Enabling Interoperable SCADA Communications for PV Inverters through Embedded Controllers

Deepthi Vaidhynathan,[1] Kumaraguru Prabakar,[1] Akanksha Singh,[1] Emma Raszmann,[1] Joel Greene,[2] Christoph Brunner,[2] and Beth Capeles[2]

1 National Renewable Energy Laboratory
2 Triangle Microworks, Inc.

NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Technical Report
NREL/TP-5D00-79386
June 2021

Enabling Interoperable SCADA Communications for PV Inverters through Embedded Controllers

Deepthi Vaidhynathan,[1] Kumaraguru Prabakar,[1] Akanksha Singh,[1] Emma Raszmann,[1] Joel Greene,[2] Christoph Brunner,[2] and Beth Capeles[2]

1 National Renewable Energy Laboratory
2 Triangle Microworks, Inc.

Suggested Citation
Vaidhynathan, Deepthi, Kumaraguru Prabakar, Akanksha Singh, Emma Raszmann, Joel Greene, Christoph Brunner, and Beth Capeles. 2021. *Enabling Interoperable SCADA Communications for PV Inverters through Embedded Controllers*. Golden, CO: National Renewable Energy Laboratory. NREL/TP-5D00-79386.
https://www.nrel.gov/docs/fy21osti/79386.pdf.

NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Technical Report
NREL/TP-5D00-79386
June 2021

National Renewable Energy Laboratory
15013 Denver West Parkway
Golden, CO 80401
303-275-3000 • www.nrel.gov

NOTICE

This work was authored in part by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by U.S. Department of Energy Office of Energy Efficiency and Renewable Energy Solar Energy Technologies Office. The views expressed herein do not necessarily represent the views of the DOE or the U.S. Government.

Cover Photos by Dennis Schroeder: (clockwise, left to right) NREL 51934, NREL 45897, NREL 42160, NREL 45891, NREL 48097, NREL 46526.

List of Acronyms

AMPVI	Additively Manufactured Photovoltaic Inverter
DNP3	Distributed Network Protocol 3
DSO	distribution system operator
DTM	Device Type Manager
ESIF	Energy Systems Integration Facility
GOOSE	Generic Object-Oriented Substation Event
ICD	IED Capability Description
IEC	International Electrotechnical Commission
IED	intelligent electronic device
IP	Internet Protocol
JSON	JavaScript Object Notation
MMS	Manufacturing Message Specification
NREL	National Renewable Energy Laboratory
PV	photovoltaic
SAV	Sampled Value
sbRIO	single-board Reconfigurable Input-Output
SCADA	supervisory control and data acquisition
SCL	Substation Configuration Description Language
TCP	Transmission Control Protocol
TMW	Triangle MicroWorks
VAR	volt ampere reactive
XML	Extensible Markup Language
ZeroMQ	Zero Message Queueing

Executive Summary

The percentage integration of photovoltaic (PV) inverters in the field has increased significantly in the past 5 years. Regardless of the size of the PV plants and the inverters (residential vs. commercial), it is becoming crucial that these devices have the capability to communicate with peers (other smart devices) and with components that are at a hierarchy above the inverters (e.g., supervisory control and data acquisition (SCADA) systems, distributed controllers, and data managers). This project aims to develop a standard SCADA software code for inverters' embedded controllers that will enable interoperability with other components in the system. To achieve this, the code will be developed using two different protocols: Distributed Network Protocol 3 and International Electrotechnical Commission 61850. The developed code is aimed to be deployed in simple embedded controllers. It will be tested in the National Renewable Energy Laboratory's (NREL's) Energy Systems Integration Facility. The tested code will then be made available through Triangle MicroWorks's (TMW's) software platform.

The primary objectives of this project include training the NREL team with TMW's embedded controller libraries, developing an interoperable communication code for embedded controllers, successfully testing, and deploying the code, and demonstrating the newly developed code in a conference.

Table of Contents

List of Figures

1 Introduction

Photovoltaic (PV) inverters and other inverter-based assets are being integrated into the distribution system at a face pace. Utilities operating the distribution system need to access information from these assets and control these assets in real time. Interoperability is a critical component toward enabling the sensing and control of assets (Kim et al. 2017). Without interoperability, utilities will require a larger number of resources to enable communications with different assets.

International Electrotechnical Commission (IEC) 618501 is used in the electric power industry to allow interoperability among the different intelligent electronic devices in a substation. Legacy PV inverters in the field might use other protocols like Distributed Network Protocol 3 (DNP3) or might not have communication capability (Nagarajan, Palmintier, and Baggu 2016).

With the increase in PV inverters in the field during the past 5 years, it is becoming crucial that inverters can communicate with peers (other smart devices) and with components that are at a hierarchy above the inverters (e.g., supervisory control and data acquisition (SCADA) systems, distributed controllers, and data managers). This project aims to develop a standard SCADA software code for inverters' embedded controllers that will enable interoperability with other components in the system. To achieve this, the code was developed using two different protocols: DNP3 and IEC 61850. The developed code aims to be deployed in simple embedded controllers (such as BeagleBone boards or Intel boards). The developed code was tested at the National Renewable Energy Laboratory's (NREL) Energy Systems Integration Facility (ESIF).

1.1 Objectives

1.1.1 Outcomes

The primary objectives of this project include developing an interoperable communication code for an embedded controller, successfully testing, and deploying the code, and demonstrating the capability of the developed code through use cases.

1.1.2 Deliverables

The final deliverable of the project is the microcontroller board with the DNP3 code, an ".icd" file for IEC 61850, and a microcontroller board with IEC 61850 compatibility.

1.1.3 End Use Application

The customer (or the targeted audience) can use the software code and the developed hardware to communicate with a PV inverter through substation automation interoperable protocols.

[1] https://www.naic.edu/~phil/hardware/sitePower/evd4/1MRK511242-UEN_-_en_Communication_protocol_manual__IEC_61850__650_series__IEC.pdf.

2 Interoperable Code Development

This report presents the development of software code that can enable interoperability in a PV inverter. Following are subsets of the advanced grid support functions available in a typical PV inverter (Seal and Ealey 2016):

- Measurement:
    - Voltage
    - Current
    - Frequency
    - Real power
    - Reactive power
    - DC measurements.
- Control:
    - Volt-volt ampere reactive (VAR)
    - Volt-watt
    - Frequency-watt.
- Protection:
    - Voltage ride-through
    - Frequency ride-through.

SCADA protocols are used to communicate with the PV inverters to acquire the status of the equipment and send control signals.

2.1 Software Requirements for the DNP3 and IEC 61850 Protocols

IEC 61850 protocol is traditionally used for substation automation (Mackiewicz 2006). It is used in the integration of protection, control, measurement, and monitoring functions in intelligent electronic devices (IEDs). Under this high-level protocol, IEC 61850 7-420 defines the logical nodes and the basic communication requirements for distributed energy resources (Cleveland 2008). Based on IEC 61850 7-420 documentation, the IEC 61850 protocol requirements for controlling a smart inverter was identified, and an IED Capability Description (ICD) file was created.

The configuration file to use this protocol is specified using the Substation Configuration Description Language (SCL). SCL is an Extensible Markup Language (XML)-based format to describe the elements of the power system. An ICD file is used to describe an IED. IEC 61850 has the following three protocols that can be used for mapping to application layer protocols:

- **MMS:** Manufacturing Message Specification (application layer protocol)
- **GOOSE:** Generic Object-Oriented Substation Event
- **SAV:** Sampled Value.

MMS is used for end-to-end communications from IEDs to the control center. MMS uses Ethernet and Transmission Control Protocol (TCP)/Internet Protocol (IP) to handle the information transport within the substation. GOOSE is typically used for reporting changes in data from the server to the client. SAV is used for communicating data at high speeds over

Ethernet. Both, GOOSE and SAV exchange information though the publisher/subscriber paradigm.

DNP3 follows IEC 61850 for the application note titled "DNP Application Note AN2018-001 – DNP3 Profile for Communications with Distributed Energy Resources." This document outlines the profile (analog and binary points) that are allocated for communication between a PV inverter (DNP3 server) and a SCADA system (DNP3 client). These functions are identified, and the point maps were identified for the software code development.

2.1.1 Software Requirements for the DNP3 Protocol

To develop the software code for the DNP3 protocol, the following software platform were utilized:

1. The DNP3 source code library from TMW.
2. Distribution test manager—a software simulator from TMW that can simulate the DNP3 client and server. This tool was used as a test SCADA platform to evaluate the DNP3 server code developed in the embedded systems.

2.1.2 Software Requirements for the IEC 61850 Protocol

To develop the software for the IEC 61850 protocol, the following software were utilized:

1. IEC 61850 protocol source-code libraries:
    A. MMS—the source code library for the MMS communication standard
    B. GOOSE—the source code library for the GOOSE communication standard
    C. SAV—the source code library for SAV communication standard.
2. SCL Navigator: This software from TMW was used to create the ICD configuration files for the devices under test.
3. Test Suite Pro: This software from TMW was used to simulate and test the IEC 61650 protocols. This serves as a client that is used to validate the application under development.

2.2 Setups Used to Evaluate the Software Codes Developed:

To evaluate the software codes developed, a simulation testbed and a hardware testbed were used.

2.2.1 Simulation Test Bed

The simulation test bed was set up to develop and validate the code in a simulation-only environment before moving to the hardware setup. It is shown in Figure 1. The setup runs the code in real time and uses a laptop to mimic the inverter controller actions, whereas the hardware setup uses an sbRIO inverter controller that was developed in the Additively Manufactured Photovoltaic Inverter (AMPVI) project funded by the U.S Department of Energy Solar Energy Technologies Office (Chinthavali et al. 2020). The simulation test bed consists of the following tools:

- TMW's Test Suite Pro to mimic a DSO that sends various control signals and monitors the health of the IEDs

- TMW's DTM to mimic a DNP3 capable inverter controller that is used to test the code being developed
- A Linux machine with arm-cross-compiler capabilities to develop and test the code for this project.

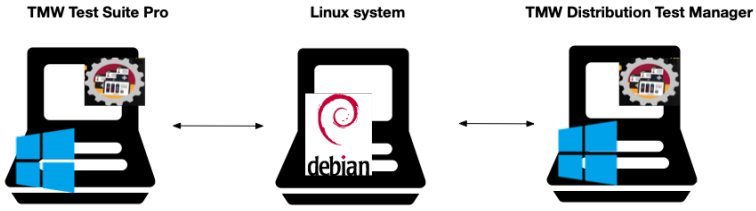


Figure 1. Simulation test bed used to evaluate the interoperability code

2.2.2 Hardware Test Bed

The hardware test bed has all the components of the simulation environment. In addition, it has the following:

- A microcontroller that will host the interoperable code
- An sbRIO controller that hosts the DNP3-capable (DNP3 server) PV inverter controller.

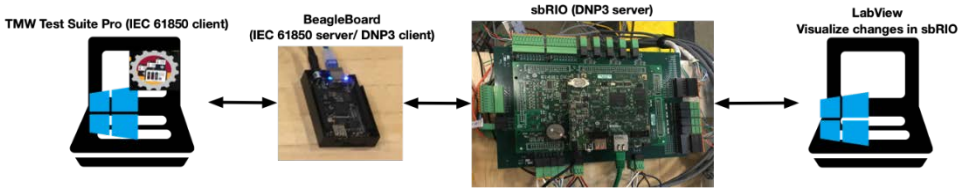


Figure 2. Hardware setup used to evaluate the developed interoperability code *Photos by the authors*

The interoperable code was also tested over an Anterix 900-MHz wireless private LTE communication channel. An experimental research license for the 900-MHz network was obtained to run experiments over a private LTE wireless in the ESIF at NREL (Raszmann et al. 2020). We tested the communication of IEC 61850 and DNP3 signals over the wireless network. The use of wireless communication for SCADA protocols like IEC 61850 and DNP3 can make it easier to integrate DERs such as PV in remote locations.

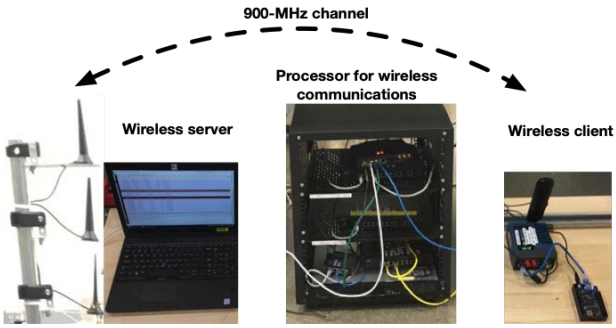


Figure 3. Experimental setup using 900-MHz communication *Photos by the authors*

3 Experimental Validation of the Developed Code

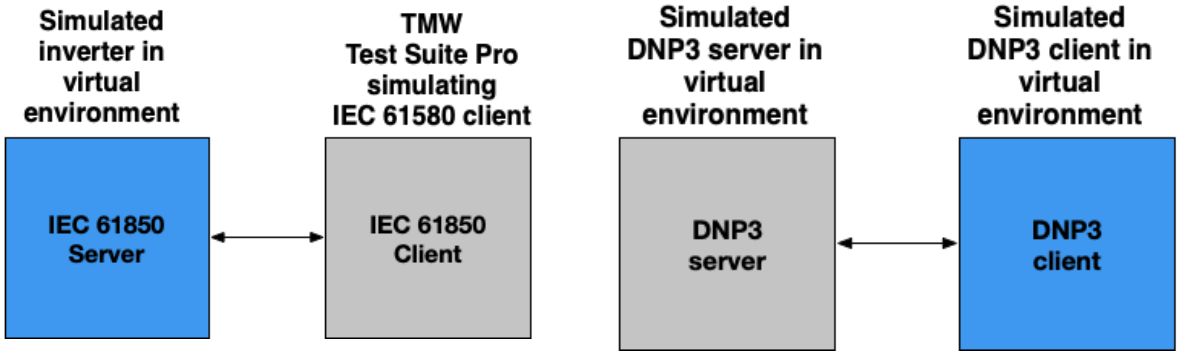
The software code to achieve interoperability between the IEC 61850 and DNP3 protocols was validated in a controller hardware setup at the ESIF.

3.1 Initial Experiments to Validate the Simple Interoperability Code

The first step in enabling IEC 61850 interoperability is the development of an xml-based ICD file. This ICD was developed for a standard PV inverter with advanced grid support functionality. An ICD file with the following functionality was created using the SCL Navigator:

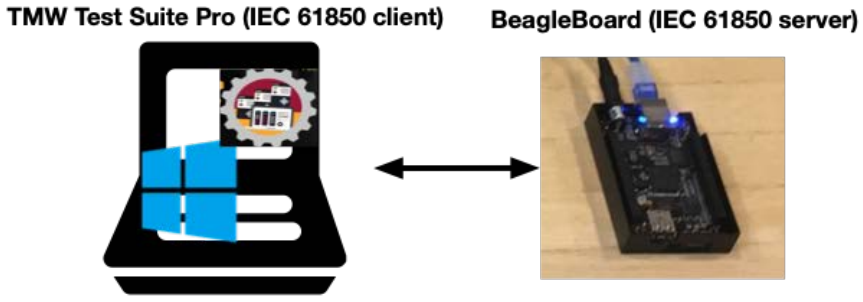
1. Volt-VAR curve
2. Volt-watt curve
3. Frequency-watt curve
4. Voltage ride-through
5. Frequency ride-through
6. Measurements from the inverter (voltage, current, power, and others).

This developed ICD file was used by a server application to enable IEC 61850 protocol. A server application in "C" programming language was developed in a Linux virtual environment, shown in Figure 4(a). This was validated using the client from TMW's Test Suite Pro.



(a) Simple experiment to evaluate the developed initial IEC 61850 code (IEC 61850 client-server simulation)

(b) Simple experiment to evaluate the developed initial DNP3 code (DNP3 client-server simulation)



(c) Application development setup

Figure 4. Experimental setup for initial validation tests to evaluate the developed code *Photos by the authors*

A simple DNP3 client was developed using TMW's DNP3 source code. This simple client was evaluated for proper performance using the Device Type Manager (DTM) software from TMW, as shown in Figure 4(b).

The client-server setup for DNP3 and IEC 61850 was validated using the setup shown in Figure 4(c). The project team ran simple shakedown experiments to test IEC 61850-7-420 codes in the BeagleBone board. The shakedown testing evaluated the normal operation of communication between two IP addresses using the IEC 61850 protocol over Ethernet. MMS, GOOSE, and SAV were communicated between the client and server.

3.2 Communication between AMPVI's Inverter Controller Board and the Embedded Board

3.2.1 Requirements for Interoperable Communication Testing

The NREL team along with the TMW team identified the requirements for a translator that can exchange data between the IEC server and the DNP3 client inside the embedded controller. This translator is not only critical to enable communication to the AMPVI controller board but also a general requirement for most legacy PV inverters. This is because most legacy inverters have the capability to communicate via DNP3 but lack the necessary libraries to communicate following IEC 61850.

The IEC 61850 server and DNP3 clients were augmented to exchange information between the two protocols. Two sets of codes were developed in this project. The first code can be used directly to enable the IEC 61850 server on a PV inverter. The second code converts the data from the IEC 61850 client, initiates communication with a DNP3 client, and exchanges information with a DNP3 server through the DNP3 client. This enables a distribution system operator (DSO) to seamlessly communicate to IEC 61850 and DNP3 distributed energy resources, as shown in Figure 5.

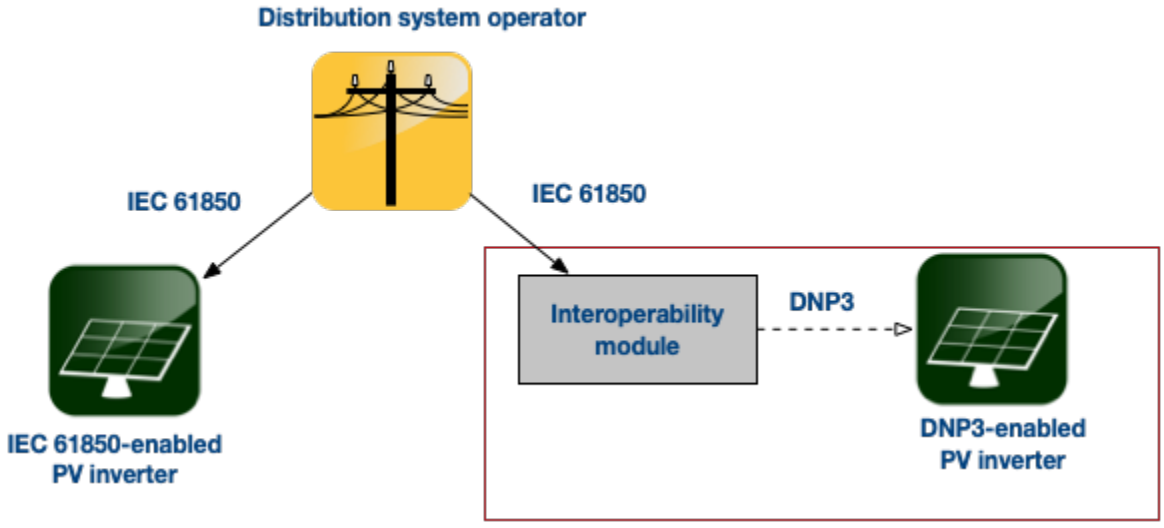


Figure 5. Usability of the codes developed by directly enabling IEC 61850 on a legacy PV inverter and on a DNP3 inverter

The communication flow to achieve interoperable communications is presented in Figure 6.

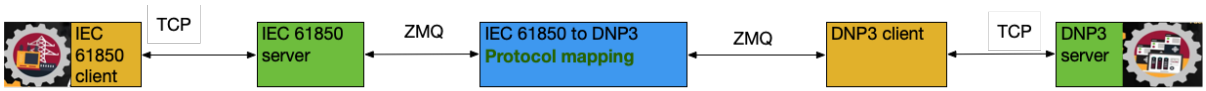


Figure 6. Communication workflow to enable communications between a PV inverter and an IEC 61850 client

The IEC 61850 configuration file was developed using TMW's SCL Navigator software. The DNP3 device register mappings for the simulation test bed were created using the DTM. The data flowing over the network communications was verified using the Wireshark network analysis tool.

Shown in Figure 6, ZeroMessage Queuing (ZeroMQ) is an open-source messaging library developed for use in distributed applications (Hintjens 2013). This library was used to translate from IEC 61850 to DNP3 and vice versa. The project team used the ZeroMQ request-reply pattern to create a message bus that enables this translation.

The protocol translation module contains the mapping points to exchange information from IEC 61850 to DNP3. The advantage of the developed module is that it can accept different mapping files. This makes it easier to translate between legacy DNP3 point maps or newer DNP3 point maps and IEC 61850 protocols.

Different types of PV inverter control methods were implemented and verified in the simulation test bed and in the hardware setup at the ESIF. The following PV inverter controls were evaluated:

- Volt-VAR
- Volt-watt
- Over-voltage
- Under-voltage
- Freq-watt
- Over-frequency
- Under-frequency.

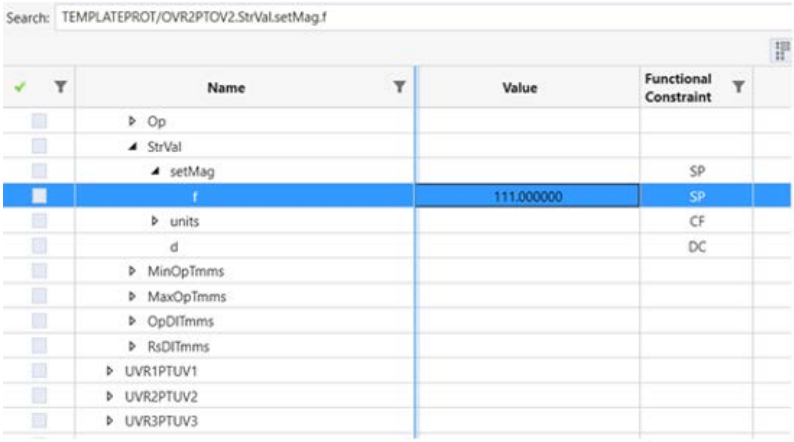
3.2.1.1 Communication from the IEC 61850 Client to the DNP3 Server

C programming language was used to develop the software code that translated the data from the IEC 61850 client and pushed it to the DNP3 server. The protocol translator converts this message to a data structure containing name, index, and value. This is encoded in a ZeroMQ message inside the embedded system (the embedded system was simulated in a laptop environment) and sent to its corresponding counterpart on the DNP3 client. The data field and index are used to map these data to the appropriate DNP3 data file and index in the DNP3 client.

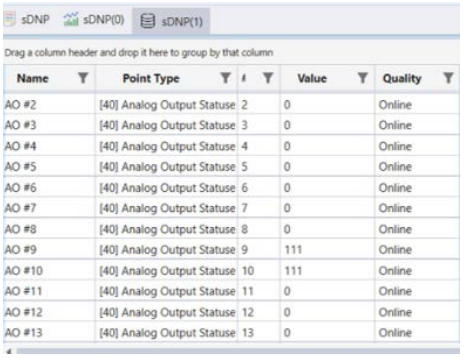
The DNP3 client in the embedded system (shown in Figure 4) packs the ZeroMQ message in DNP3 format and exchanges it with the DNP3 server modeled in the DTM tool. This DTM-based DNP3 server will be replaced with the DNP3 server in the Single Board Reconfigurable Input Output (sbRIO) board controlling the AMPVI inverter (MacCleery et al. 2012), (Prabakar

et al. 2017). The sbRIO board is a commonly used board for developing controls. This communication flow indicates communication in one direction. Messages from the DNP3 server to the IEC client follow the same approach but in the reverse order.

Successful communication from the IEC 61850 client to the DNP3 server is shown in Figure 7. Communication from the IEC 61850 MMS client to the DNP3 server . Figure 7. Communication from the IEC 61850 MMS client to the DNP3 server (a) shows the "write" command initiated in the IEC 61850 client. Figure 7. Communication from the IEC 61850 MMS client to the DNP3 server (b) shows the value "written" by the client received on the server side at AO 9.



(a) IEC 61850 client initiating the write command



(b) DNP3 server receiving the value

Figure 7. Communication from the IEC 61850 MMS client to the DNP3 server

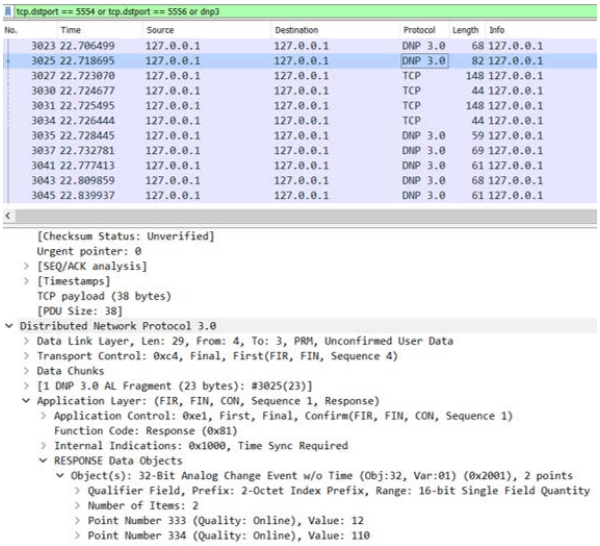
3.2.1.2 Communication from the DNP3 Server to the IEC 61850 Client

The DNP3 server sends messages as DNP3 packets to the DNP3 client. The protocol translator on the embedded system program converts the message received by the DNP3 client to a data structure that indicates the data file name, index, and value. This is encoded in a ZeroMQ message and sent to its corresponding counterpart on the IEC 61850 client.

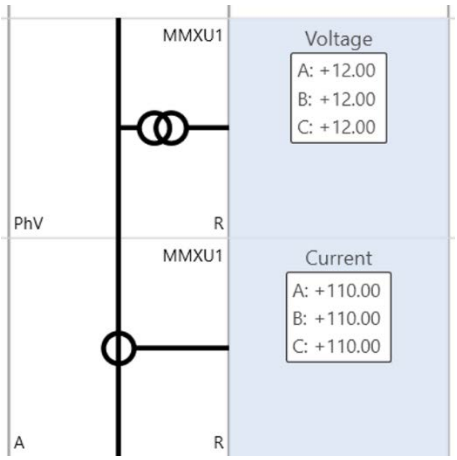
This message is split into the appropriate components: data field, index, and value. The information in the message is used to map the data to the appropriate IEC 61850 field using "S

address” mapping, which is then communicated with the IEC 61850 client over Ethernet, thereby completing a one-way communication from the DNP3 server to an IEC 61850 client.

The IEC 61850 server application in the C programming language was developed in a Linux virtual environment, shown in Figure 1. This was validated using the client from TMW’s Test Suite Pro. Successful communication from the DNP3 server to the IEC 61850 client is shown in Figure 8. Figure 8 (a) shows the data exchanged from the DNP3 server to the IEC 61850 client in point number 333 and point number 334. Figure 8 (b) shows the value “written” by the server and received on the IEC 61850 client side at the corresponding voltage and current measurements.



(a) Wireshark capture of the DNP3 server sending measurement values



(b) IEC 61850 client displaying the received values

Figure 8. Communication from the DNP3 server to the IEC 61850 client

The two-way communication between the IEC 61850 client and the DNP3 server is shown in Figure 9.

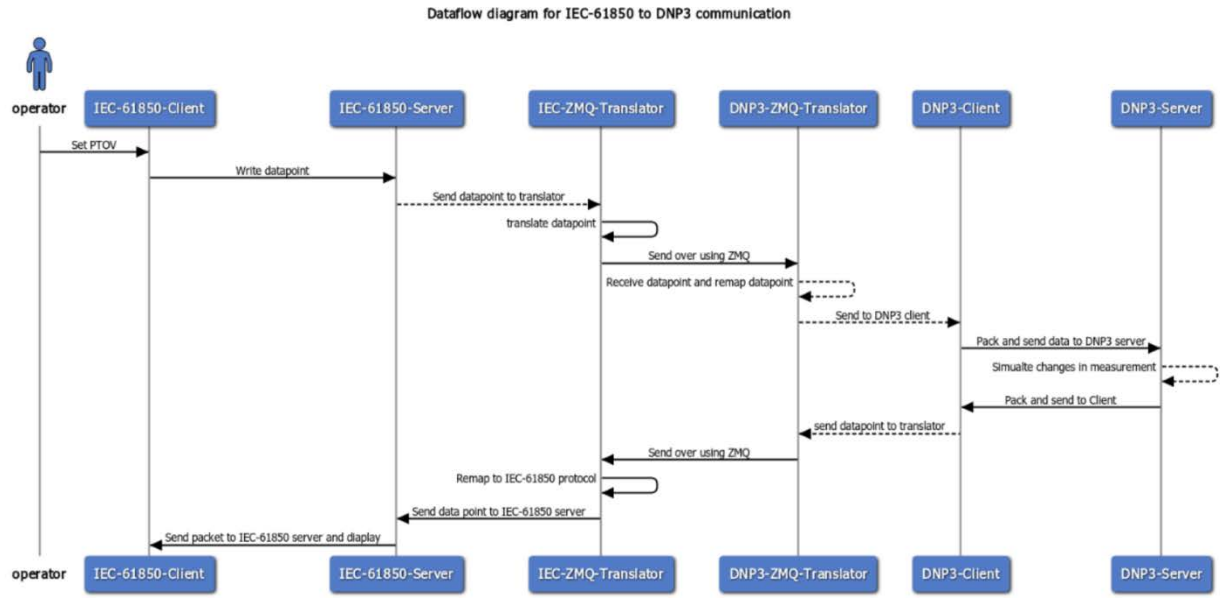


Figure 9. Data flow between the IEC 61850 MMS client and the DNP3 server

3.2.2 Details on Interoperability Code

The ICD file was created using the SCL Navigator tool from TMW. The addresses of the values of interest were then mapped to the ICD file. We needed to ensure that we assigned S addresses based on the data type of the parameter being modified. New data types can be added to the code through the XML file that accompanies the source code. Further information on assigning S addresses can be found in the TMW source code library manual. An example of adding the S address is shown in the following as a code snippet:

```
        <LN lnType="PTOV_VoltageRideThrough" lnClass="PTOV" inst="1" prefix="OVR1" desc="OV 1 Moment
ary cessation">
          <DOI name="StrVal">
           <SDI name="setMag">
            <DAI name="f" sAddr="AV18">
              <Val>110</Val>
            </DAI>
           </SDI>
OVR1PTOV1$SP$StrVal$setMag$f:pDAI->p_sAddr      = AV18 (init)
```

During the initialization phase, a map of the argument structure and the S address is created. The JavaScript Object Notation (JSON) library is used to create this dictionary. A snippet of the mapping dictionary between the logical node and S address created is shown in the following:

```
{"AV5": "MMXU1$MX$TotW$mag$f", "Q0": "MMXU1$MX$TotW$q", "T0": "MMXU1$MX$TotW$t",
"R0": "MMXU1$CF$TotW$db", "AV2": "MMXU1$MX$TotVAr$instMag$f", "AV3":
"MMXU1$MX$TotVAr$mag$f", "Q1": "MMXU1$MX$TotVAr$q", "T1": "MMXU1$MX$TotVAr$t",
"D9": "MMXU1$CF$TotVAr$db", "AV0": "MMXU1$MX$Hz$instMag$f", "Q2":
"MMXU1$MX$Hz$q", "T2": "MMXU1$MX$Hz$t", "D10": "MMXU1$CF$Hz$db", "Q3":
"MMXU1$MX$PhV$phsA$q", "T3": "MMXU1$MX$PhV$phsA$t", "D11":
"MMXU1$CF$PhV$phsA$db", "Q4": "MMXU1$MX$PhV$phsB$q", "T4": "MMXU1$MX$PhV$phsB$t",
"D12": "MMXU1$CF$PhV$phsB$db", "Q5": "MMXU1$MX$PhV$phsC$q", "T5":
"MMXU1$MX$PhV$phsC$t", "D13": "MMXU1$CF$PhV$phsC$db", "AV10":
"OFR1PTOF1$SP$StrVal$setMag$f", "Q6": "MMXU1$MX$A$phsA$q", "T6":
"MMXU1$MX$A$phsA$t", "D14": "MMXU1$CF$A$phsA$db", "Q7": "MMXU1$MX$A$phsB$q",
"T7": "MMXU1$MX$A$phsB$t", "D15": "MMXU1$CF$A$phsB$db", "Q8":
"MMXU1$MX$A$phsC$q", "T8": "MMXU1$MX$A$phsC$t", "D16": "MMXU1$CF$A$phsC$db",
"AV14": "MMXU1$DC$A$d", "AV18": "OVR1PTOV1$SP$StrVal$setMag$f", "DV0":
"OVR1PTOV1$SP$OpDlTmms$setVal", "DV1": "OVR1PTOV1$CF$OpDlTmms$minVal", "DV2":
"OVR1PTOV1$CF$OpDlTmms$maxVal", "DV3": "OVR1PTOV1$SP$RsDlTmms$setVal", "BL1":
"OVR1PTOV1$ST$Str$general", "BL0": "OVR1PTOV1$ST$Op$general", "AV20":
"OVR2PTOV2$SP$StrVal$setMag$f", "AI3": "OVR2PTOV2$SP$OpDlTmms$setVal", "":
"UFR1PTUF1$SP$RsDlTmms$setVal", "AV21": "UVR1PTUV1$SP$StrVal$setMag$f", "AV22":
"UVR1PTUV1$SP$OpDlTmms$setVal", "AV26": "UVR2PTUV2$SP$StrVal$setMag$f", "AV37":
"UVR2PTUV2$SP$OpDlTmms$setVal", "AV38": "UVR3PTUV3$SP$StrVal$setMag$f", "AV19":
"UVR3PTUV3$SP$OpDlTmms$setVal", "AV110": "OFR1PTOF1$SP$OpDlTmms$setVal", "AV120":
"OFR2PTOF2$SP$StrVal$setMag$f", "D6": "OFR2PTOF2$SP$OpDlTmms$setVal", "R13":
"OFR2PTOF2$SP$RsDlTmms$setVal", "AV130": "UFR1PTUF1$SP$StrVal$setMag$f", "D7":
"UFR1PTUF1$SP$OpDlTmms$setVal", "AF14": "UFR2PTUF2$SP$StrVal$setMag$f", "D8":
"UFR2PTUF2$SP$OpDlTmms$setVal", "AV15": "UFR2PTUF2$CF$OpDlTmms$minVal", "AV16":
"UFR2PTUF2$CF$OpDlTmms$maxVal", "AV17": "UFR2PTUF2$SP$RsDlTmms$setVal"}
```


The IEC 61850 server and the DNP3 clients have been set up with functions that monitor messages from the IEC 61850 client and the DNP3 server, respectively. The IEC 61850 server sends a message to the protocol translator when it receives a message from the DSO. The message is mapped to the respective DNP3 register and sent to the DNP3 client running on the BeagleBoard controller. The CSV file that maps between the two protocols is loaded in the protocol translator. The DNP3 client contains a module that communicates with the protocol translator. The DNP3 client has a listener function that waits for IEC 61850 mapped messages from the protocol translator. These messages are then relayed to the DNP3 server implemented in the sbRIO controller. The DNP3 client also has the capability to push the measurements it receives from the DNP3 server to the IEC 61850 server through the protocol translator. The measurements are mapped back to the MMXU logical device in the IEC 61850 server.

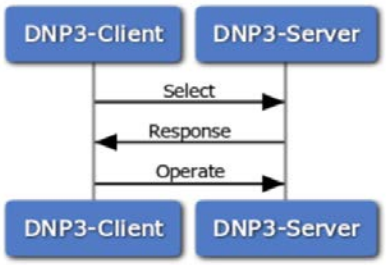


Figure 10. Typical select-before-operate in DNP3

Figure 10 shows a typical communication between the DNP3 client and DNP3 serve for a "select-before operate" (Hunt and Popescu 2015), (Weerathunga and Cioraca 2016) communication flow. In the select-before-operate, two commands are sent in sequence. The select command is used to reserve the resource and validate the control. The operate command is used act on the selected resource. The messages exchanged between the devices are captured in Wireshark and shown in Figure 11 and Figure 12.

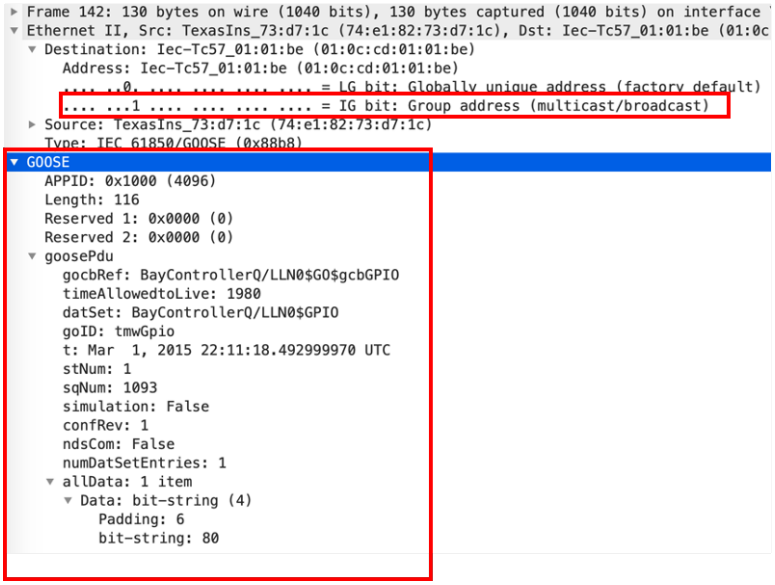


Figure 11. Wireshark data capture of GOOSE messaging between the IEC 61850 client and the IEC 61850 server

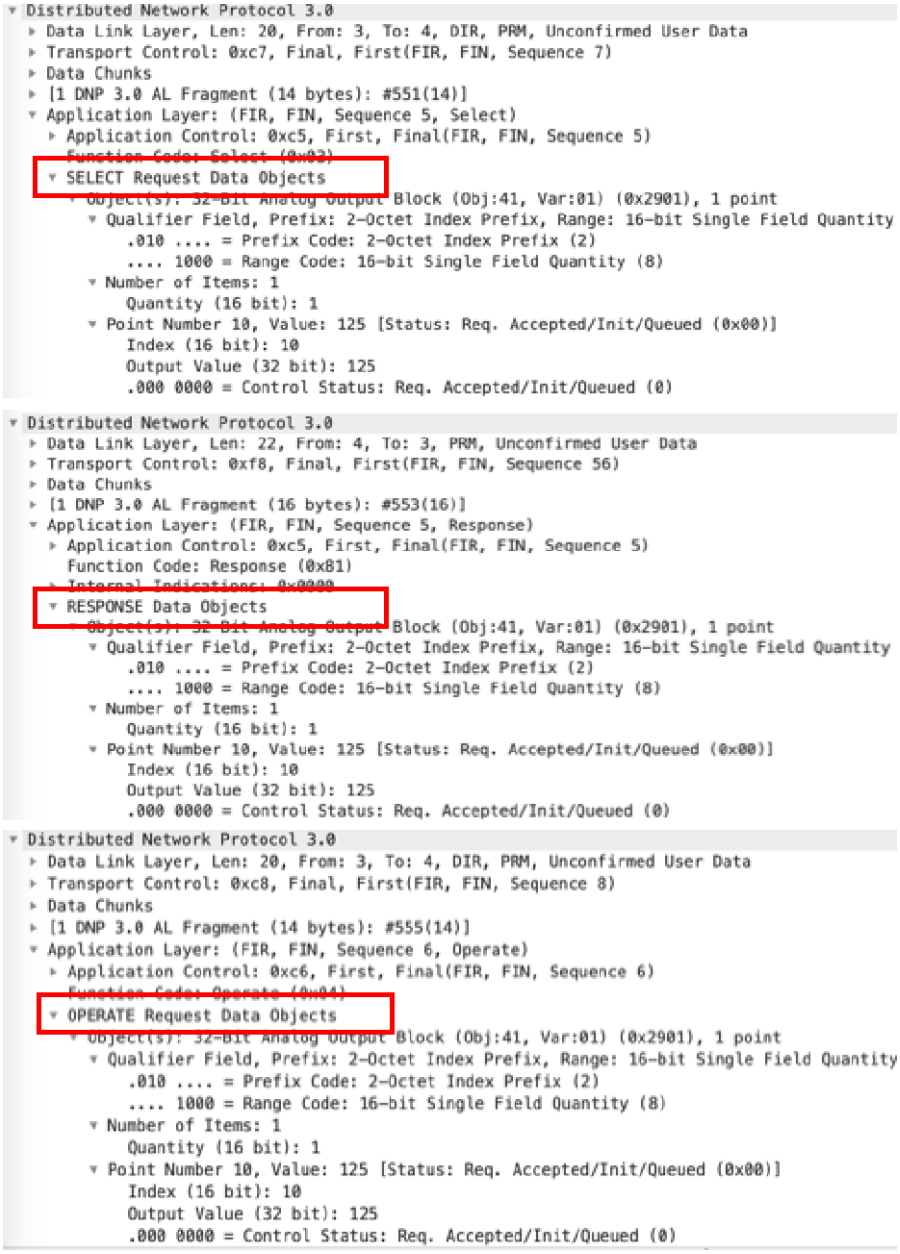


Figure 12. Wireshark data capture of select-before-operate communication using the developed software code

Figure 11 shows the Wireshark data capture of the communication between the IEC client and the IEC server through GOOSE messaging. Figure 12 shows the Wireshark data capture of the communication between the DNP3 client and the DNP3 server. Figure 12 uses the select-before-operate approach available under the standard DNP3 communication structure. Figure 13 and Figure 14 show the sequence diagram for the typical curve functions updated by the DSO. This includes volt-VAR curve and volt-watt curve settings. These sequence diagrams show the operation sequence when the DSO operator tries to update the curve. The protocol translator conveniently translates the message to the inverter controller.

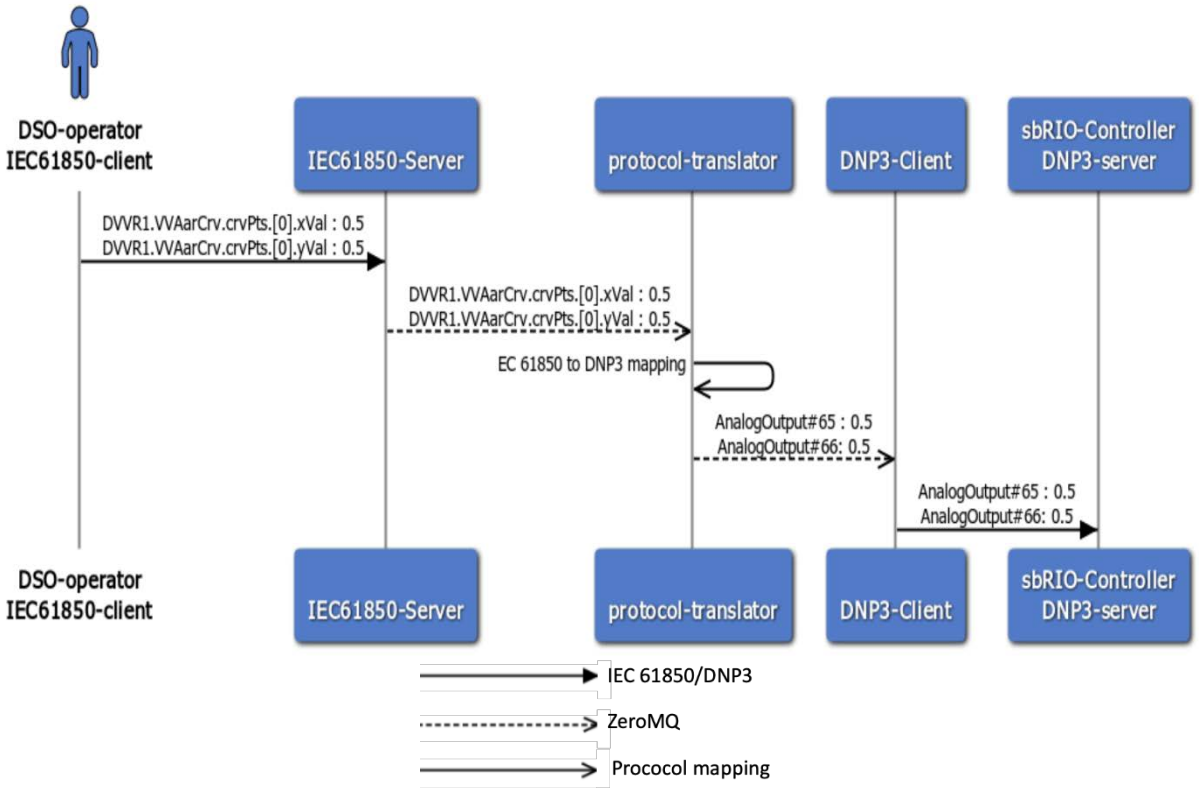


Figure 13. Volt-VAR curve update sequence diagram

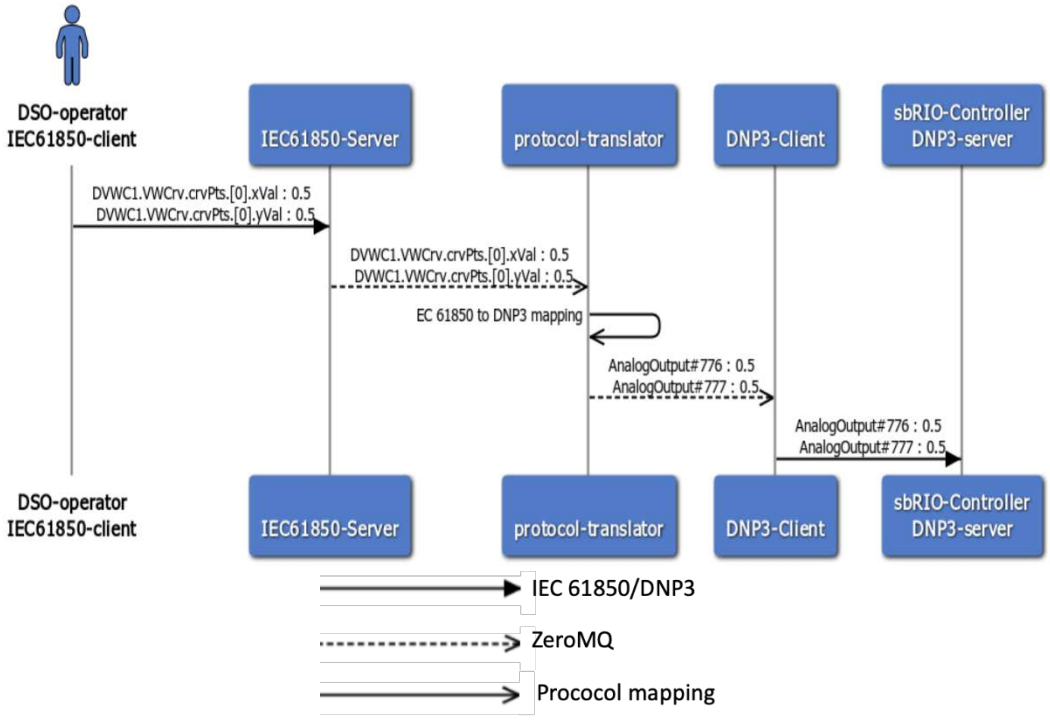


Figure 14. Volt-watt curve update sequence diagram

The schematic of the DNP3 server in the sbRIO board is shown in Figure 15.

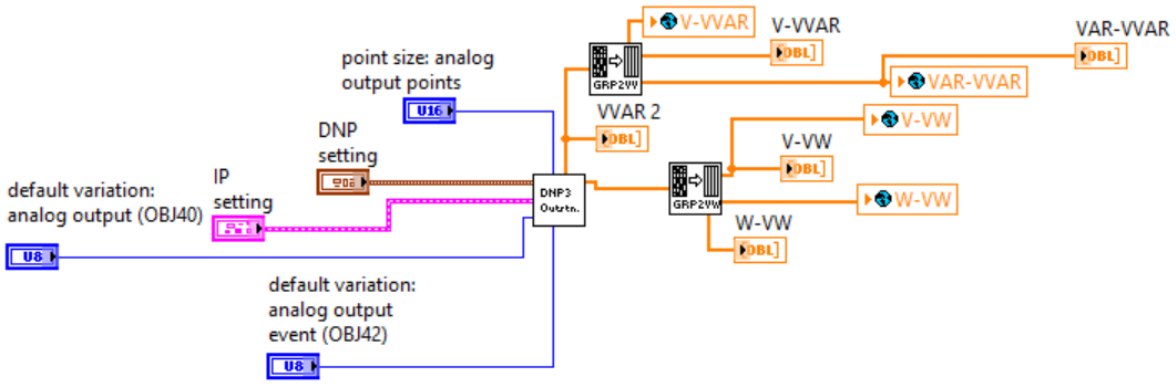


Figure 15. DNP3 server code in the sbRIO inverter controller

The communication of messages between a DNP3 client and the sbRIO server was verified by plotting the volt-VAR value change from the sbRIO controller. This is shown in Figure 16.

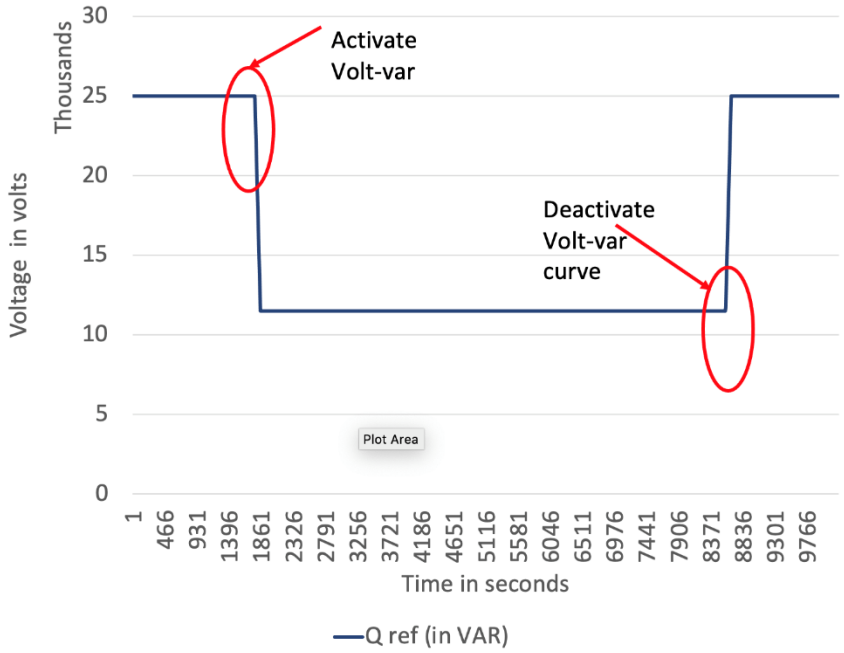


Figure 16. Volt-VAR curve being updated in the inverter controller by the central controller through the developed communication code

We developed an IEC 61850 server using the TMW's source code library, that is capable of acting as an IEC 61850 capable inverter. We also added a functionality to convert the IEC 61850 protocol to DNP3 protocol. A DNP3 client that can communicate with a DNP3 server was developed using TMW's source code library. The DNP3 client can communicate with the DNP3 server and the protocol translator modules. Hence, we enabled an IEC 61850 client to talk to an IEC 61850 capable inverter or a DNP3 capable inverter. The BeagleBoard was used to implement the protocol translator module that also includes the modified IEC 61850 server and DNP3 client. The interoperable code developed in this project was test in a controller hardware setup where the PV inverter controller was implemented using the sbRIO board.

4 Summary

The development of interoperable SCADA protocols for PV inverters will lead to wider adoption of grid-interactive PV inverters by the utilities leading to higher penetration of DERs in the grid. The interoperable module developed in this research work enables the participation of legacy inverters in the advanced distribution management functions. In addition, the interoperability code developed in this research work reduces the cost of adding additional communication protocols by inverter manufactures, which can result in cost savings to the customers.

In this research work, we developed, and evaluated a prototype code for standard SCADA software to interface to the inverters' embedded controllers. The NREL team successfully identified interoperability requirements for both DNP3 and IEC 61850 protocols, as shown in Section 2. These requirements were used toward developing the software code, which was evaluated as shown in Section 3. Communication was tested between the AMPVI inverter controller board and the BeagleBone board using both DNP3 and IEC 61850 protocols, both via wired and wireless communication methods. The code was refined based on these experimental results.

References

Chinthavali, Madhu Sudhan (ORCID:0000000282340943), Steven (ORCID:0000000304116812) Campbell, Mariko Shirazi, Sudipta Chakraborty, Akanksha Singh, Kumaraguru Prabakar, and Colin Tombari. 2020. "Additively Manufactured Photovoltaic Inverter (AMPVI)." ORNL/SPR-2019/1337. Oak Ridge National Lab. (ORNL), Oak Ridge, TN (United States). https://doi.org/10.2172/1606907.

Cleveland, F. M. 2008. "IEC 61850-7-420 Communications Standard for Distributed Energy Resources (DER)." In *2008 IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century*, 1–4. IEEE.

Hintjens, Pieter. 2013. *ZeroMQ: Messaging for Many Applications*. O'Reilly Media, Inc.

Hunt, R., and B. Popescu. 2015. "SCADA Communications for Protection Engineers." In *2015 68th Annual Conference for Protective Relay Engineers*, 881–91. https://doi.org/10.1109/CPRE.2015.7102211.

Kim, Dae-Kyoo, Alaa Alaerjan, Lunjin Lu, Hyosik Yang, and Hyuksoo Jang. 2017. "Toward Interoperability of Smart Grids." *IEEE Communications Magazine* 55 (8): 204–10. https://doi.org/10.1109/MCOM.2017.1600392.

MacCleery, Brian, Olivier Trescases, Muris Mujagic, Damon M. Bohls, Oleg Stepanov, and Garret Fick. 2012. "A New Platform and Methodology for System-Level Design of next-Generation FPGA-Based Digital SMPS." In *2012 IEEE Energy Conversion Congress and Exposition (ECCE)*, 1599–1606. IEEE.

Mackiewicz, Ralph E. 2006. "Overview of IEC 61850 and Benefits." In *2006 IEEE Power Engineering Society General Meeting*, 8-pp. IEEE.

Nagarajan, Adarsh, Bryan Palmintier, and Murali Baggu. 2016. "Advanced Inverter Functions and Communication Protocols for Distribution Management." In *2016 IEEE/PES Transmission and Distribution Conference and Exposition (T&D)*, 1–5. IEEE.

Prabakar, Kumaraguru, Mariko Shirazi, Akanksha Singh, and Sudipta Chakraborty. 2017. "Advanced Photovoltaic Inverter Control Development and Validation in a Controller-Hardware-in-the-Loop Test Bed." In *2017 IEEE Energy Conversion Congress and Exposition (ECCE)*, 1673–79. IEEE.

Raszmann, Emma, Kumaraguru Prabakar, Barry Mather, and Jim Li. 2020. "Experimental Test Bed to Enable Realistic Evaluations for Direct Transfer Trip Relaying via Private Wireless LTE Communications." In *2020 IEEE International Smart Cities Conference (ISC2)*, 1–6. IEEE.

Seal, B., and B. Ealey. 2016. "Common Functions for Smart Inverters." In *EPRI, Knoxville, TN*.

Weerathunga, P. E., and A. Cioraca. 2016. "The Importance of Testing Smart Grid IEDs against Security Vulnerabilities." In *2016 69th Annual Conference for Protective Relay Engineers (CPRE)*, 1–21. https://doi.org/10.1109/CPRE.2016.7914920.