An Electrical Mine Monitoring System Utilizing the IEC 61850 Standard

David Christopher Mazur

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
In
Mining Engineering

Gerald H. Luttrell
Joseph Sottile
Thomas Novak
Gregory T. Adel
Erik Westman

September 12, 2013
Blacksburg, Virginia, USA

Keywords: International Electrotechnical Commission (IEC) 61850, intelligent electronic device, load side management, load shedding, energy management

An Electrical Mine Monitoring System Utilizing the IEC 61850 Standard

David Christopher Mazur

Abstract

Motor control assets are foundational elements in many industrial operations. In the mining industry, these assets primarily consist of motor control centers and drives, which are available with a comprehensive assortment of control and monitoring devices. Various intelligent electronic devices (IEDs) are now used to prevent machine damage and downtime. As motor control devices have advanced in technology, so too have the IEDs that protect them. These advances have resulted in new standards, such as IEC 61850, that have embedded intelligence and a standard set of communication schemes by which IEDs can share information in a peer-to-peer or one-to-many fashion.

This dissertation investigated the steps involved in interfacing IEDs to a mining process control network via the use of the IEC 61850 standard. As a result of this study, several key technological advancements were made including the development of (i) vendor independent system to communicate with IEDs in a mining environment over IEC 61850, (ii) command and control methods for communication based assisted automation of IEDs for mining firms, (iii) effective solutions to incorporate electrical distribution data in the process control system, (iv) enhanced safety platforms through remote operation of IEDs, (v) standard visualization faceplate graphics for HMI operators with enhanced security, and (vi) new methods for time stamped dataflow to be correctly inserted into a process historian for “true” Sequence of Events Records.

Dedication

This work is dedicated to my family. Thank you for always being there and believing in me. You have taught me over the years that I am capable of anything that I put my mind to. This document is proof that anything is possible and that the North American engineering spirit is truly alive and well. Thank you!

Ce travail est dédié à ma famille. Je vous remercie d'être toujours là et de croire en moi. Vous m'avez appris au fil des ans que je suis capable de tout ce que je mets mon esprit à. Ce document est la preuve que tout est possible et que l'Amérique du Nord d'ingénierie esprit est bien vivant et bien. Je vous remercie!

Acknowledgements

I would like to take this moment in acknowledging those who have helped me along the way in the development of this research. Special thanks to Mr. Erik Syme of ProSoft Technologies for helping me along the way with the hardware and software development of the gateway module. Thanks to Rob Entzminger of Rockwell Automation for helping me understand the current problem with process historians and being there for guidance when developing the historian event insertion tool. Thank you to Jim Elhert and Lee Ward of Rockwell Automation for helping me obtain feedback from various mining and metals firms in order to provide the community a meaningful and beneficial solution. Thanks to Blake Moret for believing in this idea, by funding my research efforts.

Finally, special thanks to Joe Sottile and Tom Novak for agreeing to serve on my committee and helping me develop this research into a beneficial result for the mining industry.

Table of Contents

Table of Figures

Table Listings

Chapter 1 - Introduction

Background

As new technology continues to drive innovative applications, the mining industry must keep pace to remain competitive in an ever-changing marketplace. The trend requires replacing outdated systems with high-performance, low-cost, option-rich devices that offer improved flexibility while reducing operating costs. Many functions of new control systems are becoming increasingly distributed to smart components capable of performing localized operations that were once the responsibility of a central or master controller. The integration of intelligent devices, device-level networks, and interrogation software into motor control centers demonstrates improved diagnostics, permits early warnings for increased system reliability, offers design flexibility, and provides for simplified wiring and an enhanced level of personnel safety.

Energy consumption is an additional concern for mining operations. The ability to integrate energy consumption and the electrical distribution network into the process control system provides the mine process engineer many advantages. This dissertation focuses on technology integration of intelligent devices via the IEC 61850 standard to traditional process control networks utilizing programmable automation controllers (PACs) for both monitoring and command and control of a mine electric distribution system.

Power system and industrial-based electrical protection devices have evolved over the past three decades from electromechanical- to microprocessor-based relays. With the advent of sophisticated microprocessor based relays, more advanced protection schemes that correctly identify and clear faults in acceptable timeframes have been developed. As microprocessor-based relay technology has evolved, so too have communication networks for electrical protection. Electrical protection architectures have developed from hard-wired contacts to communication networks using serial protocols such as Modbus. Serial communication has evolved to communication protocols over the TCP/IP stack such as Modbus TCP and DNP3 LAN/WAN, which then allowed substation devices to communicate in a peer-to-peer manner to share data.

The mining industry is the third largest North American vertical growth industry, behind oil and gas, and power infrastructure. Multiple world market leaders are participating in the power distribution system market place. Because of an increase in vendors, and the growing complexity of interfaces for power system protection equipment, IEC 61850 was developed as a global standard for substation communication. Now that technology in time synchronization, protection, and fast-acting circuit breakers exists, a truly automated substation can now be realized with the implementation of IEC 61850 across multiple equipment manufacturers**.**

The implementation of IEC 61850 also benefits automation and control companies as they now can spend resources in developing SCADA systems. Additionally, other data collection and historian systems that are inclusive of process and utility system can be controlled by devices from multiple vendors. This will allow for better Human Machine Interfaces (HMIs) as well as faceplates for operators to read and control relay operations in a substation environment.

Problem Statement

The IEC 61850 standard provides a mechanism for interoperability, i.e., the ability for information from Intelligent Electronic Devices (IEDs) from multiple manufacturers to be exchanged with each other; however, the standard is limited to power system components, i.e., relays, meters, circuit breakers, etc. If this information could also be made available to the control system, i.e., programmable logic controllers (PLCs) or programmable automation controllers (PACs), the power system IEDs could be treated identically as control system components, thus permitting a wide range of novel applications for these IEDs, e.g., optimization, demand-side load management, etc.

The electrical distribution and infrastructure systems of a mining operation are not very well monitored. There needs to be an easier way to gather information to maximize process yield and efficiency. Mining applications need to improve automation and control systems to better understand the big picture of the overall operation. Current energy monitoring and management systems are not vendor independent, thus making them difficult and expensive to maintain. This dissertation solves all of these problems and presents a functional solution for interfacing a mine distribution system to its process control system.

Scope of Work

Electrical SCADA (E-SCADA) platforms that interface with the process control system of a mine were initially investigated. Research was conducted to evaluate the current E-SCADA protocols and implementations to evaluate the pros and cons of traditional solutions. After the evaluation of these traditional standards and protocols, the IEC 61850 standard and protocol suite was evaluated for multi-vendor implementation of mining electric distribution systems.

After this initial research of current technologies, a novel solution was developed, tested, and validated. The scope of this work was limited to the background research and construction of a functional system to link a process control system to an electric distribution system.

This research produces a hardware and software solution that interfaces to the electrical distribution system. A hardware gateway solution is defined and specified as well as software developed to interface to this hardware solution. Visualization graphics are developed to allow for an easy operator and engineer interface to various pieces of distribution equipment. Additionally, a solution is specified and implemented to link traditional process historians with E-SCADA systems for the benefit of long-term data storage and trending.

Thesis Format

This thesis is divided into nine chapters that describe the design and development of a proposed interface to link an electric distribution system with a process control system via the IEC 61850 standard. The following is an overview of the chapters follows.

Chapter 1 Introduction

This chapter defines the problem statement and provides an overview of the scope of work that will be conducted and produced by the end of this document.

Chapter 2 Literature Review

Supervisory Control and Data Acquisition (SCADA) systems have been around since the conception of automation and control systems. The first SCADA systems utilized data acquisition by means of panels, meters, and lights. The operator manually exercised supervisory control by adjusting control knobs. These devices still perform the supervisory control and data acquisition of plants, factories, and power generating stations. As systems became more distributed, SCADA systems played a larger role on process efficiency and yield. This chapter discusses background information and evolution of traditional SCADA systems. Other topics discussed include electrical distribution protocols and standards as well as time synchronization standards.

Chapter 3 Conceptual Design

As previously described, the objective of this research is to design and develop a system that links the protection and metering of the electrical distribution system at a mining operation to a site-wide control system. The first steps in this process are to establish the system capabilities, develop a preliminary conceptual design, and develop a working specification for the system to achieve the research objective. This chapter discusses the development of the conceptual design for the automation and control solution that was developed.

The design process of this solution begins with an initial conceptual design based upon research, industry needs, and personal experience. In order to better fortify the design for an industrial process network, the original conceptual design was presented to industry stakeholders within the following disciplines: mining, metals, oil and gas, and engineering firms.

The original conceptual design was then compared against the engineering community input in order to finalize the conceptual design presented at the end of the chapter. The chapter begins with the original conceptual design followed by input provided by the engineering community segments listed above. Subsequently, the final conceptual design is presented and defined.

Chapter 4 Hardware Implementation

This chapter summarizes the hardware gateway module developed to implement the conceptual design described in Chapter 3. A key component to this design is the multi-threaded design to interface between the IED IEC 61850 network and the process EtherNet/IP network. Each thread including the master control program is defined, and basic functionality is explained in a functional block diagram. The various types of data that are used in this project are then addressed followed by how data is packaged in packets to be sent to the process network via EtherNet/IP.

The concept of the tag database is discussed, i.e., how it is utilized as a common space of shared memory where both 61850 and EtherNet/IP drivers would read and write various tags. The use of semaphore tags is also discussed in the database to avoid collisions in the tag database. Finally, a practical example of this hardware module is used in a mine power system application is provided.

Chapter 5 Software Implementation

In order for the hardware gateway to work properly with the process control system, software needed to be developed that allowed for user interaction and configuration of the gateway module. As described in Chapter 4, the configuration files are held by the Secure Digital (SD) card. These files are downloaded to the SD card from the software tool that is presented in this chapter. The procedure for how the hardware and software work together to produce the final solution developed in this research is summarized in this chapter. A structured example detailing the creation of a Configured IED Description, specifying the mapping of information, and creating an Add-On instruction is explored in this chapter. Examples of software are shown on the graphical user interface while additional XML code is provided to show what files are created behind the scenes allowing the solution to function. Examples of importing the Add-On instruction into code are shown as well as how the data are mapped as an object into the controller data table.

In addition, a software tool was also developed to store information in the OSI PI database on a precise timescale. The chapter discusses the traditional model of process historians and how they do not adequately function for high speed SCADA and “smart” process instrumentation. A new model is explored using off the shelf components to develop a bypass to traditional historian data interface nodes. The tool creates a linkage between two databases, SQL for alarms and events, and OSI PI via the OLE DB connector for historical data, thus allowing for post processing of time critical data into the historian environment. The historian tool is also discussed in this chapter.

Chapter 6 Visualization Benefits

This chapter defines the visualization developed for this research. Items discussed in this chapter include: faceplate definition, human/machine interface discussion and definition, and data management. The motivation for visualization is to represent an IED in logic and graphics as close to the physical object itself. This provides the operator or engineer with the same look and feel experience that they have with the physical device. At the same time, the graphics had to be developed with various levels of security to allow only users with proper credentials access to various command and control functions. Additionally, the research solution must adhere to various graphics standards for both power and process control systems. This chapter discusses the challenges in implementing a compliant visualization solution.

Chapter 7 Testing and Verification

This chapter discusses the testing and validation procedures conducted throughout this research to validate and verify functionality of the proposed solution. As this research's goal was to develop a mine monitoring solution utilizing the IEC 61850 standard to link the electrical distribution system with process control system, it was determined that the best way to validate the work and functionality of the proposed system would be to benchmark it against the conceptual design. The chapter discusses each conceptual design milestone as well as the procedure used to validate that this goal was met throughout the period of conducted research. For milestones that required more than just visual inspection, the experimental setup is defined, and testing procedure and experimental results are discussed.

Chapter 8 Practical Implementation

This chapter discusses the practical implementation of the mine monitoring system presented in this dissertation.

Chapter 9 Conclusions and Future Work

This chapter summarizes the results of this research and discusses future work in this area.

Chapter 2 - Literature Review

Introduction

Supervisory Control and Data Acquisition (SCADA) systems have been around since the conception of automation and control systems. The first SCADA systems utilized data acquisition by means of panels, meters, and lights. The operator manually exercised supervisory control by adjusting control knobs. These devices still perform the supervisory control and data acquisition of plants, factories, and power generating stations. As systems become more distributed, the need and role of SCADA systems plays a larger role on process efficiency and yield. The next few sections discuss the evolution of SCADA systems from simple sensor and instrumentation panels to more complex systems and the justification for their use in mining operations.

Mining Energy Usage

In June 2007, the U.S. Department of Energy (USDOE) conducted a study on energy consumption in the mining industry. This study sampled 20 of the largest energy intensive mineral extraction and processing facilities. The results of the study can be seen in Figures 2.1 and 2.2.

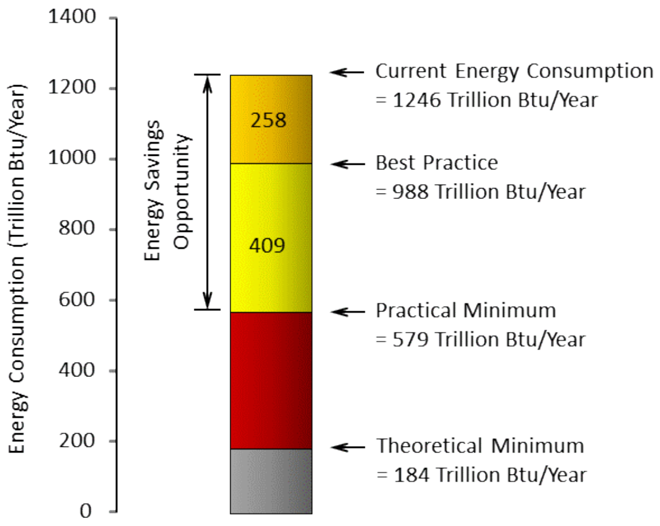


Figure 2.1 Mine Energy Usage

Figure 2.1 depicts the overall results of the study showing that a typical energy intensive mine will consume 1246 Trillion Btu/yr, with a practical minimum consumption of 570 Trillion Btu/year. The difference of 667 Trillion Btu/year represents the possible energy savings, with proper monitoring equipment [1]. Additionally, the study further analyzed various components of mining and determined that grinding operations consumed far more energy than other processes, as shown in Figure 2.2.

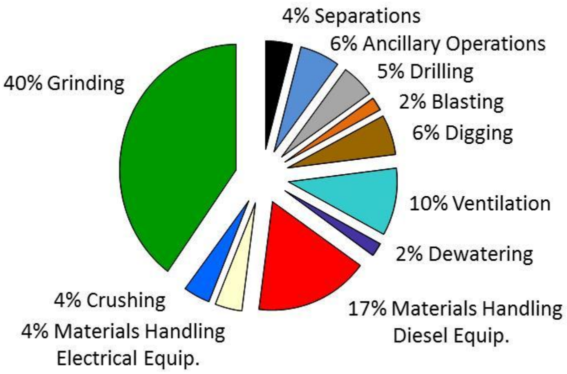


Figure 2.2 Energy Consumption by Segment

Figure 2.2 shows that grinding accounts for approximately 40% of energy usage of equipment across the mining industry, as sampled by the USDOE. A system that could monitor and trend this information would be of immense value to mine process engineers [1]. In November of 2010, the Canadian Ministry of Energy also released a study on commercial energy consumption. The results of this study can be seen in Figures 2.3 and 2.4.

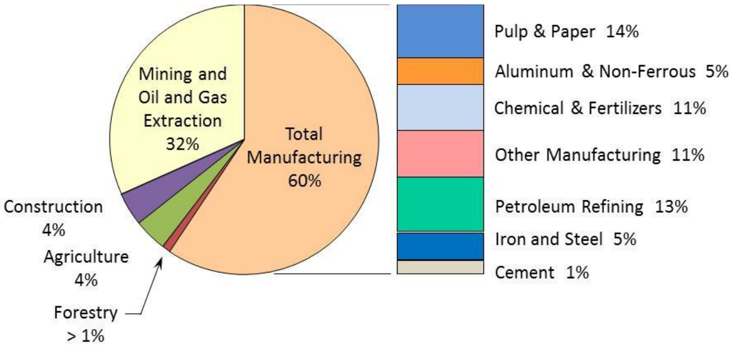


Figure 2.3 Canadian Energy Usage By Segment

Figure 2.3 depicts energy usage by industry segment. It can be seen that mining, oil, and gas extraction consume 32% of all industrial and commercial power usage in Canada [2]. Total manufacturing, which consists of many subsectors as shown on the right of Figure 2.3, consumes 60% of all energy, but the largest individual energy consumer in the nation is the mining industry. Figure 2.4 depicts the trend of energy usage. It can be seen that the mining industry has shown continuous growth from 1990-2008, while other industries have declined. As previously stated, much of this energy is consumed in crushing and grinding processes [2].

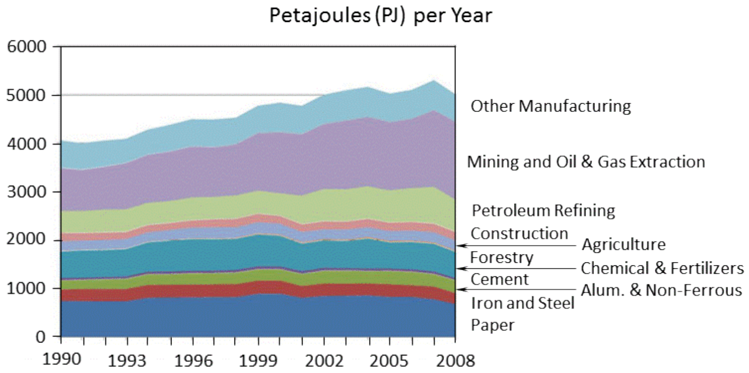


Figure 2.4 Energy Usage Per Segment

Taconite Mine Example

Taconite is an iron-bearing rock that is essential to the steel-making industry. Iron ore mining and beneficiation are very energy intensive processes that require heavy crushing and grinding procedures. To put this energy usage in perspective, in 2005 a Minnesota iron ore mine used an average of 275 MW of power and is currently Minnesota's largest consumer of energy [3]. The process begins with hauling taconite from an open pit mine to the central processing facility. At the processing facility, raw taconite is delivered to the coarse ore crusher [4]. This machine, which typically operates at 13.8 kV, reduces the mined rock into conveyor size material. The coarse crusher is typically a 7850-horsepower synchronous machine with high-torque, low-speed characteristics.

The output material of the coarse crusher is then conveyed to the fine ore crusher. The belt conveyors are typically powered by dual 4160-V, 5000-hp induction motors, that continually operate during production shifts [4]. These conveyors feed the fine crusher house, which contains multiple mills that are also driven by 4.16-kV induction machines. The mills further reduce the coarsely crushed material to 3 cm.

The fine material is then conveyed from the fine crusher house to the concentrator building, which houses rod and ball mills for grinding. These mills are driven by 4.16-kV, 2600-hp synchronous machines [4]. The output of these mills is a fine slurry mixture that is next sent to magnetic separators.

The magnetic separators remove ferrous material from non-ferrous material. The non-ferrous slurry material is pumped to the tailings pond, and the ferrous material is pumped in slurry form to the agglomeration building [4]. This building contains vacuum discs and balling drums to create taconite pellets. The pellets are then sent to a rotating kiln for drying and then to storage for shipment. These processes are performed with low voltage equipment.

A typical one-line diagram of a processing facility is shown in Figure 2.5. These taconite processing facilities are fed with 138 kV feeders from the utility.

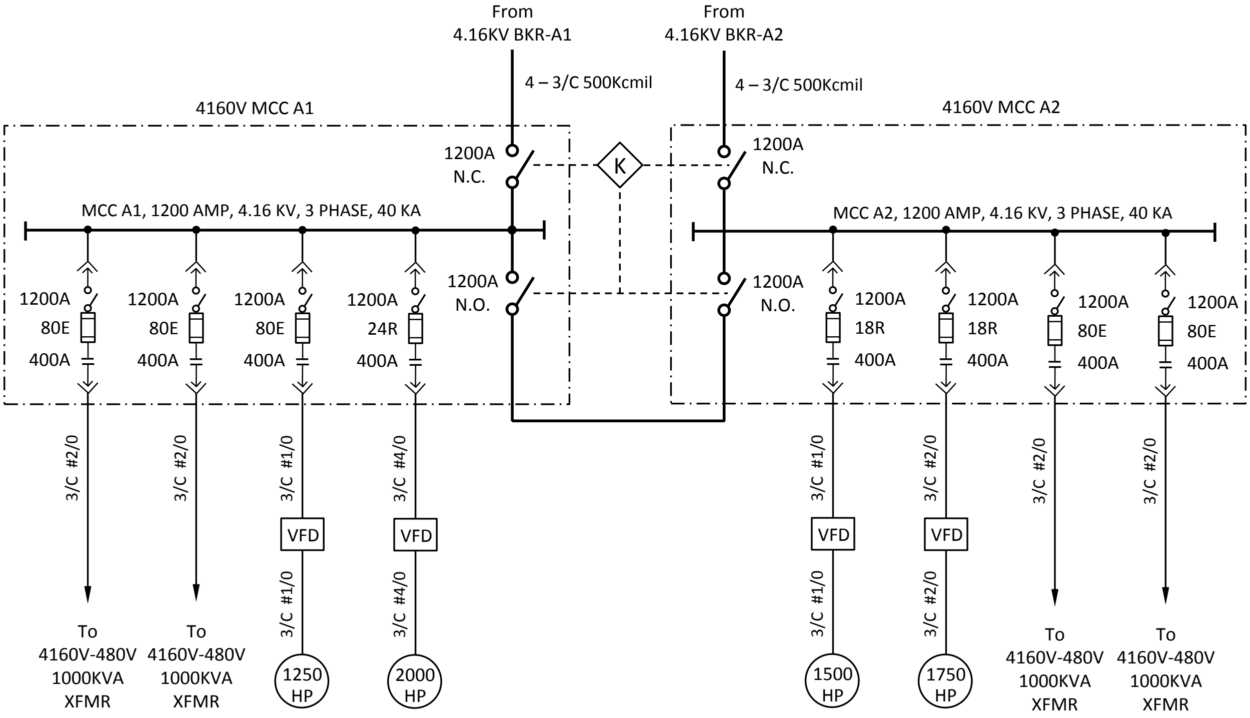


Figure 2.5 Typical Taconite Mine Distribution One Line Diagram

From this point, every facility has two feeds of 13.8-kV in a main-tie-main configuration, each with its own transformer. The 13.8-kV distribution voltage is then stepped down to 4.16-kV at each building to feed individual loads. The 4.16-kV busses are also configurable in a secondary-selective, main-tie-main scheme that can be implemented in case of bus failure. This configuration is often preferred in large mining operations because it allows critical loads to be transferred to healthy busses during times of an electrical system fault or insufficient spinning reserve. Many relays are utilized for operating and protecting these facilities, including instantaneous and time delay overcurrent (50/51 elements), differential (87 element), under-voltage (27 element), and others.

Electrical protection has evolved over the past three decades from electromechanical to microprocessor-based relays. Protection systems previously consisted of isolated, hard-wired interlocked electro-mechanical and solid-state relays that had limited system visibility. These devices only detected an electrical fault or overload and lacked a way to report additional protection information. With the advent of microprocessor-based relays, more advanced protection schemes, that correctly identify faults in acceptable timeframes, have been developed. Communication schemes for providing electrical protection have also evolved with microprocessor-based relays [5]. SCADA communications protocols and standards have been developed for the purpose of collecting information from remote process locations for use in central processing calculations. This dissertation proposes the use of the IEC 61850 standard to serve as the backbone for communications for addressing the IED-interface issue. The IEC 61850 standard can also be used for controlling main-tie-main schemes through communications rather than hardwire, in addition to its added SCADA benefits [6].

SCADA Evolution

Instrument Panels

Original SCADA systems consisted of sensor and instrumentation panels. These were panels constructed of various discreet and analog I/O sensors that were hardwired back to a reference point in a control panel [7]. A simple sensor panel can be seen in Figure 2.6.

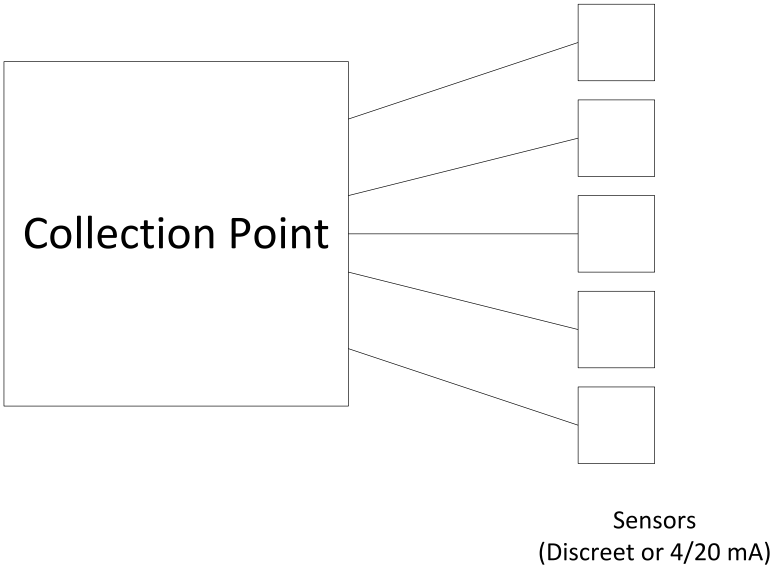


Figure 2.6 Sensor Panel

Many process owners installed these sensor panels for the many advantages that they offered, including simplicity with no CPU or programming software needed, sensors are directly connected to meters, switches, and indicators, and the cost to construct these panels is cheap.

On the other hand, these panels had the following drawbacks:

- The wire required to install a complex system could become unmanageable due to cable quantity
- The quantity and type of data received from the panel are rudimentary
- Increasing capacity of the system becomes more difficult as the system expands
- Reconfiguration becomes difficult as there is no way for data simulation using this method
- Data storage is minimal

- There is not remote monitoring of data and alarms; thus requiring a person to be present to monitor the system.

As a result of these drawbacks, the industry eventually progressed to the use of telemetry to interconnect areas of a process network [8].

PLC / DCS Systems

Modern SCADA systems often utilize telemetry to move data across long distances and connect remote nodes to their systems [7]. For example, in today's society, many heavy manufacturing and industrial processes, such as metals, mining, utilities, and security, need to connect equipment and systems often separated by large distances. These distances can range from a few feet to hundreds of miles. Via the use of telemetry, process owners and control engineers can send commands and programs and receive monitoring information over the network from various locations [9].

A modern definition of SCADA is the combination of telemetry and data acquisition. Furthermore, SCADA can be defined as the collecting of information, the process of transferring this information to a central site, performing calculations to take corrective control action of a process, and enabling actuators to take the corrective control action [7]. SCADA has also evolved to displaying and trending data on operator and engineering workstations over the past few decades.

At its conception, SCADA systems used relay logic to control production and plant processes. With the evolution of CPUs and electronic devices, manufacturers, such as Modicon, Allen-Bradley, and Siemens, began to implement this technology into relay logic equipment. The result of this technology integration birthed/created the Programmable Logic Controller (PLC). The PLC is still one of the most widely used control systems in industry today [10]. Instead of hardwiring a relay into a control panel, a sensor was simply wired back to an I/O point. Computer logic, known as ladder logic due to its similar form of a relay ladder diagram, was written to control the state of these devices [7]. As the need to monitor and control field devices grew, PLCs were distributed, and control systems began to become more intelligent and smaller in size. An example of a PLC or Distributed Control System (DCS) can be seen in Figure 2.7.

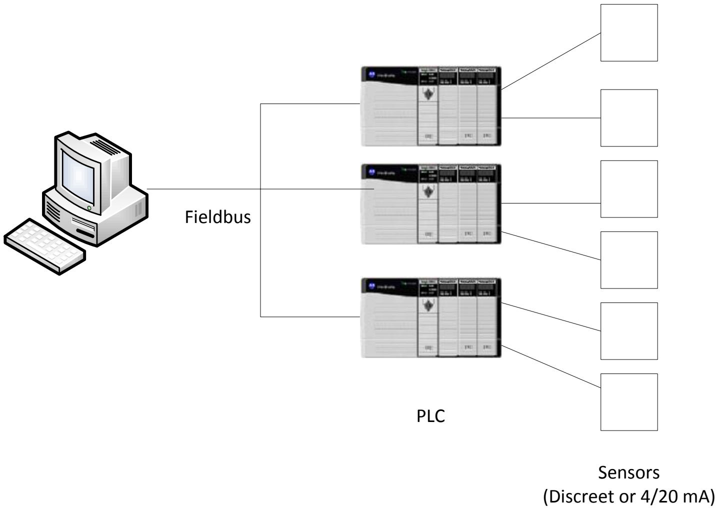


Figure 2.7 PLC System

This system had many advantages including the computer's ability to store and records a large amount of data, the user's ability to customize data, connect sensors over a wide geographic range, real time simulations for operators, and access to data from remote locations. While DCS systems have many advantages, the system is much more complex than simple panels; it requires programming knowledge to configure the system, there are wiring concerns from the sensors to the PLCs, and the operator only has vision to as far as the PLC in the network [7].

IED Systems over Fieldbus

As manufacturers and process owners began to push the envelope of SCADA and process control systems, the need for smaller and smarter systems grew, and as a result, sensors were designed with the intelligence of PLCs [7]. These new intelligent sensors became known as Intelligent Electronic Devices (IEDs). As process engineers began to design systems from process and instrumentation diagrams (P&IDs), a demand grew for these intelligent electronic devices to perform both discreet and analog control, including analog output and PID control [8]. The IEDs were located on a fieldbus network, such as Profibus, DeviceNet, or Foundation Fieldbus, where information could be exchanged from IEDs to PCs and other enterprise systems. IEDs in and of themselves have enough intelligence to acquire data, communicate in a peer to peer fashion, and contribute to the overall control system. These IEDs typically contained multiple sensors for analogs and discreet inputs. An example of an IED system over a Fieldbus network can be seen in Figure 2.8.

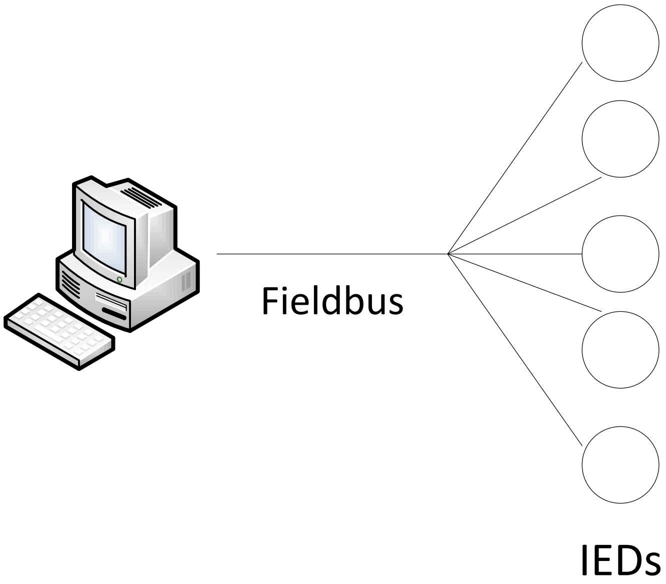


Figure 2.8 IED and Fieldbus SCADA System

Advantages of this system include the minimal need for wire, operator visibility to the sensor level, the inclusion of sensor firmware information with received data , the plug and play ability of devices for simple installation and replacement, and smaller physical footprint. The disadvantages of this system include: the cost of training required for operators, higher sensor prices, and IEDs that rely on a communication scheme for data transfer [7].

PLCs to PACs

Over the past 15 years PLCs have evolved to Programmable Automation Controllers. These platforms are a rack based automation controller that accepts multiple modules for various configurations. PACs were chosen because of its asynchronous capabilities. Most PLCs are synchronous in nature in that they execute the following procedure: read inputs, execute a program, and update output registers [11].

Asynchronous machines utilize two different sets of memory, program and I/O. These two memory banks are in separate locations and can be written to independently. This means that I/O information can get collected separately from the executing instructions in program memory. I/O memory is populated via the use of a backplane circuit and a Requested Packet Interval

(RPI). Multiple data words can be simultaneously shared between modules connected to the backplane as well as written to I/O mapped memory locations [10]. PAC processors are capable of true multitasking, thus allowing the system to asynchronously gather information and provide precise time tags while normally scheduled programs continue to operate. Many process owners today are replacing PLC SCADA systems with PAC systems that work in conjunction with IEDs on the same fieldbus for monitoring and process control.

SCADA Terminology

A SCADA system can be divided into two categories, hardware and software. It is important that these two categories function equally in order for the SCADA system itself to be self-sustaining and provide proper control for the process system.

SCADA Hardware

A traditional SCADA system consists of a number of remote terminal units (RTUs) that collect field data and send it to a master station via a communication system. The master station then displays the acquired data and allows the controls engineer to make process and control decisions for control tasks. This acquisition of data in a timely fashion allows for process optimization [7]. Other SCADA benefits include more efficient, reliable, and, most importantly, safety operations. In all, this will result in lower cost of operation compared to non-automated systems [8].

Complex SCADA systems consist of a five level hierarchy:

1. Field Level Instrumentation and Control Devices
2. Marshaling terminals and RTUs
3. Communication Systems
4. Master Station
5. Commercial data processing department computer system

The RTU provides the SCADA system interface to the field analog and digital sensors situated on every remote node. The communications system provides the connection between master and remote sites. This communication system can be defined by a number of media including: wire, fiber optic, radio, telephone line, microwave, or even satellite. Specific SCADA protocols and error detection philosophies are used for efficient and optimum data transfer between nodes [7].

Master Stations gather information from various RTUs and generally provide an operator interface for information displays and control of remote nodes.

SCADA Software

There are two types of SCADA software: proprietary and open. Manufacturers engineer proprietary software to communicate with their hardware. These are typically seen in "turnkey" solutions where all engineering, integration, and startup are provided from a single vendor or encompass group. There is one main failure with these SCADA systems: the process owner heavily depends on the supplier of the system [7]. As a result, many process owners are supporting open SCADA systems. Due to the interoperability these systems bring to the system, multiple vendor manufacturer's equipment can be used in the same system.

Major vendors of SCADA system software include Citect and WonderWare, but there are many manufacturers of SCADA software. Many packages today actually include asset management in order to provide process owners more information about their systems. Key software features for SCADA system software include user interfaces, graphical displays, alarms, trends, RTU interface, scalability, access to data, database, networking, fault tolerance and redundancy, and client/server distributed processing[8].

SCADA Protocols

This section discusses SCADA protocols. Since this dissertation addresses the electrical distribution system in mines, this section focuses on the three main electrical SCADA protocols: Modbus TCP, DNP3 LAN/WAN, IEC 60870-5-103, and IEC 61850. To be fair, there are many electrical distribution protocols that can be implemented in any automation and control system. Figure 2.9 depicts a listing of various electrical distribution SCADA protocols between the bay and station level of a traditional electrical distribution substation.

In general, an electrical distribution substation can be broken into three levels of control: station level, bay level, and process level. The process level of a substation houses the instrumentation used to collect data, i.e. potential and current transformers. The bay level consists of IEDs such as relays and meters that collect the process level data and make a distributed control decision. In this case, control decisions may be network configuration, energy usage, or load shedding, to name a few. Finally, the station level of a substation consists of Human Machine Interfaces (HMIs), central supervisory control functions, and a station gateway to send gathered status and data to further upstream substations or master controllers [8]. Traditionally, this link is usually wireless or fiber optic.

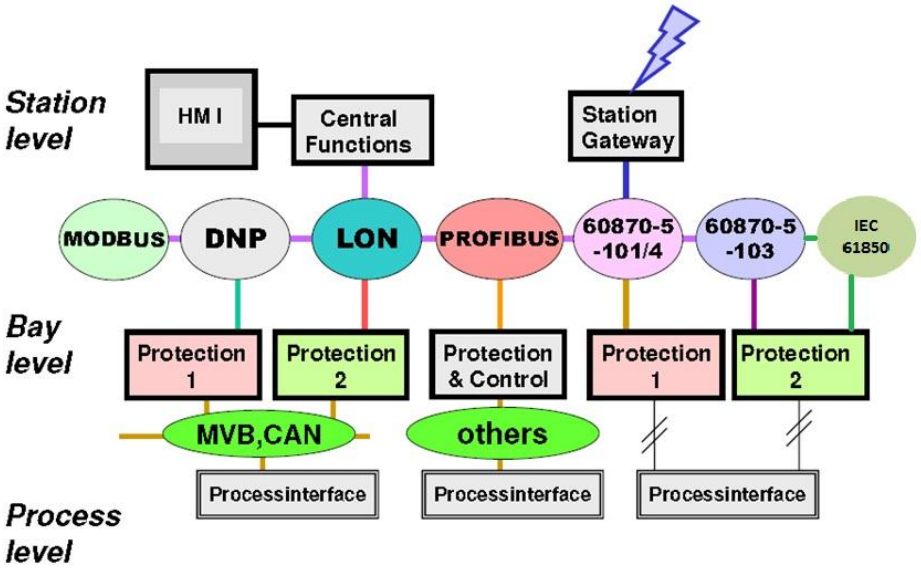


Figure 2.9 Typical Electrical SCADA System

Modbus TCP

Modbus is a master-slave communication model that was developed by Modicon in 1979. This communication protocol is well established with over seven million nodes in North America and Europe alone. This protocol is very simple to implement and commission. Although this protocol is very easy to implement, there are many drawbacks to this method. Modbus TCP is very inefficient at managing data and network bandwidth. The protocol only uses simple data types, such as integer and Boolean. Finally, it only can send static data. Despite these drawbacks, Modbus is still the de facto standard in a multi-vendor integration [12]. Additional limitations to Modbus include no time stamp of data values for Sequence of Events (SOE) Applications, no indication of a disturbance event, the need for the master to always ask slave device for data, the inability of the slave to initiate communications to master, and the lack of common data formats between devices. As a result, these limitations made it critical for the power industry to find a communication protocol that provided data that more accurately represented the events that occurred in a power application.

IEC 60870-5-101/103/104

As more services were being performed by IEDs in a substation, more intelligence was required in the protocol that linked these devices. This need, combined with the limitations of existing communication protocols, such as Modbus, led to the formation of new protocol standards. In 1990, IEC created the IEC 870-5-1. This standard was then spun off into two separate standards: IEC 60870-5-101 in 1995 and DNP 3.0 in 1993. This creation can be seen in Figure 2.10.

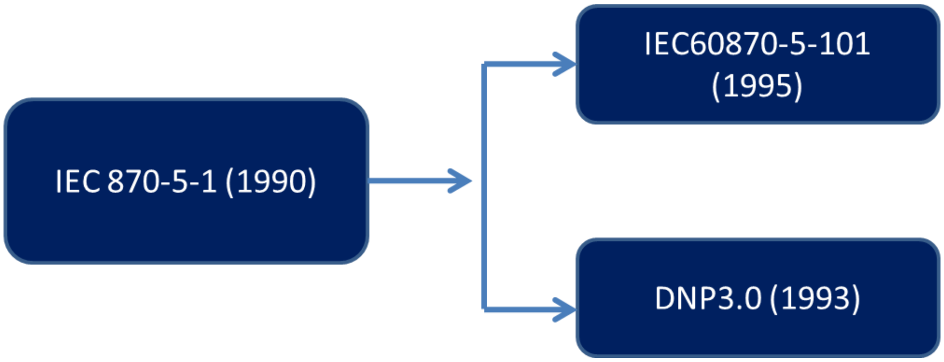


Figure 2.10 Evolution of Electrical SCADA Protocols

Industrial communications protocols, such as IEC 60870-5 and DNP 3.0, provided unique attributes that made data acquisition, data reporting, and overall communications much more efficient and provided greater levels of detail to the devices in the network. DNP 3.0 is dominant in North America, Latin America, South Africa, and Australia and is primarily used in the water/wastewater, and oil and gas segments [13]. The IEC-60870-5 protocol suite is primarily used in Europe, the Middle East, and Asia Pacific regions. Both protocols provide similar application functionality and are popular in electrical SCADA applications for communication of IEDs [14].

The IEC 60870-5-101/103/104 standards are used for power system monitoring and control in the following: 101 defines serial communication, 103 defines protection relay data formats, and 104 defines the Ethernet implementation of the protocol. IEC 60870-5-101 is the first protocol to define a hierarchy of message structure. The protocol breaks data into two formats, high priority messages (Class 1), and low priority messages (Class 2), and transfers data using separate mechanisms. The 101 section of the standard also defines cyclic and spontaneous updating schemes as well as the facility for time synchronization [15].

The 103 section of the standard defines power system control and associated communications. It defines a mechanism that enables interoperability between protection equipment and devices of a control system in a substation. The mechanism defines either two methods of data transfer: the use of specified application service data units (ASDUs) or the use of generic services for transmission of all possible information. This standard supports specific protection functions and provides the vendor a facility to incorporate its own protective functions on private data ranges [16].

The standard supports two modes of data transfer. The first mode of data transfer is unbalanced which is defined by a master initiated message. The alternative to unbalanced communication is balanced which is defined by a master/slave-initiated message. As previously mentioned, data is classified into different information objects, each with its own specific address. The process owner must then classify data into Class 1 and Class 2 data in order to define transfer mechanisms [8].

The IEEE 802.3 Ethernet portion of the standard (104) classifies data into sixteen separate data groups that can be individually interrogated. It allows for cyclic as well as spontaneous data updating of data tags in addition to allowing for the benefit of time synchronization [8]. These improvements from Modbus allowed for better data transfer with more meaning to each data point. The ability to timestamp values, group non-similar data points into custom groups, and interrogate data by two levels of priority provided large improvements over traditional Modbus schemes [15]. The first implementations of DNP 3.0 developed from the IEC 60870-5 specifications.

DNP3 LAN/WAN

DNP 3.0 was originally developed as a serial communications protocol for electrical SCADA applications. This section will refer to the Ethernet based form of this protocol known as DNP3 LAN/WAN. This protocol is primarily used for communications between a master station and IEDs or RTUs [8]. This model can be seen in Figure 2.11.

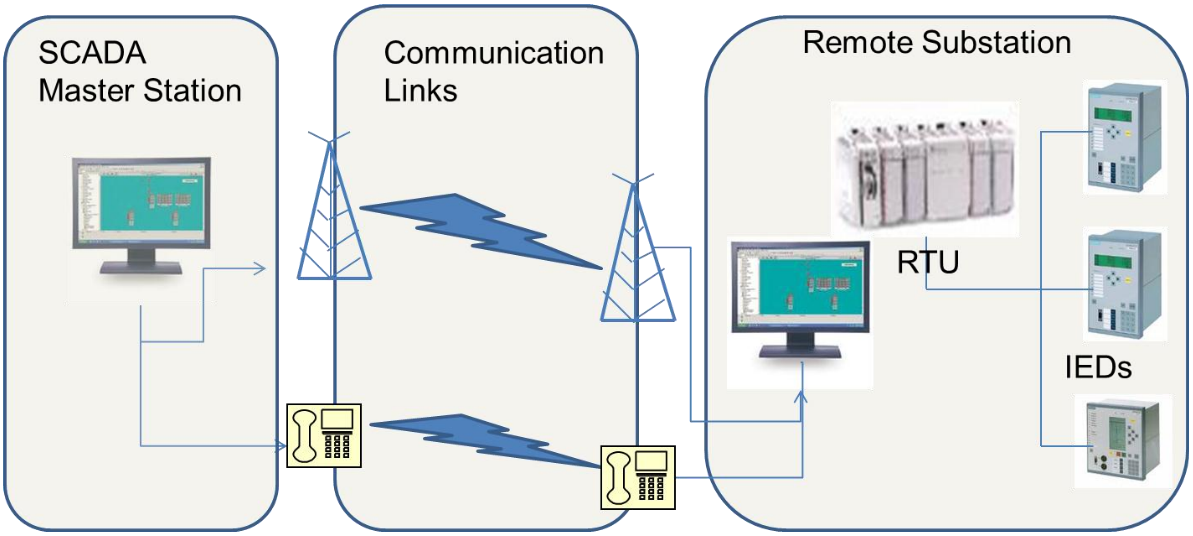


Figure 2.11 DNP3 LAN/WAN Implementation

DNP 3.0 was designed for and is heavily used in the following industries: substation automation, wind power generation, solar power generation, oil and gas, mining and minerals, and water wastewater. The traditional form of DNP 3.0 uses a multi-drop configuration, which can be seen in Figure 2.12. In this configuration, the DNP Master is connected to DNP slaves over a network. This creates a network hierarchy in which the slaves report to the DNP Master [17]. This configuration can be expanded to multiple levels as seen in Figure 2.13.

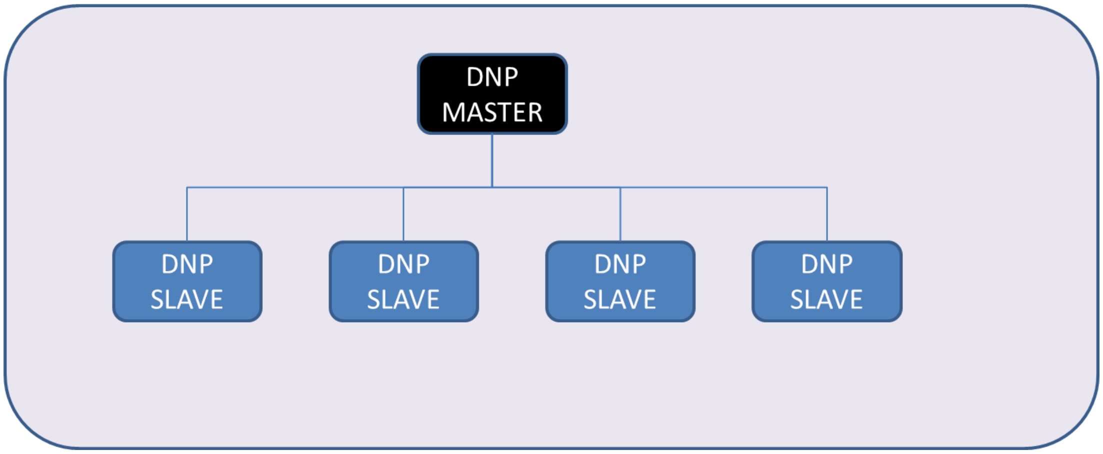


Figure 2.12 Multi-drop Configuration

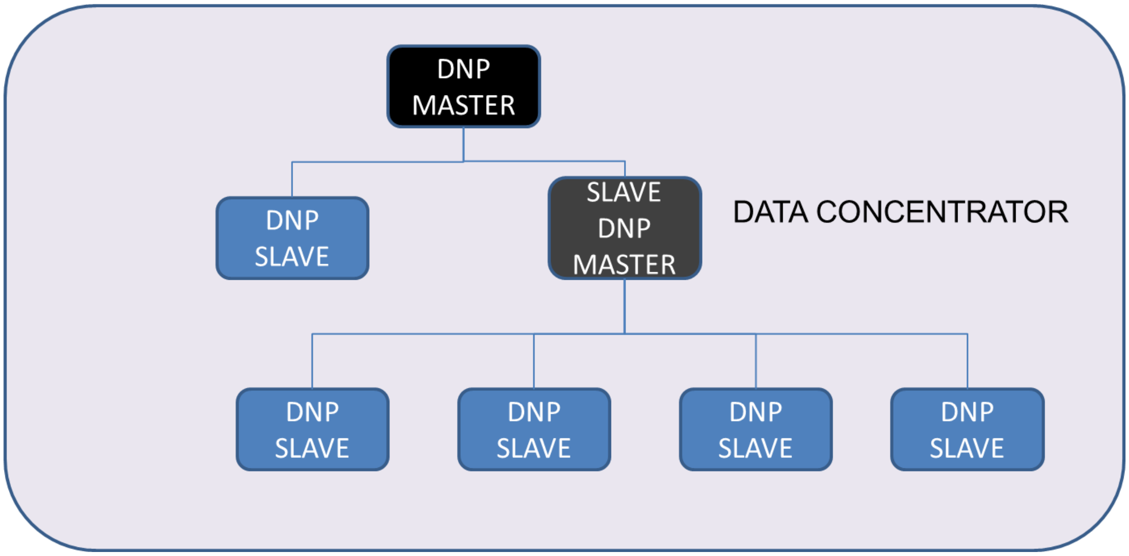


Figure 2.13 Multi-Level DNP3.0 Network

In the multi-level drop configuration, there is one ultimate master within the system (represented by black). All of the remaining nodes within the system are slaves to this master. In addition, each level of the network designates a master to the lower levels of the network (represented by gray). This device serves as a data concentrator to the devices below it to gather data and push it to higher levels of the SCADA system [17].

DNP3 LAN/WAN builds upon IEC 60870-5 in that it supports complex data types, including integers, double integers, reals, Booleans, dual point binaries, and counters with quality flags. The protocol also allows for the time stamping of records based on event change of state as well as analog deadband. The DNP3 LAN/WAN protocol also allows for unsolicited reports by exception. Data is grouped into two types, dynamic and static. All dynamic data is grouped into three levels of priority, Class 1, Class 2, and Class 3, while all static data is referenced to Class 0 [8].

Process owners have adopted DNP3 LAN/WAN because it provides standardization and interoperability. It is an open protocol and is optimized for SCADA communications. DNP3 provides interoperability between various vendors equipment and is supported by a substantial number of SCADA equipment manufacturers [8]. The protocol is not static or unchanging; it allows for extensions. Users can define complete data structures to pass industry specific information in complete context [17].

Today industry is moving away from IEC 60870 and DNP 3.0 in the adoption of a new SCADA protocol standard known as IEC 61850. This migration is occurring for a multitude of reasons including: the advantages of high speed Ethernet, the high cost for data management, the loss of information and functions during mapping of data, and the lack of consistency between vendor implementations of similar devices [16].

IEC 61850

Introduction

Power system and industrial-based electrical protection has evolved over the past three decades from electromechanical to microprocessor-based relays. With the advent of microprocessor-based relays, more advanced protections schemes that correctly identify faults in acceptable timeframes have been developed. As microprocessor-based relay technology has evolved, so too have communication networks for electrical protection [5]. Electrical protection architectures have developed from hard-wired contacts to communication networks over serial communications such as Modbus. Serial communication has evolved to communication schemes over the TCP/IP stack such as Modbus TCP and DNP3 LAN/WAN, which now allows substations devices to communicate in a peer-to-peer fashion to share data.

With power systems being the second largest North American vertical growth industry behind oil and gas, world market leaders are participating in the power system market place. Furthermore, the mining industry is the largest consumer of electric power in the world [18]. Because of an increase in vendors and growing complexity of interfaces for power system protection equipment, IEC 61850 was developed as a global standard for substation communications. Now that technology in time synchronization, protection, and fast acting circuit breakers currently exists, a truly automated substation can be realized with the implementation of IEC 61850 across multiple equipment manufacturers [19].

The implementation of IEC 61850 also benefits automation and control companies as they now can spend resources developing SCADA systems, as well as other data collection and historian systems, that interface to multiple devices over one common protocol. This allows for better Human Machine Interfaces (HMIs) as well as faceplates for operators to read and control relay operations in a substation environment.

History

IEC 61850, “Communication Networks and Systems in Substations,” is an international standard developed by Technical Committee 57 that focuses on substation automation [20]. This standard was drafted and adopted in an effort to unify substation equipment and communications on one common platform regardless of manufacturer. IEC 61850 is divided into ten sub-sections ranging from 61850-1 to 61850-10 that define protocol terminology, communication mechanisms, and conformance testing [20-35].

IEC 61850 was a combined international effort merging the Utility Communication Architecture 2.0 (UCA), an EPRI project, with IEC standard 60870-5-101, -103, and -104 to create one international standard for substation automation. UCA defined many of the protocols, data models, and abstract service definitions, while IEC 60870-5 defined the basic communication profile for sending control messages between two systems [36].

There are multiple benefits to the IEC 61850 standard. These benefits include the support of comprehensive substation functions, ease of design, specification, setup, and maintenance, strong functional support for substation communication, and its flexibility to support system evolution [36]. Additionally, the standard was structured in such a way as to accommodate current technology.

Protocol Definitions

IEC 61850 was designed as an object-oriented protocol in that data would be defined as objects, and each object would have multiple attributes. This is very similar to protocols of high level programming languages. Each physical Intelligent Electronic Device (IED) would be accessed by a network address. The physical device would then be represented by a logical device that contained all relevant, non-distributed, logical nodes (i.e. functions in the real device). Each logical node would then contain data and data attributes which would describe operations, positions, etc. of the IED. The data and data attributes are defined as dedicated data values that are structured and well-defined. These values are exchanged according to defined rules and communication mechanisms [24]. The way that this information is shared across the network is defined by communications and mapping portions of the 61850 standard [24].

The traditional grid has evolved from analog electro-mechanical relays with one-way communication that was human monitored to digital electronic relays with two-way communication and managed by advanced control schemes. As previously described, with the advent of digital IEDs, communication schemes became more advanced than the traditional hardwired contact. Communication protocols such as Modbus TCP and DNP3 LAN/WAN were developed by the power industry specifically for advanced control schemes. With the development of the IEC 61850 standard, three main forms of communications are defined: GOOSE, MMS, and SMV.

The IEC 61850 protocol is based on the Generic Substation Event Model (GSE). This model is a fast and reliable system-wide distribution of input and output data values [36]. The data transfer is based upon a publisher-subscriber mechanism. Additionally, simultaneous delivery of the same generic substation event information to more than one IED can be achieved via the use of multicast services. GSE describes the general structure for an event, and the mechanism for transferring information between devices and locations are defined by GOOSE and MMS.

The Generic Object-Oriented Substation Event (GOOSE) is one way to transmit information under the IEC 61850 protocol. GOOSE is a single message sent by an IED, but it can be received by multiple targets in a peer-to-peer fashion [32, 36]. It is used for fast, reliable transmission of substation events such as alarms, commands, and indicators. In general GOOSE is used for high speed, high priority applications such as protection interlocking and events when event information needs to be delivered in a peer-to-peer fashion.

The Manufacturing Message Specification (MMS) is an alternative to transmitting messages over GOOSE that is generally preferred for SCADA or non-time critical time data acquisition. MMS is defined by ISO standard 9506 and is widely used in control networks. The standard defines a reduced OSI stack with TCP/IP capability and Ethernet or RS232C as physical media [37]. MMS defines communication messages transferred between controllers as well as between engineering stations and controllers. Each IEC 61850 object is mapped to a corresponding MMS object, and each IEC 61850 service is also mapped to a corresponding MMS operation. All but GOOSE messages and the exchange of sampled values are mapped to the MMS protocol stack, with the exceptions of time synchronization and file transfer. The application described within this paper uses MMS to transmit report information from IED to an automation controller.

The IEC 61850 object model is based on commonly used protection, control, and metering functions in electrical substations. As shown in Figure 2.14, each physical IED can include one or more IEC 61850 logical devices. Within each logical device, the manufacturer enables logical nodes to represent the primary functions of the device. Each logical node encapsulates a collection of data tags associated with the function in question. For example, in a feeder relay commonly used for industrial power systems, the manufacturer defined logical devices for protection, metering, control, and annunciation. Contained within the protection logical device, are a number of logical nodes related to the operation of time overcurrent elements. Specifically, in metering, the logical nodes are related to the variety of metering values in the relay [38].

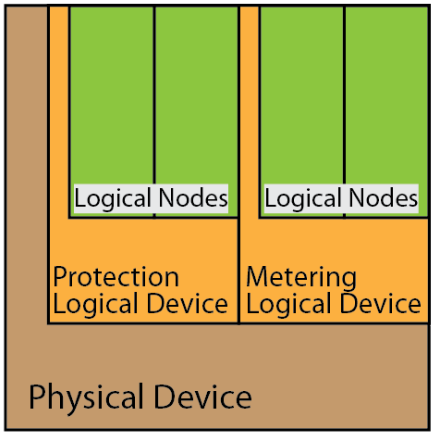


Figure 2.14 Virtual IED

Most engineers are accustomed to SCADA protocols that reference tags by address or index; one of the desirable attributes of IEC 61850 is that tags are referenced by a structured name. As an example, A-phase electrical current is represented in a Polyphase Measurement Unit (MMXU) logical node as MMXU$A$phsA$cVal. Table 2.1 describes the components of this tag name. Additionally, each IEC 61850 enabled IED can self-describe all of the logical devices and logical nodes it contains. A number of software manufacturers have IEC 61850 browsers that query devices on the network and display available data from each IED. Although engineers commonly configure network communications using an offline Substation Configuration Language (SCL) tool, which will be discussed later, self-description is a very useful function for evaluating or verifying the configuration of a particular IED [39].

Table 2.1IEC 61850 Descriptor Example

Descriptor	Component Type	Description
MMXU	Logical Node	Polyphase measurement unit
A	Data Object	Phase-to-ground amperes
phsA	Sub-Data Object	Phase A
cVal	Data Attribute	Complex Value

The authors of the IEC 61850 standard defined many types of standard logical nodes that are used by many manufacturers of protective relays. The common tag naming and structure of the logical nodes help to simplify the integration of an HMI and other systems with power system protection equipment. At the same time, it is important to understand limits to the standard.

First, the standard does not dictate which logical nodes are implemented in a given relay. Also, there is no way to define how the internal memory values of a given relay are mapped into IEC 61850 tags. Each manufacturer chooses which logical nodes to implement and how manufacturers or nodes associate their relay's functions to each logical node. Additionally, the standard allows for custom logical nodes that can be created by any manufacturer. During system design, the integration engineer needs to verify the existence and location of needed data from a target relay.

Substation Configuration Language

In order to have a common method for describing and documenting the communications network, IEC 61850-6-1 defines various SCL file types based on XML schemas. The specified file types are System Specification Description (SSD), Substation Configuration Description (SCD), IED Capability Description (ICD), and Configured IED Description (CID). This dissertation only focuses on ICD and CID files. ICD files represent the default IEC 61850 configuration of an IED. CID files follow the same schema but represent the final configuration of an IED in service.

If an ICD file were opened in a web browser or text editor, one would find the definition of all logical devices and logical nodes for the IED. In addition, the ICD file can include definitions for datasets, MMS reports, and GOOSE messages. Datasets are simply logical collections of tags (not necessarily from the same logical node). Collecting the tags into datasets allows those tags to be efficiently defined as part of a GOOSE message or MMS report.

Reports are unsolicited methods of sending datasets from an IED. The standard defines two types, buffered and unbuffered. When using buffered reports, the IED keeps track of a client message receipt so that any missed reports can be re-sent if there is a network problem. Unbuffered reports do not fill in the gap if the link is lost. The MMS protocol can also provide datasets via direct polling by the client. This method provides no buffering and eliminates deadbands on analog datasets.

Automation Limitations

Even though SCL files define how an IED communicates on an IEC 61850 network, many people do not realize that these files do not include configuration information for the protection and control functions in a relay. Each manufacturer has software and proprietary methods for enabling and configuring various protection elements and control strategies. The IEC 61850 standard also provides no standard method for designing communication-assisted automation [39].

Time Synchronization

Importance of Time in Applications

Everything in today's society is synchronized to some form of clock or source of time, and it is important that in coordinated processes that these clocks be synchronized to provide the most efficient process yield possible. Take the example of having a local controller with a remote I/O device. Currently, measurements are typically taken by a controller-based I/O to ensure that accurate timestamp information of events is recorded. Due to advancements in time synchronization, this application can now be addressed by utilizing distributed I/O [40].

A sensor now can be connected to a remote I/O device. If each device contains a clock, and they are time synchronized, discrete transitions from off-to-on and on-to-off can be captured and timestamped as an input event. This timestamp now has meaning since both clocks are now synchronized to each other, and this information can be sent across the network and used by the controller or any other enterprise level process. Since the clocks are synchronized, the controller can use the information to record sequences of events or timestamped data logs [41].

As a result, more than just traditional data can be sent across a time synchronized network. In addition to timestamped inputs, scheduled outputs and synchronized actuation can be achieved over time-synchronized networks. Four major areas of improvement can be realized with the implementation of time-synchronized networks: sequence of events recording, timestamped data logging, coordinated/synchronized operation, and motion control.

For power distribution applications, time synchronization plays a major role in everyday process administration. Examples of time synchronization can be seen in Sequence of Events recording (SER), protection schemes, power quality measurements, and data acquisition. Microprocessor relays typically have the capability to synchronize to a common reference time source such as GPS through IRIG-B [42]. Relatively newer technologies, including wide-area measurement systems using synchronized phasor measurement units (PMUs), apply a GPS time stamp to real-time data frames to capture a snapshot of the power system. More traditionally, remote terminal units (RTUs) within a substation record events from devices throughout the station and report statuses to a SCADA (Supervisory Control and Data Acquisition) Master, commonly at a control center [9]. Although SCADA is not a real-time application due to its synchronous design, a valid timestamp of events from substation equipment via either an RTU or other intelligent electronic device is beneficial for both online and offline applications.

Mine applications are no exception to this need for time synchronization for both electrical distribution systems and processes themselves. Most mines or processing facilities use some form of backup or co-generation in order to power their processes. As a result, cascading failures can occur that would cause massive system and process shutdowns, costing process owners money for every moment they are not in production. Tracing the root failure of these electromechanical systems can take a large amount of time and manpower. By having synchronized SCADA systems, the downtime to locate the problem and take corrective action becomes greatly shortened.

SCADA Timing Protocols

A point was made in each SCADA protocol definition to discuss the time synchronization capabilities of each protocol. The timing protocols that will be discussed in this section include GPS, IRIG, NTP, and PTP. Each protocol is compared to and contrasted with its counterparts based on electrical SCADA conformance.

Global Positioning System

Global Positioning System (GPS) is not only a navigation system; it has evolved to become the world's primary means of distributing precise time and frequency. GPS was developed by the United States Department of Defense (DoD) in 1978 and is still maintained by the organization today. Currently, 31 satellites orbit the earth and provide accurate time within +/- 10 ns to GPS receivers [41]. The US DoD dictates the accuracy of the system and has the right to limit this accuracy. Additionally, GPS systems can be costly to deploy over a distributed I/O network since a GPS receiver is required at each node, and the antenna must have a clear view of the sky to attain a locked signal.

Inter-Range Instrumentation Group Time Codes

In 1952, the commanders of the U.S. guided missile test ranges formed the Inter-Range Instrumentation (IRIG) group as part of the Range Commanders Council (RCC) of the U.S. Army in order to share information about range instrumentation [43]. Today, the steering committee and ten technical working groups, including the Telecommunications and Timing Group (TTG), control the IRIG time code standards. The IRIG standard was last updated in September, 2004, and is titled IRIG Serial Time Code Formats [42]. The most well-known and utilized code format is the IRIG-B time code from this standard. Despite multiple choices of IRIG code formats, IRIG-B is the mostly widely used in the power industry. In fact, more than ninety percent of substations use the IRIG-B format, which is typically better than +/- 1 ms [44].
There exist two different forms of the IRIG-B time code: modulated and un-modulated. Modulated IRIG-B is transmitted on a carrier frequency sinusoid (similar to AM radio) and as a result, must be demodulated at the receiving end in order to interpret the time data. With advancements in phase locked loop (PLL) technology, modulated IRIG-B schemes can obtain accuracies to within +/- 10 µs [45]. Installations with modulated IRIG-B must be isolated

through some form of transformer in order to prevent ground loops and signal degradation. Unmodulated IRIG-B, or level shifted IRIG, is an alternative to the traditional modulated approach. This form of time code transmission uses digital level shifting to achieve data transfer. As no demodulation is necessary at the receiving end, time synchronization accuracy improves to +/- 1µs [45].

Network Time Protocol

Network time protocol (NTP) is a protocol for synchronizing computer clocks using a data network, such as the intranet or a wide area network (WAN). This protocol was developed at the University of Delaware in 1980 and is the first protocol to address time synchronization over variable latency packet switched networks [41]. This protocol provides accuracies that depend on the setup of the network between each device and the performance of the computers' operating systems. Ideally, the connections in the network should be as short as possible, but this protocol does include methods to estimate and account for round-trip path delay. Overall, the accuracy of this protocol is in the low tens of milliseconds over wide area networks (WANs) and better than a millisecond over local area networks (LANs) [46].

Precision Time Protocol

The formal title of IEEE 1588 is Precision Clock Synchronization for Networked Measurement and Control Systems. This protocol is better known as Precision Time Protocol or PTP. The standard specifies a protocol to synchronize independent clocks operating on separate nodes of a distributed measurement and control network to a high degree of accuracy and precision [47]. PTP is often used to synchronize distributed I/O devices over variable latency packet switched networks such as Ethernet. This protocol was originally released in 2002 defining many of the regulations and specifications, and in 2008 it was revised to its current version and released as IEEE 1588v2. The original version of this standard defined ordinary and boundary clocks while version 2 defines transparent and hybrid clocks; all of these clock types are addressed in this section [19].

In its simplest form, PTP was intended to be an administration-free protocol. The clocks within a network communicate with each other across a network media and establish multiple master-slave relationships. These master-slave relationships form what is known as a hierarchy of clocks [47]. The overall intent of PTP is that distributed I/O devices manage the synchronization of clocks automatically, thus requiring little if any network administrator input. Not all devices within a network require the same level of time synchronization accuracy; as a result, PTP allows for the support of a wide spectrum of clock accuracies to support the needs of the end device or process [48]. For example, when dealing with protective relays, accuracy within a millisecond is acceptable due to mechanical component tolerances. PTP can be configured to meet the needs of both of these applications. Additionally, PTP can use multiple sources of time as an ultimate time reference including, but not limited to, Global Positioning System (GPS), IRIG, Network Time Protocol (NTP), or another PTP clock.

A system or network of clocks by definition consist of one or more devices capable of becoming a master clock, while other devices within the network serve as slave clocks [47]. Normally one master clock is designated as the grandmaster clock. Figure 2.15 depicts the combinations of clocks that PTP supports: ordinary, boundary, transparent, and hybrid. Ordinary clocks consist of a single connection port, which industry commonly refers to as a PTP port. This port can either be assigned as a master or a slave [41]. Examples of ordinary clocks include GPS receivers and logic controllers and are typically located at the end nodes of a network [49]. Boundary clocks contain multiple PTP ports that establish separate PTP domains by segmenting the synchronization path between master and slave clocks. As their name implies, boundary clocks form boundaries between PTP synchronization segments [47]. These clocks are typically found in network switches. Transparent clocks are very different from ordinary and boundary clocks in that they help compensate for the propagation delay through the network rather than segment the network [41]. Finally, hybrid clocks are defined as a combination of PTP type clocks in a device. The most common type of hybrid clock is a transparent clock paired with an ordinary clock. Hybrid clocks are usually found within motion devices and are used to perform synchronized actuation [41].

It was previously stated that PTP was designed to be an administration-free protocol in that devices negotiate amongst themselves to determine a hierarchy of clocks. The algorithm used to determine the hierarchy of these clocks is known as the Best Master Clock Algorithm (BMCA). The BMCA, as defined by IEEE 1588, is the strict arbitration process employed to determine the status of each network node, either master or slave [47]. As its name implies, the

BMCA determines the best master clock and names it the grandmaster of the PTP system. All remaining clocks within the system are ultimately synchronized to the grandmaster.

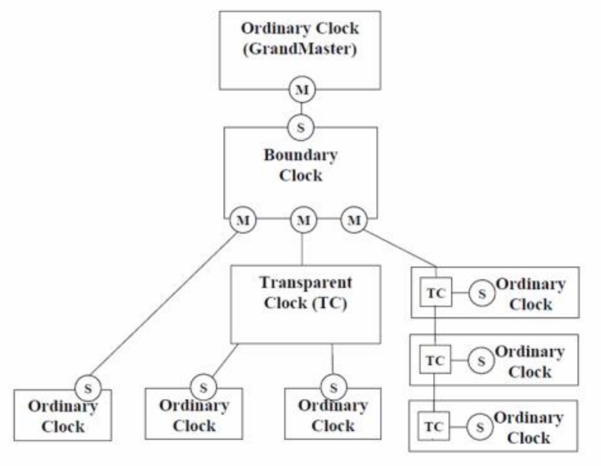


Figure 2.15 System of Clocks

Announce messages are sent approximately every two seconds for any PTP device claiming to be a master. These announce messages contain information about how well the clock attributes compare to a scale [40]. When a node receives an announce message from another device it compares the credentials that it receives to its own. The better of the two clocks the serves as the master, while the lesser acts as the slave. This process continues until the status for every clock within the network is determined.

The BMCA utilizes four criteria to determine the better of two clocks. These factors include clock class, accuracy, variance, and priority. Clock class defines the relative measure of clock quality. Accuracy defines how close the clock meters time to an absolute reference. Variance is the measure of the clock's stability. Priority is a manual override that can be established if a network administrator wants one clock to serve as grandmaster over another [47].

One main advantage to implementing PTP in a system is that it will dynamically update to topology changes [41]. For example, if the current grandmaster is removed from the system, the BMCA will attempt to designate a new grandmaster from the remaining clocks. On the other hand, if a clock with better credentials is added to the system, the BMCA will designate this new clock as grandmaster.

Figure 2.16 depicts the synchronization process for clocks utilizing the PTP protocol. PTP utilizes four messages in order to synchronize two clocks: sync message, follow up message, delay request message, and delay response message.

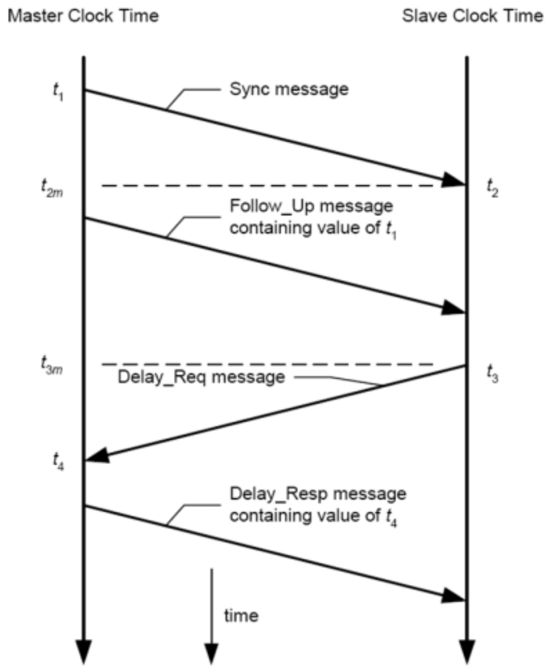
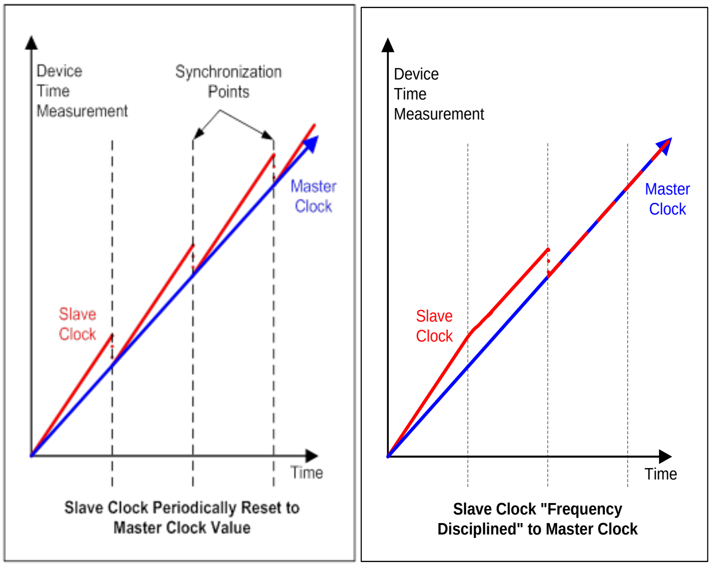


Figure 2.16 PTP Synchronization Messages

These messages are transmitted from master to slave and allow the clocks to make frequency adjustments to change the rate at which the clocks meter time. Additionally, the messages allow the clocks to measure the phase delay between master and slave and allow for a value correction. The frequency adjustment is made by utilizing the sync and follow up messages, while the phase adjustment is made by utilizing the delay request and delay response messages [41]. Every time the synchronization process occurs, timestamps t1-t4 are collected to determine the offset from master and frequency adjustments for the slave clocks. Clocks that are PTP compliant that make both a frequency and value adjustment are known as tunable clocks [41].

Tunable clocks are very important to use in any variable latency packet switched network. Take the example with two clocks, one master and one slave, and start them at the exact same time. No two clocks meter time at the exact same rate due to the natural frequencies of individual clock crystals. Consequently, the clocks will begin to diverge from each other and no longer remain synchronized.

One approach to correct this problem is to periodically reset the slave clock's value to that of the master's. This solution does not address the issue of differing clock frequencies or the rate at which the two clocks meter time. As a result, the clocks will only be brought into alignment for a moment in time, then begin to diverge again. Tunable clocks, on the other hand, allow for the proper synchronization of clocks. These clocks allow for the frequency of the slave clock to be tuned to that of its master so they will meter time at the same rate [41]. In addition to making a frequency adjustment, the tunable clock also applies a value or offset adjustment so that master and slave clocks are truly synchronized. Both synchronization phenomena can be seen in Figure 2.17.



Slave Clock Periodically Reset to Master Clock Value

Slave Clock "Frequency Disciplined" to Master Clock

Figure 2.17 Clock Synchronization Non-Tunable (Left) Tunable (Right)

It has been announced by the IEC that IEEE 1588 will be the protocol of choice to synchronize clocks over the IEC 61850 protocol. PTP is only a time synchronization protocol and must be implemented on a network that supports the protocol. IEEE 1588 was designed for packet switched networks such as Ethernet. For experiments described later in this paper, the EtherNet/IP (EtherNet/Industrial Protocol) was implemented across the network. EtherNet/IP is a protocol that is part of the Common Industrial Protocol or CIP suite.

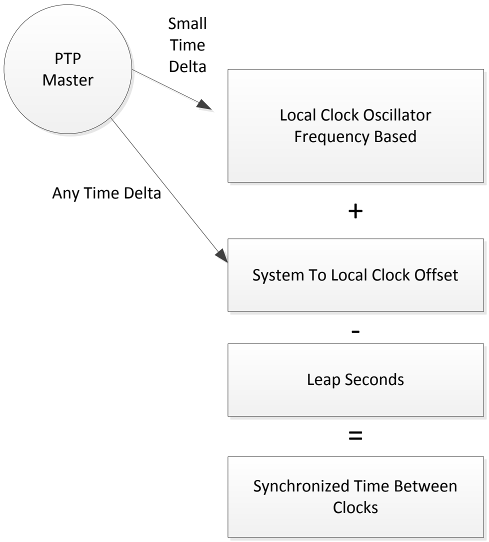
Common Industrial Protocol

The Common Industrial Protocol (CIP) is an industrial protocol suite that contains message and service instructions for automation applications pertaining to control, safety, synchronization, and motion [50]. This protocol is currently managed by the Open DeviceNet Vendors Association (ODVA) and is an open development network supported by hundreds of industrial vendors. CIP allows these applications to be implemented on enterprise-level Ethernet networks. Benefits of implementing CIP networks include seamless integration of I/O control and data collection, information flow across multiple networks, and implementable multilayer networks without the need for the implementation of network bridges [51].

CIP can be defined as an object-oriented connection based protocol that supports both explicit and implicit messaging. Explicit message connections provide generic, multipurpose communications paths between devices. These messages provide typical request/response type network communication [50]. Explicit messages are used by CIP for configurations, monitoring, and troubleshooting. Implicit messages, also known as I/O connections, provide special purpose communications paths between producing and consuming agents within a network. This I/O data is often exchanged cyclically or at a requested packet interval (RPI) [50].

The principle behind how EtherNet/IP synchronizes time across a network of distributed I/O can be attributed to the CIP Sync Object [40]. This object provides an interface to CIP Sync that allows devices, such as logic controllers, to access the synchronization mechanism. CIP Sync defines an offset clock model that addresses the requirements for various control applications. This model is necessary since PTP defines a mechanism for distributing and synchronizing time but fails to define a mechanism to compensate for step changes in time that may occur at the grandmaster source [41].

Figure 2.18 defines the CIP Sync Object at a high level. This model shows a PTP master represented by the circle and a PTP time slave represented by the top right side rectangle. In this example PTP is used to discipline a local clock so that it ticks and meters time at the same rate of the PTP master. The slave clock also maintains an offset between the local clock time and the PTP system time. Any small delta, or step change in time, causes the slave device to make a small adjustment to the "system to local clock offset" value [40]. In addition, the slave device will continue to tune its clock. A large step change results in the device updating the offset value, but not tune its clock. As a result, cyclic tasks such as SCADA reads, can be scheduled based upon the local clock and are not affected by large step changes in time at the grandmaster source.



PTP Master
Small Time Delta
Local Clock Oscillator Frequency Based
Any Time Delta
+
System To Local Clock Offset
-
Leap Seconds
=
Synchronized Time Between Clocks

Figure 2.18 CIP Sync Object Diagram

CIP Sync represents time as a 64-bit long-integer (LINT) that can be expressed in either nanoseconds or microseconds. The starting reference point in time for CIP Sync is January 1, 1970, starting at 12:00 AM. This time is represented in Universal Time Coordinated (UTC) and adjusted to include leap seconds. In order to represent this 64-bit LINT in a readable format, an algorithm must be developed to compute the current date and time in a readable, understandable format for analysis. The algorithm consisted of several mathematical operations to convert the LINT into two strings that are given in the following format: MM/DD/YYYY and HH:MM:SS:µsµsµsµs.

Acceptance of Precision Time Protocol

Recent work has shown that PTP can be successfully used in various applications. Precision time protocol was implemented successfully to measure the rotor angle of a synchronous machine [52, 53]. Additionally, programmable automation controllers have been used in conjunction with PTP and protective relays for a protection voting scheme [54]. PTP and programmable automation controllers were also used together as Sequence of Events Recorders

[55, 56]. Distributing time over Ethernet networks is becoming more common in SCADA system, and will play a crucial part in this dissertation.

Chapter 3 - Conceptual Design

Introduction

As previously described, the objective of this research is to specify and develop a system that links the protection and metering of the electrical distribution system at a mining operation to a site-wide control system. The first steps in this process are to establish the system capabilities, develop a preliminary conceptual design, and develop a working specification for the system to achieve the research objective. This chapter discusses the development of the conceptual design for the automation and control solution that was developed.

The design process of this solution began with an initial conceptual design based upon research, including industry needs and personal experience. In order to better fortify the design for an industrial process network, the original conceptual design was presented to industry stakeholders within the following disciplines:

1. Mining and Minerals
2. Metals
3. Oil and Gas
4. Engineering, Procurement, and Construction (EPC) Firms

The original conceptual design was then compared against engineering community input in order to finalize the conceptual design presented at the end of this chapter. This chapter begins with the original conceptual design followed by input provided by the engineering community segments listed above. Subsequently, the final conceptual design is presented and defined.

Original Conceptual Design

After researching the various SCADA protocols that existed for electrical distribution systems, as described in Chapter 2, it was determined that the IEC 61850 standard had a high probability of being readily accepted into the industrial marketplace. The challenge now was not to determine how to implement an IEC 61850 solution to control various intelligent electronic devices, such as relays in a mine distribution setting, but how to interface it to a platform that would be accepted by the engineering community.

Microprocessor Based or Automation Controller Based System

It was determined that developing a standalone microprocessor-based solution over PC software would not be viable or serve the community's needs. This conclusion was based on the fact that microprocessor-based solutions are difficult to maintain. Most microprocessor-based solutions are written in some form of high level structured text language, e.g., C++, C#, and Java. In addition, custom software for interfacing and database collecting would need to be developed for interfacing and storing data. Consequently, specific message instructions over TCP/IP would need to be developed in order to perform command and control of various intelligent electronic devices.

Operations, maintenance, and engineering staff at mining and other industrial facilities are not familiar with this type of system. Consequently, a microprocessor-based solution would be very costly to maintain as well as make changes to. Operations personnel typically work with ladder and function block programming in order to maintain, troubleshoot, and upgrade their systems. As a result, it was determined that the automation and control solution should interface directly with the facility's process control network. These networks are either controlled by a DCS or PLC/PAC type of controller, which are widely used in the industry. By designing a solution that interfaces to the electrical distribution system for these industrial facilities, the following advantages can immediately be seen:

1. Power monitoring capabilities
2. Predictive maintenance capabilities
3. Auto Transfer Switch (ATS) load shedding capabilities
4. Command and control of rotating machinery
5. Command and control of breakers
6. Interfacing energy measurements into speed regulator process models
7. Ventilation On Demand (VOD)
8. Fast motor bus transfer
9. Main-Tie-Main Distribution Schemes
10. Data Historian Applications
11. On Demand Topology Transitions

In Chassis Solution or Standalone Gateway Model

The next issue that had to be addressed was determining the physical form of the solution, in chassis or standalone solution. There are two practical approaches to address this question:

1. To develop a module that resides in the PLC/PAC chassis, or
2. To develop a module that stands on its own and serves as a gateway or conversion box.

The first option has several advantages, including that the module would require no additional network connections and would provide one less point of common coupling failure (PCCF). If the module resides within the chassis itself, the throughput of the module to convert IEC 61850 to a usable industrial format, such as EtherNet/IP, is much greater than if the module sits external to the local PAC rack. On the other hand, if the developed module is a standalone gateway module, it would be much more flexible with the system acceptance. For example, not every manufacturer creates the same rack-based PAC. Consequently, a module created for in-rack-only would work for a specific model solution of controller. Meanwhile, if the solution is a standalone module, multiple models of PAC and DCS systems can interface to the device. As a result, the original design for the solution would be a standalone hardware gateway that would be connected to an EtherNet/IP or equivalent industrial network. This solution provides more flexibility and is much more likely to be accepted by industry compared with a module that resides in the PLC/PAC chassis.

Quantity of Support

In order to have a solution that will be acceptable by the engineering community, it must be able to handle an acceptable number of intelligent electronic devices, (IEDs) on the network. After studying multiple drawings for both mining and metals applications, it was determined that 20 was an appropriate number of IEDs to support. Determining the maximum number of IED support for each gateway module is necessary to permit multithreading on the hardware solution side and to establish processor clock requirements. If the application required more than 20 IEDs, a second gateway module could be added to accommodate the additional devices. This process could be completed until all devices could be accounted for within the distribution system.

Communication Media

In order for the solution to be successful, a suitable communication media must be chosen to move information from the gateway module to both the IED and process networks. The following modes of communication were examined for this solution:

1. Copper 10/100/1000 Mbit/s communications
2. Fiber Single/Multimode
3. Wireless b/g/n communications

Copper media is the most common form of Ethernet based communication in North America. Traditional 100 Mbit/s communications speed would be more than adequate to move measurands and MMS messages throughout a large network of IEDs connected to the process network. The disadvantage to this solution is that it is susceptible to electromagnetic interference (EMI). The IEC 61850 standard defines various EMI specifications for devices on the network with corresponding noise immunity requirements. According to the standard, copper communications would not satisfy IEC 61850-10 for EMI if the module were within the substation environment. As a result, if the solution were to include copper communications, the gateway would have to be located in a control room in order to satisfy IEC 61850-10 for EMI.

Fiber communications would eliminate the EMI issue that a copper solution incurs, but the disadvantage of this solution is the cost of fiber. Most automation and control networks are traditionally copper, so adding a fiber interface may not be cost effective.

Wireless communications, while seemingly very attractive, would not be a viable choice for the industry. With rising security issues and cyber-attacks at an all-time high, process engineers are focused on maintaining network and system health. Wireless communications are more vulnerable to attack than other solutions. In addition, wireless signals typically do not travel well in most underground settings without the use of many repeaters and amplifiers. Considering the advantages and disadvantages of copper, fiber, and wireless communications, it was determined to select traditional copper, 10/100 Mbit/s communications for this application.

System Security

Security is a major consideration in the design and operation of industrial control systems. Good security practices help reduce control product and system susceptibility to accidental or unauthorized activities that affect safety, operational integrity, and data confidentiality. Industrial control system security relies on layers of security using multiple controls, methods and techniques that work together to protect a system's assets, operations, and those who depend on its safe, reliable operation. Technical controls, including physical and electronic mechanisms that compensate for risk, should be accompanied and balanced by non-technical controls such as company policies, procedures and guidelines. To help protect key assets, users should employ specific product-level security and protection features available within a networked process system.

Cyber-attacks are at an all-time high [11]. With the use of intelligent devices, such as managed switches, control engineers now have the ability to segment their network into zones, thus mitigating cyber threats. Today, many Information Technology (IT) personnel advocate constructing network infrastructures that are similar to the one shown in Figure 3.1. This model is based upon ISA Standard 95 and the Purdue network model. The model defines levels down the left-hand side of the figure and zones down the right. The model is separated into six different levels and four zones. In order to keep a secure facility, managed switches and other network equipment, such as firewalls, separate the plant network (levels 0-3) from the enterprise network (levels 4-5). These two networks are isolated via a firewall which does not permit web traffic, email, or process automation packets to pass directly through without passing through an approved proxy. By segmenting these networks, the process owner can limit traffic and authorized users who have virtual access to the plant floor production.

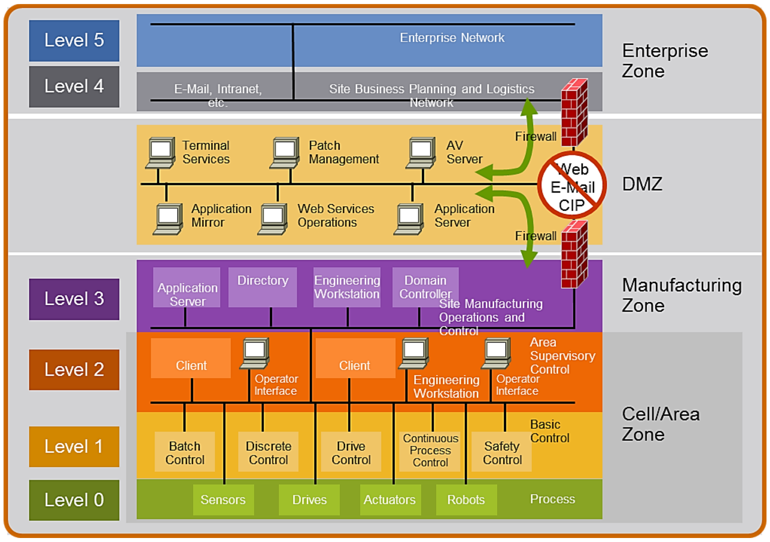


Figure 3.1 Network Security Model

The gateway module resides in Level 1, since it is part of the control solution for power distribution systems. This solution would be accessed by Level 2 equipment, such as engineering work stations (EWSs) and operator work stations (OWSs) for command and control of various process components. System security would be maintained through best practices and system architecture design by (1) keeping process control and instrumentation at the lower levels of control, and (2) keeping enterprise control at the top of the network architecture.

Summary of Original Concept

Figure 3.2 depicts the original conceptual design for the solution to integrate the electrical distribution network to the process control network of a mining and metals facility.

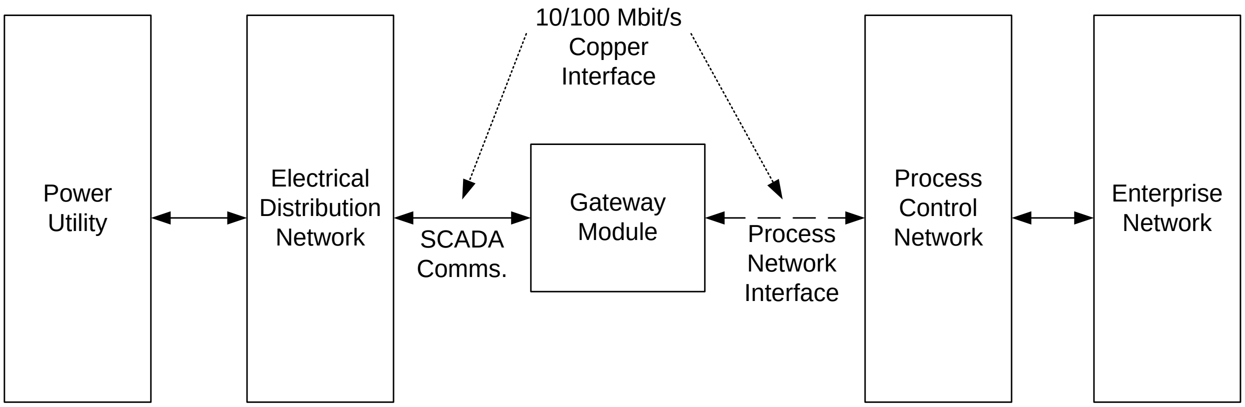


Figure 3.2 Original Conceptual Design

As described, it was determined that the module should be integrated into the process control system as a stand-alone, microprocessor-based solution that interfaces to a programmable automation controller. This solution provides more flexibility with integration into more existing process control systems compared with a module that resides in the PLC/PAC chassis. This solution bridges the electrical distribution SCADA network with the process control network and provides access from the enterprise network to the electrical distribution system as well as power utility access, if granted, to the process control network. With the advent of smart grid technology, this solution would provide bidirectional communications between a process owner and the power utility, a major goal of smart grid technology.

The next step the engineering process was to take this conceptual design to various personnel in the engineering community for additional input/feedback. The next portion of this chapter discusses how this input was obtained as well as a summary of that input.

Engineering Community Input

After developing an original conceptual design for a SCADA interfacing solution to an IED network, the concept was presented to the engineering community for input and feedback. As mentioned, various industries, including mining, metals, oil and gas, and EPC firms were surveyed for feedback. Firm names were removed to provide anonymous feedback. Since this solution bridges both the power and process control systems, survey questions were directed to both power and instrumentation and control (I&C) engineers. When security concerns were being addressed, these engineers were asked to consult with their IT Security office. As a result, both I&C and power engineers needed to work together to provide a solution that would serve as an acceptable compromise between them.

Each company was asked for their opinion and input on the following categories:

- Packaging
- Communications Media
- Number of IEDs to Support
- Safety Concerns
- Security Concerns
- User Interface Information
- Any Additional Comments or Concerns

The results of the industry feedback are summarized in the following four tables.

Mining

Table 3.1Mining Firms

Mining Firms							
Company	**Packaging**	**Communications Media**	**Device Support**	**Safety**	**Security**	**Information Requested**	**Additional Comments**
Foxtrot	Stand-Alone	Copper	Min 20	Interlocking , MSHA Safety Standards	Min 5 levels of security for command and control	Alarms and Target Trips	ISA 5.5 Compliance
Gulf	Stand-Alone	Copper Min 100 Mbit/s	Min 20	Classified Areas MSHA	Min 10 levels	Command and Control	ISA 5.5 Compliance
Hotel	Stand-Alone	Copper	Min 20	Interlocking	Min 7 levels	Alarms, Command and Control	EMI Conformance
India	Stand-Alone	Copper Min 100 Mbit/s	30+	Select Before Operate	Min 10 levels	Command and Control	Diagnostics should be included with module
Juliet	Stand Alone	Copper Min 100 Mbit/s	Min 10	Interlocking	Min7 levels	Alarms, Command and Control	Historian Interface Capability

Metals

Table 3.2 Metals Firms

Metals Firms							
Company	**Packaging**	**Communications Media**	**Device Support**	**Safety**	**Security**	**Information Requested**	**Additional Comments**
Kilo	Integrated	Fiber	Min 20	Interlocking , Select Before Operate	Min 7 levels	Command and Control, Visualization	ISA 5.5 Compliance
Lima	Integrated	Copper Min 100 Mbit/s	Min 20	Classified Areas	Min 7 levels	Command and Control, Alarms, Visualization	ISA 5.5 Compliance
Mike	Stand-Alone	Copper	Min 20	Interrogating I/O	Min 7 levels	Command and Control, Visualization	ISA 5.5 Compliance
November	Stand-Alone	Fiber	Min 20	Interrogating I/O	Min 10 levels	Command and Control, Historian	ISA 5.5 Compliance
Oscar	Stand Alone	Fiber	Min 20	Classified Areas	Min 10 levels	Alarms, Command and Control	ISA 5.5 Compliance

Oil and Gas

Table 3.3 Oil and Gas Firms

Oil and Gas Firms							
Company	**Packaging**	**Communications Media**	**Device Support**	**Safety**	**Security**	**Information Requested**	**Additional Comments**
Alpha	Stand-Alone	Fiber	Min 10	Interlocking between physical devices and operator interface	Min 5 levels of security for command and control	Alarms, Timing SOE, Visualization, Historian, Command and Control	Will use for load shedding and power monitoring
Bravo	Stand-Alone	Fiber	Min 20	Interlocking	Min 5 levels of security for command and control	Alarms, Command and Control	Must be designed for Class 1 Div 2 systems
Charlie	Stand-Alone	Copper Min 100 Mbit/s	Min 10	Interlocking	Min 7 levels	Alarms, Command and Control	Extended Temperature Range +70 C
Delta	Stand-Alone	Copper Min 100 Mbit/s	Min 10	Interlocking	Min 10 levels	Alarms, Historian	Predictive Maintenance Models can now incorporate electrical measurements
Echo	Integrated	Copper Min 100 Mbit/s	Min 10	Interlocking	Min 5 levels	Alarms, Visualization, Command and Control	Graphics must meet ISA 5.5 Standard

EPC Firms

Table 3.4 Engineering Procurement Construction Firms

EPC Firms							
Company	**Packaging**	**Communications Media**	**Device Support**	**Safety**	**Security**	**Information Requested**	**Additional Comments**
Papa	Stand-Alone	Fiber	Min 20	Interlocking , Select Before Operate	Min 7 levels	Alarms, Timing / SOE, Visualization	ISA 5.5 Compliance, Extended Temperature Range
Quebec	Stand-Alone	Copper Min 100 Mbit/s	Min 20	Classified Areas	Min 5 levels	Alarms, Targets, Visualization, Historian	ISA 5.5 Compliance, DFS Security
Romeo	Stand-Alone	Copper	Min 20	Interrogating I/O, Classified Areas, Select Before Operate	Min 5 levels	Command and Control, Visualization	ISA 5.5 Compliance, Diagnostics Web Server
Sierra	Stand-Alone	Copper	Min 20	Interrogating I/O	Min 5 levels	Historian, Command and Control	ISA 5.5 Compliance, New Look to Load Shedding
Tango	Stand Alone	Copper	Min 20	Classified Areas, Interlocking	Min 5 levels	Alarms, Command and Control	ISA 5.5 Compliance, Extended Temperature Range

Final Conceptual Design

After collecting industry input and comparing it to the original design, the final conceptual design was specified and is presented in the table below. It can be seen than 17 of the 20 industry participants felt a stand-alone gateway would provide the best solution. As most industries already had a copper Ethernet infrastructure installed, 70% of all companies surveyed felt that developing an already existing copper infrastructure was more practical. Companies that felt fiber communications were more beneficial cited EMI concerns for this opinion.

All of the industry segments surveyed felt that a minimum of 10 IEDs would need to be supported by the solution. With regard to safety companies surveyed felt that the ability to interlock between HMI screens and IED was needed to prevent unauthorized operation and mis-operation of electrical devices in an industrial setting. Company feedback also stated that security played a large role in process control. Feedback stated that HMI screens should have different levels of operation and privileges based on role. They ultimately specified a minimum of seven different levels of security settings.

Additionally, all industry segments surveyed agreed with the initial conceptual design that alarms, targets, and measurands should be displayed. Also from the industry input, it was determined that this solution should provide sequence of events reporting as well as a way to interface to database software such as PI. Finally, it was determined from industry input that all graphics must meet the current ISA standards for process control graphics. The final conceptual design specifications can be seen in Table 3.5. Figure 3.3 depicts the final conceptual design for this solution.

Table 3.5 Final Conceptual Design

Final Conceptual Design		
Parameter	**Original Design**	**Final Design**
Packaging	Stand-Alone	Stand-Alone
Communications Media	Copper Min. 100 Mbit/s	Copper Min. 100 Mbit/s
Device Support	Min. 20 device	20 devices
Safety	Interlocking	Interlocking, Select Before Operate, Classified Areas Considerations
Security	Min. 5 levels	Min. 7 levels
Display Information	Alarms, Targets, Measurands	Alarms, Targets, Measurands, Timing / SOE, Historian Interfacing Capability
Visualization Information	N/A	ISA 5.5 Compliance, Provide Diagnostic Information, Command and Control

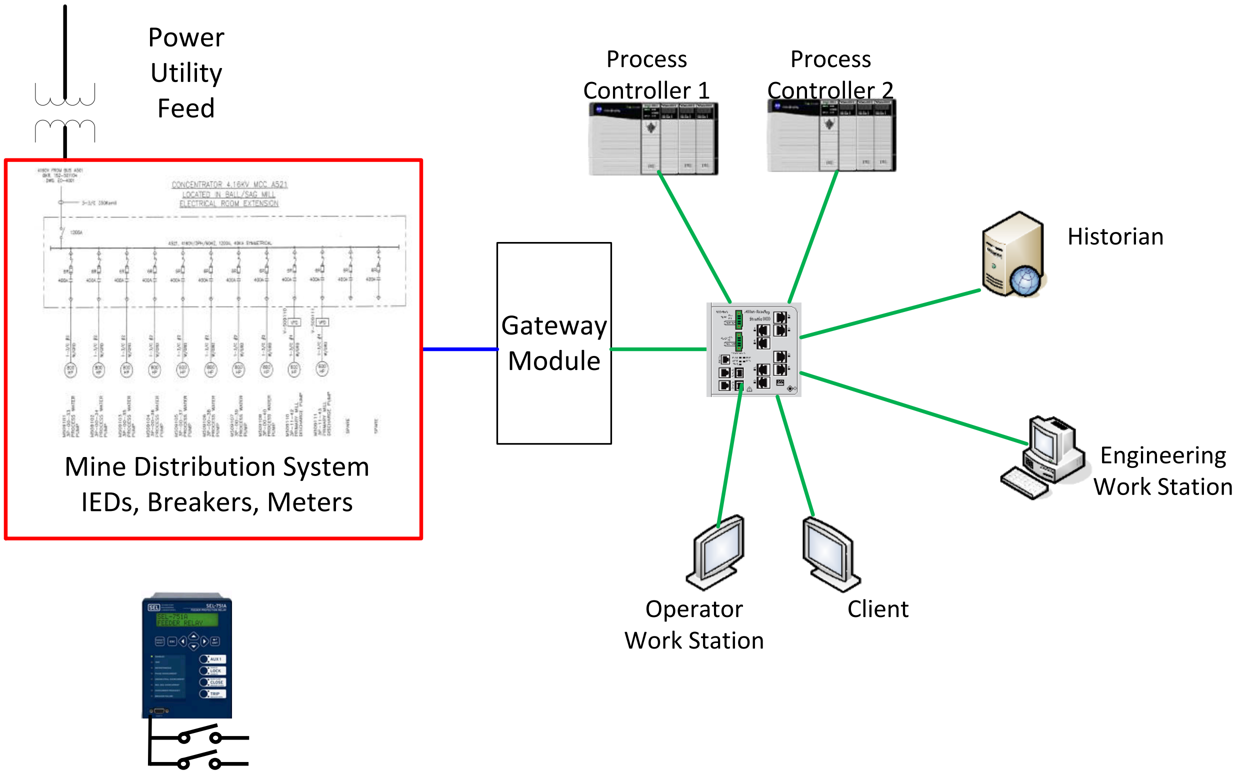


Figure 3.3 Final Conceptual Design

Conclusion

This chapter discussed the evolution of the conceptual design for an automation and control solution that implements electrical SCADA systems into mine system automation schemes. The previous chapter presented various SCADA standards and defined their benefits and shortcomings in both generic comparisons and benefits to mining protection and automation systems. The literature review of this document has shown that, with respect to interoperability between vendors and a common naming convention, the IEC 61850 standard is the best automation and control solution to interact with current systems. It was also discussed that the IEC 61850 standard possessed the following shortcomings that would need to be addressed by this solution:

1. Although the substation configuration language (SCL) file defines how an IED communicates on a 61850 network, it does not define configuration information for protection and control functions of each IED

2. Each IED manufacturer has its own proprietary software and configuration tools to enable and configure various protection elements and control strategies. The IEC 61850 standard defines no methodology for designing communication-based assisted automation.

In order to develop a successful solution for the mining market, these two concerns need to be addressed within the conceptual design of the total solution, i.e. hardware, software, and visualization. The next three chapters of this dissertation discuss the components and algorithms used to take the final specification and implement a prototype.

Chapter 4 - Hardware Implementation

Introduction

As mentioned in Chapter 3, the hardware solution developed to link the electric distribution and process control systems is a standalone hardware gateway. This chapter defines the logic behind the gateway operation and how it bridges the IEC 61850 and process control network. The gateway module consists of a main processor with various peripherals including a network 10/100 Mbit/s copper interface and Secure Disk (SD) interface to store configurations for the module as shown in Figure 4.1. It is noted here that there were several design iterations before deciding on the components described in this chapter.

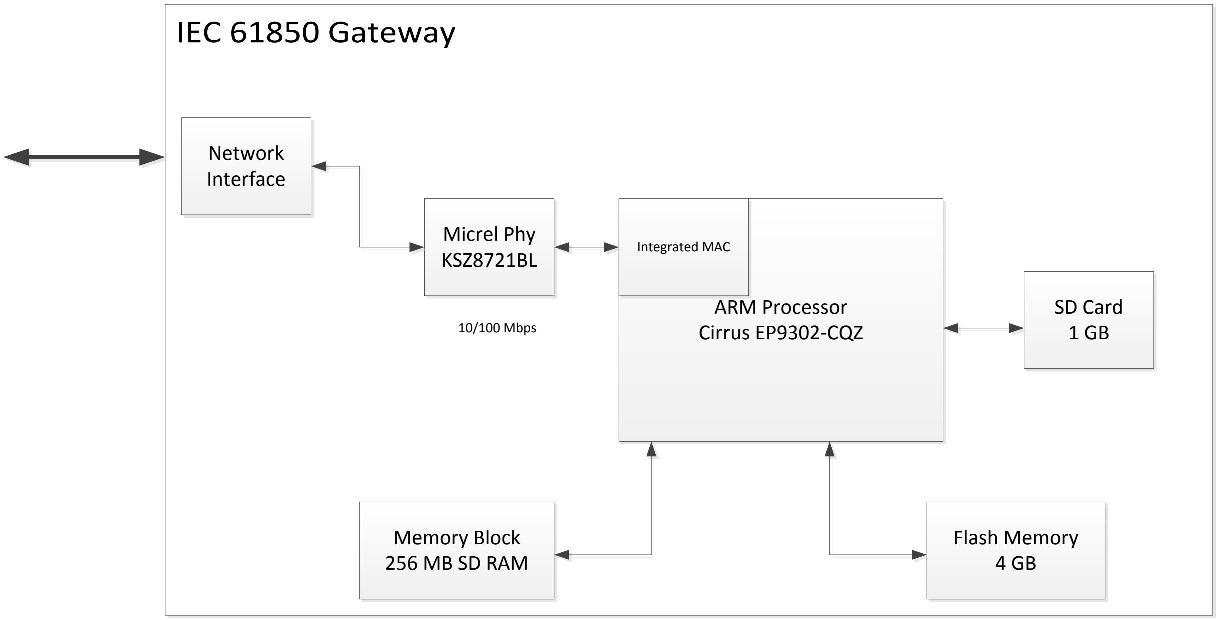


Figure 4.1 High Level Gateway Block Diagram

Figure 4.1 depicts a high-level functional block diagram defining the hardware components. The most important component in this hardware configuration is the ARM processor, shown by the largest block in the diagram. The processor, a single core 200 MHz, ARM-9 processor, is responsible for all interfaces to memory (flash or SD RAM), as well as the hardware level interface to the Micrel Phy for Ethernet based communications over a CAT-5 copper network. The processor additionally interfaces to the SD card, where configurations are stored for various modes of operation. In addition, the processor provides a hardware level interface to LEDs for both diagnostic and status indications during various modes of operation.

Three hardware blocks were considered for system communications: Network Interface, Micrel Phy, and Integrated MAC, all depicted in the top left of the Figure 4.1. Figure 4.2 describes the data link path between physical media and the ARM-9 processor via the Phy. The network interface consists of an RJ-45 receptacle to which a generic CAT-5 Ethernet cable can be inserted to provide a link to the process control network switch, as seen on the right side of Figure 4.2.

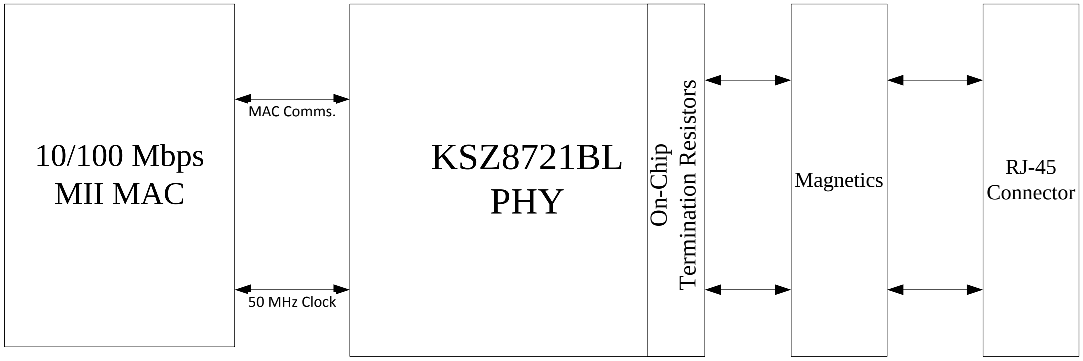


Figure 4.2 High Level PHY Block Diagram

The magnetics block is a passive EMI filter which reduces electro-magnetic noise on the wire. Additionally, an electro-static discharge (ESD) filter is built into the passive magnetic circuit. The on-chip termination resistors were chosen to provide the highest level of signal integrity, while further reducing EMI noise. The physical layer interface (PHY) is referenced by the large block labeled KSZ8721BL. This transceiver provides the hardware level connections between the processor MAC and physical media.

Figure 4.3 shows a high level block diagram of the physical layer interface chosen for this solution. The benefit of this interface is that all low-level hardware drivers were already implanted and functional. It can be seen from the block diagram level that the PHY has many responsibilities and peripheral interfaces, including the auto-negotiation of network transfer speed and the management of both a 10Mbit and 100Mbit connection. This choice of PHY substantially reduced engineering time without having to code these low level communications drivers.

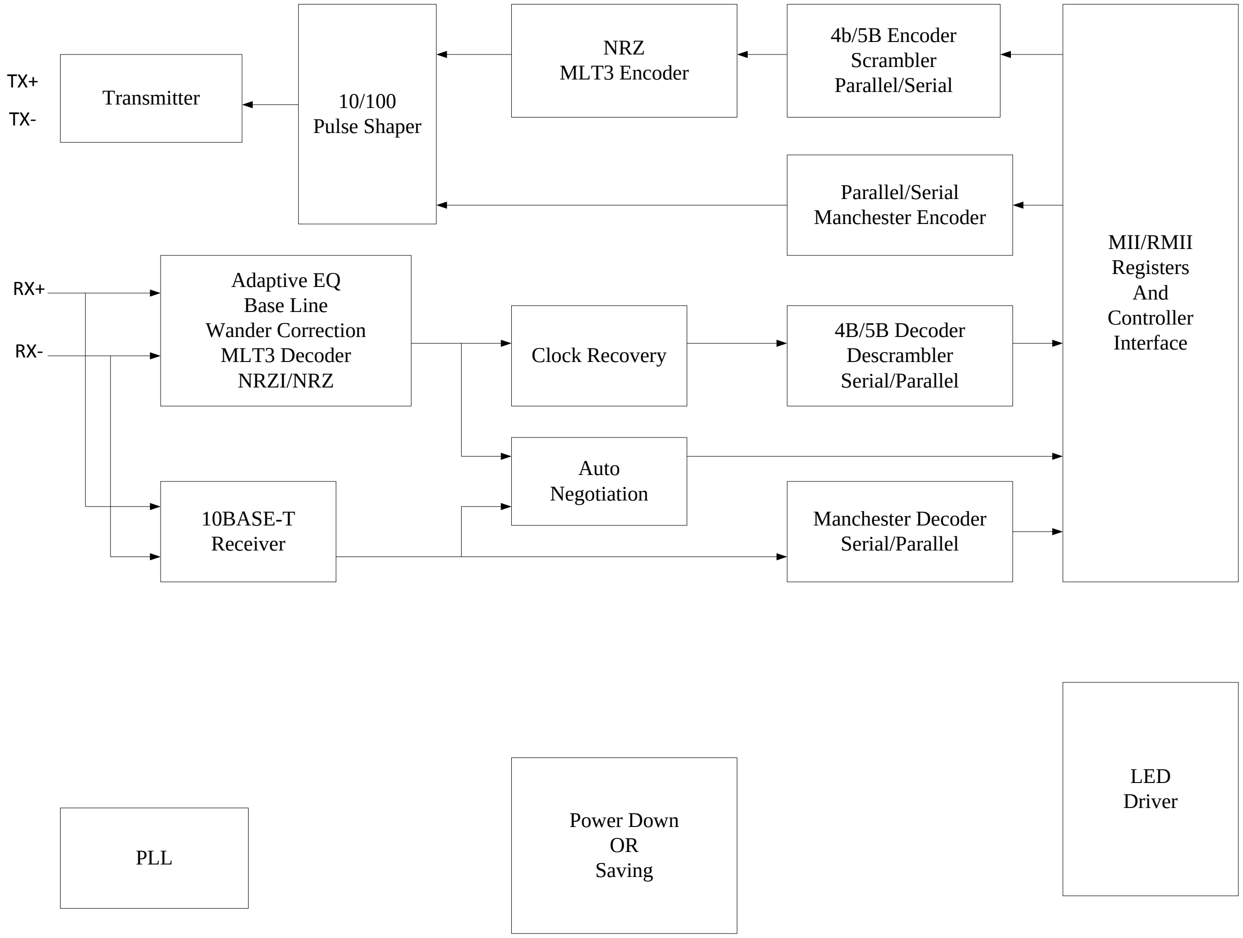


Figure 4.3 PHY Block Diagram

In addition to holding the Linux operating system for this solution, the flash memory also holds the firmware for the module as well as the web server for diagnostics and configuration. The SD RAM is used to allocate and store memory for every client connection and IED, as well as server socket connections and PAC. This memory is volatile, thus any data is lost on power cycle. The SD card is responsible for storing the device configuration files. These files are stored in XML format and read on startup, after which memory is then allocated within SD RAM. Because Linux is chosen as the operating system, the SD Card peripheral is mounted as a physical drive. Figure 4.4 depicts the startup sequence and memory allocation of the device upon power-up.

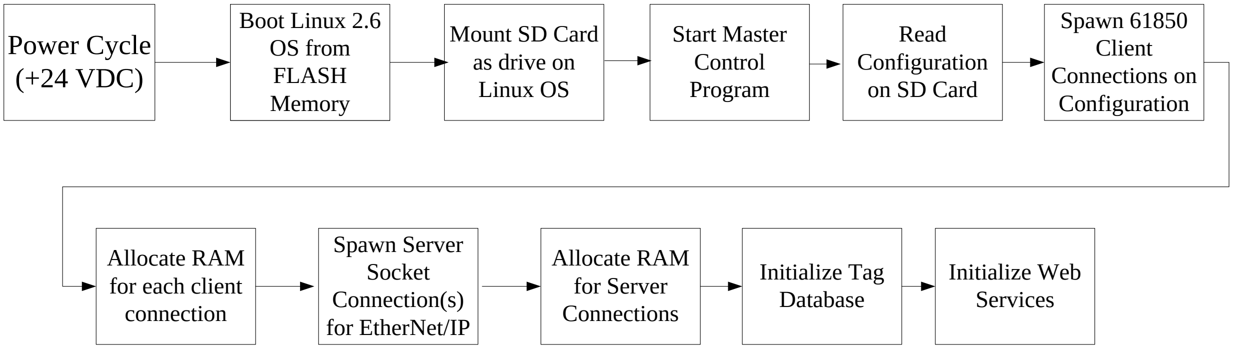

Power Cycle (+24 VDC)
Boot Linux 2.6 OS from FLASH Memory
Mount SD Card as drive on Linux OS
Start Master Control Program
Read Configuration on SD Card
Spawn 61850 Client Connections on Configuration
Allocate RAM for each client connection
Spawn Server Socket Connection(s) for EtherNet/IP
Allocate RAM for Server Connections
Initialize Tag Database
Initialize Web Services

Figure 4.4 Power Cycle State Machine Sequence

The code for this solution was written in C++, in the windows environment. The code was then compiled using the *gcc* tool to create a target file intended for the Linux 2.6 environment.

The remaining portion of this chapter focuses on the multi-threading of the processor and interaction among IEDs, the gateway module, and the controller.

Solution Overview

The process of collecting data from various intelligent electronic devices (IEDs) and converting it to an industry accepted process control protocol was performed through the concept of multi-threading. Figure 4.5 depicts the various threads that are executing at any given time during the operation of the hardware gateway.

Master Control Program (MCP)

Figure 4.5 shows six various types of threads that run parallel with communications to the processor and each other. The processor contains the master control program (MCP). On module startup, the MCP reads the configuration file from the SD card, as depicted in Figure 4.1. Based on the configuration file, the MCP determines the number of individual threads needed to be created for each driver and calls the corresponding tag database files to create and allocate memory for moving parameters.

After startup, the MCP starts each driver individually. After each threaded drive has returned a *ready* message to the MCP, the MCP changes the status of each individual (ready) driver to *run* mode. The MCP also serves as a watchdog timer running on Linux that evaluates the time to execute each running thread. If a thread gets halted or exceeds five milliseconds to execute, the MCP terminates the thread and re-instantiates the terminated thread. An example of the MCP can be seen in Figure 4.6.

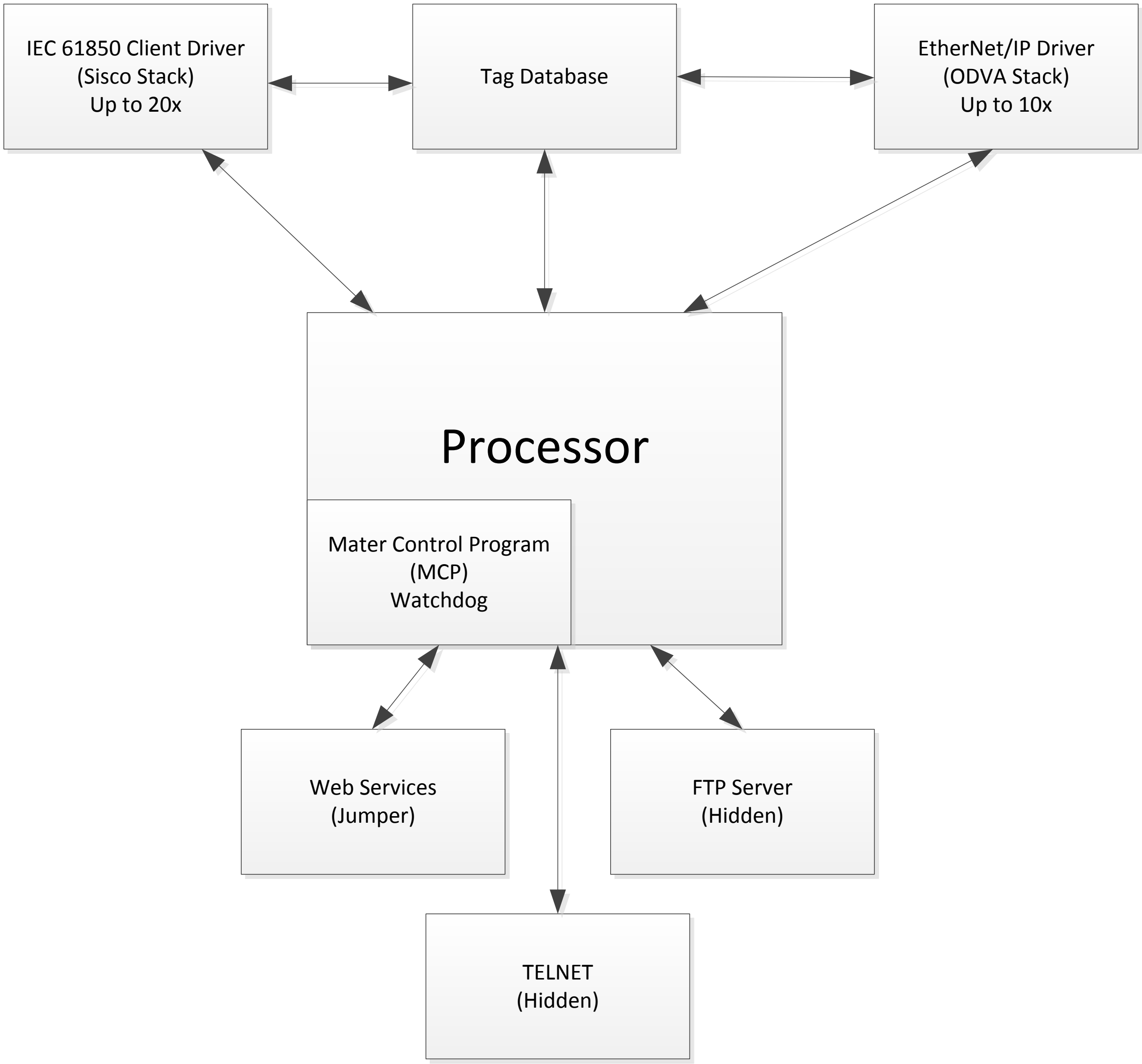


Figure 4.5 Processor Thread Diagram

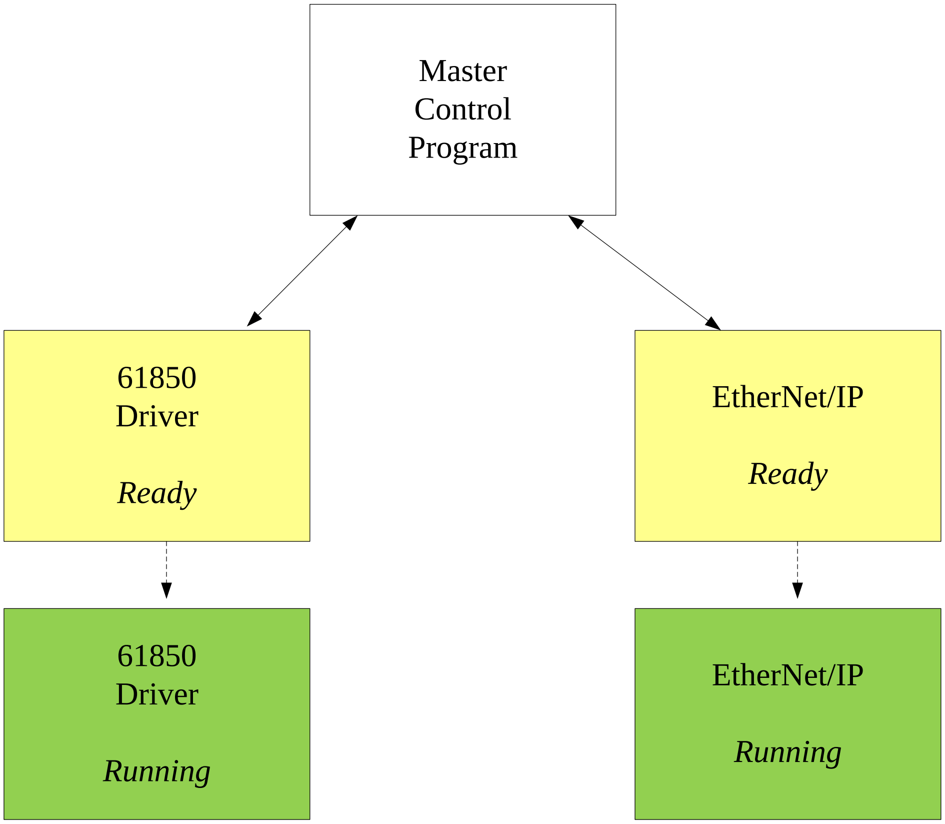


Figure 4.6 MCP Example

Should an error or fault occur during operation, the MCP will log this error in a log file stored in the flash memory, as depicted in Figure 4.1. This file can then be retrieved for debugging purposes by the configuration software that will be discussed in Chapter 5.

Hidden Threads

The telnet and ftp server threads are hidden from public use and are used exclusively for debugging and pushing new system files to the processor. The web services thread hosts a simple website whose web address is the IP address of the module. Information displayed on this web page includes model and firmware revisions, as well as simple diagnostic information about the module, such as error and fault status.

In order to enhance performance of the communication processes of the module, the web services thread is tied to a jumper that can be removed to enable/disable this functionality. By removing the web services jumper, the module, on power up, ignores the installation of the web services thread, thus allowing more processing power to be dedicated to IEC 61850 and EtherNet/IP communications. Additionally, module firmware is downloaded to the module via the web services page.

Data Mapping

When the module is configured from the user interface software, report configurations containing the parameters that are to be passed from the IED to the automation controller are stored on the SD card. Additionally, when the user maps the IEC 61850 tags to EtherNet/IP tags, each 61850 tag is allocated a certain number of bytes depending upon the data type, such as boolean, real, double integer, etc. This configuration is also defined on the SD card.
Table 4.1 lists the data types that were supported for this research.

Table 4.1 Data Types

Data Type	Definition	Bits
BOOL	Boolean	1
BYTE	Byte	8
UBYTE	Unsigned Byte	8
INT	Integer	16
UINT	Unsigned Integer	16
DINT	Double Integer	32
UDINT	Unsigned Double Integer	32
REAL	IEEE 754 Single Precision Floating Point	32
DREAL	IEEE 754 Double Precision Floating Point	64
STRING	ASCII Character Array	32 + 8*length of text
DATETIME	UTC microsecond precision date and time	64
SEMAPHORE	Internal use semaphore	128

Table 4.2 gives a description of various data types supported by this research as well as their corresponding range of values.

Table 4.2 Common Data Types Definitions

Type	Description	Range
BOOL	Boolean data type	0 or 1
BYTE	Signed 8 bit magnitude	-128 to 127
UBYTE	Unsigned 8 bit magnitude	0 to 255
INT	Signed 16 bit magnitude	-32768 to 32767
UINT	Unsigned 16 bit magnitude	0 to 65535
DINT	Signed 32 bit magnitude	-2147483648 to 2147483647
UDINT	Unsigned 32 bit magnitude	0 to 4294967295
REAL	Signed IEEE 754 32 bit floating point number	$\pm 10^{-44.85}$ to $\pm 10^{38.53}$
DREAL	Signed IEEE 754 64 bit floating point number	$\pm 10^{-323.3}$ to $\pm 10^{308.3}$

The STRING Data Type is the ASCII representation of a string of characters. The STRING Data Type is composed of two fields shown in Table 4.3. The LENGTH field contains the length of the string and DATA contains the ASCII characters. The size of the DATA array is defined at run time by the configuration settings.

Table 4.3 Uncommon Data Types Definition

Field	Data Type
Length	UINT
Data	UBYTE[]

A DATETIME data type represents a Coordinated Universal Time (UTC) date and time. The time is expressed as seconds and microseconds since the Epoch and is stored as a 64-bit structure. The epoch used is the standard UNIX epoch that corresponds to 00:00:00 UTC, January 1 1970. The UTC time is stored with a one-microsecond resolution. A DATETIME data type is composed by two fields, as shown in Table 4.4.

Table 4.4 Date/Time Definition

Field	Data Type
Seconds	UINT
Microseconds	UINT

Mapping Example

The following example was created to illustrate how data, Alpha-Hotel, is packaged and mapped in the gateway module. The data is shown in Table 4.5 and graphically mapped in Table 4.6.

Table 4.5 Data Mapping Example

Field Name	Data Type	Byte:Bit	Size (Bits)	Data Map Color
Alpha (A)	Bool	0:0	1	🟦
Bravo (B)	Bool	0:1	1	🟩
Charlie (C)	Byte	1:0	8	🟨
Delta (D)	Dint	4:0	32	🟥
Echo (E)	Byte	8:0	8	🟩
Foxtrot (F)	Int	10:0	16	🟦
Gulf (G)	Byte[3]	12:0	24	🟪
Hotel (H)	Byte	15:0	8	🟥

Table 4.6 Data Mapping Example Graphical

Bit																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Byte																															
3								2								1								0							
																🟨	🟨	🟨	C	🟨	🟨	🟨	🟨							B	A
🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	D	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
🟦	🟦	🟦	🟦	🟦	🟦	🟦	F	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦									🟩	🟩	🟩	🟩	E	🟩	🟩	🟩
🟥	🟥	🟥	H	🟥	🟥	🟥	🟥	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	G	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪

SISCO Stack for IEC 61850 Driver

When the module is booted, the SD card information is loaded into the processor and SD RAM for use. Every IED that is defined by the user in the reports has its own thread on the IEC 61850 client driver, which is based upon the SISCO stack. At any given time up to 20 threads or physical devices can be interrogated by the gateway module. These threads read the configuration of the report and request this information from the IEDs. After the information in each report is populated, the information is transferred to the tag database where it is collected by the EtherNet/IP driver.

EtherNet/IP Stack

As previously mentioned, when the user maps IEC 61850 tags to EtherNet/IP tags, a certain number of bytes are allocated to each parameter or tag. The data is transferred from the gateway to the automation controller via a Class 1 Common Industrial Protocol (CIP) message. CIP messages are limited to 500 bytes of input data, 496 bytes of output data, and a configuration header. The information in the tag database is grouped into chunks of 500-byte packets or data table reads. A parallel thread is created for every group of 500 bytes that needs to be sent to the controller. This thread is constructed using the Open Device Vendors Association (ODVA) stack for EtherNet/IP. This organization manages the configuration for the drivers for the CIP protocols.

Tag Database

The tag database is a shared piece of memory where data items can be located using a tag name. The tag database is composed of two shared memory locations, as shown in Figure 4.7.

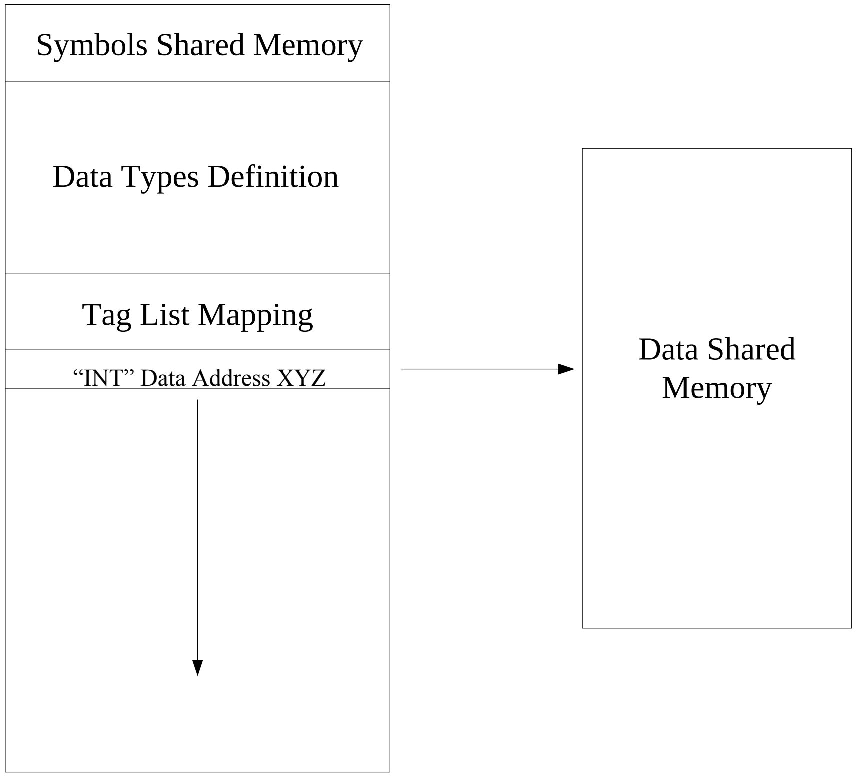


Figure 4.7 Generic Tag Database

The tag database is the central repository of process data and is the link between drivers. The data is read by a driver and copied to the tag database where another driver can write it to a device. This concept is shown in Figure 4.8.

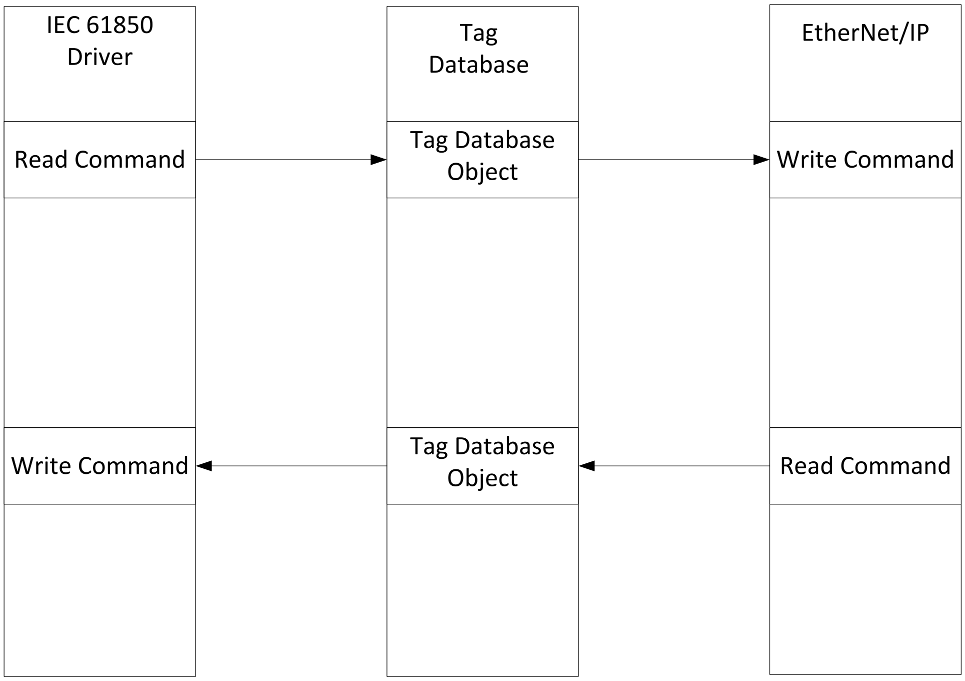


Figure 4.8 Interaction Between Drivers

The tag database thread can be thought of as a large array of data. The array begins at index zero and ends with the index of the last byte of data that needs to be transferred. This thread is the key to the operational success of the gateway module. The tag database is populated by the IEC 61850 thread, and information is read from the tag database by the EtherNet/IP thread to be sent to the process control system. Semaphore tags are used to signal drivers that information has changed and a new, read/write cycle needs to be executed.

When the 61850 thread writes to the tag database, it writes to an index offset from the base, i.e., the beginning location of where the parameter or electrical measurand lies within the array of data to be captured. The same process is performed when the parameter is to be read by EtherNet/IP driver. The process of writing to the tag database can be seen in Figures 4.9 and 4.10, respectively, for both the IEC 61850 and EtherNet/IP drivers. Figure 4.11 depicts the interactions of the two drivers with the tag database. As described in this chapter, the IEC 61850 thread, shown in Figure 4.9, is instantiated upon power-up of the module. After the SD card configuration is read by the module, the corresponding number of client connections (IED threads/drivers) is instantiated. The IEC 61850 driver waits in an endless loop until the write command is issued by the tag database for information to be passed from one driver to the next. Also, on startup, the server-socket connection is created to transfer information from the tag database to the controller via the EtherNet/IP driver. The EtherNet/IP driver, as shown in Figure 4.10, waits for a semaphore tag to update. This semaphore tag can be thought of as a collection of boolean flags corresponding to various measurands and data that are designed to be passed from the IED network to the process network. When the semaphore tag is updated, an *interrupt* is triggered that allows the EtherNet/IP driver to read the tag database, construct data tables of information to be sent to the controller, and issue a write command to the controller signaling that it is ready to pass data from the gateway to the controller processor.

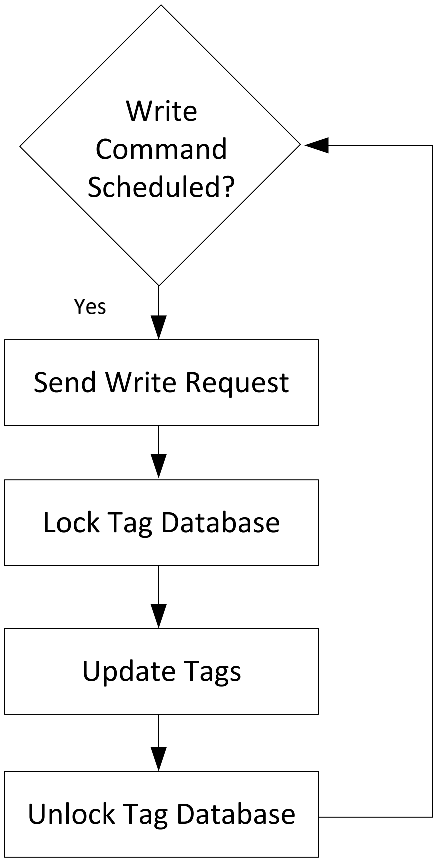


Figure 4.9 IEC 61850 Driver Information Flow

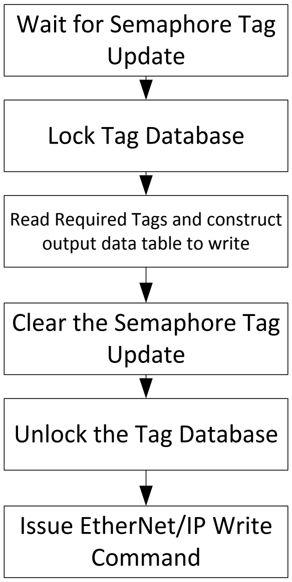

Wait for Semaphore Tag Update
Lock Tag Database
Read Required Tags and construct output data table to write
Clear the Semaphore Tag Update
Unlock the Tag Database
Issue EtherNet/IP Write Command

Figure 4.10 EtherNet/IP Driver Information Flow

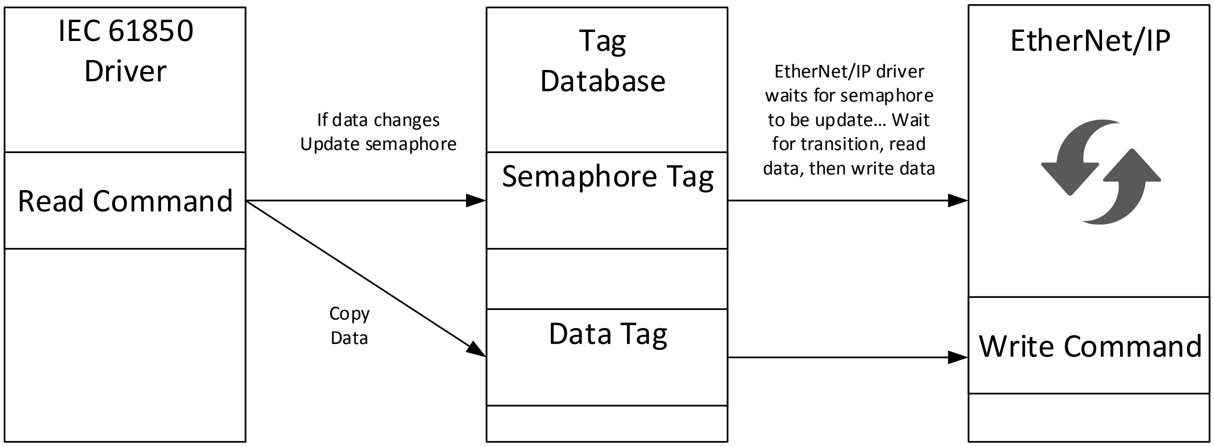

IEC 61850 Driver
Read Command
If data changes Update semaphore
Copy Data
Tag Database
Semaphore Tag
Data Tag
EtherNet/IP driver waits for semaphore to be update... Wait for transition, read data, then write data
EtherNet/IP
Write Command

Figure 4.11 Functional Flow of Driver Traffic

Illustrative Example

An example of how this technology could be applied in a mining application is through the one-line diagram of a simplified underground mine power system, as shown in Figure 4.12. This figure shows the substation, which reduces the utility voltage to mine distribution levels, the distribution cables, a switch house for allowing the system to branch, the load centers, which reduce the distribution voltage to utilization levels, and trailing cables that connect the loads to the power center.

Ordinarily, the switch house (as well as the substation and power centers) has devices that protect each outgoing circuit. Current transformers (CTs) are typically used to measure current while potential transformers (PTs) are used to measure voltage. A relay receives signals from the PTs and CTs, processes the information, and the determines if there are any abnormal conditions such as overloads, short circuits, ground faults, or under-voltage. If an abnormal condition is detected by the relay, a trip signal is sent from the relay to the circuit breaker to de-energize the offending circuit.

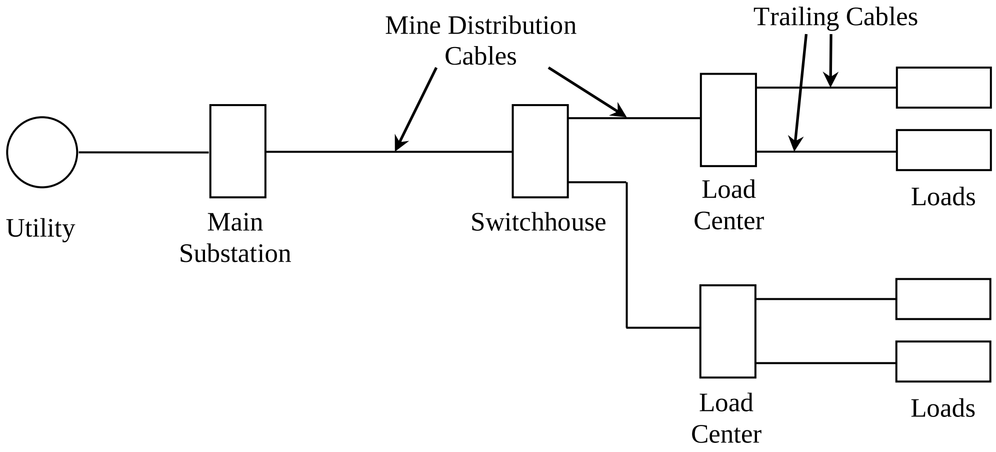


Figure 4.12 Mine Power System Example

Figure 4.13 illustrates these protective devices in the switch house. Each power conductor passes through a line current CT. In addition, all three power conductors pass through the zero-sequence CT. The zero-sequence CT is used to detect ground faults through the application of Kirchhoff's Current Law (and the theory of symmetrical components). In nearly all types of normal and abnormal conditions, including line-to-line and three-phase faults, the sum of the three line currents are zero, so there is no current induced in the zero-sequence CT. However, during a ground fault, the sum of the three line currents is not zero, and the current measured by the zero-sequence CT is equal to the ground fault current. Because many mining operations use high resistance grounding to limit ground fault current to low levels to avoid arc and flash hazards, zero sequence relaying can be used to detect low levels of ground current in a power system having load currents many times higher than the ground fault current. For example, a ground fault of 10 Amperes can easily be detected in the presence of load currents of several hundred Amperes.

Relay elements shown in this diagram include 50-G and 50, and 51. Element 50 is an instantaneous line overcurrent element typically used for detecting short circuits. Element 51 is an inverse-time line overcurrent element typically used for detecting overloads. Element 50-G is an instantaneous ground overcurrent element used to detect ground faults. It should/must be noted that modern solid-state relays typically contain many additional elements, such as reverse power flow, overvoltage, and under-voltage, that are not included in this example.

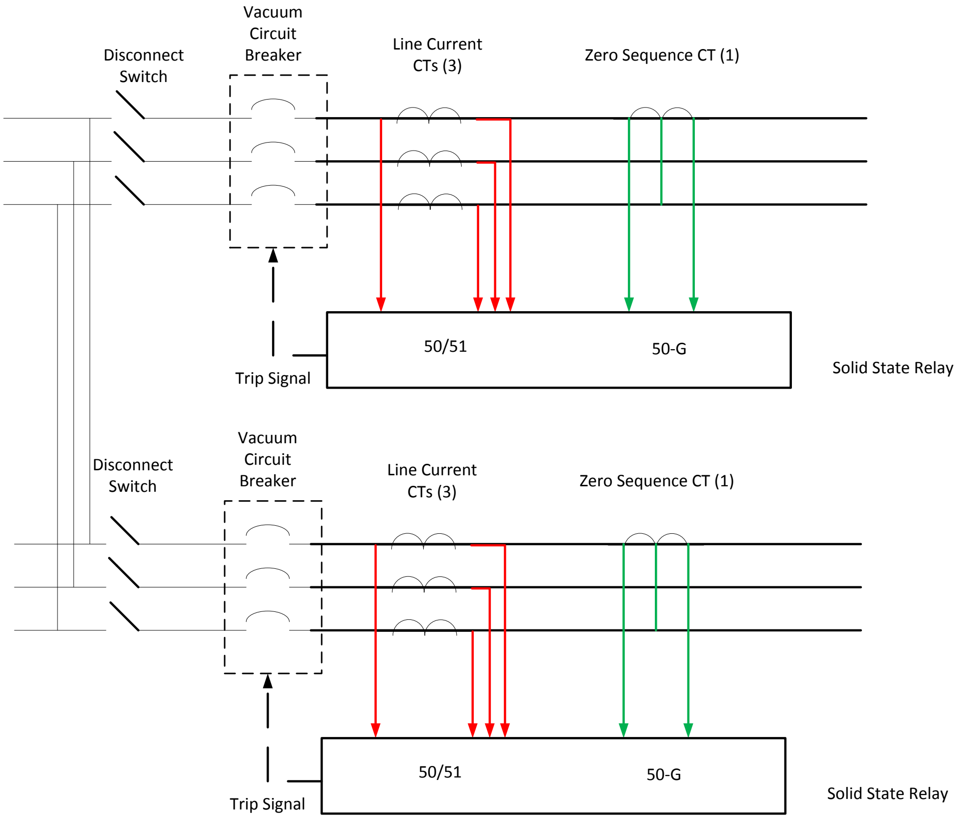


Figure 4.13 Traditional Feeder Protection Scheme

Consider that a ground fault occurs on one of the branch circuits on the output of the switchhouse, tripping the associated circuit breaker. All that would be known to downstream personnel is that the power is off at their location, i.e., mine personnel would not know the extent of, or reason for, the power outage. Consequently, they would begin calling other sections of the mine, or outside, to inquire about the extent of the power outage. After many minutes, the switch house eventually would be identified as the location where the circuit breaker tripped, and an employee would be dispatched to determine the cause of the trip. Upon arrival at the switchhouse, the individual would be able to determine from the relay panel that a ground fault occurred and request that an electrician be sent to determine the cause of the ground fault and initiate repairs.

Next, consider the situation depicted in Figure 4.14, in which the gateway module is present. Immediately after a ground fault occurs, an individual at the monitoring-system computer would know the location and reason for the power outage. As a result the correct personnel could be dispatched immediately to conduct repairs, and while personnel in the affected portions of the mine could be informed of the source and extent of the problem.

Although this is a relatively simple example of the application of the gateway module, they show the potential behind the use of this module in other applications. Specifically, the module provides the mine operator with the ability to not only monitor the status of the IED (trip/not trip), but also to monitor the values or measurand of interest of the IED and make this information available throughout the existing SCADA system. By having these electrical parameters in addition to traditional process parameters, mines have the ability to expand the use of these existing devices, such as implementing load shedding for energy management or conducting root-cause analysis of failures through reconstructing sequence of events.

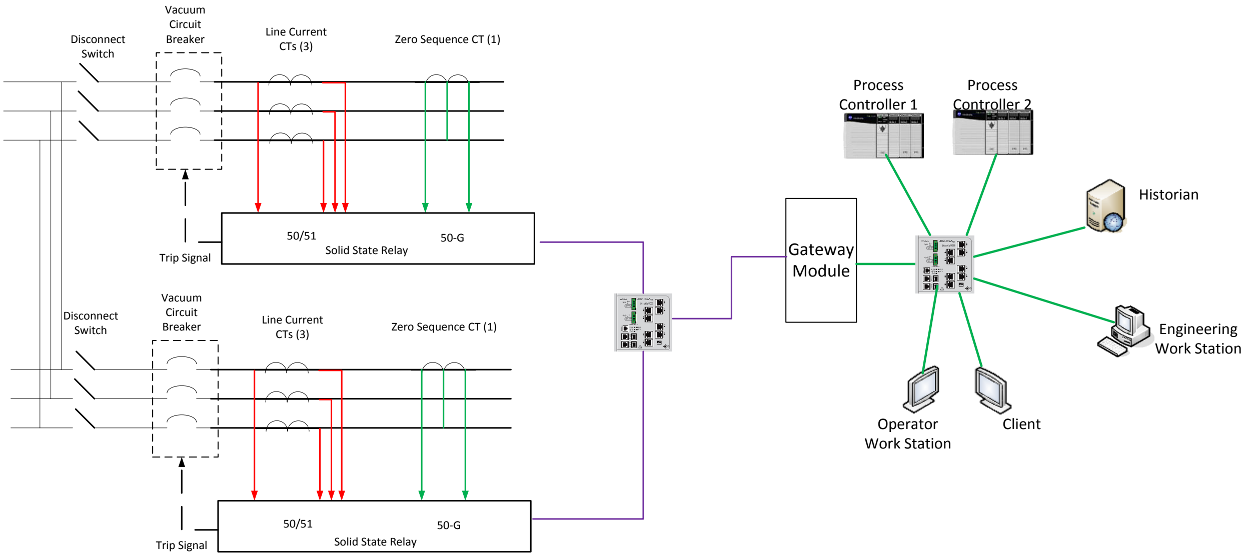


Figure 4.14 Gateway Module Implementation

Conclusions

This chapter has summarized the hardware gateway module which was developed to implement the conceptual design described in Chapter 3. A key component to this concept is the multi-threaded design to interface between the IED IEC 61850 network and the process EtherNet/IP network. Each thread, including the master control program, was defined, and basic functionality was explained in functional block diagram. The various types of data that were used in this project were then addressed, followed by how data is packaged in packets to be sent to the process network via EtherNet/IP.

The concept of the tag database was then discussed, i.e., how it is utilized as a common space of shared memory where both 61850 and EtherNet/IP drivers read and write various tags. Also discussed was the use of semaphore tags in the database to avoid collisions in the tag database. Finally, a practical example of this hardware module could be used in a mine power system application was provided. The next chapter focuses on how the user interacts with the hardware via the use of a graphical user interface (GUI) software.

Chapter 5 - Software Implementation

Introduction

In order for the hardware gateway to work properly with the process control system, software needed to be developed that allows the user to interact and configure the gateway module. As described in Chapter 4, the configuration files are held by the Secure Digital (SD) card. These files are downloaded to the SD card from the software tool that is presented in this chapter. In addition, a software tool was developed to store information in the OSI PI database on a precise timescale. This tool is also discussed in detail later in this chapter.

The procedure describing how the hardware and software work together to produce the final solution developed in this research involves several steps. First, the user creates a Configured IED Description (CID) file for each IED that is connected to the gateway module. For example, if we consider a solid state relay, the user would typically configure it for various parameters or measurands, such as active and reactive power, voltage magnitudes and angles, current magnitudes and angles, and sequence components of current and voltage. This step is conducted using the software provided by the IED vendor, following the vendor's instructions.

Next, the CID configuration is downloaded to the gateway module using the IEC 61850 standard. This step is accomplished by the software developed for the Gateway Module and is described in this chapter. It is noted here that this software was developed to work with any IED that follows the IEC 61850 standard; it is neither product nor device specific.

Next, the user selects specific information from the CID configuration that will be transferred to the programmable automation controller (PAC). The software was developed to allow the user to easily select any or all of the measurands or parameters included in the CID configuration. With regard to the relay, for example, the user may be interested in sending the voltage, current, and power quantities to the PAC, but not the sequence components of voltage and current.

The software then maps these data to EtherNet/IP via the Open Device Vendor's Association (ODVA) standard. The software also creates and exports an Add-On instruction (AOI) for the PAC. Both of these steps are done automatically by the software developed for the gateway module. For this information to be passed on to the PAC, the user only needs to add the gateway module to the controller project and import the Add-On instruction into the controller program. At this point, the IED appears as a device on a rung in the controller project and can be used like any other automation and control device connected to the controller. (It is noted that the software was developed specifically for Program Automation Controllers, manufactured by Rockwell Automation.)

The remainder of this chapter describes how the software accomplishes these tasks and also addresses process control historian issues.

Creating a Configured IED Description (CID) File

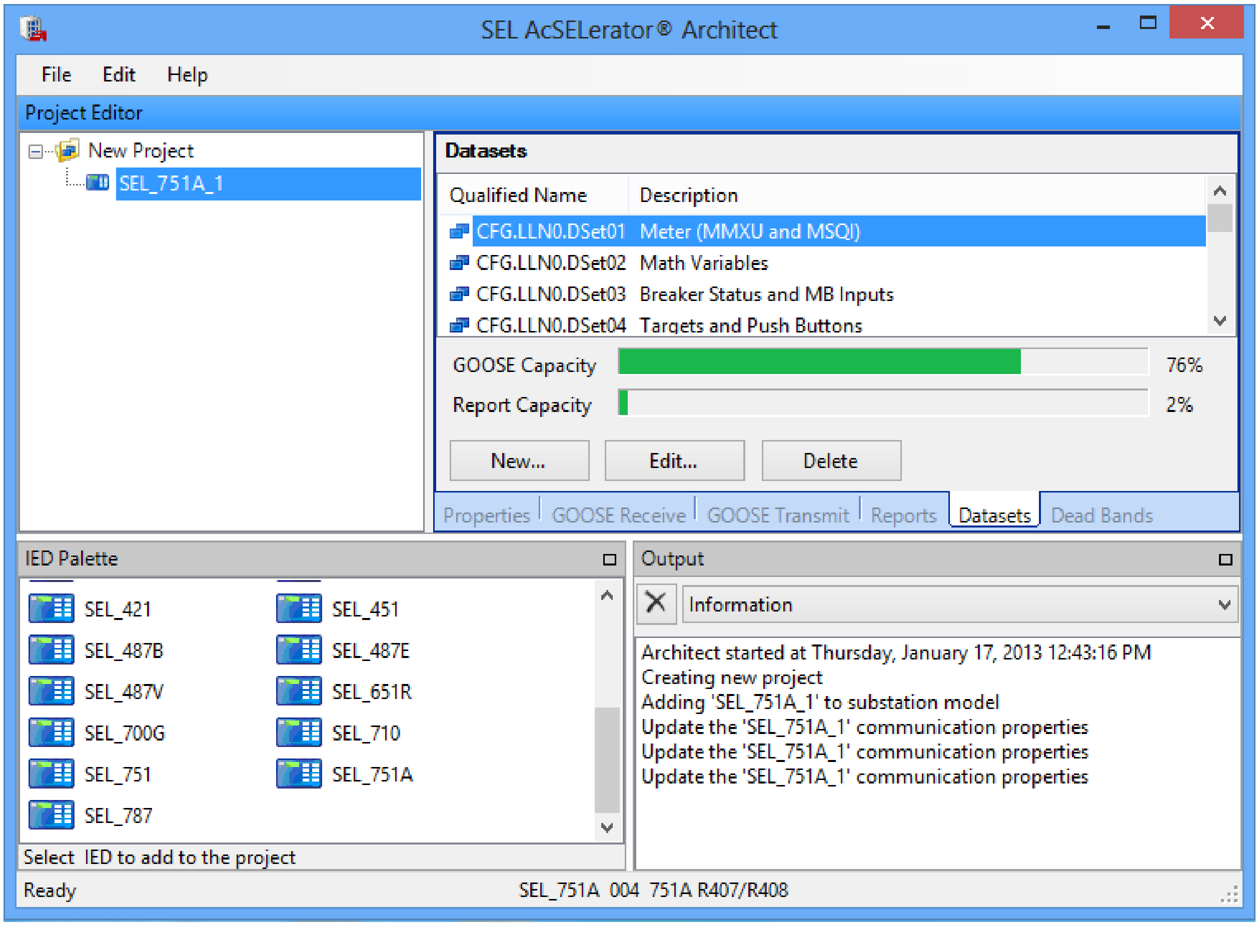
In order to have a common method for describing and documenting the communications network, IEC 61850-6-1 defines various SCL file types based on XML schemas. The specified file types are System Specification Description (SSD), Substation Configuration Description (SCD), IED Capability Description (ICD), and Configured IED Description (CID). For the purposes of this dissertation, only ICD and CID files will be considered. ICD files represent the default IEC 61850 configuration of an IED. CID files follow the same schema but represent the final configuration of an IED in service.

When opening an ICD file in a web browser or text editor, the definition of all logical devices and logical nodes for the IED is listed. In addition, the ICD file can include definitions for datasets, MMS reports, and GOOSE messages. Datasets are simply a logical collection of tags (not necessarily from the same logical node). Collecting the parameters into datasets allows the data to be efficiently used as part of a GOOSE message or MMS report.

Reports are unsolicited methods of sending datasets from an IED. The standard defines two types: buffered and unbuffered. When using buffered reports, the IED keeps track of client message receipts so that any missed reports can be re-sent if there is a network problem. Unbuffered reports do not maintain the missing data if the link is lost. The MMS protocol can also provide datasets via direct polling by the client. This method provides no buffering and eliminates deadbands on analogs.

Figure 5.1 is a screenshot of an IEC 61850 configuration tool developed by a major IED manufacturer. It can be seen from the figure that this project is configuring a 751, or feeder protection relay. For the purposes of this research, the data sets and reports tab are of most interest in this example. As previously defined, data sets are groups of measurands or variables that are grouped together for reporting purposes via the 61850 standard. When designing

SCADA systems, as defined in Chapter 2, Manufacturing Message Specification (MMS) is the mechanism for the transfer of information from server to client.



Qualified Name	Description
CFG.LLN0.DSet01	Meter (MMXU and MSQI)
CFG.LLN0.DSet02	Math Variables
CFG.LLN0.DSet03	Breaker Status and MB Inputs
CFG.LLN0.DSet04	Targets and Push Buttons

GOOSE Capacity 76%

Report Capacity 2%

```
Architect started at Thursday, January 17, 2013 12:43:16 PM
Creating new project
Adding 'SEL_751A_1' to substation model
Update the 'SEL_751A_1' communication properties
Update the 'SEL_751A_1' communication properties
Update the 'SEL_751A_1' communication properties
```

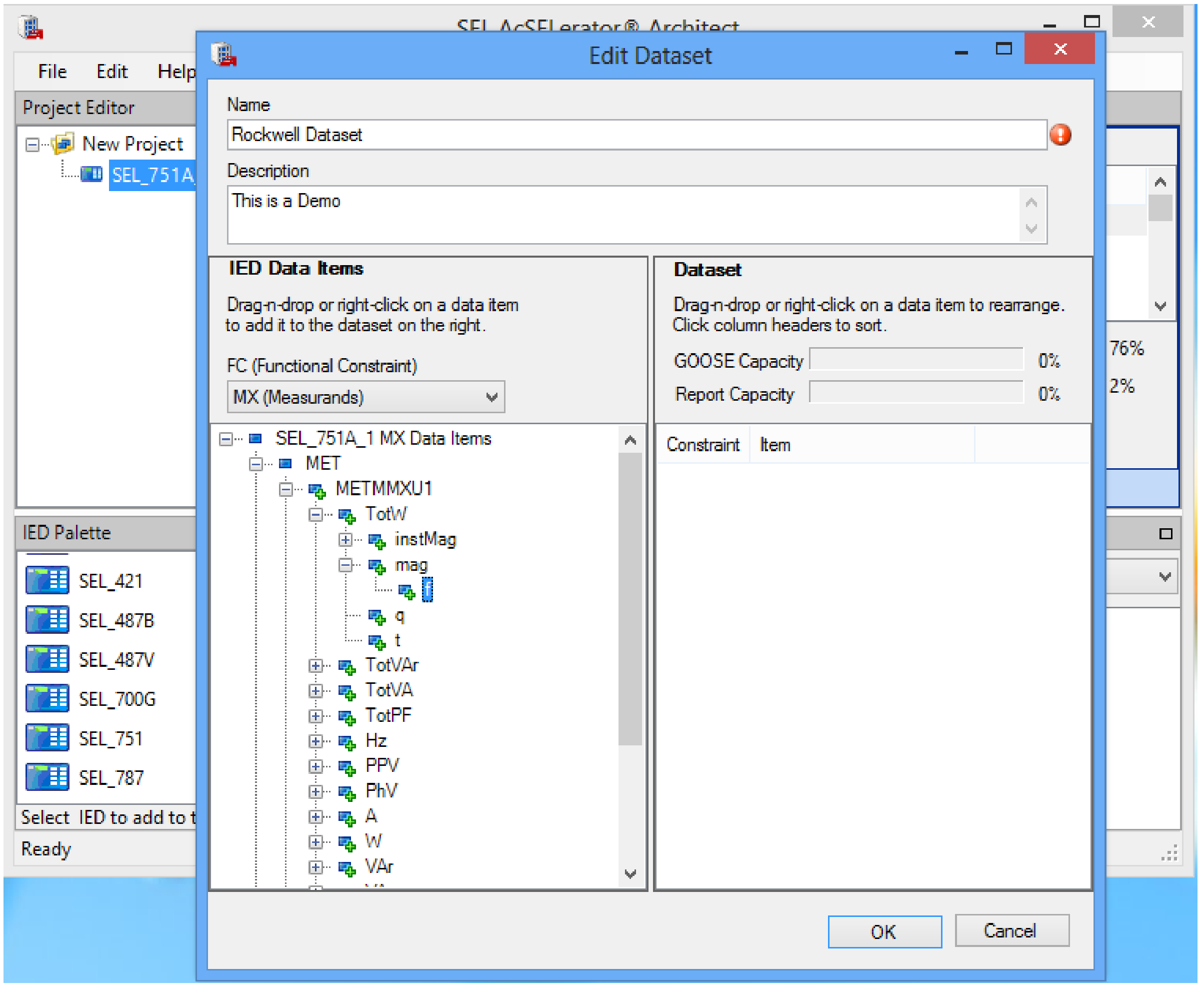
SEL_751A 004 751A R407/R408

Figure 5.1 Manufacturer IEC 61850 Configuration Tool

Figure 5.2 shows how data sets are configured using the drag and drop feature of the vendor software. As shown, all of the data is classified into various logical nodes as defined by the standard. In this example, the data set contains fundamental measurands, since tags are being chosen from the MX or Measurands Logical Node of the device. After the user is satisfied with the information created in the data set, the data set is saved and assigned to a report.

After the data set is assigned to a corresponding report, the CID file is downloaded to the IED via an FTP connection. In addition, the CID file is exported for use in this research. Figure 5.3 shows a CID file for the feeder relay configured in this example. The portion of the CID file shown corresponds to the data set configured for the example with the name RA for Rockwell

Data Set and its corresponding description. Also, the definition of the assignment of the data set, highlighted in grey, is to buffered report one. This report has been configured to send information to the controller regarding data change, quality change, or integrity period (poll period) of 1000 milliseconds or every one second. All of these values can be adjusted for each configuration and system setup.



Edit Dataset

Name: Rockwell Dataset

Description: This is a Demo

IED Data Items

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

FC (Functional Constraint): MX (Measurands)

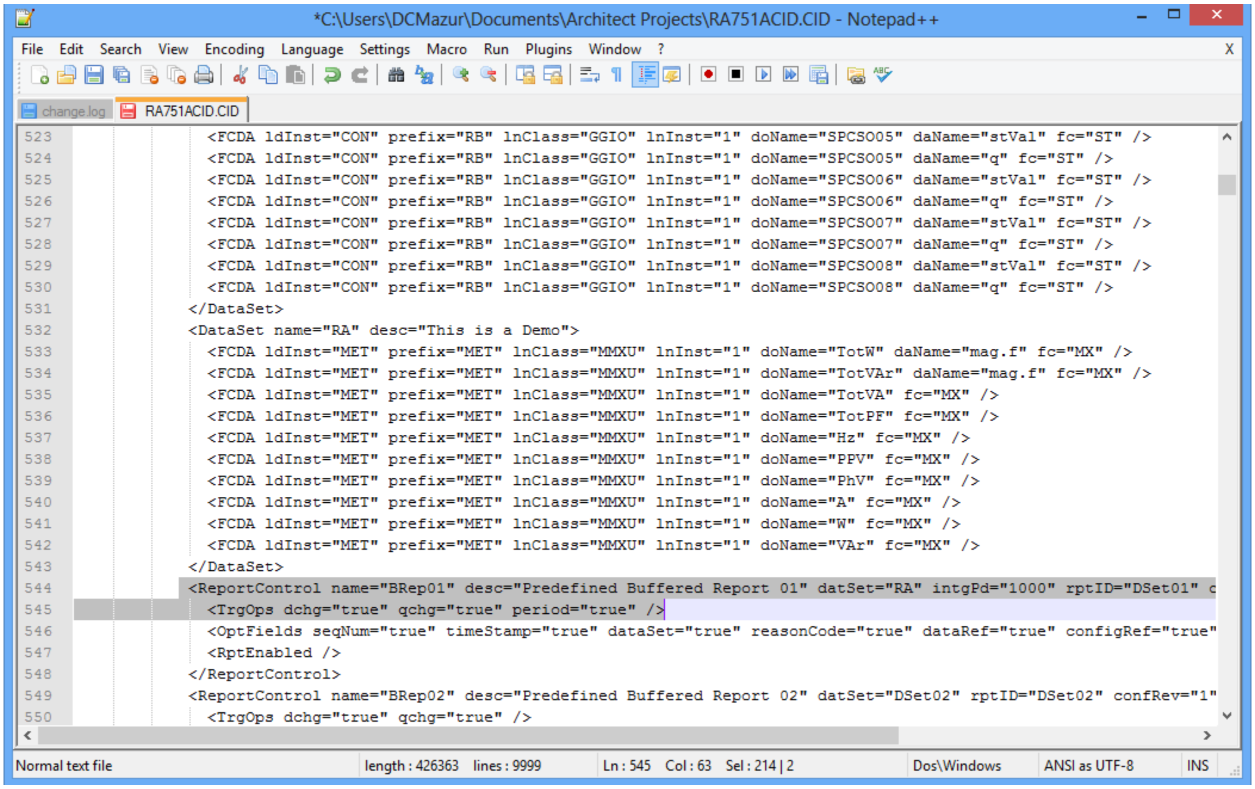
Dataset

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

GOOSE Capacity 0%

Report Capacity 0%

Figure 5.2 Editing a Data Set



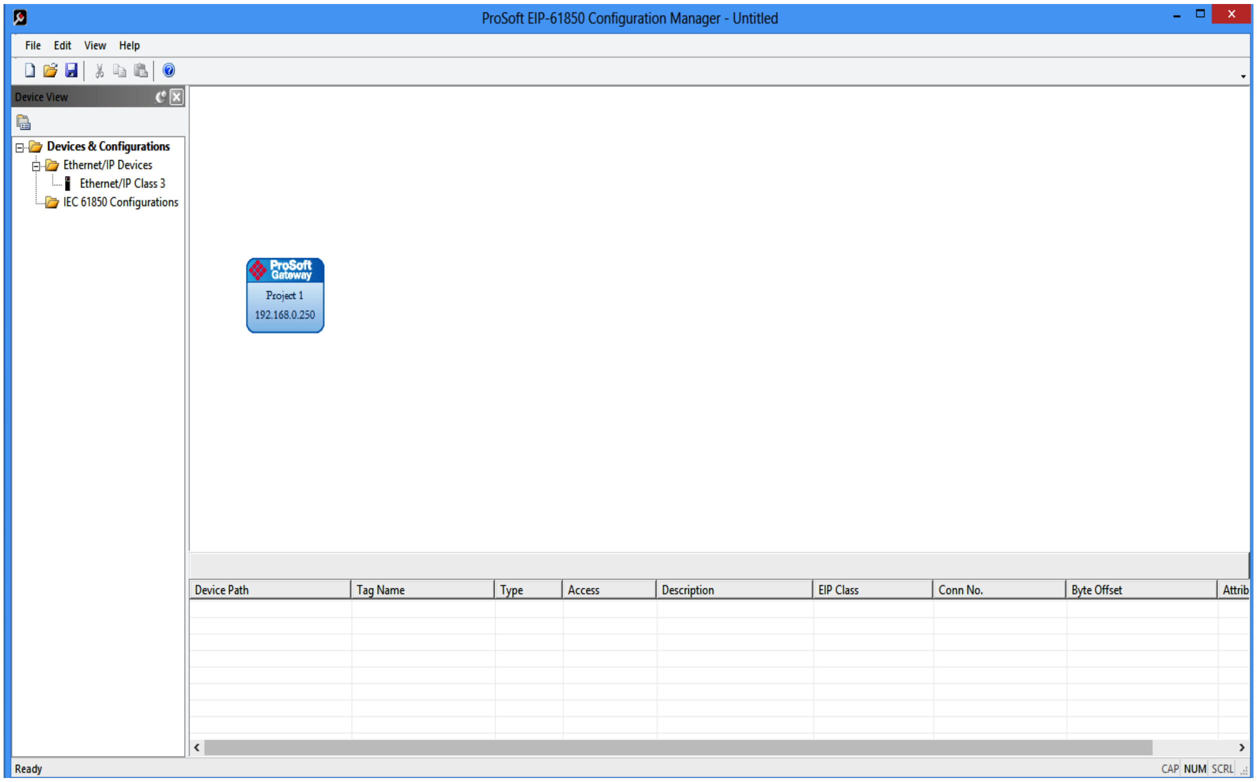
```
      <FCDA ldInst="CON" prefix="RB" lnClass="GGIO" lnInst="1" doName="SPCSO05" daName="stVal" fc="ST" />
      <FCDA ldInst="CON" prefix="RB" lnClass="GGIO" lnInst="1" doName="SPCSO05" daName="q" fc="ST" />
      <FCDA ldInst="CON" prefix="RB" lnClass="GGIO" lnInst="1" doName="SPCSO06" daName="stVal" fc="ST" />
      <FCDA ldInst="CON" prefix="RB" lnClass="GGIO" lnInst="1" doName="SPCSO06" daName="q" fc="ST" />
      <FCDA ldInst="CON" prefix="RB" lnClass="GGIO" lnInst="1" doName="SPCSO07" daName="stVal" fc="ST" />
      <FCDA ldInst="CON" prefix="RB" lnClass="GGIO" lnInst="1" doName="SPCSO07" daName="q" fc="ST" />
      <FCDA ldInst="CON" prefix="RB" lnClass="GGIO" lnInst="1" doName="SPCSO08" daName="stVal" fc="ST" />
      <FCDA ldInst="CON" prefix="RB" lnClass="GGIO" lnInst="1" doName="SPCSO08" daName="q" fc="ST" />
    </DataSet>
    <DataSet name="RA" desc="This is a Demo">
      <FCDA ldInst="MET" prefix="MET" lnClass="MMXU" lnInst="1" doName="TotW" daName="mag.f" fc="MX" />
      <FCDA ldInst="MET" prefix="MET" lnClass="MMXU" lnInst="1" doName="TotVAr" daName="mag.f" fc="MX" />
      <FCDA ldInst="MET" prefix="MET" lnClass="MMXU" lnInst="1" doName="TotVA" fc="MX" />
      <FCDA ldInst="MET" prefix="MET" lnClass="MMXU" lnInst="1" doName="TotPF" fc="MX" />
      <FCDA ldInst="MET" prefix="MET" lnClass="MMXU" lnInst="1" doName="Hz" fc="MX" />
      <FCDA ldInst="MET" prefix="MET" lnClass="MMXU" lnInst="1" doName="PPV" fc="MX" />
      <FCDA ldInst="MET" prefix="MET" lnClass="MMXU" lnInst="1" doName="PhV" fc="MX" />
      <FCDA ldInst="MET" prefix="MET" lnClass="MMXU" lnInst="1" doName="A" fc="MX" />
      <FCDA ldInst="MET" prefix="MET" lnClass="MMXU" lnInst="1" doName="W" fc="MX" />
      <FCDA ldInst="MET" prefix="MET" lnClass="MMXU" lnInst="1" doName="VAr" fc="MX" />
    </DataSet>
    <ReportControl name="BRep01" desc="Predefined Buffered Report 01" datSet="RA" intgPd="1000" rptID="DSet01" c
      <TrgOps dchg="true" qchg="true" period="true" />
      <OptFields seqNum="true" timeStamp="true" dataSet="true" reasonCode="true" dataRef="true" configRef="true"
      <RptEnabled />
    </ReportControl>
    <ReportControl name="BRep02" desc="Predefined Buffered Report 02" datSet="DSet02" rptID="DSet02" confRev="1"
      <TrgOps dchg="true" qchg="true" />
```

Figure 5.3 CID File Example Feeder Relay

Importing the CID File into Software

As mentioned, the software tool was created to simplify the user's experience with the gateway module and ease of implementation constructing complex SCADA systems. The program was created in the Microsoft Visual Studio environment using Visual C++. The software is responsible for reading the CID file from the relay, parsing the information, allowing the user to configure final reports, mapping the data from IEC 61850 tags to EtherNet/IP tags, writing a final configuration in XML for the gateway module, and downloading this configuration to the SD card of the gateway.

Figure 5.4 depicts the workspace where the user interacts with the gateway module. When the user opens a new project in the software, the gateway module is the only device that is defined in the project by the gateway block. In this example the gateway is assigned an IP address of 192.168.0.250. The left hand pane of the work space defines the devices that are currently available to be configured; the project has the available EtherNet/IP connection for communications to the controller. The bottom pane of the workspace defines a listing of all the tags that are configured to be transferred from the electric distribution network to the process control network.



Device Path	Tag Name	Type	Access	Description	EIP Class	Conn No.	Byte Offset	Attrib

Figure 5.4 Gateway Module Workspace

By right clicking on the gateway block and selecting *properties*, as shown in Figure 5.5, the user is prompted with the screen shown in Figure 5.6. In this menu, the user is able to assign the IP address of his or her choosing to the gateway module as well as define corresponding network information such as subnet mask and the default gateway in the networked system. The user then provides a name for the project so that the configuration can be saved. At this point, the user is ready to import the CID files to create a virtualized IED network in the software configuration tool.

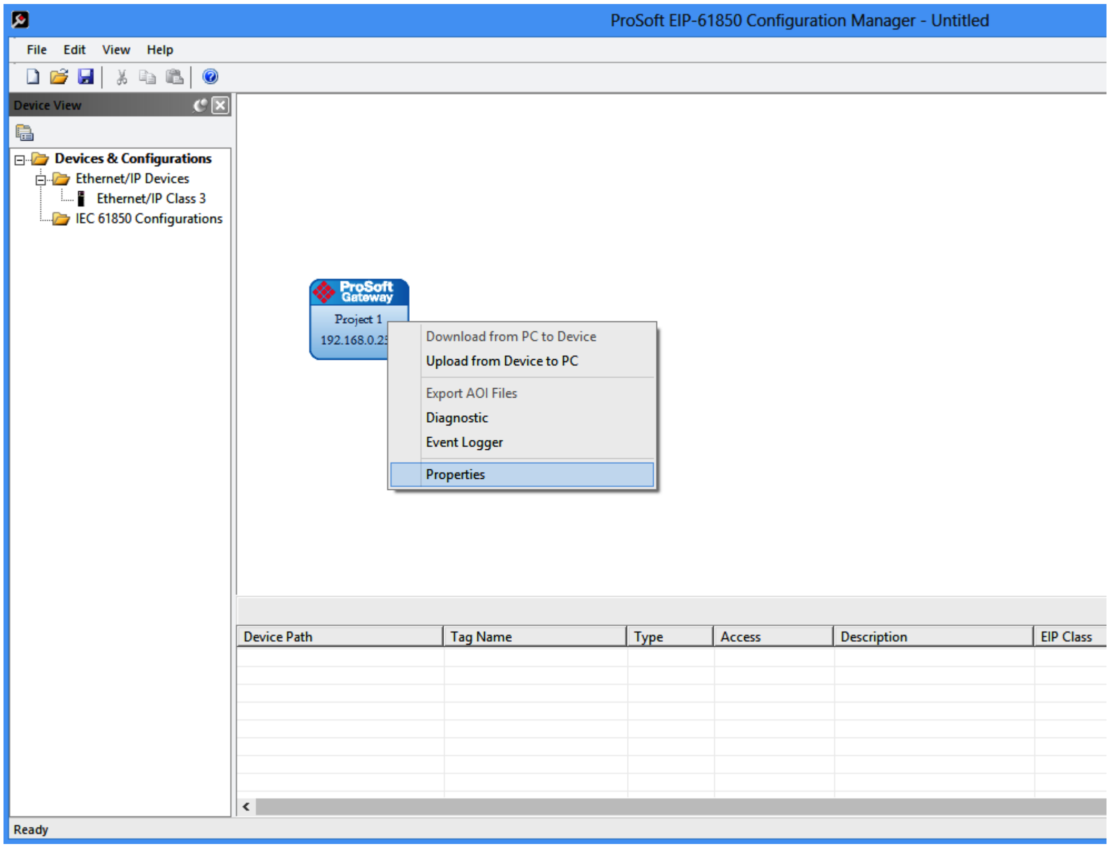
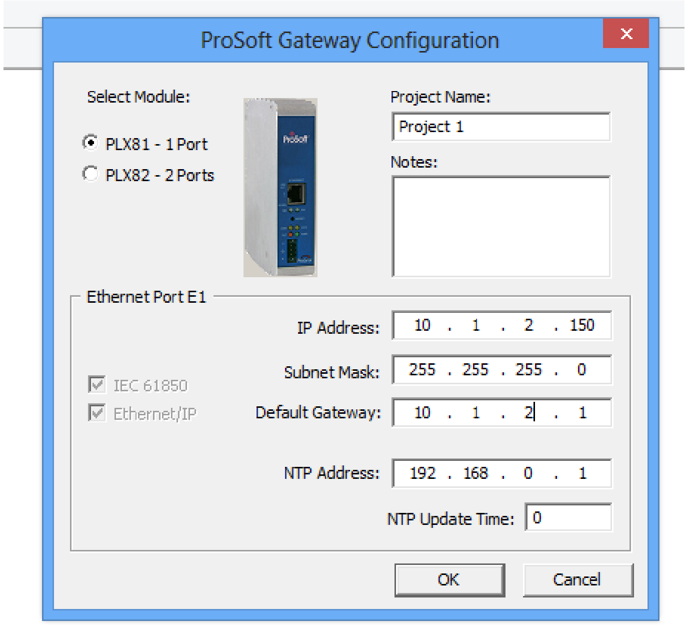


Figure 5.5 Configure Gateway Module



Select Module:

● PLX81 - 1 Port

○ PLX82 - 2 Ports

Project Name: Project 1

Notes:

Ethernet Port E1

☑ IEC 61850

☑ Ethernet/IP

IP Address: 10 . 1 . 2 . 150

Subnet Mask: 255 . 255 . 255 . 0

Default Gateway: 10 . 1 . 2 . 1

NTP Address: 192 . 168 . 0 . 1

NTP Update Time: 0

Figure 5.6 Configure Gateway IP Address

After defining the IP address of the gateway, the user can add IEDs to the virtualized network as previously defined. In the left window pane of the workspace, the user can add devices to the IEC 61850 configurations folder, as seen in Figure 5.7. The user then points to the CID file that was exported from the manufacturer’s relay configuration software and imports the virtualized IED into the workspace. Figure 5.8 shows the 751A feeder relay added to the project which is recognized by the software through the corresponding manufacturer assigned to the name. The IED can then be dragged and dropped to the workspace as shown in Figure 5.9. The virtualized IED is defined just like the gateway; it is a function block with its own assets. Up to 20 IEDs, regardless of vendor, can be added to the gateway module.

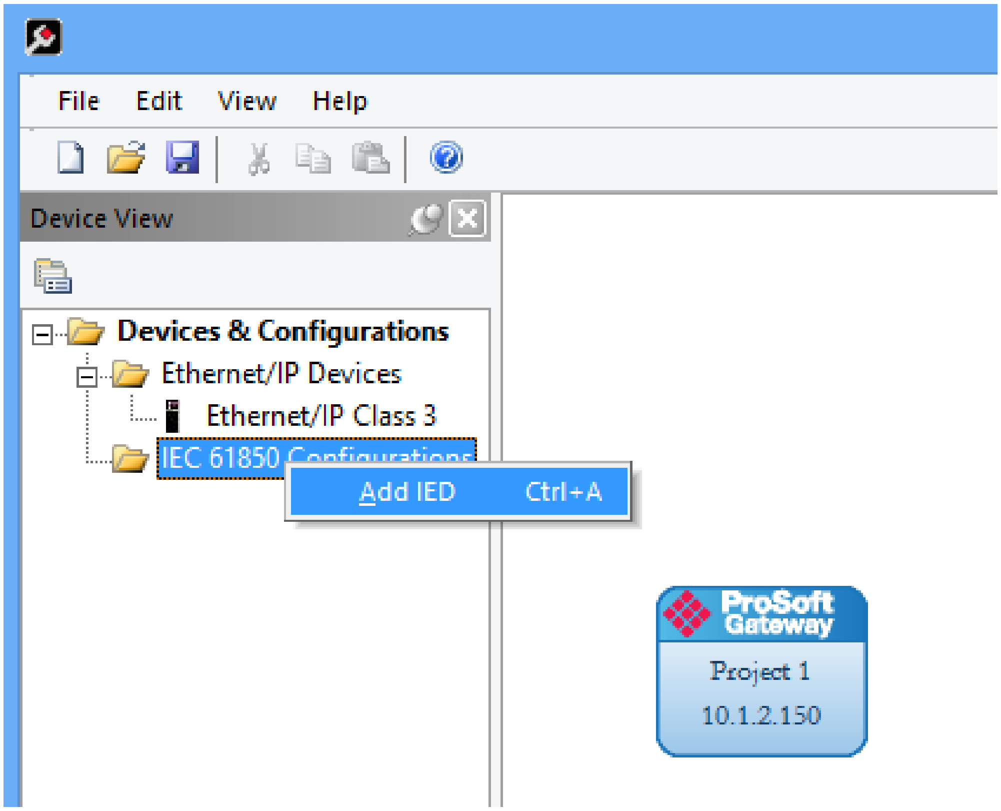


Figure 5.7 Add IED from CID File

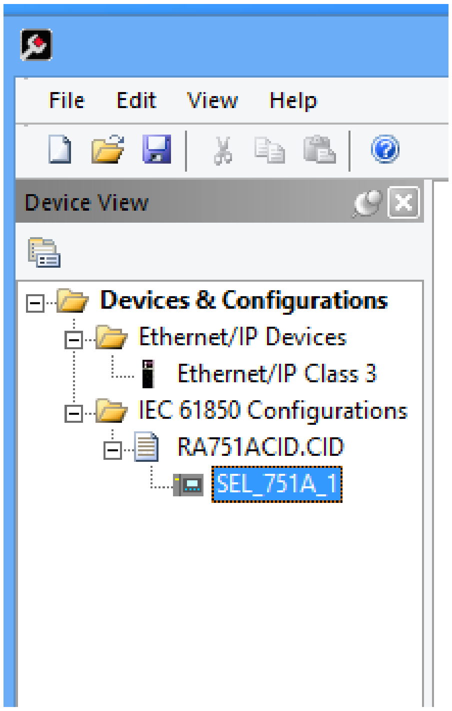


Figure 5.8 Feeder Relay Added to Project

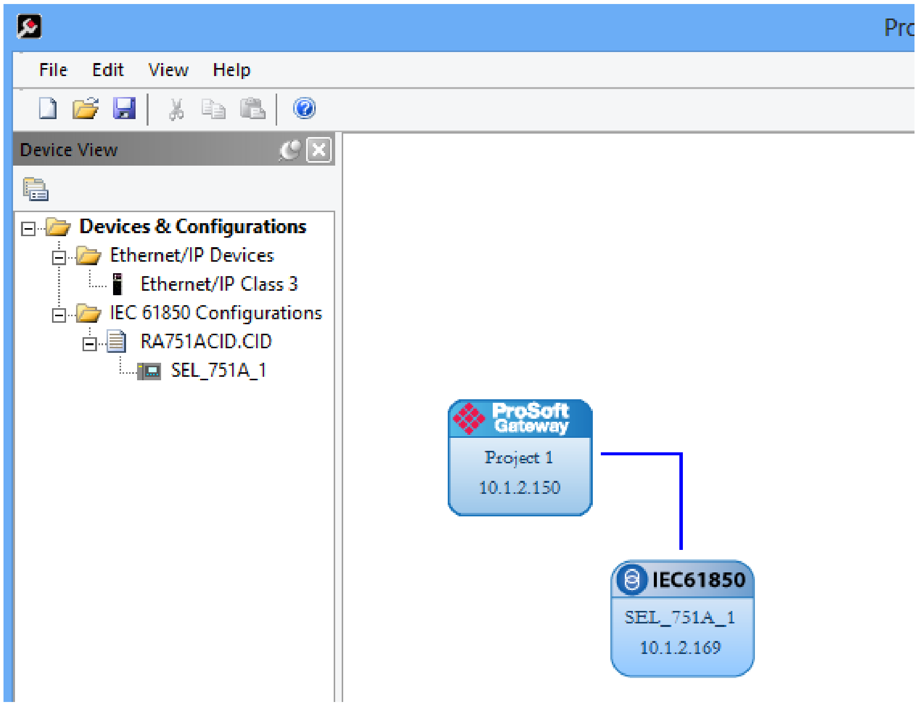


Figure 5.9 Add IED to Workspace

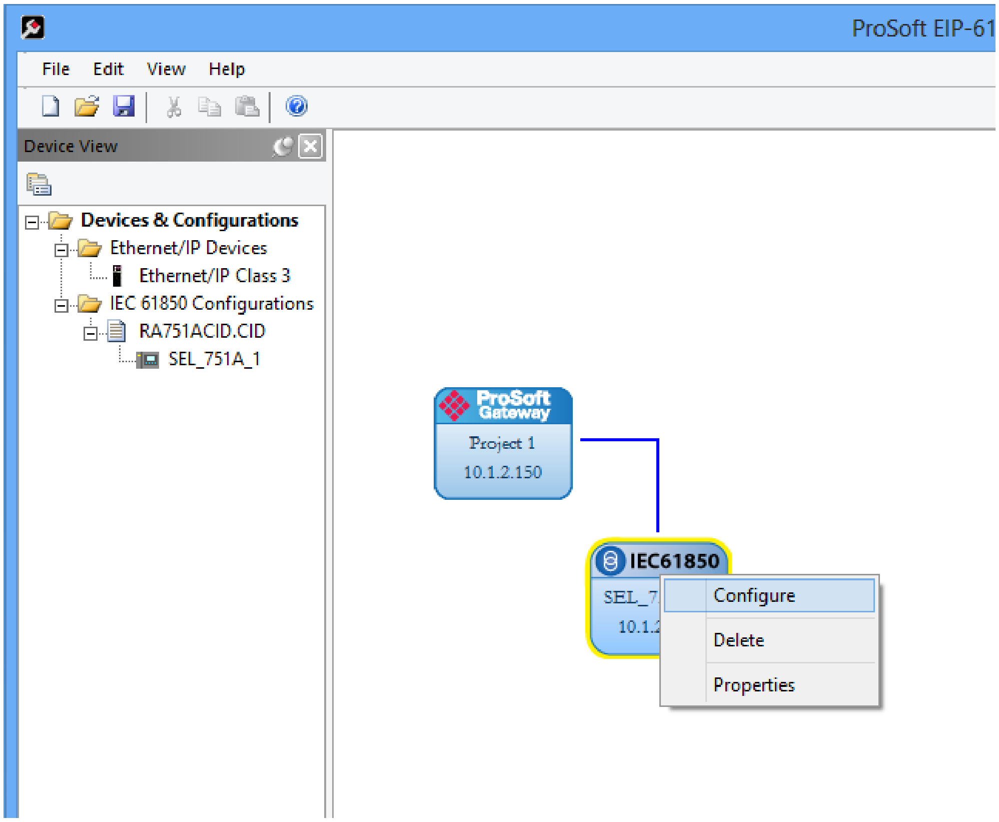


Figure 5.10 Configuring IED in Workspace

As previously shown for the gateway module, the individual IED is shown with the IP address as defined in the CID file. After importing the virtualized IED into the IEC 61850 network, the user now needs to configure what information is to be mapped from the relay to the controller. It is important to note that all reports that were defined in the CID file could automatically be mapped; however, it was decided to provide the increased flexibility of allowing the user to specify what reports and additional information should be mapped to the process control system. After choosing *configure* under the drop down menu, the screen shown in Figure 5.11 is depicted.

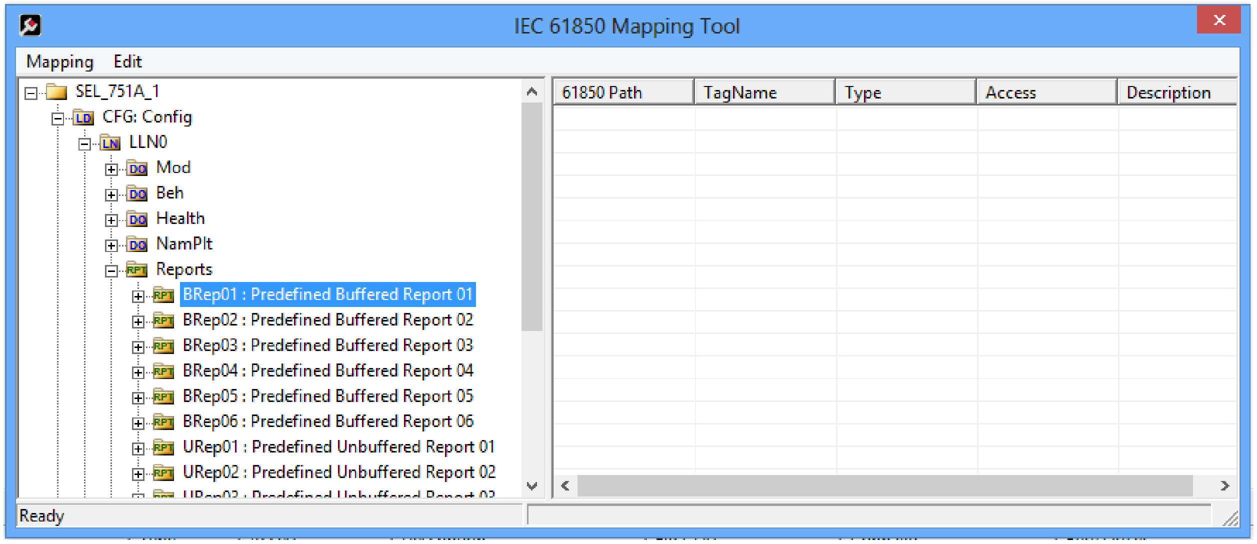
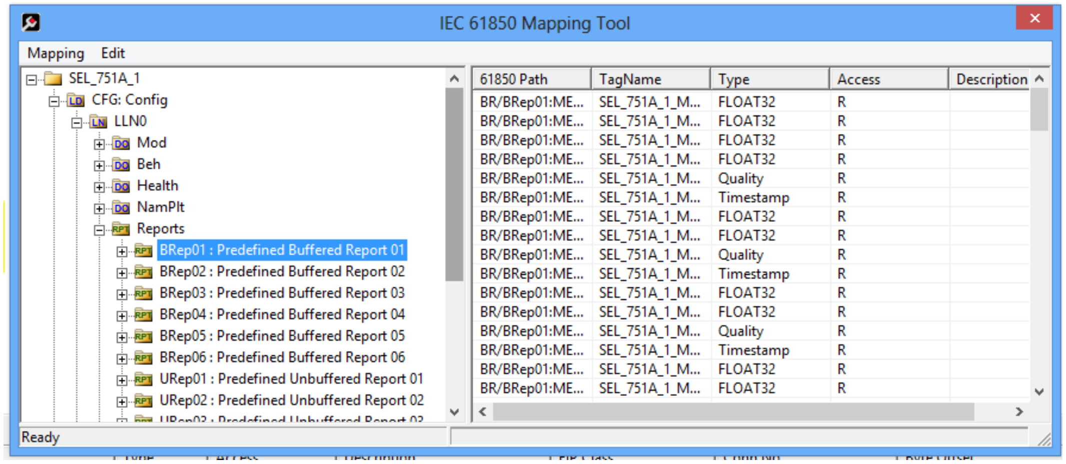

IEC 61850 Mapping Tool
Mapping
Edit
SEL_751A_1
CFG: Config
LLN0
Mod
Beh
Health
NamPlt
Reports
BRep01 : Predefined Buffered Report 01
BRep02 : Predefined Buffered Report 02
BRep03 : Predefined Buffered Report 03
BRep04 : Predefined Buffered Report 04
BRep05 : Predefined Buffered Report 05
BRep06 : Predefined Buffered Report 06
URep01 : Predefined Unbuffered Report 01
URep02 : Predefined Unbuffered Report 02
URep03 : Predefined Unbuffered Report 03
61850 Path
TagName
Type
Access
Description
Ready

Figure 5.11 Non-Configured IEC 61850 Tags

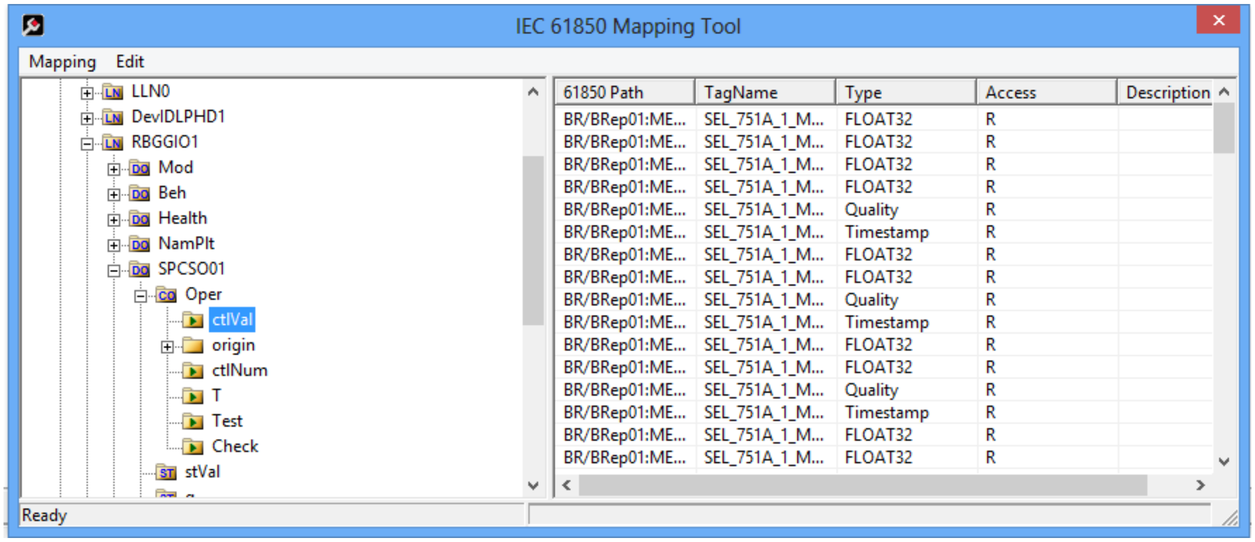
After the configuration section is chosen for an individual IED, the user can browse through the parsed CID file and has the freedom to browse through all configured logical nodes. In this example the LLN0, or base logical node of the feeder relay, is selected. Furthermore, the *reports* folder is chosen and buffered report one (BREP01) is selected to be mapped to the controller. By dragging and dropping the desired information, i.e. the report folder, the information is parsed from the report and displayed in the right workspace panel.

Once the information is mapped, the user sees a screen similar to that in Figure 5.12. When the user wants to map remote bits for command and control, the user maps the control value (ctval) of the remote bit (RBGGIO1) logical node. These bits are simple Boolean values that can be mapped to actuate the IED, for instance with open/close or remote trip. The mapping of these bits can be seen in Figure 5.13. The values mapped from reports are known as controller inputs, while remote bits are known as controller outputs.


IEC 61850 Mapping Tool
Mapping
Edit
SEL_751A_1
CFG: Config
LLN0
Mod
Beh
Health
NamPlt
Reports
BRep01 : Predefined Buffered Report 01
BRep02 : Predefined Buffered Report 02
BRep03 : Predefined Buffered Report 03
BRep04 : Predefined Buffered Report 04
BRep05 : Predefined Buffered Report 05
BRep06 : Predefined Buffered Report 06
URep01 : Predefined Unbuffered Report 01
URep02 : Predefined Unbuffered Report 02
URep03 : Predefined Unbuffered Report 03
Ready

61850 Path	TagName	Type	Access	Description
BR/BRep01:ME...	SEL_751A_1_M...	FLOAT32	R	
BR/BRep01:ME...	SEL_751A_1_M...	FLOAT32	R	
BR/BRep01:ME...	SEL_751A_1_M...	FLOAT32	R	
BR/BRep01:ME...	SEL_751A_1_M...	FLOAT32	R	
BR/BRep01:ME...	SEL_751A_1_M...	Quality	R	
BR/BRep01:ME...	SEL_751A_1_M...	Timestamp	R	
BR/BRep01:ME...	SEL_751A_1_M...	FLOAT32	R	
BR/BRep01:ME...	SEL_751A_1_M...	FLOAT32	R	
BR/BRep01:ME...	SEL_751A_1_M...	Quality	R	
BR/BRep01:ME...	SEL_751A_1_M...	Timestamp	R	
BR/BRep01:ME...	SEL_751A_1_M...	FLOAT32	R	
BR/BRep01:ME...	SEL_751A_1_M...	FLOAT32	R	
BR/BRep01:ME...	SEL_751A_1_M...	Quality	R	
BR/BRep01:ME...	SEL_751A_1_M...	Timestamp	R	
BR/BRep01:ME...	SEL_751A_1_M...	FLOAT32	R	
BR/BRep01:ME...	SEL_751A_1_M...	FLOAT32	R	

Figure 5.12 Configured IEC 61850 Tags



61850 Path	TagName	Type	Access	Description
BR/BRep01:ME...	SEL_751A_1_M...	FLOAT32	R	
BR/BRep01:ME...	SEL_751A_1_M...	FLOAT32	R	
BR/BRep01:ME...	SEL_751A_1_M...	FLOAT32	R	
BR/BRep01:ME...	SEL_751A_1_M...	FLOAT32	R	
BR/BRep01:ME...	SEL_751A_1_M...	Quality	R	
BR/BRep01:ME...	SEL_751A_1_M...	Timestamp	R	
BR/BRep01:ME...	SEL_751A_1_M...	FLOAT32	R	
BR/BRep01:ME...	SEL_751A_1_M...	FLOAT32	R	
BR/BRep01:ME...	SEL_751A_1_M...	Quality	R	
BR/BRep01:ME...	SEL_751A_1_M...	Timestamp	R	
BR/BRep01:ME...	SEL_751A_1_M...	FLOAT32	R	
BR/BRep01:ME...	SEL_751A_1_M...	FLOAT32	R	
BR/BRep01:ME...	SEL_751A_1_M...	Quality	R	
BR/BRep01:ME...	SEL_751A_1_M...	Timestamp	R	
BR/BRep01:ME...	SEL_751A_1_M...	FLOAT32	R	
BR/BRep01:ME...	SEL_751A_1_M...	FLOAT32	R	

Figure 5.13 Adding Remote Bits

Figure 5.14 shows the virtualized substation configuration in the software once the 751A feeder relay is imported into the configuration. After the IED is imported into the software, dragged and dropped into the configuration, and the data is mapped for the IEC 61850 driver, the connection to the control system via EtherNet/IP needs to be defined. By right clicking the EtherNet/IP Device block and selecting *configure* from the drop down menu, the user is able to map and define how these IEC 61850 tags will be mapped to ControlLogix control system.

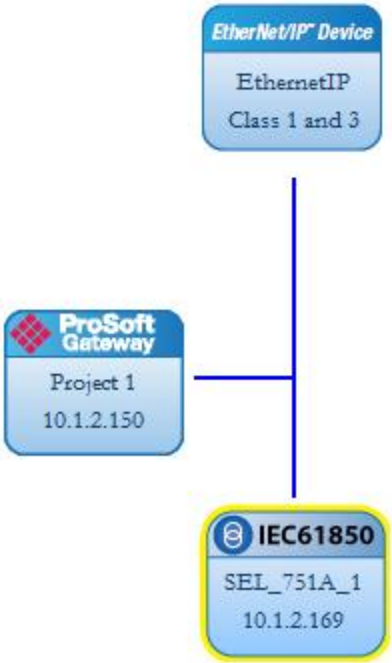
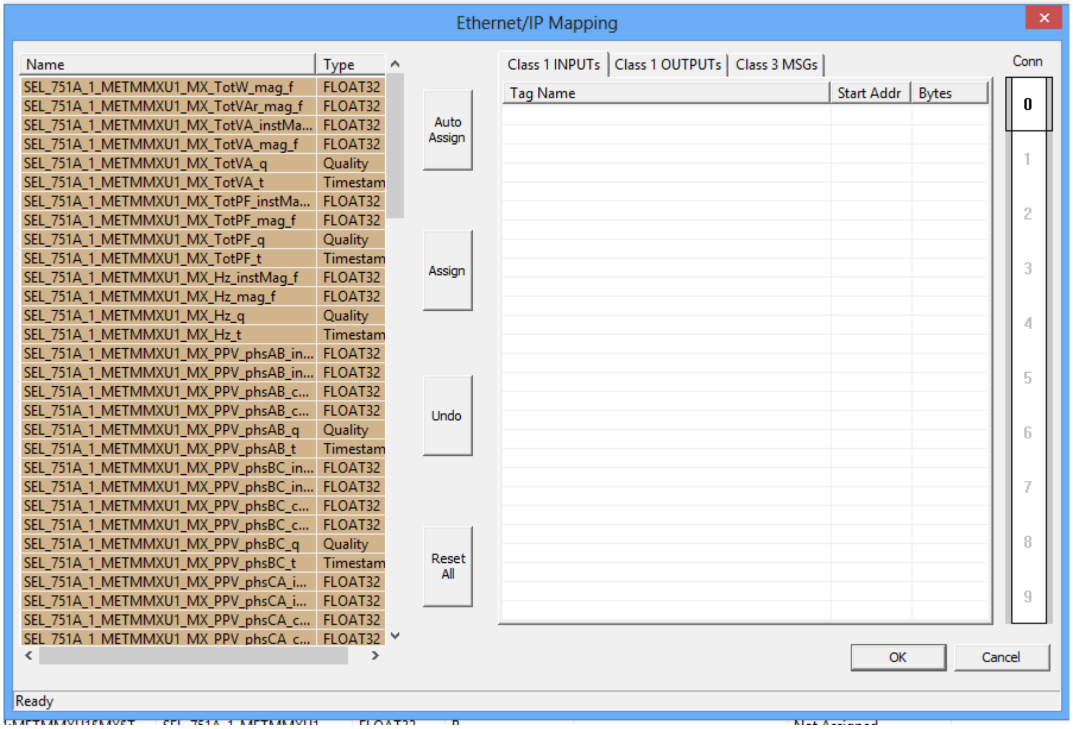


Figure 5.14 Adding EtherNet/IP Controller

Figures 5.15 and 5.16 show IEC 61850 tags are mapped to Class 1 CIP connections to deliver them, via EtherNet/IP, to the automation controller. Prior to this screen being displayed, the software reads the tag database, as defined by the IEC 61850 tag configurations. The tag database is then parsed, and each tag is assigned a corresponding data type, each with the corresponding number of bytes to be mapped to the controller. The user then has the option to auto-assign all of the data to various CIP connections or assign each data tag point by point. For purposes of this research and practical implementation, it is recommended that the auto-assign feature be used to configure the system if no data priority is needed/required.

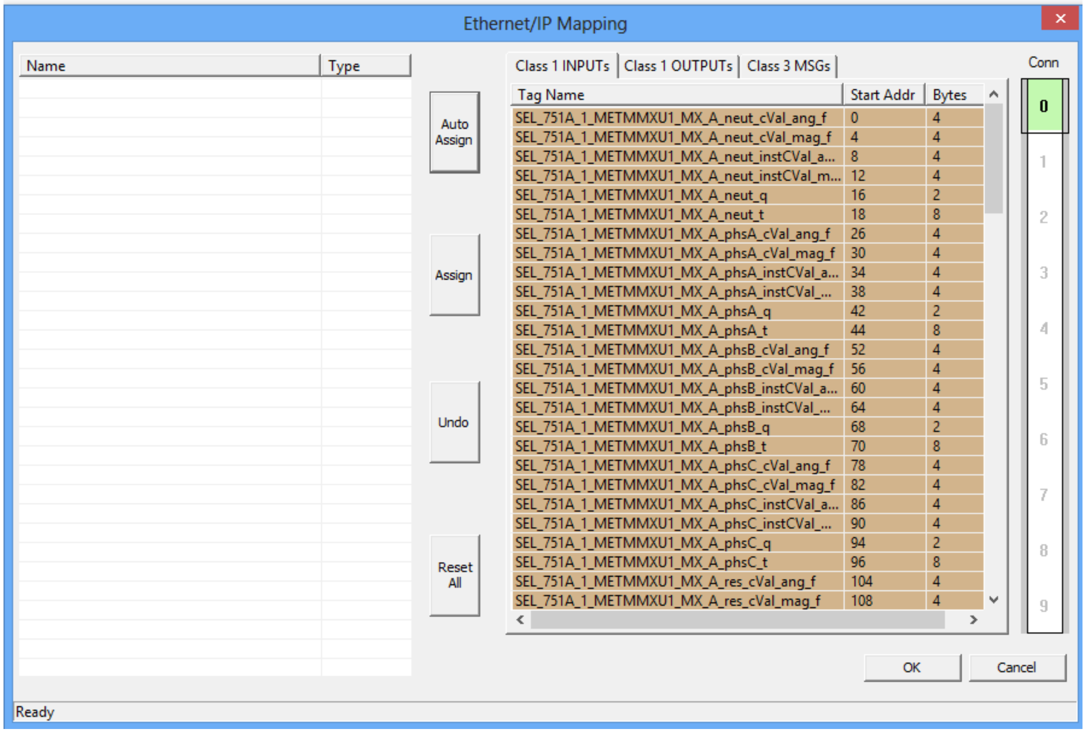
Once the data is mapped, as seen in Figure 5.16, various tabs in the right-hand workspace window pane can be used to navigate between Class 1 inputs, Class 1 outputs, and Class 3 message instructions. It can be seen that the starting address of each data point is incremented by the number of bytes that the data type corresponds. For example, the first item in Figure 5.16 defines the phase angle of the neutral current with a starting address at index zero of the first connection with a length of four bytes. The second item in the list is the corresponding magnitude to the neutral current, which in-turn starts at index four. The indicator down the right side of the window in Figure 5.16 provides the user with a visual indication of the number of CIP connections used, based on the data mapped from the gateway module to the control system.



Ethernet/IP Mapping

Name	Type
SEL_751A_1_METMMXU1_MX_TotW_mag_f	FLOAT32
SEL_751A_1_METMMXU1_MX_TotVAr_mag_f	FLOAT32
SEL_751A_1_METMMXU1_MX_TotVA_instMa...	FLOAT32
SEL_751A_1_METMMXU1_MX_TotVA_mag_f	FLOAT32
SEL_751A_1_METMMXU1_MX_TotVA_q	Quality
SEL_751A_1_METMMXU1_MX_TotVA_t	Timestam
SEL_751A_1_METMMXU1_MX_TotPF_instMa...	FLOAT32
SEL_751A_1_METMMXU1_MX_TotPF_mag_f	FLOAT32
SEL_751A_1_METMMXU1_MX_TotPF_q	Quality
SEL_751A_1_METMMXU1_MX_TotPF_t	Timestam
SEL_751A_1_METMMXU1_MX_Hz_instMag_f	FLOAT32
SEL_751A_1_METMMXU1_MX_Hz_mag_f	FLOAT32
SEL_751A_1_METMMXU1_MX_Hz_q	Quality
SEL_751A_1_METMMXU1_MX_Hz_t	Timestam
SEL_751A_1_METMMXU1_MX_PPV_phsAB_in...	FLOAT32
SEL_751A_1_METMMXU1_MX_PPV_phsAB_in...	FLOAT32
SEL_751A_1_METMMXU1_MX_PPV_phsAB_c...	FLOAT32
SEL_751A_1_METMMXU1_MX_PPV_phsAB_c...	FLOAT32
SEL_751A_1_METMMXU1_MX_PPV_phsAB_q	Quality
SEL_751A_1_METMMXU1_MX_PPV_phsAB_t	Timestam
SEL_751A_1_METMMXU1_MX_PPV_phsBC_in...	FLOAT32
SEL_751A_1_METMMXU1_MX_PPV_phsBC_in...	FLOAT32
SEL_751A_1_METMMXU1_MX_PPV_phsBC_c...	FLOAT32
SEL_751A_1_METMMXU1_MX_PPV_phsBC_c...	FLOAT32
SEL_751A_1_METMMXU1_MX_PPV_phsBC_q	Quality
SEL_751A_1_METMMXU1_MX_PPV_phsBC_t	Timestam
SEL_751A_1_METMMXU1_MX_PPV_phsCA_i...	FLOAT32
SEL_751A_1_METMMXU1_MX_PPV_phsCA_i...	FLOAT32
SEL_751A_1_METMMXU1_MX_PPV_phsCA_c...	FLOAT32
SEL_751A_1_METMMXU1_MX_PPV_phsCA_c...	FLOAT32

Figure 5.15 Mapping IEC 61850 Tags to EtherNet/IP Tags



Ethernet/IP Mapping

Tag Name	Start Addr	Bytes
SEL_751A_1_METMMXU1_MX_A_neut_cVal_ang_f	0	4
SEL_751A_1_METMMXU1_MX_A_neut_cVal_mag_f	4	4
SEL_751A_1_METMMXU1_MX_A_neut_instCVal_a...	8	4
SEL_751A_1_METMMXU1_MX_A_neut_instCVal_m...	12	4
SEL_751A_1_METMMXU1_MX_A_neut_q	16	2
SEL_751A_1_METMMXU1_MX_A_neut_t	18	8
SEL_751A_1_METMMXU1_MX_A_phsA_cVal_ang_f	26	4
SEL_751A_1_METMMXU1_MX_A_phsA_cVal_mag_f	30	4
SEL_751A_1_METMMXU1_MX_A_phsA_instCVal_a...	34	4
SEL_751A_1_METMMXU1_MX_A_phsA_instCVal_...	38	4
SEL_751A_1_METMMXU1_MX_A_phsA_q	42	2
SEL_751A_1_METMMXU1_MX_A_phsA_t	44	8
SEL_751A_1_METMMXU1_MX_A_phsB_cVal_ang_f	52	4
SEL_751A_1_METMMXU1_MX_A_phsB_cVal_mag_f	56	4
SEL_751A_1_METMMXU1_MX_A_phsB_instCVal_a...	60	4
SEL_751A_1_METMMXU1_MX_A_phsB_instCVal_...	64	4
SEL_751A_1_METMMXU1_MX_A_phsB_q	68	2
SEL_751A_1_METMMXU1_MX_A_phsB_t	70	8
SEL_751A_1_METMMXU1_MX_A_phsC_cVal_ang_f	78	4
SEL_751A_1_METMMXU1_MX_A_phsC_cVal_mag_f	82	4
SEL_751A_1_METMMXU1_MX_A_phsC_instCVal_a...	86	4
SEL_751A_1_METMMXU1_MX_A_phsC_instCVal_...	90	4
SEL_751A_1_METMMXU1_MX_A_phsC_q	94	2
SEL_751A_1_METMMXU1_MX_A_phsC_t	96	8
SEL_751A_1_METMMXU1_MX_A_res_cVal_ang_f	104	4
SEL_751A_1_METMMXU1_MX_A_res_cVal_mag_f	108	4

Figure 5.16 Configured EtherNet/IP Mapping Scheme

After defining the mapping scheme from IEC 61850 to EtherNet/IP, the user can download the project and configuration to the gateway as well as export Add-On Instructions (AOIs) to the controller software so that the controller can correctly interpret and parse data coming from the gateway module. This procedure is completed by right clicking the gateway block in the workspace and selecting the export AOI option, as shown in Figure 5.17.

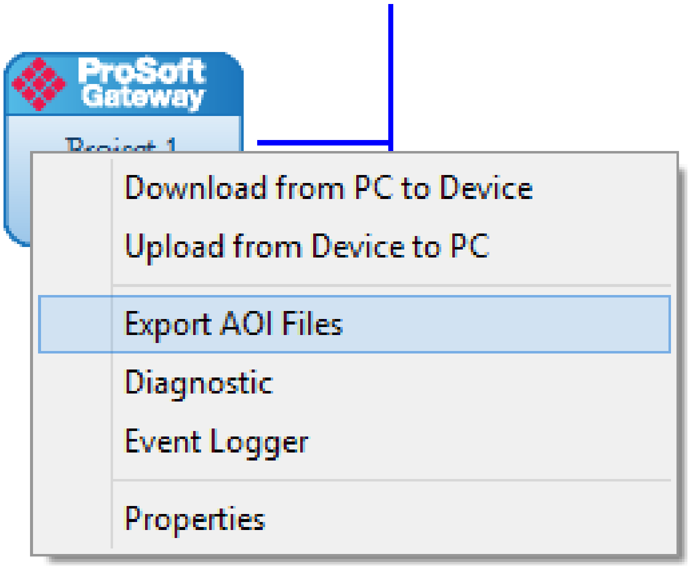
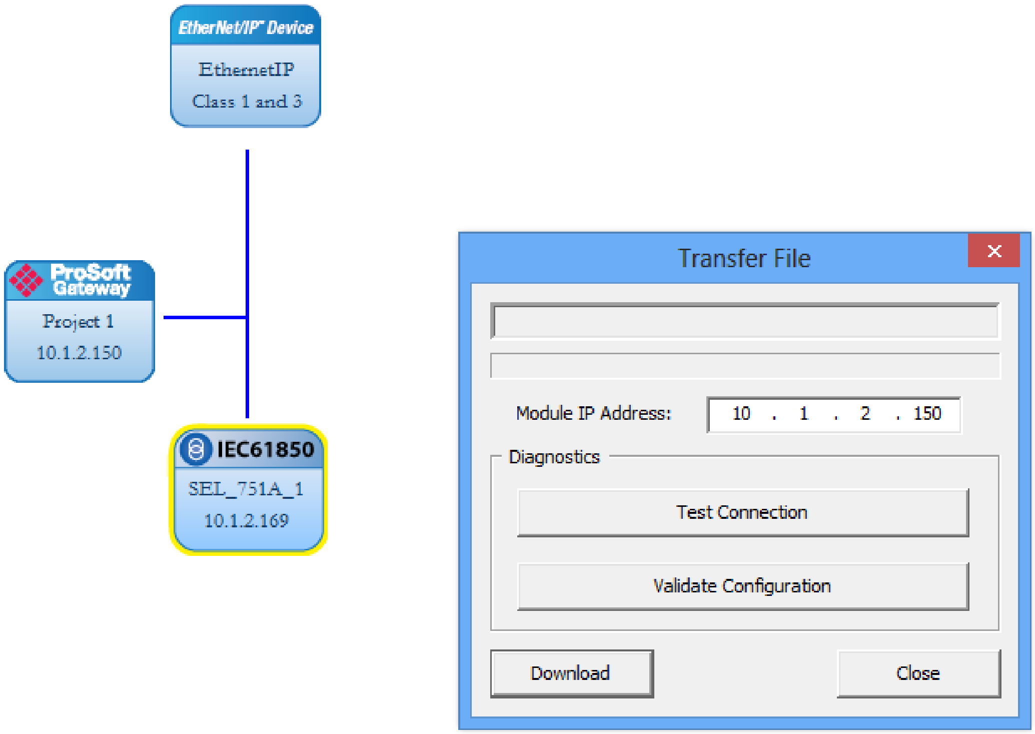


Figure 5.17 Exporting Add-On Instructions

Figure 5.18 shows the screens the user sees when he or she attempts to download the project and configuration to the gateway module. When the user is ready to download the configuration, the gateway block is selected, right clicked, and the download from PC to device option is selected. The transfer file window appears and displays the IP address of the module. The user should first select the test connection button to make sure that communication can be established with the module. After successful communication is established, the download button should be depressed. After the module downloads the configuration to the SD card, "*validate configuration*" should be selected to ensure that the configuration was downloaded successfully. At this point, the user has successfully configured both the IEDs and gateway module to send and receive information over the IEC 61850 standard.

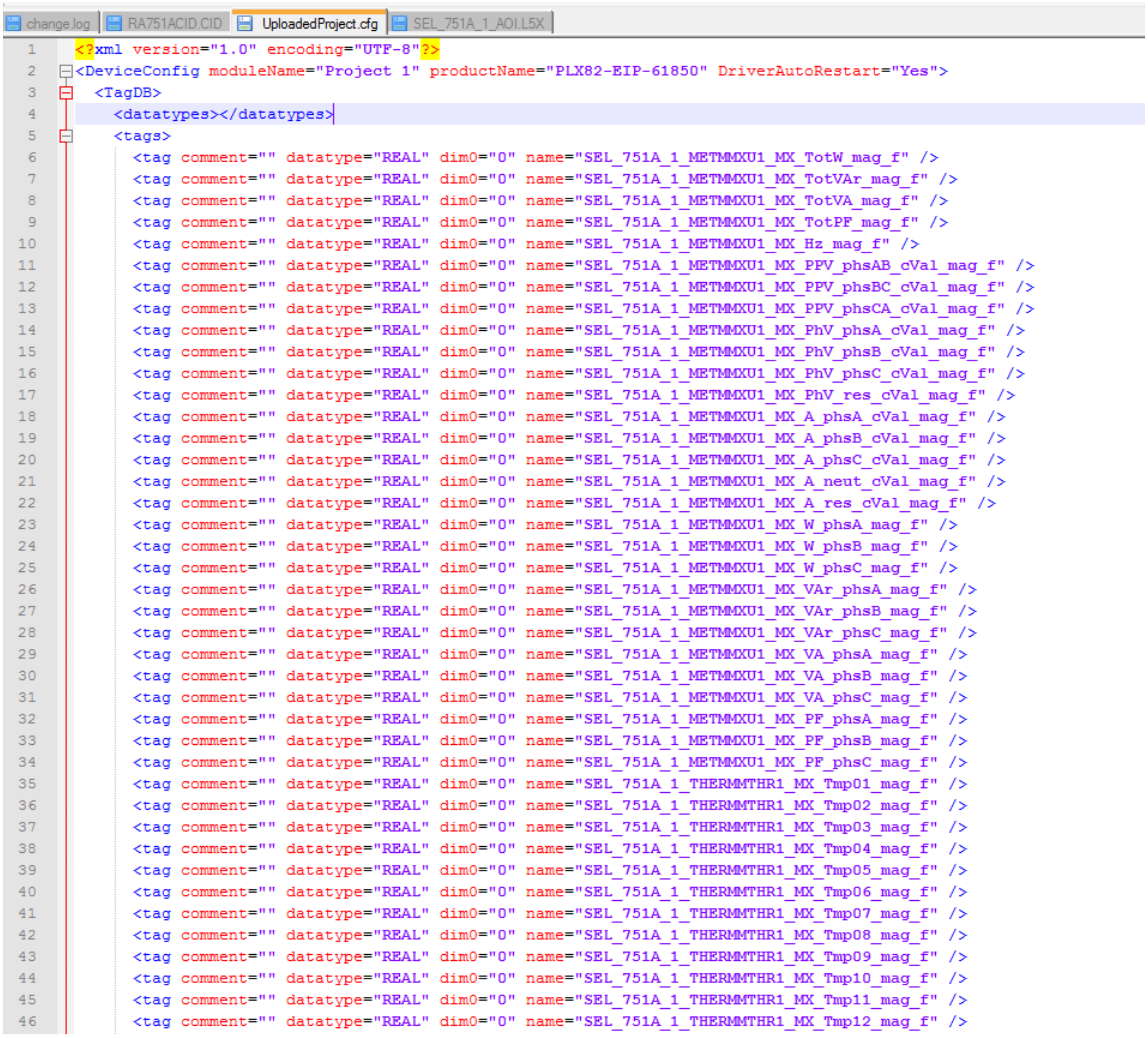


Module IP Address: 10 . 1 . 2 . 150

Figure 5.18 Downloading Configuration to Gateway Module

The Configuration File

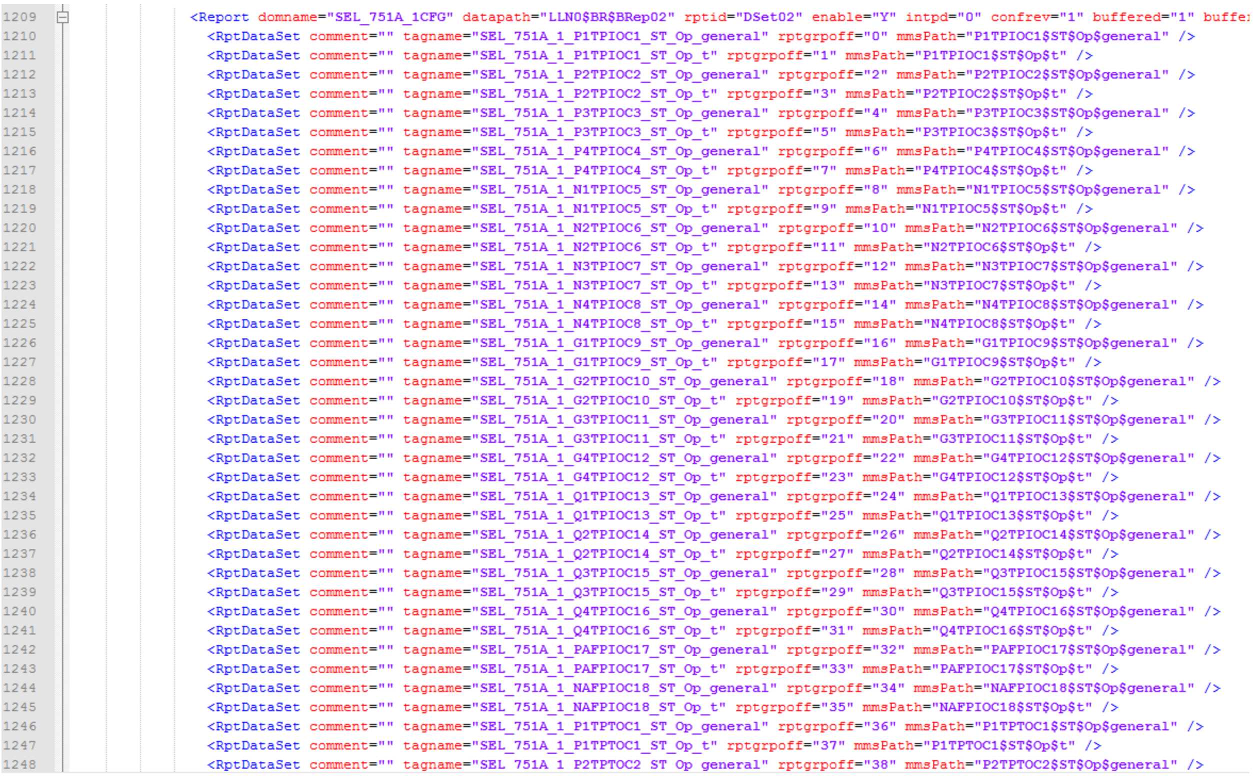
The software creates a running configuration file that is ultimately downloaded to the gateway module for every device added to the system. The process is transparent to the user. This configuration file can be seen in Figure 5.19. While Figure 5.19 shows only a small fragment of the configuration file, it does show the definition of the tag database based on the information mapped from the IEC 61850 CID files. As seen in the figure, the 751A feeder relay parameters that were mapped in the software configuration tool are shown in XML format as definitions of tags within the tag database (<TagDB>).



```xml
<?xml version="1.0" encoding="UTF-8"?>
<DeviceConfig moduleName="Project 1" productName="PLX82-EIP-61850" DriverAutoRestart="Yes">
  <TagDB>
    <datatypes></datatypes>
    <tags>
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_TotW_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_TotVAr_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_TotVA_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_TotPF_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_Hz_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_PPV_phsAB_cVal_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_PPV_phsBC_cVal_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_PPV_phsCA_cVal_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_PhV_phsA_cVal_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_PhV_phsB_cVal_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_PhV_phsC_cVal_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_PhV_res_cVal_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_A_phsA_cVal_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_A_phsB_cVal_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_A_phsC_cVal_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_A_neut_cVal_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_A_res_cVal_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_W_phsA_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_W_phsB_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_W_phsC_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_VAr_phsA_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_VAr_phsB_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_VAr_phsC_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_VA_phsA_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_VA_phsB_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_VA_phsC_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_PF_phsA_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_PF_phsB_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_METMMXU1_MX_PF_phsC_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_THERMMTHR1_MX_Tmp01_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_THERMMTHR1_MX_Tmp02_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_THERMMTHR1_MX_Tmp03_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_THERMMTHR1_MX_Tmp04_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_THERMMTHR1_MX_Tmp05_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_THERMMTHR1_MX_Tmp06_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_THERMMTHR1_MX_Tmp07_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_THERMMTHR1_MX_Tmp08_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_THERMMTHR1_MX_Tmp09_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_THERMMTHR1_MX_Tmp10_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_THERMMTHR1_MX_Tmp11_mag_f" />
      <tag comment="" datatype="REAL" dim0="0" name="SEL_751A_1_THERMMTHR1_MX_Tmp12_mag_f" />
```

Figure 5.19 Tag Database of Configuration File

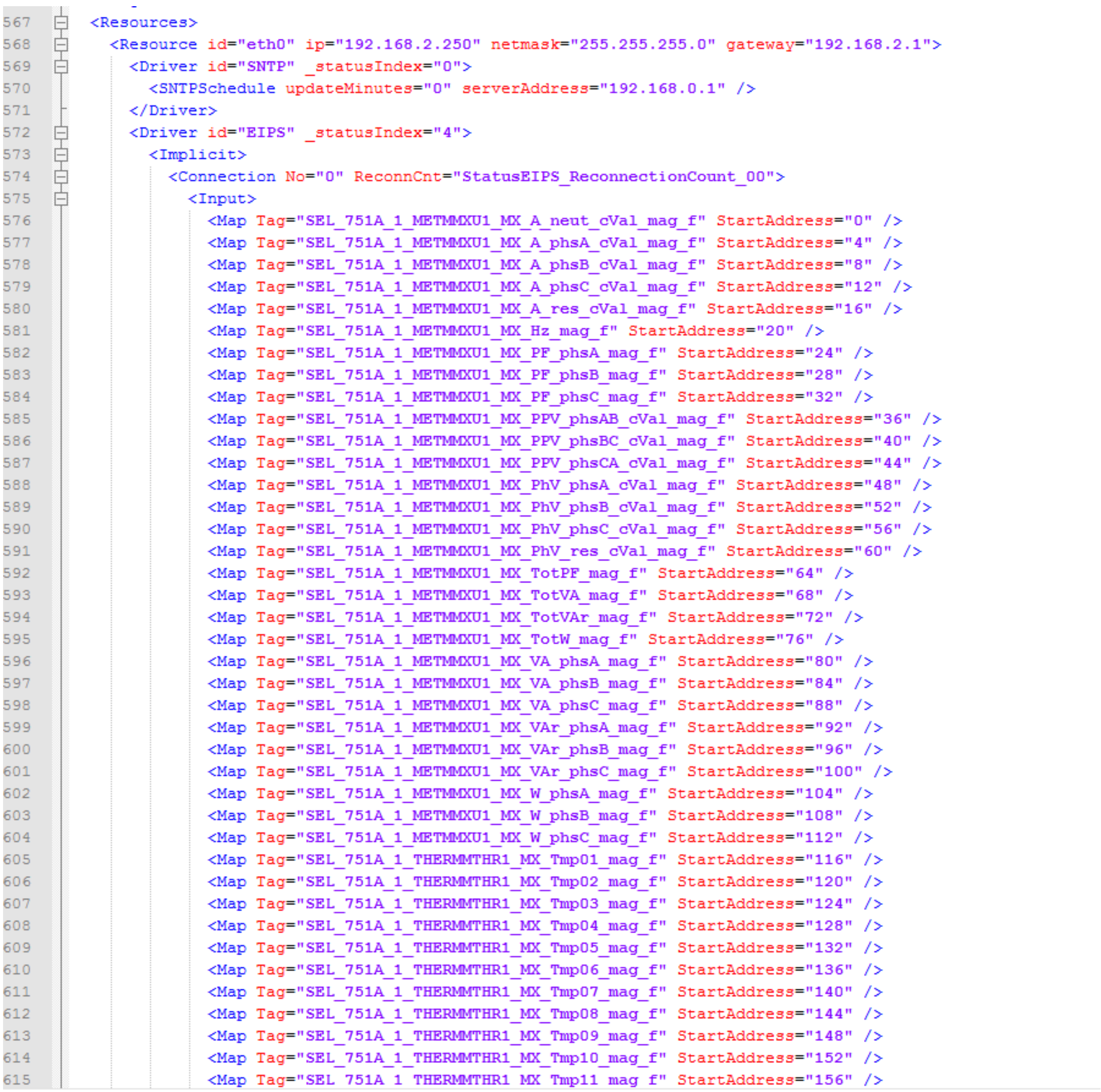
Figure 5.20 shows the portion of the configuration file where the reports were defined and selected by the configuration software. In this example, Buffered Report 2, as defined by the CID file and mapped by the configuration tool, is shown. This portion of the configuration file is passed to the IEC 61850 driver in order to specify the information it should look for. It also includes additional security information that allows a handshake to occur between the IED and gateway module before data transfer.



```
<Report domname="SEL_751A_1CFG" datapath="LLN0$BR$BRep02" rptid="DSet02" enable="Y" intpd="0" confrev="1" buffered="1" buffe
  <RptDataSet comment="" tagname="SEL_751A_1_P1TPIOC1_ST_Op_general" rptgrpoff="0" mmsPath="P1TPIOC1$ST$Op$general" />
  <RptDataSet comment="" tagname="SEL_751A_1_P1TPIOC1_ST_Op_t" rptgrpoff="1" mmsPath="P1TPIOC1$ST$Op$t" />
  <RptDataSet comment="" tagname="SEL_751A_1_P2TPIOC2_ST_Op_general" rptgrpoff="2" mmsPath="P2TPIOC2$ST$Op$general" />
  <RptDataSet comment="" tagname="SEL_751A_1_P2TPIOC2_ST_Op_t" rptgrpoff="3" mmsPath="P2TPIOC2$ST$Op$t" />
  <RptDataSet comment="" tagname="SEL_751A_1_P3TPIOC3_ST_Op_general" rptgrpoff="4" mmsPath="P3TPIOC3$ST$Op$general" />
  <RptDataSet comment="" tagname="SEL_751A_1_P3TPIOC3_ST_Op_t" rptgrpoff="5" mmsPath="P3TPIOC3$ST$Op$t" />
  <RptDataSet comment="" tagname="SEL_751A_1_P4TPIOC4_ST_Op_general" rptgrpoff="6" mmsPath="P4TPIOC4$ST$Op$general" />
  <RptDataSet comment="" tagname="SEL_751A_1_P4TPIOC4_ST_Op_t" rptgrpoff="7" mmsPath="P4TPIOC4$ST$Op$t" />
  <RptDataSet comment="" tagname="SEL_751A_1_N1TPIOC5_ST_Op_general" rptgrpoff="8" mmsPath="N1TPIOC5$ST$Op$general" />
  <RptDataSet comment="" tagname="SEL_751A_1_N1TPIOC5_ST_Op_t" rptgrpoff="9" mmsPath="N1TPIOC5$ST$Op$t" />
  <RptDataSet comment="" tagname="SEL_751A_1_N2TPIOC6_ST_Op_general" rptgrpoff="10" mmsPath="N2TPIOC6$ST$Op$general" />
  <RptDataSet comment="" tagname="SEL_751A_1_N2TPIOC6_ST_Op_t" rptgrpoff="11" mmsPath="N2TPIOC6$ST$Op$t" />
  <RptDataSet comment="" tagname="SEL_751A_1_N3TPIOC7_ST_Op_general" rptgrpoff="12" mmsPath="N3TPIOC7$ST$Op$general" />
  <RptDataSet comment="" tagname="SEL_751A_1_N3TPIOC7_ST_Op_t" rptgrpoff="13" mmsPath="N3TPIOC7$ST$Op$t" />
  <RptDataSet comment="" tagname="SEL_751A_1_N4TPIOC8_ST_Op_general" rptgrpoff="14" mmsPath="N4TPIOC8$ST$Op$general" />
  <RptDataSet comment="" tagname="SEL_751A_1_N4TPIOC8_ST_Op_t" rptgrpoff="15" mmsPath="N4TPIOC8$ST$Op$t" />
  <RptDataSet comment="" tagname="SEL_751A_1_G1TPIOC9_ST_Op_general" rptgrpoff="16" mmsPath="G1TPIOC9$ST$Op$general" />
  <RptDataSet comment="" tagname="SEL_751A_1_G1TPIOC9_ST_Op_t" rptgrpoff="17" mmsPath="G1TPIOC9$ST$Op$t" />
  <RptDataSet comment="" tagname="SEL_751A_1_G2TPIOC10_ST_Op_general" rptgrpoff="18" mmsPath="G2TPIOC10$ST$Op$general" />
  <RptDataSet comment="" tagname="SEL_751A_1_G2TPIOC10_ST_Op_t" rptgrpoff="19" mmsPath="G2TPIOC10$ST$Op$t" />
  <RptDataSet comment="" tagname="SEL_751A_1_G3TPIOC11_ST_Op_general" rptgrpoff="20" mmsPath="G3TPIOC11$ST$Op$general" />
  <RptDataSet comment="" tagname="SEL_751A_1_G3TPIOC11_ST_Op_t" rptgrpoff="21" mmsPath="G3TPIOC11$ST$Op$t" />
  <RptDataSet comment="" tagname="SEL_751A_1_G4TPIOC12_ST_Op_general" rptgrpoff="22" mmsPath="G4TPIOC12$ST$Op$general" />
  <RptDataSet comment="" tagname="SEL_751A_1_G4TPIOC12_ST_Op_t" rptgrpoff="23" mmsPath="G4TPIOC12$ST$Op$t" />
  <RptDataSet comment="" tagname="SEL_751A_1_Q1TPIOC13_ST_Op_general" rptgrpoff="24" mmsPath="Q1TPIOC13$ST$Op$general" />
  <RptDataSet comment="" tagname="SEL_751A_1_Q1TPIOC13_ST_Op_t" rptgrpoff="25" mmsPath="Q1TPIOC13$ST$Op$t" />
  <RptDataSet comment="" tagname="SEL_751A_1_Q2TPIOC14_ST_Op_general" rptgrpoff="26" mmsPath="Q2TPIOC14$ST$Op$general" />
  <RptDataSet comment="" tagname="SEL_751A_1_Q2TPIOC14_ST_Op_t" rptgrpoff="27" mmsPath="Q2TPIOC14$ST$Op$t" />
  <RptDataSet comment="" tagname="SEL_751A_1_Q3TPIOC15_ST_Op_general" rptgrpoff="28" mmsPath="Q3TPIOC15$ST$Op$general" />
  <RptDataSet comment="" tagname="SEL_751A_1_Q3TPIOC15_ST_Op_t" rptgrpoff="29" mmsPath="Q3TPIOC15$ST$Op$t" />
  <RptDataSet comment="" tagname="SEL_751A_1_Q4TPIOC16_ST_Op_general" rptgrpoff="30" mmsPath="Q4TPIOC16$ST$Op$general" />
  <RptDataSet comment="" tagname="SEL_751A_1_Q4TPIOC16_ST_Op_t" rptgrpoff="31" mmsPath="Q4TPIOC16$ST$Op$t" />
  <RptDataSet comment="" tagname="SEL_751A_1_PAFPIOC17_ST_Op_general" rptgrpoff="32" mmsPath="PAFPIOC17$ST$Op$general" />
  <RptDataSet comment="" tagname="SEL_751A_1_PAFPIOC17_ST_Op_t" rptgrpoff="33" mmsPath="PAFPIOC17$ST$Op$t" />
  <RptDataSet comment="" tagname="SEL_751A_1_NAFPIOC18_ST_Op_general" rptgrpoff="34" mmsPath="NAFPIOC18$ST$Op$general" />
  <RptDataSet comment="" tagname="SEL_751A_1_NAFPIOC18_ST_Op_t" rptgrpoff="35" mmsPath="NAFPIOC18$ST$Op$t" />
  <RptDataSet comment="" tagname="SEL_751A_1_P1TPTOC1_ST_Op_general" rptgrpoff="36" mmsPath="P1TPTOC1$ST$Op$general" />
  <RptDataSet comment="" tagname="SEL_751A_1_P1TPTOC1_ST_Op_t" rptgrpoff="37" mmsPath="P1TPTOC1$ST$Op$t" />
  <RptDataSet comment="" tagname="SEL_751A_1_P2TPTOC2_ST_Op_general" rptgrpoff="38" mmsPath="P2TPTOC2$ST$Op$general" />
```

Figure 5.20 IEC 61850 Configuration

Figure 5.21 shows the definition of mapping tags from the tag database to the EtherNet/IP driver for transfer to the automation controller. The information for each CIP connection, as defined by the configuration software, is defined as a resource to the EIPS schedule, which is then grabbed by the EtherNet/IP driver for publication to the controller. The user defined this mapping based on his or her selection in the EtherNet/IP mapping screen in the configuration software. It can be seen that the start address, in bytes, is defined for each data point.



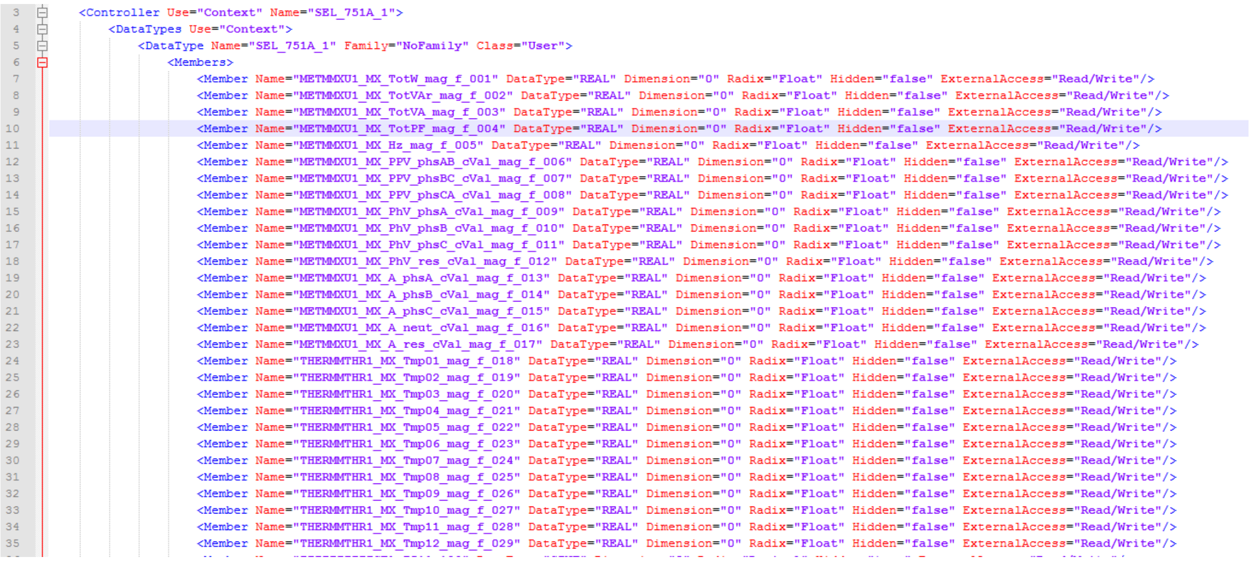
```
<Resources>
  <Resource id="eth0" ip="192.168.2.250" netmask="255.255.255.0" gateway="192.168.2.1">
    <Driver id="SNTP" _statusIndex="0">
      <SNTPSchedule updateMinutes="0" serverAddress="192.168.0.1" />
    </Driver>
    <Driver id="EIPS" _statusIndex="4">
      <Implicit>
        <Connection No="0" ReconnCnt="StatusEIPS_ReconnectionCount_00">
          <Input>
            <Map Tag="SEL_751A_1_METMMXU1_MX_A_neut_cVal_mag_f" StartAddress="0" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_A_phsA_cVal_mag_f" StartAddress="4" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_A_phsB_cVal_mag_f" StartAddress="8" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_A_phsC_cVal_mag_f" StartAddress="12" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_A_res_cVal_mag_f" StartAddress="16" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_Hz_mag_f" StartAddress="20" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_PF_phsA_mag_f" StartAddress="24" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_PF_phsB_mag_f" StartAddress="28" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_PF_phsC_mag_f" StartAddress="32" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_PPV_phsAB_cVal_mag_f" StartAddress="36" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_PPV_phsBC_cVal_mag_f" StartAddress="40" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_PPV_phsCA_cVal_mag_f" StartAddress="44" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_PhV_phsA_cVal_mag_f" StartAddress="48" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_PhV_phsB_cVal_mag_f" StartAddress="52" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_PhV_phsC_cVal_mag_f" StartAddress="56" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_PhV_res_cVal_mag_f" StartAddress="60" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_TotPF_mag_f" StartAddress="64" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_TotVA_mag_f" StartAddress="68" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_TotVAr_mag_f" StartAddress="72" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_TotW_mag_f" StartAddress="76" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_VA_phsA_mag_f" StartAddress="80" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_VA_phsB_mag_f" StartAddress="84" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_VA_phsC_mag_f" StartAddress="88" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_VAr_phsA_mag_f" StartAddress="92" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_VAr_phsB_mag_f" StartAddress="96" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_VAr_phsC_mag_f" StartAddress="100" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_W_phsA_mag_f" StartAddress="104" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_W_phsB_mag_f" StartAddress="108" />
            <Map Tag="SEL_751A_1_METMMXU1_MX_W_phsC_mag_f" StartAddress="112" />
            <Map Tag="SEL_751A_1_THERMMTHR1_MX_Tmp01_mag_f" StartAddress="116" />
            <Map Tag="SEL_751A_1_THERMMTHR1_MX_Tmp02_mag_f" StartAddress="120" />
            <Map Tag="SEL_751A_1_THERMMTHR1_MX_Tmp03_mag_f" StartAddress="124" />
            <Map Tag="SEL_751A_1_THERMMTHR1_MX_Tmp04_mag_f" StartAddress="128" />
            <Map Tag="SEL_751A_1_THERMMTHR1_MX_Tmp05_mag_f" StartAddress="132" />
            <Map Tag="SEL_751A_1_THERMMTHR1_MX_Tmp06_mag_f" StartAddress="136" />
            <Map Tag="SEL_751A_1_THERMMTHR1_MX_Tmp07_mag_f" StartAddress="140" />
            <Map Tag="SEL_751A_1_THERMMTHR1_MX_Tmp08_mag_f" StartAddress="144" />
            <Map Tag="SEL_751A_1_THERMMTHR1_MX_Tmp09_mag_f" StartAddress="148" />
            <Map Tag="SEL_751A_1_THERMMTHR1_MX_Tmp10_mag_f" StartAddress="152" />
            <Map Tag="SEL 751A 1 THERMMTHR1 MX Tmp11 mag f" StartAddress="156" />
```

Figure 5.21 EtherNet/IP Definition

Add-On Instruction

As mentioned in the introduction, the Add-On instruction defines the data that is sent by the gateway to the controller and determines how that data is sent over EtherNet/IP CIP connections. A screen shot of the code behind the AOI is shown in Figure 5.22. There are two components to the AOI: the User-Defined Data Type (UDT) and information mapping. In

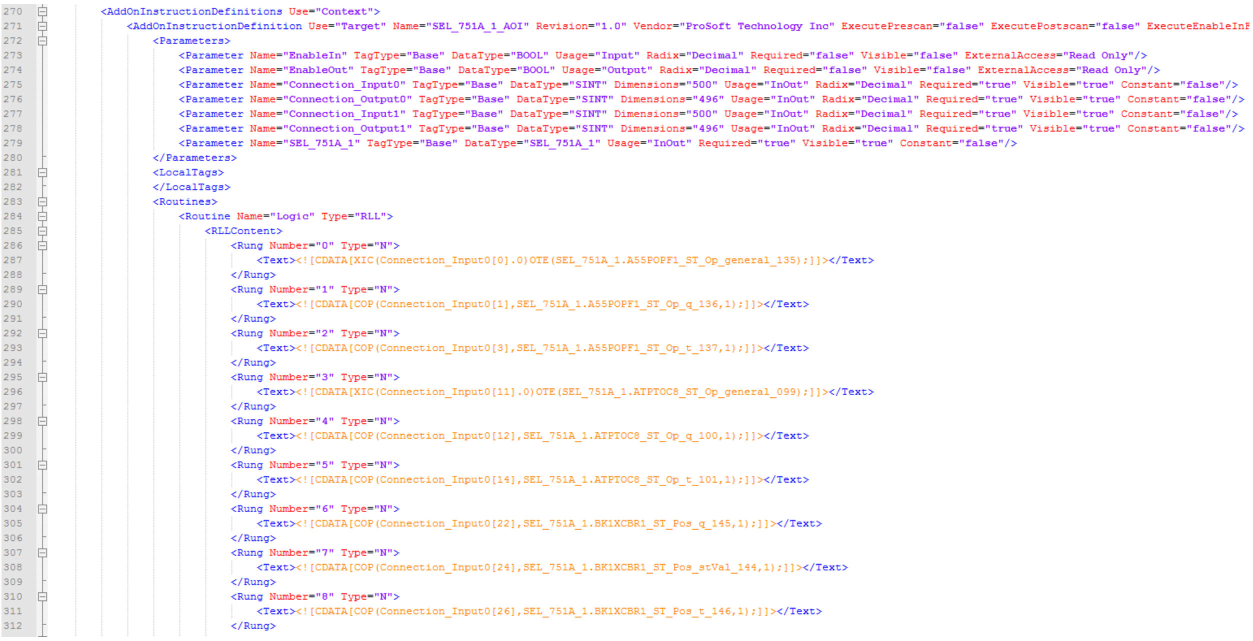
Figure 5.22, the object getting mapped to the automation controller is the 751A feeder relay, defined in Line 3. The UDT name is defined as SEL_751A_1, as seen in Line 5. In Lines 7 through 35 (and beyond), the members of the object are defined. The UDT is an object with multiple attributes, including members and modifying functions. Members are those attributes associated with the 751A feeder relay. For this particular example, the UDT contains the same mapped IEC 61850 information that have appeared in screens/or figures already presented in this chapter.



```
3      <Controller Use="Context" Name="SEL_751A_1">
4          <DataTypes Use="Context">
5              <DataType Name="SEL_751A_1" Family="NoFamily" Class="User">
6                  <Members>
7                      <Member Name="METMMXU1_MX_TotW_mag_f_001" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
8                      <Member Name="METMMXU1_MX_TotVAr_mag_f_002" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
9                      <Member Name="METMMXU1_MX_TotVA_mag_f_003" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
10                     <Member Name="METMMXU1_MX_TotPF_mag_f_004" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
11                     <Member Name="METMMXU1_MX_Hz_mag_f_005" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
12                     <Member Name="METMMXU1_MX_PPV_phsAB_cVal_mag_f_006" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
13                     <Member Name="METMMXU1_MX_PPV_phsBC_cVal_mag_f_007" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
14                     <Member Name="METMMXU1_MX_PPV_phsCA_cVal_mag_f_008" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
15                     <Member Name="METMMXU1_MX_PhV_phsA_cVal_mag_f_009" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
16                     <Member Name="METMMXU1_MX_PhV_phsB_cVal_mag_f_010" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
17                     <Member Name="METMMXU1_MX_PhV_phsC_cVal_mag_f_011" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
18                     <Member Name="METMMXU1_MX_PhV_res_cVal_mag_f_012" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
19                     <Member Name="METMMXU1_MX_A_phsA_cVal_mag_f_013" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
20                     <Member Name="METMMXU1_MX_A_phsB_cVal_mag_f_014" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
21                     <Member Name="METMMXU1_MX_A_phsC_cVal_mag_f_015" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
22                     <Member Name="METMMXU1_MX_A_neut_cVal_mag_f_016" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
23                     <Member Name="METMMXU1_MX_A_res_cVal_mag_f_017" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
24                     <Member Name="THERMMTHR1_MX_Tmp01_mag_f_018" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
25                     <Member Name="THERMMTHR1_MX_Tmp02_mag_f_019" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
26                     <Member Name="THERMMTHR1_MX_Tmp03_mag_f_020" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
27                     <Member Name="THERMMTHR1_MX_Tmp04_mag_f_021" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
28                     <Member Name="THERMMTHR1_MX_Tmp05_mag_f_022" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
29                     <Member Name="THERMMTHR1_MX_Tmp06_mag_f_023" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
30                     <Member Name="THERMMTHR1_MX_Tmp07_mag_f_024" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
31                     <Member Name="THERMMTHR1_MX_Tmp08_mag_f_025" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
32                     <Member Name="THERMMTHR1_MX_Tmp09_mag_f_026" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
33                     <Member Name="THERMMTHR1_MX_Tmp10_mag_f_027" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
34                     <Member Name="THERMMTHR1_MX_Tmp11_mag_f_028" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
35                     <Member Name="THERMMTHR1_MX_Tmp12_mag_f_029" DataType="REAL" Dimension="0" Radix="Float" Hidden="false" ExternalAccess="Read/Write"/>
```

Figure 5.22 User-Defined Data Type Definition

The later portion of the AOI file that is generated by the configuration software includes a ladder routing that is imported into the automation controller code. The AOI defines how the data is scanned by the controller and the data tables containing the information from the IEDs are updated. The script can be seen in Figure 5.23. Lines 270-282 define the scope of the UDT and what connections populate the data table. The <Routine> section of the code defines the ladder logic which individually scans each object of the data stream, parse it, and copy its value to the corresponding tag in the controller data table. The figure shows that each rung has its own rung number, and the <text> field in each line defines the logic used to parse and copy the data.

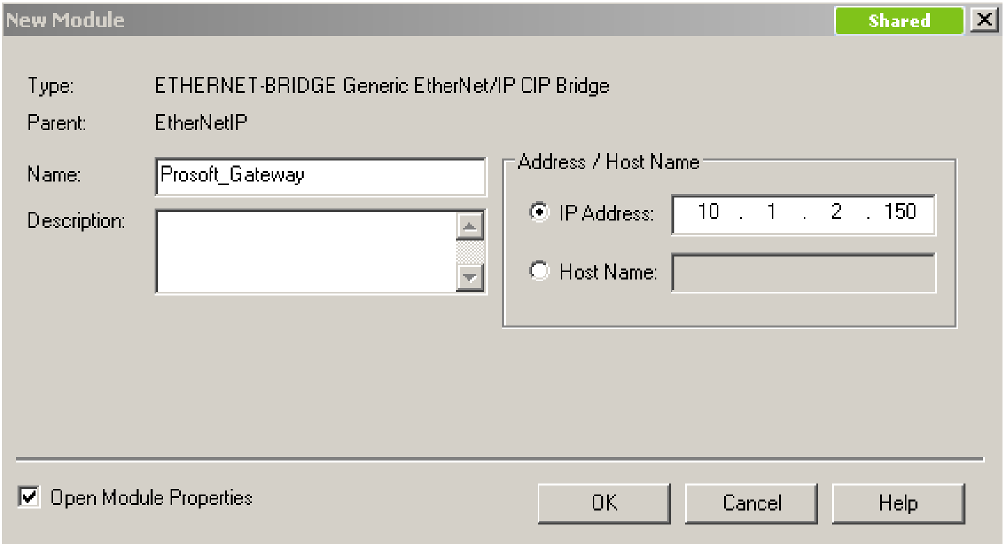


```
<AddOnInstructionDefinitions Use="Context">
    <AddOnInstructionDefinition Use="Target" Name="SEL_751A_1_AOI" Revision="1.0" Vendor="ProSoft Technology Inc" ExecutePrescan="false" ExecutePostscan="false" ExecuteEnableInF
        <Parameters>
            <Parameter Name="EnableIn" TagType="Base" DataType="BOOL" Usage="Input" Radix="Decimal" Required="false" Visible="false" ExternalAccess="Read Only"/>
            <Parameter Name="EnableOut" TagType="Base" DataType="BOOL" Usage="Output" Radix="Decimal" Required="false" Visible="false" ExternalAccess="Read Only"/>
            <Parameter Name="Connection_Input0" TagType="Base" DataType="SINT" Dimensions="500" Usage="InOut" Radix="Decimal" Required="true" Visible="true" Constant="false"/>
            <Parameter Name="Connection_Output0" TagType="Base" DataType="SINT" Dimensions="496" Usage="InOut" Radix="Decimal" Required="true" Visible="true" Constant="false"/>
            <Parameter Name="Connection_Input1" TagType="Base" DataType="SINT" Dimensions="500" Usage="InOut" Radix="Decimal" Required="true" Visible="true" Constant="false"/>
            <Parameter Name="Connection_Output1" TagType="Base" DataType="SINT" Dimensions="496" Usage="InOut" Radix="Decimal" Required="true" Visible="true" Constant="false"/>
            <Parameter Name="SEL_751A_1" TagType="Base" DataType="SEL_751A_1" Usage="InOut" Required="true" Visible="true" Constant="false"/>
        </Parameters>
        <LocalTags>
        </LocalTags>
        <Routines>
            <Routine Name="Logic" Type="RLL">
                <RLLContent>
                    <Rung Number="0" Type="N">
                        <Text><![CDATA[XIC(Connection_Input0[0].0)OTE(SEL_751A_1.A55POPF1_ST_Op_general_135);]]></Text>
                    </Rung>
                    <Rung Number="1" Type="N">
                        <Text><![CDATA[COP(Connection_Input0[1],SEL_751A_1.A55POPF1_ST_Op_q_136,1);]]></Text>
                    </Rung>
                    <Rung Number="2" Type="N">
                        <Text><![CDATA[COP(Connection_Input0[3],SEL_751A_1.A55POPF1_ST_Op_t_137,1);]]></Text>
                    </Rung>
                    <Rung Number="3" Type="N">
                        <Text><![CDATA[XIC(Connection_Input0[11].0)OTE(SEL_751A_1.ATPTOC8_ST_Op_general_099);]]></Text>
                    </Rung>
                    <Rung Number="4" Type="N">
                        <Text><![CDATA[COP(Connection_Input0[12],SEL_751A_1.ATPTOC8_ST_Op_q_100,1);]]></Text>
                    </Rung>
                    <Rung Number="5" Type="N">
                        <Text><![CDATA[COP(Connection_Input0[14],SEL_751A_1.ATPTOC8_ST_Op_t_101,1);]]></Text>
                    </Rung>
                    <Rung Number="6" Type="N">
                        <Text><![CDATA[COP(Connection_Input0[22],SEL_751A_1.BK1XCBR1_ST_Pos_q_145,1);]]></Text>
                    </Rung>
                    <Rung Number="7" Type="N">
                        <Text><![CDATA[COP(Connection_Input0[24],SEL_751A_1.BK1XCBR1_ST_Pos_stVal_144,1);]]></Text>
                    </Rung>
                    <Rung Number="8" Type="N">
                        <Text><![CDATA[COP(Connection_Input0[26],SEL_751A_1.BK1XCBR1_ST_Pos_t_146,1);]]></Text>
                    </Rung>
```

Figure 5.23 AOI Ladder Definition

Importing Code to RS Logix

The last step in configuring the automation and control system to read data from the gateway module is to configure the controller to interpret information being delivered by the gateway module. The gateway is defined as a Generic CIP Bridge, which is shown in Figure 5.24. The IP address of the gateway is set to correspond to that of the gateway module.



Type: ETHERNET-BRIDGE Generic EtherNet/IP CIP Bridge

Parent: EtherNetIP

Name: Prosoft_Gateway

Description:

Address / Host Name

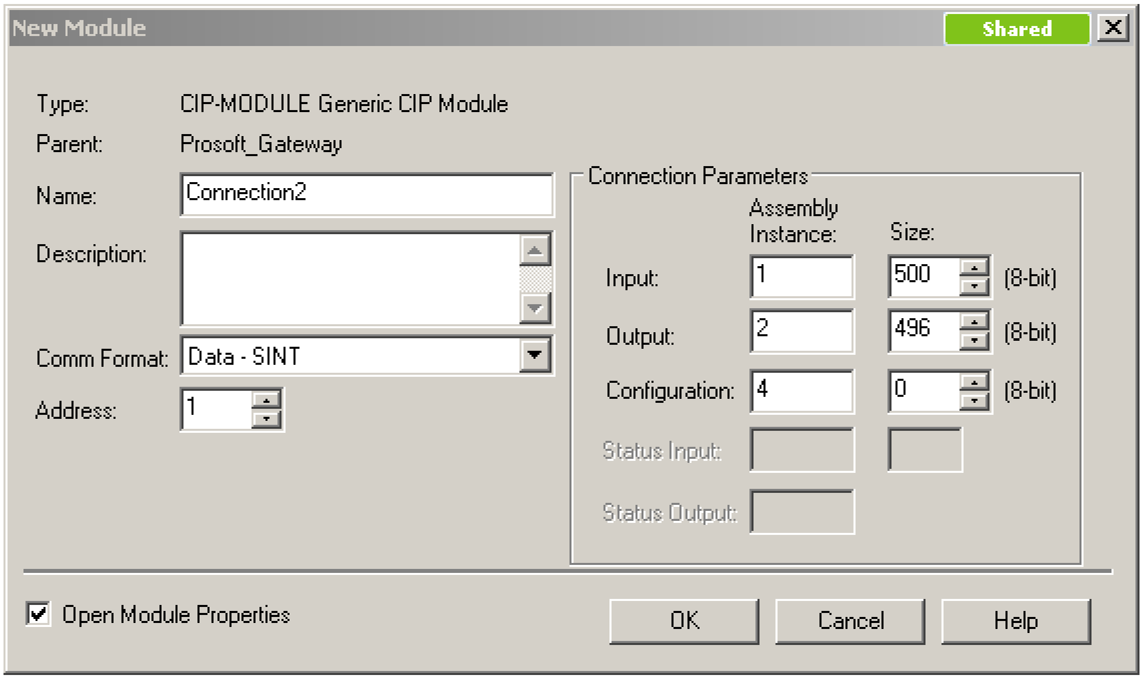
● IP Address: 10 . 1 . 2 . 150

○ Host Name:

☑ Open Module Properties

Figure 5.24 Configuring the Gateway

Depending on the number of CIP connections required to move all of the data, the user needs to define the corresponding number of CIP connections in software. Figure 5.25 shows the process of defining the CIP connections in the controller software environment. Data is defined as Short Integers (SINTS), or bytes, with each connection containing 500 bytes of input data, 496 bytes of output data, and a configuration header. Figure 5.26 shows a completed configuration example, with the gateway module defined with two associated Class 1 CIP connections.



New Module

Shared

Type: CIP-MODULE Generic CIP Module

Parent: Prosoft_Gateway

Name: Connection2

Description:

Comm Format: Data - SINT

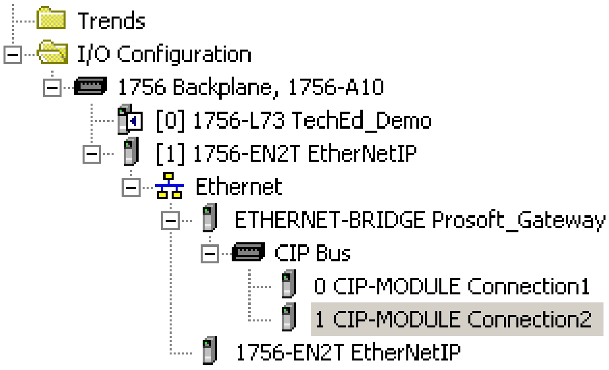
Address: 1

Connection Parameters

	Assembly Instance:	Size:	
Input:	1	500	(8-bit)
Output:	2	496	(8-bit)
Configuration:	4	0	(8-bit)
Status Input:			
Status Output:			

☑ Open Module Properties

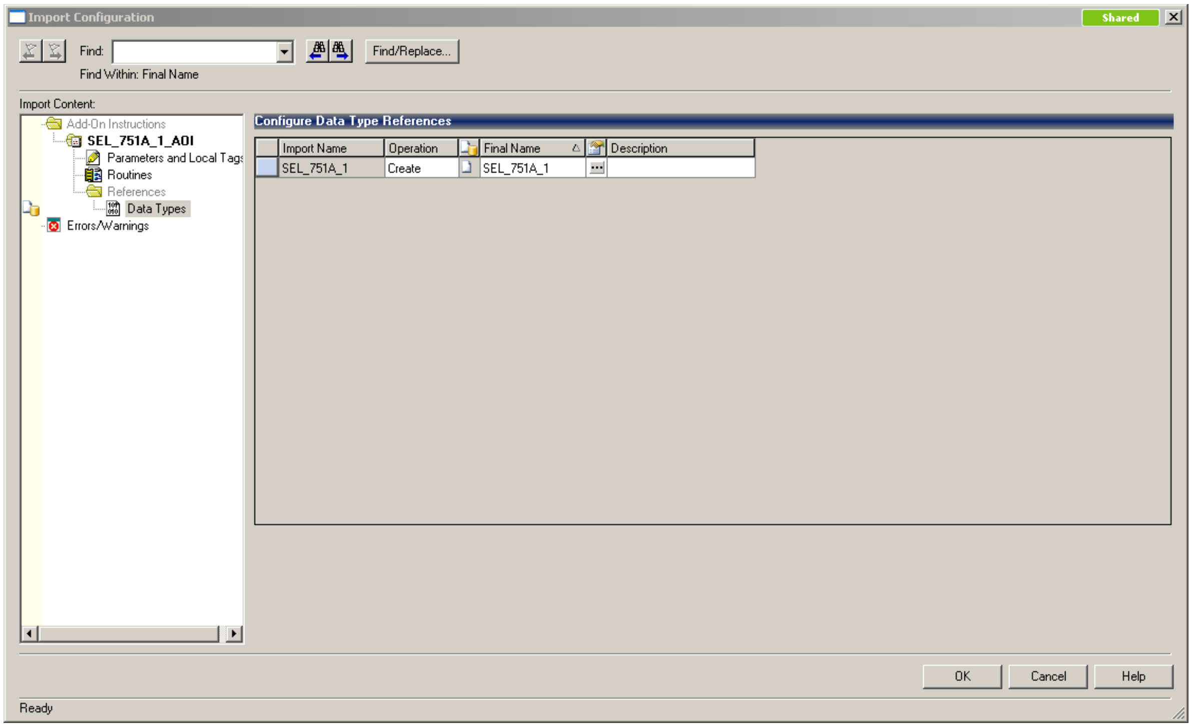
Figure 5.25 Defining CIP Connections



```
Trends
I/O Configuration
  1756 Backplane, 1756-A10
    [0] 1756-L73 TechEd_Demo
    [1] 1756-EN2T EtherNetIP
      Ethernet
        ETHERNET-BRIDGE Prosoft_Gateway
          CIP Bus
            0 CIP-MODULE Connection1
            1 CIP-MODULE Connection2
        1756-EN2T EtherNetIP
```

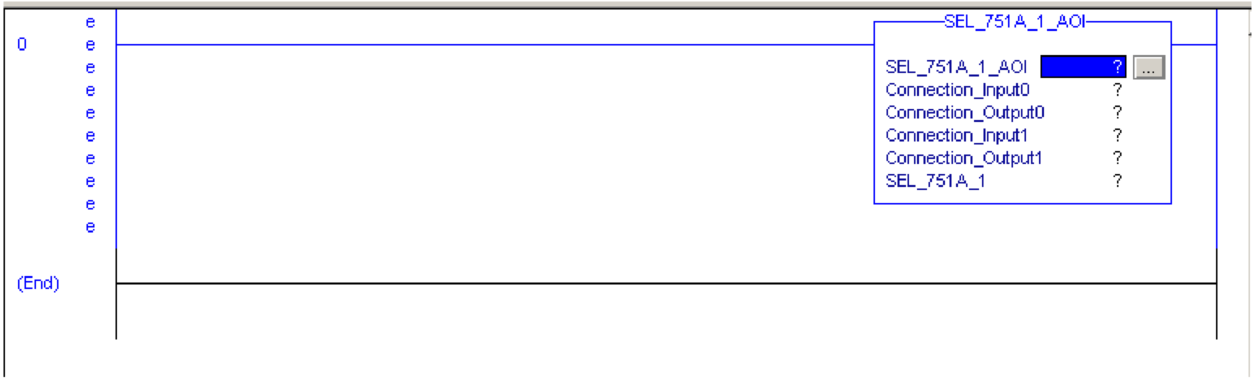
Figure 5.26 Gateway Module in I/O Tree

After defining the gateway module as a generic CIP bridge, the user imports the AOI, which was described and depicted in Figures 5.22 and 5.23. The XML script written in these files runs in the background as a wizard that creates the linkages and an instruction block which can be imported into ladder code. The wizard can be seen in Figure 5.27, while the AOI can be seen in Figure 5.28.



Import Name	Operation	Final Name	Description
SEL_751A_1	Create	SEL_751A_1	

Figure 5.27 AOI Import Wizard

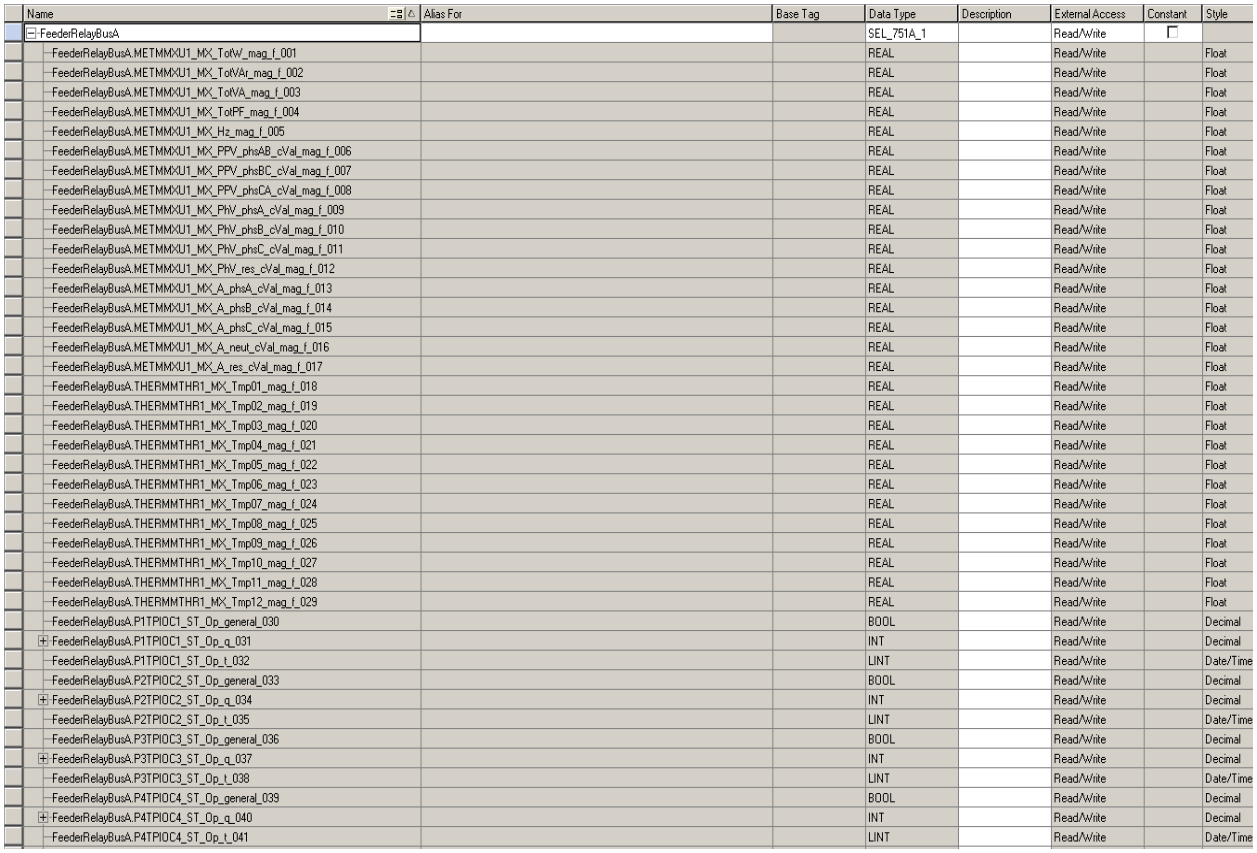


```
SEL_751A_1_AOI          ?
Connection_Input0       ?
Connection_Output0      ?
Connection_Input1       ?
Connection_Output1      ?
SEL_751A_1              ?
```

Figure 5.28 AOI in Ladder Code

The first input to the AOI asks for a user-defined name. The connection parameters ask the user to point to the corresponding input and output arrays of data as defined by each created CIP connection. That last input of the AOI is the name that the user wants to give the IED, for example “*FeederRelay*.”

Subsequently, this AOI is inserted into the automation controller code, and the parameters are populated. The data table of the controller is now populated with information from corresponding IEDs, as shown in Figure 5.29. Note that the IEC 61850 tags that were originally mapped from the example in the beginning of this chapter are now displayed in the controller data table. It should also be noted that each data type and naming convention is preserved between the IED network and the process control network.



Name	Alias For	Base Tag	Data Type	Description	External Access	Constant	Style
⊟ FeederRelayBusA			SEL_751A_1		Read/Write	□	
FeederRelayBusA.METMMXU1_MX_TotW_mag_f_001			REAL		Read/Write		Float
FeederRelayBusA.METMMXU1_MX_TotVAr_mag_f_002			REAL		Read/Write		Float
FeederRelayBusA.METMMXU1_MX_TotVA_mag_f_003			REAL		Read/Write		Float
FeederRelayBusA.METMMXU1_MX_TotPF_mag_f_004			REAL		Read/Write		Float
FeederRelayBusA.METMMXU1_MX_Hz_mag_f_005			REAL		Read/Write		Float
FeederRelayBusA.METMMXU1_MX_PPV_phsAB_cVal_mag_f_006			REAL		Read/Write		Float
FeederRelayBusA.METMMXU1_MX_PPV_phsBC_cVal_mag_f_007			REAL		Read/Write		Float
FeederRelayBusA.METMMXU1_MX_PPV_phsCA_cVal_mag_f_008			REAL		Read/Write		Float
FeederRelayBusA.METMMXU1_MX_PhV_phsA_cVal_mag_f_009			REAL		Read/Write		Float
FeederRelayBusA.METMMXU1_MX_PhV_phsB_cVal_mag_f_010			REAL		Read/Write		Float
FeederRelayBusA.METMMXU1_MX_PhV_phsC_cVal_mag_f_011			REAL		Read/Write		Float
FeederRelayBusA.METMMXU1_MX_PhV_res_cVal_mag_f_012			REAL		Read/Write		Float
FeederRelayBusA.METMMXU1_MX_A_phsA_cVal_mag_f_013			REAL		Read/Write		Float
FeederRelayBusA.METMMXU1_MX_A_phsB_cVal_mag_f_014			REAL		Read/Write		Float
FeederRelayBusA.METMMXU1_MX_A_phsC_cVal_mag_f_015			REAL		Read/Write		Float
FeederRelayBusA.METMMXU1_MX_A_neut_cVal_mag_f_016			REAL		Read/Write		Float
FeederRelayBusA.METMMXU1_MX_A_res_cVal_mag_f_017			REAL		Read/Write		Float
FeederRelayBusA.THERMMTHR1_MX_Tmp01_mag_f_018			REAL		Read/Write		Float
FeederRelayBusA.THERMMTHR1_MX_Tmp02_mag_f_019			REAL		Read/Write		Float
FeederRelayBusA.THERMMTHR1_MX_Tmp03_mag_f_020			REAL		Read/Write		Float
FeederRelayBusA.THERMMTHR1_MX_Tmp04_mag_f_021			REAL		Read/Write		Float
FeederRelayBusA.THERMMTHR1_MX_Tmp05_mag_f_022			REAL		Read/Write		Float
FeederRelayBusA.THERMMTHR1_MX_Tmp06_mag_f_023			REAL		Read/Write		Float
FeederRelayBusA.THERMMTHR1_MX_Tmp07_mag_f_024			REAL		Read/Write		Float
FeederRelayBusA.THERMMTHR1_MX_Tmp08_mag_f_025			REAL		Read/Write		Float
FeederRelayBusA.THERMMTHR1_MX_Tmp09_mag_f_026			REAL		Read/Write		Float
FeederRelayBusA.THERMMTHR1_MX_Tmp10_mag_f_027			REAL		Read/Write		Float
FeederRelayBusA.THERMMTHR1_MX_Tmp11_mag_f_028			REAL		Read/Write		Float
FeederRelayBusA.THERMMTHR1_MX_Tmp12_mag_f_029			REAL		Read/Write		Float
FeederRelayBusA.P1TPIOC1_ST_Op_general_030			BOOL		Read/Write		Decimal
⊞ FeederRelayBusA.P1TPIOC1_ST_Op_q_031			INT		Read/Write		Decimal
FeederRelayBusA.P1TPIOC1_ST_Op_t_032			LINT		Read/Write		Date/Time
FeederRelayBusA.P2TPIOC2_ST_Op_general_033			BOOL		Read/Write		Decimal
⊞ FeederRelayBusA.P2TPIOC2_ST_Op_q_034			INT		Read/Write		Decimal
FeederRelayBusA.P2TPIOC2_ST_Op_t_035			LINT		Read/Write		Date/Time
FeederRelayBusA.P3TPIOC3_ST_Op_general_036			BOOL		Read/Write		Decimal
⊞ FeederRelayBusA.P3TPIOC3_ST_Op_q_037			INT		Read/Write		Decimal
FeederRelayBusA.P3TPIOC3_ST_Op_t_038			LINT		Read/Write		Date/Time
FeederRelayBusA.P4TPIOC4_ST_Op_general_039			BOOL		Read/Write		Decimal
⊞ FeederRelayBusA.P4TPIOC4_ST_Op_q_040			INT		Read/Write		Decimal
FeederRelayBusA.P4TPIOC4_ST_Op_t_041			LINT		Read/Write		Date/Time

Figure 5.29 Controller Data Table

Addressing Process Control Historian Issues

Traditional process historian consists of four major components: controller, data server, interface node, and historian repository. Typically, the event timestamp is applied at the interface node. Interface nodes collect, interrogate, and qualify information provided by the data server. If the data values collected at the interface node exceed pre-defined dead band thresholds, a timestamp is applied and the data is transferred to the historian repository. If the data does not exceed the threshold, the data point is disregarded and not archived. This process is better known as *exception testing* and can be seen in Figure 5.30.

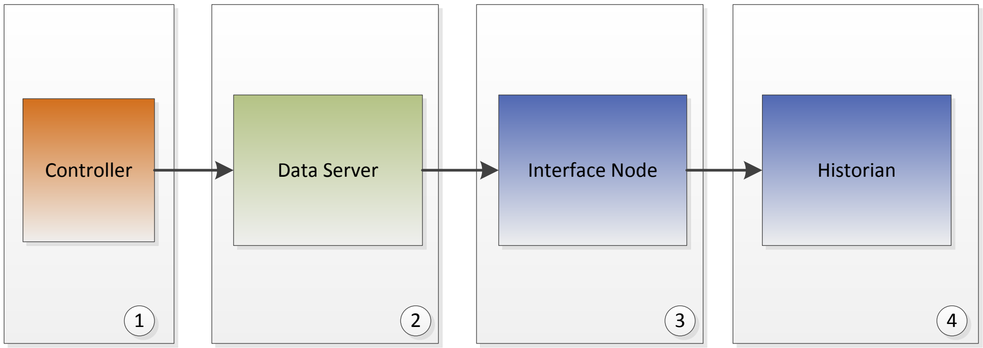


Figure 5.30 Traditional Data Collection Scheme

This procedure for collecting and timestamping data does not work well for events that are time stamped at the source device. In the above example, the event that is passed to the interface node is value based, not time driven. After it has been determined that an exception has occurred, a timestamp is generated and applied to the event. For example, if a process engineer is trying to log and store events from IEDs, i.e., trips, alarms, etc., the timestamp of these events is generated by the IED. Using value-based exception reporting will not suffice in transferring meaningful timestamps from the IED to the historian repository. Referring back to Figure 5.30, the controller does not provide a timestamp to the data server, but only a process value. For most manufacturing environments, this schema is acceptable as time delay error is within the poll rate of the data server.

In an IED system with IEC 61850 communications, the event drives both value and timestamp changes in the data collection system. Both the event data and timestamp are created at the IED and must travel together from the source of the event to the repository to be meaningful. The previous two statements are the root cause of the problem in attempting to store electrical distribution values in a process control data historian. The block diagram that describes the solution to this problem can be seen in Figure 5.31.

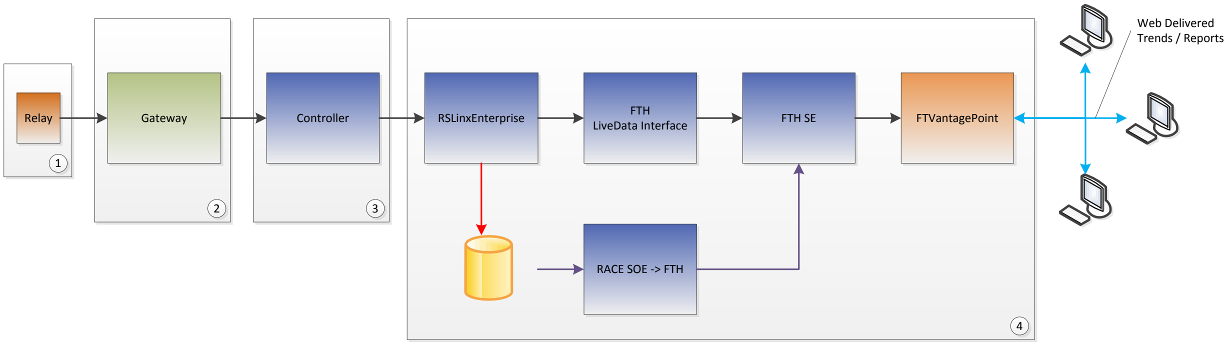


Figure 5.31 Proposed Software Solution

Figure 5.31 shows the progression of data through the block diagram system. Block one defines the physical IED. The IED(s) is (are) connected to the gateway module via an Ethernet connection. Additionally, the gateway module is connected to the controller represented by the Controller block. The timestamp that is generated on the event by the IED is passed through the gateway to the controller, as shown by the progression from Boxes 1-3. Although the controller data table contains both the event value and IED timestamp, it is not inherently passed to the historian repository. In order to persist an event with an historian tag, the configuration of an ALMD (Digital Alarm) instruction associates the event to the point ID of the historian tag. An historian point ID is a unique identifier for a tag in the historian repository. When an alarm occurs, the following parameters are known: timestamp of IED, unique point ID, timestamp of entering alarm, value of alarm, and the Event Association ID. An example of this logic is shown in Figure 5.32.

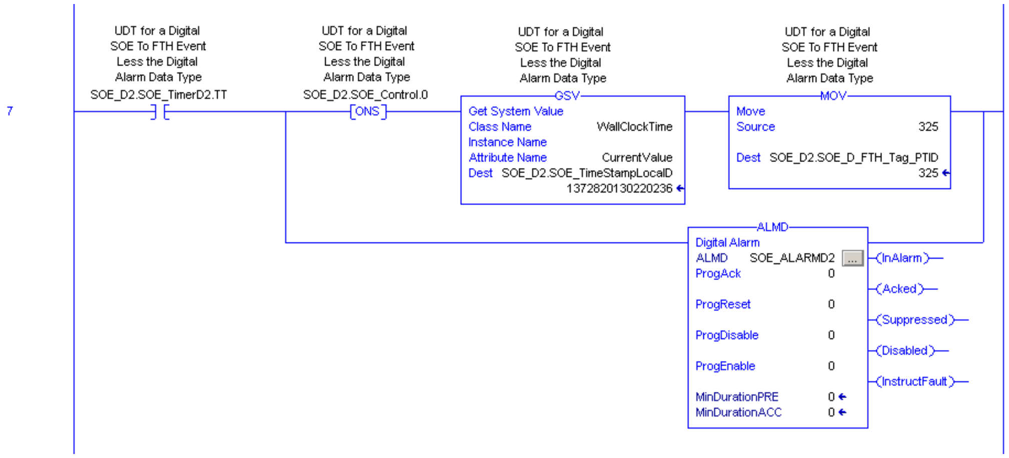
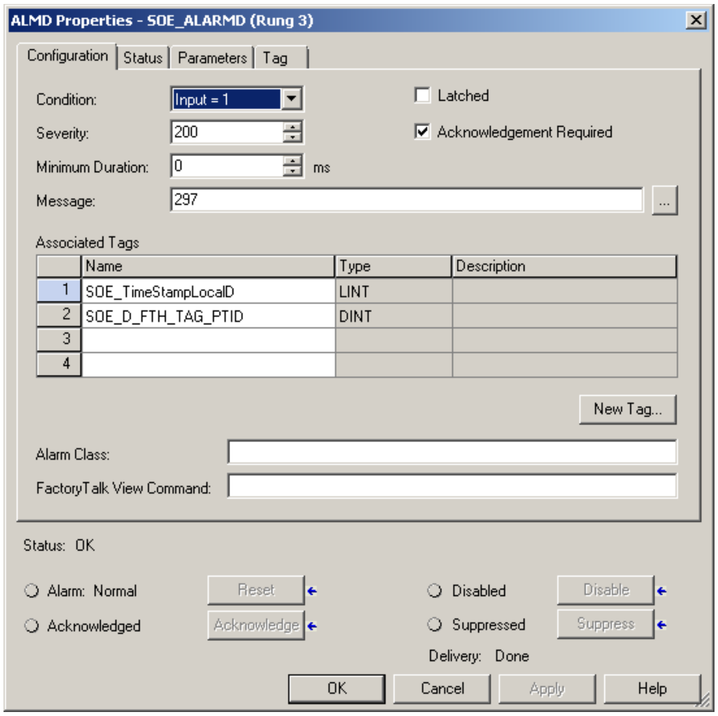

7
UDT for a Digital SOE To FTH Event Less the Digital Alarm Data Type
SOE_D2.SOE_TimerD2.TT
UDT for a Digital SOE To FTH Event Less the Digital Alarm Data Type
SOE_D2.SOE_Control.0
ONS
UDT for a Digital SOE To FTH Event Less the Digital Alarm Data Type
GSV
Get System Value
Class Name WallClockTime
Instance Name
Attribute Name CurrentValue
Dest SOE_D2.SOE_TimeStampLocalD
1372820130220236
UDT for a Digital SOE To FTH Event Less the Digital Alarm Data Type
MOV
Move
Source 325
Dest SOE_D2.SOE_D_FTH_Tag_PTID
325
ALMD
Digital Alarm
ALMD SOE_ALARMD2
ProgAck 0
ProgReset 0
ProgDisable 0
ProgEnable 0
MinDurationPRE 0
MinDurationACC 0
InAlarm
Acked
Suppressed
Disabled
InstructFault

Figure 5.32 Example Timestamp Logic ALMD

Figure 5.33 shows the configuration of the ALMD instruction with the associated IED timestamp (1) and point ID (2). It is important to note that the severity is set to 200 (default 500), in order to provide the control system a way to differentiate tags as electrical distribution tags. The data type of the time stamp is that of Long Integer (LINT), which is a 64-bit representation of the number of microseconds from January 1, 1970, also known as the Unix Epoch.



Condition: Input = 1

□ Latched

Severity: 200

☑ Acknowledgement Required

Minimum Duration: 0 ms

Message: 297

Associated Tags

	Name	Type	Description
1	SOE_TimeStampLocalD	LINT	
2	SOE_D_FTH_TAG_PTID	DINT	
3			
4			

Alarm Class:

FactoryTalk View Command:

Status: OK

○ Alarm: Normal

○ Acknowledged

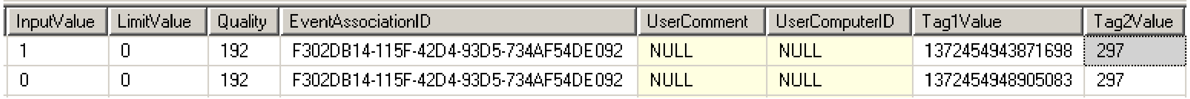
○ Disabled

○ Suppressed

Delivery: Done

Figure 5.33 ALMD Configuration

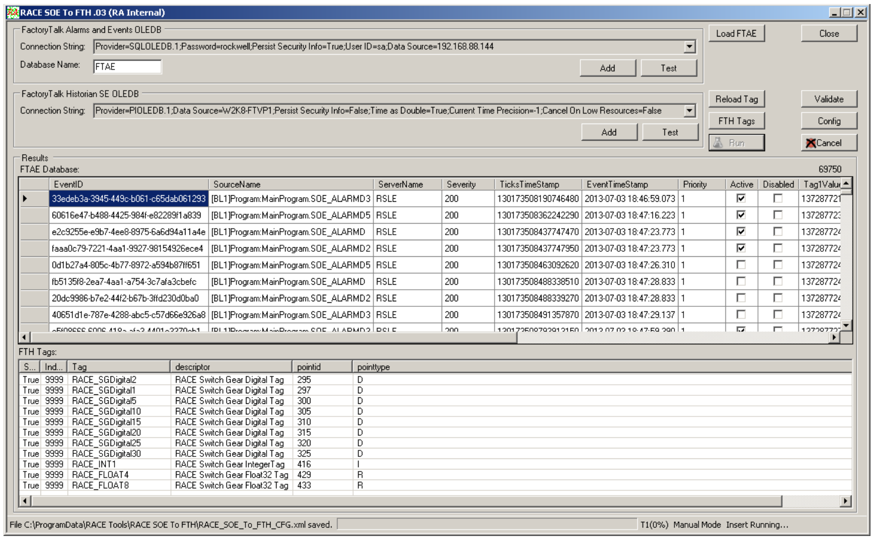
When the alarm is triggered, a message is sent to the Alarm and Events (A&E) data server (RS Linx Enterprise) as depicted in 5.31. RS Linx Enterprise then populates the ConditionEvent Table in the A&E database (red arrow). Figure 5.34 shows an example of this table. The ConditionEvent Table shows the InputValue, EventAssociationID, Tag1Value (IED timestamp), and Tag2Value (point ID). Figure 5.34 also shows the off/on and on/off transition of an alarm event, represented by EventAssociationIDs being equal.



InputValue	LimitValue	Quality	EventAssociationID	UserComment	UserComputerID	Tag1Value	Tag2Value
1	0	192	F302DB14-115F-42D4-93D5-734AF54DE092	NULL	NULL	1372454943871698	297
0	0	192	F302DB14-115F-42D4-93D5-734AF54DE092	NULL	NULL	1372454948905083	297

Figure 5.34 ConditionEvent Table (Partial)

In Figure 5.31, the top path in Block 4 represents the traditional data flow within the process historian system. The solution developed for this research takes an alternative approach to providing data to the historian repository, as represented by the lower path through the RACE SOE / FTH tool. This GUI tool is shown in Figure 5.35. The GUI program was written using the .NET framework. The Results workgroup represents the two collections: A&E records and historian tags. Connections to both the A&E database, as well as the process historian, are defined in OLEDB connection strings.



RACE SOE To FTH .03 (RA Internal)

FactoryTalk Alarms and Events OLEDB

Connection String: Provider=SQLOLEDB.1;Password=rockwell;Persist Security Info=True;User ID=sa;Data Source=192.168.88.144

Database Name: FTAE

FactoryTalk Historian SE OLEDB

Connection String: Provider=PIOLEDB.1;Data Source=W2K8-FTVP1;Persist Security Info=False;Time as Double=True;Current Time Precision=-1;Cancel On Low Resources=False

Results

FTAE Database: 69750

	EventID	SourceName	ServerName	Severity	TicksTimeStamp	EventTimeStamp	Priority	Active	Disabled	Tag1Value
▶	33edeb3a-3945-449c-b061-c65dab061293	[BL1]Program:MainProgram.SOE_ALARMD3	RSLE	200	130173508190746480	2013-07-03 18:46:59.073	1	☑	☐	137287721
	60616e47-b488-4425-984f-e82289f1a839	[BL1]Program:MainProgram.SOE_ALARMD5	RSLE	200	130173508362242290	2013-07-03 18:47:16.223	1	☑	☐	13728772
	e2c9255e-e9b7-4ee8-8975-6a6d94a11a4e	[BL1]Program:MainProgram.SOE_ALARMD	RSLE	200	130173508437747470	2013-07-03 18:47:23.773	1	☑	☐	13728772
	faaa0c79-7221-4aa1-9927-98154926ece4	[BL1]Program:MainProgram.SOE_ALARMD2	RSLE	200	130173508437747950	2013-07-03 18:47:23.773	1	☑	☐	13728772
	0d1b27a4-805c-4b77-8972-a594b87ff651	[BL1]Program:MainProgram.SOE_ALARMD5	RSLE	200	130173508463092620	2013-07-03 18:47:26.310	1	☐	☐	13728772
	fb5135f8-2ea7-4aa1-a754-3c7afa3cbefc	[BL1]Program:MainProgram.SOE_ALARMD	RSLE	200	130173508488338510	2013-07-03 18:47:28.833	1	☐	☐	13728772
	20dc9986-b7e2-44f2-b67b-3ffd230d0ba0	[BL1]Program:MainProgram.SOE_ALARMD2	RSLE	200	130173508488339270	2013-07-03 18:47:28.833	1	☐	☐	13728772
	40651d1e-787e-4288-abc5-c57d66e926a8	[BL1]Program:MainProgram.SOE_ALARMD3	RSLE	200	130173508491357870	2013-07-03 18:47:29.137	1	☐	☐	13728772

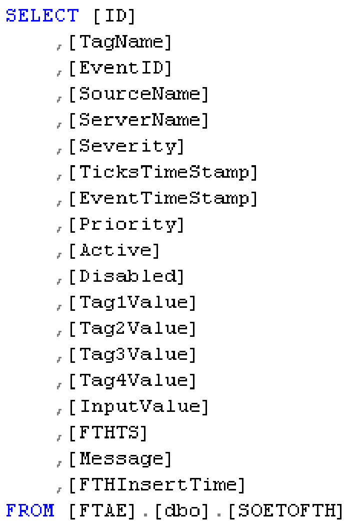
FTH Tags:

S...	Ind...	Tag	descriptor	pointid	pointtype
True	9999	RACE_SGDigital2	RACE Switch Gear Digital Tag	295	D
True	9999	RACE_SGDigital1	RACE Switch Gear Digital Tag	297	D
True	9999	RACE_SGDigital5	RACE Switch Gear Digital Tag	300	D
True	9999	RACE_SGDigital10	RACE Switch Gear Digital Tag	305	D
True	9999	RACE_SGDigital15	RACE Switch Gear Digital Tag	310	D
True	9999	RACE_SGDigital20	RACE Switch Gear Digital Tag	315	D
True	9999	RACE_SGDigital25	RACE Switch Gear Digital Tag	320	D
True	9999	RACE_SGDigital30	RACE Switch Gear Digital Tag	325	D
True	9999	RACE_INT1	RACE Switch Gear IntegerTag	416	I
True	9999	RACE_FLOAT4	RACE Switch Gear Float32 Tag	429	R
True	9999	RACE_FLOAT8	RACE Switch Gear Float32 Tag	433	R

File C:\ProgramData\RACE Tools\RACE SOE To FTH\RACE_SOE_To_FTH_CFG.xml saved. T1(0%) Manual Mode Insert Running...

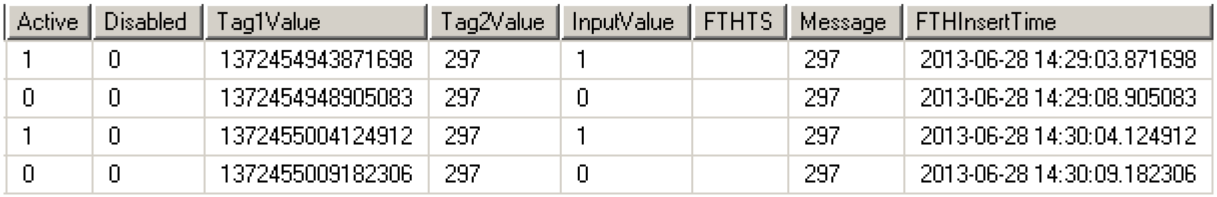
Figure 5.35 RACE GUI Tool

Now that an event with point ID 297 has been captured in the A&E database, the information, including the associated IED timestamp, needs to be persisted to the historian tag. This is accomplished by the user with an insert query to the process historian. After the insert is successful, an artifact record is written to an additional table in the FT A&E database called SOEtoFTH. Figure 5.36 defines the query to display information in the SOEtoFTH table. The result of this query can be seen in Figure 5.37.



```
SELECT [ID]
    ,[TagName]
    ,[EventID]
    ,[SourceName]
    ,[ServerName]
    ,[Severity]
    ,[TicksTimeStamp]
    ,[EventTimeStamp]
    ,[Priority]
    ,[Active]
    ,[Disabled]
    ,[Tag1Value]
    ,[Tag2Value]
    ,[Tag3Value]
    ,[Tag4Value]
    ,[InputValue]
    ,[FTHTS]
    ,[Message]
    ,[FTHInsertTime]
FROM [FTAE].[dbo].[SOETOFTH]
```

Figure 5.36 Query of SOEtoFTH

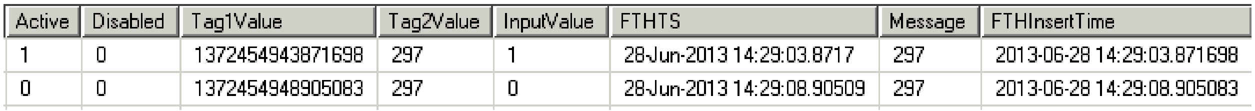


Active	Disabled	Tag1Value	Tag2Value	InputValue	FTHTS	Message	FTHInsertTime
1	0	1372454943871698	297	1		297	2013-06-28 14:29:03.871698
0	0	1372454948905083	297	0		297	2013-06-28 14:29:08.905083
1	0	1372455004124912	297	1		297	2013-06-28 14:30:04.124912
0	0	1372455009182306	297	0		297	2013-06-28 14:30:09.182306

Figure 5.37 Query Results (Partial)

Note that Active is the input state of the alarm instruction (ALMD). Tag1Value is the time stamp from the IED, Tag2Value is the point ID of the historian tag, and FTHTS is the Factory Talk Historian Time Stamp. The FTHInsertTime is the time used in the insert to the historian. The FTHTS and FTHInsertTime can be up to +/- 15 microseconds, as this is the highest resolution that this timestamp object can represent.

Figure 5.38 shows the complete run of the GUI tool with the FTHTS column populated. The difference between the IED time and Historian time for the first two events are 2 and 7 microseconds, respectively, well within the error bars of the defined system.



Active	Disabled	Tag1Value	Tag2Value	InputValue	FTHTS	Message	FTHInsertTime
1	0	1372454943871698	297	1	28-Jun-2013 14:29:03.8717	297	2013-06-28 14:29:03.871698
0	0	1372454948905083	297	0	28-Jun-2013 14:29:08.90509	297	2013-06-28 14:29:08.905083

Figure 5.38 Populated Query Results (Partial)

The logic behind how the software tool functions is shown in Figure 5.39. This tool, much like the hardware, is multithreaded so that it does not burden the machine upon which it is installed.

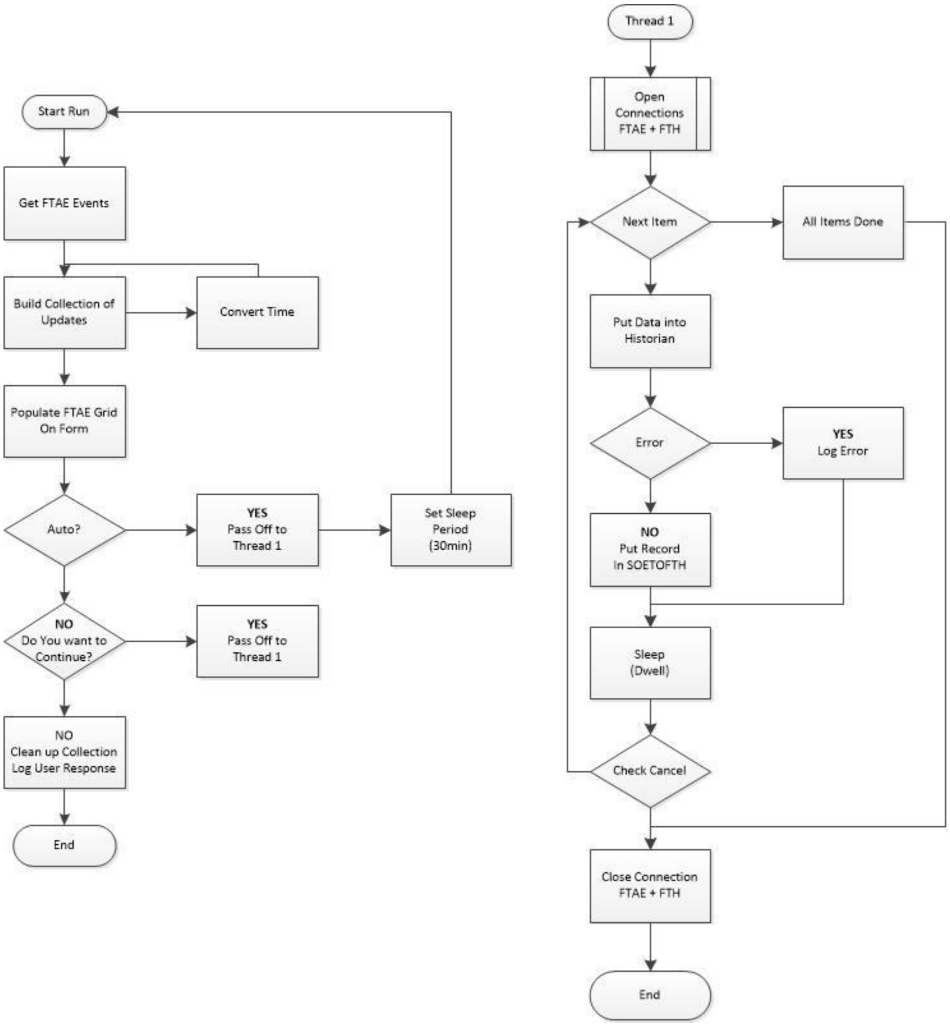


Figure 5.39 Historian Tool Algorithm

Figure 5.40 shows the flow of information created by the development of the software discussed in this chapter. As described, information moves from IEDs in the field through the gateway module to the controller. From the controller, these values can then be inserted into historian and trended in reports on view screens for visualization and reporting.

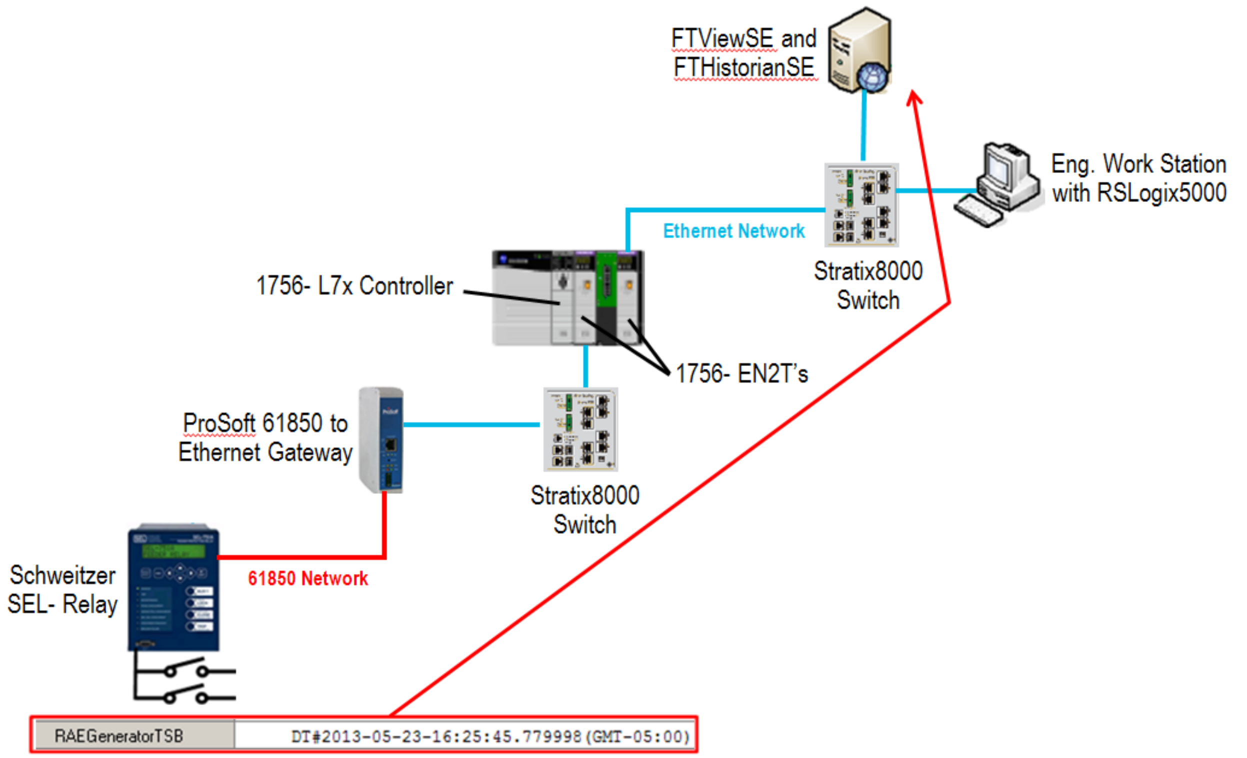


Figure 5.40 Practical Data Flow Diagram

Summary

In conclusion, this chapter has discussed software developments resulting from this research that provide a functional, user-friendly solution to what program the gateway module to convert IEC 61850 messages to EtherNet/IP messages. This chapter was broken down into two sections: software created to interface with the gateway module, and software created to move information to a process historian. Software developed to interface to the gateway module reads CID files and allows the user to map information that is required by the automation controller. After this information is defined as tags, these tags are mapped to corresponding EtherNet/IP tags according to the ODVA standard. Once this mapping has occurred, the module configuration is downloaded to the module and Add-on instruction is? generated so that the data stream can be interpreted by the controller. Once the Add-on instruction is imported by the controller, information is correctly parsed from the data stream and inserted into the controller data table.

The historian software reads information from the Alarms and Events software database and inserts the corresponding timestamp in post processing to the process historian within 15 microseconds. Once the data is in the automation controller from hardware and software configurations, the remainder of this developed solution is to provide operators and engineers visual aids that enhance command and control of a mine process control system.

Chapter 6 - Visualization

Introduction

This chapter defines the visualization developed for this research. Items discussed in this chapter include: faceplate definition, human machine interface discussion and definition, and data management solution. The motivation for visualization is to represent an IED in logic and graphics as close to the physical object itself. This provides the operator or engineer with the same look and feel experience they would have if interacting with the physical device. At the same time, the graphics were developed with various levels of security to allow only users with proper credentials access to various command and control functions. Additionally, the research solution must adhere to various graphics standards for both power and process control systems.

Faceplate Solution

Process control companies have developed solutions for process visualization. This research takes process visualization one step further into the electrical distribution system. The IEC 61850 standard allows for the visual representation of reports and alarms at the process control level. With the development of more sophisticated Human Machine Interface (HMI) screens, global objects have been introduced into the automation graphics. The use of these global objects has allowed for the creation of faceplates.

A faceplate is defined as a reusable standard object. The advantage of the faceplate is that it is a standard, prebuilt object that can be implemented repeatedly. Each faceplate has security levels built into the objects themselves. These security features can be customized, based upon user and application requirements. A prime example of the security benefit is the command and control of IEDs.

The command and control portion of this research allows operators and engineers to change settings and allow relay (or other IED) actuation from a remote site. Even though the IEC 61850 standard defines how an IED communicates on an IEC 61850 network, many people do not realize that these files do not include configuration information for the protection and control functions in an IED. The IEC 61850 standard also does not provide any uniform method for designing communication-assisted automation. Using faceplates as a solution addresses both of these concerns, offering operation's personnel a standard methodology not only to gather information from the system, but to control it as well.

Global Objects and Images

The faceplates developed for this research were constructed using standard ISA global objects. These symbols are widely accepted by the process and power community, with each graphic having a standard definition. Figure 6.1 shows the standard global objects used in the creation of faceplates. It can be seen that the control tab is defined by standard home, configuration, engineering, trending, and various alarm buttons. Tabs are defined to comply with ISA standards with respect to the grey shading and definitions when certain pages are selected.

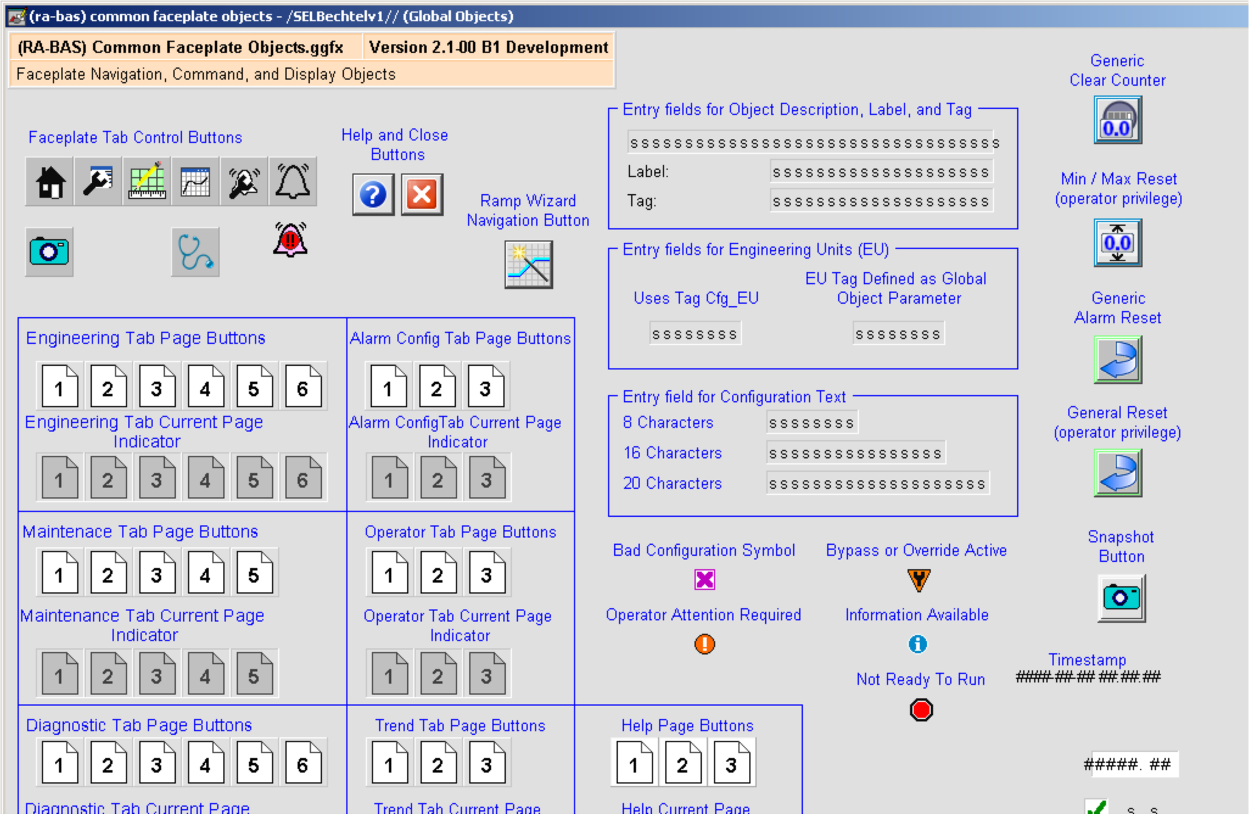

(ra-bas) common faceplate objects - /SELBechtelv1// (Global Objects)
(RA-BAS) Common Faceplate Objects.ggfx
Version 2.1-00 B1 Development
Faceplate Navigation, Command, and Display Objects
Faceplate Tab Control Buttons
Help and Close Buttons
Ramp Wizard Navigation Button
Entry fields for Object Description, Label, and Tag
ssssssssssssssssssssssssssssssss
Label:
ssssssssssssssssssss
Tag:
ssssssssssssssssssss
Entry fields for Engineering Units (EU)
Uses Tag Cfg_EU
EU Tag Defined as Global Object Parameter
ssssssss
ssssssss
Entry field for Configuration Text
8 Characters
ssssssss
16 Characters
ssssssssssssssss
20 Characters
ssssssssssssssssssss
Generic Clear Counter
0.0
Min / Max Reset (operator privilege)
0.0
Generic Alarm Reset
General Reset (operator privilege)
Snapshot Button
Engineering Tab Page Buttons
1 2 3 4 5 6
Engineering Tab Current Page Indicator
1 2 3 4 5 6
Alarm Config Tab Page Buttons
1 2 3
Alarm ConfigTab Current Page Indicator
1 2 3
Maintenace Tab Page Buttons
1 2 3 4 5
Maintenance Tab Current Page Indicator
1 2 3 4 5
Operator Tab Page Buttons
1 2 3
Operator Tab Current Page Indicator
1 2 3
Bad Configuration Symbol
Bypass or Override Active
Operator Attention Required
Information Available
Not Ready To Run
Timestamp
####-##-## ##:##:##
Diagnostic Tab Page Buttons
1 2 3 4 5 6
Diagnostic Tab Current Page
Trend Tab Page Buttons
1 2 3
Trend Tab Current Page
Help Page Buttons
1 2 3
Help Current Page
#####. ##

Figure 6.1 Global Objects

Additional global objects were created in this research for various operations of IEDs, to include open/close breakers, start/stop rotating machinery, lock/unlock relay options, target reset, select, and synchronize. These symbols are shown in Figure 6.2. The images used for the faceplates match National Electrical Manufactures Association (NEMA) standards. The color schema adheres to that of the IEEE C37.1 definition of Supervisory Control and Data Acquisition (SCADA) systems. The color schema differ from traditional process control systems in that red usually references a stop or removal of power operation. However, in distribution systems, red implies an application of power to terminals.

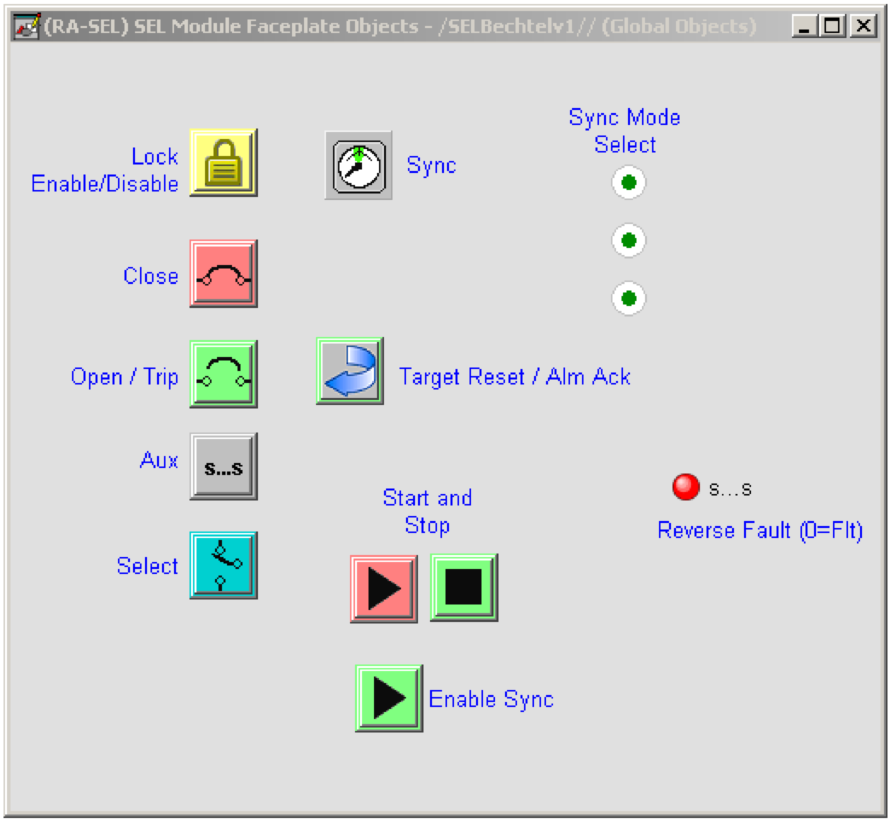
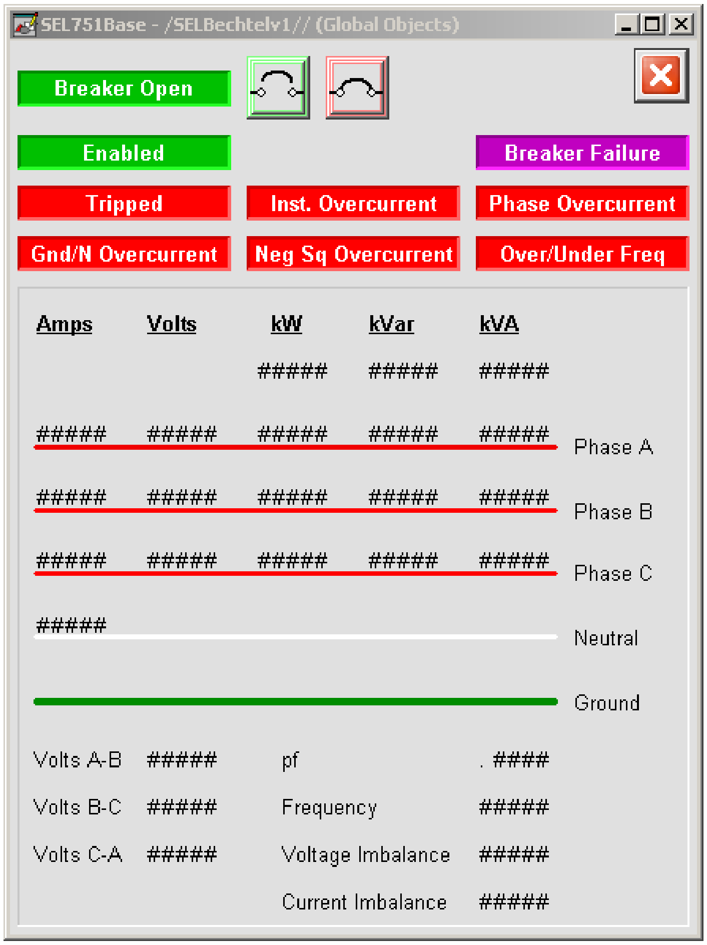


Figure 6.2 Relay Faceplate Global Objects

Additionally, to make the faceplates easily reusable, a standard base faceplate was developed. Figure 6.3 shows the standard object and table formats used for the IED faceplates. These objects are shown as visible when various tabs of the faceplate are displayed. Again, the color scheme matches ISA 5.5 and complies with IEEE C37.1. Information tables, such as the line to neutral voltages and currents table, are implemented in the engineering tab of each faceplate. The "#" field indicates that the text field will be a numeric. The power conductors are represented by red, the neutral conductor by white, and grounding conductor(s) by green. A logo was also requested from various manufacturers for implementation in the faceplates. These images are used on the home page to provide the user with the same look and feel as the physical device. An example of a manufacturer logo can be seen in Figure 6.4.



Amps	Volts	kW	kVar	kVA	
		#####	#####	#####	
#####	#####	#####	#####	#####	Phase A
#####	#####	#####	#####	#####	Phase B
#####	#####	#####	#####	#####	Phase C
#####					Neutral
					Ground

Volts A-B	#####	pf	. ####
Volts B-C	#####	Frequency	#####
Volts C-A	#####	Voltage Imbalance	#####
		Current Imbalance	#####

Figure 6.3 Relay Base Objects

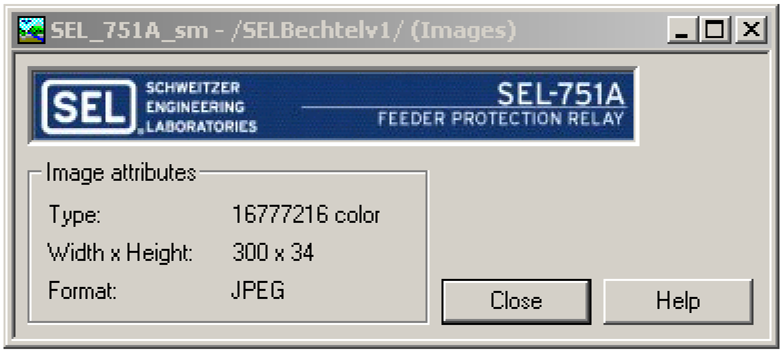
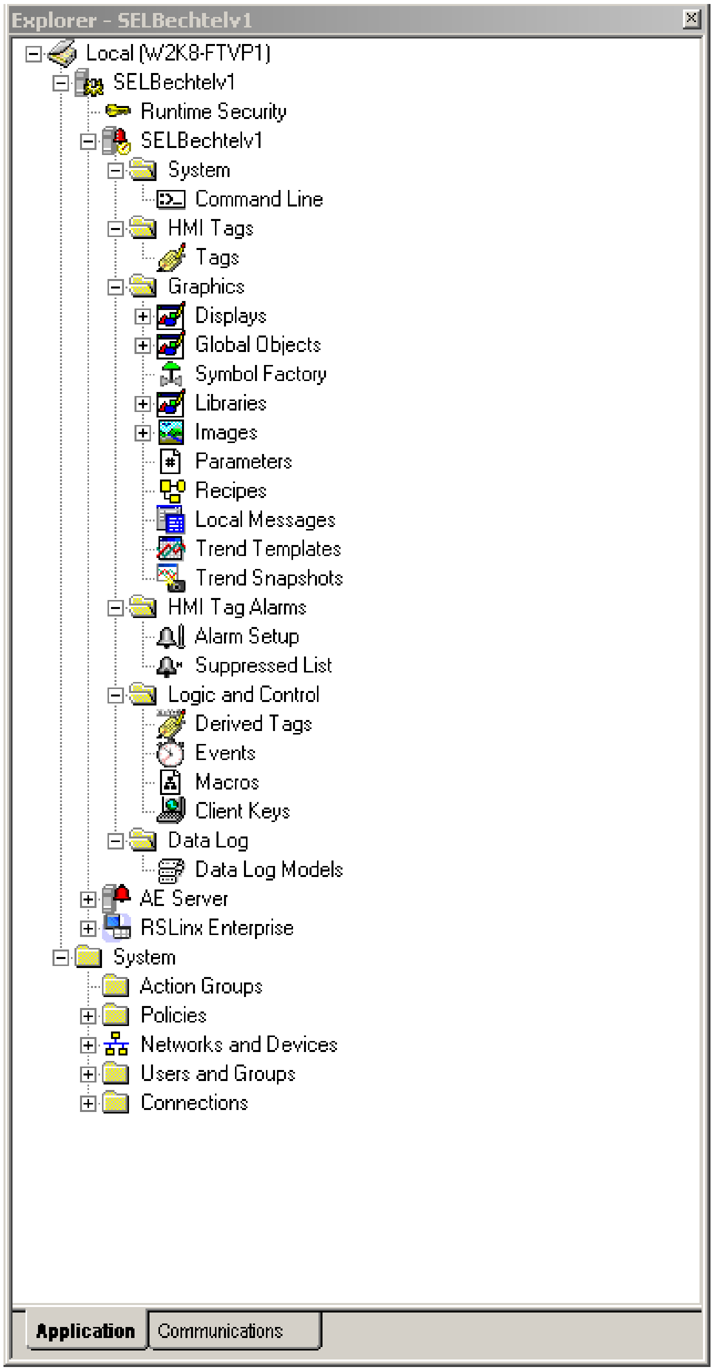

SEL_751A_sm - /SELBechtelv1/ (Images)
SEL SCHWEITZER ENGINEERING LABORATORIES
SEL-751A
FEEDER PROTECTION RELAY
Close
Help

Image attributes

Type:	16777216 color
Width x Height:	300 x 34
Format:	JPEG

Figure 6.4 Manufacturer Logo

Each of these objects is then mapped into an RS View project. RS View is a software HMI package that allows for the creation of various HMI screens and visual interfaces. Figure 6.5 shows an example of how the global objects, images, alarms and events are all mapped into a visualization project.

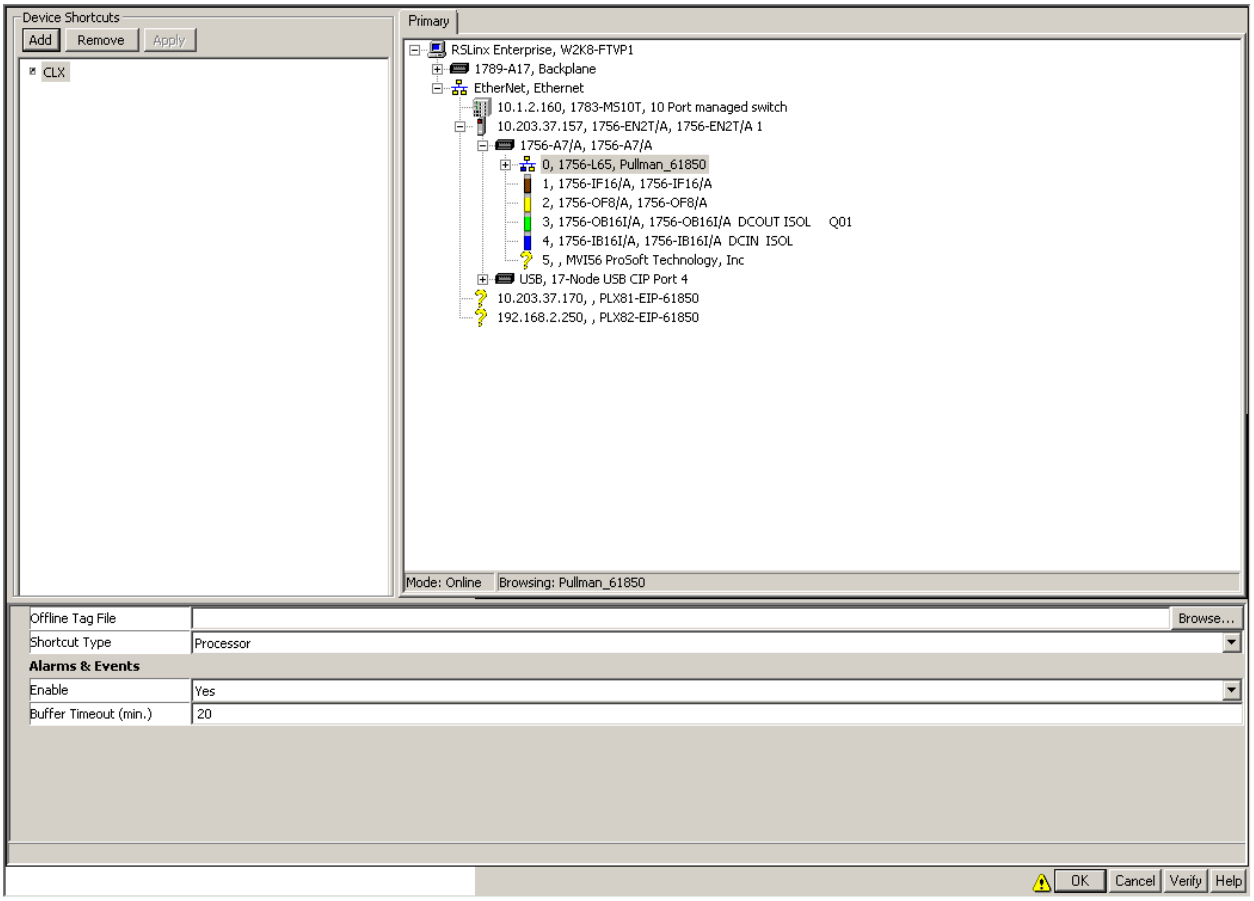


```
Explorer - SELBechtelv1
⊟ Local (W2K8-FTVP1)
  ⊟ SELBechtelv1
      Runtime Security
    ⊟ SELBechtelv1
      ⊟ System
          Command Line
      ⊟ HMI Tags
          Tags
      ⊟ Graphics
        ⊞ Displays
        ⊞ Global Objects
          Symbol Factory
        ⊞ Libraries
        ⊞ Images
          Parameters
          Recipes
          Local Messages
          Trend Templates
          Trend Snapshots
      ⊟ HMI Tag Alarms
          Alarm Setup
          Suppressed List
      ⊟ Logic and Control
          Derived Tags
          Events
          Macros
          Client Keys
      ⊟ Data Log
          Data Log Models
    ⊞ AE Server
    ⊞ RSLinx Enterprise
  ⊟ System
      Action Groups
    ⊞ Policies
    ⊞ Networks and Devices
    ⊞ Users and Groups
    ⊞ Connections
```

Figure 6.5 RS View Window Object Pane

Linking Controller Tags to Graphics

In order for the graphics and faceplates to have meaningful representations, the fields for text and numerical values must be linked to controller tags. This is done by creating an RS Linx Enterprise connection to the controller processor. This process is depicted in Figure 6.6. In this figure, the left workspace pane defines the shortcut "CLX" as the active data path. The user browses to the EtherNet/IP module and points to the corresponding processor, in this case "Pullman_61850." After this is completed, the user selects the "Apply" button and also selects the controller project for the offline tag database. Once this controller connection is made, the data can begin to flow from controller to HMI server where information can populate the faceplates.



```
RSLinx Enterprise, W2K8-FTVP1
  1789-A17, Backplane
  EtherNet, Ethernet
    10.1.2.160, 1783-MS10T, 10 Port managed switch
    10.203.37.157, 1756-EN2T/A, 1756-EN2T/A 1
      1756-A7/A, 1756-A7/A
        0, 1756-L65, Pullman_61850
        1, 1756-IF16/A, 1756-IF16/A
        2, 1756-OF8/A, 1756-OF8/A
        3, 1756-OB16I/A, 1756-OB16I/A  DCOUT ISOL    Q01
        4, 1756-IB16I/A, 1756-IB16I/A  DCIN  ISOL
        5, , MVI56 ProSoft Technology, Inc
      USB, 17-Node USB CIP Port 4
    10.203.37.170, , PLX81-EIP-61850
    192.168.2.250, , PLX82-EIP-61850
```

Mode: Online   Browsing: Pullman_61850

Offline Tag File	
Shortcut Type	Processor
Alarms & Events	
Enable	Yes
Buffer Timeout (min.)	20

Figure 6.6 Linking HMI to Controller

Mapping Tags to Data Fields

The faceplates created for this research are multi-layered frames. This means that each tab is its own screen with attributes of visibility. When certain tabs at the top of the faceplate (as seen in Figure 6.7) are selected, the selected frame is set visible, while the remaining frames have their visibilities set to false. Also, each numeric, text, or button, needs to be assigned to a controller tag in order to populate the field with meaningful power parameter information. Figure 6.7 shows two panels: the faceplate to the left, and the object explorer to the right. The "VA_RMS51" tag is selected in grey, which corresponds to the "Frequency" numeric field as shown in the faceplate. Every tag in the object explorer is linked to a text, numeric, or button field in the faceplate.

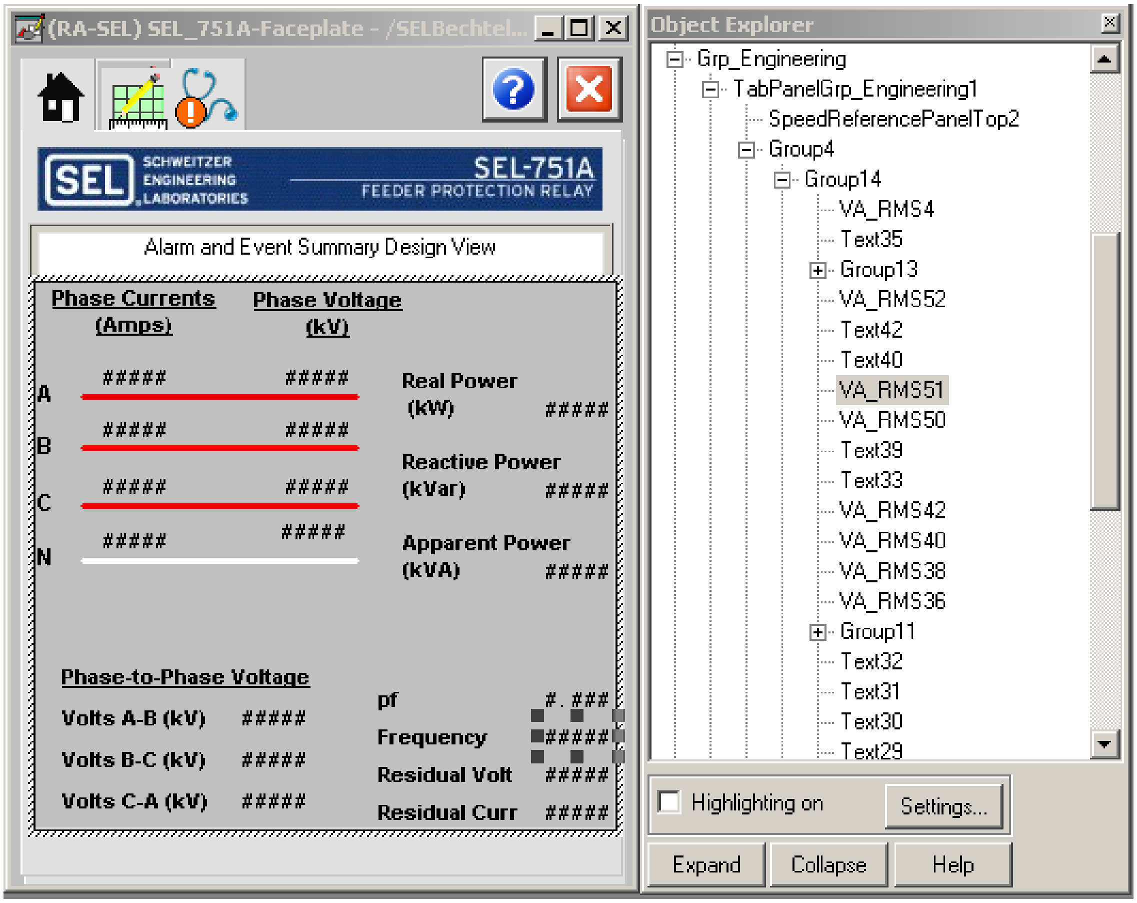
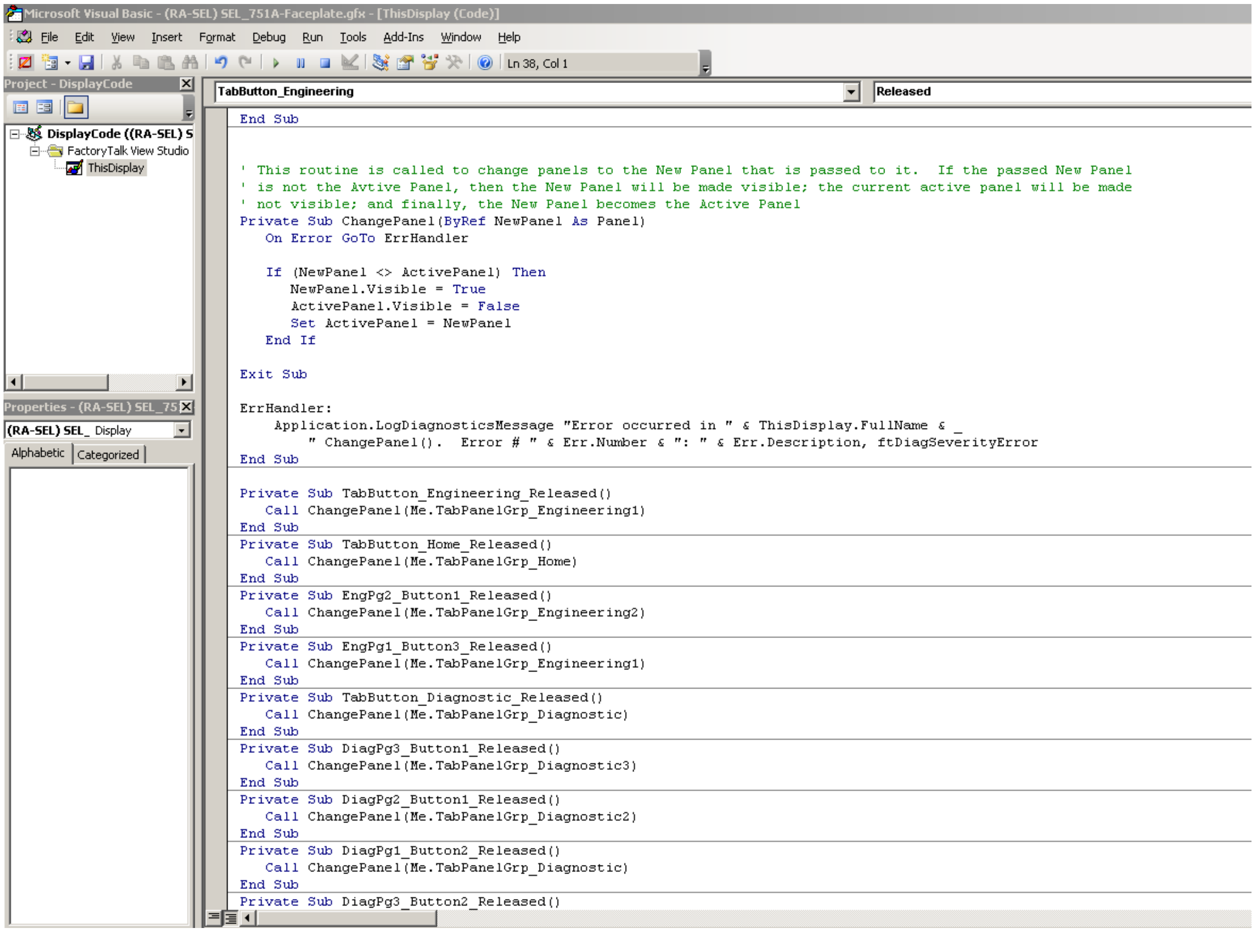


Figure 6.7 Frames of a Faceplate

In order to move between tabs of the faceplate, Visual Basic (VB) script was written to set various tabs to visible and invisible. The code to perform these functions can be seen in Figure 6.8.

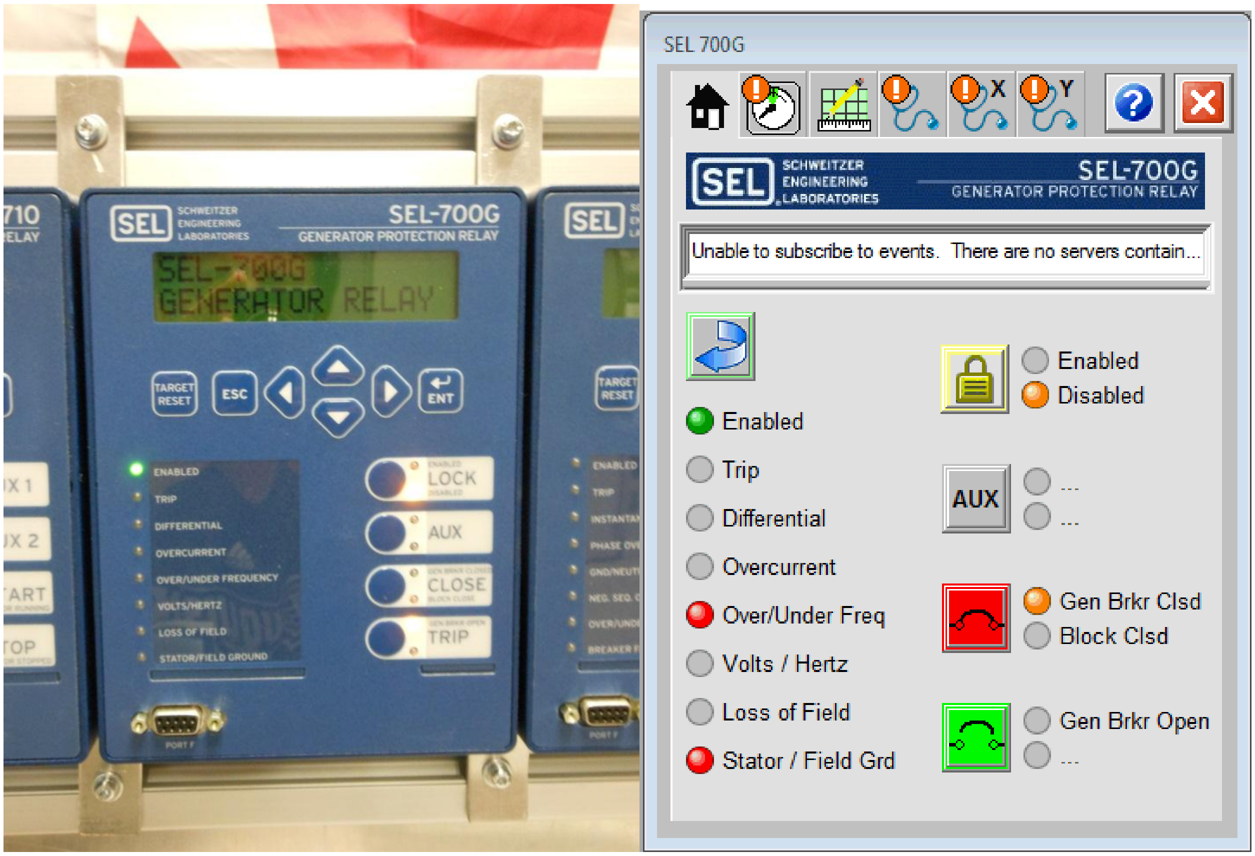

Microsoft Visual Basic - (RA-SEL) SEL_751A-Faceplate.gfx - [ThisDisplay (Code)]
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Ln 38, Col 1
Project - DisplayCode
DisplayCode ((RA-SEL) S
FactoryTalk View Studio
ThisDisplay
Properties - (RA-SEL) SEL_75
(RA-SEL) SEL_ Display
Alphabetic
Categorized
TabButton_Engineering
Released

```
End Sub

' This routine is called to change panels to the New Panel that is passed to it.  If the passed New Panel
' is not the Avtive Panel, then the New Panel will be made visible; the current active panel will be made
' not visible; and finally, the New Panel becomes the Active Panel
Private Sub ChangePanel(ByRef NewPanel As Panel)
   On Error GoTo ErrHandler

   If (NewPanel <> ActivePanel) Then
      NewPanel.Visible = True
      ActivePanel.Visible = False
      Set ActivePanel = NewPanel
   End If

Exit Sub

ErrHandler:
    Application.LogDiagnosticsMessage "Error occurred in " & ThisDisplay.FullName & _
        " ChangePanel().  Error # " & Err.Number & ": " & Err.Description, ftDiagSeverityError
End Sub

Private Sub TabButton_Engineering_Released()
   Call ChangePanel(Me.TabPanelGrp_Engineering1)
End Sub
Private Sub TabButton_Home_Released()
   Call ChangePanel(Me.TabPanelGrp_Home)
End Sub
Private Sub EngPg2_Button1_Released()
   Call ChangePanel(Me.TabPanelGrp_Engineering2)
End Sub
Private Sub EngPg1_Button3_Released()
   Call ChangePanel(Me.TabPanelGrp_Engineering1)
End Sub
Private Sub TabButton_Diagnostic_Released()
   Call ChangePanel(Me.TabPanelGrp_Diagnostic)
End Sub
Private Sub DiagPg3_Button1_Released()
   Call ChangePanel(Me.TabPanelGrp_Diagnostic3)
End Sub
Private Sub DiagPg2_Button1_Released()
   Call ChangePanel(Me.TabPanelGrp_Diagnostic2)
End Sub
Private Sub DiagPg1_Button2_Released()
   Call ChangePanel(Me.TabPanelGrp_Diagnostic)
End Sub
Private Sub DiagPg3_Button2_Released()
```

Figure 6.8 Visual Basic Script to Browse Tabs

Look and Feel

The concept behind developing faceplates is to allow for remote monitoring as well as command and control of various power distribution components. At the same time, the graphics must have the same look and feel as the physical device for engineers to understand how to use the graphic from a remote location.



Unable to subscribe to events. There are no servers contain...

Figure 6.9 Physical and Graphical Comparison

Figure 6.9 shows a comparison of the physical IED to the virtualized representation of the IED in the form of a faceplate. The graphic on the left is a photograph taken of an SEL 700G generator protection relay. This IED is used to protect rotating synchronous generators that are found in mining applications on sites that have their own co-generation. The graphic on the right represents the IED in faceplate format. It can be seen that the faceplate is identified as a 700G, or generator protection relay, from the title bar of the faceplate window as well as from the manufacturer’s image mark implemented as a global object. The target LED panels down the left represent the same text that is represented on the physical IED, for example, Enabled, Trip, Differential, etc. As all of these labels are configurable on the IED, the same text is also left up to the user to define. For example, if the Differential LED was set to Reverse Power flow in the IED, the faceplate could be altered to represent this change. The pushbutton LEDs, (right column), are also graphically represented in the faceplate. Each button has a maximum of three lines of text that could be assigned. The first and third lines of text refer to the operation of the amber LEDs, while the second line of text references the operation of the push button. Graphically this information is represented as follows:

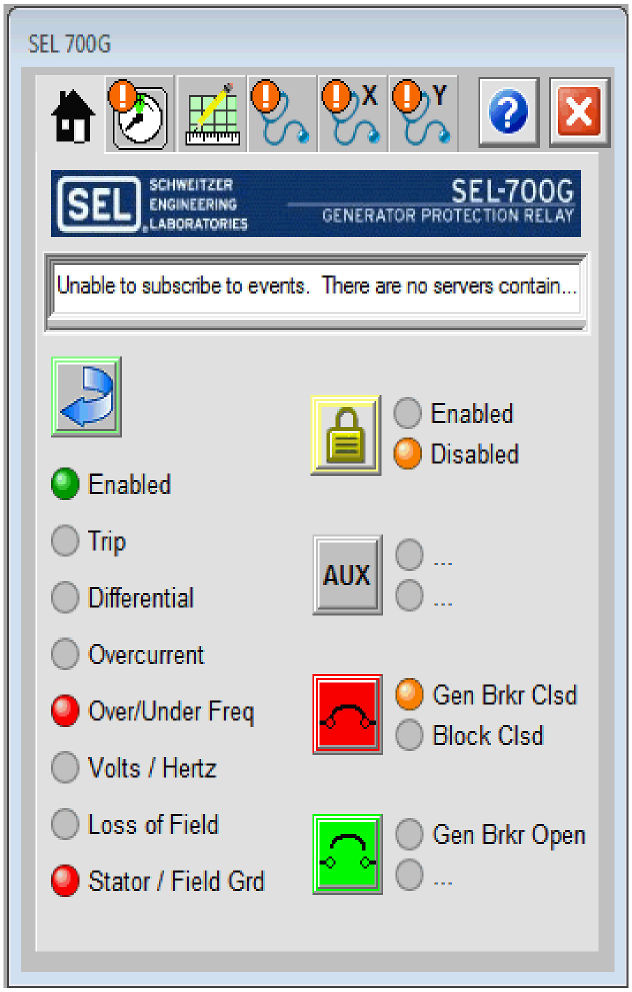
1. The text on the second line with reference to the push buttons is graphically represented as a faceplate pushbutton
2. The text on the first and third line reference the text with amber LEDs on the IEDs themselves. These LEDs represent the internal state of the IED.

Additionally, each faceplate has an alarm and events banner built into the faceplate. This allows the operator or engineer quick access on the home screen to determine any warning, diagnostics, or trip information from the relay. This alarm banner is populated from logic written in the controller.

The tabs across the top of the faceplate represent home, synchronization, engineering, overall faults, X-side (Low Voltage) faults, and Y-side (High Voltage) faults. The next section of this chapter presents the various screens for the IED faceplates developed in this research. Faceplates were developed for four IEDs: generator, feeder, motor, and differential. These can be seen in Figures 6.10-6.22.

Generator Faceplate

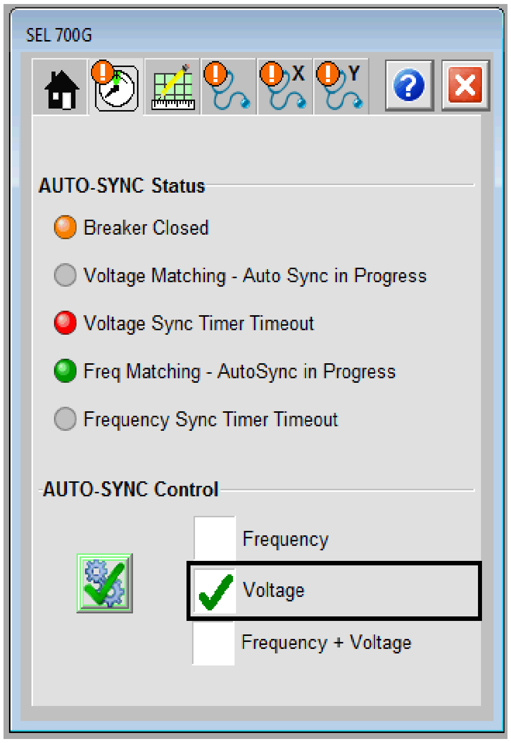
Home Screen



Unable to subscribe to events. There are no servers contain...

Figure 6.10 Generator Relay Home Screen

Synchronization Screen



SEL 700G

AUTO-SYNC Status

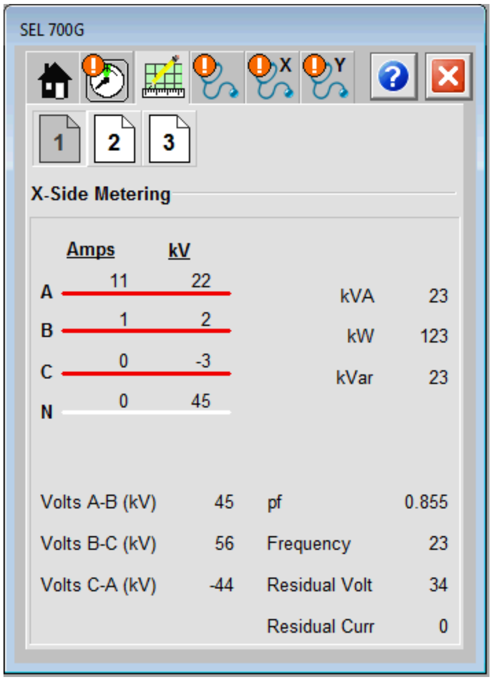
- ● Breaker Closed
- ○ Voltage Matching - Auto Sync in Progress
- ● Voltage Sync Timer Timeout
- ● Freq Matching - AutoSync in Progress
- ○ Frequency Sync Timer Timeout

AUTO-SYNC Control

- □ Frequency
- ☑ Voltage
- □ Frequency + Voltage

Figure 6.11 Generator Relay Synchronization Screen

Measurands Screen



X-Side Metering

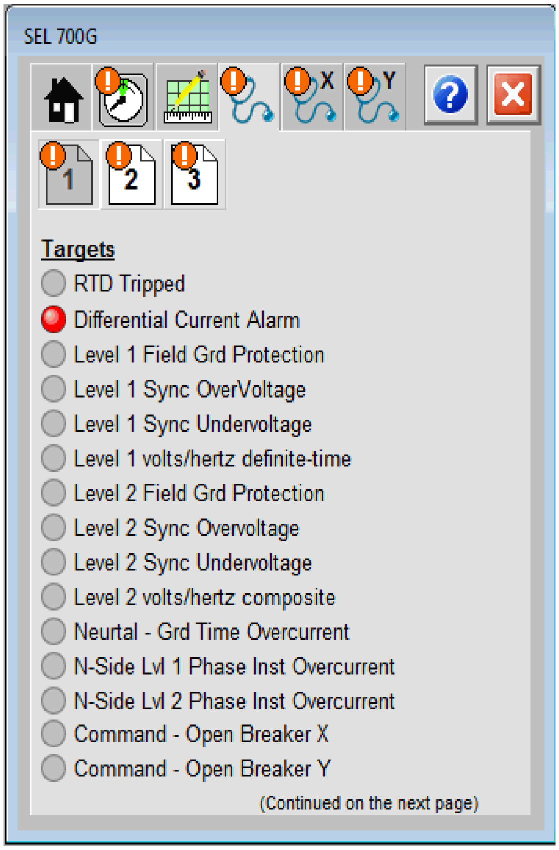
	Amps	kV
A	11	22
B	1	2
C	0	-3
N	0	45

kVA	23
kW	123
kVar	23

Volts A-B (kV)	45	pf	0.855
Volts B-C (kV)	56	Frequency	23
Volts C-A (kV)	-44	Residual Volt	34
		Residual Curr	0

Figure 6.12 Generator Relay Metering Screen

Faults Screen



SEL 700G

Targets

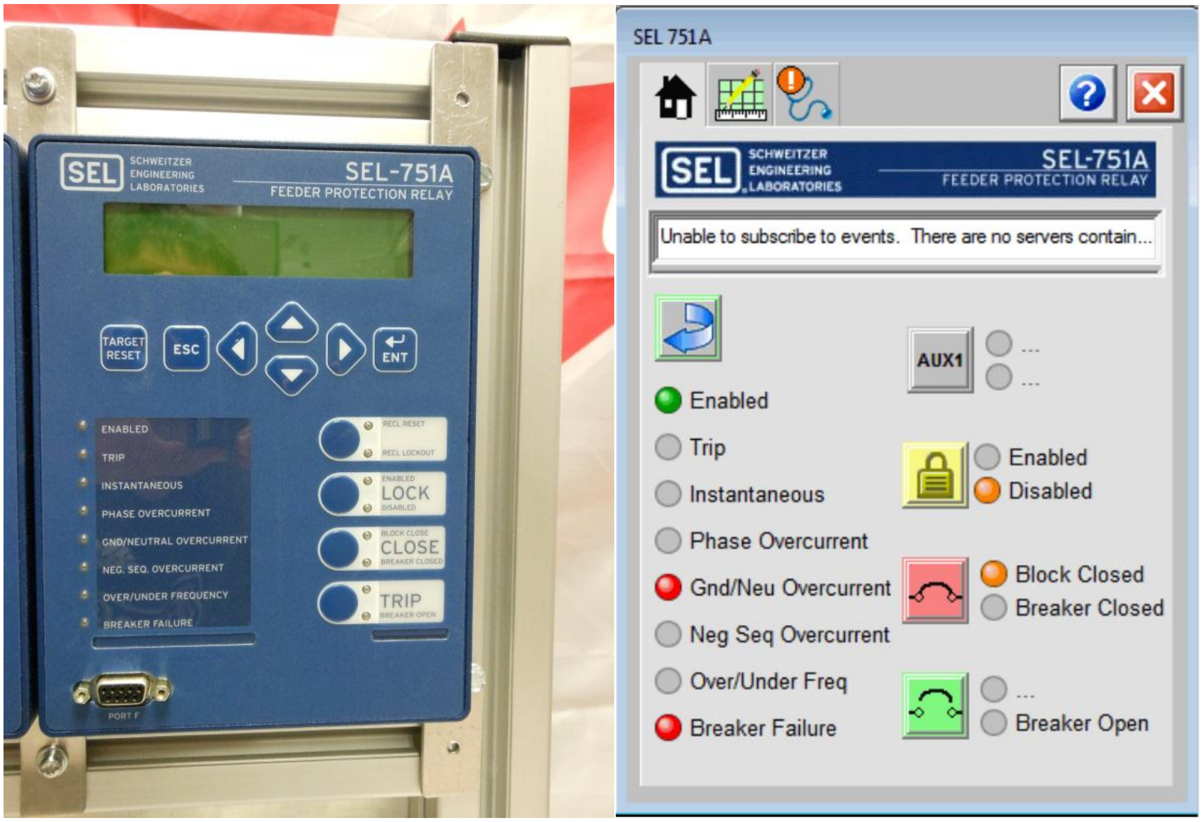
○ RTD Tripped
● Differential Current Alarm
○ Level 1 Field Grd Protection
○ Level 1 Sync OverVoltage
○ Level 1 Sync Undervoltage
○ Level 1 volts/hertz definite-time
○ Level 2 Field Grd Protection
○ Level 2 Sync Overvoltage
○ Level 2 Sync Undervoltage
○ Level 2 volts/hertz composite
○ Neurtal - Grd Time Overcurrent
○ N-Side Lvl 1 Phase Inst Overcurrent
○ N-Side Lvl 2 Phase Inst Overcurrent
○ Command - Open Breaker X
○ Command - Open Breaker Y

(Continued on the next page)

Figure 6.13 Generator Relay Faults Screen

Feeder Relay Faceplate

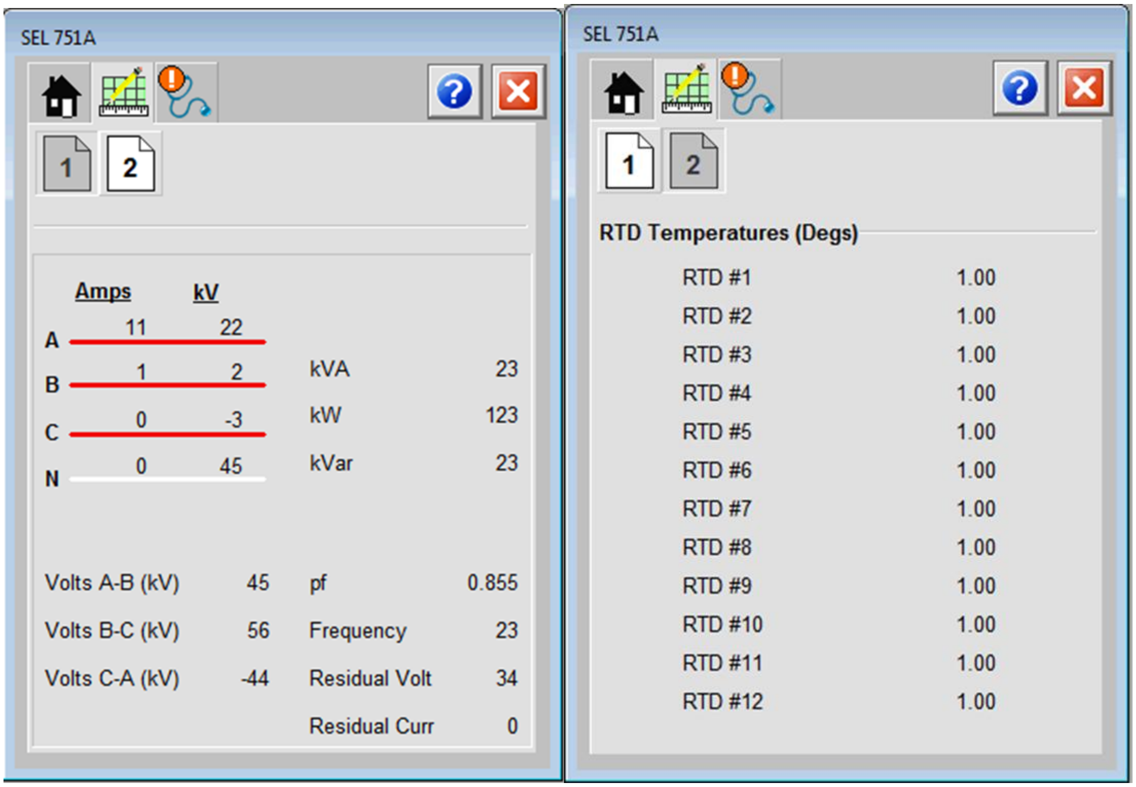
Home Screen



Unable to subscribe to events. There are no servers contain...

Figure 6.14 Feeder Relay Home Screen

Measurands Screen



	Amps	kV
A	11	22
B	1	2
C	0	-3
N	0	45

kVA	23
kW	123
kVar	23

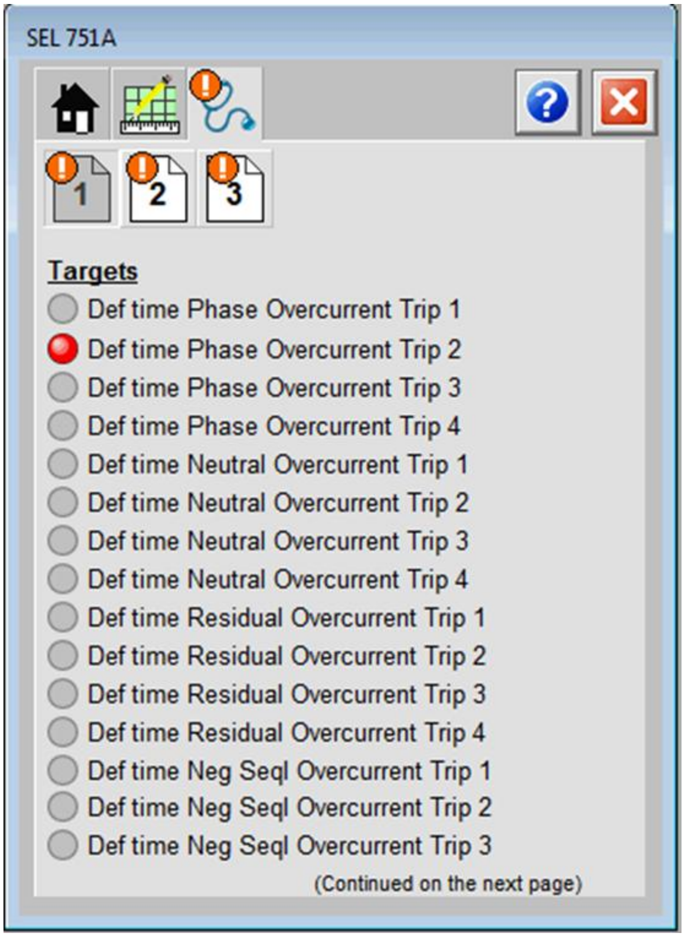
Volts A-B (kV)	45	pf	0.855
Volts B-C (kV)	56	Frequency	23
Volts C-A (kV)	-44	Residual Volt	34
		Residual Curr	0

RTD Temperatures (Degs)

RTD #1	1.00
RTD #2	1.00
RTD #3	1.00
RTD #4	1.00
RTD #5	1.00
RTD #6	1.00
RTD #7	1.00
RTD #8	1.00
RTD #9	1.00
RTD #10	1.00
RTD #11	1.00
RTD #12	1.00

Figure 6.15 Feeder Relay Engineering Screen

Faults Screen



SEL 751A

Targets

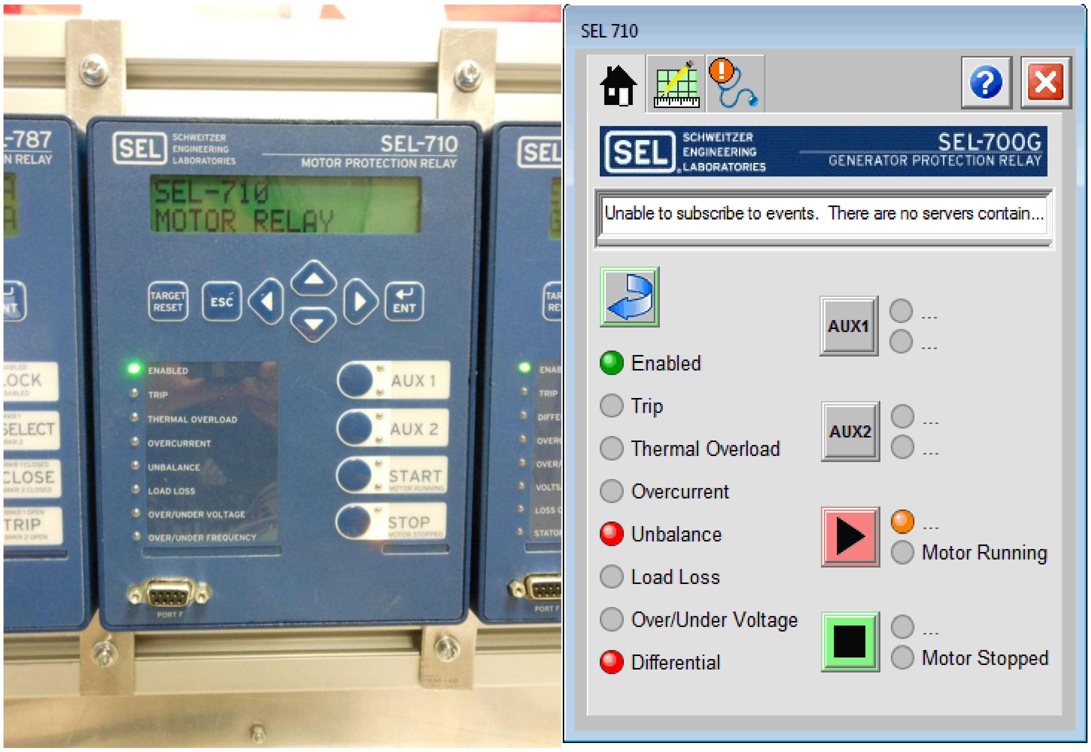
○ Def time Phase Overcurrent Trip 1
● Def time Phase Overcurrent Trip 2
○ Def time Phase Overcurrent Trip 3
○ Def time Phase Overcurrent Trip 4
○ Def time Neutral Overcurrent Trip 1
○ Def time Neutral Overcurrent Trip 2
○ Def time Neutral Overcurrent Trip 3
○ Def time Neutral Overcurrent Trip 4
○ Def time Residual Overcurrent Trip 1
○ Def time Residual Overcurrent Trip 2
○ Def time Residual Overcurrent Trip 3
○ Def time Residual Overcurrent Trip 4
○ Def time Neg Seql Overcurrent Trip 1
○ Def time Neg Seql Overcurrent Trip 2
○ Def time Neg Seql Overcurrent Trip 3

(Continued on the next page)

Figure 6.16 Feeder Relay Faults Screen

Motor Relay Faceplate

Home Screen

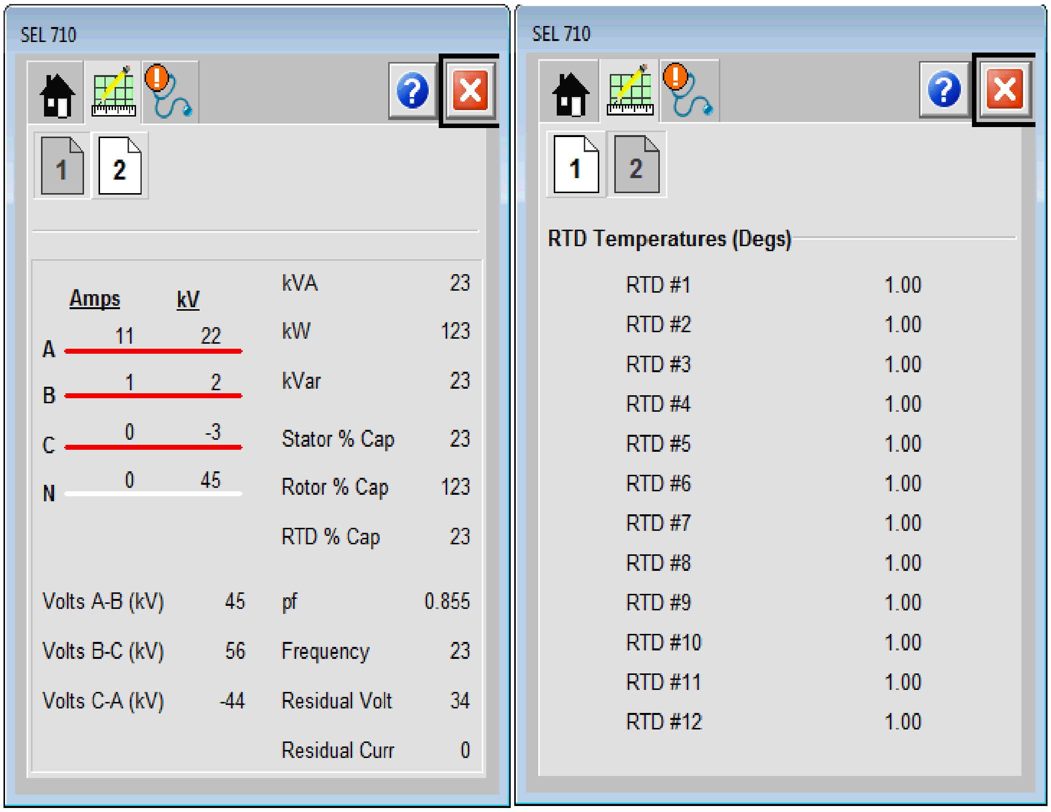


SEL 710

Unable to subscribe to events. There are no servers contain...

Figure 6.17 Motor Relay Home Screen

Measurands Screen



SEL 710

1 2

	Amps	kV
A	11	22
B	1	2
C	0	-3
N	0	45

kVA	23
kW	123
kVar	23
Stator % Cap	23
Rotor % Cap	123
RTD % Cap	23

Volts A-B (kV)	45
Volts B-C (kV)	56
Volts C-A (kV)	-44

pf	0.855
Frequency	23
Residual Volt	34
Residual Curr	0

SEL 710

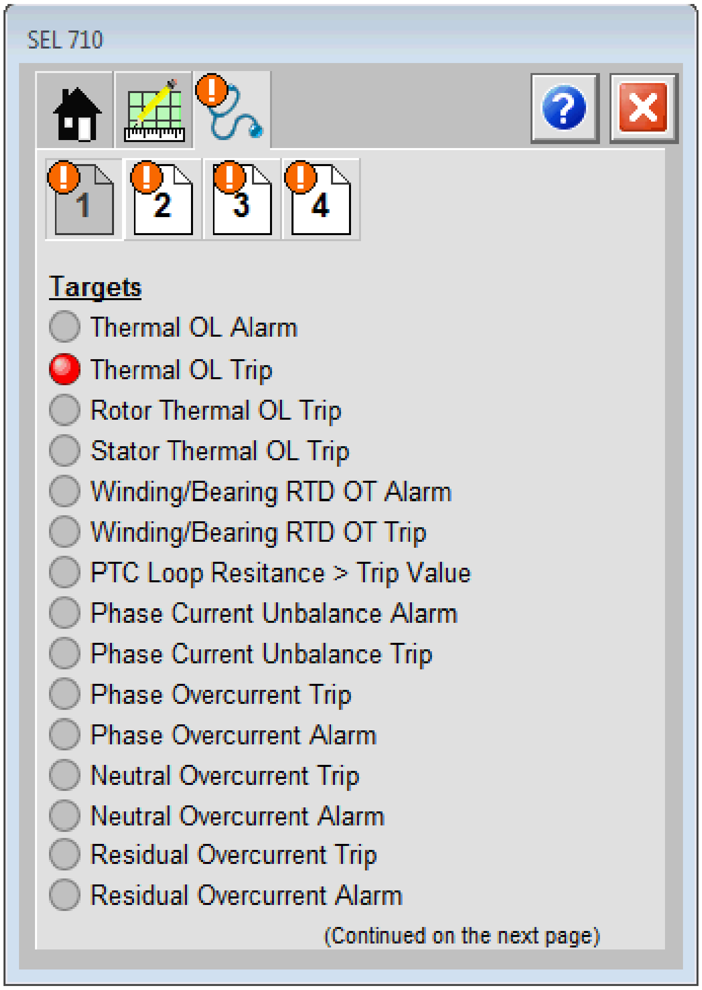
1 2

RTD Temperatures (Degs)

RTD #1	1.00
RTD #2	1.00
RTD #3	1.00
RTD #4	1.00
RTD #5	1.00
RTD #6	1.00
RTD #7	1.00
RTD #8	1.00
RTD #9	1.00
RTD #10	1.00
RTD #11	1.00
RTD #12	1.00

Figure 6.18 Motor Relay Measurands Screen

Faults Screen



Targets

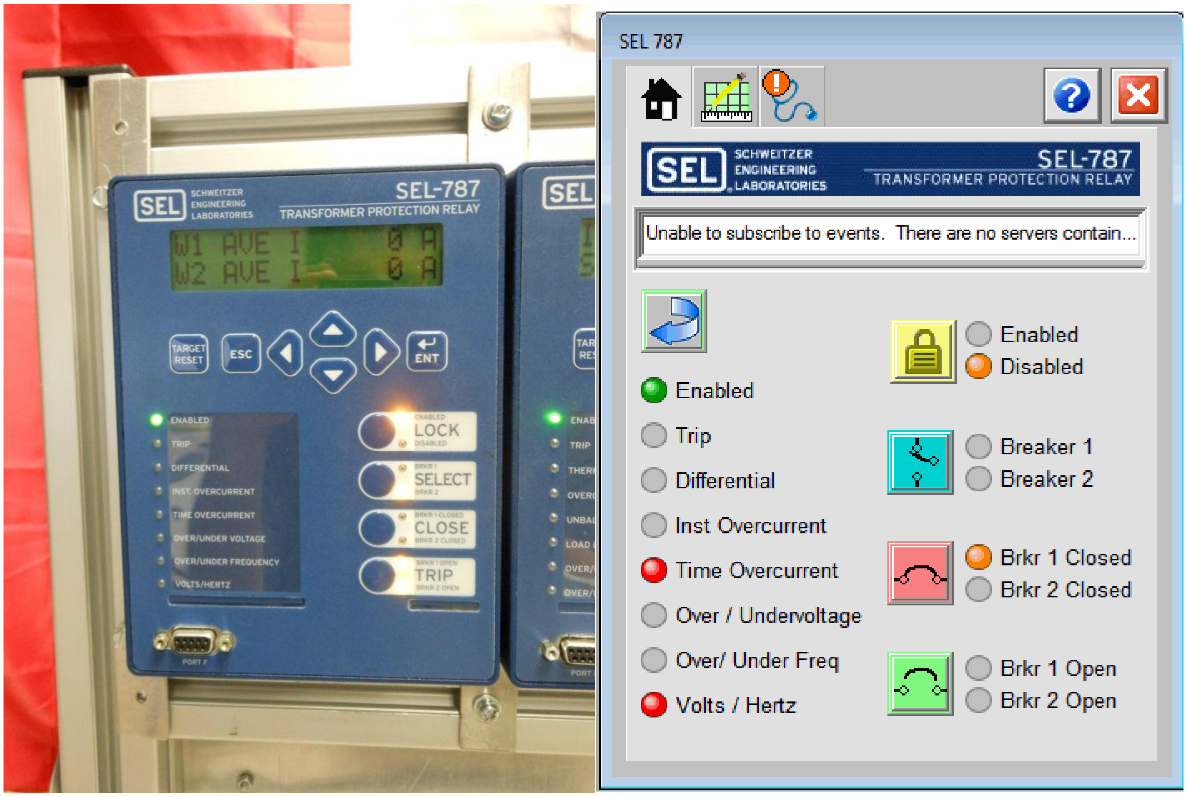
○ Thermal OL Alarm
● Thermal OL Trip
○ Rotor Thermal OL Trip
○ Stator Thermal OL Trip
○ Winding/Bearing RTD OT Alarm
○ Winding/Bearing RTD OT Trip
○ PTC Loop Resitance > Trip Value
○ Phase Current Unbalance Alarm
○ Phase Current Unbalance Trip
○ Phase Overcurrent Trip
○ Phase Overcurrent Alarm
○ Neutral Overcurrent Trip
○ Neutral Overcurrent Alarm
○ Residual Overcurrent Trip
○ Residual Overcurrent Alarm

(Continued on the next page)

Figure 6.19 Motor Relay Faults Screen

Differential Relay Faceplate

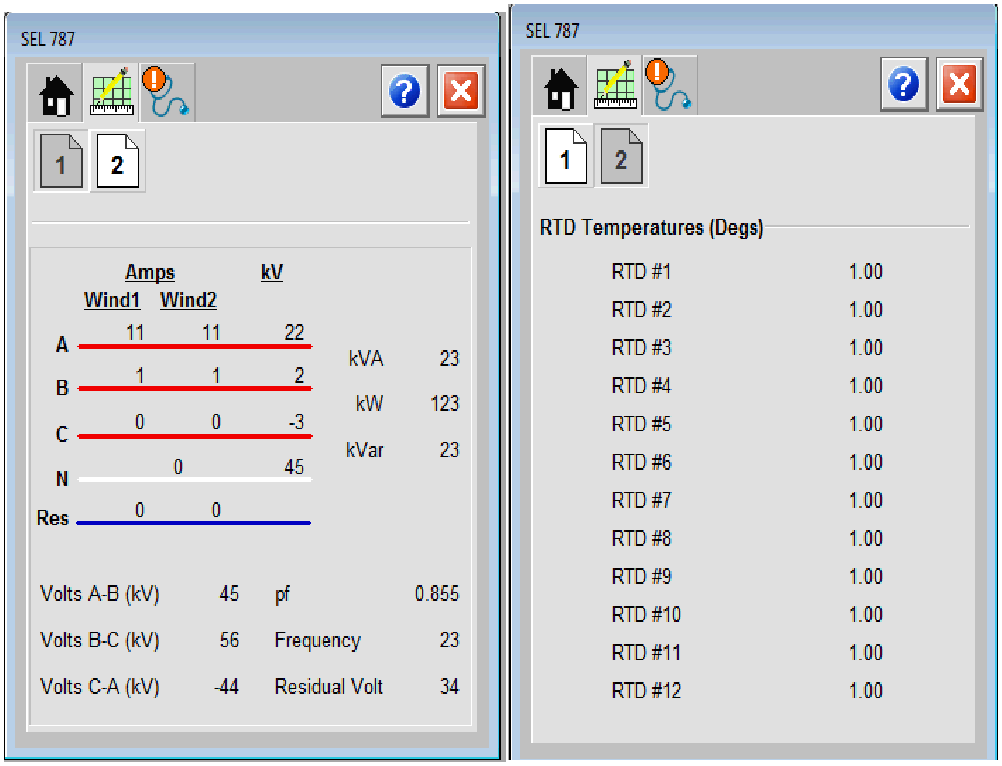
Home Screen



Unable to subscribe to events. There are no servers contain...

Figure 6.20 Transformer Protection Relay Home Screen

Measurands Screen



SEL 787

	Amps		kV		
	Wind1	Wind2			
A	11	11	22	kVA	23
B	1	1	2	kW	123
C	0	0	-3	kVar	23
N	0		45		
Res	0	0			

Volts A-B (kV)	45	pf	0.855
Volts B-C (kV)	56	Frequency	23
Volts C-A (kV)	-44	Residual Volt	34

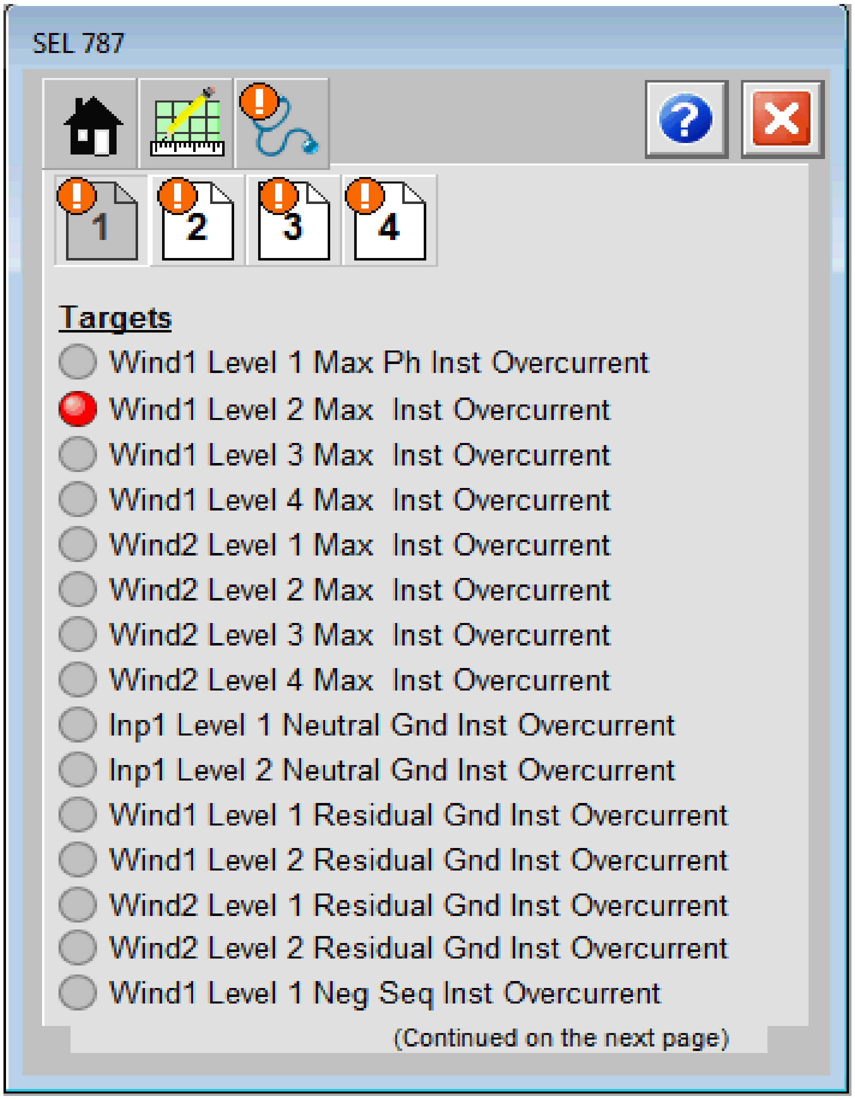
SEL 787

RTD Temperatures (Degs)

RTD #1	1.00
RTD #2	1.00
RTD #3	1.00
RTD #4	1.00
RTD #5	1.00
RTD #6	1.00
RTD #7	1.00
RTD #8	1.00
RTD #9	1.00
RTD #10	1.00
RTD #11	1.00
RTD #12	1.00

Figure 6.21 Differential Relay Measurands Screen

Faults Screen



SEL 787

Targets

○ Wind1 Level 1 Max Ph Inst Overcurrent
● Wind1 Level 2 Max Inst Overcurrent
○ Wind1 Level 3 Max Inst Overcurrent
○ Wind1 Level 4 Max Inst Overcurrent
○ Wind2 Level 1 Max Inst Overcurrent
○ Wind2 Level 2 Max Inst Overcurrent
○ Wind2 Level 3 Max Inst Overcurrent
○ Wind2 Level 4 Max Inst Overcurrent
○ Inp1 Level 1 Neutral Gnd Inst Overcurrent
○ Inp1 Level 2 Neutral Gnd Inst Overcurrent
○ Wind1 Level 1 Residual Gnd Inst Overcurrent
○ Wind1 Level 2 Residual Gnd Inst Overcurrent
○ Wind2 Level 1 Residual Gnd Inst Overcurrent
○ Wind2 Level 2 Residual Gnd Inst Overcurrent
○ Wind1 Level 1 Neg Seq Inst Overcurrent

(Continued on the next page)

Figure 6.22 Differential Relay Faults Screen

Adding Faceplates to a Project

After developing the faceplates for this research, the next component that needed to be considered was the ease of implementation and commissioning of the global faceplates. The faceplate are populated from two AOIs: one from the gateway module defining the user-defined data type (UDT) and one for the faceplate defining user-defined text strings that populate the home screen of each faceplate. Figure 6.23 shows how the Add-On Instruction is added to the project in order to populate the faceplate.

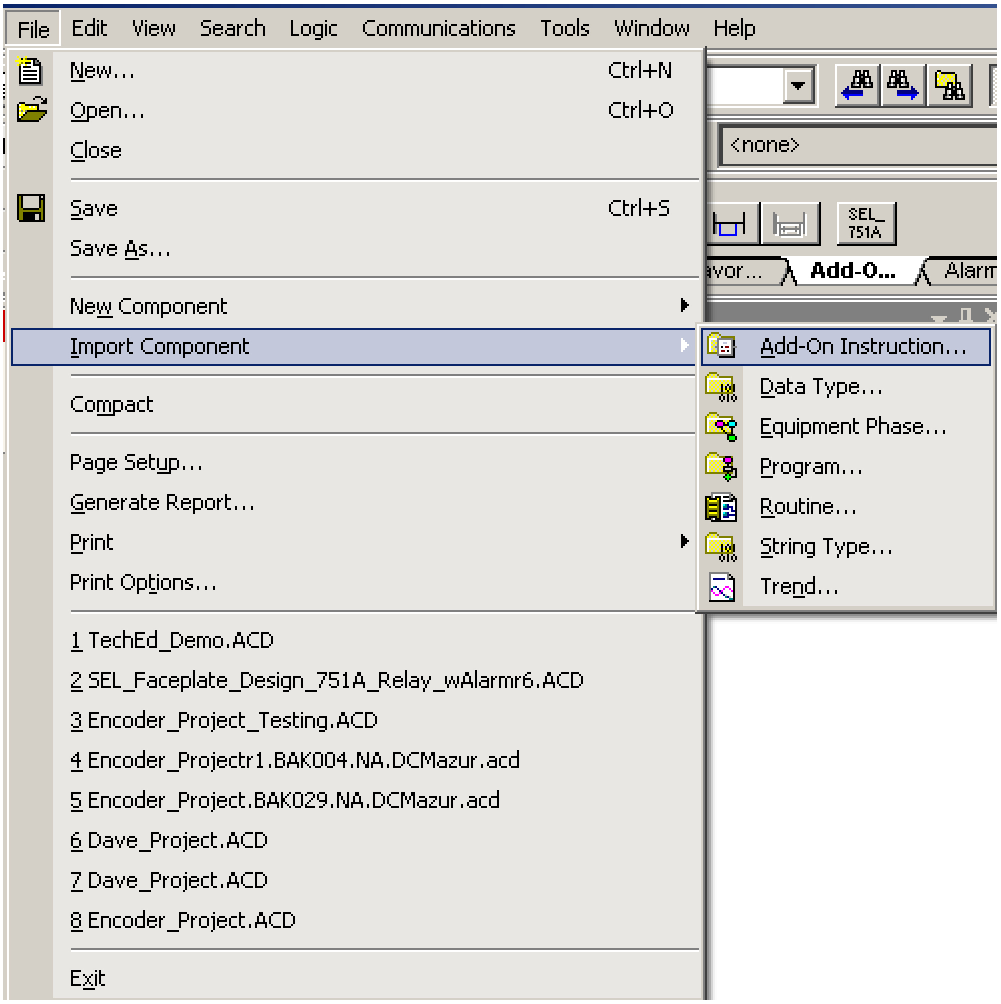
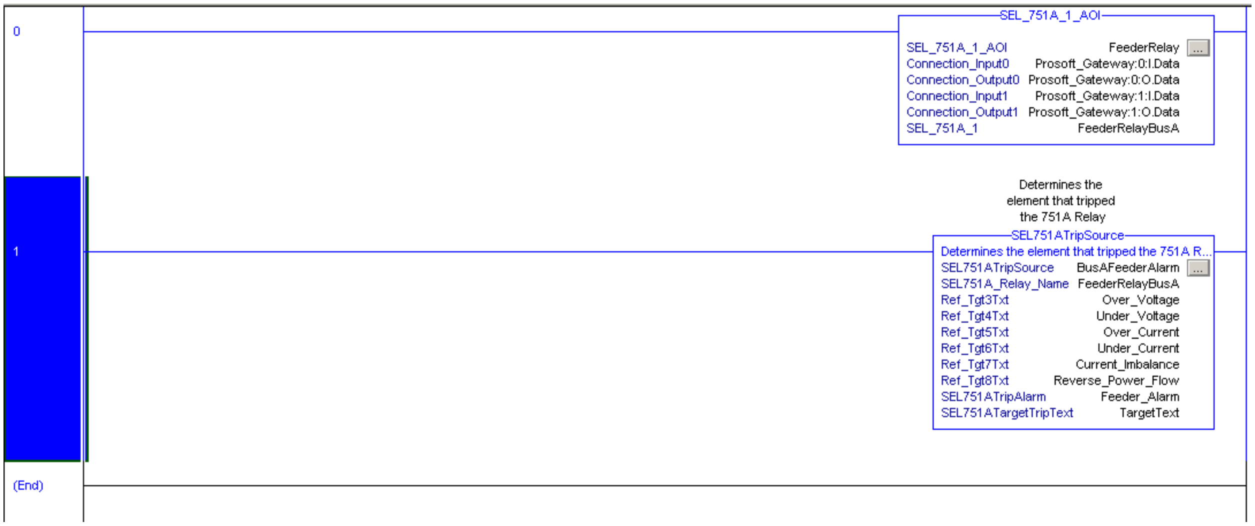


Figure 6.23 Importing Add-On Instruction

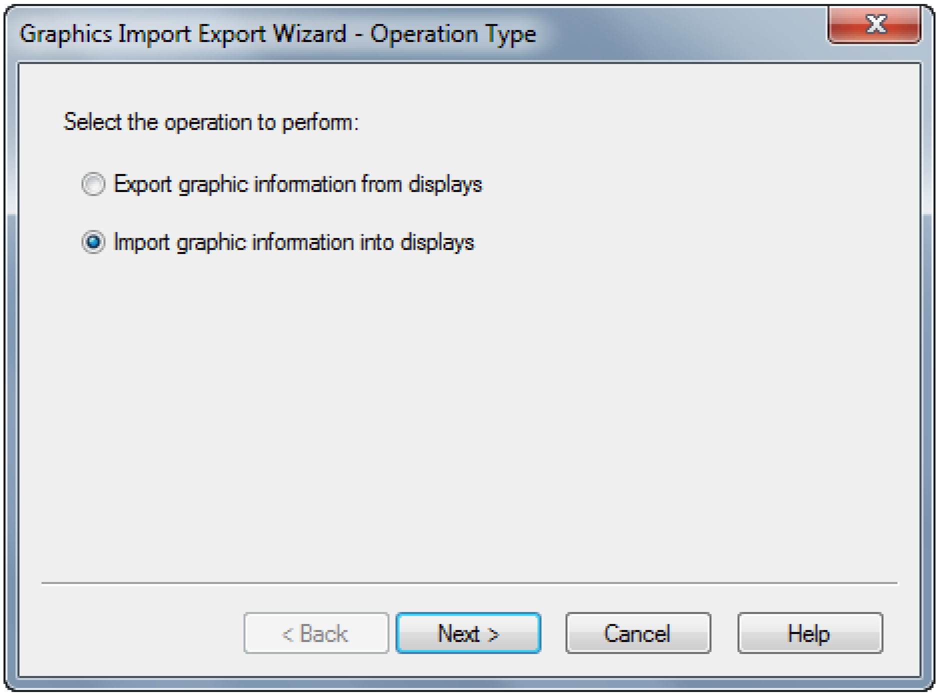
Figure 6.24 shows a sample ladder program with two AOIs: one defining the UDT (rung 0) and the other defining the AOI for the faceplate (rung 1). The AOI in rung 1 defines 11 different input tags. The reference tags refer to the text of the target LEDs that the user wants to show on the Faceplate when it is displayed. In this example, targets such as overvoltage, under voltage, and overcurrent, can be specified depending upon system definition. The relay name input parameter refers to the user-defined data type (UDT) as described in the software chapter. In this example, the relay name is FeederRelayBusA. The alarm for this faceplate references the name for the alarm that appears in the alarm and events database when this instruction is executed.



```
0
                 SEL_751A_1_AOI
SEL_751A_1_AOI                  FeederRelay  ...
Connection_Input0     Prosoft_Gateway:0:I.Data
Connection_Output0    Prosoft_Gateway:0:O.Data
Connection_Input1     Prosoft_Gateway:1:I.Data
Connection_Output1    Prosoft_Gateway:1:O.Data
SEL_751A_1                   FeederRelayBusA

1
                 Determines the
              element that tripped
                 the 751A Relay
               SEL751ATripSource
Determines the element that tripped the 751A R...
SEL751ATripSource          BusAFeederAlarm  ...
SEL751A_Relay_Name         FeederRelayBusA
Ref_Tgt3Txt                    Over_Voltage
Ref_Tgt4Txt                   Under_Voltage
Ref_Tgt5Txt                    Over_Current
Ref_Tgt6Txt                   Under_Current
Ref_Tgt7Txt               Current_Imbalance
Ref_Tgt8Txt              Reverse_Power_Flow
SEL751ATripAlarm               Feeder_Alarm
SEL751ATargetTripText            TargetText

(End)
```

Figure 6.24 Add-On Instruction Ladder

Figures 6.25 and 6.26 depict the steps to import the faceplate graphics into an existing HMI project. From the file menu, the user selects the import wizard tool. When the wizard loads, the user selects the import graphic displays into project option. After selecting this option and backing up the project for revision control, the user then points to the faceplate(s) that are intended to be imported into the project. These files are .gfx or graphics files that are designed for FactoryTalk View Studio applications.

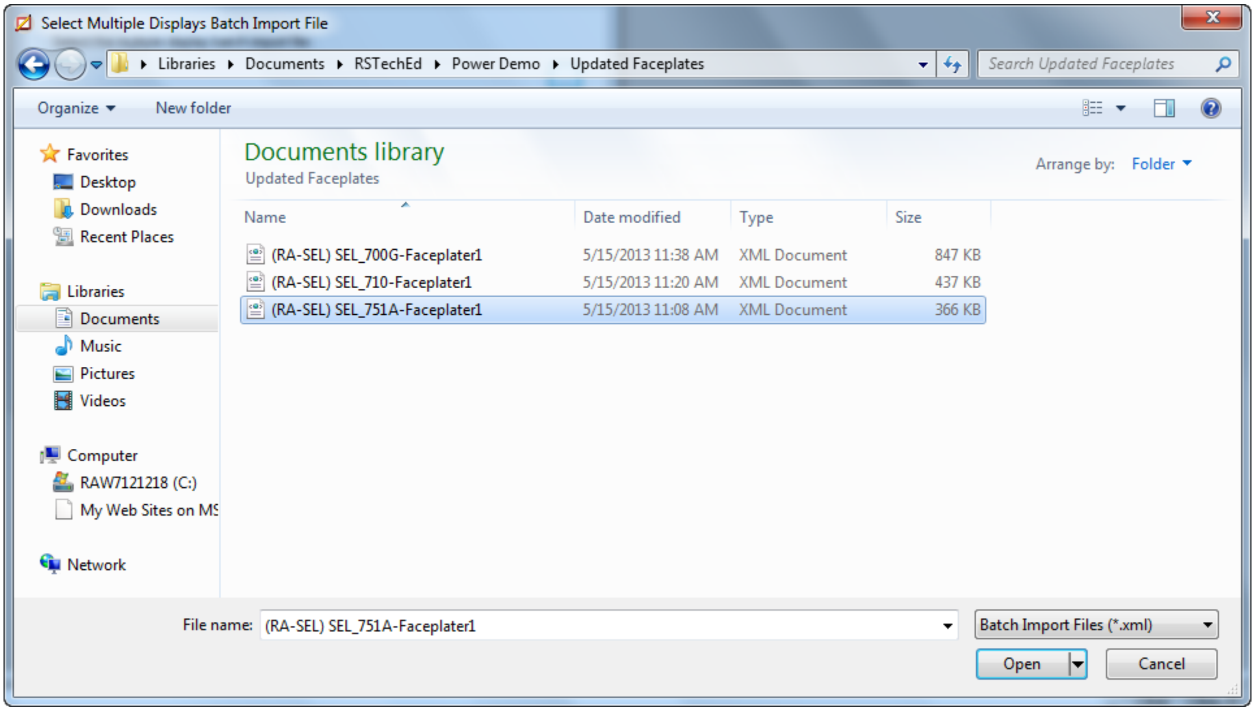


Select the operation to perform:

○ Export graphic information from displays

● Import graphic information into displays

Figure 6.25 Import GFX File



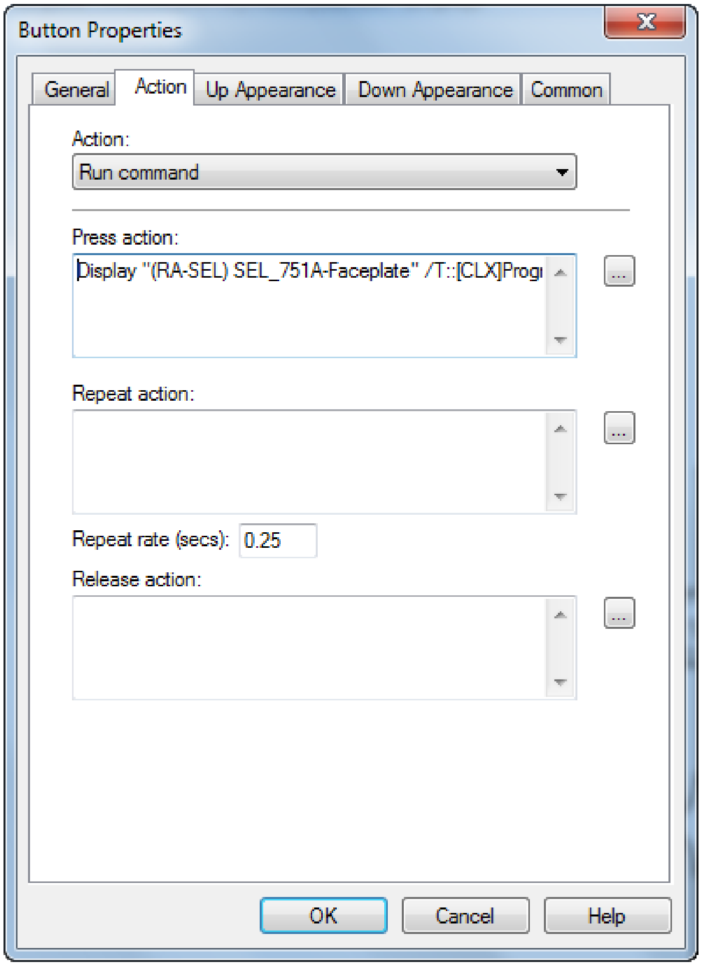
Name	Date modified	Type	Size
(RA-SEL) SEL_700G-Faceplater1	5/15/2013 11:38 AM	XML Document	847 KB
(RA-SEL) SEL_710-Faceplater1	5/15/2013 11:20 AM	XML Document	437 KB
(RA-SEL) SEL_751A-Faceplater1	5/15/2013 11:08 AM	XML Document	366 KB

File name: (RA-SEL) SEL_751A-Faceplater1

Batch Import Files (*.xml)

Figure 6.26 Select Correct Faceplate

Figure 6.27 defines the procedure for displaying the faceplate upon a button press in the HMI screen. In this example, the faceplate is displayed on the press action of a button. The first part of the command is <Display> and the name of the faceplate in this case is the 751A feeder relay. The second portion of the command is </T>, which looks for tag input parameters to populate the faceplate. The faceplates are populated by two parameters: the UDT of the 751A feeder relay (defined by the gateway) and the user defined input of the faceplate (target strings) defined by the faceplate AOI.



Action: Run command

Press action: Display "(RA-SEL) SEL_751A-Faceplate" /T::[CLX]Progı

Repeat action:

Repeat rate (secs): 0.25

Release action:

Figure 6.27 Display Commands

Note: /T::[CLX]Program:MainProgram.SEL751A, Points to SEL751A UDT (#1)

::[CLX]Program:MainProgram.Rockwell751AAddon, Points to Tag Architecture in AOI Instruction

Historian Interaction

With the development of the SOEtoFTH software tool as described in the software (Chapter 5), there is now an ability to display information from the historian graphically. Because the information from the electrical distribution system can now be stored in the repository in conjunction with standard process variables, information can now be plotted over trend periods. Figure 6.28 shows a trend screen with three IEDs represented by blue, red, and green. The blue and red relays have both tripped twice, as represented from the off to on transition. The corresponding timestamps are also gathered and populated on the x-axis of the visual trend. These distribution parameters can also be combined with process parameters and overlaid.

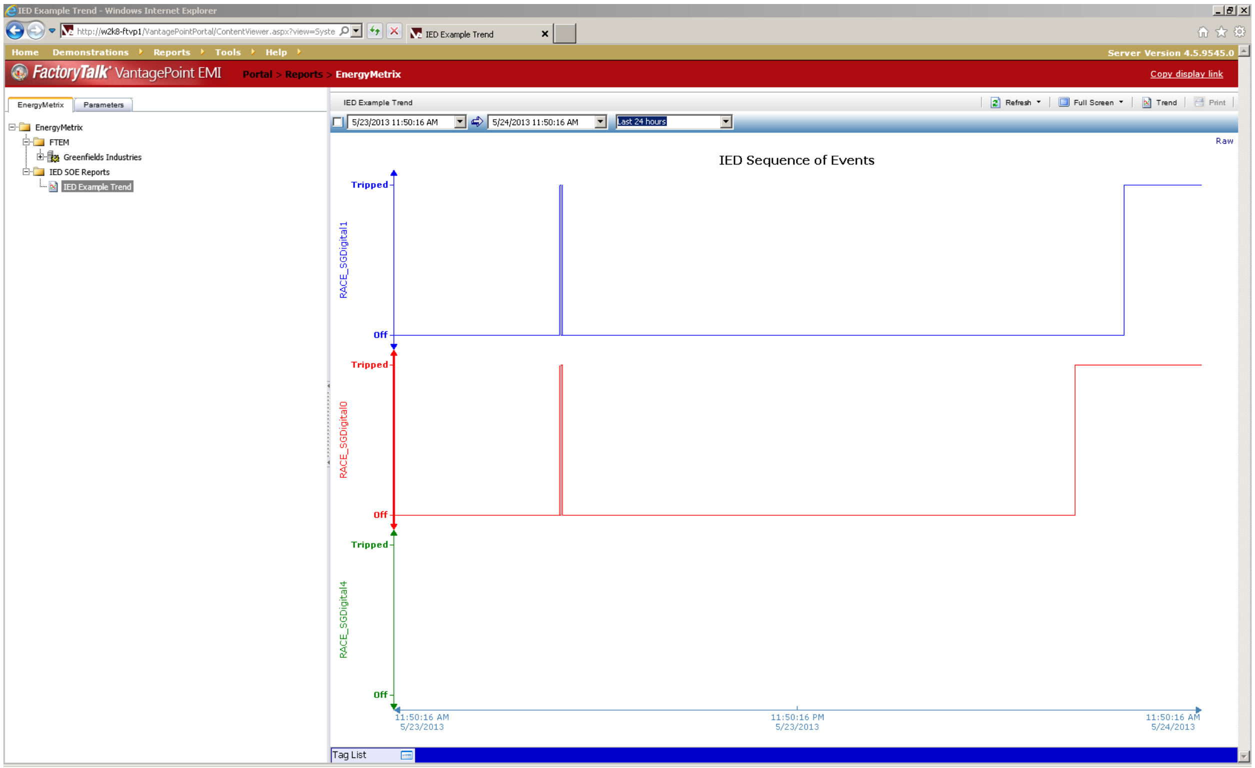


Figure 6.28 IED SOE Data on Vantage Point

Mining Example

A typical one-line diagram of a processing facility is shown in Figure 6.29. For this example, the utility provides 138 kV to the processing facility. From this point, the facility has two 13.8 kV feeds in a main-tie-main configuration, each with its own transformer. The 13.8 kV distribution voltage is then stepped down to 4.16 kV to feed individual loads. The 4.16 kV busses are also configurable in a secondary-selective, main-tie-main scheme that can be implemented in case of bus failure. This configuration is often preferred in large mining operations since it allows critical loads to be transferred to healthy busses during times of an electrical system fault or insufficient spinning reserve. Many relays are utilized for operating and protecting these facilities, including instantaneous and time delay overcurrent (50/51 elements), differential (87 element), under-voltage (27 element), and others.

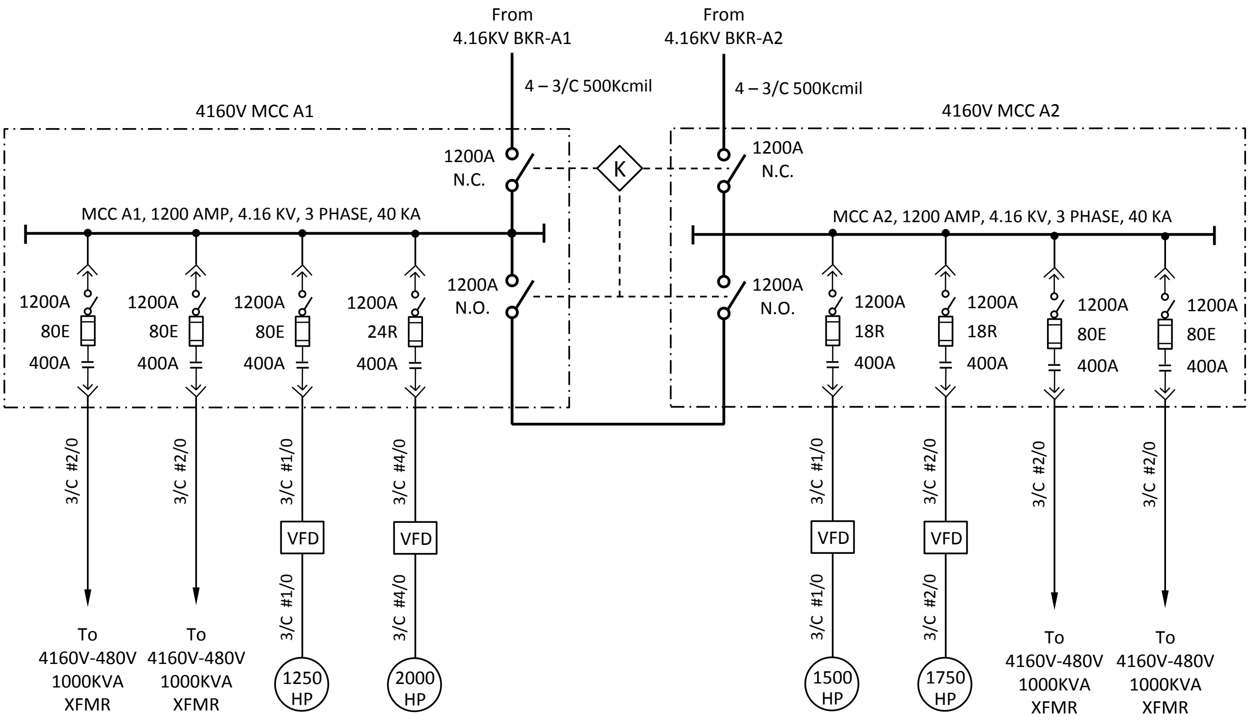


Figure 6.29 Main Tie Main Scheme

The equivalent HMI system for control of the scheme depicted in Figure 6.29 is shown in Figure 6.30. The loads, in this case rotating machines, are protected by 710 motor protection relays, while the feeders are protected by 751A relays. Any co-generation is protected by a generation relay, and the transformer is protected by a differential relay. Each of the white boxes displays the corresponding faceplate for monitoring, as well as command and control of the loads from a remote location. Figure 6.31 shows the Alarm and Events banner, with an example of the generator relay tripping as defined by the red box in the figure.

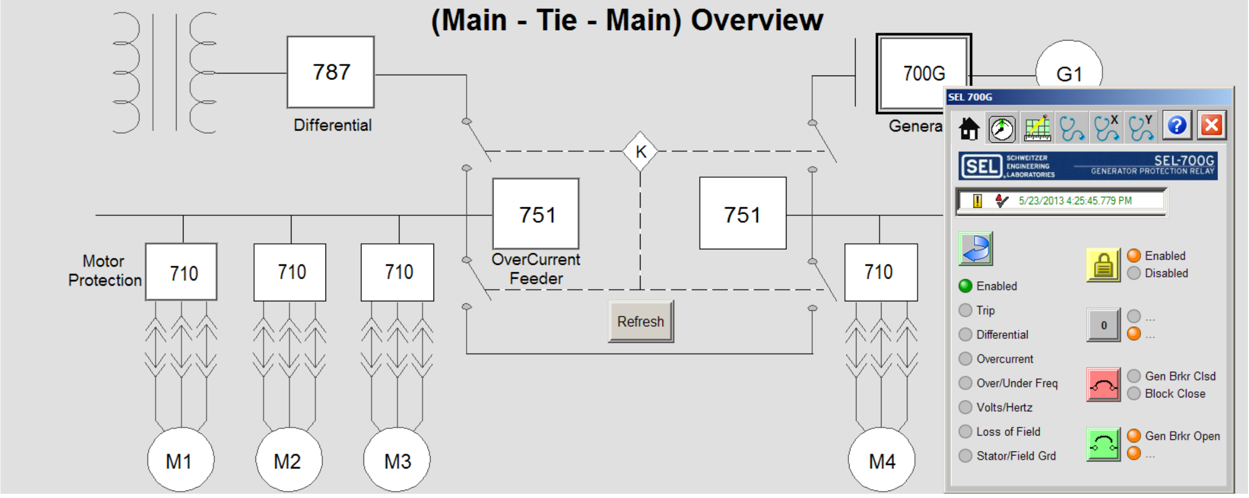
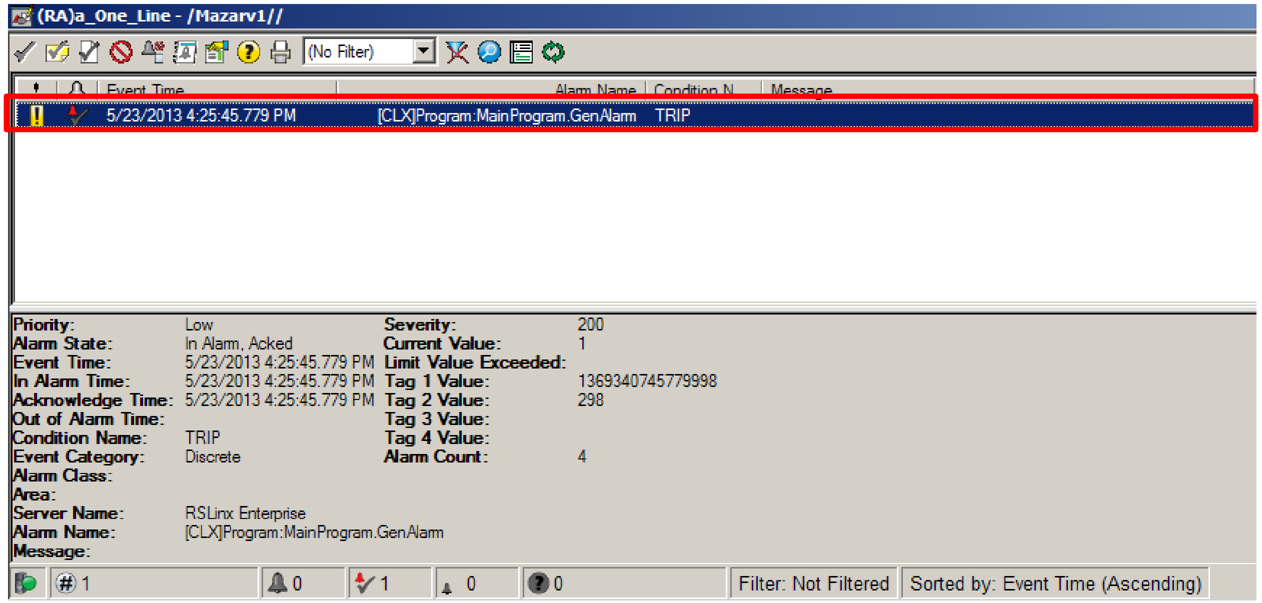


Figure 6.30 HMI Main Tie Main



5/23/2013 4:25:45.779 PM [CLX]Program:MainProgram.GenAlarm TRIP

Priority: Low
Alarm State: In Alarm, Acked
Event Time: 5/23/2013 4:25:45.779 PM
In Alarm Time: 5/23/2013 4:25:45.779 PM
Acknowledge Time: 5/23/2013 4:25:45.779 PM
Out of Alarm Time:
Condition Name: TRIP
Event Category: Discrete
Alarm Class:
Area:
Server Name: RSLinx Enterprise
Alarm Name: [CLX]Program:MainProgram.GenAlarm
Message:

Severity: 200
Current Value: 1
Limit Value Exceeded:
Tag 1 Value: 1369340745779998
Tag 2 Value: 298
Tag 3 Value:
Tag 4 Value:
Alarm Count: 4

1 | 0 | 1 | 0 | 0 | Filter: Not Filtered | Sorted by: Event Time (Ascending)

Figure 6.31 Alarm and Events Banner

Conclusions

This chapter described the visualization developed for this research. Items discussed included faceplate definition, human machine interface discussion and definition, and data management solution. The goal of visualization in this research was to represent an IED in logic and graphics which produced a virtual faceplate that would be familiar to users accustomed to a particular IED. The virtual faceplate gives operators and engineers the same look and feel they experience with the physical device. At the same time, the graphics were developed with various levels of security to allow only users with proper credentials access to various command and control functions.

Additionally, the research solution adhered to various graphics standards for both power and process control systems. The visualization trending functionality of the solution was then discussed by providing an example of a trend from a process historian repository. Finally, an example of a secondary selective main-tie-main scheme was presented for a processing facility. The one-line diagram was converted to an HMI screen, and multiple faceplates were tied to various instrumentation blocks in the diagram to enable remote monitoring and control of the system.

Chapter 7 - Verification and Acceptance Testing

Introduction

This chapter discusses the testing and verification procedures conducted to validate and verify functionality of the proposed solution. Since a goal of this research is to develop a mine monitoring solution utilizing the IEC 61850 standard to link the electrical distribution system with the process control system, it was determined that the best way to validate the work and functionality is to benchmark gateway performance tests against the conceptual design. The chapter discusses each conceptual design milestone as well as the procedures used to validate that this goal was met. For milestones requiring more than just visual inspection, the experimental setup is defined, testing procedures are described, and the experimental results discussed.

Conceptual Design Milestones

The original conceptual design of the solution proposed in this research was discussed in Chapter 3. After surveying multiple industrial firms ranging from EPCs to heavy industry segments, and comparing the feedback against information gathered in the literature review, this research produced the results shown in Table 7.1. Table 7.1 shows seven major milestones categories that were created for this research which include packaging, communications media, device support, safety, security, display information, and visualization information.

Many of the physical milestones can be validated by simple inspection. For example, packaging and media communications can be validated easily by examining a standalone gateway module. In addition to inspection, further testing and validation needs to be performed to benchmark the module's performance in various situations including being lightly loaded, heavily loaded, and how it would react in dynamic situations.

The benchmarks tested in this chapter are the self-imposed "Final Design" specifications shown in Table 7.1. Items that can be validated by inspection are qualified in paragraph form in the next few sections, while testing procedures are thoroughly described throughout the remainder of this chapter.

Table 7.1 Final Conceptual Design Milestones

Parameter	Original Design	Final Design
Packaging	Stand-Alone	Stand-Alone
Communications Media	Copper Min. 100 Mbit/s	Copper Min. 100 Mbit/s
Device Support	Min. 20 device	20 devices
Safety	Interlocking	Interlocking, Select Before Operate, Classified Areas Considerations
Security	Min. 5 levels	Min. 7 levels
Display Information	Alarms, Targets, Measurands	Alarms, Targets, Measurands, Timing / SOE, Historian Interfacing Capability
Visualization Information	N/A	ISA 5.5 Compliance, Provide Diagnostic Information, Command and Control

Packaging & Communications Media

Once creating the final conceptual design specification, it was determined that a standalone gateway module is the most versatile solution for linking the power distribution system to the process control system. Additionally, in order to be functional with most of the existing infrastructure, copper Ethernet media was determined to be the best form of data transfer. Figure 7.1 shows a sample experimental system consisting of a controller, unmanaged Ethernet switch, gateway module, 24 VDC power supply, and feeder protection relay. The gateway module in Figure 7.1 is a standalone device (blue module) which is powered by the 24 VDC power supply.

Additionally, Figure 7.1 shows that the gateway module uses an RJ-45 copper media Ethernet jack as the communications media. As defined in the hardware section, hardware for the communications PHY was selected for 100 Mbit/s TCP/IP Ethernet communications, thus satisfying the conception design milestone. As a result the packaging and media communications milestones were successfully met.

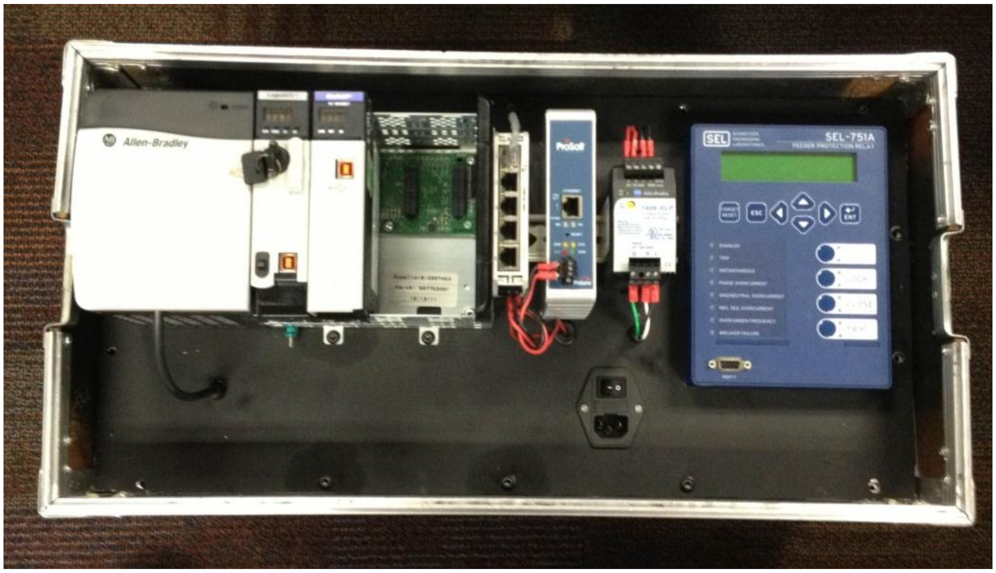


Figure 7.1 Sample System Setup

Visualization Information

ISA 5.5 Compliance

After developing the final conceptual design specifications for visualization information, it was important to ensure that any information displayed on process screens would be displayed in a standard, acceptable fashion. The Instrumentation Society of America (ISA) defines proper visualization in its 5.5 standard, entitled “Graphic Symbols for Process Displays.”

This standard was chosen since many mining facilities follow these guidelines as it is similar to MSHA standards. One problem with some of the graphics developed for this research is that power distribution and control often use color schemes opposite that of their process counterparts. For example, the breaker closure is often represented as red, while the breaker opening is often represented as green in power systems, whereas the breaker closure represents the start of a process and breaker closure represents the stopping of a process. The IEEE Industry Application Society Industrial and the Commercial Power Systems (IEEE IAS I&CPS) color schemes were chosen for power since their safe, non-energized states are often referred to with a green color, while their energized or possible danger states are symbolized by red.

All symbols used within the graphical faceplates for visualization follow ISA 5.5 guidelines, with the following exceptions shown in Figure 7.2. The symbols shown in Figure 7.2 are not defined in the ISA 5.5 symbols library and were added by exception to allow for full functionality of the solution developed as part of this research. are all NEMA symbols that are utilized in the power distribution realm and follow traditional power systems color schemes as defined by the SCADA standard IEEE C37.1.

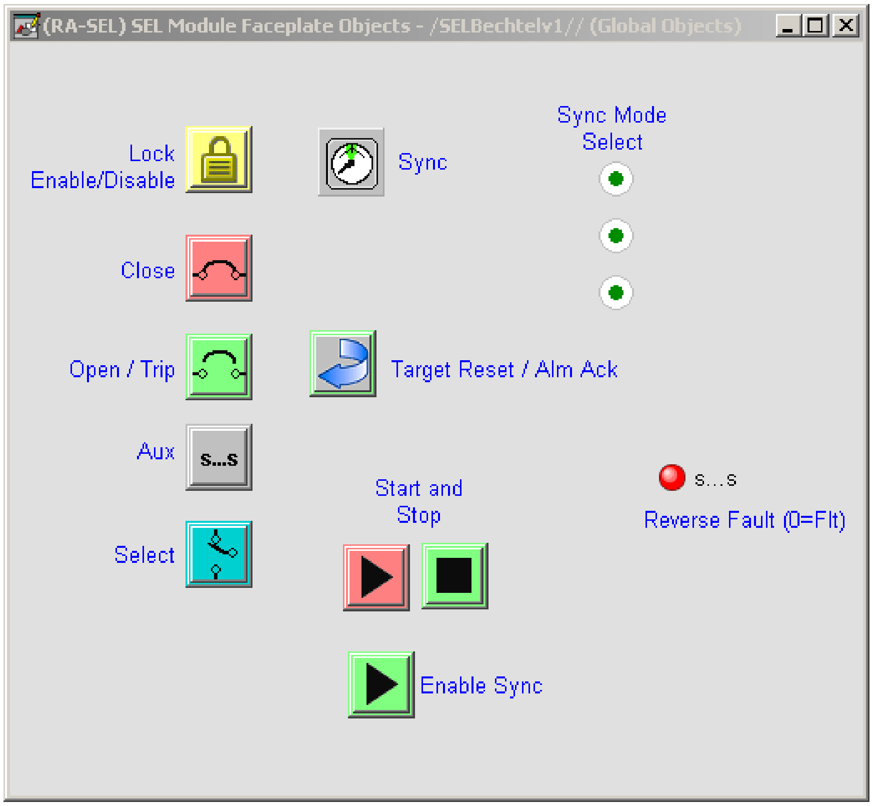


Figure 7.2 ISA Symbol Exceptions

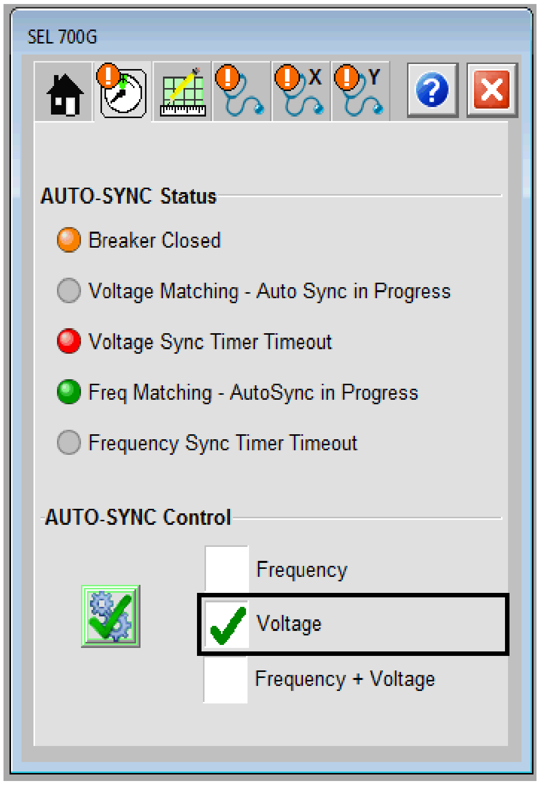
Diagnostic Information

According to the IEC 61850 standard, diagnostic information is kept in the quality fields of each relay. If the user decides to map the quality information from any parameters, these values can be monitored for IED health. In addition, information about potential and current transformers can also be brought back from the system. All of these quality flags from the IED are transferred to the controller where they are constantly monitored for bad data. These quality flags are then logically “ORed” together to determine if there was a diagnostic problem with the relay. If there was a problem, an alarm was generated, and the corresponding faceplate was populated with a diagnostic alarm to alert the operator that there are bad data coming from the relay.

Command and Control

Utilizing the benefits of the two-way communications specified by the IEC 61850 standard, command and control was implemented into this research solution. By utilizing the ability to map remote bits from IEDs to the controller, the operator or engineer can perform various actuations from the relay, including synchronizing a generator, opening or closing a breaker, and starting or stopping a rotating machine.

Figure 7.3 depicts the synchronization screen for a generator relay on the visualization faceplate. In this example, the user has selected the mechanism for synchronization: frequency, voltage, or frequency and voltage. Once the user selects the mechanism and depresses the sync button, the corresponding remote bits are set, and the operator receives diagnostic information from the relay as the synchronization process occurs.



AUTO-SYNC Status

● Breaker Closed

○ Voltage Matching - Auto Sync in Progress

● Voltage Sync Timer Timeout

● Freq Matching - AutoSync in Progress

○ Frequency Sync Timer Timeout

AUTO-SYNC Control

□ Frequency

☑ Voltage

□ Frequency + Voltage

Figure 7.3 Generator Synchronization Screen

Table 7.2 shows the command and control functions that were tested for various faceplates. Each of these features was tested 50 times with the operations of the corresponding graphical faceplate. The tests resulted in a 100% success rate for all command and control functionality with the exception of the 700G relay. Unfortunately at the time of testing, no generator was available to test the functionality of the synchronization features of the 700G relay. In the case of these tests, the relay failed on synchronization timeout for each of the 50 trials conducted.

Table 7.2 Tested Command and Control Scenarios

IED	Function
751A	Lock
751A	Unlock
751A	Open
751A	Close
700G	Lock
700G	Unlock
700G	Open
700G	Close
700G	Sync
710	Lock
710	Unlock
710	Start
710	Stop
787	Lock
787	Unlock
787	Select
787	Open
787	Close

Security

The original security specifications called for five distinct levels for the system, but after industry consultation, the final conceptual design called for seven levels of security to be built into the HMI graphics. The intent for levels of security is that those without proper credentials cannot/are not able to perform command and control of the electric distribution system. The solution developed in this research creates the possibility of safety risks if engineers or personnel with improper credentials make unauthorized changes to the mine power system during operation. As a result, seven levels of security were implemented to prevent unauthorized actuation of power equipment. The seven levels of security implemented in the design solution can be seen in Table 7.3. Tables 7.4-7.7 show the permissions for each security level.

Table 7.3 Security Levels

Security Level	Description
1	Administrator
2	Engineer II
3	Engineer I
4	Senior Operator
5	Operator
6	Maintenance
7	Remote Access

In order to test the integrity of the security built into the system, various trials were run on each developed relay faceplate in order to ensure that no level could access data or command and control actuation of a device for which it was not authorized. Tables 7.4-7.7 describe the testing conducted on each relay faceplate with descriptions of each test case. A red ‘X’ indicates that access should not be and was not permitted, while a green check indicates that access was granted for command and control or viewing of information. Each case was tested 50 times and included closing the faceplate and opening it again for consistency. The results shown in Tables 7.4-7.7 represent a 100% successful testing of each test case.

Table 7.4 Feeder Relay Security Features

	Description						
Security Level	Home Screen	Engineering Screen	Diagnostics	Target Reset	Lock/Unlock	Breaker Open	Breaker Close
Administrator	✓	✓	✓	✓	✓	✓	✓
Engineer II	✓	✓	✓	✓	✓	✓	✓
Engineer I	✓	✓	✓	✓	✓	✓	✗
Senior Operator	✓	✓	✓	✗	✓	✓	✗
Operator	✓	✓	✓	✗	✗	✓	✗
Maintenance	✓	✓	✓	✗	✓	✓	✗
Remote Access	✓	✓	✗	✗	✗	✗	✗

Table 7.5 Motor Relay Security Features

	Description					
Security Level	Home Screen	Engineering Screen	Diagnostics	Target Reset	Motor Stop	Motor Start
Administrator	✓	✓	✓	✓	✓	✓
Engineer II	✓	✓	✓	✓	✓	✓
Engineer I	✓	✓	✓	✓	✓	✗
Senior Operator	✓	✓	✓	✗	✓	✗
Operator	✓	✓	✓	✗	✓	✗
Maintenance	✓	✓	✓	✗	✓	✗
Remote Access	✓	✓	✗	✗	✗	✗

Table 7.6 Differential Relay Security Features

	Description							
Security Level	Home Screen	Engineering Screen	Diagnostics	Target Reset	Lock/Unlock	Select Breaker	Breaker Open	Breaker Close
Administrator	✓	✓	✓	✓	✓	✓	✓	✓
Engineer II	✓	✓	✓	✓	✓	✓	✓	✓
Engineer I	✓	✓	✓	✓	✓	✗	✓	✗
Senior Operator	✓	✓	✓	✗	✓	✓	✓	✗
Operator	✓	✓	✓	✗	✗	✗	✓	✗
Maintenance	✓	✓	✓	✗	✓	✗	✓	✗
Remote Access	✓	✓	✗	✗	✗	✗	✗	✗

Table 7.7 Generator Relay Security Features

	Description							
Security Level	Home Screen	Engineering Screen	Diagnostics	Synchronization	Target Reset	Lock/Unlock	Breaker Open	Breaker Close
Administrator	✓	✓	✓	✓	✓	✓	✓	✓
Engineer II	✓	✓	✓	✓	✓	✓	✓	✓
Engineer I	✓	✓	✓	✓	✓	✓	✓	✗
Senior Operator	✓	✓	✓	✗	✗	✓	✓	✗
Operator	✓	✓	✓	✗	✗	✗	✓	✗
Maintenance	✓	✓	✓	✗	✗	✓	✓	✗
Remote Access	✓	✓	✗	✗	✗	✗	✗	✗

Display

When reviewing the final conceptual design for the solution proposed in this research, four distinct target milestones were established:

1. Display Alarms and Targets
2. Display Measurands
3. Display Sequence of Events Data
4. Provide Historian Interfacing Capability

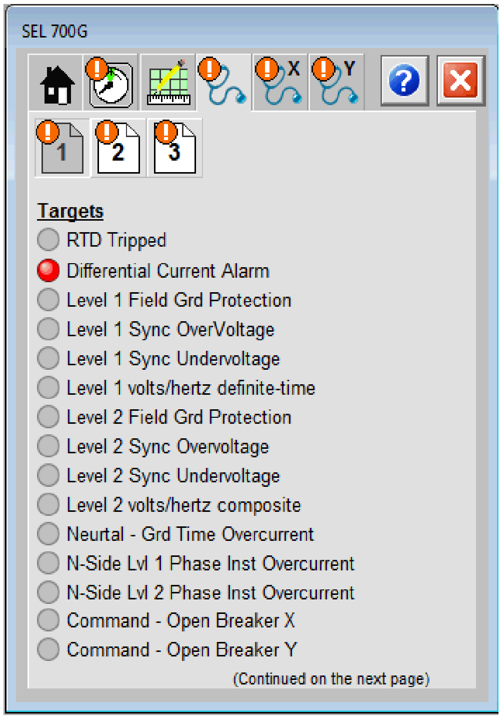
Each of these milestones are discussed in this section as well as how the information is displayed to the benefit of the operator or engineer.

Alarms and Targets

The first and most important functionality of IEDs is electrical protection of the devices in the system. With this in mind, IEDs must be able to provide alarms and warnings to operators and engineers to alert them that there is an electrical fault or other anomaly in their mining system that could harm personnel or damage equipment. Prior to microprocessor-based relays, SCADA systems were alerted by the closure of normally open and normally closed contacts on the relays to make or break alarming circuits from the device.

Unfortunately, this method only provided information that the relay had picked up or alarmed, but no diagnostic information as to why the relay had picked up or alarmed. As a result, post mortem analysis of why faults occurred was time consuming, thus affecting process yield due to downtime. With microprocessor based relays, more diagnostic information (such as the reason for picking up) can be provided by the IED, but there is still no easy way in a distributed system to retrieve this information without being directly plugged into the front port of the relay. The solution provided in this research allows trips, alarms, and warnings to be graphically displayed in a centralized location via the use of the graphical faceplates described in Chapter 6.

The diagnostic tab in each relay faceplate provides all of the diagnostic information for alarms or targets that may pick up during facility operation. This screen can be seen in Figure 7.4, while Figure 7.5 displays the generic targets on the home screen of the faceplate for easy identification , in the event the relay picked up, and a summary view of why the relay picked up.

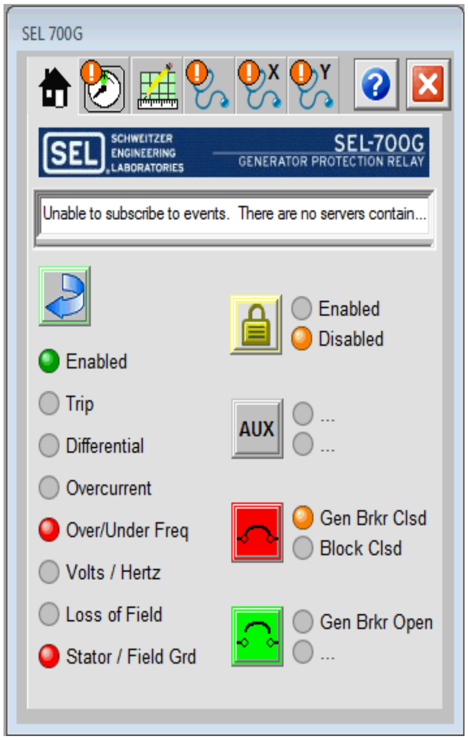


Targets

○ RTD Tripped
● Differential Current Alarm
○ Level 1 Field Grd Protection
○ Level 1 Sync OverVoltage
○ Level 1 Sync Undervoltage
○ Level 1 volts/hertz definite-time
○ Level 2 Field Grd Protection
○ Level 2 Sync Overvoltage
○ Level 2 Sync Undervoltage
○ Level 2 volts/hertz composite
○ Neurtal - Grd Time Overcurrent
○ N-Side Lvl 1 Phase Inst Overcurrent
○ N-Side Lvl 2 Phase Inst Overcurrent
○ Command - Open Breaker X
○ Command - Open Breaker Y

(Continued on the next page)

Figure 7.4 Diagnostic Screen Generator Relay


SEL 700G
X
Y
SEL SCHWEITZER ENGINEERING LABORATORIES
SEL-700G
GENERATOR PROTECTION RELAY
AUX

Unable to subscribe to events. There are no servers contain...

○ Enabled
● Disabled

● Enabled
○ Trip
○ Differential
○ Overcurrent
● Over/Under Freq
○ Volts / Hertz
○ Loss of Field
● Stator / Field Grd

○ ...
○ ...

● Gen Brkr Clsd
○ Block Clsd

○ Gen Brkr Open
○ ...

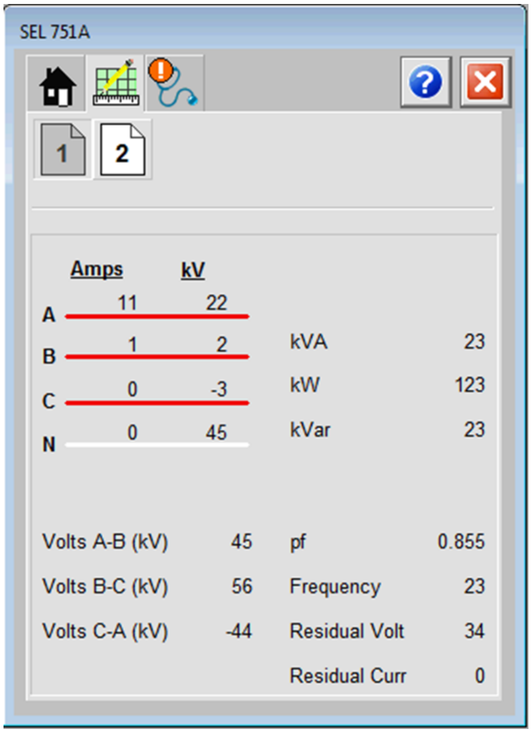
Figure 7.5 Generator Relay Target Overview

Each of the relay targets was tested via simulation software by forcing the target to generate a trip signal based on computer forced command. Each target was tested five times in conjunction with the relay faceplate and checked whether there was agreement between the controller data table point and the faceplate graphic tag. The result of this testing indicates that all relay pickup points were mapped correctly and operated correctly.

Measurands

With the advancement of microprocessor-based relays, individual measurands, such as voltage and current, can now be gathered by the IED. Measurands for the solution proposed in this research for mine monitoring and control play a major role in indicating the health of the electric distribution system as well as the electromechanical equipment. These fundamental measurands, both electrical and thermal, are important for establishing the long term health and conducting the predictive maintenance of a system. As a result, these fundamental values are included in the visualization portion of the relay faceplates, as this information, when provided at a centralized facility from a distributed system, allows engineers to quickly asses the overall performance of an operation. Figures 7.6 and 7.7 depict the measurands that are displayed during the operation of the mine system.

In order to simulate values going to the relays, the IEDs generator, feeder, differential, and motor were connected to a variable voltage source with a motor load. Voltages and loading were then adjusted in order to provide for a range of electrical measurands, ensuring that the relay updated these values accordingly. In addition, faceplate graphics were checked to ensure they were mapped correctly to controller tags, thus appropriately updating from information in the controller data table. It was determined after testing each relay and faceplate that the information was correctly mapped and was being correctly transferred from IED to PAC.

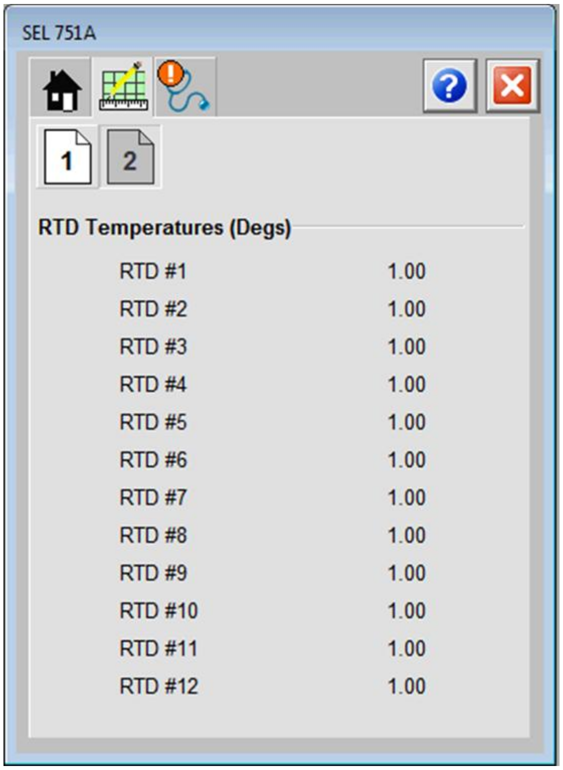


	Amps	kV
A	11	22
B	1	2
C	0	-3
N	0	45

kVA	23
kW	123
kVar	23

Volts A-B (kV)	45	pf	0.855
Volts B-C (kV)	56	Frequency	23
Volts C-A (kV)	-44	Residual Volt	34
		Residual Curr	0

Figure 7.6 Measurands Electrical



RTD Temperatures (Degs)

RTD #1	1.00
RTD #2	1.00
RTD #3	1.00
RTD #4	1.00
RTD #5	1.00
RTD #6	1.00
RTD #7	1.00
RTD #8	1.00
RTD #9	1.00
RTD #10	1.00
RTD #11	1.00
RTD #12	1.00

Figure 7.7 Measurands Thermal

Timing and Sequence of Events

With the advent of GPS timing, all IEDs from a distributed system can be referenced and synchronized to the same time clock or reference to time. This allows for easier analysis of cascading events in an electrical distribution system, and for reference to when events occurred to an ultimate time source. Two places needed to be tested to ensure that timestamps were properly passed throughout the system. The first place was between the IED and the automation controller, and the second was between the automation controller and HMI software.

Timing between IED and PAC

The first step in validating proper time transfer was to establish a network of IEDs, a gateway module, and controllers for receiving data from the IEDs. This setup is shown in Figure 7.8. The relays were configured to pick up every two seconds and reset automatically. When the relays picked up, they were configured to send an IEC 61850 report, with corresponding timestamp, to the gateway module, which in-turn published the information to the automation controller.

First the timestamp between the IED sequence of events record (SER) and published IEC 61850 report was checked for having the same value. It was determined that timestamps from trips were assigned in the same processing cycle, leaving a time difference of zero microseconds between the actual protection element assertion time and the relay trip published time. This point was later found as a part of the IEC 61850 annex defining that all IED manufactures should follow this procedure to provide the most accurate information to SCADA systems.

The reported trip time was then compared against the data table LINT timestamp. It was determined after 1000 tests of all four relays that the time difference between the IED timestamp and controller data table timestamp was zero microseconds, thus meeting the milestone of preserving the timestamp from the IEC 61850 network to the EtherNet/IP network.

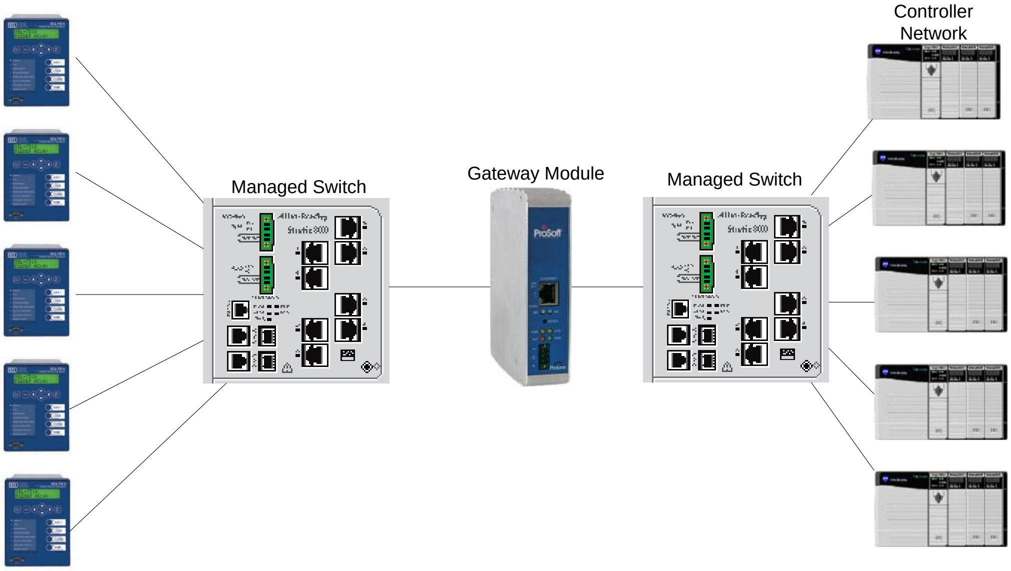


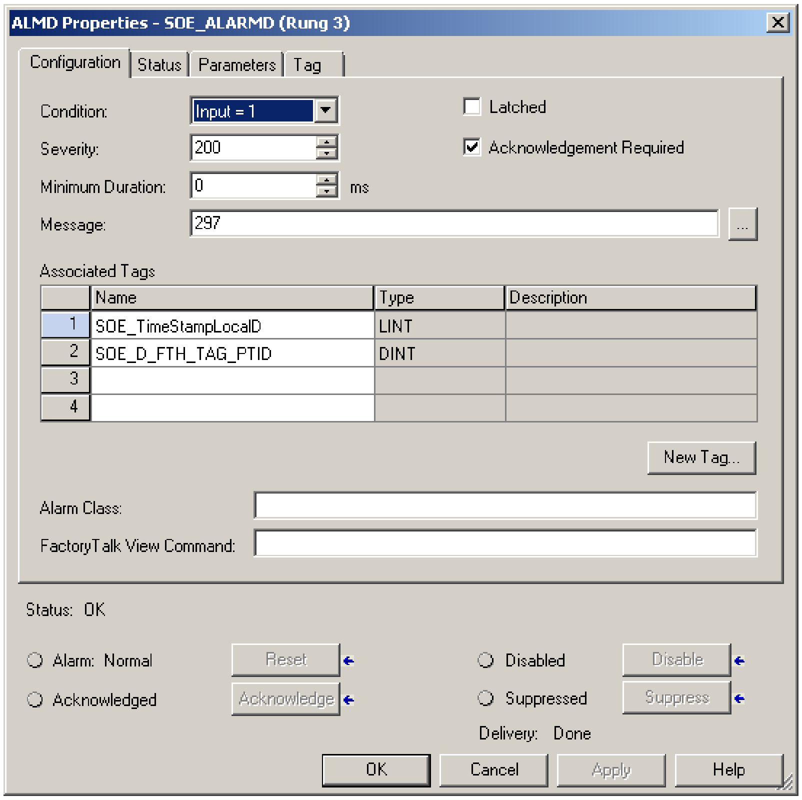
Figure 7.8 Timing Test Setup

Timing between PAC and HMI

The intent of this solution is to provide a time scale that is as accurate as possible, given current industry products and the capability of the system hardware. The ControlLogix controller is capable of using a 64bit Integer (LINT) that represents the number of micro-seconds from Jan 1, 1970. This allows for a date timestamp to be held in either two registers (DINTS) or one (LINT).

The FactoryTalk Alarms and Events server software is capable of storing, in a Microsoft SQL Server Database, the event time at a resolution of one millisecond. This resolution is due to the use of the DateTime data type. Unfortunately, at this time the DateTime field as specified by Microsoft only accommodates a 32-bit timestamp, or one millisecond. This can be seen in Figure 7.9. Microsoft is currently developing a DateTime2 field which represents the full 64-bit range, or one microsecond, resolution.

Although there is no direct way to time a 64-bit timestamp to an alarm, it can still be preserved within the alarm object. The object has four other tags storage locations with each Alarm Event, stored in the FTAE database. As a result of this data object definition being able to pass the LINT timestamp as an associated alarm tag, the timestamp can still be preserved in full form and passed to a process historian for data vaulting. This feature can be seen in Figure 7.9.



ALMD Properties - SOE_ALARMD (Rung 3)

Condition: Input = 1

Severity: 200

Minimum Duration: 0 ms

Message: 297

☐ Latched

☑ Acknowledgement Required

Associated Tags

	Name	Type	Description
1	SOE_TimeStampLocalD	LINT	
2	SOE_D_FTH_TAG_PTID	DINT	
3			
4			

Alarm Class:

FactoryTalk View Command:

Status: OK

○ Alarm: Normal

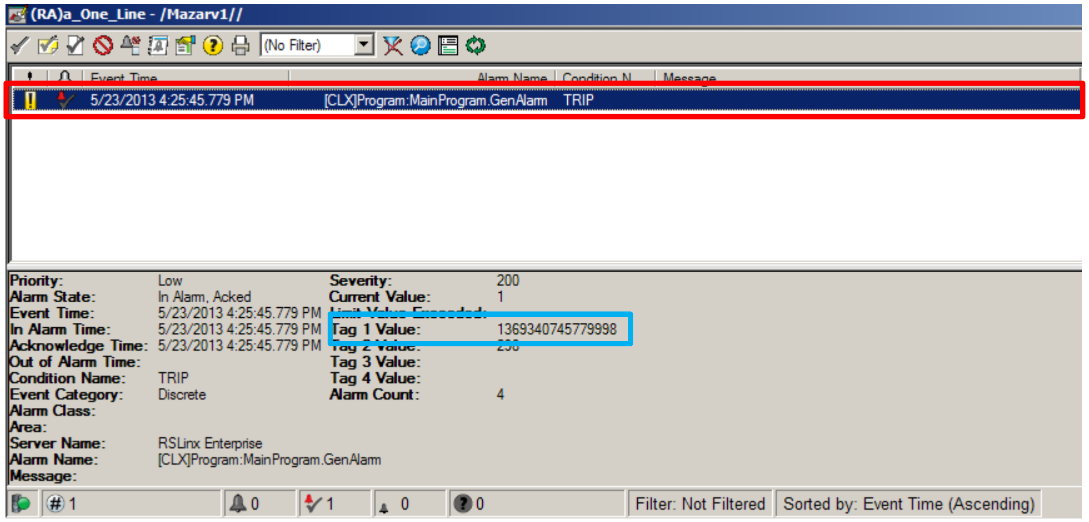
○ Acknowledged

○ Disabled

○ Suppressed

Delivery: Done

Figure 7.9 Mapping Associated LINT Timestamp



(RA)a_One_Line - /Mazarv1//

5/23/2013 4:25:45.779 PM [CLX]Program:MainProgram.GenAlarm TRIP

Priority:	Low	Severity:	200
Alarm State:	In Alarm, Acked	Current Value:	1
Event Time:	5/23/2013 4:25:45.779 PM	Limit Value Exceeded:	
In Alarm Time:	5/23/2013 4:25:45.779 PM	Tag 1 Value:	1369340745779998
Acknowledge Time:	5/23/2013 4:25:45.779 PM	Tag 2 Value:	298
Out of Alarm Time:		Tag 3 Value:	
Condition Name:	TRIP	Tag 4 Value:	
Event Category:	Discrete	Alarm Count:	4
Alarm Class:			
Area:			
Server Name:	RSLinx Enterprise		
Alarm Name:	[CLX]Program:MainProgram.GenAlarm		
Message:			

Filter: Not Filtered

Sorted by: Event Time (Ascending)

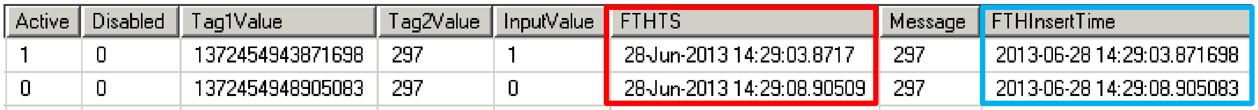
Figure 7.10 Alarm and Events Software Results

The result of mapping the timestamp to Tag 1 also can be seen in the blue box in Figure 7.10, where the timestamp represents the same time microsecond format. It was determined after 1000 tests of each relay that the LINT timestamp, represented in the number of microseconds located in Tag 1, matched exactly the timestamp in the data table of the PAC as well as the timestamp transferred from the IED network. As a result, the preservation of the time milestone was achieved for this solution.

Storing Values in Historian

As mentioned in Chapter 6, OSI PI historian does not accept foreign timestamps due to the way traditional process historians function. Chapter 6 also discussed a solution for how to address this issue with the creation of a software tool that bridges the Alarms and Events database and OSI PI historian database. In order to test the functionality of this software tool, 2000 alarms were created and tested on IEDs in a system.

As long as a timestamp inserted into the historian was within 15 microseconds of the IED published time, the tool was within tolerance and no further error propagated throughout the system. The timestamps that were compared are shown in Figure 7.11. The red timestamp corresponds to the timestamp published to the process historian, and the blue timestamp corresponds timestamp published by the IED network. As long as the time difference between the red and blue timestamps is less than or equal to 15 microseconds, the tool has no added delay outside the tolerance of current process historian systems. The results of the testing can be seen in Table 7.8 and Figure 7.12.



Active	Disabled	Tag1Value	Tag2Value	InputValue	FTHTS	Message	FTHInsertTime
1	0	1372454943871698	297	1	28-Jun-2013 14:29:03.8717	297	2013-06-28 14:29:03.871698
0	0	1372454948905083	297	0	28-Jun-2013 14:29:08.90509	297	2013-06-28 14:29:08.905083

Figure 7.11 Time Comparisons

Table 7.8 Numerical Time Error Results

																	Total
Time Error [us]	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Number of results	172	176	137	116	97	164	170	152	151	161	115	112	92	85	63	37	**2000**

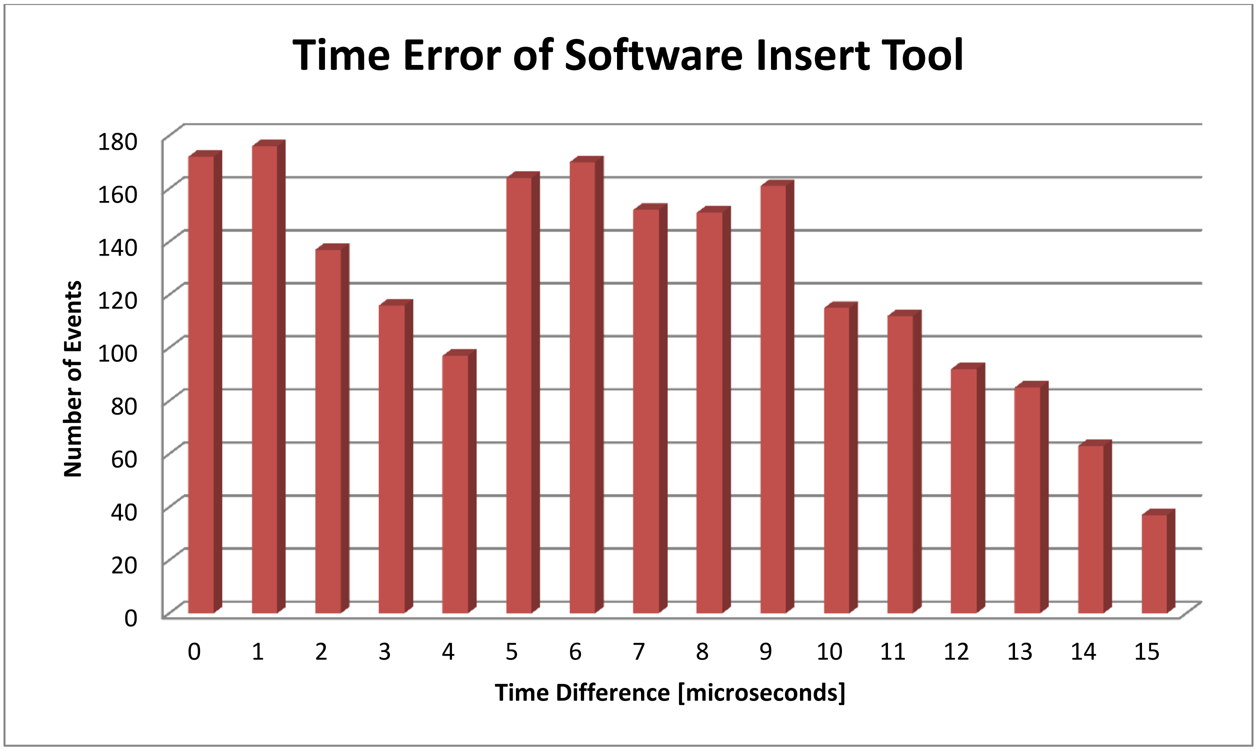


Figure 7.12 Graphical Time Error Results

Table 7.8 and Figure 7.12 show that, after 2000 test trials, all resulting data points are within the error bars of 15 microseconds. The data suggests that 74.7% of the results are within 9 microseconds of each other, well within the 15 microsecond tolerance provided by OSI PI.

In addition to the testing of time discrepancy, this test also checked whether fundamental parameters could be trended against electrical faults in the system. To check this, a second vendor's IED was tested as it had a built in simulator to test relay functionality and trip settings. This software can be seen in Figure 7.13. By adjusting the slider bars in the simulator, the currents, voltages, and harmonics of the corresponding waveforms were adjusted. The relay tripped when the set thresholds of these parameters were exceeded, thus triggering an alarm to be logged by the controller and historian system.

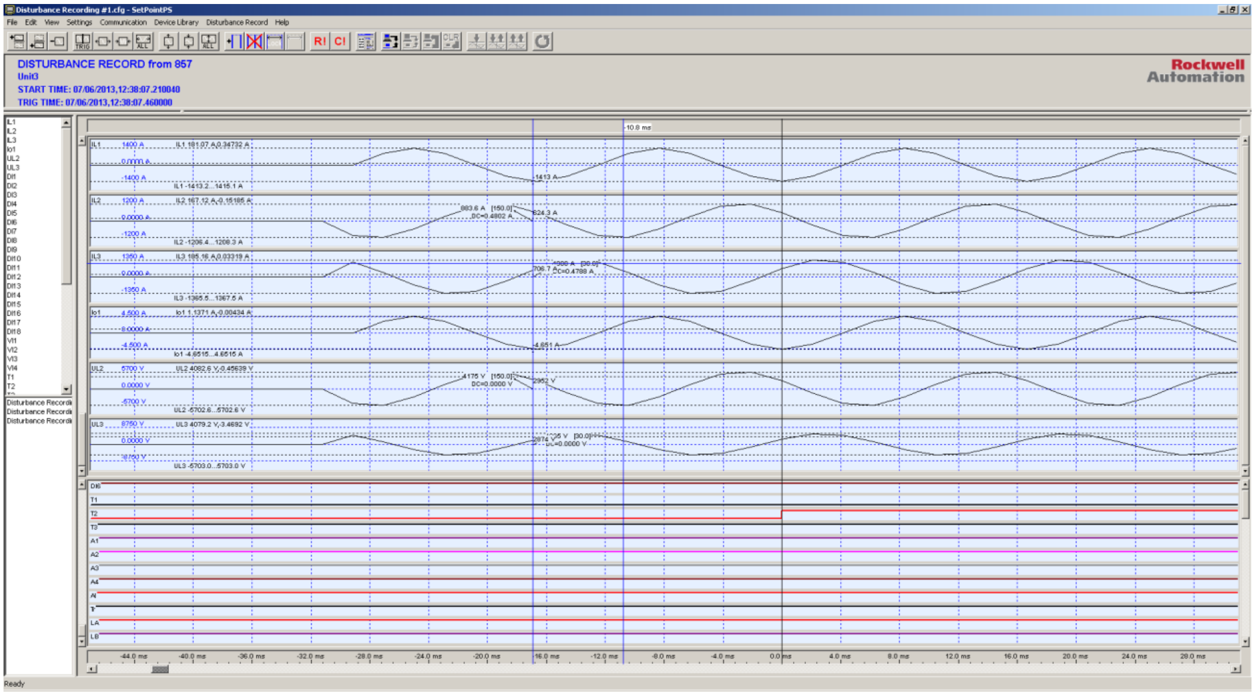
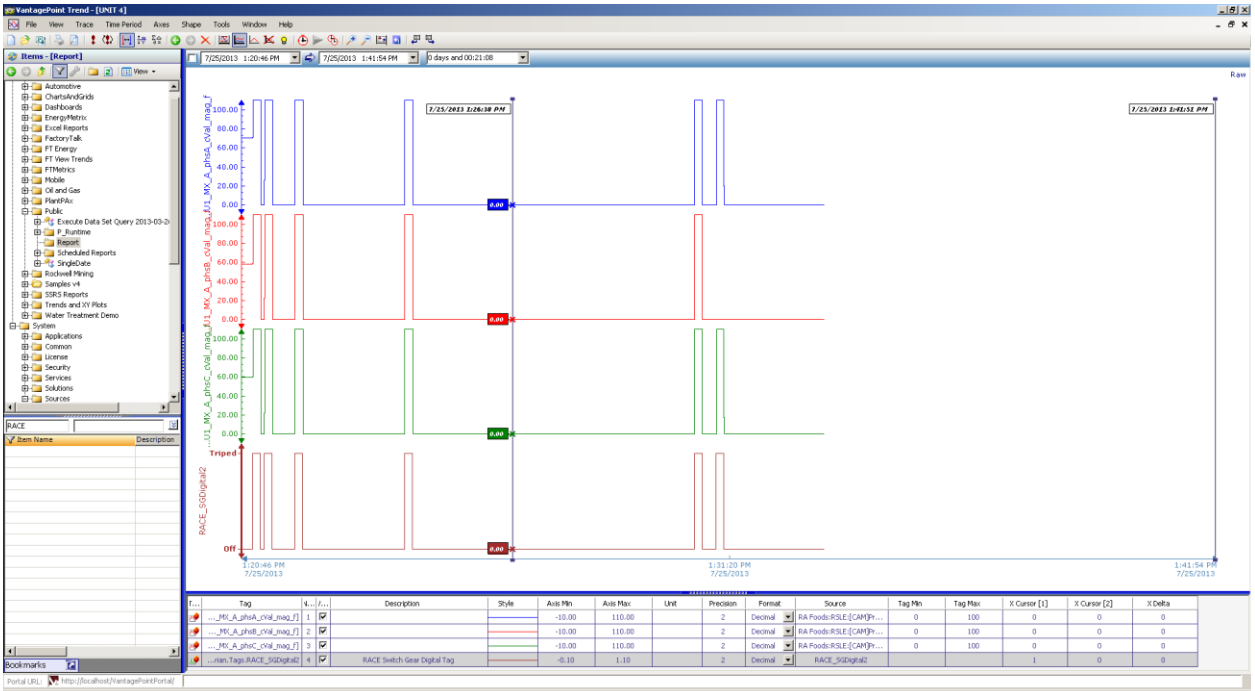


Figure 7.13 Software Simulator

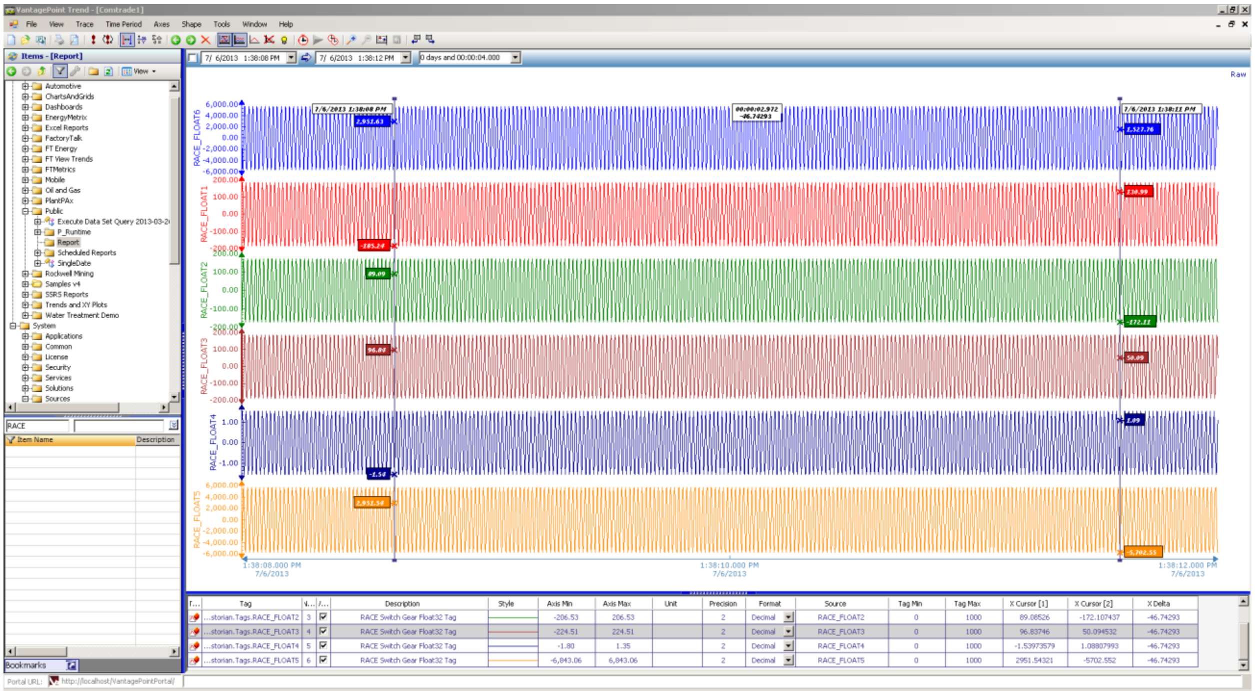
Information for analog parameters can also be logged and trended in historian as seen in Figure 7.14. In this figure, the maroon lines represent the trip status of the relay, which, in this example, was set to trip on a ground current greater than 60 A (green), regardless of the currents on phases A and B (blue and red). As shown in the figure, the maroon line is asserted any time the process parameter, i.e., ground current (green), exceeds 60 Amps and resets when the relay resets. In this example this occurs when the ground current is less than 60 Amps.

Analog waveforms can be depicted if the IED manufacturer allows this information to be sent over the IEC 61850 standard. These waveforms can be seen in Figures 7.15 and 7.16, respectively. Figure 7.15 shows waveforms over a four-second time interval, while Figure 7.16 shows those same waveforms zoomed to approximately eight electrical cycles.



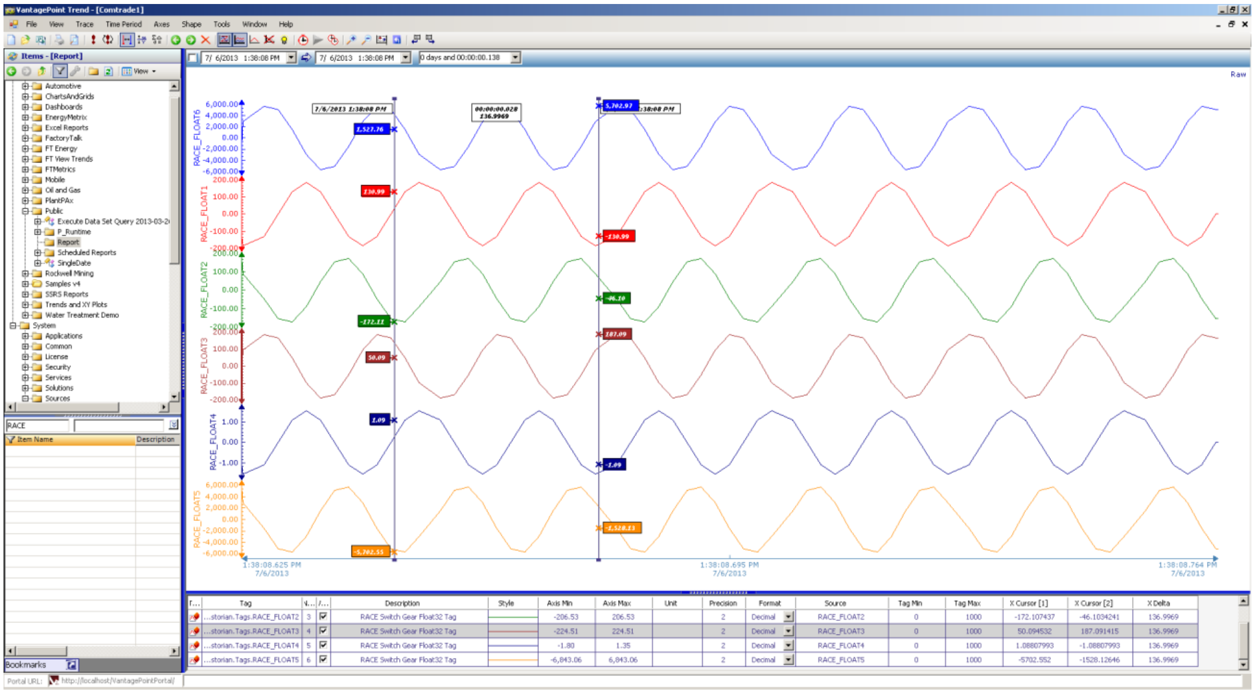
Γ...	Tag	N...	/...	Description	Style	Axis Min	Axis Max	Unit	Precision	Format	Source	Tag Min	Tag Max	X Cursor [1]	X Cursor [2]	X Delta
	..._MX_A_phsA_cVal_mag_f]	1	☑		—	-10.00	110.00		2	Decimal	RA Foods:RSLE:[CAM]Pr...	0	100	0	0	0
	..._MX_A_phsB_cVal_mag_f]	2	☑		—	-10.00	110.00		2	Decimal	RA Foods:RSLE:[CAM]Pr...	0	100	0	0	0
	..._MX_A_phsC_cVal_mag_f]	3	☑		—	-10.00	110.00		2	Decimal	RA Foods:RSLE:[CAM]Pr...	0	100	0	0	0
	...rian.Tags.RACE_SGDigital2	4	☑	RACE Switch Gear Digital Tag	—	-0.10	1.10		2	Decimal	RACE_SGDigital2			1	0	0

Figure 7.14 Trending of Analog and Electrical Parameters



Γ...	Tag	N...	/...	Description	Style	Axis Min	Axis Max	Unit	Precision	Format	Source	Tag Min	Tag Max	X Cursor [1]	X Cursor [2]	X Delta
	...storian.Tags.RACE_FLOAT2	3	☑	RACE Switch Gear Float32 Tag	—	-206.53	206.53		2	Decimal	RACE_FLOAT2	0	1000	89.08526	-172.107437	-46.74293
	...storian.Tags.RACE_FLOAT3	4	☑	RACE Switch Gear Float32 Tag	—	-224.51	224.51		2	Decimal	RACE_FLOAT3	0	1000	96.83746	50.094532	-46.74293
	...storian.Tags.RACE_FLOAT4	5	☑	RACE Switch Gear Float32 Tag	—	-1.80	1.35		2	Decimal	RACE_FLOAT4	0	1000	-1.53973579	1.08807993	-46.74293
	...storian.Tags.RACE_FLOAT5	6	☑	RACE Switch Gear Float32 Tag	—	-6,843.06	6,843.06		2	Decimal	RACE_FLOAT5	0	1000	2951.54321	-5702.552	-46.74293

Figure 7.15 Analog Waveforms



T...	Tag	N...	/...	Description	Style	Axis Min	Axis Max	Unit	Precision	Format	Source	Tag Min	Tag Max	X Cursor [1]	X Cursor [2]	X Delta
	...storian.Tags.RACE_FLOAT2	3	☑	RACE Switch Gear Float32 Tag		-206.53	206.53		2	Decimal	RACE_FLOAT2	0	1000	-172.107437	-46.1034241	136.9969
	...storian.Tags.RACE_FLOAT3	4	☑	RACE Switch Gear Float32 Tag		-224.51	224.51		2	Decimal	RACE_FLOAT3	0	1000	50.094532	187.091415	136.9969
	...storian.Tags.RACE_FLOAT4	5	☑	RACE Switch Gear Float32 Tag		-1.80	1.35		2	Decimal	RACE_FLOAT4	0	1000	1.08807993	-1.08807993	136.9969
	...storian.Tags.RACE_FLOAT5	6	☑	RACE Switch Gear Float32 Tag		-6,843.06	6,843.06		2	Decimal	RACE_FLOAT5	0	1000	-5702.552	-1528.12646	136.9969

Figure 7.16 Zoomed In Analogue Waveforms

Throughout the testing process design milestones were validated by successfully transferring information from the IED network to the PAC and from the controller to a process historian for storage and trending.

Device Support

In order to test quantity and throughput for device support, IEDs were instrumented with a packet analyzer known as “wire shark” which captures and transfers data. Wire Shark is a software network analyzer, recognized globally as an industry standard for determining connectivity, data rates, and network transfers. In order to obtain information about the network, a managed Ethernet switch was utilized. Managed switches have the ability to map an addition port as a span port. The span port mimics or becomes an exact copy of the port to which it is assigned to mirror. In this case, the switch port to which the gateway module was inserted was mirrored to another port. so that a computer running wire shark could capture the data. This setup is shown in Figure 7.17.

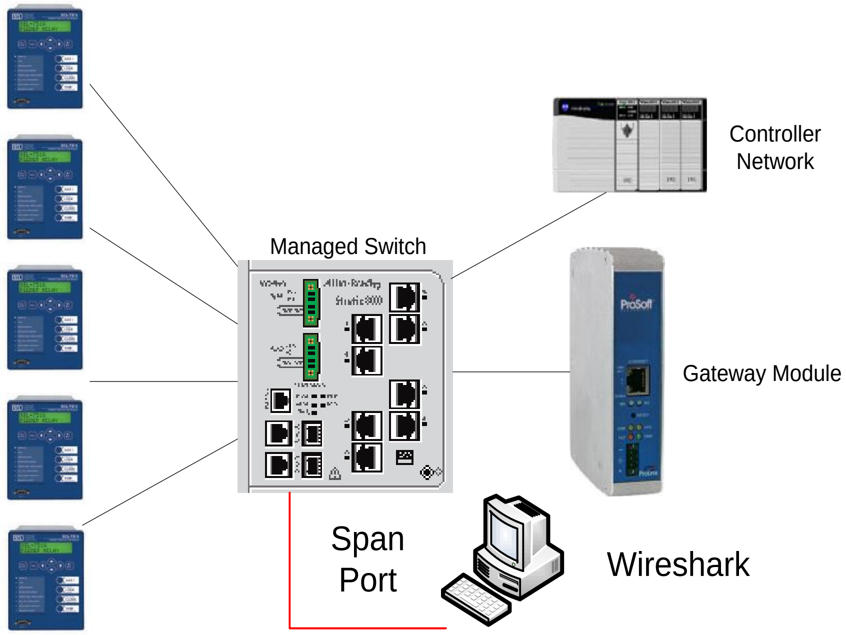


Figure 7.17 System Setup

For the first test, three relays were set up and configured with a generic report and configured to be sent to the gateway module on five-second intervals, much faster than any SCADA system uses for gathering data. The plot shown in Figure 7.18 depicts three types of data: relay data (blue), controller data (red), and gateway data (green).

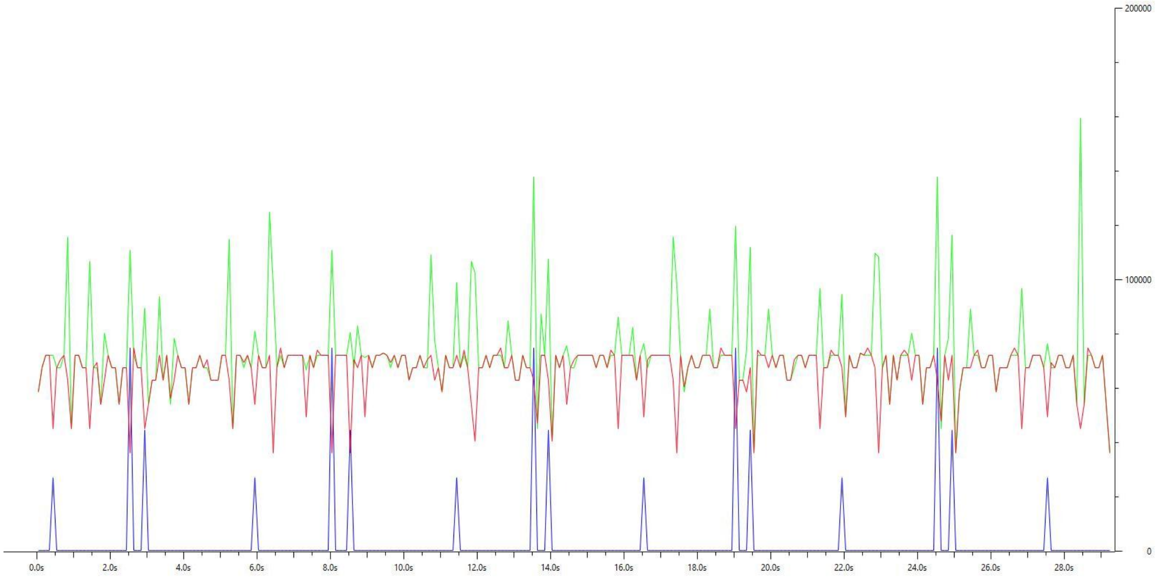
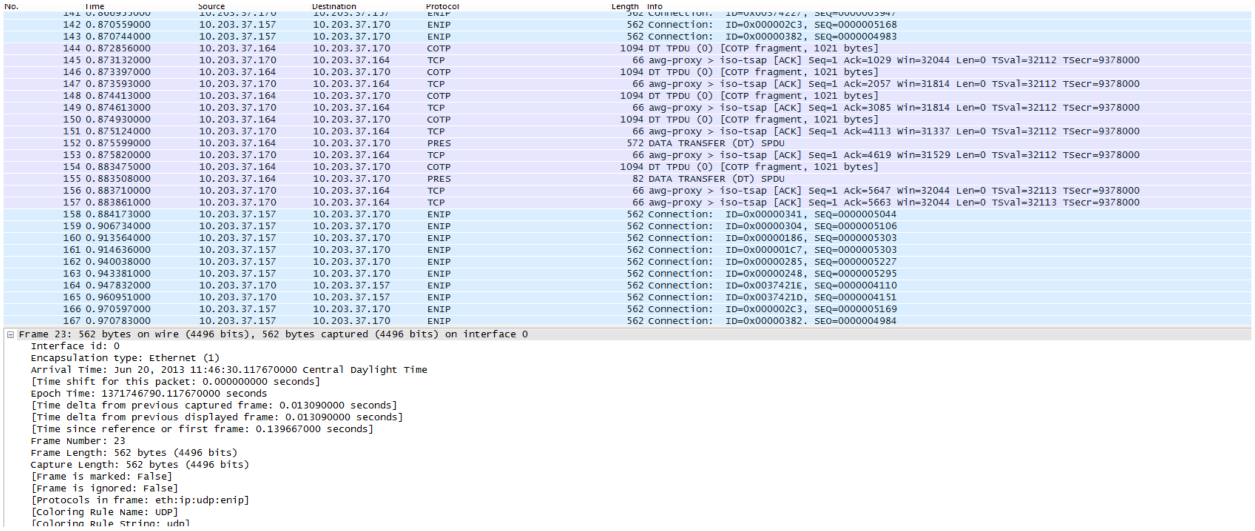


Figure 7.18 Data Flow in System

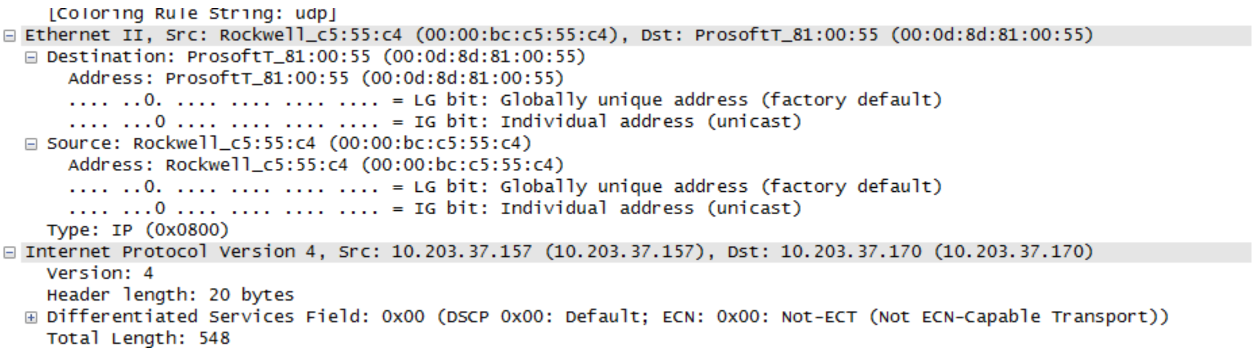
The blue plot in the diagram represents the IEC 61850 MMS data being transferred from the IED to the gateway module. There is a distinct pattern of the blue trace consisting of three distinct peaks, each representing a report being sent from an IED to the gateway module. The x-axis represents time in seconds of the data capture, and the y-axis represents the number of bits transferred. At approximately every 5 seconds the same pattern of three reports is generated and sent to the gateway module, with each report marked by a blue peak. As IEDs are designed first for protection and second for SCADA transfer, the timeframe is not exactly five seconds; however, the error is less than 300 milliseconds on data transfer rate. The data flow to the gateway module also spikes at these moments, as the green line spikes at the time of data transfer.

Figures 7.19-7.21 show wire shark screen captures that provide information on packet transfer between devices. Figure 7.19 shows information from the various frames captured by wire shark with the corresponding protocols of each message. Figure 7.20 shows specific information about an EtherNet/IP message, in this case between the gateway and the controller. Figure 7.21 shows IEC MMS communications between the IED and gateway module.



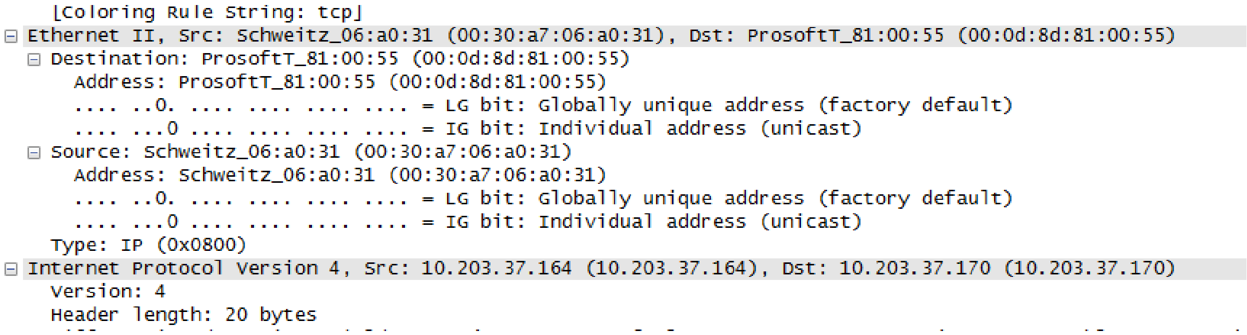
```
No.    Time          Source          Destination     Protocol        Length  Info
   141 0.866933000   10.203.37.170   10.203.37.157   ENIP               562 Connection:  ID=0x00374227, SEQ=0000003947
   142 0.870559000   10.203.37.157   10.203.37.170   ENIP               562 Connection:  ID=0x000002C3, SEQ=0000005168
   143 0.870744000   10.203.37.157   10.203.37.170   ENIP               562 Connection:  ID=0x00000382, SEQ=0000004983
   144 0.872856000   10.203.37.164   10.203.37.170   COTP              1094 DT TPDU (0) [COTP fragment, 1021 bytes]
   145 0.873132000   10.203.37.170   10.203.37.164   TCP                 66 awg-proxy > iso-tsap [ACK] Seq=1 Ack=1029 Win=32044 Len=0 TSval=32112 TSecr=9378000
   146 0.873397000   10.203.37.164   10.203.37.170   COTP              1094 DT TPDU (0) [COTP fragment, 1021 bytes]
   147 0.873593000   10.203.37.170   10.203.37.164   TCP                 66 awg-proxy > iso-tsap [ACK] Seq=1 Ack=2057 Win=31814 Len=0 TSval=32112 TSecr=9378000
   148 0.874413000   10.203.37.164   10.203.37.170   COTP              1094 DT TPDU (0) [COTP fragment, 1021 bytes]
   149 0.874613000   10.203.37.170   10.203.37.164   TCP                 66 awg-proxy > iso-tsap [ACK] Seq=1 Ack=3085 Win=31814 Len=0 TSval=32112 TSecr=9378000
   150 0.874930000   10.203.37.164   10.203.37.170   COTP              1094 DT TPDU (0) [COTP fragment, 1021 bytes]
   151 0.875124000   10.203.37.170   10.203.37.164   TCP                 66 awg-proxy > iso-tsap [ACK] Seq=1 Ack=4113 Win=31337 Len=0 TSval=32112 TSecr=9378000
   152 0.875599000   10.203.37.164   10.203.37.170   PRES               572 DATA TRANSFER (DT) SPDU
   153 0.875820000   10.203.37.170   10.203.37.164   TCP                 66 awg-proxy > iso-tsap [ACK] Seq=1 Ack=4619 Win=31529 Len=0 TSval=32112 TSecr=9378000
   154 0.883475000   10.203.37.164   10.203.37.170   COTP              1094 DT TPDU (0) [COTP fragment, 1021 bytes]
   155 0.883508000   10.203.37.164   10.203.37.170   PRES                82 DATA TRANSFER (DT) SPDU
   156 0.883710000   10.203.37.170   10.203.37.164   TCP                 66 awg-proxy > iso-tsap [ACK] Seq=1 Ack=5647 Win=32044 Len=0 TSval=32113 TSecr=9378000
   157 0.883861000   10.203.37.170   10.203.37.164   TCP                 66 awg-proxy > iso-tsap [ACK] Seq=1 Ack=5663 Win=32044 Len=0 TSval=32113 TSecr=9378000
   158 0.884173000   10.203.37.157   10.203.37.170   ENIP               562 Connection:  ID=0x00000341, SEQ=0000005044
   159 0.906734000   10.203.37.157   10.203.37.170   ENIP               562 Connection:  ID=0x00000304, SEQ=0000005106
   160 0.913564000   10.203.37.157   10.203.37.170   ENIP               562 Connection:  ID=0x00000186, SEQ=0000005303
   161 0.914636000   10.203.37.157   10.203.37.170   ENIP               562 Connection:  ID=0x000001C7, SEQ=0000005303
   162 0.940038000   10.203.37.157   10.203.37.170   ENIP               562 Connection:  ID=0x00000285, SEQ=0000005227
   163 0.943381000   10.203.37.157   10.203.37.170   ENIP               562 Connection:  ID=0x00000248, SEQ=0000005295
   164 0.947832000   10.203.37.170   10.203.37.157   ENIP               562 Connection:  ID=0x0037421E, SEQ=0000004110
   165 0.960951000   10.203.37.170   10.203.37.157   ENIP               562 Connection:  ID=0x0037421D, SEQ=0000004151
   166 0.970597000   10.203.37.157   10.203.37.170   ENIP               562 Connection:  ID=0x000002C3, SEQ=0000005169
   167 0.970783000   10.203.37.157   10.203.37.170   ENIP               562 Connection:  ID=0x00000382, SEQ=0000004984
⊟ Frame 23: 562 bytes on wire (4496 bits), 562 bytes captured (4496 bits) on interface 0
    Interface id: 0
    Encapsulation type: Ethernet (1)
    Arrival Time: Jun 20, 2013 11:46:30.117670000 Central Daylight Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1371746790.117670000 seconds
    [Time delta from previous captured frame: 0.013090000 seconds]
    [Time delta from previous displayed frame: 0.013090000 seconds]
    [Time since reference or first frame: 0.139667000 seconds]
    Frame Number: 23
    Frame Length: 562 bytes (4496 bits)
    Capture Length: 562 bytes (4496 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:udp:enip]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
```

Figure 7.19 Frame Capture



```
    [Coloring Rule String: udp]
⊟ Ethernet II, Src: Rockwell_c5:55:c4 (00:00:bc:c5:55:c4), Dst: ProsoftT_81:00:55 (00:0d:8d:81:00:55)
  ⊟ Destination: ProsoftT_81:00:55 (00:0d:8d:81:00:55)
      Address: ProsoftT_81:00:55 (00:0d:8d:81:00:55)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ⊟ Source: Rockwell_c5:55:c4 (00:00:bc:c5:55:c4)
      Address: Rockwell_c5:55:c4 (00:00:bc:c5:55:c4)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IP (0x0800)
⊟ Internet Protocol Version 4, Src: 10.203.37.157 (10.203.37.157), Dst: 10.203.37.170 (10.203.37.170)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    Total Length: 548
```

Figure 7.20 EtherNet/IP Communication

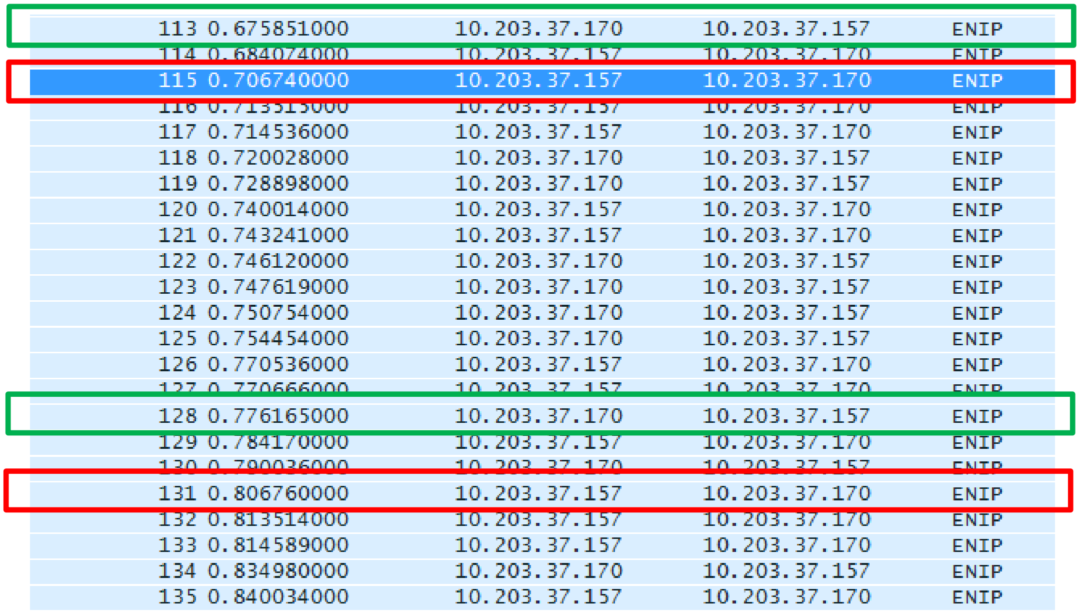


```
    [Coloring Rule String: tcp]
⊟ Ethernet II, Src: Schweitz_06:a0:31 (00:30:a7:06:a0:31), Dst: ProsoftT_81:00:55 (00:0d:8d:81:00:55)
  ⊟ Destination: ProsoftT_81:00:55 (00:0d:8d:81:00:55)
      Address: ProsoftT_81:00:55 (00:0d:8d:81:00:55)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ⊟ Source: Schweitz_06:a0:31 (00:30:a7:06:a0:31)
      Address: Schweitz_06:a0:31 (00:30:a7:06:a0:31)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IP (0x0800)
⊟ Internet Protocol Version 4, Src: 10.203.37.164 (10.203.37.164), Dst: 10.203.37.170 (10.203.37.170)
    Version: 4
    Header length: 20 bytes
```

Figure 7.21 Relay to Gateway Communications

In order to validate the system, packets were examined upon report MMS report header. Taking the timestamp between consecutive reports on the IEC side determines the degradation of gateway performance while loading. In addition, the time between EtherNet/IP messages to the controller, based upon connection streams, determines how the gateway module can be loaded on the controller side. Figure 7.22 shows an example of how analysis was performed. In this example, the RPI, or requested packet interval, from the gateway to the controller is defined as 100 milliseconds. This means that information is first read from the gateway to the controller, then written from the controller to the gateway. For this example, 10.203.37.57 is the gateway module and 10.203.37.170 is the automation controller. The green boxes represent writing from the PAC to the gateway, and the red boxes represent the reading from gateway to controller. In this example, the first read from the controller occurs at 0.706740000, and the next occurs at 0.806760000. This is a difference of 1.020 milliseconds, an acceptable rate based upon a 100-millisecond RPI time. The difference between the two green boxes for output comparison is 1.3

milliseconds, which is still acceptable. For the upcoming tests, the input stream from the gateway to the controller is analyzed.



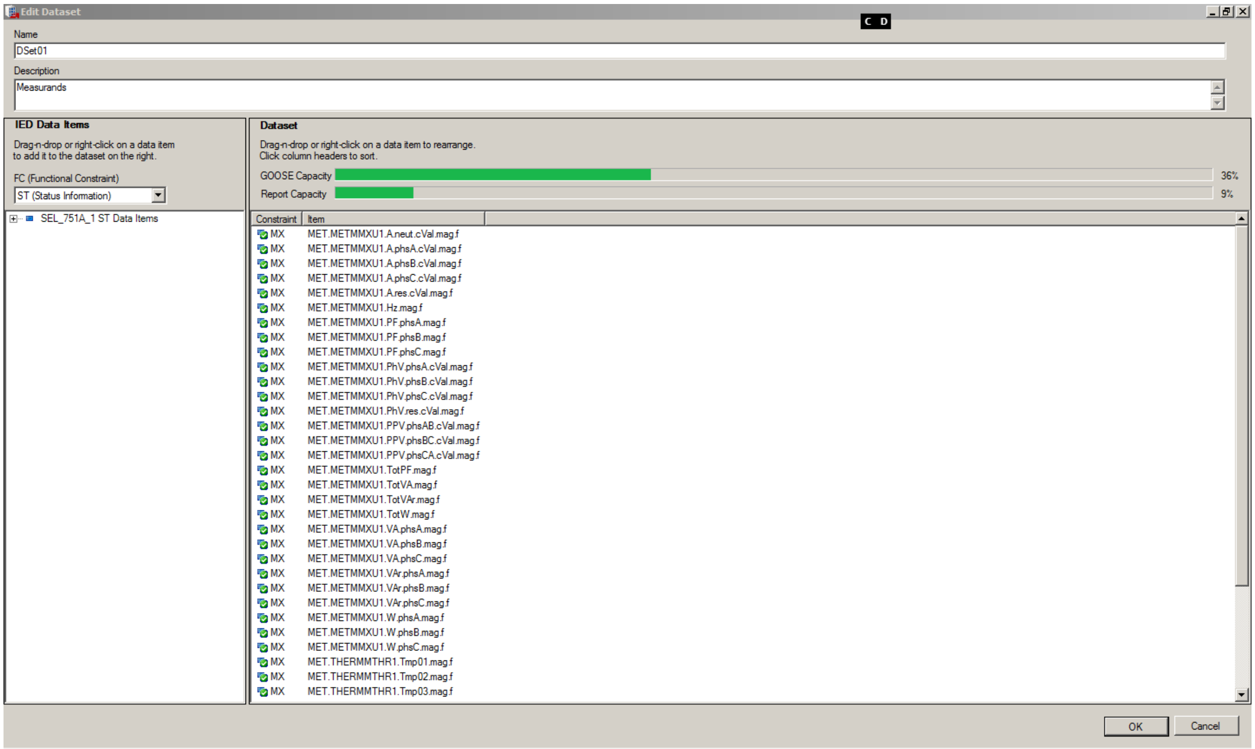
```
113 0.675851000   10.203.37.170   10.203.37.157   ENIP
114 0.684074000   10.203.37.157   10.203.37.170   ENIP
115 0.706740000   10.203.37.157   10.203.37.170   ENIP
116 0.713513000   10.203.37.157   10.203.37.170   ENIP
117 0.714536000   10.203.37.157   10.203.37.170   ENIP
118 0.720028000   10.203.37.170   10.203.37.157   ENIP
119 0.728898000   10.203.37.170   10.203.37.157   ENIP
120 0.740014000   10.203.37.157   10.203.37.170   ENIP
121 0.743241000   10.203.37.157   10.203.37.170   ENIP
122 0.746120000   10.203.37.170   10.203.37.157   ENIP
123 0.747619000   10.203.37.170   10.203.37.157   ENIP
124 0.750754000   10.203.37.170   10.203.37.157   ENIP
125 0.754454000   10.203.37.170   10.203.37.157   ENIP
126 0.770536000   10.203.37.157   10.203.37.170   ENIP
127 0.770666000   10.203.37.157   10.203.37.170   ENIP
128 0.776165000   10.203.37.170   10.203.37.157   ENIP
129 0.784170000   10.203.37.157   10.203.37.170   ENIP
130 0.790036000   10.203.37.170   10.203.37.157   ENIP
131 0.806760000   10.203.37.157   10.203.37.170   ENIP
132 0.813514000   10.203.37.157   10.203.37.170   ENIP
133 0.814589000   10.203.37.157   10.203.37.170   ENIP
134 0.834980000   10.203.37.170   10.203.37.157   ENIP
135 0.840034000   10.203.37.157   10.203.37.170   ENIP
```

Figure 7.22 Data Transfer Example

In order to validate the device support of the gateway module solution, the following test was conducted. Twenty identical feeder relays were gathered and assembled. A sample CID file, including one data set of 40 fundamental parameters, was then created and downloaded to each relay. One by one, a relay was added to the gateway module and controller.

The RPI, or requested packet interval, between the controller and the gateway module was then reduced in increments of five milliseconds until the module did not respond or the CIP connection was broken, i.e., there was no response by 2.5x the RPI time.

The dataset generated and downloaded to each relay can be seen in Figure 7.23. This dataset consists of both fundamental electrical parameters, such as voltages, currents, and powers, as well as thermal metering information from various RTD points. This dataset was selected because it was a good representation of parameters that are of interest for mining applications.



Name: DSet01

Description: Measurands

IED Data Items

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)

SEL_751A_1 ST Data Items

Dataset

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

GOOSE Capacity 36%
Report Capacity 9%

Constraint	Item
MX	MET.METMMXU1.A.neut.cVal.mag.f
MX	MET.METMMXU1.A.phsA.cVal.mag.f
MX	MET.METMMXU1.A.phsB.cVal.mag.f
MX	MET.METMMXU1.A.phsC.cVal.mag.f
MX	MET.METMMXU1.A.res.cVal.mag.f
MX	MET.METMMXU1.Hz.mag.f
MX	MET.METMMXU1.PF.phsA.mag.f
MX	MET.METMMXU1.PF.phsB.mag.f
MX	MET.METMMXU1.PF.phsC.mag.f
MX	MET.METMMXU1.PhV.phsA.cVal.mag.f
MX	MET.METMMXU1.PhV.phsB.cVal.mag.f
MX	MET.METMMXU1.PhV.phsC.cVal.mag.f
MX	MET.METMMXU1.PhV.res.cVal.mag.f
MX	MET.METMMXU1.PPV.phsAB.cVal.mag.f
MX	MET.METMMXU1.PPV.phsBC.cVal.mag.f
MX	MET.METMMXU1.PPV.phsCA.cVal.mag.f
MX	MET.METMMXU1.TotPF.mag.f
MX	MET.METMMXU1.TotVA.mag.f
MX	MET.METMMXU1.TotVAr.mag.f
MX	MET.METMMXU1.TotW.mag.f
MX	MET.METMMXU1.VA.phsA.mag.f
MX	MET.METMMXU1.VA.phsB.mag.f
MX	MET.METMMXU1.VA.phsC.mag.f
MX	MET.METMMXU1.VAr.phsA.mag.f
MX	MET.METMMXU1.VAr.phsB.mag.f
MX	MET.METMMXU1.VAr.phsC.mag.f
MX	MET.METMMXU1.W.phsA.mag.f
MX	MET.METMMXU1.W.phsB.mag.f
MX	MET.METMMXU1.W.phsC.mag.f
MX	MET.THERMMTHR1.Tmp01.mag.f
MX	MET.THERMMTHR1.Tmp02.mag.f
MX	MET.THERMMTHR1.Tmp03.mag.f

Figure 7.23 Test Dataset

A system was constructed, as shown in Figure 7.17, with a network of IEDs, managed switch, PAC, and computer running the wire shark packet analyzer. To begin the testing procedure, one feeder relay was added to the system. The RPI time between the controller and gateway module was set to a value that resulted in a stable connection, in this case 40 milliseconds, as shown in Figure 7.24. Once the RPI change was made, the first three wire shark time differences were taken between report publication to the controller and averaged, and the resulting value was displayed in the chart as the actual RPI time. The RPI time from the controller was then dropped by five milliseconds, and the procedure was repeated. The procedure was repeated until the actual RPI time exceeded 2.5 times the requested RPI time, which is the point defined by ODVA when the connection between the controller and remote I/O device, i.e., the gateway, is to be terminated. The red line in Figure 7.24 shows this relationship graphically. When the blue line intersects the red line in the plot, the connection timed out and became faulted. At this point, the connection was terminated and the test trial was concluded. An additional IED was then be added to the system and the process would be repeated. It can be seen that the system failed at the 15 ms RPI time.

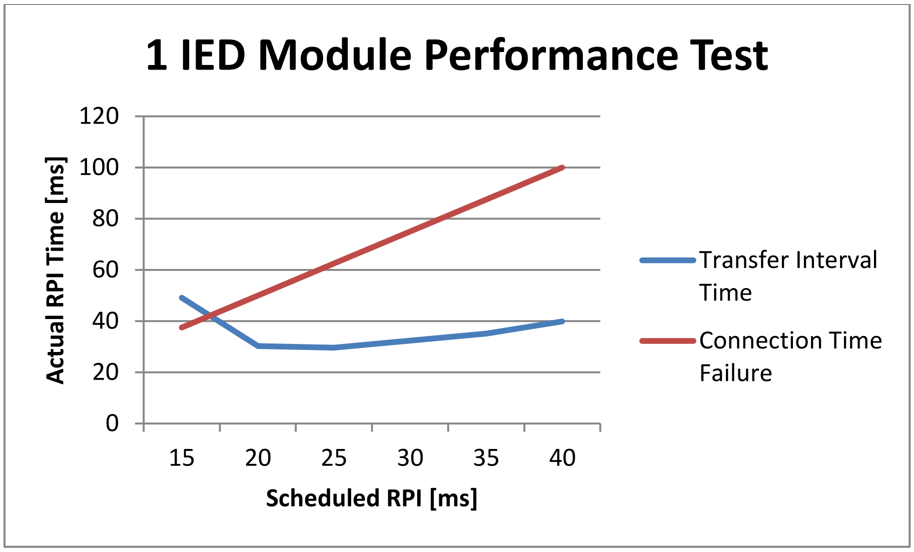


Figure 7.24 1 IED Test Results

Figure 8.25 depicts test results when a second IED was added to the system. It can again be seen that the connection between controller and gateway times out when the scheduled RPI time is set to 15 milliseconds. As the number of IEDs is increased to 10, as shown in Figure 8.26, it can be seen that the RPI time between the controller to the gateway must be increased to 50 milliseconds to allow the gateway sufficient processing time to manage both the IEC 61850 network and package data on the EtherNet/IP network for the control system.

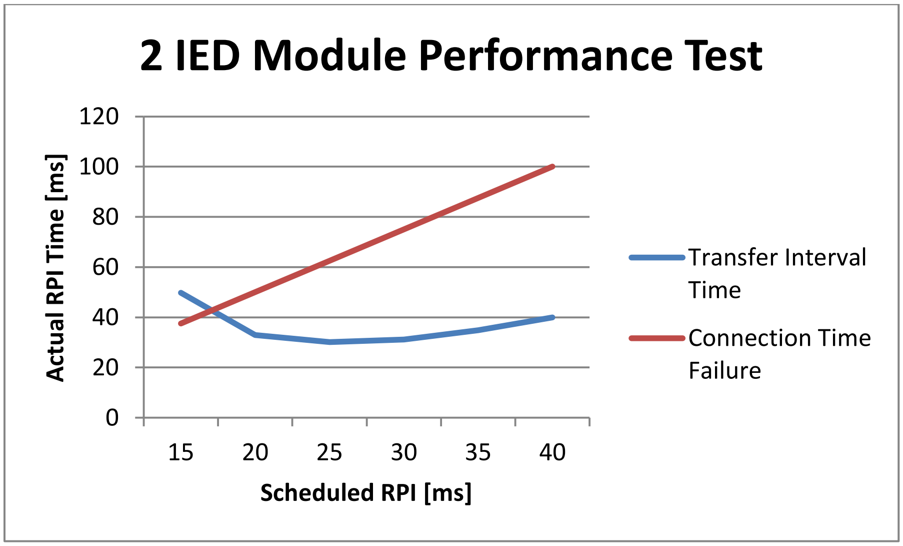


Figure 7.25 2 IED Test Result

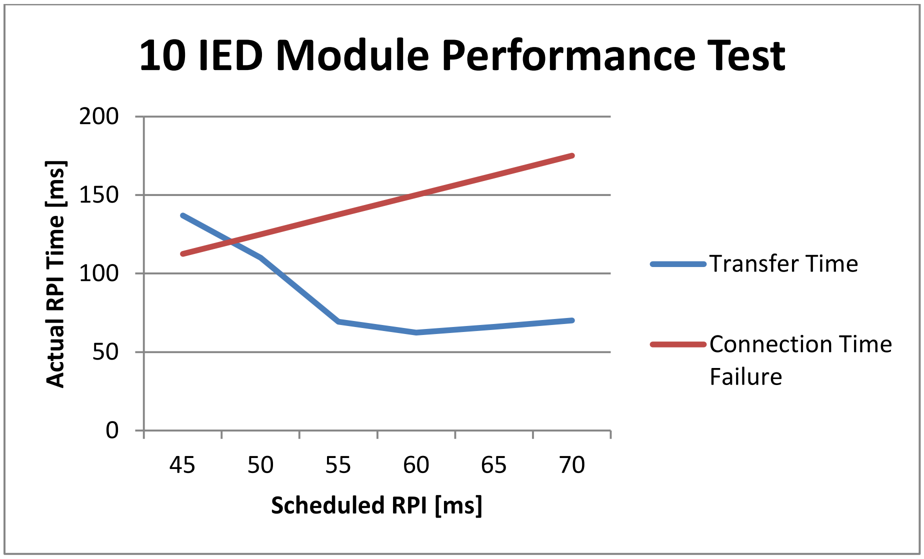


Figure 7.26 10 IED Test Results

Figure 8.27 shows test results for all 20 IEDs (in this case, feeder relays) for the system. The curve can also be described as the capability curve of the system since it defines the maximum operating points for the gateway module at each discrete transition in the system, i.e., the addition of an IED. It can be seen that the system can accommodate up to 20 IEDs as specified by the conceptual design. Moving 40 parameters from 20 independent relays can be achieved at an RPI rate of 100 milliseconds, leaving 300 milliseconds, or 75% of the time, for other calculations.

The maximum number of IEDs supported by this module is 20 as that is the maximum number of IEC 61850 driver threads that can run at once. The maximum scenario was tested with a valid set of parameters to transfer to the control system and performance was set at 100 milliseconds for the data to transfer from gateway to controller for a stable operating point. It was determined that the RPI time of 100 milliseconds was acceptable based upon current load shedding algorithms. As a result, the device support portion of the conceptual design was met by this testing.

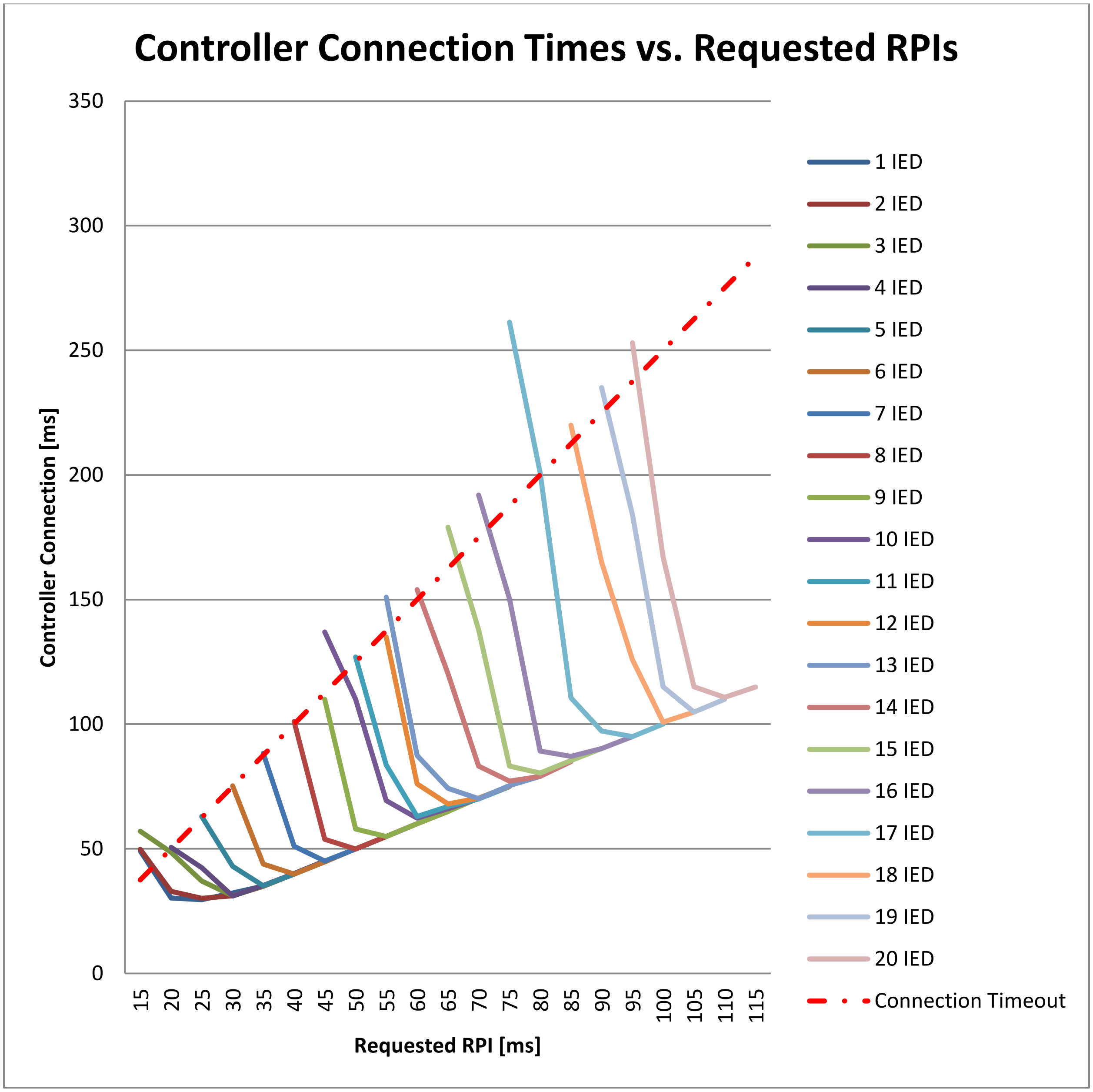


Figure 7.27 Capability Curve of System

Conclusions

This chapter discussed the testing and verification procedures conducted throughout this research to validate and verify functionality of the proposed solution. As the research goal was to develop a mine monitoring solution utilizing the IEC 61850 standard to link the electrical distribution system with the process control system, it was determined that the best way to validate the work and functionality of the proposed system was to benchmark it against the conceptual design. The chapter discussed each conceptual design milestone as well as the procedure used to validate that this goal was met throughout the period of conducted research. For milestones requiring more than just visual inspection, the experimental setup was defined, testing procedure described, and experimental results discussed. It was determined that the proposed solution meets all of the required specifications imposed by the conceptual design.

Chapter 8 - Practical Implementation

Introduction

This chapter describes the practical implementation of the mine monitoring system described in this dissertation. The practical implementation of the monitoring system was installed at a bauxite mine to monitor the overall energy consumption and provide critical load shedding functionality for the process. This chapter describes the overall system and how it is utilized in real time production of the mine.

System Description

The system contains multiple sites of a mining operation that are separated over multiple kilometers (km) of distance. In total there are approximately 350 IEDs distributed throughout the system. This system is fed from two 22-kV feeders that monitor an intertie in the system, as shown in Figure 8.1. Two important locations to be noted in Figure 8.1 are the 11-kV Generator Buses A and B, located on the left of the diagram. An electrical intertie is located between them that needs constant monitoring for load shedding possibilities.

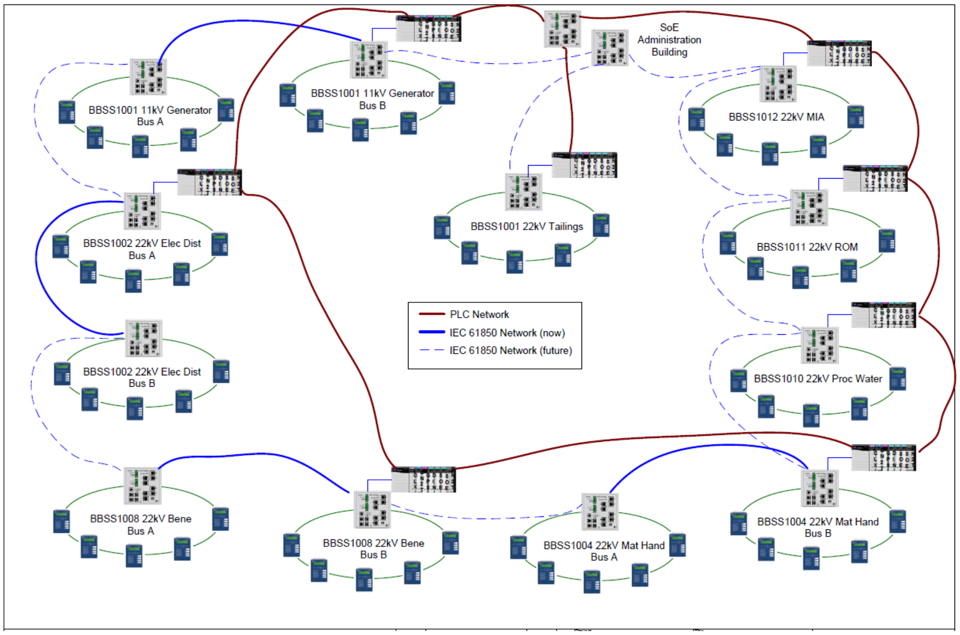

SoE Administration Building
BBSS1001 11kV Generator Bus A
BBSS1001 11kV Generator Bus B
BBSS1012 22kV MIA
BBSS1001 22kV Tailings
BBSS1002 22kV Elec Dist Bus A
BBSS1011 22kV ROM
PLC Network
IEC 61850 Network (now)
IEC 61850 Network (future)
BBSS1002 22kV Elec Dist Bus B
BBSS1010 22kV Proc Water
BBSS1008 22kV Bene Bus A
BBSS1008 22kV Bene Bus B
BBSS1004 22kV Mat Hand Bus A
BBSS1004 22kV Mat Hand Bus B

Figure 8.1 Bauxite Mine Distribution System

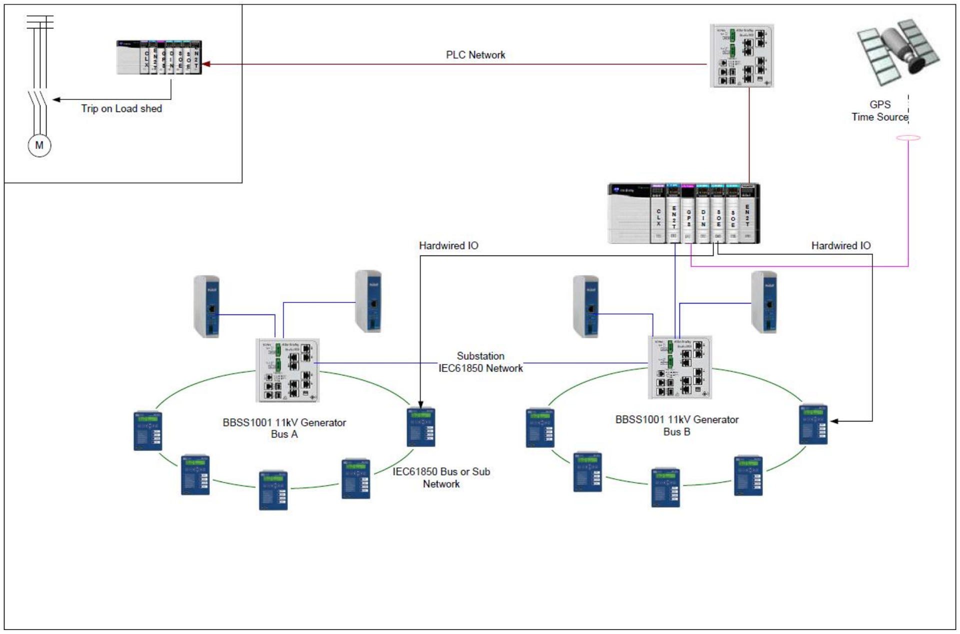


Figure 8.2 Intertie Monitoring

The monitoring of the intertie between two generator buses is crucial to detect electrical islanding and other unwanted power issues. Typical algorithms for islanding detection use values such as rates of change of frequency, voltage, and current. The client uses the capabilities of the IEC 61850 implementation to gather information, run calculation logic, and execute the proper load shedding decision in an acceptable, long-time-constant scenario. A diagram of the load shedding system can be seen in Figure 8.2.

With the ability to connect the electric distribution system to the process control system, load shedding at the automation controller is realized. Measurands from various IEDs can be configured to be sent on various time intervals to the controllers. Controllers can run simple algorithms and make correct load shedding decisions, either over hard wired outputs or by utilizing the remote bit’s strategy for command and control over IEC 61850. Many load shedding algorithms for magnitude protection are long-time constant algorithms. This means that the time to propagate information from IED, through the system, to controller is on the order of hundreds of milliseconds.

An example of data path propagation can be seen in Figure 8.3. By grouping parameters of interest in a single report, the RPI of this Class 1 connection can be reduced to small RPI times, from 10-20 milliseconds, in order to populate the controller data table.

The load shedding algorithm monitors frequency, voltage, and current magnitudes to detect abnormal generator or motor behavior. These values do not themselves give good indications of system performance, but the rates of change of these fundamental parameters are used to detect abnormal and unwanted power system conditions. Since IEC 61850 provides the timestamp with the associated measurand, a rolling window of data points (i.e. 5 frequencies and times) is collected, and the first derivative is calculated and averaged over the window. Based on this average calculated value, the automation controller can make a decision on whether to shed load or take remedial action in order to provide a more stable operating point.

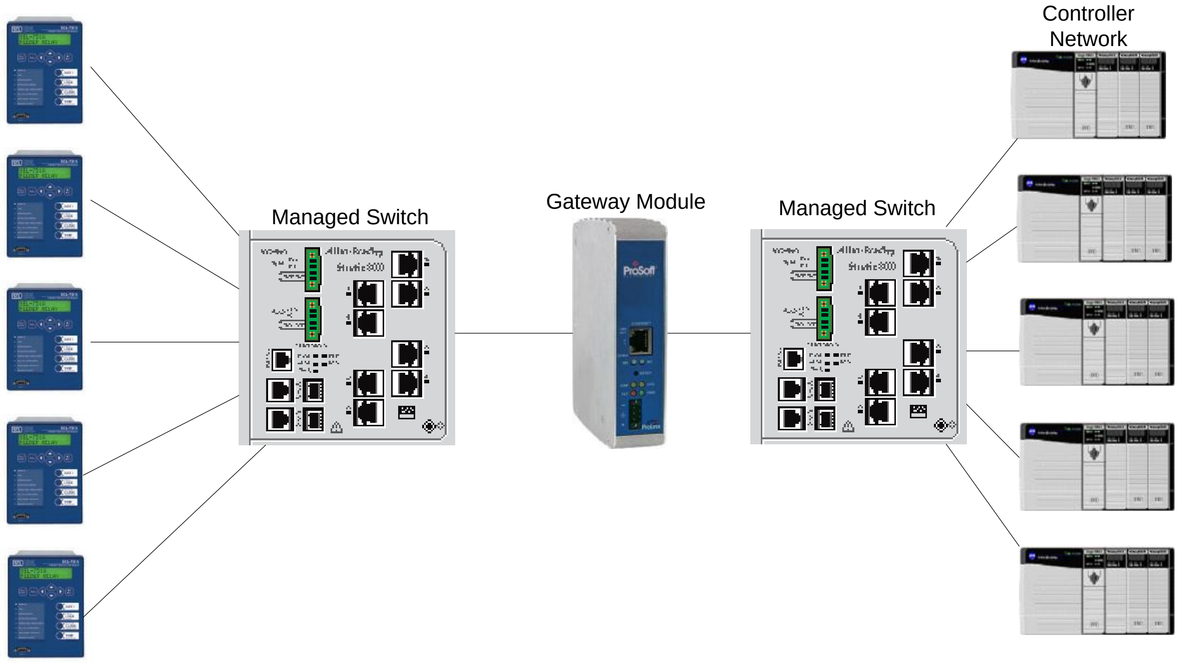


Figure 8.3 Load Shedding Data Flow

Conclusions

In addition to monitoring various fundamental electrical measurands in a SCADA application, the technology developed during this research was implemented in a bauxite mine in order to provide critical load shedding. The system was interfaced to 350 IEDs distributed over various sites that were separated by 25-30 kilometers. The load shedding algorithm collected data from the IEC 61850 network and used the concept of rolling windows to calculate fundamental changes in frequency, voltage, and current, in order to assess system health.

Chapter 9 - Conclusions and Future Work

Overall Conclusions

The IEC 61850 standard suite of protocols and methods provides a convenient method to integrate power system IEDs into control system networks in order to provide greater electrical system visibility for plant operators and engineers. Connecting these devices to the Supervisory Control and Data Acquisition system through the gateway, as described in this dissertation, can greatly enhance system command and control functionality.

This dissertation presented various SCADA standards and defined their benefits and shortcomings in both generic comparisons and benefits to mining protection and automation systems. The literature review of this document has shown that, with respect to interoperability between vendors and a common naming convention, the IEC 61850 standard was the best candidate to implement an automation and control solution that could interact with current systems. It was also discussed that the IEC 61850 standard possessed the following shortcomings that would need to be addressed by this solution: although the substation configuration language (SCL) file defines how an IED communicates on a 61850 network, it does not define configuration information for protection and control functions of each IED.

Each IED manufacturer has proprietary software and configuration tools used for enabling and configuring various protection elements and control strategies. The IEC 61850 standard defines no methodology for designing communication-based assisted automation. In order to develop a successful solution for the mining market, these two concerns needed to be addressed within the conceptual design of the total solution, i.e., hardware, software, and visualization.

The dissertation summarized the hardware gateway module developed to implement the conceptual design. A key component to this design is a multi-threaded design to interface between the IED IEC 61850 network and the process EtherNet/IP network. Each thread, including the master control program, was defined and its basic functionality explained in functional block diagram. The various types of data used in this project were then addressed, followed by how data is packaged in packets to be sent to the process network via EtherNet/IP.

The concept of the tag database was then discussed, i.e., how it is utilized as a common space of shared memory where both 61850 and EtherNet/IP drivers read and write various tags. Also discussed was the use of semaphore tags in the database to avoid collisions in the tag database.

This research presented software developments that provide a functional, user-friendly solution to aid engineers in link the electric power system of a mine with its process control system. The software developed to support this research had two components: software created to interface with the gateway module and software created to move information to a process historian. Software developed to interface with the gateway module reads CID files and allows the user to map information that is required by the automation controller. After this information is defined, these tags are mapped to corresponding EtherNet/IP tags according to the ODVA standard. Once this mapping has occurred, the module configuration is downloaded to the module and an Add-on instruction is generated so that the data stream can be interpreted by the controller. Once the Add-on instruction is imported by the controller, information is correctly parsed from the data stream and inserted in to the controller data table.

The historian software reads information from the Alarms and Events software database and inserts the corresponding timestamp in post processing to the process historian within 15 microseconds. Once the data is in the automation controller from hardware and software configurations, the remainder of this solution provides operators and engineers visual aids to help enhance command and control of a mine process control system.

Visualization tools were also developed in this research. Items include faceplate definition, human/machine interface discussion and definition, and data management solutions. The goal of visualization was to represent an IED in logic and graphics to produce a virtual faceplate that is familiar to users accustomed with a particular IED. This gives operators and engineers the same look and feel experience they have had with the physical device. At the same time, graphics were developed with various levels of security to allow only users with proper credentials access to various command and control functions.

Additionally, the research solution adhered to various graphics standards for both power and process control systems. The visualization trending functionality of the solution was then discussed by providing an example of a trend from a process historian repository. Finally, an example of a secondary selective main-tie-main scheme was presented for a processing facility. The one-line diagram was converted to an HMI screen, and multiple faceplates were tied to various instrumentation blocks in the diagram to enable remote monitoring and control of the system.

Finally, testing and verification procedures conducted throughout this research were presented and described. As this research's goal was to develop a mine monitoring solution utilizing the IEC 61850 standard to link the electrical distribution system with the process control system, it was determined that the best way to validate the work and functionality of the proposed system was to benchmark performance test results against the conceptual design. For milestones requiring more than just visual inspection, the experimental setup was defined, testing procedure discussed, and experimental results discussed. It was determined that the proposed solution meets all of the required specifications imposed by the conceptual design.

The proposed system developed in this research was then installed at a bauxite mine in order to provide energy data to the process control system as well as provide data for critical load shedding decisions.

Additionally, this dissertation presented how the structured IEC 61850 tag names were maintained in the automation controller software, which provides greater transparency of the data values and faster commissioning of a system. The protocols allow for the interconnection of all IEDs and control systems at the industrial level, including mining, metals, pulp and paper, semiconductor, oil and gas, and more. The faceplate solution described within this document provides many advantages to process owners.

This solution is an all-encompassing plant-wide solution for not only the electrical distribution system, but also for functions on the electrical process network. The solution provides process owners with the ability to extend control down to the individual breaker or contactor for functions such as load monitoring and load shedding. By incorporating a solution that functions with a multitude of vendors, process owners can now easily manage their electrical systems with a simple, standard network infrastructure.

Future Work Recommendations

The IEC 61850 standard provides a uniform method of communication among multivendor intelligent electronic devices used for system monitoring, control, and protection. Connecting these devices to the Supervisory Control and Data Acquisition system through the gateway, as described in this dissertation, can greatly enhance system command and control functionality. Benefits to the mining industry include improved system monitoring, improved safety through remote control of power system components, reduction in downtime through the reconstruction of the sequence of events leading to system failures, and the implementation of demand management. Therefore, there are opportunities for future work in the development of algorithms for achieving specific applications for monitoring and control of power systems. These include

1. Power monitoring
2. Predictive maintenance
3. Demand side load management
4. Auto Transfer Switch (ATS) load shedding
5. Command and control of rotating machinery
6. Command and control of breakers
7. Interfacing energy measurements into speed regulator process models
8. Ventilation On Demand (VOD)
9. Fast motor bus transfer
10. Main-Tie-Main Distribution Schemes
11. Data Historian Applications
12. On Demand Topology Transitions

References

[1] U. S. D. o. Energy, "Mining Industry Bandwidth Study," 2007.

[2] C. Ministry of Energy, "Industrial Energy Use in Canada Emerging Energy Trends," 2010.

[3] S. o. Minnesota, "Energy Consumption Minnesota," 2005.

[4] AIST, "The Making Shaping and Treatment of Steel: Iron Making Volume," ed, 2012.

[5] S. H. Horowitz and A. G. Phadke, *Power System Relaying*: John Wiley & Sons, 2008.

[6] D. L. Ransom and C. Chelmecki, "Using GOOSE messages in a main-tie-main scheme," in *Industry Applications Society Annual Meeting (IAS), 2012 IEEE*, 2012, pp. 1-8.

[7] D. Bailey and E. Wright, *Practical SCADA for Industry*: Elsevier Science, 2003.

[8] G. Clarke and D. Reynders, *Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems*: Elsevier Science, 2004.

[9] I. PES, "IEEE Standard for SCADA and Automation Systems," vol. IEEE Std C37.1-2007 (Revision of IEEE Std C37.1-1994), ed, 8 May 2008.

[10] R. A. Inc., "Control Logix System," R. A. Inc., Ed., ed. Milwaukee, WI: Rockwell Automation Inc., 2011, p. 44.

[11] R. A. Inc. (2011). *1756-UM001L-EN-P-May-2011*.

[12] M. Organization, "Modbus Messagin on TCP/IP Implementation Guide," ed. www.modbus.org, 2000.

[13] J. Momoh, *Smart Grid: Fundamentals of Design and Analysis*: John Wiley & Sons, 2012.

[14] A. A. Sallam and O. P. Malik, *Electric Distribution Systems*: John Wiley & Sons, 2011.

[15] I. E. Commission and C. E. Internationale, *International Standard IEC 60870-5-101: Telecontrol Equipment and Systems. Part 5-101 : Transmission Protocols, Companion standard for basic telecontrol tasks : Amendment 2*: IEC, 2001.

[16] J. M. P.E. (2010). *Substation Automation Basics*. Available: www.electricenergy.com

[17] D. U. Group, "DNP3 Specification Version 2.0," in *DNP3 Introduction*, ed. DNP Users Group: DNP Users Group, 2002.

[18] I. I. Resources, "Power Industry Electrical Generation," in *Electric Power Generation Coverage*, ed. http://www.industrialinfo.com/marketcoverage.jsp?pagerequest=marketcoverage01_intl: Industrial Info Resources Global Power Database, 2012.

[19] IEC, "Precision clock synchronization protocol for networked measurement and control systems," vol. 61588, ed. www.iec.ch: IEC/IEEE, 2009, p. 292.

[20] IEC, "Communication networks and systems in substations," in *Part 1: Introduction and Overview* vol. 61850-1, ed. www.iec.ch: IEC, 2003, p. 40.

[21] IEC, "Communication networks and systems in substations," in *Part 3: General Requirements* vol. 61850-3, ed. www.iec.ch: IEC, 2002, p. 36.

[22] IEC, "Communication networks and systems for power utility automation," in *Part 4: System and project management* vol. 61850-4, ed. www.iec.ch: IEC, 2011, p. 78.

[23] IEC, "Communication networks and systems for power utility automation," in *Part 6: Configuration description language for communication in electrical substations related to IEDs* vol. 61850-6, ed. www.iec.ch: IEC, 2009, p. 220.

[24] IEC, "Communication networks and systems for power utility automation," in *Part 7-2: Basic information and communication structure - Abstract communication service interface (ACSI)* vol. 81850-7-2, ed. www.iec.ch: IEC, 2010, p. 218.

[25] IEC, "Communication networks and systems for power utility automation-," in *Part 7-410: Hydroelectric power plants - Communication for monitoring and control* vol. 61850-7-410, ed. www.iec.ch: IEC, 2007, p. 90.

[26] IEC, "Communication networks and systems for power utility automation," in *Part 7-510: Basic communication structure - Hydroelectric power plants - Modelling concepts and guidelines* vol. 61850-7-510, ed. www.iec.ch: IEC, 2012, p. 86.

[27] IEC, "Communication networks and systems in substations," in *Part 10: Conformance testing* vol. 61850-10, ed. www.iec.ch: IEC, 2005, p. 46.

[28] IEC, "Communication networks and systems in substations," in *Part 2: Glossary* vol. 61850-2, ed. www.iec.ch: IEC, 2003, p. 46.

[29] IEC, "Communication networks and systems in substations," in *Part 5: Communication requirements for function and device models* vol. 61850-5, ed. www.iec.ch: IEC, 2003, p. 134.

[30] IEC, "Communication networks and systems for power utility automation," in *Part 7-1: Basic Communication Structure - Principles and models* vol. 61850-7-1, ed. www.iec.ch: IEC, 2011, p. 294.

[31] IEC, "Communication networks and systems for power utility automation," in *Part 7-3: Basic communication structure - Common data classes* vol. 61850-7-3, ed. www.iec.ch: IEC, 2010, p. 186.

[32] IEC, "Communication networks and systems for power utility automation," in *Part 7-4: Basic communication strucuture - Compatible logical nodes classes and data object classes* vol. 61850-7-4, ed. www.iec.ch: IEC, 2010, p. 184.

[33] IEC, "Communication networks and systems for power utility automation," in *Part 7-420: Basic communication structure - Distributed energy resources logical nodes* vol. 61850-7-420, ed. www.iec.ch: IEC, 2009, p. 104.

[34] IEC, "Communication networks and systems for power utility automation," in *Part 8-1: Specific communication service mapping (SCSM) - Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3* vol. 61850-8-1, ed. www.iec.ch: IEC, 2011, p. 390.

[35] IEC, "Communication networks and systems for power utility automation," in *Part 9-2: Specific communication service mapping (SCSM) - Sampled values over ISO/IEC 8802-3* vol. 61850-9-2, ed. www.iec.ch: IEC, 2011, p. 70.

[36] D. S. K. Shukla, "IEC 61850 Overview," ed. Virginia Tech, 2011.

[37] B. S. I. Staff, *Communication Networks and Systems in Substations. Specific Communication Service Mapping (SCSM). Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3*: B S I Standards, 2004.

[38] Mackiewicz, "Technical Overview and Benefits of the IEC 61850 Standard for Substation Automation," www.sisconet.com2010.

[39] Dolezilek, "IEC 61850: What You Need to Know About Functionality and Practical Implementation," Schweitzer Engineering Labs, www.selinc.com2011.

[40] R. A. Inc. (2010). *Integrated Architecture and CIP Sync Configureation*.

[41] R. A. Inc., "Precision Time Protocol Over EtherNet/IP," in *Motion Over EtherNet/IP Learning Series*, R. A. Inc., Ed., ed. Milwaukee, WI: Rockwell Automation Inc., 2011.

[42] R. C. Council, "IRIG Serial Time Codes Formats," ed: Range Commanders Council Telecommunications and Timing Group, 2004.

[43] B. W. Pike, "IRIG, Inter-Range Instrumentation Group----History, Functions and Status, 1959," *Space Electronics and Telemetry, IRE Transactions on,* vol. SET-6, pp. 59-61, 1960.

[44] B. Dickerson, "Time in the Power Industry: How and Why We Use It," *Arbiter Systems, Inc.*

[45] I. P. PSRC, "IRIG-B Time Code Connection Requirements."

[46] D. L. Mills, "Internet time synchronization: the network time protocol," *Communications, IEEE Transactions on,* vol. 39, pp. 1482-1493, 1991.

[47] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002),* pp. c1-269, 2008.

[48] Cisco, "CIP Sync Sequence of Events," in *Converged Plantwide Ethernet (CPwE) Design and Implementation Guide*, Cisco, Ed., ed, 2011, p. 72.

[49] S. Hekmat. (January 2011, Communication Networks. 24-39.

[50] L. M. Surhone, M. T. Tennoe, and S. F. Henssonow, *Common Industrial Protocol*: VDM Verlag Dr. Mueller AG & Co. Kg, 2010.

[51] ODVA, "The Common Industrial Protocol (CIP) and the Family of CIP Networks," ODVA, Ed., ed, 2010, p. 92.

[52] D. C. Mazur, "Synchronized Rotor Angle Measurements of Synchronous Machines," Master of Science, Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, 2012.

[53] D. C. M. Jaime De La Ree, "Synchronized Rotor Angle Measurements of Synchronous Machines," presented at the IEEE Industry Applications Society Annual Meeting, Las Vegas, NV, 2012.

[54] D. C. M. Ryan D. Quint, Noah G. Badayos, "A Protective Relay Voting Scheme," presented at the IEEE Industry Application Society Annual Meeting, Las Vegas, NV, 2012.

[55] R. D. Q. David Christopher Mazur, Virgilio A Centeno, "Time Synchronization of Automation Controllers for Power Applications," presented at the IEEE Industry Applications Society Annual Meeting, Las Vegas, Nevada, 2012.

[56] S. C. Greg Pfister, David Roop, "A Syncrhonized Sequence of Events Recorder for Power Applications," presented at the IEEE Industry Application Society Annual Meeting, Las Vegas, NV, 2012.

Appendix I - Glossary of Terms

AOI- Add-On Instruction; A user defined custom piece of code that is encapsulated within a single logic instruction within the controller environment

ALMD- Alarm Digital; An alarm instruction inside RS Logix 5000 used to indicate there is an alarm within the system

Architect- IEC 61850 configuration software developed by Schweitzer Engineering Labs

ATS-Auto-transfer Switch; A mechanism used in load shedding and topology management for a “bumpless” transition of connecting or disconnecting devices from the plant

CID- Configured IED Description; Configuration file used to define what information will be served from IED to the SCADA system

Class 1 Connection- High speed communications usually within a publisher-subscriber mechanism that is set at an RPI rate; used between modules of a PAC or remote I/O and the PAC processor

Class 3 Connection- Message instruction, a request/response type message where individual pieces of information are gathered on a master/slave type response

ControlLogix- A brand of PAC manufactured by Rockwell Automation

Current Transformer (CT)- Analog Sensor to measure presence of current in power system

DCS- Distributed Control System

EMI- Electromagnetic Interference

EPC- Engineering Procurement Construction

EtherNet/IP – Protocol owned by ODVA that specifies transfer of data for manufacturing control applications over traditional 802.3 Ethernet.

EWS- Engineering Work Station; Graphical interface or terminal where information is provided to engineers for calculations or operation

GOOSE- Generic Object Oriented Substation Event; Protocol defined by IEC 61850 for peer-to-peer or one-to-many fast exchange of data

Global Object- images that will be repeated reused throughout an application; given global scope so they can be referenced from any visualization screen

GUI-Graphical User Interface
I&C- Instrumentation and Control; usually applied to an engineering sector that defines, specifies, and codes for process instrumentation inside and industrial process

ICD- IED Capability Description; A file containing all the information that can be published by the IED

IEC – International Electrotechnic Commission; Governing body of standards based in Europe

IEC 61850- Standard developed by IEC for interfacing to Intelligent Electronic Devices

IED- Intelligent Electronic Device; e.g. relay, circuit breaker, meter

IEEE C37.1- A standard developed by IEEE that defines the performance of SCADA systems

Interface Node- A piece of software or hardware that qualifies data from the process control system to be passed to the historian repository

ISA- Instrumentation Society of America

ISA 5.5- A standard developed by ISA which specifies schemes for graphics on process instrumentation

IT- Information Technology

MAC- Media Access Control; Firmware address of a device defined for identification purposes

Main-Tie-Main- An electrical scheme where loads can be added or shed from the system in the case of a fault so that loads can be prioritized, and the plant not completely shut down

MCP- Master Control Program; Program run on startup of the gateway module

MMS- Manufacturing Message Specification; Protocol defined by IEC 61850 for block transfer of data from publisher to subscriber

ODVA- Open Device Vendor's Association; Organization that owns industrial protocols such as EtherNet/IP

OSI-PI- A manufacturer of a popular process historian

OWS- Operator Work Station; Interface, usually graphical that will allow status feedback and command and control of a process

PAC- Programmable Automation Controller

Persist- A database term used to “bind” data to a tag and populate it through the entire process control system
PHY- Physical Interface; e.g. RJ45 Ethernet adapter

PLC- Programmable Logic Controller

PT-Potential Transformer; Analog sensor used to measure a presence of voltage on an electrical system

PTP- Precision Time Protocol; Standard defined by IEEE 1588 v.2 which defines time synchronization of devices over a standard 802.3 Ethernet media.

Rockwell Automation- A company dedicated to manufacturing PACs and other industrial automation equipment

RPI-Requested Packet Interval; The rate at which information is shared across an EtherNet/IP network, also used for timeout of certain devices

RS Logix 5000- PAC programming software developed by Rockwell Automation

SCADA- Supervisory Control and Data Acquisition; A scheme to gather data from distributed sections of facilities

SCD- Substation Configuration Description; A file that defines all the IEDs on a network

SCL- Substation Configuration Language; XML scripted language defined by IEC 61850

SD- Secure Digital; A method for storing information on an external format

SEL- Schweitzer Engineering Labs; An IED manufacturer

SINT- Short Integer; 8 bits or one byte of data

SISCO- Systems Integration Specialists Company; A manufacturer of the IEC 61850 stack

Tag Database- Shared piece of memory where data items can be located using a tag name. The shared space where IEC 61850 tags and EtherNet/IP tags are shared

Target- A specific relay assertion element of a protective device, e.g. overcurrent

UDT- User Defined Data type; Object Data type that is customizable by user to make data objects

XML- Extensible Markup Language; A schema used in IEC 61850 to define file data types, and markup for defining data objects within the standards

Zero Sequence Currents- a part of a symmetrical currents calculation that leads to an indication of ground current presence within a system
50/51- Relay used to detect instantaneous and time delay overcurrent conditions

50G- Relay used to detect ground fault current