic to support local automation schemes
- Peer-to-peer communications to support future automatic isolation, restoration, source transfer, and loop schemes
- SCADA access to device internal diagnostics
- Ethernet interface for future DNP/TCP communications

- Remote or local engineering access to retrieve event reports, data logs, and/or modify settings or reports
- Technician access for troubleshooting and maintenance
- Single software suite for remote or local configuration

### *B. RTU Requirements*

To meet application and configuration requirements for any switch installation, the utility standardized on three manufacturers supporting ten unique configurations. Switch manufacturers and types include the following:

- S&C PME 3, 9, 10, and 11 (air-break switches)
- S&C Vista 330, 440, and 422 (SF6 switches)
- Trayer 4, 5, and 6 (vacuum switches)

Each switch offers different I/O configurations. These include the following:

- PME installations support 1 to 4 ways and include up to 4 CT/PT connections, 24 digital inputs (DIs), 16 digital outputs (DOs), and 8 analog inputs (AIs).
- Vista installations support 2 to 4 ways and include up to 4 CT/PT connections, 32 digital inputs, 16 digital outputs, and 8 analog inputs.
- Trayer installations support 4 to 6 ways and include up to 6 CT/PT connections, 40 digital inputs, 24 digital outputs, and 8 analog inputs.

Because installations vary depending on switch type and configuration, a distributed architecture was incorporated, allowing the utility to match the RTU configuration to the specific switch configuration. This distributed design also helped meet the 18” X 16” X 10” RTU footprint in the switch control cabinet.

For this case study, the utility selected the PME 10 switch. This four-way air-break switch includes the following configuration:

- Four PT connections
- Four CT connections
- 24 Digital inputs
- 16 Digital outputs
- 2 Analog inputs

SCADA monitor and control points from this switch include the following:

- Four single-phase currents
- Four single-phase voltages
- Three-phase watts and vars
- Power factor
- Frequency
- Battery voltage
- Battery charger output voltage
- 24 Digital inputs (18 SER, 6 status)
- 16 Digital outputs

### *C. Innovative and Flexible Network of Components*

This case study demonstrates the process of using hardened Ethernet components and standardized communications to construct a compact but distributed RTU. This method is useful for any RTU replacement and is demonstrated to be cost effective as well. Further, this design provides new operations and maintenance features that will become standard requirements in the future. Although the new data acquisition methods promise to reduce overall labor, they also dramatically change the technology used for system configuration, installation, commissioning, and maintenance. This paper demonstrates the new technologies and processes that provide an understanding of unseen data flow inside the communications network and message transfer configuration.

## II. PREVIOUS RTU DESIGN REPLACED

Prior to the new design, pad-mounted RTUs were procured from SCADA vendors that relied on traditional monolithic and centralized hardware designs. The CPU and I/O boards all shared a common power supply and physical proprietary communications bus interconnected with ribbon cables or edge connectors. The move away from this traditional design is evident in the distributed designs of the communications industry, process control industry, and now the computing industry. The RTU-replacement design incorporates this new distributed architecture, based on Ethernet communications and internationally standard protocols. This case study demonstrates a tightly integrated and localized RTU to fit the space requirements of the pad-mounted switchgear; however, it is clear from the integrated communications that the hardware components can be located remotely from one another and perform equally well. Further, this case study demonstrates that it is now very easy to integrate I/O from other existing IEDs, such as relays, using peer-to-peer communications. This data integration does not impact SCADA or protection communications and coexists on the same communications local-area network (LAN).

## III. BUSINESS CASE OVERVIEW

The customer has numerous sites to add and many to retrofit. This project was more cost sensitive than most, simply because of the sheer quantity of devices needed.

### *A. Cost Savings Over Existing Application*

The many retrofits are necessary due to longevity and quality issues of the in-service RTUs. Therefore, reliable hardware was also a priority. Reliability, or lack thereof, is actually also a cost issue because maintenance and field replacements create out-of-service costs and labor and hardware expenses. Also, distribution circuits are measured for their various performance indices; and poor reliability directly and proportionally impacts the number and duration of outages. This new innovative method will provide dramatic cost savings due to the following:

- PACs may be programmed to perform both SCADA and distribution automation (DA) functions, eliminating the need to install and maintain several devices.
- The use of industry-standard protocols, such as IEC 61850 and DNP3, facilitates communication between

- relays and other manufacturers' devices and permits application of the most cost-effective hardware.
- The same hardware design can be applied to various DA switch arrangements and voltage classes, reducing time for engineering and testing.
- Space requirements are minimized because PAC units are incrementally added as needed to support the I/O requirements.
- Wiring can be kept the same as the previous generation design with the use of wiring harnesses. Each type of switch can also be satisfied with a universal wiring harness that is built at the vendor facility. Harnesses prewired to the I/O enable the field wiring to match the customer-specific desires. In this case, field I/O terminations are identical for each switch type.
- Integrated network diagnostics tools used for commissioning and/or troubleshooting quickly identify and isolate communications issues.

### *B. Cost Comparison of Available Products*

The RTU previously used in this application was no longer available from the manufacturer. Most other offerings did not meet the I/O, temperature range, or reliability requirements. Traditional substation RTUs meet the I/O requirements, but not size, temperature range, or reliability criteria. Also, the traditional RTU does not calculate watts, vars, and power factor, so these values must be provided via external transducers. The PAC solution meets all criteria and offers previously mentioned additional features.

Although every system is different, a traditional RTU for this specific system will cost about $14,000 and require 12 transducers at $400 each for a total hardware cost of $18,800. The PAC system includes HMI and diagnostic tools at no additional cost and includes the Ethernet network equipment. This Ethernet network provides much more connectivity than simply RTU replacement and supports peer-to-peer distribution automation. The PAC solution used in this example cost $11,223.

## IV. CHOOSE AND VERIFY PAC HARDWARE

A single PAC satisfies numerous unique installations due to its several optional internal contact I/O cards, including the following:

- Analog inputs for dc process values
- Digital inputs to sense status of contacts
- Digital outputs to control external devices
- Combination of digital inputs and outputs
- Combination of analog inputs and outputs
- AC voltage inputs
- AC current inputs
- Combination of ac current and ac voltage inputs
- RTD inputs

Fig. 1 illustrates the functional logic diagram of a PAC network. The PAC CPU collects field data from local I/O cards, serial connections, and network connections. It then processes these data based on settings and logic, creates databases, and shares these data via serial and network connections.

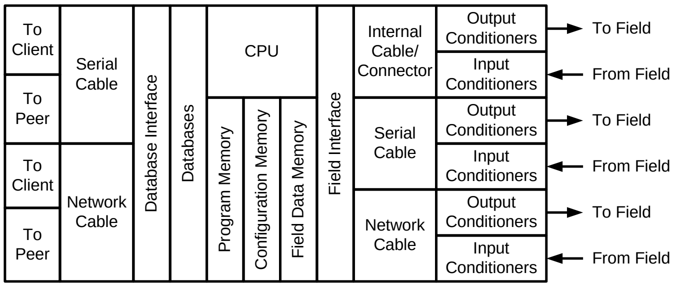


Fig. 1. PAC Network Functional Logic Diagram

Additional combinations of digital inputs and outputs are easily added locally or remotely via a MIRRORED BITS® peer-to-peer serial connection to a remote I/O module. Even more I/O combinations are satisfied when two or more PACs are networked together using IEC 61850 GOOSE messages over Ethernet.

In order to satisfy each specific installation, the corresponding PAC system hardware must be procured after the design phase indicates the required I/O. Each PAC has an HMI for local display and control, a CPU for communications and logic processing, and card slots to accept numerous I/O combinations. Based on the choice of I/O, the number of required cards dictates the number of PACs that will be connected together to create the RTU replacement. This flexibility, and the use of peer-to-peer communications among PACs, supports the replacement of any RTU configuration.

## V. PAC COMMUNICATIONS

Within traditional RTUs, the I/O is polled over the communications bus at a fixed rate using a proprietary message system. Some more elaborate RTUs also detect digital inputs when they change state. IEC 61850 GOOSE messages essentially combine both methods. Each GOOSE message has a customized group of data within it, called a dataset. At a minimum, these data are published from the PAC at the slowest frequency, set by a fixed, configurable rate. Each GOOSE message is also published whenever one or more of the contents of the dataset changes. Any time a digital point changes state or an analog change is larger than the associated reporting dead band amount, a new GOOSE message is created and published. After the change, this new GOOSE message is published more frequently than the fixed rate and then slows in frequency until the rate equals the fixed rate. This process repeats whenever the contents change. This messaging method is very effective for numerous distributed and peer-to-peer applications. Here we use it to concentrate I/O from several PACs into one, which becomes the DNP3 server.

The communications methods described in this case study are also used to share data among any quantity and type of IEDs in addition to, or instead of, PACs. Using GOOSE messages, it is possible to combine I/O from other IEDs as well. Therefore, in a substation or industrial process network, I/O can be collected from devices installed for other purposes like protection and control. Fig. 2 illustrates the RTU application built using I/O cards distributed among several PACs.

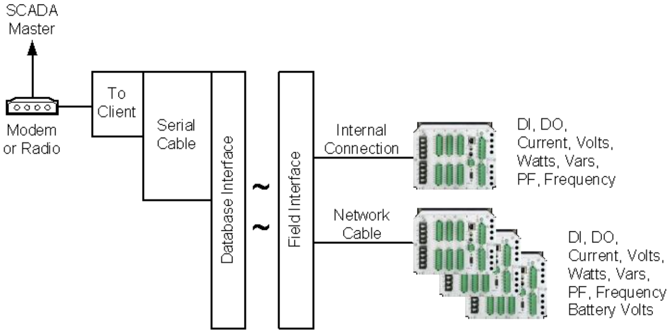


Fig. 2. RTU Replacement Using Distributed I/O

Four PACs with various combinations of I/O cards satisfy every switch configuration in this distribution utility. The PAC_MASTER acts as the data concentrator and combines its own local I/O with I/O from the other three PACs received via GOOSE across the network. The PAC_MASTER communicates these data via DNP3 to the SCADA master. It also responds to control commands from SCADA via DNP3. The PAC_MASTER drives its local digital outputs in response to these DNP3 commands. It also uses a GOOSE publication to distribute commands received via DNP3 that are destined for output cards on the other PACs. The other three PACs that make up the RTU are named PAC_SLAVE_A, PAC_SLAVE_B, and PAC_SLAVE_C. The flexibility of this PAC supports mapping all of the networked data into a single internal DNP3 map and address that match the existing SCADA requirements. In this way, new and retrofit installations support the exact datamap and address expected by the existing SCADA master. Fig. 3 illustrates the network of PACs acting as an RTU.

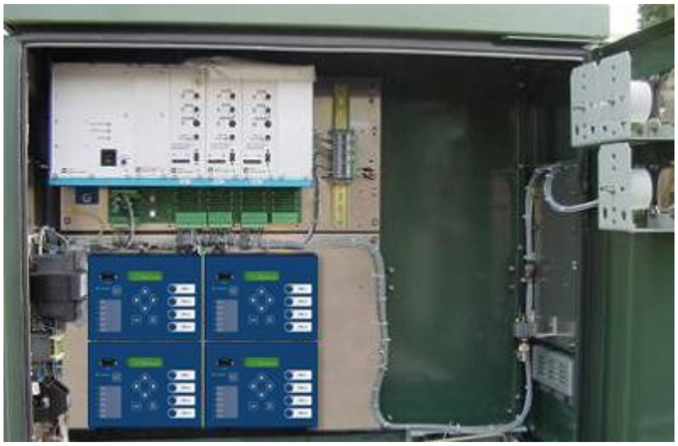
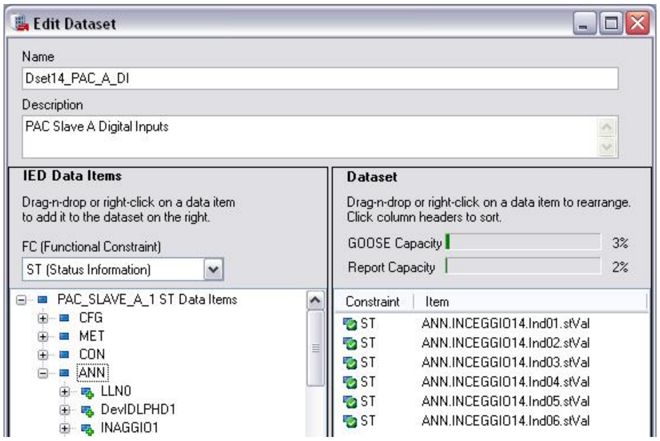


Fig. 3. PAC Network Acting as an RTU

## A. Configure PAC GOOSE Message Contents

Using PAC configuration software, the utility easily configures GOOSE message contents to move only required data, chosen from a comprehensive internal IED database. The message contents map to a dataset associated with a GOOSE message. All types of data can be put into a single dataset; however, only digital and analog values are communicated in this RTU replacement. Also, for consistency and simplicity of testing and commissioning, the digital and analog data are separated into different GOOSE messages. Fig. 4 illustrates a digital dataset being created for PAC_SLAVE_A, and Fig. 5 illustrates an analog dataset being created for PAC_SLAVE_A. As needs change in the future, these datasets are easily changed at any time to reflect new I/O added to the PAC and/or new calculations and logic process results. Therefore, the RTU replacement has the flexibility during installation and into the future to fit all applications as requirements change and expand.



Name: Dset14_PAC_A_DI

Description: PAC Slave A Digital Inputs

Drag-n-drop or right-click on a data item to add it to the dataset on the right.

FC (Functional Constraint): ST (Status Information)

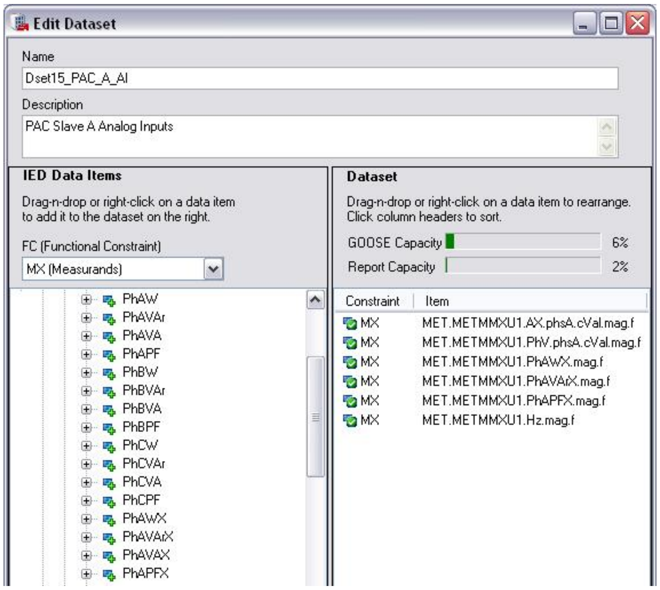
Drag-n-drop or right-click on a data item to rearrange. Click column headers to sort.

GOOSE Capacity 3%

Report Capacity 2%

| Constraint | Item |
|---|---|
| ST | ANN.INCEGGIO14.Ind01.stVal |
| ST | ANN.INCEGGIO14.Ind02.stVal |
| ST | ANN.INCEGGIO14.Ind03.stVal |
| ST | ANN.INCEGGIO14.Ind04.stVal |
| ST | ANN.INCEGGIO14.Ind05.stVal |
| ST | ANN.INCEGGIO14.Ind06.stVal |

Fig. 4. Digital Dataset Being Created for PAC_SLAVE_A



Name: Dset15_PAC_A_AI

Description: PAC Slave A Analog Inputs

Drag-n-drop or right-click on a data item to add it to the dataset on the right.

FC (Functional Constraint): MX (Measurands)

Drag-n-drop or right-click on a data item to rearrange. Click column headers to sort.

GOOSE Capacity 6%

Report Capacity 2%

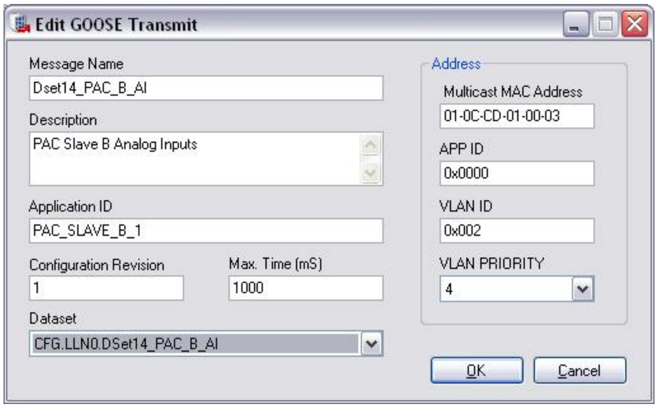
| Constraint | Item |
|---|---|
| MX | MET.METMMXU1.AX.phsA.cVal.mag.f |
| MX | MET.METMMXU1.PhV.phsA.cVal.mag.f |
| MX | MET.METMMXU1.PhAWX.mag.f |
| MX | MET.METMMXU1.PhAVArX.mag.f |
| MX | MET.METMMXU1.PhAPFX.mag.f |
| MX | MET.METMMXU1.Hz.mag.f |

Fig. 5. Analog Dataset Being Created for PAC_SLAVE_A

## B. Configure PAC GOOSE Message Publication

The fixed rate publication of GOOSE messages when the dataset contents do not change acts as a heartbeat. Since the publisher can never receive positive acknowledgement that the subscribers each received the GOOSE message, frequent receipt helps the subscriber recognize that the publisher is active and functioning properly. This method is less than optimal for time-critical interlocking, protection, and automation. It is improved with the time-to-live (TTL) value. Each time a message is published because of a state change or because the maximum delay timer times out, the message includes a TTL value for the subscriber. This time tells the subscriber the maximum amount of time until another GOOSE message is published. This is a configurable value so that the network can be tuned for devices or LAN components that drop messages. In this example with four devices and a substation-hardened switch, it is unlikely that any messages will be interrupted, so we made the maximum time a full second. Each PAC GOOSE message is given a unique multicast address and assigned a dataset. Fig. 6 illustrates a GOOSE message publication being configured within PAC_SLAVE_B. The publication parameters are as follows:

- Message Name is Dset14_PAC_B_AI. This name identifies this message to devices that would like to subscribe to it.
- Description is PAC Slave B Analog Inputs. This description documents that the purpose is to move analog values from PAC_SLAVE_B.
- Application ID is PAC_SLAVE_B_1. This is the human-readable identifier of the source IED.
- Max. Time (ms) is 1000. This maximum delay time is used to calculate the TTL value published within the GOOSE message. It is also the GOOSE message fixed rate.
- Dataset is CFG.LLNO.DSet14_PAC_B_AI. This is the internal label of the dataset that satisfies the description.
- Multicast MAC Address is 01-0C-CD-01-00-03. This address is used as the first step for subscribers to decide if they are configured to accept and process each received GOOSE message.



Message Name: Dset14_PAC_B_AI
Description: PAC Slave B Analog Inputs
Application ID: PAC_SLAVE_B_1
Configuration Revision: 1
Max. Time (mS): 1000
Dataset: CFG.LLN0.DSet14_PAC_B_AI
Multicast MAC Address: 01-0C-CD-01-00-03
APP ID: 0x0000
VLAN ID: 0x002
VLAN PRIORITY: 4

Fig. 6. GOOSE Message Publication Configuration for PAC_SLAVE_B

## C. Configure PAC GOOSE Message Subscription

Each PAC is configured to subscribe to the appropriate GOOSE messages to be published by the other PACs. In this case, the PAC_MASTER is subscribing to GOOSE messages containing analog values from each of the other PACs. It is also subscribing to a GOOSE message containing digital values from PAC_SLAVE_A. Fig. 7 illustrates the four PACs within the project **RTU_Replacement_02** in the left pane of the configuration software. The **GOOSE Receive** pane lists all of the GOOSE messages available for subscription. The fourth GOOSE message is expanded to illustrate the contents available for use. These contents reflect the dataset similar to the one illustrated in Fig. 4.

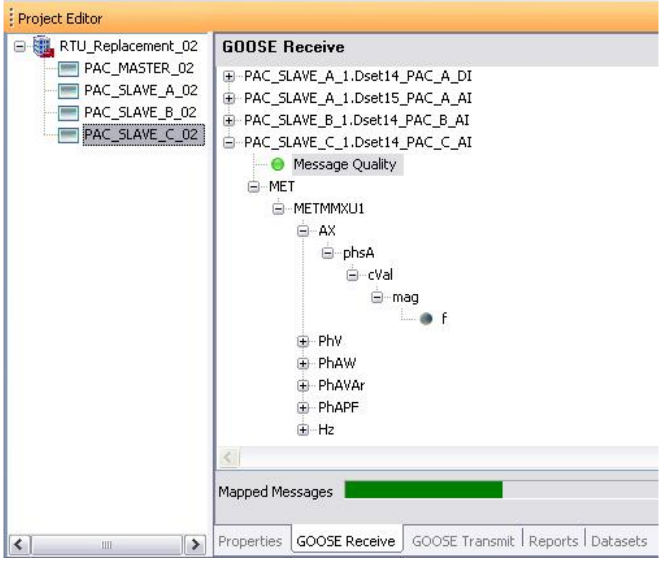
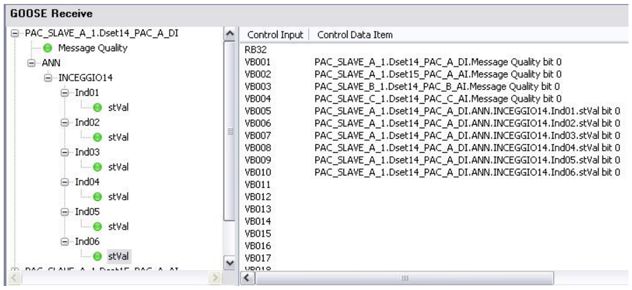


Fig. 7. GOOSE Message Subscription Configuration

The PAC configuration software automatically creates subscription settings associated with the human-readable GOOSE message label. It also automatically creates datamap settings in response to manual drag and drop selections that associate received dataset contents with internal PAC control points. These control points include different types of logic bits, such as remote bits (RB) and virtual bits (VB), as well as analog values such as remote analog (RA) points. Fig. 8 illustrates the mapping function that associates contents of the received GOOSE message to the controllable inputs within the PAC_MASTER. In this illustration, the quality of each incoming GOOSE message is mapped to a virtual bit, VB001–VB004. Also, the six incoming digital inputs from PAC_SLAVE_A are also mapped to virtual bits. When viewed in color, the **GOOSE Receive** pane indicates that each possible data element has been mapped because the circle icon to the left of each has changed to green. In this case, the Message Quality and the status value (stVal) of each of the six digital input indications (Ind01–Ind06) are Control Data Items mapped to Control Inputs.

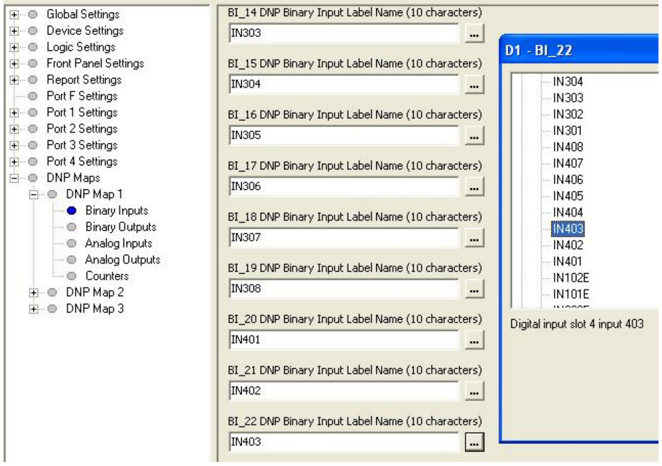


| Control Input | Control Data Item |
|---|---|
| RB32 | |
| VB001 | PAC_SLAVE_A_1.Dset14_PAC_A_DI.Message Quality bit 0 |
| VB002 | PAC_SLAVE_A_1.Dset15_PAC_A_AI.Message Quality bit 0 |
| VB003 | PAC_SLAVE_B_1.Dset14_PAC_B_AI.Message Quality bit 0 |
| VB004 | PAC_SLAVE_C_1.Dset14_PAC_C_AI.Message Quality bit 0 |
| VB005 | PAC_SLAVE_A_1.Dset14_PAC_A_DI.ANN.INCEGGIO14.Ind01.stVal bit 0 |
| VB006 | PAC_SLAVE_A_1.Dset14_PAC_A_DI.ANN.INCEGGIO14.Ind02.stVal bit 0 |
| VB007 | PAC_SLAVE_A_1.Dset14_PAC_A_DI.ANN.INCEGGIO14.Ind03.stVal bit 0 |
| VB008 | PAC_SLAVE_A_1.Dset14_PAC_A_DI.ANN.INCEGGIO14.Ind04.stVal bit 0 |
| VB009 | PAC_SLAVE_A_1.Dset14_PAC_A_DI.ANN.INCEGGIO14.Ind05.stVal bit 0 |
| VB010 | PAC_SLAVE_A_1.Dset14_PAC_A_DI.ANN.INCEGGIO14.Ind06.stVal bit 0 |
| VB011 | |
| VB012 | |
| VB013 | |
| VB014 | |
| VB015 | |
| VB016 | |
| VB017 | |

Fig. 8. GOOSE Message Dataset Mapping to Internal PAC Control Inputs

## D. Configure PAC DNP3 Map

The PAC_MASTER is configured to provide the locally acquired inputs, and those acquired through GOOSE subscription, via its DNP3 server. Fig. 9 illustrates the DNP3 map configuration of locally acquired digital inputs from the digital inputs card in Slot 4. The tool allows direct entry of point names or selection from the pick list as shown here.

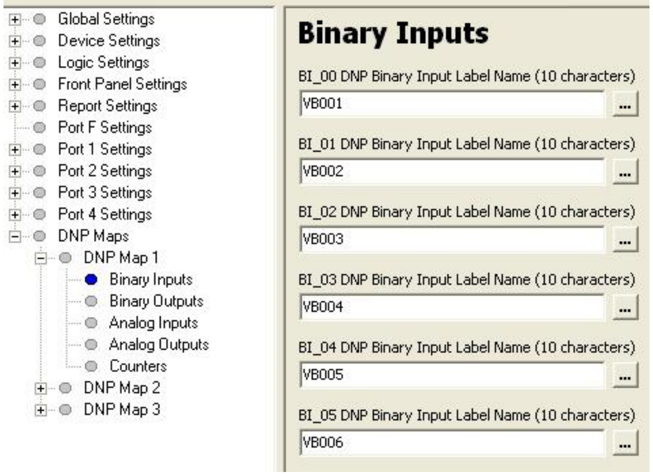


BI_14 DNP Binary Input Label Name (10 characters): IN303
BI_15 DNP Binary Input Label Name (10 characters): IN304
BI_16 DNP Binary Input Label Name (10 characters): IN305
BI_17 DNP Binary Input Label Name (10 characters): IN306
BI_18 DNP Binary Input Label Name (10 characters): IN307
BI_19 DNP Binary Input Label Name (10 characters): IN308
BI_20 DNP Binary Input Label Name (10 characters): IN401
BI_21 DNP Binary Input Label Name (10 characters): IN402
BI_22 DNP Binary Input Label Name (10 characters): IN403

Digital input slot 4 input 403

Fig. 9. DNP Map Configuration of Local Digital Inputs Within PAC_MASTER

Fig. 10 illustrates the DNP3 map configuration of remotely acquired digital inputs. In this example, the recently subscribed control point virtual bits, VB001–VB005, are mapped to DNP binary inputs (BI). Using similar methods, other local, remote, and virtual bits are mapped into the DNP3 map.

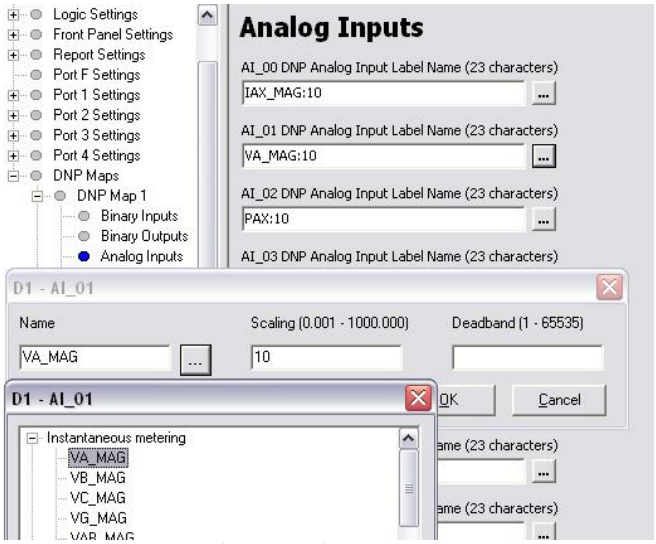


**Binary Inputs**

BI_00 DNP Binary Input Label Name (10 characters): VB001
BI_01 DNP Binary Input Label Name (10 characters): VB002
BI_02 DNP Binary Input Label Name (10 characters): VB003
BI_03 DNP Binary Input Label Name (10 characters): VB004
BI_04 DNP Binary Input Label Name (10 characters): VB005
BI_05 DNP Binary Input Label Name (10 characters): VB006

Fig. 10. DNP3 Map Configuration of Remotely Acquired Digital Inputs Within the PAC_MASTER

Mapping analog inputs, either by name or pick list, also configures the scale factor and customized reporting dead band. Fig. 11 illustrates local A-phase voltage magnitude being mapped to DNP3 analog inputs Index 25 with a scale factor of 10. The value is chosen from the **Instantaneous metering** pick list (see Fig. 11) and the default reporting dead band is used when the entry field is left empty.



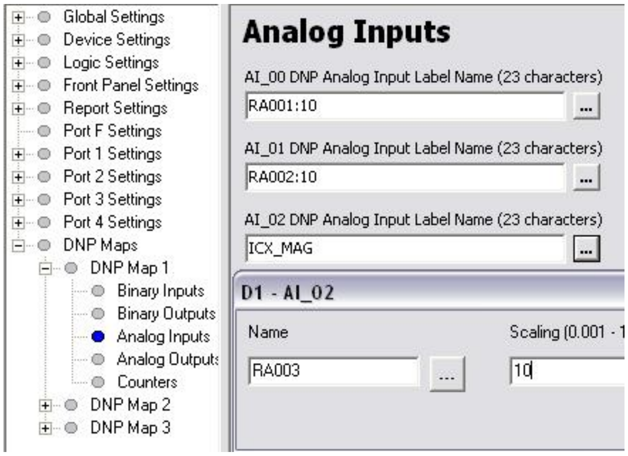
**Analog Inputs**

AI_00 DNP Analog Input Label Name (23 characters): IAX_MAG:10
AI_01 DNP Analog Input Label Name (23 characters): VA_MAG:10
AI_02 DNP Analog Input Label Name (23 characters): PAX:10
AI_03 DNP Analog Input Label Name (23 characters)

Name: VA_MAG
Scaling (0.001 - 1000.000): 10
Deadband (1 - 65535):

Fig. 11. DNP3 Map Configuration of Local Analog Inputs Within PAC_MASTER Showing Simplicity of Using Scaling and Pick Lists

Fig. 12 illustrates remote analogs RA001–RA005, which are GOOSE contents received into the PAC_MASTER from another PAC, being mapped to analog input registers in the PAC_MASTER DNP3 map.



**Analog Inputs**

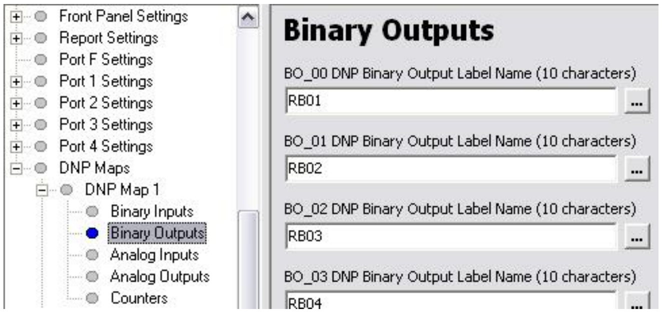
AI_00 DNP Analog Input Label Name (23 characters): RA001:10
AI_01 DNP Analog Input Label Name (23 characters): RA002:10
AI_02 DNP Analog Input Label Name (23 characters): ICX_MAG

Name: RA003
Scaling (0.001 - 1: 10

Fig. 12. DNP3 Map Configuration of Remote Analogs as Analog Inputs Within PAC_MASTER via a Pick List

## *E. Configure PAC DNP3 Controls*

Within the PAC_MASTER, incoming DNP3 commands are automatically mapped to remote bits and used locally or passed to other PACs as GOOSE publications. Fig. 13 illustrates the DNP3 controls being mapped directly to remote bits within the PAC_MASTER.

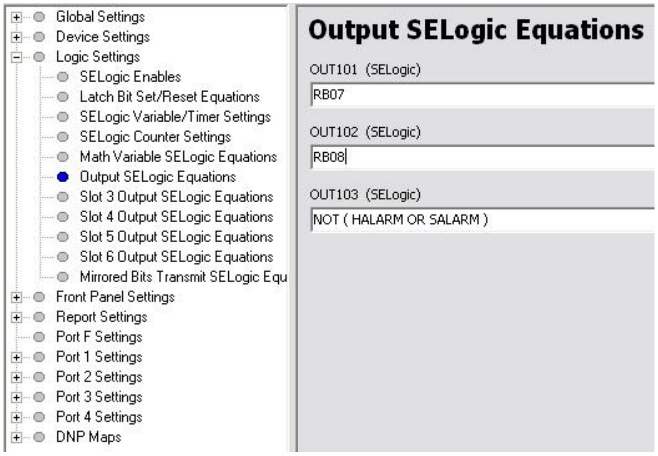


**Binary Outputs**

BO_00 DNP Binary Output Label Name (10 characters): RB01
BO_01 DNP Binary Output Label Name (10 characters): RB02
BO_02 DNP Binary Output Label Name (10 characters): RB03
BO_03 DNP Binary Output Label Name (10 characters): RB04

Fig. 13. DNP3 Controls Mapped to Remote Bits in the PAC_MASTER

Fig. 14 illustrates remote bits associated with DNP3 controls being mapped to two of three output contacts provided on the PAC base unit. The third output contact is used as a security and self-test alarm. Fig. 15 illustrates remote bits associated with DNP3 controls being mapped to output contacts on an optional digital output board in the PAC_MASTER.



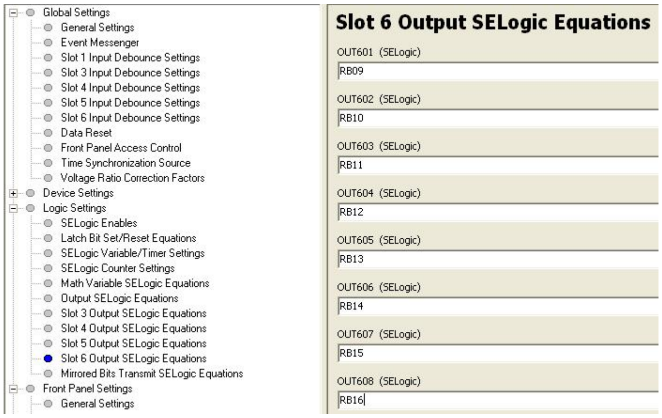
**Output SELogic Equations**

OUT101 (SELogic)
RB07

OUT102 (SELogic)
RB08

OUT103 (SELogic)
NOT ( HALARM OR SALARM )

Fig. 14. Remote Bits Associated With DNP3 Controls Mapped to the Output Contacts in the Base Unit of the PAC_MASTER



**Slot 6 Output SELogic Equations**

OUT601 (SELogic)
RB09

OUT602 (SELogic)
RB10

OUT603 (SELogic)
RB11

OUT604 (SELogic)
RB12

OUT605 (SELogic)
RB13

OUT606 (SELogic)
RB14

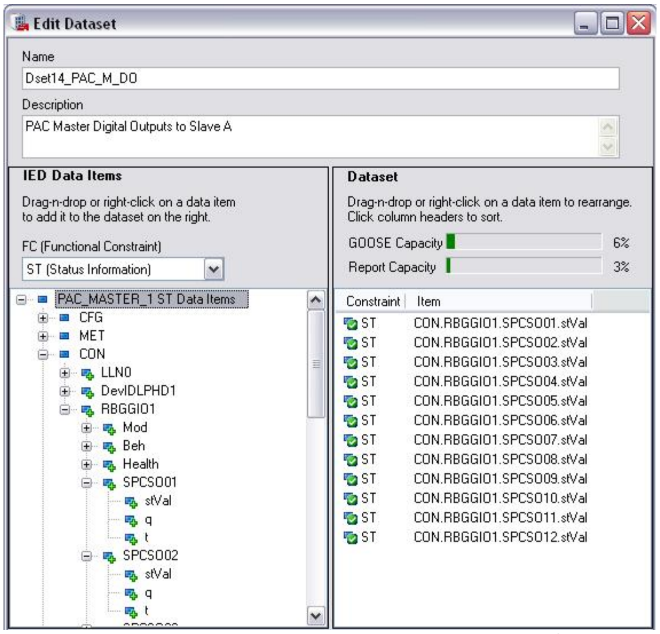
OUT607 (SELogic)
RB15

OUT608 (SELogic)
RB16

Fig. 15. Remote Bits Associated With DNP3 Controls Mapped to the Output Contacts on a Digital Output Board in the PAC_MASTER

### *F. Configure PAC_MASTER Pass-Through DNP3 Controls*

Fig. 16 illustrates six DNP3 control bit pairs being mapped to PAC_SLAVE_A as six pairs of remote bits. Once mapped, they are published in the GOOSE dataset Dset14_PAC_M_DO. These bits are found in the CON, or control, logical device within the RBGGIO1 logical node, or remote bit generic I/O. The points are mapped as single-point controls (SPCs) and the status value (stVal) represents the state of the DNP3 bit in the PAC_MASTER.



Name
Dset14_PAC_M_DO

Description
PAC Master Digital Outputs to Slave A

Drag-n-drop or right-click on a data item to add it to the dataset on the right.

FC (Functional Constraint)
ST (Status Information)

Drag-n-drop or right-click on a data item to rearrange. Click column headers to sort.

GOOSE Capacity 6%
Report Capacity 3%

| Constraint | Item |
|---|---|
| ST | CON.RBGGIO1.SPCSO01.stVal |
| ST | CON.RBGGIO1.SPCSO02.stVal |
| ST | CON.RBGGIO1.SPCSO03.stVal |
| ST | CON.RBGGIO1.SPCSO04.stVal |
| ST | CON.RBGGIO1.SPCSO05.stVal |
| ST | CON.RBGGIO1.SPCSO06.stVal |
| ST | CON.RBGGIO1.SPCSO07.stVal |
| ST | CON.RBGGIO1.SPCSO08.stVal |
| ST | CON.RBGGIO1.SPCSO09.stVal |
| ST | CON.RBGGIO1.SPCSO10.stVal |
| ST | CON.RBGGIO1.SPCSO11.stVal |
| ST | CON.RBGGIO1.SPCSO12.stVal |

Fig. 16. Six DNP3 Control Bit Pairs Mapped as Status Values of Twelve Individual Single-Point Controls

### *G. Configure PAC DNP Controls Subscription*

PAC_SLAVE_A is configured to map the incoming dataset contents to remote bits. Within PAC_SLAVE_A, these remote bits are mapped to digital outputs using the same process as in the PAC_MASTER. PAC remote bits are updated via internal logic algorithms, local digital inputs, MIRRORED BITS communications messages, DNP3 command messages, and/or GOOSE messages. In this RTU-replacement example, PAC_MASTER digital outputs are triggered by DNP3 commands, which write to remote bits. The PAC_SLAVE_A digital outputs are triggered by DNP3 commands. However, PAC_Slave_A does not directly receive the DNP3 commands, rather, the commands are accepted by the PAC_MASTER. The PAC_MASTER remote bits, which these commands trigger, are in turn mapped to a GOOSE dataset sent to PAC_SLAVE_A. These incoming GOOSE messages trigger local remote bits and then control digital outputs within PAC_SLAVE_A.

## VI. PAC SCADA I/O PERFORMANCE

This new PAC topology offers much more flexibility and many new features not available in traditional RTUs. Additionally, tests of response speed to DNP3 commands and poll requests demonstrate that the PAC topology performs equal to or better than single-purpose RTUs. A typical serial DNP3 SCADA master connection to the PAC_MASTER was staged with a 1000 millisecond poll rate. Testing was done to reveal performance of I/O directly in the PAC_MASTER as well as the I/O in the connected PAC_SLAVE_A, B, and C.

### *A. Configure PAC GOOSE Message Contents*

As within components of traditional RTUs, the I/O cards in a PAC are polled over the communications bus at a fixed rate using a proprietary message system. However, unlike RTU components that are combined using proprietary connectors, ribbon cables, and still more proprietary messaging, PACs are combined using internationally standard connectors, cables, and messaging. Once configured to match the specific RTU replacement in this case study, performance testing was done. Though capable of much more, the new connections, protocols, and architectures were verified to support SCADA functions in a timely manner. Table I illustrates elapsed times for various SCADA functions from a DNP3 client to the DNP3 server in the PAC_MASTER and PAC control outputs. Of course, the component of timing associated with the DNP3 client computer, and time of the message traversing the serial cable to and from the DNP3 client, is not affected by using a PAC. This time is unchanged whether for a traditional RTU or for PAC topology, but it is a part of the overall timing measurement.

The stages of the command and response sequence were identified as parts of the round trip message sequence scheme. The SER function within the PAC makes time elements easy to measure with great accuracy. This function timestamps and records the changes by name as they occur. The PACs were all synchronized to a GPS IRIG-B clock, and so the timestamps are recorded with millisecond resolution and are accurate to the millisecond as well.

Elements of a single PAC timing test include the following:

1. Time from selection of command at operator console, to subsequent DNP3 command message being sent from client and completely received by PAC_MASTER DNP3 server interface.
2. Time from complete receipt of message at PAC_MASTER DNP3 server interface, through subsequent processing of message, through acceptance of DNP3 command, to close action of digital output contact on PAC_MASTER.
3. **Control Time**—time from command initiation by SCADA operator at DNP3 client, to resulting close action of digital output contact on PAC_MASTER.
4. Time from close action of digital output contact on PAC_MASTER, to detected closure at connected digital input contact feedback on PAC_MASTER.
5. Time from detected closure at connected digital input contact on PAC_MASTER, to data change available for DNP3 poll.
6. **Total Feedback Time**—time from feedback data change available for DNP3 poll in PAC_MASTER, to detection of this feedback in DNP3 client.

TABLE I
SINGLE PAC SCADA FUNCTION TIMING RESULTS

| **Interaction With I/O on PAC_MASTER** | **Time Duration of SCADA Messaging Function** |
|---|---|
| Time Element 1 | Ave 0.522 s<br>Max 0.823 s<br>Min 0.298 s |
| Time Element 2 | Ave 0.007 s<br>Max 0.009 s<br>Min 0.004 s |
| Time Element 3<br>**Control Time** | Ave 0.529 s<br>Max 0.824 s<br>Min 0.305 s |
| Time Element 4 | Ave 0.0027 s<br>Max 0.0030 s<br>Min 0.0025 s |
| Time Element 5 | Ave 0.007 s<br>Max 0.009 s<br>Min 0.004 s |
| Time Element 6<br>**Total Feedback Time** | Ave 1.022 s<br>Max 1.317 s<br>Min 0.798 s |

Table II illustrates elapsed times for various SCADA functions between a DNP3 client and the DNP3 server in the PAC_MASTER, which in turn uses GOOSE messages to interact with PAC_SLAVE_A. With the extra step of communications between the PAC components that make up the RTU, latency to I/O in the distributed PACs is slightly longer than latency to I/O in the PAC_MASTER. The result is the same as with a large RTU with communications among individual card cages or other components. As with traditional RTUs, digital inputs that require SER accuracy are acquired directly by the primary component, in this case, the PAC_MASTER. Unlike RTUs, however, the PAC is capable of true one-millisecond absolute accuracy and resolution, providing much better data management.

Elements of a multiple PAC timing test include the following:

1. Time from selection of command at operator console, through subsequent DNP3 command message being sent from client, to complete receipt by PAC_MASTER DNP3 server interface.
2. Time from complete receipt of message at PAC_MASTER DNP3 server interface, through subsequent processing of message, through acceptance of DNP3 command, to preparation of dataset.
3. **Peer-to-Peer GOOSE Time**—time from dataset preparation, through subsequent publication from PAC_MASTER, through complete receipt, to processing of GOOSE message dataset at PAC_SLAVE_A.
4. Time from processing of GOOSE message dataset, to subsequent close action of digital output contact on PAC_SLAVE_A.
5. **Control Time**—time from command initiation by SCADA operator at DNP3 client, to resulting close action of digital output contact on PAC_SLAVE_A.
6. Time from close action of digital output contact on PAC_SLAVE_A, to detected closure at connected digital input contact feedback on PAC_SLAVE_A.
7. **PAC-to-PAC Feedback Time**—time from detected closure at connected digital input contact on PAC_SLAVE_A, through dataset preparation with feedback data change, through subsequent GOOSE publication in PAC_SLAVE_A, through complete receipt and processing of

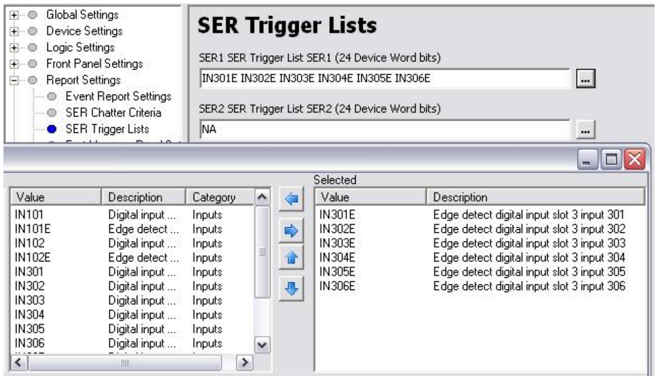
GOOSE message dataset, to feedback data change available for DNP3 poll at PAC_MASTER.

8. **Total Feedback Time**—time from detected closure at connected digital input contact on PAC_SLAVE_A, through dataset preparation, through subsequent GOOSE publication in PAC_SLAVE_A, through complete receipt and processing of GOOSE message dataset in PAC_MASTER, through DNP3 poll of PAC_MASTER, to detection of this feedback in DNP3 client.

TABLE II
MULTIPLE PAC SCADA FUNCTION TIMING RESULTS

| Interaction With I/O on PAC_MASTER | Time Duration of SCADA Messaging Function |
|---|---|
| Time Element 1 | Ave 0.587 s<br>Max 1.232 s<br>Min 0.214 s |
| Time Element 2 | Ave 0.007 s<br>Max 0.009 s<br>Min 0.004 s |
| Time Element 3<br>**Peer-to-Peer GOOSE Time** | Ave 0.003 s<br>Max 0.005 s<br>Min 0.001 s |
| Time Element 4 | Ave 0.041 s<br>Max 0.042 s<br>Min 0.041 s |
| Time Element 5<br>**Control Time** | Ave 0.601 s<br>Max 1.243 s<br>Min 0.226 s |
| Time Element 6 | Ave 0.002 s<br>Max 0.002 s<br>Min 0.002 s |
| Time Element 7<br>**PAC-to-PAC Feedback Time** | Ave 0.009 s<br>Max 0.011 s<br>Min 0.007 s |
| Time Element 8<br>**Total Feedback Time** | Ave 1.103 s<br>Max 1.744 s<br>Min 0.731 s |

As can be seen in Tables I and II, both single and multiple PAC architectures perform SCADA functions faster than most traditional RTUs. Though traditional RTUs provide millisecond timestamp resolution and claim relative timestamp accuracies of several milliseconds, this only means that data changes detected within the RTU don't differ from one another by more than this value. However, RTUs do not accurately create an absolute timestamp related to the actual real-time occurrence of the event. This lack of absolute accuracy reduces the value of RTU data as part of an IED network, where order of events and time duration between them is important. This PAC is the perfect network IED for this purpose due to its ability to create timestamps accurate to the millisecond. Fig. 17 illustrates configuration of inputs within this PAC as SER points that will be stored locally for a forensic report. This PAC immediately reports the SER points when they occur, for use in exception-based data acquisition.



SER1 SER Trigger List SER1 (24 Device Word bits): IN301E IN302E IN303E IN304E IN305E IN306E

SER2 SER Trigger List SER2 (24 Device Word bits): NA

| Value | Description | Category |
|---|---|---|
| IN101 | Digital input ... | Inputs |
| IN101E | Edge detect ... | Inputs |
| IN102 | Digital input ... | Inputs |
| IN102E | Edge detect ... | Inputs |
| IN301 | Digital input ... | Inputs |
| IN302 | Digital input ... | Inputs |
| IN303 | Digital input ... | Inputs |
| IN304 | Digital input ... | Inputs |
| IN305 | Digital input ... | Inputs |
| IN306 | Digital input ... | Inputs |

| Value | Description |
|---|---|
| IN301E | Edge detect digital input slot 3 input 301 |
| IN302E | Edge detect digital input slot 3 input 302 |
| IN303E | Edge detect digital input slot 3 input 303 |
| IN304E | Edge detect digital input slot 3 input 304 |
| IN305E | Edge detect digital input slot 3 input 305 |
| IN306E | Edge detect digital input slot 3 input 306 |

Fig. 17. Selection of Digital Input Points for Inclusion in PAC SER Reports and Unsolicited Updates via a Pick List

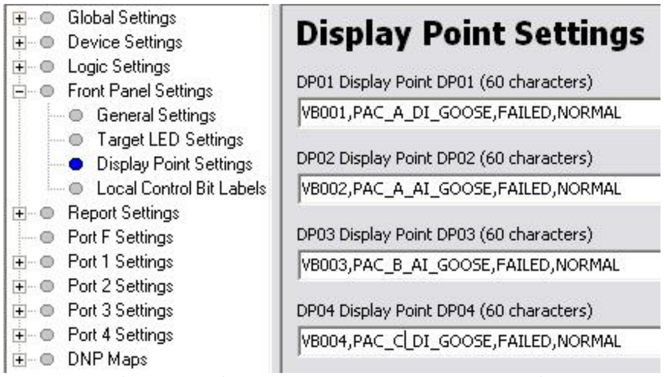
## VII. PAC DISPLAY, ALARM, AND NOTIFICATION

Another differentiator between a PAC and a traditional RTU is the PAC's local HMI communications capabilities. Local and remote data are quickly made visible on the front-panel HMI and also communicated via non-SCADA methods directly to utility personnel.

### A. *Configure Display Points*

This PAC software automatically generates much of the information displayed on the front-panel HMI. The HMI displays much of the acquired and calculated values as LEDs and as text on the LCD display. In this example, we also customized the PAC_MASTER to display the status of each of the incoming GOOSE messages. As shown in Fig. 8, the quality of each incoming GOOSE message is mapped to a virtual bit, VB001–VB004. Another benefit of the PAC over the RTU is that these quality bits are available within algorithms using data from the different interconnected components. When data become unavailable for any reason, such as a bad cable or CPU failure, the software immediately notifies each PAC and modifies its processing accordingly. Traditional RTUs are not capable of detecting an internal cable or part failure and often interpret a lack of response as a zero value or no-data change.

In this example, virtual bits are assigned to customizable PAC display points and made visible on the front panel. Fig. 18 illustrates configuring display points to show the status of the incoming GOOSE messages on the local display. The setting field includes the internal point name, label used for local display on the HMI, status text for set state, and status text for cleared state, each separated by a comma.



**Display Point Settings**

DP01 Display Point DP01 (60 characters)
VB001,PAC_A_DI_GOOSE,FAILED,NORMAL

DP02 Display Point DP02 (60 characters)
VB002,PAC_A_AI_GOOSE,FAILED,NORMAL

DP03 Display Point DP03 (60 characters)
VB003,PAC_B_AI_GOOSE,FAILED,NORMAL

DP04 Display Point DP04 (60 characters)
VB004,PAC_C_DI_GOOSE,FAILED,NORMAL

Fig. 18. Configuration of Display Points to Show the Status of the Incoming GOOSE Messages

Fig. 19 illustrates the view of the PAC_MASTER HMI LCD display with the Ethernet cable to PAC_SLAVE_C disconnected and the resulting failure status.

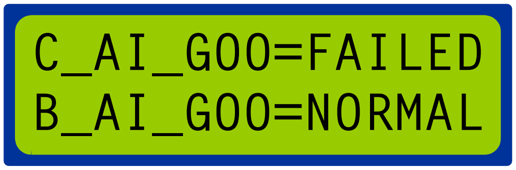
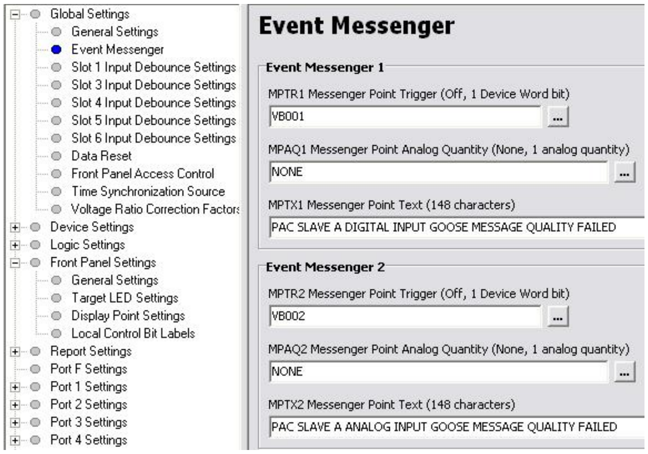

C_AI_GOO=FAILED
B_AI_GOO=NORMAL

Fig. 19. PAC HMI View of GOOSE Message Quality Display Point

## *B. Configure Event Messenger*

The PAC also has the capability to use virtually any available communications media to send a text-based message containing an alert or alarm. Fig. 20 illustrates the configuration of a text message destined for a communications port rather than the local display. This unsolicited message is sent from a communications port when the associated virtual bit is set, upon failure of the incoming GOOSE message.



**Event Messenger**

**Event Messenger 1**

MPTR1 Messenger Point Trigger (Off, 1 Device Word bit)
VB001

MPAQ1 Messenger Point Analog Quantity (None, 1 analog quantity)
NONE

MPTX1 Messenger Point Text (148 characters)
PAC SLAVE A DIGITAL INPUT GOOSE MESSAGE QUALITY FAILED

**Event Messenger 2**

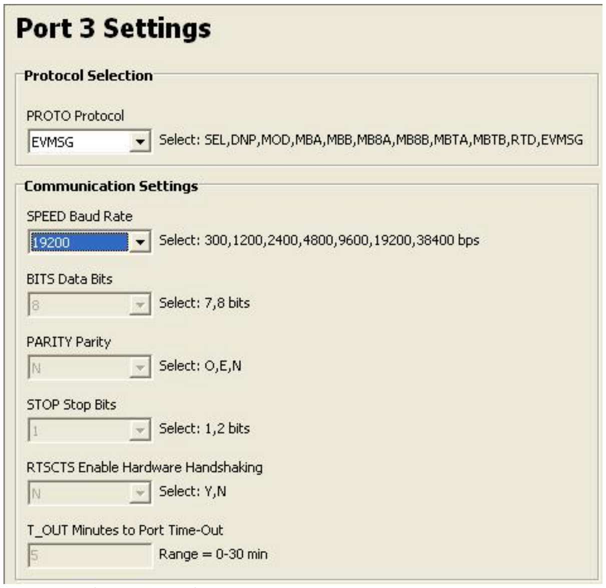
MPTR2 Messenger Point Trigger (Off, 1 Device Word bit)
VB002

MPAQ2 Messenger Point Analog Quantity (None, 1 analog quantity)
NONE

MPTX2 Messenger Point Text (148 characters)
PAC SLAVE A ANALOG INPUT GOOSE MESSAGE QUALITY FAILED

Fig. 20. Configuration of Event Messenger Points to Report the Status of the Incoming GOOSE Messages

Fig. 21 illustrates the serial port settings used to send these unsolicited messages to report alerts and alarms. These messages travel over virtually any media capable of supporting text communications, such as a text message over a cellular telephone modem. Additionally, they can be sent to a device that converts them to audio and then calls and reports the message audibly.


**Port 3 Settings**

**Protocol Selection**

PROTO Protocol
EVMSG
Select: SEL,DNP,MOD,MBA,MBB,MB8A,MB8B,MBTA,MBTB,RTD,EVMSG

**Communication Settings**

SPEED Baud Rate
19200
Select: 300,1200,2400,4800,9600,19200,38400 bps

BITS Data Bits
8
Select: 7,8 bits

PARITY Parity
N
Select: O,E,N

STOP Stop Bits
1
Select: 1,2 bits

RTSCTS Enable Hardware Handshaking
N
Select: Y,N

T_OUT Minutes to Port Time-Out
5
Range = 0-30 min

Fig. 21. Configuration of Port 3 on the PAC_MASTER to Report Event Messenger Points

## *C. Configure Email Notification*

In addition to the previously mentioned methods of sending the event message as a text message and an audible voice message, it can also be converted into an email message. In this case, a serial-to-Ethernet transceiver is connected to Port 3, which captures the text message and then converts it to an email. This email is then immediately sent to an individual or a mail group address and is available at the PC, Blackberry™, or other PDA.

# VIII. USE FEATURES UNAVAILABLE IN RTUS TO IMPROVE COMMISSIONING

As mentioned previously, PACs automatically monitor the communications among the RTU system components to determine message quality. Each device detects errors in received messages and failure to receive expected messages from other devices and performs remediation immediately.

## *A. Calculate Intracomponent Communications Status*

Each PAC verifies the message quality by combining the status of each of the codes shown in Table III. If the PAC detects any of these to be true, it sets the message quality to failure.

TABLE III
GOOSE MESSAGE ERROR CODES

| Message Statistics | Error Code |
|---|---|
| Configuration revision mismatch between publisher and subscriber | CONF REV MISMA |
| Publisher indicates that it needs commissioning | NEED COMMISSIO |
| Publisher is in test center | TEST MODE |
| Received message is decoding and reveals error | MSG CORRUPTED |
| Message received out of sequence | OUT OF SEQUENC |
| Message time-to-live expired | TTL EXPIRED |

### B. *Uniquely Identify Each Configuration Revision in PAC*

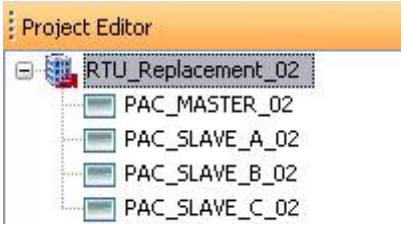
The name designated for the PAC during the IEC 61850 configuration is also used as the “iedName” and is the prefix of the various GOOSE configuration labels. To accurately install, commission, and troubleshoot, the utility must be certain of how each PAC is configured and have the ability to view and manage the configuration files using PC commissioning and configuration software tools. The IEC 61850 Communications Standard describes the substation configuration language (SCL) and configuration files, which configure devices for IEC 61850 communications.

The preferred method is to actually load a configuration file into the PAC so the PAC can use the information to communicate properly. Loading the file directly into the PAC has several advantages over the legacy method of sending settings to the PAC. A very important advantage is the ability to identify what communications behavior the PAC is configured for, and then cross-reference that to other PACs and the configuration software. Engineering access commands in the PAC reveal which file is being used within the PAC.

Without this capability, other manufacturers rely instead on the legacy method of sending settings, and it is impossible to later verify the PAC configuration from the device itself. If configuration is performed using settings instead of files, after initial configuration, it is possible to know the PAC configuration only by retrieving the settings from the PAC. Communications and network designers will have no method of confirming configuration as they monitor PAC behavior by capturing the actual communications messages sent from the PAC onto the network. This complicates communications commissioning and diagnostics because these engineers and technicians will need to learn and use IED settings software to verify network parameters, while being careful not to disrupt other settings. This makes settings revision management difficult among the various groups and increases the necessary licenses for configuration software packages. These packages are often expensive and must be kept up to date in order to communicate with the IEDs.

Further, by separating IEC 61850 configuration from other PAC and protection settings via the SCL configuration file download, it is possible to be certain that no other settings were accidently modified or affected. This provides security by minimizing the impact to the system, minimizing the recommissioning after a change, and eliminating risk of unintentionally affecting the other processes within the system.

Using a PAC with the flexibility to load files, modify the iedName, and support the GOOSE report command greatly improves the system. Essentially, this method makes it possible for the utility to be certain about how the PAC is configured and that the correct file is loaded by the configuration software for visible review and modification. Using this method, the utility can also be certain that the correct files are archived. Using files enables remote experts to receive the configuration files for review and support. Alternatively, remote experts can perform the configuration or modifications and send only the file to the substation via email or other means. Changing the iedName each time a new configuration file is created allows the operator to determine which revision is being used. Fig. 22 illustrates the Project Editor window of the PAC configuration software where the various PACs are added to the project and then given a unique name. This screen capture reflects project Revision 2 where “02” is appended to each IED name. The appended revision number for the project and IEDs is incremented via the rename feature each time the configuration is changed. Project revisions and IED revisions are then loaded into the PACs and archived on the PC. Using this feature, it is always possible to learn from the PAC what configuration file is active and then to view that file with the PC configuration tool.

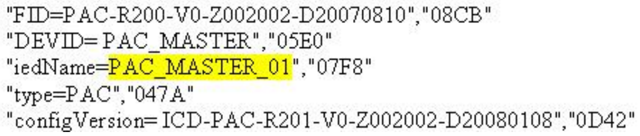


RTU_Replacement_02
- PAC_MASTER_02
- PAC_SLAVE_A_02
- PAC_SLAVE_B_02
- PAC_SLAVE_C_02

Fig. 22. Project Editor Window of IEC 61850 Configuration Software for Revision 2

### C. *Collect and Review Thorough Diagnostic Report*

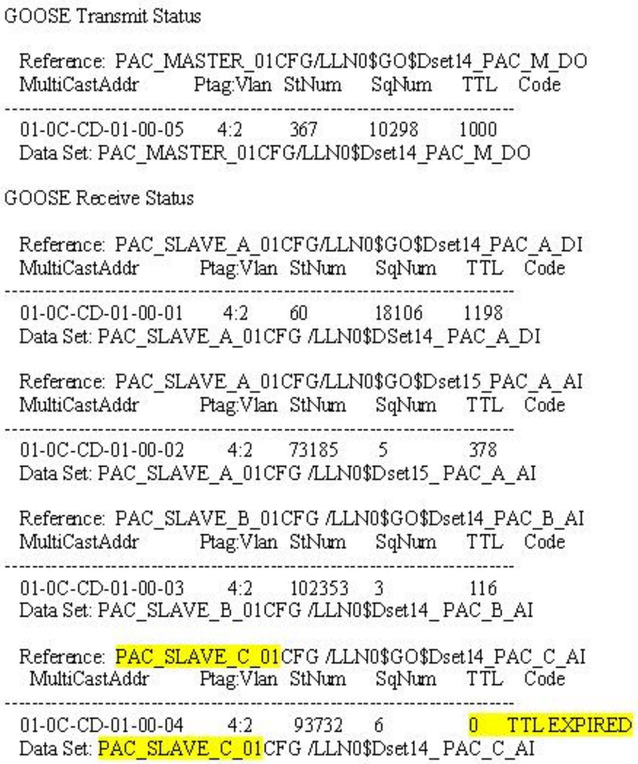
Using an engineering access port, PAC reports and archived data are retrieved to a PC for reuse and/or visualization. Fig. 23 displays the PAC report that results from sending the identification request command, **ID**, via the engineering access connection. This confirms that the PC is communicating with PAC_MASTER, Revision 1.



```
"FID=PAC-R200-V0-Z002002-D20070810","08CB"
"DEVID= PAC_MASTER","05E0"
"iedName=PAC_MASTER_01","07F8"
"type=PAC","047A"
"configVersion= ICD-PAC-R201-V0-Z002002-D20080108","0D42"
```

Fig. 23. ID Command Report From PAC_MASTER

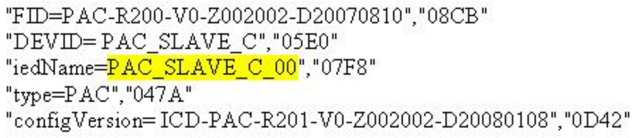
The GOOSE report includes the configuration and status of incoming and outgoing GOOSE messages. Each report includes message configuration and performance information. Configuration information for each GOOSE message includes a message label, multicast address, priority tag, virtual LAN identifier, and dataset name. Real-time statistics for each message include the status number, sequence number, time-to-live value, and error code. If detected, an error code from Table III is displayed. The report shown in Fig. 24 illustrates that the PAC_MASTER is unsuccessful in communicating to PAC_SLAVE_C. Highlighted in the report are the IED name and error code. For PAC_SLAVE_C_01, the time-to-live (TTL) value has decremented to zero and the associated error code, **TTL EXPIRED**, is displayed. Recall that the suffix _01 represents that the PAC is presently configured with SCL file Revision 1 that was created as part of Revision 1 of the project entitled, RTU Replacement. PAC_SLAVE_A analog input and digital input GOOSE messages as well as PAC_SLAVE_B analog input GOOSE messages are communicating without errors.



```
GOOSE Transmit Status

  Reference:  PAC_MASTER_01CFG/LLN0$GO$Dset14_PAC_M_DO
  MultiCastAddr        Ptag:Vlan  StNum     SqNum     TTL   Code
--------------------------------------------------------------------------
  01-0C-CD-01-00-05     4:2      367       10298      1000
  Data Set: PAC_MASTER_01CFG/LLN0$Dset14_PAC_M_DO

GOOSE Receive Status

  Reference:  PAC_SLAVE_A_01CFG/LLN0$GO$Dset14_PAC_A_DI
  MultiCastAddr         Ptag:Vlan  StNum     SqNum     TTL    Code
--------------------------------------------------------------------------
  01-0C-CD-01-00-01      4:2      60         18106      1198
  Data Set: PAC_SLAVE_A_01CFG /LLN0$DSet14_ PAC_A_DI

  Reference:  PAC_SLAVE_A_01CFG/LLN0$GO$Dset15_PAC_A_AI
  MultiCastAddr         Ptag:Vlan  StNum     SqNum     TTL    Code
--------------------------------------------------------------------------
  01-0C-CD-01-00-02      4:2      73185      5          378
  Data Set: PAC_SLAVE_A_01CFG /LLN0$Dset15_ PAC_A_AI

  Reference:  PAC_SLAVE_B_01CFG /LLN0$GO$Dset14_PAC_B_AI
  MultiCastAddr         Ptag:Vlan  StNum     SqNum     TTL    Code
--------------------------------------------------------------------------
  01-0C-CD-01-00-03      4:2      102353     3          116
  Data Set: PAC_SLAVE_B_01CFG /LLN0$Dset14_ PAC_B_AI

  Reference:  PAC_SLAVE_C_01CFG /LLN0$GO$Dset14_PAC_C_AI
   MultiCastAddr        Ptag:Vlan  StNum     SqNum     TTL    Code
--------------------------------------------------------------------------
  01-0C-CD-01-00-04      4:2      93732      6          0     TTL EXPIRED
  Data Set: PAC_SLAVE_C_01CFG /LLN0$Dset14_ PAC_C_AI
```

Fig. 24. GOOSE Report From PAC_MASTER Showing Expected GOOSE Configuration and Present Failure Code

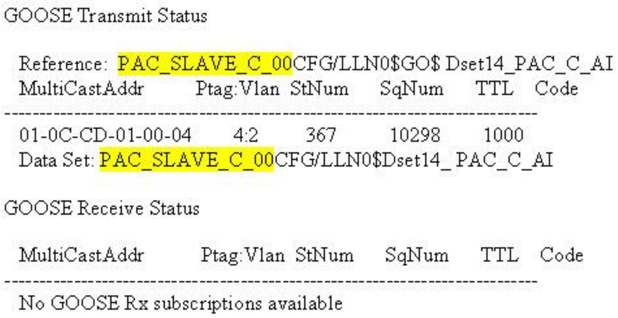
Further diagnostics are possible by retrieving the identification report from PAC_SLAVE_C as shown in Fig. 25. This report reveals that the PAC_SLAVE_C_00 is presently configured with Revision 0, and thus, the communications with PAC_MASTER failed.



```
"FID=PAC-R200-V0-Z002002-D20070810","08CB"
"DEVID= PAC_SLAVE_C","05E0"
"iedName=PAC_SLAVE_C_00","07F8"
"type=PAC","047A"
"configVersion= ICD-PAC-R201-V0-Z002002-D20080108","0D42"
```

Fig. 25. Identification Report From PAC_SLAVE_C Showing That the IED Is Presently Configured With Revision 0

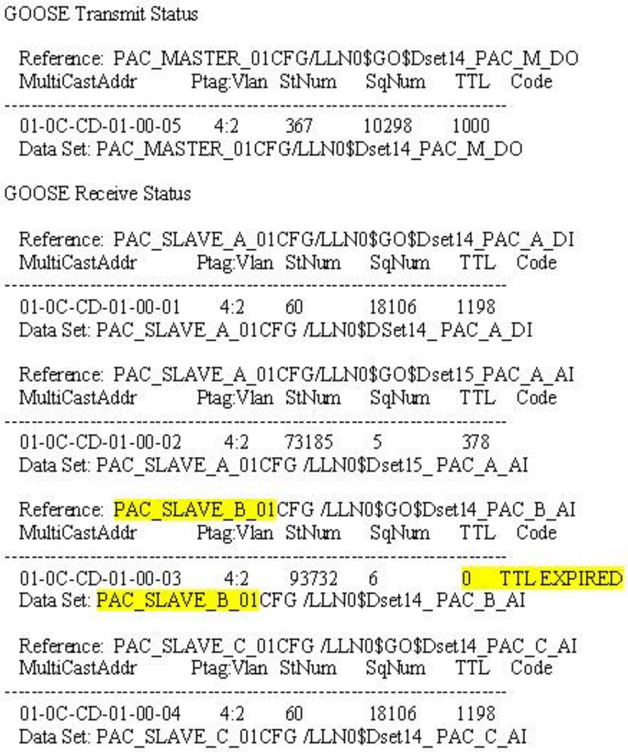
Further analysis is done by retrieving the GOOSE report from PAC_SLAVE_C, as shown in Fig. 26. This confirms that it is presently publishing GOOSE with the prefix PAC_SLAVE_C_00. Because this does not match the configuration within the PAC_MASTER, which is expecting Revision 1, it is not accepted. The message is denied because the message purpose and contents may have changed. This type of filtering is essential because the nature of Ethernet precludes disabling receipt of some but not all messages. If the cable is unplugged, as may happen with direct serial peer-to-peer messaging RTU component connections, all communication is lost.



```
GOOSE Transmit Status

  Reference:  PAC_SLAVE_C_00CFG/LLN0$GO$ Dset14_PAC_C_AI
  MultiCastAddr        Ptag:Vlan  StNum      SqNum     TTL    Code
-----------------------------------------------------------------------------
  01-0C-CD-01-00-04      4:2       367        10298      1000
  Data Set: PAC_SLAVE_C_00CFG/LLN0$Dset14_ PAC_C_AI

GOOSE Receive Status

  MultiCastAddr         Ptag:Vlan  StNum     SqNum      TTL    Code
-----------------------------------------------------------------------------
  No GOOSE Rx subscriptions available
```

Fig. 26. GOOSE Report From PAC_SLAVE_C Showing GOOSE Configuration Is Presently Revision 0

If, however, review of the GOOSE and identification reports confirms that the configuration in each PAC is correct and there is still a failure condition, the connections must be confirmed. Fig. 27 displays the result of disconnecting the cable to PAC_SLAVE_B where the time-to-live (TTL) value has decremented to zero and the associated error code, **TTL EXPIRED**, is displayed.

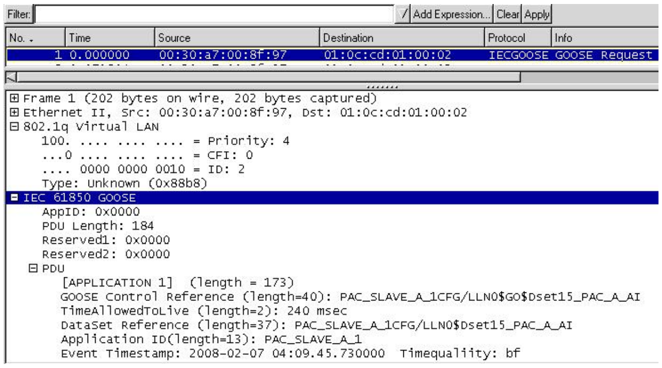


```
GOOSE Transmit Status

  Reference:  PAC_MASTER_01CFG/LLN0$GO$Dset14_PAC_M_DO
  MultiCastAddr        Ptag:Vlan  StNum     SqNum     TTL   Code
--------------------------------------------------------------------------
  01-0C-CD-01-00-05     4:2       367        10298      1000
  Data Set: PAC_MASTER_01CFG/LLN0$Dset14_PAC_M_DO

GOOSE Receive Status

  Reference:  PAC_SLAVE_A_01CFG/LLN0$GO$Dset14_PAC_A_DI
  MultiCastAddr         Ptag:Vlan  StNum     SqNum     TTL    Code
--------------------------------------------------------------------------
  01-0C-CD-01-00-01      4:2      60         18106      1198
  Data Set: PAC_SLAVE_A_01CFG /LLN0$DSet14_ PAC_A_DI

  Reference:  PAC_SLAVE_A_01CFG/LLN0$GO$Dset15_PAC_A_AI
  MultiCastAddr         Ptag:Vlan  StNum     SqNum     TTL    Code
--------------------------------------------------------------------------
  01-0C-CD-01-00-02      4:2      73185      5          378
  Data Set: PAC_SLAVE_A_01CFG /LLN0$Dset15_ PAC_A_AI

  Reference:  PAC_SLAVE_B_01CFG /LLN0$GO$Dset14_PAC_B_AI
  MultiCastAddr         Ptag:Vlan  StNum     SqNum     TTL    Code
--------------------------------------------------------------------------
  01-0C-CD-01-00-03      4:2      93732      6          0     TTL EXPIRED
  Data Set: PAC_SLAVE_B_01CFG /LLN0$Dset14_ PAC_B_AI

  Reference:  PAC_SLAVE_C_01CFG /LLN0$GO$Dset14_PAC_C_AI
  MultiCastAddr        Ptag:Vlan  StNum     SqNum     TTL    Code
--------------------------------------------------------------------------
  01-0C-CD-01-00-04      4:2      60         18106      1198
  Data Set: PAC_SLAVE_C_01CFG /LLN0$Dset14_ PAC_C_AI
```

Fig. 27. Goose Status Report Revealing Correct Configuration but Failed Communications

### *D. Use Common Ethernet Tools to Aid Troubleshooting*

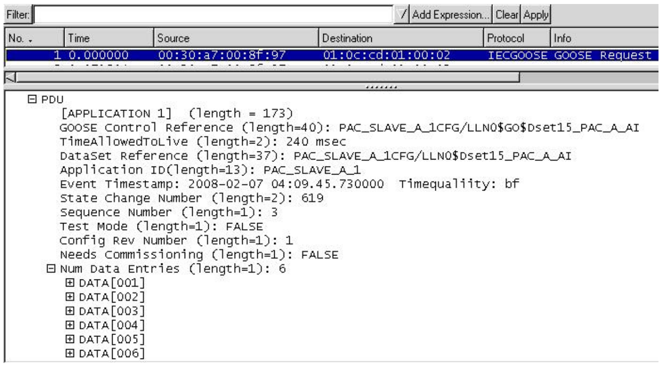
Since the IEC 61850 GOOSE messages travel over Ethernet, Ethernet recording software that collects many Ether types, not just TCP/IP traffic, will capture and store message traffic. Fig. 28 illustrates a display within Ethereal® software that deconstructs one of the GOOSE messages. Review of the file shows that the captured message has the destination (Dst) multicast address 01:0c:cd:01:00:02 and is from PAC_ SLAVE_A, Revision 1. Fig. 29 illustrates the protocol data unit (PDU) of the same message displaying the GOOSE parameters and that the payload, or dataset in this case, contains six entries. Fig. 30 illustrates even more detail, and even though the contents of GOOSE messages are anonymous, the data types and values are displayed. The dataset description in Fig. 5 documents the name of each floating-point value. The last analog in the dataset is frequency, and its value is 60 in this GOOSE message. The PDU in Fig. 31 illustrates the status of the digital inputs in SLAVE_A, being published as Booleans with the present value of zero. Fig. 4 shows the digital inputs being configured.



```
No.    Time        Source              Destination         Protocol  Info
     1 0.000000    00:30:a7:00:8f:97   01:0c:cd:01:00:02   IECGOOSE  GOOSE Request
```

```
⊞ Frame 1 (202 bytes on wire, 202 bytes captured)
⊞ Ethernet II, Src: 00:30:a7:00:8f:97, Dst: 01:0c:cd:01:00:02
⊟ 802.1q Virtual LAN
    100. .... .... .... = Priority: 4
    ...0 .... .... .... = CFI: 0
    .... 0000 0000 0010 = ID: 2
    Type: Unknown (0x88b8)
⊟ IEC 61850 GOOSE
    AppID: 0x0000
    PDU Length: 184
    Reserved1: 0x0000
    Reserved2: 0x0000
  ⊟ PDU
      [APPLICATION 1]  (length = 173)
      GOOSE Control Reference (length=40): PAC_SLAVE_A_1CFG/LLN0$GO$Dset15_PAC_A_AI
      TimeAllowedToLive (length=2): 240 msec
      DataSet Reference (length=37): PAC_SLAVE_A_1CFG/LLN0$Dset15_PAC_A_AI
      Application ID(length=13): PAC_SLAVE_A_1
      Event Timestamp: 2008-02-07 04:09.45.730000  Timequality: bf
```

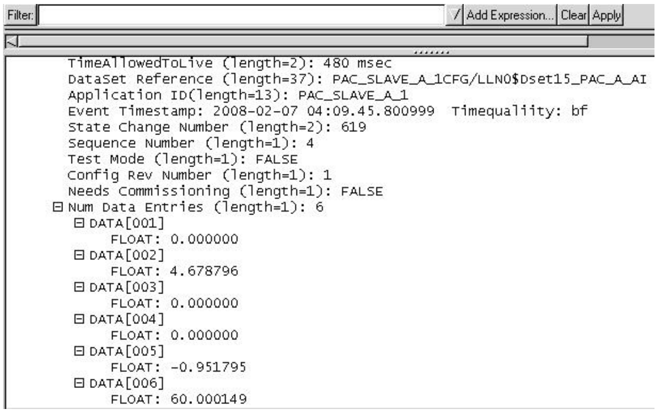
Fig. 28. Ethereal Software Deconstruction of Header of GOOSE Message From PAC_SLAVE_A Revision 1



```
No.    Time        Source              Destination         Protocol  Info
     1 0.000000    00:30:a7:00:8f:97   01:0c:cd:01:00:02   IECGOOSE  GOOSE Request
```

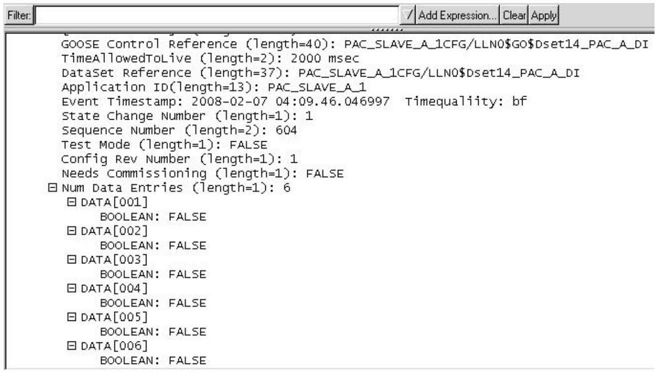
```
  ⊟ PDU
      [APPLICATION 1]  (length = 173)
      GOOSE Control Reference (length=40): PAC_SLAVE_A_1CFG/LLN0$GO$Dset15_PAC_A_AI
      TimeAllowedToLive (length=2): 240 msec
      DataSet Reference (length=37): PAC_SLAVE_A_1CFG/LLN0$Dset15_PAC_A_AI
      Application ID(length=13): PAC_SLAVE_A_1
      Event Timestamp: 2008-02-07 04:09.45.730000  Timequality: bf
      State Change Number (length=2): 619
      Sequence Number (length=1): 3
      Test Mode (length=1): FALSE
      Config Rev Number (length=1): 1
      Needs Commissioning (length=1): FALSE
    ⊟ Num Data Entries (length=1): 6
      ⊞ DATA[001]
      ⊞ DATA[002]
      ⊞ DATA[003]
      ⊞ DATA[004]
      ⊞ DATA[005]
      ⊞ DATA[006]
```

Fig. 29. Ethereal Software Deconstruction of PDU Header of GOOSE Message From PAC_SLAVE_A Revision 1



```
      TimeAllowedToLive (length=2): 480 msec
      DataSet Reference (length=37): PAC_SLAVE_A_1CFG/LLN0$Dset15_PAC_A_AI
      Application ID(length=13): PAC_SLAVE_A_1
      Event Timestamp: 2008-02-07 04:09.45.800999  Timequality: bf
      State Change Number (length=2): 619
      Sequence Number (length=1): 4
      Test Mode (length=1): FALSE
      Config Rev Number (length=1): 1
      Needs Commissioning (length=1): FALSE
    ⊟ Num Data Entries (length=1): 6
      ⊟ DATA[001]
          FLOAT: 0.000000
      ⊟ DATA[002]
          FLOAT: 4.678796
      ⊟ DATA[003]
          FLOAT: 0.000000
      ⊟ DATA[004]
          FLOAT: 0.000000
      ⊟ DATA[005]
          FLOAT: -0.951795
      ⊟ DATA[006]
          FLOAT: 60.000149
```

Fig. 30. Ethereal Software Deconstruction of PDU Analog Dataset of GOOSE Message From PAC_SLAVE_A Revision 1



```
      GOOSE Control Reference (length=40): PAC_SLAVE_A_1CFG/LLN0$GO$Dset14_PAC_A_DI
      TimeAllowedToLive (length=2): 2000 msec
      DataSet Reference (length=37): PAC_SLAVE_A_1CFG/LLN0$Dset14_PAC_A_DI
      Application ID(length=13): PAC_SLAVE_A_1
      Event Timestamp: 2008-02-07 04:09.46.046997  Timequality: bf
      State Change Number (length=1): 1
      Sequence Number (length=2): 604
      Test Mode (length=1): FALSE
      Config Rev Number (length=1): 1
      Needs Commissioning (length=1): FALSE
    ⊟ Num Data Entries (length=1): 6
      ⊟ DATA[001]
          BOOLEAN: FALSE
      ⊟ DATA[002]
          BOOLEAN: FALSE
      ⊟ DATA[003]
          BOOLEAN: FALSE
      ⊟ DATA[004]
          BOOLEAN: FALSE
      ⊟ DATA[005]
          BOOLEAN: FALSE
      ⊟ DATA[006]
          BOOLEAN: FALSE
```

Fig. 31. Ethereal Software Deconstruction of PDU Digital Dataset of GOOSE Message From PAC_SLAVE_A Revision 1

### *E. Calculate GOOSE Message Reliability and Channel Availability*

Once recorded as a timestamped SER, each GOOSE message quality status is collected as a system-wide diagnostic. After commissioning, message quality fails only when a message is corrupted or not received. The observation of failures indicates the reliability of individual GOOSE messages. If the message quality failure is intermittent, the duration of the failures is calculated as the difference between timestamps. The aggregate of failure duration over a given amount of time determines the channel availability.

### *F. Quickly Identify Source of Scaling or Wiring Problems*

Using the PAC front-panel HMI dramatically reduces troubleshooting, scaling, and database problems. Due to the fact that traditional RTUs have no integrated front-panel HMI, scaling and database problems are only visible at the SCADA console. However, the problem could be in the SCADA database, SCADA communications link, RTU database, or RTU termination wiring. Troubleshooting these legacy systems is very time consuming because a PC must be connected to a maintenance port on the RTU while a separate communications link collects the value in question. If the problem is visible, it is then assumed to be in the RTU communications to the PC, the RTU database, or the RTU terminations, but not the SCADA communications or SCADA database. However, it is never possible to be certain where the problem is, and it can be found only by trial and error modifications. The local LCD display on the PAC changes all of this. It provides a direct view of the database value, and troubleshooting can begin at once. Further, it is possible to be certain of the measured or calculated value within the PAC.

## IX. ZERO SETTINGS HMI AVAILABLE ANYWHERE, ANYTIME

Some protocols within IEC 61850 are fully routable, meaning that its message exchange provides both network and device addresses. The message contents are self-described and include values, format, attributes, and descriptions. With all this information in each message, users do not need previous knowledge of the source. This message transparency and routability require sophisticated security if the messages travel outside a security perimeter [1].

In contrast, a suite of nonroutable protocols, provided by the manufacturer of this PAC, was developed by experts with knowledge of the available data sources, destinations, and communications media to satisfy all substation data flow requirements. Some of these native PAC protocols include values, format, attributes, and descriptions to provide context and understanding of what, why, where, and when something happened on the power system, but they are not routable.

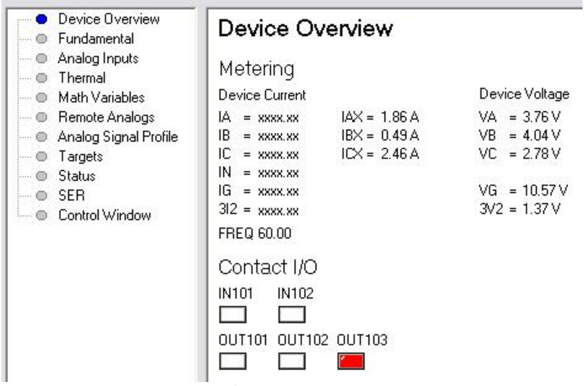
DNP3, IEC 61850, and one of the native PAC protocols offer the flexibility to customize each datamap and dataset to reduce overhead and improve performance. The other native PAC protocols serve fixed, predefined datasets containing present or historical values in response to frequent polls, constantly unsolicited, and in response to unexpected ad-hoc requests. Once a connection is established, the messages travel constantly and act as a "heartbeat" so that all receivers know that the senders are active and functioning properly.

The fixed datasets within the native PAC protocols provide the ability to be certain what data will be provided from any IED regardless of when it was manufactured and its location anywhere in the world. Thus, PC applications know what data to expect and then perform visualization and logic automatically, without the need for any settings.

The innovative message transfer methods of the native PAC protocols provide fast performance without the huge bandwidth and processing overhead of routable protocols. These native PAC protocols and DNP3 can also be safely routed via TCP/IP, but the messages retain the anonymity and obscurity of nonroutable messages.

These unique traits of the native PAC protocols make them secure, deterministic, and safely routable. These features, combined with the fact that every IED provides known, fixed datasets, enables communications without any settings. PC applications, including the PAC configuration software, use these protocols to connect to any type of IED, built at any time, with any version of firmware, and installed anywhere. These applications automatically collect messages and provide visualization, logic, and diagnostic tools. This is all done in parallel to the other protocol communications. It cannot be supported by any other known protocol because it is not possible to anticipate dataset contents.

The PAC configuration software displays the data collected from the PAC and any other IED supporting these protocols. It provides helpful descriptions, comments, graphics, and control points. Fig. 32 illustrates the list of preconfigured HMI displays and the default Device Overview.



Device Overview

Metering

Device Current

IA = xxxx.xx IAX = 1.86 A
IB = xxxx.xx IBX = 0.49 A
IC = xxxx.xx ICX = 2.46 A
IN = xxxx.xx
IG = xxxx.xx
3I2 = xxxx.xx
FREQ 60.00

Device Voltage

VA = 3.76 V
VB = 4.04 V
VC = 2.78 V
VG = 10.57 V
3V2 = 1.37 V

Contact I/O

IN101 IN102

OUT101 OUT102 OUT103

Fig. 32. Zero Settings HMI Software Pick List and Device Overview Display

Fig. 33 illustrates the Front-Panel display that matches the layout of the device LEDs and direct-action pushbuttons. The top of the screen displays the status of user-defined targets, or IED content indicators. Again, without any settings, the HMI operator picks and chooses what PAC contents to monitor and what color they want the indicator to display in the false state. The point name is automatically shown. This display was configured to show the virtual bits, contact inputs and outputs, and remote bits that are being used for the RTU Replacement project. The device Front-Panel display shows the indicators that were set in the PAC. The HMI operator customizes the display with user-defined labels. These labels are chosen to help each individual HMI operator and can be changed for each use of the HMI to satisfy the needs and understanding of multiple, separate HMI operators.

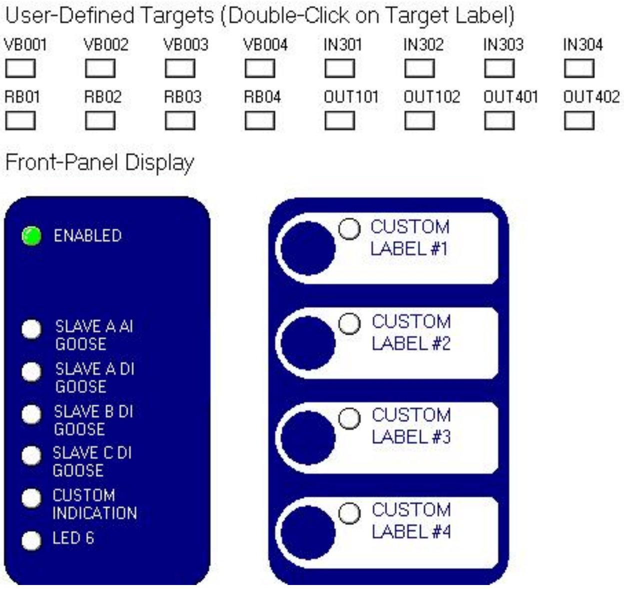
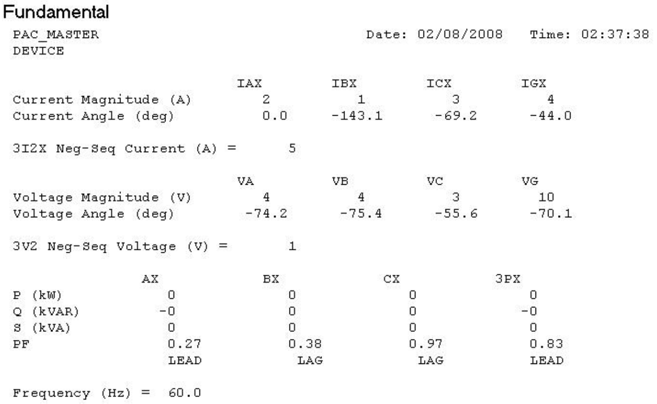


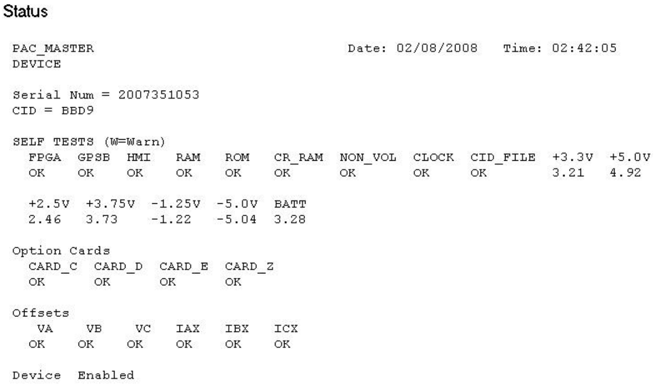
Fig. 33. Device Front-Panel Display and User-Defined Monitoring

Fig. 34 illustrates the display of a report retrieved from the device. In this example, the PAC was queried for a Fundamental Metering report. The PAC software was programmed with the appropriate retrieval command to collect, parse, and present the report in an HMI display. Thus, a text response is automatically converted into a simple, well organized, and IED-specific display. Fig. 35 illustrates a similar report display. In this case, a status command retrieves a text string that is organized into an HMI display showing configuration, self test, and diagnostic details.



```
Fundamental
PAC_MASTER                                  Date: 02/08/2008   Time: 02:37:38
DEVICE

                          IAX        IBX        ICX        IGX
Current Magnitude (A)        2          1          3          4
Current Angle (deg)        0.0     -143.1      -69.2      -44.0

3I2X Neg-Seq Current (A) =      5

                          VA         VB         VC         VG
Voltage Magnitude (V)        4          4          3         10
Voltage Angle (deg)      -74.2      -75.4      -55.6      -70.1

3V2 Neg-Seq Voltage (V) =       1

                 AX             BX             CX             3PX
P (kW)            0              0              0              0
Q (kVAR)         -0              0              0             -0
S (kVA)           0              0              0              0
PF                0.27           0.38           0.97           0.83
                  LEAD            LAG            LAG           LEAD

Frequency (Hz) =  60.0
```

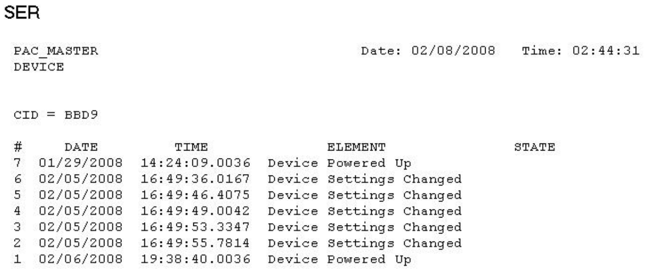
Fig. 34. Fundamental Metering Report Display



```
Status

PAC_MASTER                                Date: 02/08/2008   Time: 02:42:05
DEVICE

Serial Num = 2007351053
CID = BBD9

SELF TESTS (W=Warn)
  FPGA  GPSB  HMI   RAM   ROM   CR_RAM  NON_VOL  CLOCK  CID_FILE  +3.3V  +5.0V
  OK    OK    OK    OK    OK    OK      OK       OK     OK        3.21   4.92

  +2.5V  +3.75V  -1.25V  -5.0V  BATT
  2.46   3.73    -1.22   -5.04  3.28

Option Cards
  CARD_C  CARD_D  CARD_E  CARD_Z
  OK      OK      OK      OK

Offsets
   VA    VB    VC   IAX   IBX   ICX
  OK    OK    OK    OK    OK    OK

Device  Enabled
```

Fig. 35. PAC Status Report Display

Fig. 36 illustrates an SER report display. This shows the time and date of power cycle and settings changes. Thus, the end user can correlate these events to expected or unexpected changes in PAC behavior. Once in service, this display will also contain change-of-state SER including the user-defined state label.



```
SER

PAC_MASTER                                 Date: 02/08/2008   Time: 02:44:31
DEVICE


CID = BBD9

#     DATE         TIME              ELEMENT                STATE
7  01/29/2008  14:24:09.0036  Device Powered Up
6  02/05/2008  16:49:36.0167  Device Settings Changed
5  02/05/2008  16:49:46.4075  Device Settings Changed
4  02/05/2008  16:49:49.0042  Device Settings Changed
3  02/05/2008  16:49:53.3347  Device Settings Changed
2  02/05/2008  16:49:55.7814  Device Settings Changed
1  02/06/2008  19:38:40.0036  Device Powered Up
```

Fig. 36. SER Report Display

Fig. 37 illustrates the HMI Control display. From this display, the HMI operator clears reports, sets IED time, resets targeted alarms, and pulses output contacts. The HMI operator also toggles remote bits within the PAC to simulate changes of state to test, commission, and troubleshoot logic and communications.

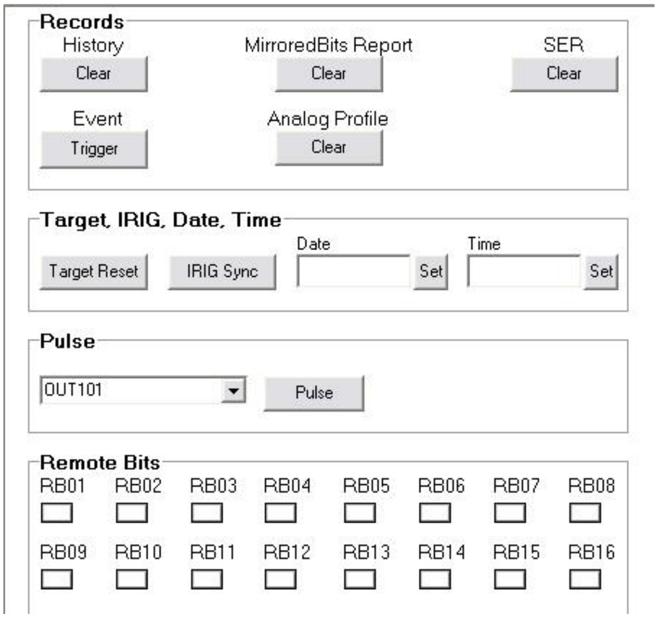


Fig. 37. HMI Control Display

This zero settings, automatic HMI is available to every end user with appropriate security access from any local or remote location. With appropriate security and supervision, it allows remotely located experts to monitor and test system performance over public communications networks, regardless of the communications links between the PAC and the end user. These remote links can be permanently disabled or disabled until they are needed.

## X. SCHEDULE MAINTENANCE VIA STANDARDIZED PAC COMPREHENSIVE REPORTING

The PAC comprehensive reporting is used to understand events, schedule maintenance, detect unfavorable trends, modify loads, and satisfy information requirements of supervisory equipment. Additional automatic HMI displays, such as Fig. 39, provide visualization of stored and trended data that illustrate system health and performance. These stored data are also retrieved and used within common PC applications to create even more useful visualization displays, such as the use of Microsoft® Excel® in Fig. 38.

### *A. Trend Function Stores Multiple Timestamped Analog Values*

When combined, SER data from individual PACs become a system-wide log of changes to monitor digital information. Once synchronized with IRIG-B time code, the system report data align perfectly. Another nontraditional RTU function performed by the PAC is analog trending. Similar to the SER for digital inputs, the trend function tracks analog channels and records the time of acquisitions and the magnitude of each quantity. Fig. 38 illustrates the value of extracting the trend report to a PC and using Excel to quickly plot the data.

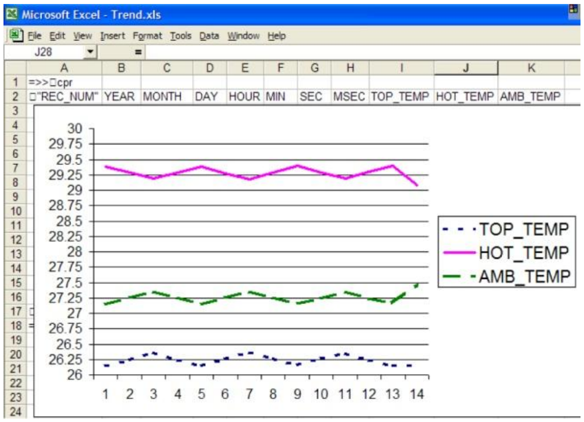


Fig. 38. Example Excel Plot of Analog Data Recorded and Trended Within a PAC

### *B. Relate Power System Disturbances to Monitored Processes*

Power system event reports are optimized for recording power disturbances and are standard offerings for power system protection devices. Now, these reports are becoming essential to network designers who want to understand not only the instrumentation and control of the primary process, but also how the underlying power system is functioning and perhaps affecting the primary process. To that end, event reports are automatically created within PACs so that they are easily related to the substation automation or industrial process.

Event reports contain ac current, voltage, and digital inputs and outputs. The report automatically adjusts content to the I/O cards used within the PAC. Reports are stored in nonvolatile memory to protect data even if power is lost. Fig. 39 illustrates an event report waveform and digital status plot created by PC software from a PAC event record.

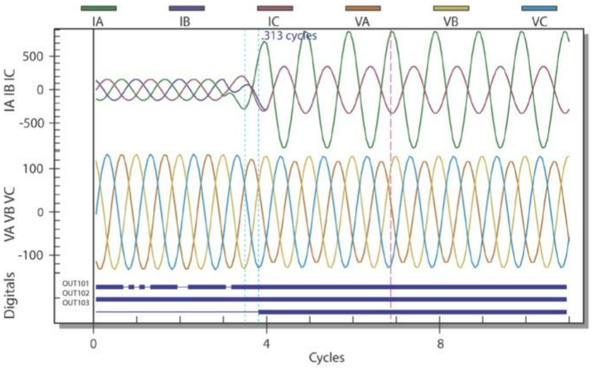


Fig. 39. Example Waveform Plot of PAC Event Report

## XI. NETWORKED IEDS IMPROVE SYSTEM CAPABILITIES

Other traits of the PAC make it an improvement over the traditional RTUs it replaces. In addition to the visualization benefit of the front-panel LCD and LEDs, the PAC also has front-panel direct-action pushbuttons. These are programmed to perform local logic or control, or to trigger GOOSE messages to another PAC; they eliminate the need for a separate control device. Electronic tags and permissions stored within the PAC supervise the controls and logic.

PACs also support serial peer-to-peer communications. These links add RTU I/O from IEDs or PACs that do not support IEC 61850. This method also enables inclusion of RTU I/O from inexpensive remote I/O modules over any distance and can even be used with electromechanical devices.

Finally, although this case study demonstrates building an RTU from dedicated PACs, the use of IEC 61850 GOOSE further blurs the definition of RTU I/O. GOOSE messages are available from protective relays, meters, PLCs, PACs, controllers, and computers without affecting their other processes or SCADA connections. As such, they can provide free I/O and will not jeopardize SCADA connections or integrated communications used for substation automation. In the future, using I/O via GOOSE messages from previously existing IEDs and new IEDs installed for other purposes will become commonplace, and the I/O in these other IEDs will grow as the traditional RTU fades away.

Integration provides significant system benefits compared with traditional methods of measuring multiple field terminations, regardless of the protocol(s) or communications media used [2].

Systems constructed with integrated IEDs networked via wireless, copper, fiber, serial, or Ethernet connections combined into a LAN offer the following benefits:

1. **Reduced field terminations, associated wiring, labor, and maintenance.**

   Reuse data detected by a single IED and digitally communicate to integrated IEDs and other data clients.

2. **Reduced quantity of unsupervised process and apparatus functions.**

   Use IEDs that, in addition to their primary functions, also perform ongoing diagnostics of their own performance and that of the equipment they are monitoring.

3. **Minimized distance of the unsupervised data path between the field source and data client(s), resulting in improved data availability.**

   Use IEDs that confirm the availability and reliability of the method by which the data are collected and alarm when the data path is broken. Supervision is maximized by replacing traditional, unmonitored copper terminations with monitored digital communications at the IED closest to the field data. This, in turn, immediately detects and alarms for communications problems.

4. **Reduced IED quantities.**

   Use newer multifunction IEDs to replace multiple, individual-purpose IEDs. IED data integration can eliminate several traditional stand-alone systems, including those that perform SCADA, metering, SER, and digital fault recording.

5. **Increased process and apparatus monitoring and control capabilities.**

   Exchange and aggregate data among many IED data sources rather than use a traditional implementation of only one IED and one data source per function. This ability to freely allocate data sources among IEDs that are networked using serial or Ethernet networks minimizes the importance of determining which IED is the data source and leads to more functional, flexible, and data-rich systems.

## XII. CONCLUSION

Using new IEC 61850 peer-to-peer GOOSE messages to replace traditional intracomponent proprietary RTU communications improves flexibility, vendor autonomy, reliability, and diagnostics.

RTUs were easily replaced with a PAC network by using only the I/O functionality. All of the additional PAC functionality creates a new generation of RTUs with much more powerful HMIs, commissioning and troubleshooting tools, and most importantly, a certainty about configuration and operation.

With these new capabilities, remote applications can transcend the RTU's traditional role of monitoring, controlling, and providing I/O logic. These applications include the following:

- PACs exchanging peer-to-peer messages between pad-mounted installation and, when used with automation capabilities, implementing intelligent isolation and restoration schemes.
- Enhancing automation and protection schemes by integrating PACs with IEC 61850-compatible protective relays.
- Using integrated troubleshooting applications, either locally or remotely, including event reports, data logs, and true millisecond accuracy SER reports to reduce outage duration and frequency.
- Using the connection-oriented nonroutable protocols with fixed datasets and file formats in the PAC to support HMI and diagnostic tools without added settings.
- Communicating via nonroutable protocols with fixed datasets and file formats to support HMI diagnostic tools without added settings. This provides communications with additional security, low processing overhead, and certainty of available data.

Messaging interoperability between peers depends on the device properties and system architecture. Commissioning tests must be performed to verify that the communications behavior of a device as a system component is compatible with the overall network design. Stand-alone network test devices, HMI applications designed to observe network messaging, and internal IED diagnostics are all essential to configure, verify, and troubleshoot network communications.

Implementing distributed PACs is an ideal way for a utility to gain experience with IEC 61850. The hybrid implementation uses the tools discussed in this paper to take advantage of IEC 61850 technologies available in new IEDs, without requiring replacement of each IED in the network.

## XIII. REFERENCES

[1] CIP Standard 005-1, Cyber Security, June 1, 2006.

[2] Fischer, H. et al., "Case Study: Revised Engineering and Testing Practices Resulting From Migration to IEC 61850," proceedings of the Power Systems Conference 2008, Clemson, SC, Mar. 11–14, 2008.

## XIV. BIOGRAPHIES

**Robin Jenkins** has a BSET degree from California State University, Chico. From 1984 to 1988, he was employed as a systems integration engineer for Atkinson System Technologies. From 1988 to 1999, he was with the California Department of Water Resources, where he worked as an associate and then senior control system engineer. In 1999, he joined Schweitzer Engineering Laboratories, Inc. where he currently holds the position of senior integration application engineer and is responsible for technical support, application assistance, and training for SEL customers in the southwestern United States.

**David Dolezilek** received his BSEE from Montana State University and is the technology director of Schweitzer Engineering Laboratories, Inc. He has experience in electric power protection, integration, automation, communications, control, SCADA, and EMS. He has authored numerous technical papers and continues to research innovative technology affecting our industry. David is a patented inventor and participates in numerous working groups and technical committees. He is a member of the IEEE, the IEEE Reliability Society, CIGRE working groups, and two International Electrotechnical Commission (IEC) technical committees tasked with global standardization and security of communications networks and systems in substations.

20090623 • TP6314-01