Cape Peninsula
University of Technology

DEVELOPMENT OF AN IEC 61850 BASED POWER QUALITY MONITORING DEVICE

by

MAYAMIKO HARA

Thesis submitted in partial fulfilment of the requirements for the degree

MASTER OF ENGINEERING: ELECTRICAL ENGINEERING: SMART GRID

in the FACULTY OF ENGINEERING AND THE BUILT ENVIRONMENT

at the Cape Peninsula University of Technology

Supervisor: Dr C. Kriger

Bellville Campus

November 2020

DECLARATION

I, Mayamiko Hara, declare that the contents of this thesis represent my unaided work and that the thesis has not previously been submitted for academic examination towards any qualification. Furthermore, it represents my own opinions and not necessarily those of the Cape Peninsula University of Technology.

Signed   Date

ABSTRACT

The term “Smart Grid” holds varying definitions in the increasingly digitised technological landscape of energy generation, distribution, control and monitoring. Three defining characteristics of the “Smart Grid” that are common among existing definitions are the following:

1) The integration of Information and Communications Technology (ICT) to improve grid reliability, efficiency and situational awareness.
2) To utilise advanced metering infrastructure (AMI) for bidirectional communications with customer meters and encourage customer participation.
3) To enhance the control, monitoring and optimisation of distributed energy resources (DERs) and bidirectional power flow in an increasingly decentralised electric power grid.

The decentralisation of generation plants mainly stems from the adoption of Small-scale Embedded Generators (SSEG), or Distributed Energy Resources (DERs) as they are more commonly known. DERs pose a unique challenge to Network Operators within the “Smart Grid” and may potentially jeopardise grid reliability. The use of power electronic converters coupled with the often-bidirectional nature of DERs power flow, along with the inconsistent and intermittent generation characteristics of renewable energy offer unique challenges in maintaining and monitoring power quality.

Due to recent regulatory changes and the increasingly constrained ESKOM generation infrastructure; South African municipalities have seen the benefit of employing more renewable energy and encourage customer participation to lessen the strain and improve grid reliability. The increasing number of small-scale generation plants in South Africa would, in future, require a solution that is interoperable with existing substation automation hardware and communications infrastructure, but also cost-effective.

This research project presents a methodology for designing, modelling and implementing an IEC 61850-based power quality monitoring embedded systems device. Primarily aimed at systems with a large number of low-voltage Small-scale Embedded Generation plants; the goal is to achieve a balance between cost-effectiveness and power quality monitoring standards compliant performance, while crucially maintaining full interoperability with existing grid infrastructure.

The thesis deliverables provide a novel proof of concept for a power quality monitoring device that may be deployed and modified depending on the use case and specific application. The following research contributions are identified:

1) Modelling and implementation of a viable, interoperable and low-cost IEC 61850-based power quality monitoring device using off-the-shelf components and an open-source software library. The implementation of this device provides a platform that is easily modifiable for use in a range of applications.
2) The method of modelling IEC 61850 logical nodes and functions to analogue front-end measurement registers presented in this research demonstrates how to design and map information to equivalent logical nodes. The knowledge demonstrated shows how device modelling techniques can be used to extract additional information for mapping to existing logical nodes or to develop new ones.

The thesis findings and deliverables result in a low-cost IEC 61850-based power quality monitoring device. The device may additionally be used for general utility-scale monitoring of DER plants, check metering, IEC 61850-based power quality mitigation algorithms and short-term power quality studies. It may also function as a low-cost entry point to IEC 61850-based communications in smaller municipalities, and additionally as a lab test and prototyping tool for technicians and engineers.

The research work may also contribute to educating university student on topics such as Linux operating systems, networking, embedded systems, substation automation and the IEC 61850 standard. It may also find use as a prototyping platform for new logical nodes.

The device proposed in this research project has the potential to act as an enabling technology for the adoption of DERs due to lowering the barriers of entry in terms of cost and functionality.

Keywords: Power quality monitoring, distributed energy resources, renewable energy, IEC 61850, GOOSE Messaging, MMS, SCADA, Communications infrastructure, Open-Source Software, Single-Board Computers, Linux Operating System.

ACKNOWLEDGEMENTS

I wish to thank:

- My partner Elizabeth for her unending support and patience during this very challenging and stressful period of my life.
- My parents for providing me with the means to achieve this goal through their sacrifice and dedication.
- The Centre for Substation Automation and Energy Management Systems at (CSAEMS) CPUT for providing the necessary support, expertise and equipment that allowed me to complete my research project.
- The MEng: Smart Grid coursework facilitators and lecturers for imparting the knowledge necessary for me to complete this research.
- My supervisor Dr Carl Kriger for the insight, guidance and motivation he provided me throughout my journey; and a special thank you to Professor Raynitchka Tzoneva, Head of the CSAEMS, whose enthusiasm and belief in me was a substantial motivating factor.

Mayamiko Hara

Bellville, 2020

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

Digital Signal Processor

A specialised microprocessor module designed for digital signal processing and used to measure, filter and compress analogue signals for further processing and interpretation.

Distributed Energy

Generation provided by small, decentralised generation plants that are usually grid-connected. A group of distributed generators are known as distributed energy resources (DERs)

Electromagnetic Compatibility

Defined as the compatibility of a given device with the electromagnetism (EM) in its environment as well as its lack of electromagnetic emission

GOOSE Messaging

A multi-cast, high-speed communications model for transmitting time-critical data in an electrical substation network as described in the IEC 61850 standard

IEC 61850 Standard

An international standard that defines communication protocol specifically for an intelligent electronic device at the substation level. The protocols are TCP/IP based and designed to run over high-speed substation local area networks and switched ethernet.

Intelligent Electronic Device

A microprocessor-based device used to control equipment such as transformers, circuit-breakers, disconnectors and capacitor banks

Manufacturing Message Specification

An interoperable IP based international standard for exchanging control information between computer systems, control systems and intelligent devices.

Open-Source Software

A type of software whose source-code may be used, studied, changed and distributed to anyone. The license or copyright holder determines the open-source nature.

Power Quality

The power quality is the grid’s ability to supply a stable, clean power supply defined by a pure sinusoidal wave operating in specified voltage and frequency ranges.

Power Quality Monitoring

The monitoring of power quality with the aim of improving performance and maintaining the quality of supply. The data gathered by a power quality monitor may be used for preventative maintenance, which could potentially save money

Power-Electronic Converter

A solid-state electronic device used to control and convert electrical signals to different forms or changing the magnitude; e.g. from AC to DC, DC to AC or DC to DC. Devices typically operate at high frequency due to the switching action used in the conversion process.

SCADA

Allows the system operator to monitor and control electric grids, generation plants and other control processes in a distributed environment.

Single-Board Computer

A full computer system integrated into a single printed circuit board. They are usually constructed using System-on-chip (SoC) and ARM (Advanced Risk Machines) architecture.

Smart Grid

An electricity network that enables the two-way flow of electricity and data using advanced ICT technologies. This allows the control system to detect, react and pro-act to changes in the network. The self-healing capabilities of Smart Grids and bidirectionality of smart meters enable customers to become active participants.

Total Harmonic Distortion

The measurement of harmonic distortion in a given signal. It is the ratio of all summed powers of all harmonic components to the power of the fundamental frequency.

NOMENCLATURE

A	Amps
AC	Alternating Current
ACSI	Abstract Communication Service Interface
ADC	Analogue to Digital Converter
ADE	Analog Devices
AFE	Analog Front-End
AMI	Advanced Metering Infrastructure
ANSI	American National Standards Institutes
API	Application Programming Interface
ARM	Advanced RISC Machine
CID	Configured IED Descriptor
CMV	Complex Measured Value
COMTRADE	Common format for Transient Data Exchange
CORBA	Common Object Request Broker Architecture
CRT	Cathode Ray Tube
CSAEMS	Centre for Substation Automation and Energy Management Systems
CT	Current Transformer
DA	Data Attribute
DAC	Digital to Analog Controller
DAQ	Data Acquisition
DC	Direct Current
DERs	Distributed Energy Resources
DG	Distributed Generation
DHCP	Dynamic Host Control Protocol
DO	Data Object
DSL	Digital Subscriber Line
DSO	Distribution System Operator
DSP	Digital Signal Processing
EMC	Electromagnetic Compatibility
EMP	Electromagnetic Pulse

EMS	Energy Management System
EN	European Standards
FC	Functional Constraint
FCDA	Functional Constraint Data Attributes
FPGA	Field Programmable Gate Array
FTP	File Transfer Protocol
GNU	GNU's Not Linux
GoCB	GOOSE Control Block
GOMSFE	Generic Object Models for Substation and Feeder Equipment
GOOSE	Generic Object-Oriented Substation Event
GSE	Generic Substation Event
GSM	Global System for Mobile Communications
HAL	Hardware Abstraction Layer
HMI	Human-Machine Interface
HV	High Voltage
Hz	Hertz
I2C	Inter-Integrated Circuit
IC	Integrated Circuit
ICD	IED Capability Descriptor
IDE	Integrated Development Environment
IEC	International Electrotechnical Committee
IED	Intelligent Electronic Device
IEEE	Institute of Electrical and Electronics Engineers
IGBT	Insulated Gate Bipolar Transistor
IID	Instantiated IED Description
IoT	Internet of Things
ISO	International Organization for Standardisation
LAN	Local Area Network
LLN0	Logical Node Zero
LN	Logical Node
LV	Low Voltage
MCU	Micro-Controller Unit

MMS	Manufacturing Message Specification
MV	Medium Voltage
NRS	National Regulatory Services
OPNET	Open Networks Simulator
PC	Personal Computer
PCB	Printed Circuit Board
PCC	Point of Common Coupling
PDC	Phasor Data Concentrator
PDU	Protocol Data Unit
Pf	Power Factor
PhV	Phase Voltage
PLC	Power Line Communication
PMU	Phasor Measurement Unit
PQ	Power Quality
PQDIF	Power Quality Data Interchange Format
PQM	Power Quality Metering
PU	Per Unit
PV	Photovoltaic
RAM	Random Access Memory
RMS	Root Mean Squared
RTAC	Real-Time Automation Controller
RTDS	Real-Time Digital Simulator
SAS	Substation Automation System
SCADA	Supervisory Control and Data Acquisition
SCD	Substation Configuration Descriptor
SCL	Substation Configuration Language
SEL	Schweitzer Engineering Laboratories
SMS	Short Message Service
SoC	System-on-Chip
SPI	Serial Peripheral Interface
SSD	System Specification Description
SSH	Secure Shell

SV	Sampled Values
TAL	Time Allowed to Live
TC	Technical Committee
TCP/IP	Transport Control Protocol Internet Protocol
THD	Total Harmonic Distortion
UART	Universal Asynchronous Receiver/Transmitter
UCA	Utility Communications Architecture
UML	Unified Modelling Language
URL	Uniformed Resource Locator
USART	Universal Synchronous Asynchronous Receiver/Transmitter
V	Volts
VNC	Virtual Network Computing
VSFTPD	Very Secure File Transfer Protocol Daemon
VT	Voltage Transformer
W	Watt
WAN	Wide Area Network
XML	eXtensible Mark-up Language

CHAPTER ONE

INTRODUCTION

1.1. Introduction

Traditionally, the electric grid has always been centralised. The generation phase took place at large power plants, while the transmission phase took place over high voltage lines managed by large utilities. The distribution phase is traditionally handled by local municipalities or distribution network operators. The resulting topology of the classic electric power grid is a "supplier and consumer" system, but this is increasingly no longer the case, with the advent of the "smart grid" and distributed energy resources (Alonso et al., 2020).

Figure 1-1 illustrates the difference between the traditional grid and the "smart grid". The power flow in the "smart grid" (to the right-hand side of Figure 1-1) does not originate from a single source in a distributed environment as it does in a conventional grid, thus significantly increasing the points of supply to be monitored. The web-like system topology formed by a combination of consumers and distributed generators increases the overall complexity of the distribution network as opposed to the traditional power grid. Consumers and generators being nearby and occasionally reversing the power flow by becoming producers as in the case of microgrids, have the potential to negatively impact the power quality if not monitored sufficiently (Dong et al., 2012). The presence of generation plants operating at varying voltage levels also presents a challenge as it necessitates the usage of harmonic-injecting power electronic converters, further deteriorating the power quality of the system (Bracale et al., 2011).

In the traditional power grid (left-hand side of Figure 1-1), utilities and distribution network operators are responsible for maintaining a steady supply of electricity as well as the power quality of said electricity being delivered to consumers. Utilities and consumers agree to a certain level of quality according to the grid code, standards set out in National Regulatory Service (NRS 048 - in a South African context) and the sensitivity of the on-site equipment. In the SANS 10142 standard Part 1-2 (Specific requirements for embedded generation installations connected to the low voltage distribution network in South Africa) it is recommended that the Embedded Generation Installation (EGI) designer performs necessary power quality measurements (SANS, 2002).

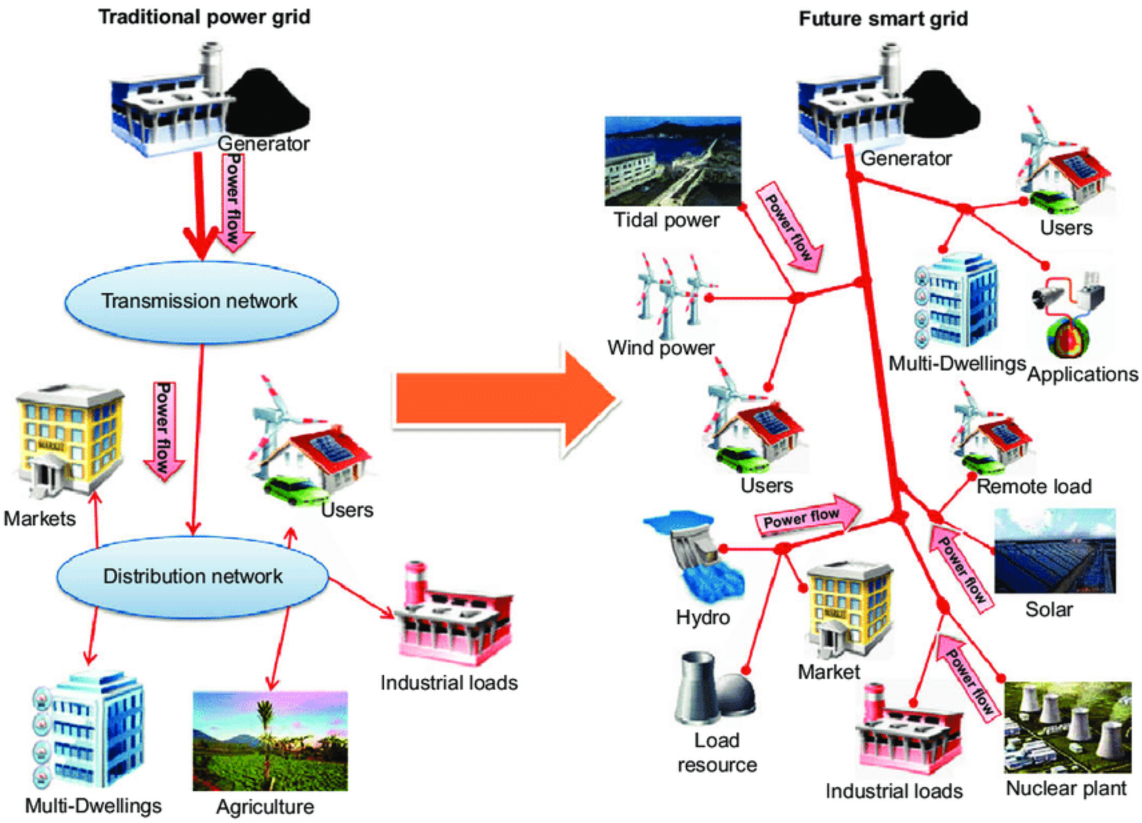


Figure 1-1: Traditional Power Grid vs Smart Grid (Dong et al., 2012)

The increasing number of PV installations as distributed energy resources has started to alter the traditional grid topology, with consumers becoming producers. What this has done is transform not only the traditional grid but also create a quality of supply monitoring problem: How does one monitor and control thousands of grid-connected producers located over a large area and generate electricity that is inherently unpredictable and inconsistent? This scenario creates a far more complex situation that could compromise the power quality in certain parts of the grid. It makes it difficult for utilities to maintain a good quality of supply.

1.2. Awareness of the problem

The majority of large South African municipalities now permit the installation of Small-scale Embedded Generation (SSEG) plants, which allow them to operate as generators instead of ESKOM (South Africa’s power utility and energy producer) resellers. According to Filipova and Morris (2018), SSEG plants have significantly become more desirable with consumers due to the ever-increasing ESKOM tariffs, and with local municipalities now offering feed-in tariffs (FITs) to plant owners, which further incentivises investment into SSEGs. Municipalities such as The City of Cape Town have set guidelines and requirements for new SSEGs due to the high demand from residential and commercial customers (City of Cape Town, 2017).

The scenario mentioned above has grown the installed capacity of photovoltaic-based SSEGs in South Africa's four largest metros (eKurhuleni, Johannesburg, eThekwini and Cape Town) to 38.46 MWp, with 29.08 MWp of the total comprising of commercial customers. Figure 1-2 visually illustrates the amount of installed capacity of photovoltaic SSEGs in all the major South African municipalities across three sectors. As can be seen in Figure 1-2, the adoption of SSEGs has been overwhelmingly prevalent in the commercial industry (as indicated by the orange bars). This increased adoption has been mainly due to the electricity generation and supply issues being experienced by ESKOM, South Africa's largest utility and primary energy supplier (Filipova & Morris, 2018).

Although 38.46 MWp is not a large installed capacity in the broader South African context, it is not insignificant. It presents an opportunity to develop frameworks, processes and technologies for managing and monitoring independent generation while the sector is in its infancy. National Energy Regulator of South Africa (NERSA) has the quality of supply standards stipulated in NRS 048-2, and these same standards would thus need to be applied to SSEGs. Any instrument developed to monitor the power quality of SSEGs should be designed with adherence to the stipulated standards.

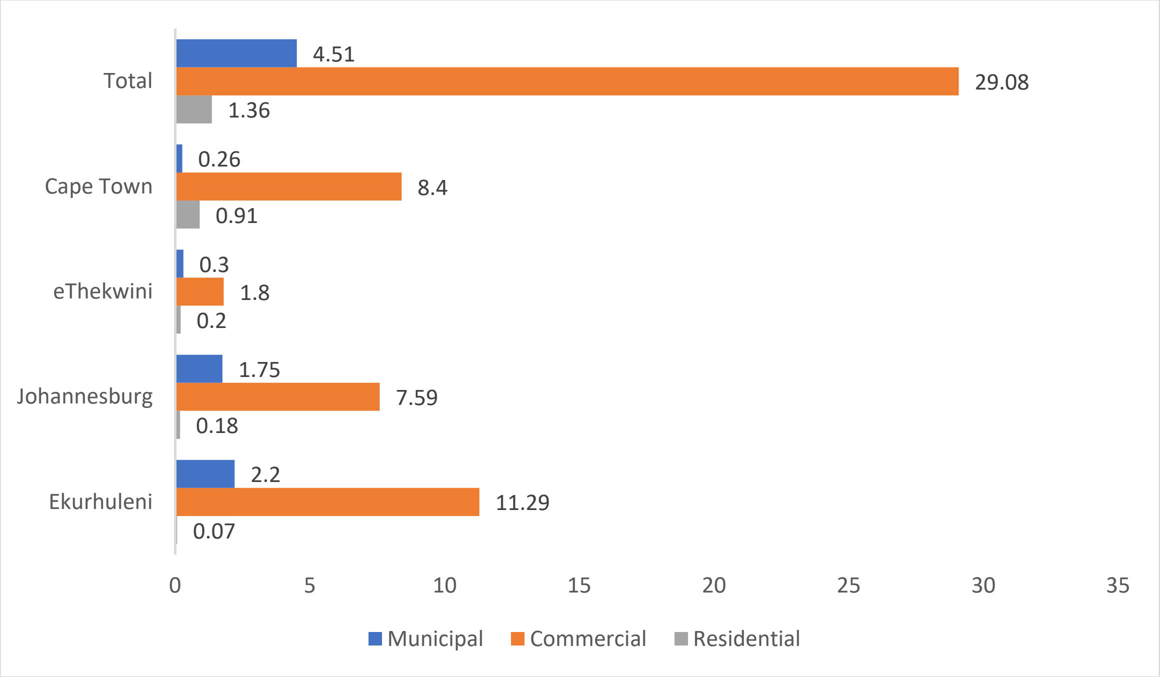


Figure 1-2: Legally Installed SSEGs by Municipality (Filipova & Morris, 2018)

1.3. Power Quality

In 1999, the council directive 85/374/EEC as amended by the European Parliament, classified electricity as a product (Council of the European Union, 1985). The classification of electricity as a product implies that it needs to adhere to specific quality control standards as with all other goods sold to the public. According to the IEEE 1159 standard (IEEE, 1995), power quality refers to a variety of electromagnetic phenomena that affect the voltage and current characteristics at a given time and location in a power system. A unique attribute about electricity is that unlike other products, it is often consumed immediately after generation; thus measurement and analysis of power quality must be done simultaneously to consumption (Moreno-Munoz, 2007).

1.3.1. Electromagnetic Interference

Electromagnetic Interference (EMI) is defined as an electromagnetic phenomenon that negatively affects the performance of devices, equipment or systems (Moreno-Munoz, 2007). In principle, every electromechanical device generates a certain level of electromagnetic interference. Two types of electromagnetic interference exist; radiated interference and conducted interference. Radiated interference is inductive and does not require physical contact between the source and the victim; radiated interference can manifest as common mode noise, differential mode noise and mixed-mode noise. Conducted interference requires a physical connection between the source and the victim. The switching frequency required for conducted interference is higher than that of radiated interference resulting in higher-order harmonics (Shadare et al., 2017).

In a Distributed Generation scheme, electromagnetic interference is mainly manifested from power electronics interfaces, particularly the power converter circuits used to interface renewable energy resources with the electric grid. Power converters that operate at high frequency are a verifiable source of EMI; this is very apparent in Insulated Gate Bipolar Transistor (IGBT) based conversion circuits which can emit 9 kHz to 30 MHz worth of electromagnetic radiation resulting from the switching of the devices. With the growth of renewables and distributed energy, especially in the Low Voltage (LV) and Medium Voltage (MV) networks, EMI is going to increasingly become an issue (Shadare et al., 2017).

1.3.2. Electromagnetic phenomena classification

As mentioned in the previous section, power quality issues can manifest in several ways as electromagnetic interference. Power quality disturbances can be grouped according to which characteristic of the voltage wave is being affected by the disturbance, or by the type of electromagnetic interference (radiated or conducted) the system is experiencing. According to the IEEE 1159 (1995) standard, the International Electrotechnical Committee (IEC) classifies electromagnetic disturbances by the types of electromagnetic disturbances, which can then be subdivided according to the duration of the interruption and the characteristic being affected. As such, for this research, the IEC classification and terminology for power quality phenomena shall be used subsequently.

The IEC classification of various phenomena is shown in Table 1. Categories listed are conducted low and high frequency, radiated low and high frequency, electrostatic discharge and nuclear electromagnetic pulse. Within each classification are several electromagnetic disturbances, which can be further broken down to either steady-state phenomena or non-steady-state phenomena (IEEE, 1995).

Conducted phenomena can be described as disturbances that directly affect the “victim” signal; common types of these interruptions are voltage dips and swell as well as harmonics and interharmonics. Typically, the causes of such interruptions are certainly more evident as compared to radiated phenomena. It could be the result of faulty or incompatible devices connected onto the temporary network loss of supply, or even high/low voltage switching action causing a dip or swell. In contrast, radiated phenomena often originate from outside the system. They can be caused by magnetic fields and electric fields created by high or low-frequency telecommunications equipment and appliances such as microwaves.

Electrostatic discharge phenomena occur when two electrically charged object come into contact with one another; a common form of electrostatic discharge occurs between the human body and electronic devices. The human body has the potential to generate an electrical charge of several thousand volts which can enter the electronic device, resulting in malfunction or damage of integrated circuits. Currently, there are not electrostatic discharge phenomena classified under the IEC standard, which is also the case with nuclear electromagnetic pulses (EMP). An EMP is a high velocity, an invisible burst of electromagnetic energy which has the capability of destroying electronics. A nuclear EMP is a more extreme form of an EMP (which can occur naturally due to lightning strikes) caused by a nuclear explosion which can cover a much larger area with a devastating effect.

Table 1-1: Principal phenomena causing electromagnetic disturbances as classified by the IEC (IEEE, 1995).

Conducted low-frequency phenomena	**Harmonics, inter-harmonics**
	Signal systems (power line carrier)
	Voltage fluctuations
	Voltage dip and interruptions
	Voltage imbalance
	Power-frequency variations
	Induced low-frequency voltages
	DC in ac networks
Radiated low-frequency phenomena	**Magnetic fields**
	Electric fields
Conducted high-frequency phenomena	**Induced continuous-wave voltages or currents**
	Unidirectional transients
	Oscillatory transients
Radiated high-frequency phenomena	**Magnetic fields**
	Electric fields
	Electromagnetic fields
	Continuous waves
	Transients
Electrostatic discharge phenomena	—
Nuclear electromagnetic pulse	—

Appendix A describes the duration and magnitude of power quality interruptions. Power quality interruptions can be categorised based on the spectral content, duration and voltage magnitude. Interestingly, sags, swells and interruptions are sub-categorised into three different groups (instantaneous, temporary, momentary) based solely on their duration and voltage magnitude, describing any of the three phenomena above requires clarity in this regard.

1.4. Analysis of Quality of Supply issues

The continued introduction of renewable energy, as well as other more conventional forms of distributed generation, has increased the necessity of power electronics for conversion and interfacing with the power grid. This increase has resulted in the manifestation of various power quality issues that have arisen as a result of varying consumer loads and component operating characteristics.

Power electronics inverters tend to inject harmonics into a power system. However, they do not propagate throughout the system due to their high order (which is discussed in the next section), which creates additional distortion that was not present (Hariri & Faruque, 2014).

Another significant issue is the voltage fluctuation due to the variability of distributed generators; this variability has the effect of causing excessive operation of voltage regulation devices in the network, and potential damage to customer equipment depending on the severity of the fluctuation.

Voltage forms the basis of most power quality studies; the magnitude is typically calculated using the Root Mean Square (RMS) voltage, the peak voltage or the fundamental voltage (Moreno-Munoz, 2007). Most commonly used in digital measurement systems is the RMS voltage, which can be obtained using Equation 1.1:

$$V_{rms} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} V_i^2} \tag{1.1}$$

Where:

i = the voltage sample number

V_i = the sampled voltage

N = the number of samples taken in a separate window.

1.4.1. Harmonics

Harmonics can be described as a mutated sinusoidal wave, containing the sums of several sine waves varying in magnitude, resulting in a non-sinusoidal waveform. Harmonics are manifested as a result of the non-linear characteristics of devices and loads. The frequencies that they typically operate at are integers multiples of the fundamental frequency of the power system (50 or 60 Hz). The waveform distortion is mainly produced when the actual voltage and current are combined (IEEE, 1995).

When measuring harmonic distortion, most instruments are configured to measure the Total Harmonic Distortion (THD). THD is the ratio of the RMS value of the sum of all the harmonic components up to a specified order; to the RMS value of the fundamental component. Equation 1.2 is used for determining the THD.

$$THD = \sqrt{\sum_{n=2}^{H} \left(\frac{G_n}{G_1}\right)^2} \tag{1.2}$$

Where:

G_n is the Harmonic component

n is the harmonic order: a running number

G_1 is the Fundamental component

H is the Harmonic order

The devices that are typically the culprits for causing harmonic distortion are treated as harmonic current injectors, the resulting voltage distortion stemming from the injected current causes voltage drops in the system (IEEE, 1995). Harmonics are becoming more of an issue as the culprits are typically power electronic converters. Figure 1-3 shows the waveform produced by the harmonic current injected into the system. The green waveform represents a system operating at a fundamental frequency of 50 Hz, whilst the blue signal illustrates a signal “polluting” the fundamental waveform. This signal operates at 150 Hz, meaning it is operating at the 3rd harmonic. Adding the fundamental waveform with the 3rd harmonic waveform creates a distorted sine wave such as the red one seen in Figure 1-3, known as harmonic distortion.

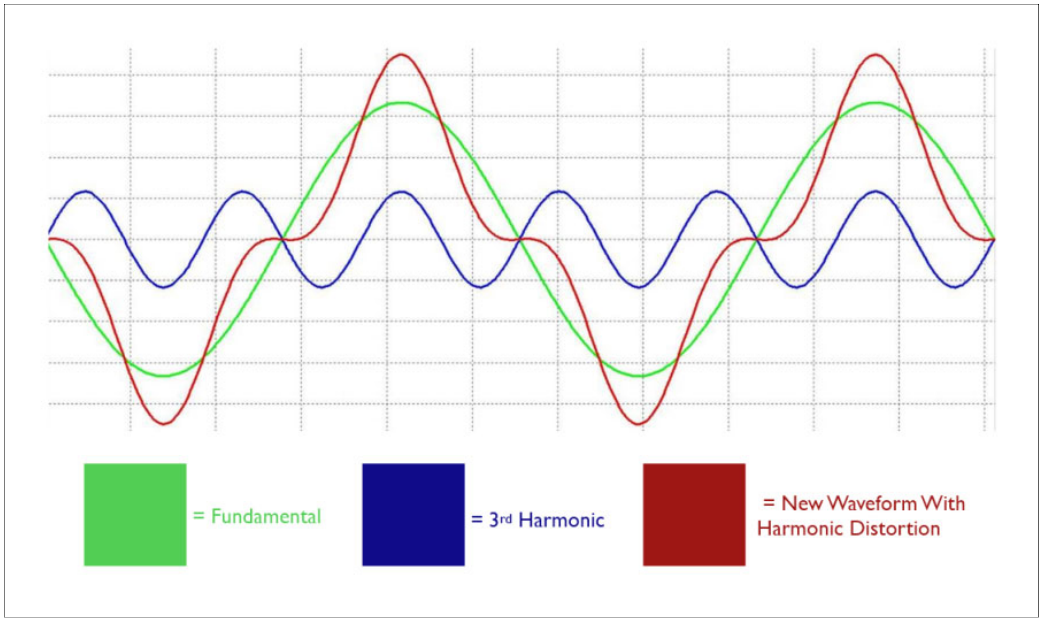


Figure 1-3: Formation of a Waveform with Harmonic Distortion (Inverter Drive Systems LTD, 2019)

1.4.2. Interharmonics

Interharmonics are discrete frequencies or wide-band spectrums that are present in all voltage levels. They are usually caused by induction motors, static frequency converters, cyclo-converters, power line carriers and arcing devices. Interharmonics may cause flickering in visual devices such as CRT based computer monitors or televisions.

1.4.3. Noise

Noise is the result of unwanted electrical signals of frequencies below 200 kHz manifested in the voltage or current signal of neutral or phase conductors or signal lines. The typical cause of the noise is power electronic devices, solid-state rectifiers, arcing equipment and control circuits. Improper grounding can make the problem worse. Noise typically has a magnitude of less than 1% of the voltage magnitude and is especially harmful to Programmable Logic Controllers (PLCs) and computer equipment (IEEE, 1995). Figure 1-6 illustrates the noise phenomena experienced in the aforementioned power electronic, rectifier and control circuits. The noise continuously manifests all along with the sine wave as small peaks and troughs. It can be noted that Figure 1-6 represents a specific noise phenomenon called “notching”.

1.4.4. Frequency variation

The frequency of a given network is dependent on the load and the generation capacity. The amount of load and the system frequency are inversely proportional, as the system load increases the system frequency falls if there is not enough generation capacity. Frequency variation is a result of the changes in this dynamic balance changing. The severity of the frequency changes and the duration is dependent on the types and behaviour of the loads and how generators respond to load changes (IEEE, 1995).

Faults on the transmission system, massive outages and loss of generation can cause frequency variations that fall outside the acceptable limits determined by the grid operator. Figure 1-4 shows a signal with varying amplitudes, indicating voltage variation in the signal. The waveform also exhibits distortion which affects the sinusoidal nature of the waveform, signals such as the one shown in Figure 1-4 could prove detrimental to sensitive electrical equipment which requires a higher quality voltage signal.

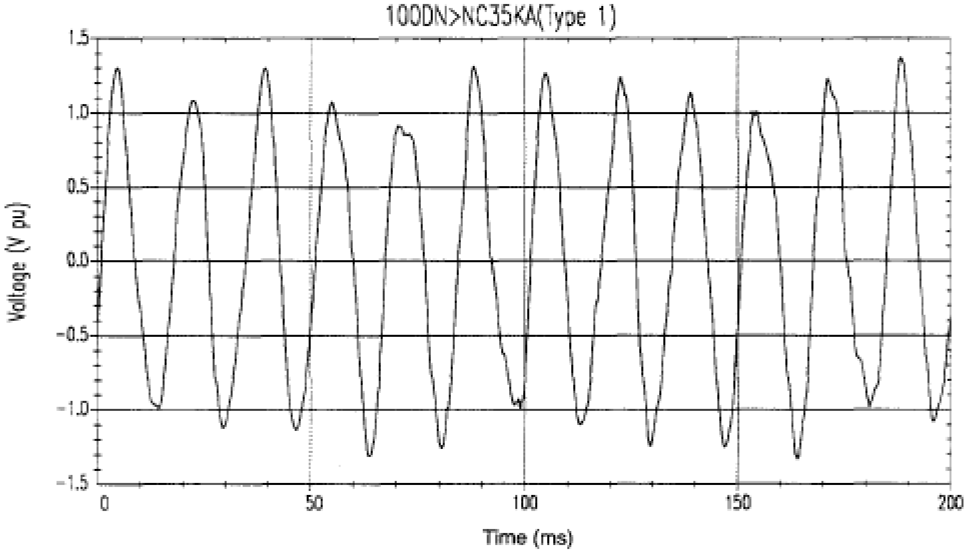


Figure 1-4: Frequency Variation (IEEE, 1995)

1.4.5. Interruption

Interruptions occur when the load current or supply voltage drops to a value of less than 0.1 pu for less than 1 min. The duration measures severity of an interruption; this is because the magnitude of an interrupted voltage signal is always below 10% of the nominal voltage. The duration is very dependent on the reaction times of the protective devices on the network and the root cause of the event. Interruptions are typically the result of system faults, control malfunctions or equipment failure.

On occasions where interruptions are caused by faults originating in the source system, dips may precede the interruption. In these situations, a dip would occur within the time the fault starts, and the system's protection operates; the feeder where the fault originated would experience an interruption straight after the sag. Figure 1-5 illustrates an example of an interruption caused by a reclosure operation. An auto recloser is a protective circuit breaker with the ability to reclose the breaker automatically after tripping due to a fault current. Auto reclosers are typically found on overhead lines where a falling branch or errant bird might cause a temporary fault current that requires protective action but does not require external intervention for restoration. The auto-reclose function can typically take 200ms to 2s to operate; well within the time frame for a temporary interruption. This example clearly illustrates how switching action can result in various power quality events.

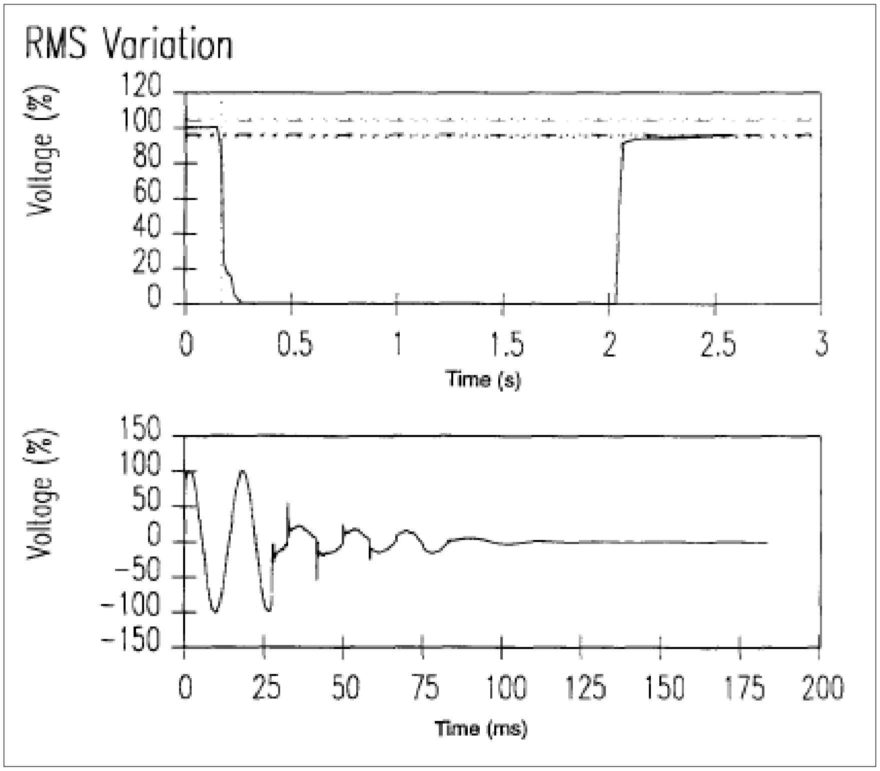


Figure 1-5: A momentary interruption due to a fault with a reclosure operation (IEEE, 1995)

1.4.6. Notching

The regular operation of power electronic devices can cause a periodic disturbance of voltage called "notching"; the commutation of current causes this from one phase to another. Notching can be considered both a transient interruption and a form of harmonic distortion; this is because it is a steady-state event that occurs periodically and can this be seen through the harmonic spectrum.

Notching is most commonly caused by three-phase power electronic converters producing constant DC, where the commutation from one phase to another causes a temporary short circuit. This phenomenon can be observed in Figure 1-6 as described earlier in 1.4.3, and as such, notching can be considered a form of signal noise.

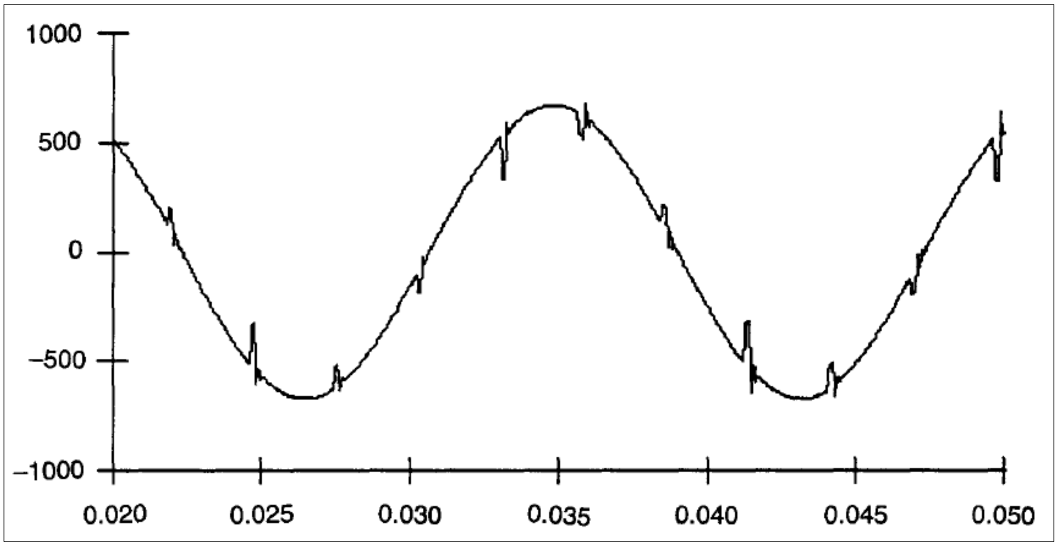


Figure 1-6: Voltage Notching Caused by a Power Electronic Converter (IEEE, 1995)

1.4.7. Dips

A voltage dip is characterised by the line voltage dropping to 20% of the nominal value; this should not be confused with the voltage dropping by 20%. When measuring a voltage dip, a typical disturbance recorder reports the remaining voltage instead of the severity of the dip. System faults are the leading cause of a dip, but large motors starting and switching on a heavily loaded network could also cause a dip. For example, induction motors draw approximately 6 to 10 times its load current at startup. Voltage dips are also commonly referred to as “sags” according to the IEEE 1159 definition.

Voltage dips typically last for between 3 to 30 cycles, but this is dependent on the magnitude of the fault current and the type of algorithm used for overcurrent detection and other interruptions. Dips can be subdivided into three categories - instantaneous, momentary and temporary; each with their duration and voltage magnitude ranges (See Appendix A). Figure 1-7 graphs a voltage dip caused by a line-to-ground fault on a single-phase where the fault lasts for approximately 75ms and spans three cycles. Typically, the clearing time for a fault can range between 3 to 30 cycles depending on the fault current magnitude and the fault detection algorithm, which is well within the range of a voltage dip.

Dips may also be the result of a fault on a parallel feeder which causes a voltage drop on other feeders until the fault is cleared (IEEE, 1995).

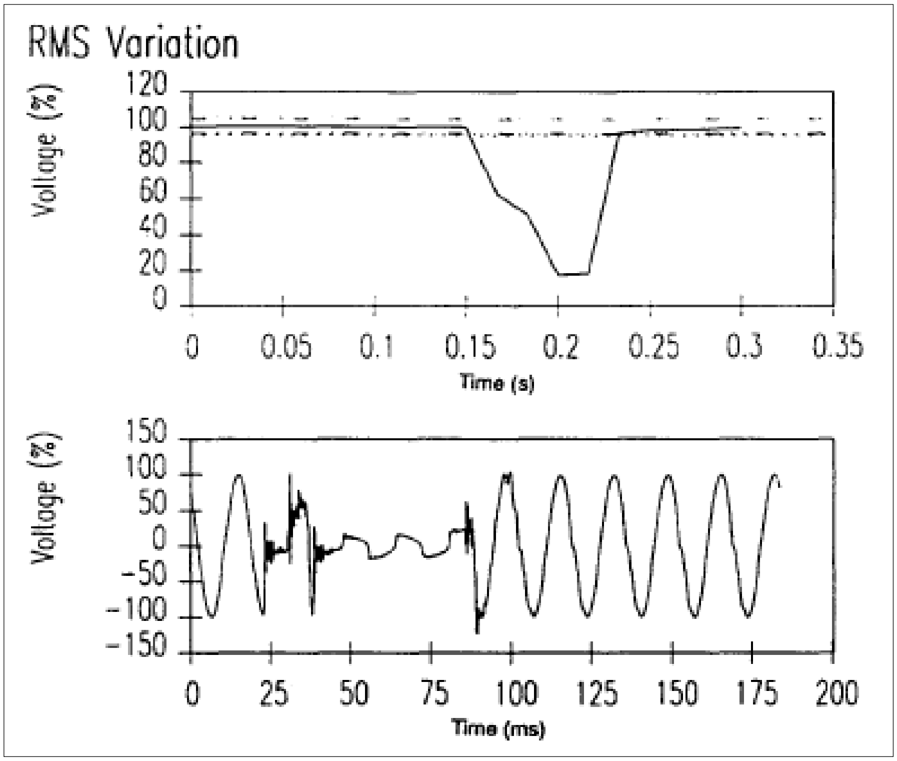


Figure 1-7: Voltage Dip on a single phase (IEEE, 1995)

1.4.8. Swell

Swells can occur on both voltage and current; they can be characterised by an increase in the respective quantities' magnitudes at the power system frequency for durations between 0.5 cycles to 1 minute. The magnitudes of these swells can range between 1.1 and 1.8 PU. Like dips, swells can be characterised magnitude and duration of the fault, as shown in Appendix A (2.1 to 2.3). Momentary overvoltage is another commonly used term for voltage swells.

Similar to dips, swells are typically a result of system faults, although much less common in their occurrence. The swell usually manifests when a fault occurs on a particular phase, and the temporary rise in voltage is experienced on the phase without the fault present. They can also occur due to switching off large loads or capacitor banks. Figure 1-8 presents an example of a swell occurring on a single-phase caused by a line-to-ground fault. In this instance, the fault resulted in a transient voltage increase on the other phases for just under 100ms, well within the definition of a voltage swell. In the sine wave, it can be noted that the swell seems insignificant in its magnitude and duration. Still, even this tiny spike in voltage could damage electrical equipment without the correct tolerances.

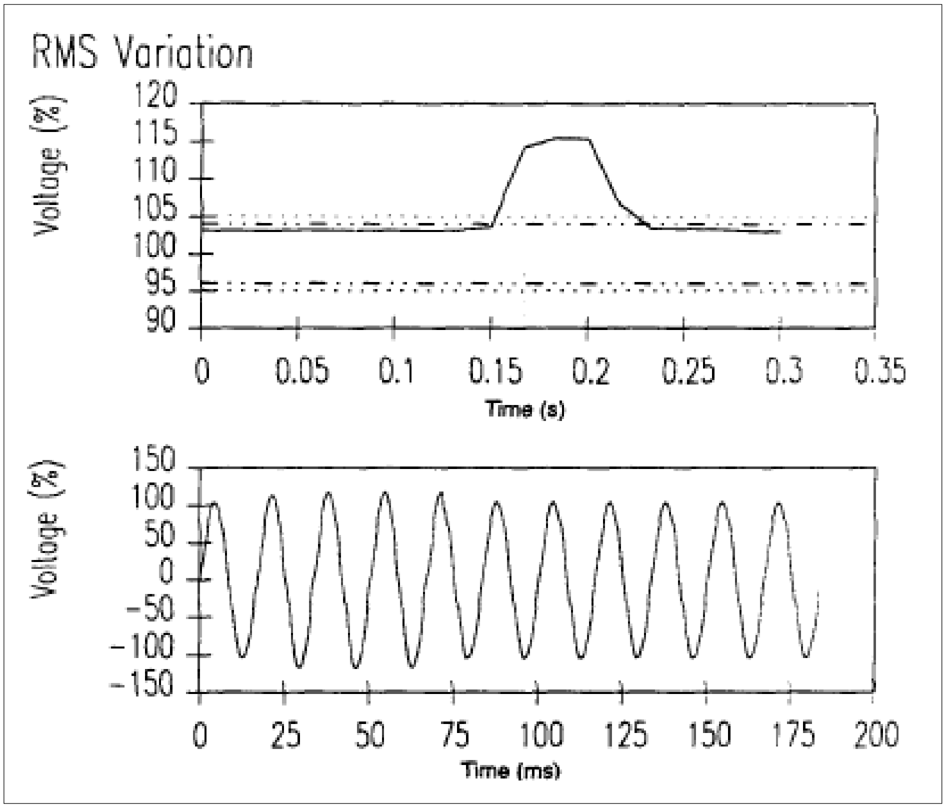


Figure 1-8: Voltage Swell on a Single Phase (IEEE, 1995)

1.4.9. Voltage unbalance

Voltage unbalance (also referred to as imbalance) is the ratio of the negative or zero sequence component to the positive sequence component (Putrus et al., 2007). Unbalanced loads are the main cause of the negative or zero sequence voltages. Unbalances are usually estimated by determining the maximum deviation from the average voltage or current in a three-phase system, then dividing that by the averages of voltage and current. The equation is as follows:

voltage imbalance = 100 x (max deviation from average voltage)/average voltage

Figure 1-9 illustrates an unbalanced three-phase system with varying voltage magnitudes over a specific period. Example: If the phase voltages are 225 V for red, 230 V for blue and 232V for red, the average is 229 V. With the average determined, the maximum deviation can be calculated by finding the difference between 229 V and each of the three-phase voltage readings. In this case, the maximum deviation is 4 -> (229 – 225). Finally, the imbalance in percentage can be calculated as 100 x 4/229 = 1.7%.

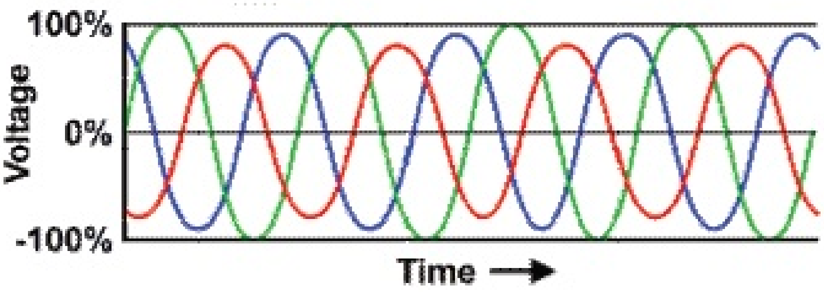


Figure 1-9: Unbalanced Three-Phase Circuit (Power Quality World, 2011)

1.4.10. Undervoltage

A system can be in an Undervoltage state due to the following conditions:

- A large load being switched on without proper compensation.
- Compromised reactive compensation. e.g. capacitor banks switched being off.
- Tap change malfunction.
- An overloaded circuit.

At times, a system operator may decide to operate the system in an Undervoltage state to reduce power delivery; this is sometimes called a brownout.

1.4.11. Overvoltage

Over-voltage may be considered the inverse of under-voltage, where the overvoltage conditions are typically the result of the following:

- Load switching. e.g. a large load being switched off without proper compensation.
- Compromised reactive compensation. e.g. capacitor banks switched on.
- Tap change malfunction.
- Generally insufficient system voltage regulation.

This section presented power quality issues encountered in a typical power system. The next section presents the Problem statement for this research work.

1.5. Problem statement

The proliferation of DERs and renewable energy as a whole has seen the increased adoption of power electronic converters in medium voltage (MV) and low voltage (LV) networks. These power electronic converters can cause power quality issues to manifest in the form of harmonics, notches, voltage fluctuations, swells, dips and temporary over/under-voltage conditions. The monitoring of the power quality indices in a distributed environment requires advanced ICT and smart grid technologies as well as cost-effectiveness due to the scale of deployment.

Problem Statement: *To investigate the development of an IEC 61850-based power quality monitoring device. The cost to performance ratio has to be considered to aid possible deployment into a more extensive network*

1.6. Sub-Problems

The following sub-problems that require consideration have been identified.

1.6.1. Investigate the appropriate system topology

Several system topologies for monitoring power quality are possible. Examples: A monitoring system may make use of disturbance recorders located on consumer premises communicating remotely via GSM or using the IEC 61850 Ethernet-based communications of IEDs situated in the utility's point of common coupling (PCC)

1.6.2. Determine evaluation criteria

The criteria for evaluating the performance of a power quality monitoring device needs to be identified. Accuracy, reliability, availability, interoperability, footprint and cost could all be determining factors for the implementation.

1.6.3. Determination of monitored indices

Prioritisation of which indices need to be monitored depends on the application, device, voltage level and the types of loads being measured. This project primarily focuses on DERs, which make use of power electronic converters, so any implementation has to focus on indices appropriate for power electronics.

1.6.4. Design and implementation of a hardware platform

The hardware architecture affects the capabilities of the monitoring device. Based on the application and critical nature of the equipment being monitored, a complex, more rugged and less cost-effective specialised hardware solution is preferable. A more straightforward but more cost-effective solution may be preferable in certain circumstances. i.e. in situations where the device needs to be deployed at multiple supply points.

1.6.5. Implementation of IEC 61850 functions

IEC 61850 logical nodes for power quality monitoring exist, but not all are readily available on commercially available power quality monitoring IEDs. Implementing new logical nodes may be necessary on devices missing certain required functions.

1.6.6. Power Quality Monitoring assessment

The performance of the implemented power quality monitoring system needs to be assessed according to the predefined criteria. For this purpose, a test bench must be implemented, taking into account all of the possible variables to get accurate results.

1.6.7. Testing of IEC 61850 communications

Tests may need to be performed on the device to prove interoperability and possible compliance with MMS and GOOSE communications.

This section presented all the sub-problems that are identified in attempting to provide a solution to the stated problem. The following section presents the Research aim and objectives identified for this research work.

1.7. Research aim and objectives

1.7.1. Aim:

This project aims to design and implement an IEC 61850-based power quality and DER monitoring system by developing new logical nodes or implementing existing ones on a selected hardware platform. This new monitoring system is aimed at providing an interoperable, cost-effective and scalable device to increase the deployment of power quality monitoring devices in a distributed environment to make the monitoring of DER supply points easier.

1.7.2. Objectives:

Maintaining a good quality of supply shall become more complicated in power systems with multiple points of supply, power electronic converters and bidirectional DER’s. With this in mind, power quality monitoring needs to become more widespread and smarter. Deployment of IEC 61850-compliant devices at the DER’s supply points allows for more robust monitoring by enabling them to interoperate with existing IEC 61850-based substation automation devices. To achieve the stated research project aim, the following objectives have been set out:

- To perform a thorough literature review of existing papers for power quality monitoring systems to determine the best practice, procedures and technologies for measurement and data transmission.
- To define the indices, algorithms and logical nodes necessary for monitoring the power quality of DERs at points of supply within the electric power grid.
- To select the appropriate hardware platform for the implementation of the software algorithms and the development of the IEC 61850 logical nodes.
- To implement and calibrate the power quality detection algorithms on the selected hardware platform/device.
- To implement IEC 61850 logical nodes on the selected hardware platform of choice using an open-source protocol stack.
- To establish, transmit and receive IEC 61850-based communications from devices in a substation automation environment.
- To test and evaluate the implemented logical nodes and power quality detection algorithms using test and simulation software and hardware components
- To prove interoperability by configuring the device to communicate with an industrial-grade IEC 61850 compliant IED or automation controller.

1.8. Hypothesis

It is possible to develop an IEC 61850-based power quality monitoring device for use in a distributed environment using open-source hardware, software and development of new software algorithms for data acquisition and transmission. The continued evolution of wired/wireless communications, coupled with the existence of open-source IEC 61850 libraries, and the availability of inexpensive single-board computers makes it feasible to implement a practical and cost-effective power quality monitoring device.

1.8.1. Assumptions

The following assumptions are considered during the conception, design and implementation phase, and are based on the current state of research in this field.

I. The chosen hardware is fully compliant with the power quality monitoring standards and thus only require calibration.
II. The proposed device to be developed is primarily used for monitoring low voltage DERs and not large installations.
III. The proposed monitoring device is placed optimally in the network, as close to the power electronic converter as possible.
IV. Omicron Test Universe contains all of the necessary power quality injection signals.

1.9. Delimitation of research

The focus of this research work is solely on the design, implementation and evaluation of an IEC 61850-based power quality monitoring device. Previous methods and existing alternative methods are taken into account as these could prove vital in the final implementation of the system. The research document serves as a proof of concept for the development of a flexible, configurable and cost-effective device to monitor a large number of distributed generators.

1.9.1. Within the scope

The following limitations are considered within the scope of this research:

- Perform a theoretical investigation on power quality issues commonly affecting electric power grids, with an emphasis on distributed energy resources (DER).
- Design and implementation of a novel power quality monitoring device using IEC 61850-based information and device modelling.
- The measurements modelled and implemented on the device are limited to the crucial power quality measurements as identified in the theoretical investigation.
- The device should be low-cost and configurable with preferably open-source hardware and software used for implementation.
- Validation of measurements against a calibrated reference meter in a controlled lab environment using a voltage and current injection test set.
- Confirmation of interoperability with industrial-grade IEC 61850-compliant IEDs and substation controllers.

1.9.2. Out of scope

The following is considered out of scope within this research:

- Implementation of measurement functions that are determined to be non-essential is considered out of the scope of this research.

1.10. Research design and methodology

In this section, a brief overview of the design process, the methodology, considerations and system architecture are presented.

First, the evolution and fundamental components of power quality monitoring techniques are reviewed. Then, an analysis of the crucial power quality issues affecting is carried out to determine the algorithms and metrics to be implemented on the device. Since the IEC 61850 standard is to form the basis of the data exchange and communications system, the modelling thereof on the device is carried out. Next, the implementation of data acquisition hardware, software algorithms and IEC 61850 protocol stack are carried out. Finally, the detection and data acquisition hardware are tested using a signal injection test set to validate power quality measurements, and the IEC 61850-based communications protocols are tested with the aid of an industrial-grade IED or substation controller. To achieve the stated aims and objectives, the following shall be done:

1.10.1. Literature review

The different methods of power quality monitoring are researched to provide a historical context and document the currently most popular methods utilised for power quality monitoring. The Literature review is not limited to IEC 61850 methods but casts a net over the entire research field of power quality monitoring

1.10.2. Identification of power quality indices

Common power quality issues in distributed energy resources and power electronic converters are assessed and based on the assessment, and the most critical indices are chosen for implementation.

1.10.3. Power Quality Monitoring in IEC 61850

The development of power quality monitoring is charted to determine the best industry practice for monitoring using IEC 61850. The following aspects are explored:

- Suggested system topology in a substation automation environment
- The appropriate communications mapping for data transmission of indices. i.e. GOOSE, MMS or Sampled Values
- The availability of logical nodes for the chosen indices.
- Communications channel requirements, including but not limited to availability, speed and distance.

1.10.4. Finalisation of system architecture

All of the required software and hardware components are combined to create a unified system architecture which contains the following subsystems and stages:

I. Data acquisition system
II. Disturbance detection algorithm
III. Computation and processing
IV. Mapping of data to IEC 61850 communication protocols
V. Transmission of data to a substation controller/IED/SCADA System
VI. Remote device/system, successfully processing received data. Confirming interoperability

1.10.5. Evaluation and selection of hardware

The hardware components are selected based on the needs and objectives of the project. The hardware needs to be able to compute the indices chosen with accuracies compliant with the IEC 61000-4-30 standard. Cost and ease of use are factors considered in the selection process as well.

1.10.6. IEC 61850 device modelling

An IEC 61850 Information Model is developed, based on the aforementioned measurement values, indices and hardware selection. The information model is then used to create a device model, mapping the IEC 61850 functions to the device's data acquisition algorithms.

1.10.7. Firmware and software development

Software development is highly dependent on the choice of hardware. Suppose a specialised piece of hardware such as a metering IED or bay controller is selected. In that case, software development may be minimal and mainly comprise of the configuration of detection and transmission procedures.

A development board may require a more involved and complex suite of software (such as a third party IEC 61850 protocol stack) to be conceived and engineered to accurately recreate the functionality of a vendor built IED.

1.10.8. Test bench and evaluation

The test bench requires the evaluation of every aspect of power quality monitoring. The performance of algorithms, adherence to monitoring standards, instrument sensitivity and the performance of the IEC 61850 functionality are taken into account, to acquire this data, the following is required:

A. Power Quality (PQ) Signal Injection Device
B. Calibrated Power Quality Meter
C. IEC 61850 compliant companion IED or substation controller

1.11. Thesis Chapters

This research document is divided into 9 Chapters, followed by an Appendix. The thesis is comprised of the following chapters:

1.11.1. Chapter 1

This chapter introduces the research project and provides a background of power quality issues within the modern electric power grid with an emphasis on SSEGs/DERs. The problem awareness, problem statement, aims and objective, hypothesis, delimitation of research, motivation, assumptions and the design and methodology of the research project.

1.11.2. Chapter 2

A literature review on power quality monitoring is covered in this chapter, with particular emphasis on telecommunications and DSP-based systems. Many relevant papers are compared and analysed.

1.11.3. Chapter 3

The theoretical aspects of power quality issues affecting DERs are very briefly discussed in this chapter.

1.11.4. Chapter 4

This chapter explores the device modelling of an IEC 61850-based system with an emphasis on the information model and SCL file types. Then a device is modelled based on the functions of a chosen DSP-based analogue front-end device.

1.11.5. Chapter 5

In this chapter, an open-source IEC 61850 library implemented and tested on a chosen single-board computer hardware platform. An ICD configuration file is also generated based on the device model created in Chapter 4.

1.11.6. Chapter 6

A data acquisition system is implemented to interface the computing platform with the data acquisition analogue front-end device. A PCB is designed after component sizing and software is written for reading and writing data from and to the analogue-front end. The IEC 61850 MMS and GOOSE communications are also implemented and tested in this chapter.

1.11.7. Chapter 7

Evaluation of the implemented devices measurement accuracy is performed by first designing a testbed and injecting signals using the OMICRON CMC. The data is acquired and analysed against a reference meter wherever possible. MMS and ACSI service mapping is also tested by running selected IEC 61850-10 conformance test.

1.11.8. Chapter 8

This chapter confirms interoperability by creating a substation automation project, including the implemented device’s ICD file along with a vendor-built IED or automation controller. A SCADA demo is designed to demonstrate how the device would function in a common scenario.

1.11.9. Chapter 9

This final chapter summarises and discussed the critical findings of the research and proposes some ideas for future research.

1.12. Conclusion

This chapter provided an introduction to the research performed in this document. The aims and objectives, deliverables, research hypothesis and delimitation of research were presented. The problem statement and motivation for the research, the research design methodology and assumptions of the research were also presented.

This first section of this chapter provided an overview of the research project, the purpose and significance of the research and the possible questions that need answering to achieve the objectives related to power quality monitoring. The theoretical concepts of power quality were described to provide more context to the problem. The next section provides an in-depth review of the research and evolution of power quality monitoring. The literature is gathered and analysed better to understand the best practices for guidance and knowledge acquisition.

Chapter 2 presents a detailed literature review of work previously done in the field of power quality monitoring with communications technologies, data acquisition methods and system architectures being the primary focus on the review.

CHAPTER TWO

LITERATURE REVIEW

2.1. Introduction

This chapter reviews and analyses the existing research, advancements, and knowledge in the field of power quality monitoring. The common types of power quality issues and their causes, as well as the importance of power quality monitoring, have been discussed in Chapter 1. This chapter describes the available and most popular monitoring methods, equipment and standards such as EN510690 and IEC 61000-4-30. There is a specific focus on the implementation of power quality monitoring that is based on the IEC 61850 standard, with an emphasis on Edition 2 Logical Nodes, which are ultimately the foundation upon which this project is built. Each of the above topics is reviewed based on literature gathered, analysed and compared to provide a complete understanding of the current research in this field. The remarks and analysis present in the literature review aid and facilitate the research and conclusion thereof.

Section 2.2 provides an overview of the keywords and topics included in the literature search; section 2.3 briefly defines power quality monitoring standards and the essential components of a power quality monitoring system. In section 2.4, traditional methods of power quality monitoring are described. Section 2.5 provides a brief overview of Digital Signal Processing (DSP) and its application in power quality monitoring. The following section (2.6) expands on the application of DSPs by exploring the increasing adoption of smart meters and their role in power quality monitoring and general demand-side management. Then section 2.8 introduces the use of the IEC 61850 standard in power quality monitoring.

The literature review is performed in sections 2.9 to 2.11; each section deals with the specific sub-topics in power quality monitoring determined in section 2.2. Many papers are gathered and tabulated based on specific keywords and search terms, the findings of the papers are summarised, then analysed in comparative form. Section 2.12 provides a proposed system architecture based on the finding of the literature review, and finally, section 2.13 is the conclusion

2.2. Literature search

As stated in the introduction, the South African electric grid is starting to include an increasing number of Distributed Generation (DG) sources, both conventional and unconventional. This increased adoption creates a challenge in both monitoring and controlling power generation to prevent grid instability, as most modern electricity distribution systems are optimised to deliver power in one direction at higher voltage levels. Understanding the challenges that may be faced in the future and the best methods of solving or mitigating them, requires an establishment of context and understanding of the solutions available. Context and understanding may be gained by performing a literature search and subsequently, a review of the obtained literature with the objective of the research in mind. To provide a full picture of power quality monitoring, historical, current and future technological advancements have to be taken into consideration, with this in mind, three main focus areas that require further research and analysis have been identified:

- **Historical:** Traditional Power Quality Monitoring
- **Current:** IEC 61850-based Power Quality Monitoring
- **Future**: Smart Grid and Distributed Power Quality Monitoring

To effectively find relevant literature related to the topics mentioned above, several keywords and search terms are required for database and search engine input, these include:

- **Traditional Power Quality Monitoring**: Power Quality, Disturbance Recording, Power Quality Monitoring, DSP Power Quality
- **IEC 61850-based Power Quality Monitoring**: IEC 61850 Power Quality, 61850 Power Quality
- **Smart Grid and Distributed Power Quality Monitoring**: Smart Grid Power Quality, Distributed Power Quality

The literature search is primarily performed using the online Cape Peninsula University of Technology Library Database. To provide further context with regards to the content of the literature found, the following sections (2.3 to 2.8) describe the basics of a power quality monitoring systems as well as several power quality monitoring concepts, techniques and technologies. The abovementioned keywords and search terms aided in finding the resources containing the following definitions and concepts.

2.3. Power Quality Monitoring

Power quality standards have been developed to determine the levels at which power quality is acceptable or unacceptable. Standards such as IEC 61000 and EN 50160 were designed to detect when one of the phenomena described in the IEEE 1159 standard (refer to Table 1 of this document) was currently affecting the system. IEC 61000 is a family of standards that encompasses every aspect of Electromagnetic Compatibility (EMC). The standard includes IEC 61000-4-30, which deals explicitly with power quality measurement, and set outs performance characteristics that groups measurement equipment into one of three classes (Class A, Class S and Class B). EN 50160 can be considered a companion standard to the European Union directive mentioned in Chapter 1 that classifies electricity as a product. The EN 50160 acts as a product standard that quantifies the limits set out in IEEE 1159. These limits can then be used to calibrate measurement instruments to ensure compliance with EN 50160.

Figure 2-1 shows the basic architecture of a power quality meter. All meters follow the same basic system architecture. Typically, a power quality meter consists of the following components (Moreno-Munoz, 2007):

a) **Signal conditioning**: Manipulates the signal to prepare it for the next stage of processing
b) **Analogue-to-digital converter**: Converts and analogue signal (typically a sinusoidal wave) into a digital signal. Translating it for data processing purposes
c) **Digital signal processing**: a microprocessor mathematically manipulates digitised real-world signals such as voltage signals
d) **Data storage and management**: Data is stored as either plain text or in a database as raw data
e) **Postprocessing**: The data is further manipulated, typically to be transmitted via a specific communications protocol or scaled to a more readable format.
f) **Communications module**: Handles the communications procedures between the data acquisition device and external devices
g) **Control subsystem**: Depending on the power quality monitoring system, control may be required to make adjustments to the output signal using a control algorithm. Supervision can occur at any stage of the data acquisition process.

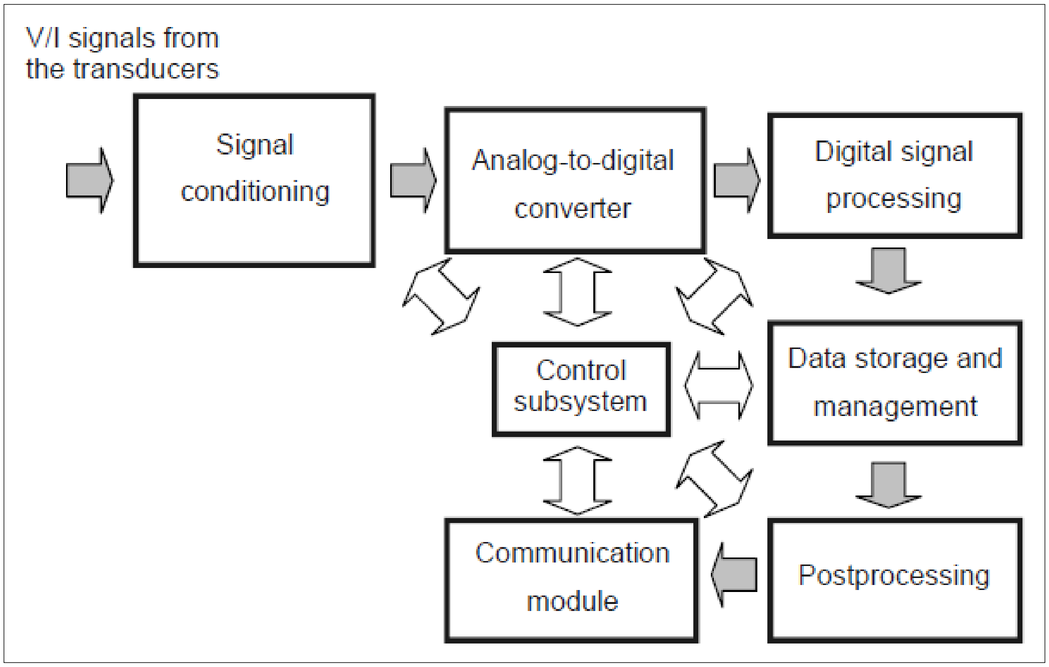

V/I signals from the transducers
Signal conditioning
Analog-to-digital converter
Digital signal processing
Control subsystem
Data storage and management
Communication module
Postprocessing

Figure 2-1: Block Diagram of a Power Quality Monitoring System (Moreno-Munoz, 2007)

Any instrument designed to measure power quality has to be compliant to a recognised standard depending on the region and application; for this research project, the standards are not reviewed in detail. One of the primary outcomes is to identify the most commonly used international standards in terms of accuracy and use it as a basis to prove the effectiveness of the implemented system. This requirement is a crucial aspect as it narrows the component selection to only the most accurate and robust devices.

2.4. Traditional measurement instruments

Power quality monitoring has evolved over the past decades, from general-purpose instruments to specialised monitoring equipment developed for the sole purpose of measuring transient events and interruptions based on the defined IEEE 1159 standard for electromagnetic disturbances. Traditional measurement instruments can include anything from general-purpose handheld multimeters to special purpose instruments such as spectrographs and power disturbance monitors. Clemmensen (1990), evaluated several measurement instruments and assessed the requirements and techniques for their usage. It was found that only the most expensive special-purpose instruments were suitable for measuring power quality indices specified in IEEE 1159, and in some instances had to be done with a combination of computer-based analysis.

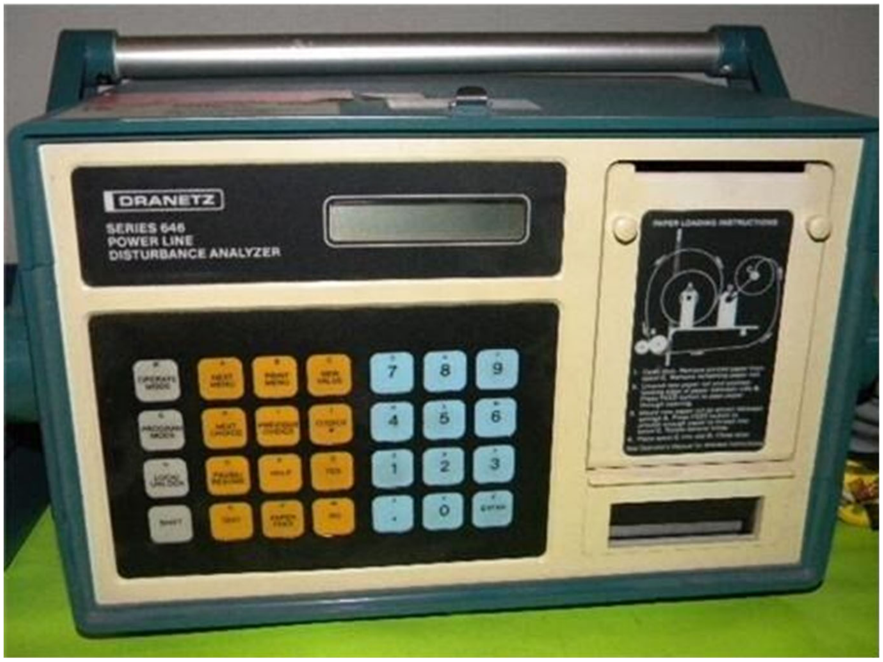


Figure 2-2: Dranetz 646-3 Disturbance Recorder (EBay, 2019)

Research into integrating traditional power quality disturbance monitors with SCADA functionality was done by Hairabedian (1989). His findings indicated the usefulness of combining telecommunications technologies with the Dranetz 646-3 power line disturbance monitor (shown in Figure 2-2). The system consisted of a number of these disturbance recorders connected to a central monitoring PC using telephone lines.

Figure 2-3 is a diagram of the system mentioned above, which represents an early example of a connected power quality monitoring system. Similarities between this and current systems can be observed, with devices such as the Dranetz 646-3 now being replaced by IEDs and smart meters, and copper telephone lines with fibre and wireless communications. In their time, although devices like the 646-3 were very functional and robust, the cost was prohibitive (with devices currently listed for over R10 000 on eBay as of 2020) and communications non-standard. The high initial costs limited the deployment of systems such as the one featured in Figure 2-3 to smaller, critical networks.

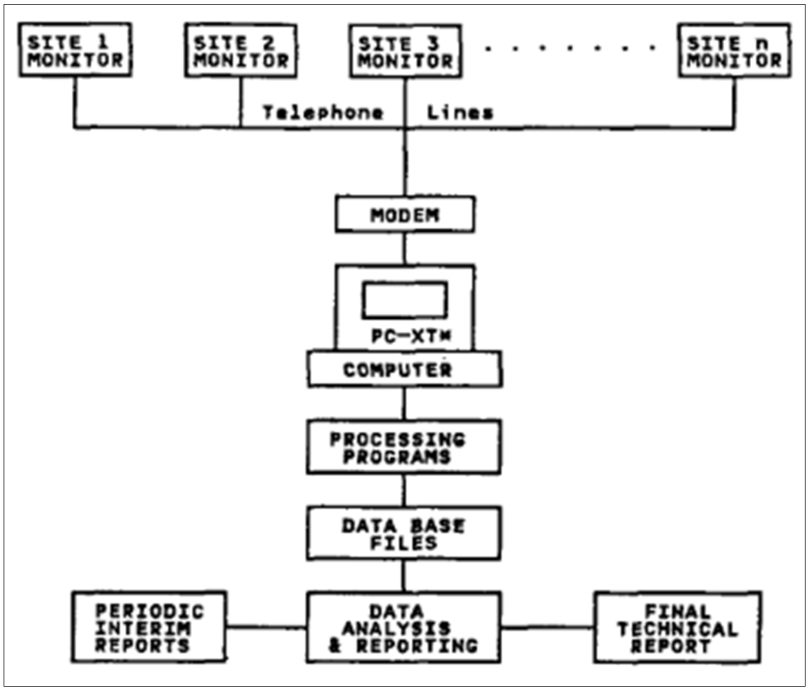


Figure 2-3: Monitoring System Using networked disturbance monitors (Hairabedian, 1989)

2.5. Digital Signal Processing and Embedded Systems

According to Analog Devices (2019), a digital signal processor (DSP) takes real-world signals like sound, video, temperature and electricity and digitises them so that they can be manipulated mathematically. The signals are converted from analogue to digital using an analogue to digital converter (ADC), then fed to the DSP, which captures the digitised data and processes them to be used in the real world.

Figure 2-4 shows the internal components of a typical DSP that contains a few of the standard elements of a regular embedded processor but are considered special-purpose devices. The compute engine handles all computational functions of the DSP and acts as the brains of the devices. Data is obtained or stored by the compute engine from the DSP's own program memory or an external storage device. The use of Serial Peripheral Interfaces (SPI), traditional serial ports, timers, host ports, external ports and link ports all form a part of the potential interfaces for relaying data to and from the outside world. DSPs are usually paired with a microcontroller or any other secondary computing device such as a PC; this is typically the case when it comes to DSP-based power quality monitoring.

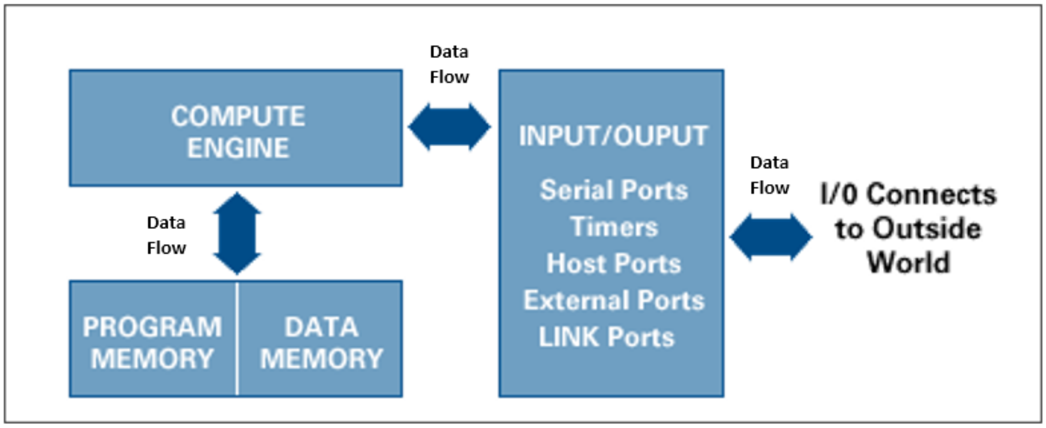


Figure 2-4: DSP Components (Analog Devices, 2019)

Ananth and Morcos (1999), investigated the use of DSP-based power quality monitoring in the electric power industry. It was found that the inherent advantage of using DSP-based instruments was the software-based approach. DSP’s, like other processor-based devices, require software to run the algorithms needed to calculate power quality indices. DSPs allow engineers to optimise and implement new, more efficient algorithms rapidly. Options to write code in a more efficient programming language such as Assembler, the introduction of interrupt-driven routines and the ever-increasing processing power due to improved silicon fabrication processes have allowed DSP’s to become very capable and accurate instruments for implementing power quality monitoring.

In their case study, Ananth and Morcos (1999) noted that the measurement error of a DSP based instrument was found to be less than the 5% maximum allowable error. In modern DSP-based smart meters, the accuracy has been proven to be consistently below 0.5% (Artale et al., 2018), which puts them comfortably within IEC 61000-4-30 compliance. With smart meters become ever more prevalent, the DSP is poised to continue playing a role in affordable yet effective energy and power quality monitoring.

2.6. Power Quality Monitoring Smart Meters

Smart meters have begun to play a more significant role in the monitoring and energy management of consumers. They are mainly used for revenue purposes, but there has been some suggestion of utilising them for network visibility purposes (Artale et al., 2018). As such, smart meters have become an integral part of the smart grid framework and form a part of the composition of measurement instruments installed at different voltage levels, such as power quality meters and phasor measurement units (Artale et al., 2018). As discussed in the introductory paragraph, the increase in the use of distributed generation and non-linear loads has necessitated the use of power quality measurement instruments at lower voltage levels.

Artale *et al.* (2018) suggest the possibility of implementing PQ metrics into a modern-day, commonly used smart metering platform. This possibility is especially pertinent as the embedded platforms that smart meters are typically based on, contain firmware that can be modified and extended to provide new functionality. The use of existing hardware with proven vendor developed DSP metrics implements standards-compliant PQ metrics in smart meter hardware a cost-effective and elegant solution. Advanced Metering Infrastructure also makes it possible to interconnect smart meters with SCADA control centres or substation automation systems for integration with protection functions. Figure 2-5 illustrates a typical advanced monitoring infrastructure system architecture. An advanced metering infrastructure (AMI) allows for bidirectional communication for data collection, energy management, load control and dynamic pricing. In brief, a complete AMI consists of the following subsystems:

Customer Relationship Management System (CRM): Manage the interaction between the utility and current customers by analysing historical customer data.

Billing: Smart meter data forms the basis of the billing database.

Energy Management System (EMS): Contains applications for monitoring, control and optimisation of the generation and transmission system.

Geographical Information System (GIS): Geocoded data with customer and equipment locations is stored on the GIS database, typically used for asset management, weather data and Outage Management Systems (OMS)

The next level below is the master station, where control system applications and databases reside.

Measurement Data Collection System (MDCS): Collects metering data collected smart meters, also functions as a front-end for data communications. Data is processed in the meter data management system.

Meter Data Management System (MDMS): Preparation for use in external applications in terms of formatting and data organisation.

The master station receives meter data via a variety of communications mediums such as GPRS/GSM, Ethernet, serial and fibre-optic networks. The field equipment gathers data from customers, a network of data concentrators connected to smart meters provides the metering for end users.

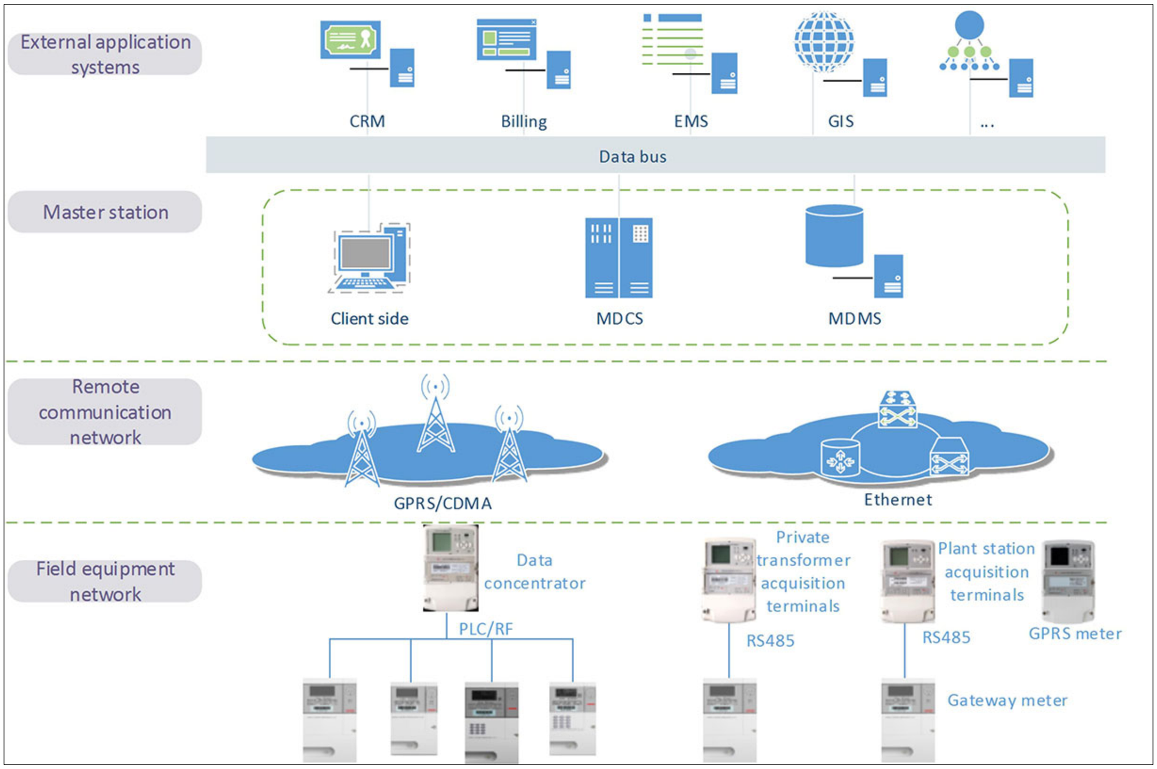


Figure 2-5: Advanced Metering Infrastructure (Topscomm, 2016)

2.7. IEC 61850-based Power Quality Monitoring

IEC 61850 is an Ethernet-based standard that defines communications for IEDs in electrical substations. The primary purpose of the standard is to allow for interoperability between devices from different vendors. To that end, specific functions that are common to the various devices were identified and encapsulated into logical nodes. Logical nodes are data objects that define a particular function in an IED and describe the different attributes that can be derived from it. In the case of logical nodes developed explicitly for power quality monitoring, Edition 1 of the standard defined a limited number that was designed under the logical node class M (Metering). More specifically, these logical nodes were; MHAI (Harmonics and Inter-harmonics), MHAN (non-phase related harmonics), MSQI (sequence and imbalance). As of Edition 2 of the standard, power quality events have been provided with their class, which is denoted by the letter Q. Table 2-1 summarises the new logical nodes.

Table 2-1: Logical Nodes for Power Quality Events

Logical Node	Description
QFVR	Frequency Variation
QITR	Current Transient
QIUB	Current Unbalance Variation
QVUB	Voltage Unbalance Variation
QVTR	Voltage Transient
QVVR	Voltage Variation

The logical nodes described in Table 2-1 can be used to model a power quality monitoring system. IEC 61850 allows for flexibility in the way real-world equipment is modelled. Apostolov (2007), demonstrates the utilization of various logical nodes to achieve a distributed power quality monitoring system, as shown in Figure 2-6. The drawing reflects a model of a system consisting of a power monitoring IED, a merging unit and CB monitoring unit. Logical nodes TCTR (Current Transformer) and TVTR (Voltage Transformer) are present in the merging unit with TVTR Logical Node (LN) transmitting metered quantities to the MMXU LN on the power monitoring IED as sampled values. The MMXU LN sends the quantities as measured values to the QVVR1 and QVVR2 LNs to detect whether a voltage variation event has occurred, with QVVR2 containing an additional input from XCBR LN originating from the CB unit.

IEDs consisting of differing functionalities may exchange information, using the abovementioned system model as an example, the merging unit may send sampled values via the TVTR LN to be processed by the power quality monitoring IED. These values are then used as measured values by the QVVR LN via the MMXU logical node to determine the waveform and whether a voltage sag has occurred. The XCBR LN transmits the breaker status via GOOSE messaging and relays the single or three-phase status of the circuit breaker to perform a control action (Apostolov, 2007).

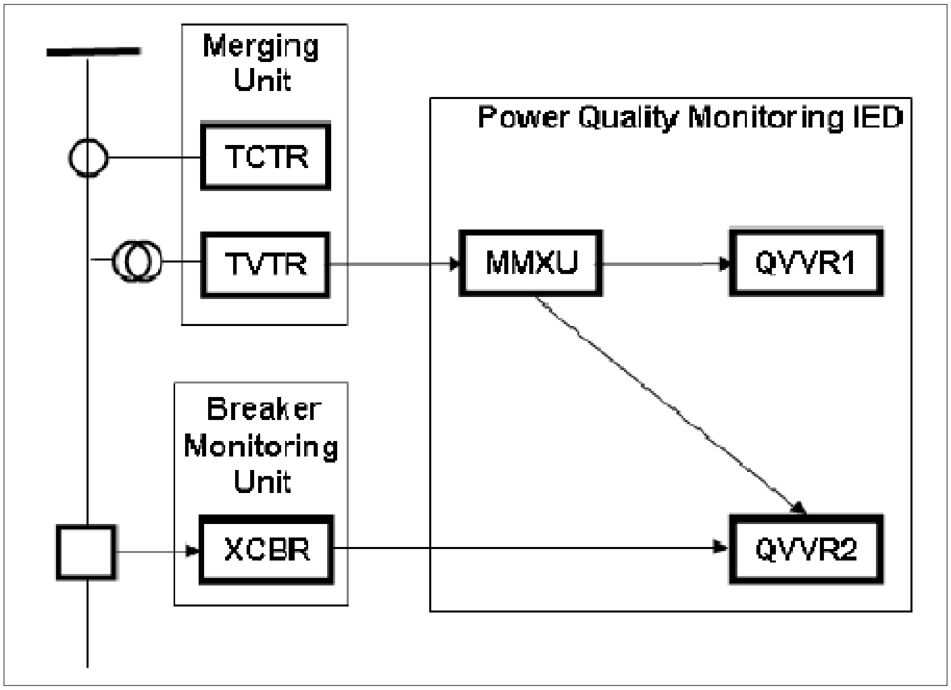


Figure 2-6: IEC 61850 Power Quality Monitoring Distributed Environments (Apostolov, 2007)

2.8. Literature review of existing papers on Power Quality Monitoring

The literature review was done on power quality monitoring techniques. The research was focused on the methods of communications, system architectures, protocols and signal processing techniques. Special attention was also was given to adherence to power quality monitoring standards.

2.8.1. Existing papers on current power quality monitoring

Figure 2-7 graphically presents the number of papers published by year for traditional and current power quality monitoring techniques. All documents presented were chosen to show the progression of power quality monitoring between the years of 1989 and 2019. As such, papers were grouped according to the innovations and findings related to the topic mentioned above. The dataset used to generate the graph consisted of relevant papers obtained by searching for the keywords “Power Quality Monitoring”. Selection criteria were as follows:

1. The paper should cover the topic of monitoring power quality or quality of supply studies.
2. A data acquisition system needs to be described or covered in some capacity.
3. System architecture must be clearly defined.

4. The system described should employ techniques and technologies currently or previously widely adopted in the industry.

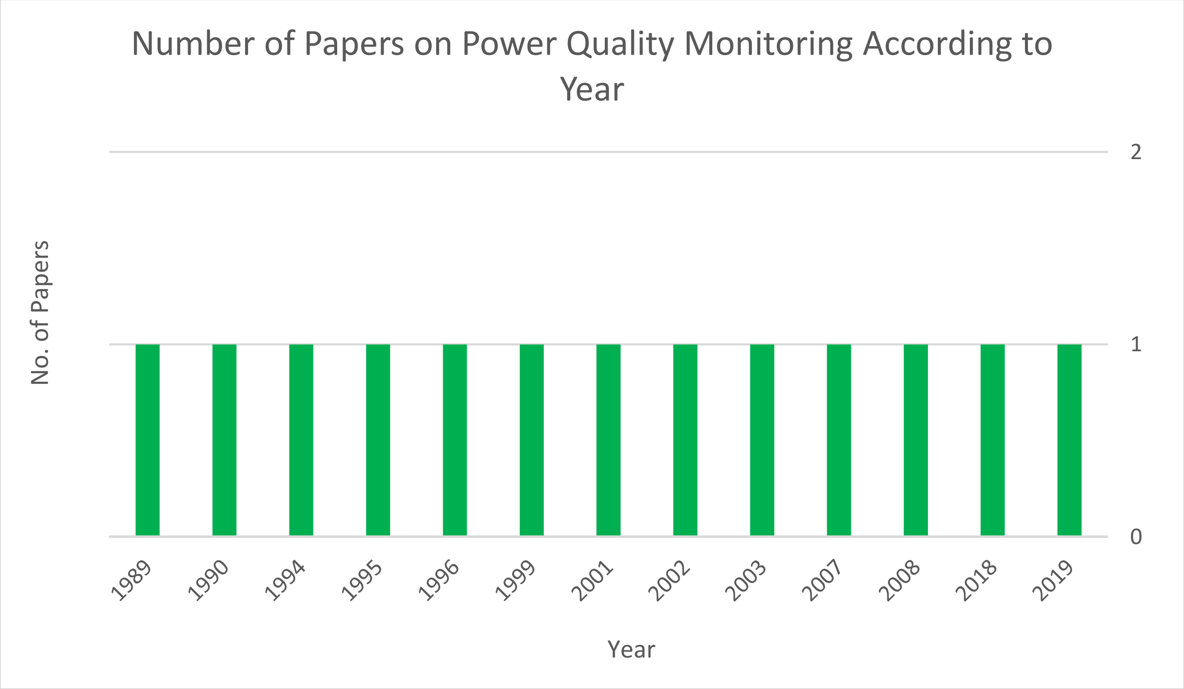


Figure 2-7: Number of papers per year

Once papers were obtained and tallied, it was observed that the number of relevant papers remained mostly consistent, with research into the topic continuously being done within 30 years albeit in a limited fashion. It can be noted that in Figure 2-7, all of the featured years contain a single research paper. Table 2-2 illustrates many different methods and technologies researched over 30 years between 1989 and 2020 to monitor power quality effectively. The table is divided into five columns; the first column indicates the author and the year of publishing, the second column describes the overall aim of the paper, column three provides a brief description of the system architecture. The required hardware and software for the implementation are listed in column four, and finally, the author's conclusions are described in the final column.

Table 2-2: Existing Papers on Power Quality Monitoring

Paper	Aim of the paper	System Overview	Hardware/ Software Required	Author's Conclusions
(Hairabedian, 1989)	Development of a power quality surveying system utilizing off-the-shelf disturbance monitors, telephone lines interfaced with modems. A PC is used for processing and data storage.	Remotely connected disturbances using public telephone lines to transmit data to a central processing PC	Dranetz 646-3 power line monitors Hayes Smartmodem 1200 IBM PC-XT Floppy disks for off-PC data storage Data processing and database programs are written in Basic	The system was implemented and deployed at IBM Kingston and able to remotely communicate disturbances in the electrical network. The power line monitors' memory can store up to 500 power quality events. Volatile memory means that the data is lost if the device is powered down. The two basic programs for processing the raw data stored on the PC reside on floppy disks, one for summary data and the other for general data. This setup means that there is no automated function for processing the data and storing it in readable text files and the available database.
(Clemmensen, 1990)	Summarizes measurement techniques and instruments for site surveys and long-term power	Handheld single purpose meters and multipurpose/ specialised fixed meter test setups	A variety of measurement instruments	Power surveys were found to be useful for monitoring existing power quality issues.

Paper	Aim of the paper	System Overview	Hardware/ Software Required	Author's Conclusions
	quality monitoring.			• A challenge was selecting the appropriate instruments for the given measurement. • The benefit of the longer-term survey is the descriptive information provided regarding the overall PQ of the site. Longer-term surveys require fixed PQ instruments, preferably with remote connection.
(Chandrasekaran et al., 1994)	To develop software for power quality analysis of a power electronics system.	PQ monitoring workstation running analysis software for disturbance classification and identification	Power electronics system Windows-based workstation Database Expert system Analysis software Neural Network	• A PQ analysis system can be used to diagnose and identify PQ issues in power electronics systems by use of computer technology such as AI and a library of equipment to feed information into the algorithm • This is useful for determining the possible PQ that could arise from specific power electronics systems,

Paper	Aim of the paper	System Overview	Hardware/ Software Required	Author’s Conclusions
				the utilisation of power electronics by DERs could be an area of interest.
(Ruiz et al., 1995)	The development of a multi-channel measurement and monitoring system for electric power supply in real-time.	PC with dedicated data acquisition and development card containing a dedicated Digital Signal Processor	CT and VT Signal Conditioner Anti-aliasing and Gain Control A/D Converter DSP Development card PC	The system can effectively monitor low-frequency disturbances. Accuracy was equivalent to traditional instruments used for PQ monitoring. Dynamic conditions could not be analysed due to a lack of commercial equipment to do so. The use of DSP connected to a PC means that the monitoring setup can be expanded upon through software modules.
(Rauch et al., 1996)	The concepts and design philosophies around power quality monitoring systems are discussed. Instrumentation selection, deployment,	Investigative paper	N/A	Important design considerations for a power quality monitoring system are as follows: • Needs to help maintain the quality of service expected from the customer • Data formats contained in the

Paper	Aim of the paper	System Overview	Hardware/ Software Required	Author's Conclusions
	data collection, management and analysis are all explored.			database must be standard and not proprietary - Communication with customers is essential for maintaining good power quality - Integration with the internet could provide an excellent opportunity for customer participation
(Ananth & Morcos, 1999)	The use of Digital Signal Processing (DSP) in power quality monitoring applications is investigated. DSP architecture, operations, and general effectiveness as a hardware/software platform for processing voltage signals.	DSP interfaced to with a host computer	Digital Signal Processor PC	It was demonstrated that DSP could be used to significant effect for power quality monitoring, able to determine up to the 31st harmonic. The versatility of the DSP meant that it could be calibrated to less than the maximum allowable measurement error of less than 5%. The real-time processing kernel allows for multi-tasking and can be ported to other platforms to develop further control and monitoring applications.

Paper	Aim of the paper	System Overview	Hardware/ Software Required	Author’s Conclusions
(Garcez et al., 2001)	The integration of power quality monitoring in a distribution control centre is described in this paper	Power quality monitors send voltage readings and report on any transient or more prolonged interruptions. The voltage signal is processed by a microcontroller unit which then transmits the data to the master station, which acts as a concentrator. The utility host computers in the Supervisory Control and Data Acquisition (SCADA) system stores and displays data. Calculating the PQ indices	PQ Monitoring Unit Microcontroller PC data Concentrator SCADA Master Client PCs	• Provides the utility with the ability to determine whether a fault is localized or system-wide based on the PQ disturbances recorded at various nodes being monitored. • PQ monitoring with SCADA capabilities enables the utility to conveniently determine if they are fulfilling their mandate of maintaining a good quality of supply.
(Hong et al., 2002)	Power Quality monitoring using power line communications (PLC) is dealt with in this paper. The system conforms to the	Various PLC Data Acquisition Systems (Data is acquired via DSP) connected to a PLC modem. The PLC modem is connected to a PC via RS232	Signal Generator PQ Data Acquisition System PLC Modems PC running Java program	• The use of PLC eliminates the need for additional communications lines such as LAN. • The power line modem allows for remote monitoring, although PLC itself

Paper	Aim of the paper	System Overview	Hardware/ Software Required	Author's Conclusions
	IEEE 1159 standard.			has limitations in distance
(Auler & D'Amore, 2003)	A new protocol is presented to monitor power quality via Ethernet. The protocol allows for real-time monitoring and control.	Remote modules capable or monitoring various analogue quantities are connected via fibre optics or twisted pair to a central computer.	Data acquisition modules Ethernet communications network Network switches Central computer	The new protocol was successfully implemented. • No packet loss was detected when the CPU clock speed was at 600 MHz, but lowering the frequency to 166 MHz resulted in a loss of data. • Remote modules of varying processing power can be used, allowing for low-cost DSP based solutions to potentially used alongside industry-grade equipment. • The implementation of encryption was also found to be a possibility but was never fully explored.
(Putrus et al., 2007)	An overview of the power quality events and their characteristics, effects and causes; along	Multiple PQ monitoring techniques are examined. Handheld, Portable, Fixed and Networked meter	None. Research is entirely theoretical.	• Commercially available power quality monitors are increasingly more powerful while getting smaller in size.

Paper	Aim of the paper	System Overview	Hardware/ Software Required	Author's Conclusions
	with the recent trends in monitoring power quality	types are briefly reviewed.		• Necessary monitoring and harmonics analysis are possible on most available equipment. • Artificial intelligence is used in networked multipoint meters; this allows for the development of low cost, but robust PQ monitoring • Long-term analysis of disturbances could help identify trends and possible solutions.
(Ferrigno et al., 2008)	A measurement instrument is built around a field-programmable gate array that is compatible with IEC 61000-4-30 and IEC 61000-4-7. Good accuracy and cost-effectiveness are some of the main objectives of the paper.	A MAX180 IC is used as the data acquisition system. Simulated signals are feed through the MAX180 which is connected to an Altera Stratix II EPS2S60 FPGA. The field programmable logic controller (FPGA) firmware acquires the data, decimates the input signals, then runs them through algorithms	MAX180 Data Acquisition IC Altera EPS2S60 FPGA PC Running FFT Algorithm	• The FPGA applied a novel digital filter approach to estimate the harmonics and interharmonics • The right conditions in the signal injection appeared to generate results in compliance with IEC 61000-4-7 • Testing focused on harmonics but can be expanded to other indices. • Firmware optimisation could be done for all

Paper	Aim of the paper	System Overview	Hardware/ Software Required	Author’s Conclusions
		based on 4-30 and 4-7 to determine whether voltage/harmonic interruption has occurred.		implementations in low-cost FPGAs.
(Artale et al., 2018)	This research project aims to implement power quality monitoring on low-cost smart metering hardware.	STCOMET is a System on Chip (SoC) board found in many existing smart meters. They are typically used for energy metering and PLC functions. The new PQ monitoring functions are programmed directly into the memory of the STCOMET, essentially extending the functionality of the board	EVLKSTCOMET10-1 evaluation board	New PQ metrics were successfully integrated directly into the firmware of the STCOMET. • The accuracy limits were within the IEC 61000-4-30 standard for frequency, magnitude, dips, swells and interruptions. • The feasibility of added PQ metrics to smart meters with no modification to existing hardware has been proven.
(Hassan et al., 2019)	Techniques for power quality monitoring are reviewed in this paper, with a focus on Power Quality Monitoring and	Five aspects of power quality monitoring are examined; power quality monitoring systems and techniques, meter placement, monitoring	No hardware and software are implemented. A case study is performed comparing hypothetical systems.	• SEMCE DAQ developed by (Grigorescu et al., 2013) is found to be the most effective, cost-efficient and flexible method for power quality monitoring.

Paper	Aim of the paper	System Overview	Hardware/ Software Required	Author’s Conclusions
	management systems.	techniques, policies and practises and struggles with power quality improvement. Each aspect consists of several concepts, analysed and compared.		• Wireless communication technologies make it possible to access the monitoring systems from a distance.

2.8.2. Comparative analysis of the existing literature listed in Table 2-2

When acquiring data, instrument selection is a crucial aspect of the data acquisition process. Instrument selection becomes even more critical in the case of power quality monitoring, where multiple types of indices are measured and recorded. Clemmensen (1990) performed research summarising the challenges and considerations for conducting power quality studies. It was found that long term studies were important for determining the overall power quality of a site, instruments installed on-site with a remote connection to a central system was crucial. Putrus, Wijayakulasooriya and Minns (2007) performed a similar study in the 2000s with updated equipment and concepts; a notable similarity is a conclusion that long-term analysis remains the best methods of identifying trends and possible solutions. Other notable findings were that the advancements in technology have made commercially available PQ meters more powerful and feature-rich. However, much smaller in size and the possible usage of artificial intelligence in networked multipoint meters could offer additional benefits in cost-effectiveness and robustness.

Hassan *et al.* (2019) ran a comparative study of various aspects of power quality monitoring, including metering techniques, meter placement and best practices. Several power quality monitoring methods are compared, these included:

- **High-resolution flicker metering:** Frequency spectrum analysis is used to observe and extract the flicker component of a signal.
- **S-Transform Based Power Quality Analysis:** The frequency spectrum is analysed by finding the S-Transform of a signal.

- **SEMCE DAQ:** Data Acquisition System for measuring power quality metrics in smart grids.
- **Online Power Quality Monitoring system:** Web servers and communications systems are merged with PQM systems
- **An integrated power quality monitoring system:** Data is collected from multiple IEDs, the devices all follow the IEC 61000-4-30 standard.
- **FPGA Based Power Quality Monitoring:** Low-cost system connect to a computer using a Wi-Fi module.
- **Power Quality Monitoring Using OPNET:** OPNET (Open Networks Simulator) is used to simulate a network PQM monitoring system and its performance
- **Spectrum Analysis Based Power Quality Monitoring:** Data collected in a typical PQ monitoring system is in the time domain. Spectrum Analysis analyses time-domain data after a frequency domain transform.
- **Power Quality Monitoring Using Pseudo-Measurements:** An intelligent application is used to identify PQ issues in large distribution networks
- **Co-Operative Sensor Network for Power Quality Monitoring:** A distributed system using wireless sensor nodes for power quality monitoring.
- **Web-Services Framework for Power Quality Monitoring:** A web services platform is used for monitoring and processing power quality events. Fault detection and real-time monitoring are possible (Hassan et al., 2019).

The advantages, disadvantages and attributes of each technique and system were tabled and analysed. The research performed in this paper is significant as it provides a comprehensive account of the various power quality monitoring techniques researched over the years. Overall, it was found that the SEMCE DAQ, a power quality monitoring system specially designed for smart grids, proved to be the most appropriate method due to its flexibility, accuracy and use of wireless technologies.

Hairabedian (1989) implemented one of the earlier examples of a networked power quality monitoring system by making use of commercially available disturbance monitors in communication with a central computer via the telephone line. By today’s standards, the implementation may be considered “primitive”, but proves the advantage of remote monitoring through the generation of reports comprising of metering readings from instruments located throughout the system. In comparison, Garcez *et al.* (2001) demonstrations a SCADA system with similar, but improved functionality. The system still used low-speed telephone lines for communications but employed digital signal processing microcontrollers for data acquisition and processing instead of commercially available disturbance monitors. The main advantage of this approach as opposed to the one taken by Hairabedian (1989), is the real-time nature of the data being recorded in the computer system and the inherently low cost of microcontroller vs commercial disturbance monitors.

A significant advancement in the early 2000s has been the continued evolution and adoption of Ethernet for a wide range of remote communications-based applications. Auler and D'Amore (2003) present a new protocol combining MODBUS/TCP, SNMP and COMTRADE communicating over Ethernet. This implementation combines the high speed of Ethernet with the real-time data capture and processing of the approach presented by Garcez *et al.* (2001). The usage of commonly available protocols and Ethernet also makes the system somewhat interoperable with existing SCADA and networked systems. Another advantage of this approach is the flexibility it offers, and the system is not limited to a specific type of measuring instruments. The ability to connect to Ethernet using MODBUS/TCP was the only limiting factor; this paved the way for DSP-based solutions as long an appropriate protocol stack is available.

A commonality among recent and current research on power quality monitoring is the prominence of Digital Signal Processors or DSPs. Digital signal processing became a direction which researchers tended to move towards from the mid-1990s until the present. Ruiz *et al.* (1995) and similarly Ananth and Morcos, (1999) presented a simple yet effective system for measuring power quality using DSP. In both instances, the system was proven to be as sensitive and accurate as traditional instruments such as the disturbance monitors employed by Hairabedian (1989). Since the DSP development card was connected to a PC, the algorithms could be expanded and improved upon through software. The ability to expand the capabilities of the systems was useful for research and development, but could not be used for real-world applications.

Hong, Lee and Choi, (2002) presents a practical, real-world use case of DSP's for power quality monitoring using new technologies. The unique aspects of this implementation are the use of Power Line Communications (PLC) for data transfer to the central PC which eliminated the need for an additional Local Area Network (LAN), and the specialised Texas Instruments TMS320C32 DSP used as a Data Acquisition System. As with previous research on the topic of DSP-based PQ monitoring, this system proved to be effective at obtaining accurate data, with the additional benefit of conforming to the IEEE 1159 standard. The research done by Hong, Lee and Choi, (2002) could be seen as a precursor to the work done by Artale et al. (2018). Their approach bore a similarity to the research mentioned above. However, it took it further by using the more advanced STCOMET System on Chip (SoC), a chip found in many existing smart meters. The EVLKSTCOMET10-1 evaluation board used in the research also contains built-in PLC functionality. It has the additional advantage of being compliant with the IEC 61000-4-30 standard for frequency, dips, swells and interruptions. By the time the work of Artale et al. (2018) was published, second-generation smart meters were already being employed in Italy, the home country of the authors (Volkwyn, 2017).

Research on different methods of data acquisition and processing was performed throughout the ’90s and 2000s; a notable example is research done by Ferrigno, Landi and Laracca (2008). A data acquisition system consisting of a MAX180 connected to an Altera Stratix II EPS2S60 FPGA programmed to run algorithms for IEC 61000-4-30 and IEC 61000-4-7 standard measurements. A novel digital filtering approach was applied to estimate harmonics and interharmonics. The advantage of this approach is that it can be applied to any compatible FPGA, making it possible to develop a low-cost power quality monitoring instrument. There has also been (by the time of writing) limited research on artificial intelligence-based algorithms for processing, diagnosing and identifying PQ issues. Chandrasekaran, Kollipara and Sundaram (1994) are one of the earliest examples of research into this approach of diagnosis and categorisation. The proposed system uses expert systems containing a library of power electronic devices and equipment to feed information to the identification algorithm and identify specific types of PQ issues.

2.9. Literature Review of existing papers on IEC 61850-based power quality monitoring

The second literature review was done on IEC 61850-based power quality monitoring. The research was focused on the implementation of the standard, methods of communications, power quality specific updates to the standard and system architectures for monitoring systems.

2.9.1. Existing papers on IEC 61850-based power quality monitoring

Figure 2-8 graphically presents the number of papers published by year. All of the documents presented were selected as the most relevant to show the progression of research in IEC 61850-based power quality monitoring between 2009 and 2020. The dataset used to generate the graph consisted of relevant papers obtained by searching for the keywords “IEC 61850 Power Quality”. Selection criteria were as follows:

1. The paper should cover the topic of IEC 61850-based power quality.

2. Parts of the standard being used for implementation must be described.
3. System architecture must be clearly defined.
4. A selection of papers describing systems with telecommunications-based data transfer along with communications protocol.

Once papers were obtained and tallied, it was observed that the papers only span 11 years between 2009 and 2020. This short period indicates that research on IEC 61850-based power quality monitoring only came into the fore in the 2010s and is still in its infancy. The IEC 61850 standard only started seeing adoption in the early 2000s, so the lack of research of any significance is expected, as such, a small number of relevant papers were found using the keyword “IEC 61850 Power Quality”.

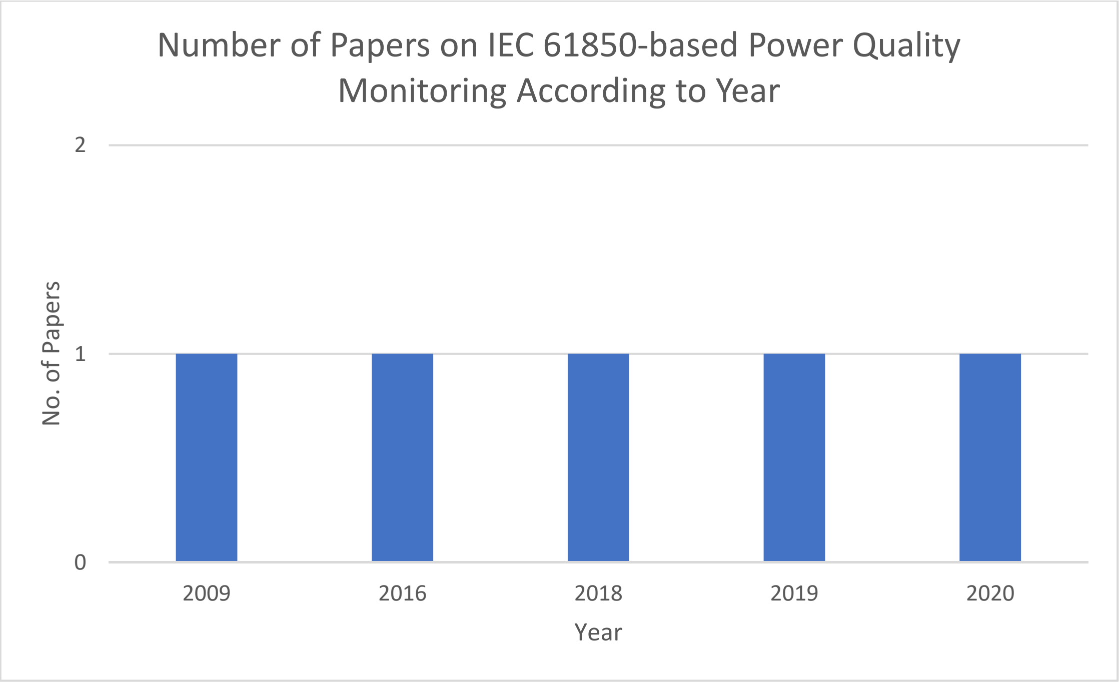


Figure 2-8: Number of papers per year covering the topic of IEC 61580-based power quality monitoring

Table 2-3 is a collection of research papers focusing on IEC 61850-based implementations of power quality monitoring over 11 years. Similar to Table 2-2, the table is divided into five columns. The first column indicates the author and the year of publishing; the second column describes the overall aim of the paper. Column three provides a brief description of the system architecture. The required hardware and software for the implementation are listed in column four, and finally, the author’s conclusions are described in the final column.

Table 2-3: Existing Papers on IEC 61850-based Power Quality Monitoring

Paper	Aim of the paper	System Overview	Hardware/ Software Required	Author's Conclusions
(Apostolov, 2009)	The object modelling of IEC 61850-based power quality monitoring devices, as well as the testing of said functions, is described in this paper.	Two distributed system architectures are discussed: 1. A merging unit-based architecture comprising of a synchronized unit connected to merging units which are interfaced with the PQ monitoring and recording unit via Ethernet. 2. A protection IED based system comprising of IEDs connected to the CTs and VTs, these IEDs communicate with the PQ monitoring and recording unit via Ethernet	N/A: Research and knowledge-based paper	• The IEC 61850 standard allows for the creation of a variety of PQ monitoring systems with varying topologies based on equipment availability and capability. • A high sampling rate of up to 256 samples/cycle is useful for PQ applications • Multicast sampled values allow for multiple readings from IEDs and merging units to be acquired and processed by the recording unit • An added Synchronizer could allow for a time accuracy close to 1 microsecond
(Steel, 2016)	This paper introduces the control and monitoring of the PV energy	At the top level of the system architecture, there is a zone substation in charge of managing the	IEC 61850-Based substation controller	• The use of GOOSE allows for quicker detection of power quality issues and faster fault clearing

Paper	Aim of the paper	System Overview	Hardware/ Software Required	Author's Conclusions
	storage scheme using IEC 61850 within a wireless communications network.	given area's PV plants. The substation communicates with PQ monitors in the field via IEC 61850. Field events are communicated to the substation via GOOSE; this allows for fast detection and response times. Locally, the data is sent to an aggregator that can be used to view data onsite.	Power Quality IEDs Suburban data aggregator Local monitoring and control PC	time, thus improving grid resilience. • The IEC 61850-based power quality relays allow for much-improved smoothing speed and voltage ramping to compensate for the variability of PVs and the effects of this variability on power quality. • Wireless technologies have the disadvantage of being shared mediums. Although collision detection is available, there is still the possibility of other devices using the available bandwidth. An IRIG-B time source could allow for synchrophasor data to be read reliably fast.
(Aftab et al., 2018)	An IEC 61850 communication-based energy management system within a	Several case studies demonstrating the usage of XMPP (Extensible Messaging Presence	IEC 61850-7-420 DER Logical Nodes	• An Energy Management system based on IEC 61850 based standardised communication and

Paper	Aim of the paper	System Overview	Hardware/ Software Required	Author's Conclusions
	microgrid containing electric vehicles.	Protocol) as a communications middleware for transporting MMS data over a wide area network in a microgrid with EV charging infrastructure.	IEC 61850-90-8 E-Mobility Data Model XMPP Messaging Protocol	modelling catered for microgrids with a large number of EV chargers are proposed. • Vehicle-to-Grid operation (discharging of EV to the grid) is made possible through coordination via IEC 61850 and XMPP.
(Jinhao et al., 2019)	This paper implements a communications protocol test on a system based on SISCO MMS-ease lite and IEEE P1159.3 (PQDIF).	The power quality monitoring platform proposed in this paper consists of digital and analogue signal injection devices connected to power quality monitoring IEDs. The client PC is connected to the power quality monitor IEDs via Rj45 and fibre, respectively.	Fluke 6100 power quality calibrator (analogue power source) RT Series digital power source Digital power quality monitor Analogue power quality monitor PC	• Verification of PQDIF files is made possible using the system • IEC 61850 standard messages communicating via MMS can also be verified
(Della Giustina et al., 2020)	A model-based approach in designing automation logic for fault location and restoration within medium	An OMICRON CMC's outputs are connected to several IEDs which form MV feeders. Four "substations" are created using groups of three IEDs, each	OMICRON CMC 353 signal generator Schneider Electric Saitel DR Platform IEDs	• The use of IEC 61850 is a key enabler for a more straightforward implementation of complex automation algorithms. • The IEC 61850-90-6 and IEC 61850-7-420

Paper	Aim of the paper	System Overview	Hardware/ Software Required	Author’s Conclusions
	voltage networks.	simulating an individual circuit breaker. Each substation contains an RTU which serves as an IEC 61850 gateway, and this is connected to a PC for engineering and configuration.	Sequential Function Chart (SFC) IEC 61131-3 programming language	may be used to simplify complex systems further.

2.9.2. Comparative analysis of the existing literature listed in Table 2-3

The modelling approach of an IEC 61850-based system was found to be one of the critical determinants in implementing a power quality monitoring system with an appropriate topology according to the scale, complexity and technological capabilities of the system being modelled. Apostolov (2009) details the various methods of modelling a power quality monitoring system that makes use of IEC 61850 logical nodes. In this paper, two distributed system architectures are discussed;

- A merging unit-based architecture comprising of a synchronized unit connected to merging units which are interfaced with the PQ monitoring and recording unit via Ethernet
- A protection IED based system comprising of IEDs connected to the CTs and VTs; these IEDs communicate with the PQ monitoring and recording unit via Ethernet.

It was found that the IEC 61850 standard allows for the development of PQ monitoring systems with various topologies based on the available IED’s and their capabilities. The Edition 2 logical nodes such as QVVR implemented in power quality monitoring IEDs could easily interface with existing merging units and bay control relays to extract additional information as long as the algorithms for computing voltage variation events allows it. These could be implemented on an existing PQ metering IED or a DSP-based development board running and IEC 61850 protocol stack. A high sampling rate of 256 samples/cycle combined with multicast sampled values could allow for multiple readings from IEDs and merging units to be acquired for processing by the recorder and quickly react to the changes in conditions.

The modelling approach described by Apostolov (2009) can be noted as a framework with which Steel (2016) uses to develop his network of PV plants communicating with deployed PQ monitors via IEC 61850. In this system, field events are transmitted between devices using GOOSE messaging, allowing for faster response times for voltage ramping and smoothing. In this aspect, the work done by Apostolov (2009), and Steel (2016) can be seen as companion pieces in conception and implementation, respectively. The advantageousness of high sampling rates, multicast sampled values, and the use of synchrophasors coupled with phasor data concentrators (PDCs) networked with IEC 61850-based power quality monitors are showcased in both papers. For utility applications, a secure yet vendor interoperable system is vital, particularly with the increase in the use of fluctuating renewable resources that require a quick response when power quality issues are detected and demonstrated by Steel (2016). This implementation yielded better power quality by mitigating the variability in PV generation; this is only possible in a network comprising of interoperable vendor instruments.

A practical application of the IEC 61850-based communications can be seen in the research performed by Jinhao et al. (2019). In this research, it is noted that Power Quality Interchange Data Format (PQDIF) files do not always adhere to the formatting standards stipulated in IEEE 1159.3. To remedy the lack of standard formatting, a test power quality system is evaluated for data consistency at PQDIF level. The data is sent to a remote PC using standard IEC 61850 MMS communications instead of File Transfer Protocol (FTP). The data contained in the generated PQDIF file is sent as several datasets containing multiple logical nodes such as MMXU, MHAI and QVVR. This data is broken up into two logical devices defined in the configuration file. This research proves the importance of interoperable, standard protocols in communications systems, and how IEC 61850 could help mitigate the impact of non-standard implementations of file formats and protocols such as PQDIF by providing a standard method of transporting the data between devices and systems.

The IEC 61850 standard contains several parts that act as extensions to the standard. These “extensions” are typically modelling paradigms, and logical nodes developed as technology progresses, and the need arises. An example of this is the work done by Aftab *et al.* (2018), which presents a system for the energy management of microgrids with a high penetration of electric vehicles. It uses the extended IEC 61850-90-8 part (Object model for E-mobility) and IEC 61850-7-420 part (Distributed energy resources logical nodes) of the standard to model the various exchanges of data. A key takeaway from this research is the control of bidirectional vehicle-to-grid (EV feeding into the grid) and grid-to-vehicle (EV drawing energy from the grid) operations. A system such as the one described would present challenges in maintaining good power quality due to the bidirectionality of the system. In this case, Edition 2 logical nodes for power quality event are integrated with the various measurements to monitor power quality and electric vehicle charging points. Della Giustina *et al.* (2020) use a similar approach, using extended IEC 61850 models such as the IEC 61850-90-6 part (Use of IEC 61850 for Distribution Automation Systems) combined with the IEC 61850-7-420 part to model a system for fault localisation, isolation and restoration. The system is tested using an OMICRON CMC and several IED's to simulate MV substation feeders. This test illustrates how the modelling process of IEC 61850 systems can be used to implement a power quality monitoring system using standard object models regardless of the application.

2.10. Literature Review of existing papers on Smart Grid technologies in Power Quality Monitoring

The literature review was done on smart grid technologies and their integration in power quality monitoring. The research is focused on the methods of communication, system architectures, protocols and signal processing techniques. Special attention is also given to adherence to power quality monitoring standards.

2.10.1. Existing papers on Smart Grid technologies in power quality monitoring

Figure 2-9 graphically presents the number of papers published by year. All of the documents presented were chosen to show the evolution of smart technologies in power quality monitoring from 1996 to 2020. The dataset used to generate the graph consisted of relevant papers obtained by searching for the keywords “Smart Grid Power Quality” to return research papers featuring smart grid and advanced telecommunications technologies. The search term “Distributed Power Quality” is used to return topics related to distributed generation and distributed power quality monitoring. Selection criteria were as follows:

1. The paper should cover the topic of power quality monitoring using smart grid technologies or in distributed power grids.
2. A data acquisition system needs to be described or covered in some capacity.
3. System architecture must be clearly defined.

4. A selection of papers describing systems with telecommunications-based data transfer along with the communications protocols.

2.11. Once papers were obtained and tallied, it was observed that the number of relevant papers remained mostly consistent until the mid-2010s, with research into the topic continuously being done within 24 years. After the mid-2010s, the number of research papers spiked, which is likely in part due to the emergence of DSPs as a viable technology for power quality monitoring. The prominence of DSP may be a result of the rise in the adoption of smart meters. Before the 2000s, DSP-based remote systems were discussed typically as a proof of concept in a lab environment. DSP-based remote systems along with the development of the smart grid framework, spurned interest in power quality monitoring, specifically using advanced information and communications technology (ICT) in a distributed environment. In recent years, there has been an emphasis on power quality monitoring as an integral component in power system optimisation in the smart grid. Research papers focusing more on the redefined roles of power quality monitoring devices in the electric grid in terms of revenue maximisation, grid optimisation and maintaining a good quality of supply in an increasingly solar and wind energy-focused generation environment.

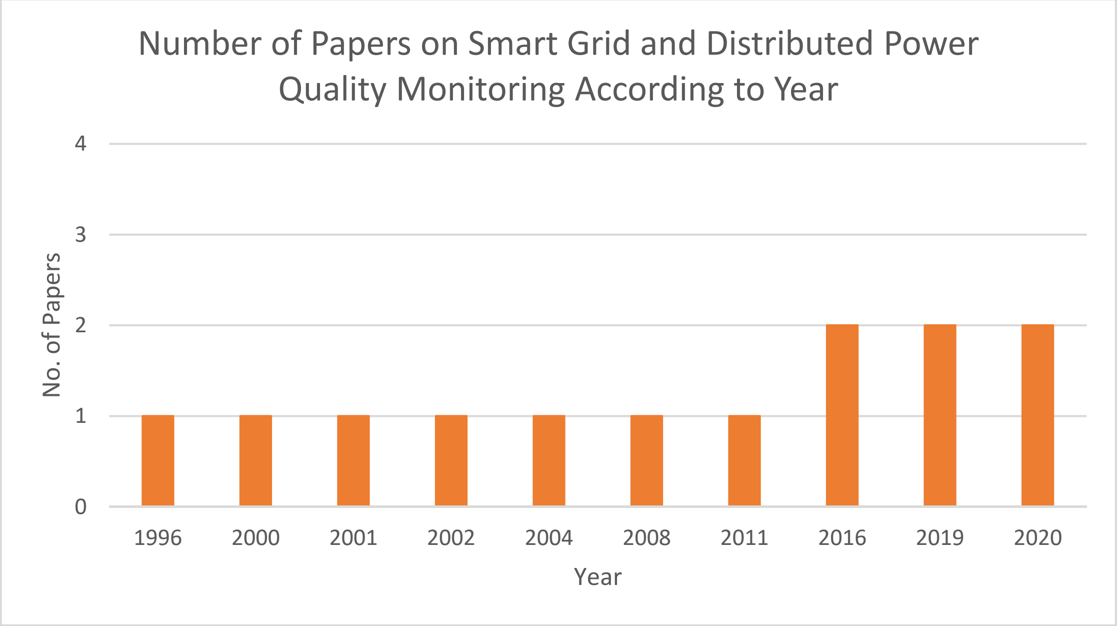


Figure 2-9: Number of papers per year dealing with smart grid technologies in power quality monitoring

Table 2-4 illustrates many different methods and technologies researched over 24 years between 1996 and 2020 to monitor power quality effectively. Similar to Table 2-2 and 2.3, the table is divided into five columns. The first column indicates the author and the year of publishing; the second column describes the overall aim of the paper; column three provides a brief description of the system architecture. The required hardware and software for the implementation are listed in column four, and finally, the author's conclusions are described in the final column.

Table 2-4: Existing Papers on Smart Grid and Distributed Power Quality Monitoring

Paper	Aim of the paper	System Overview	Hardware/ Software Required	Author's Conclusions
(Saari et al., 1996)	Paper aims to implement remote power quality monitoring in existing remotely connected kWh meters.	kWh meters are connected to central servers using a local area network. The servers are connected to the database and clients running software for billing and operations.	kWh meters (smart meters) Reading server Job manager Database server Billing and data analysis server Control client PCs	• Increases versatility and usefulness of the existing kWh meters • A low-cost solution for PQ monitoring • Easier mass rollout due to low cost • Combination of meters can help give an overall view of the power quality in the network
(Sakis Meliopoulos et al., 2000)	The paper aims to demonstrate the use of GPS synchronized Phasor Measurement Units to measure the presence of harmonic distortion present in a transmission system	PMU connected to a CT and VT for currents and voltage measurements, respectively. The PMU is time synchronised using GPS and connected to a PC which then transmits data to various locations for further processing.	PMU GPS antenna A Personal Computer running data acquisition software Harmonic Data Concentrating server	The benefits of the time-synchronized measurement system are shown with the increased redundancy of the system due to the presence of additional phase angle measurements. Accuracy is also another factor to be noted; the use of 16-bit ADC's improves this aspect. The system captures snapshots of the

Paper	Aim of the paper	System Overview	Hardware/ Software Required	Author's Conclusions
				harmonic state of the system using harmonic state estimation and capturing it every 15 minutes.
(Leou et al., 2001)	This paper presents the concept of a web server being used for remotely monitoring power quality	The proposed system has a power quality monitor transmitting data to a PC which is stored in a local database. The local database is connected to a webserver which outside clients can access via the internet.	Power Quality Monitor Data Acquisition PC containing a database and a connection to a web server A web server PC connected to the internet Client machines	• (Active Server Pages) ASP and Java made interacting connecting and interacting with the database directly from the webserver possible. • The ability of the webserver to access the monitoring data database gives engineers the ability to monitor power quality from anywhere. • Role-based access allows for a limited amount of security, but not enough for sensitive data such as power quality readings
(Lee et al., 2002)	The architecture and design philosophy of an extensive web-based multi-	A mixed topology WAN is comprising of several PQ monitoring units installed on customer	PML 7600ION Power Quality Meters Dial-up modems	• Monitoring multiple remote sites could result in challenges such as limited communication

Paper	Aim of the paper	System Overview	Hardware/ Software Required	Author’s Conclusions
	channel power quality monitoring system is described.	premises. All the PQ indices are communicated via an Ethernet network to a central server system which contains web servers to host the data.	Digital Subscriber Line (DSL) Modems Hubs Switches Servers	speed, availability and reliability • Cybersecurity concerns of the end-user are also a significant issue • Capital investment of such a system would be high due to the infrastructure and specialised equipment required • Wavelet Transforms could improve transient detection.
(Chen, 2004)	This paper proposes a new approach to enhance interoperability between power quality instruments from different vendors. The aim is to demonstrate that interoperability is possible using existing software and	The system allows different distributed devices to communicate with each other in a protocol-agnostic manner using typical object request broker architecture (CORBA). Various devices communicate via object request brokers (ORB) configured as servers and clients.	Windows 2000 based PCs are acting data acquisition units. Typical Object Request Broker Architecture (CORBA) programming interface.	• The project establishes a way for devices from different vendors to share data regardless of protocol or file format (PQDIF or COMTRADE). • However, the interfaces ~~would~~ only gain acceptance if recognised by international bodies such as the IEEE and IEC.

Paper	Aim of the paper	System Overview	Hardware/ Software Required	Author’s Conclusions
	networking technologies.			
(Lee & Lai, 2008)	This paper explores the use of SMS for a low-cost power quality monitoring solution. The intention is to create a system whereby the facility manager receives an SMS informing them of % voltage dip in 160 characters or less.	An energy meter is connected to a PLC via RS485. The PLC is connected to GSM Modem running MODBUS firmware which transmits data via the GSM/GPRS network to another SMS server storing the data in a database, or users subscribed to the message group.	Energy meter PLC GSM Modems SMS Server Database for historical data Cell phones for receiving messages	- The concept of an SMS based monitoring system is simply introduced. The inherent limitations are the character limits of SMS, which means only low data volumes may be transmitted, the reliance of critical data on public infrastructure and the data security risks. - Some advantages are the low cost and simplicity of the system architecture.
(Montoya et al., 2011)	This paper introduces an open-source software-based platform for energy management and power quality monitoring.	The system is software-based and consists of a web server presenting data from a database. The data is captured by various smart sensors and transmitted to the PC hosting the software modules for processing and	Measurement sensor Data Acquisition System X86 based PC Webserver Database HTML5 based webpage	A modular, open-source web-based program was developed. The most notable aspect of the project was the development of a completely free, license-free tool for energy and power quality monitoring. The project was built using standard, non-proprietary tools. The resultant system makes it

Paper	Aim of the paper	System Overview	Hardware/ Software Required	Author's Conclusions
		computing the samples. The webpage can be accessed from the web	JavaScript and AJAX scripting languages	easier to quickly deploy a PQ monitoring system making use of low-cost embedded systems and smart sensors.
(Brittes et al., 2016)	This article proposes the "smartening" of power quality assessment, information exchange, data analytics from a corporate level and for external players.	The proposed system relies on a deregulated electricity market and smart grid technologies to enable customers, utilities and business to optimize the monitoring of PQ by utilising data analytics at every level (IED, substation functionality and system-wide). The PQ monitoring portion consists of low-cost IEDs that collect 3 kHz snapshots of current and voltage. These snapshots are stored locally and saved as COMTRADE files which are then stored in the substation's local database. The calculated indices	Low-Cost IEDs Advanced Metering Infrastructure Smart Sensors Smart Substations SCADA System Business Analytics system	• The assessment of PQ becomes a systemic high-level analytics-based process • The use of smart IEDs and PQ hardware is essential • Focus is on data that is of technical and economic relevance based on information and knowledge • All stakeholders become participants in the monitoring of the protection, metering, maintenance, and quality of service • The maintenance of good power quality could maximise profits and operational benefits of all stakeholders.

Paper	Aim of the paper	System Overview	Hardware/ Software Required	Author’s Conclusions
		and load data can be accessed remotely via a webpage.		
(Regula et al., 2016)	This paper deals with PQ monitoring in a SCADA smart grid system containing several distributed energy resources.	The system consists of a controllable grid created on LabVIEW. The main VI is a distribution network consisting of PV, Hydro, Wind DERs. Consumers consist of an industrial zone, a city and two villages, sub-VI’s were built for Metering. The simulation model was built using MATLAB.	LabVIEW MATLAB (Matrix Laboratory) /Simulink	• The simulated Lab View environment provides a useful tool for simulating the effects of DERs on power quality. • The connection of DERs to requires careful consideration of location or placement and the types of power electronic converters used.
(Ferreira et al., 2019)	This paper describes the implementation of a multi-protocol sensor node (Pycom FiPy) connected using LoRaWAN and Wi-Fi. A performance assessment between the two communications protocols is performed.	Pycom FiPy sensor nodes deployed into the field transmit data acquired by a Microchip ATM90E26 microcontroller. This setup is connected wirelessly using the LoRaWAN protocol with a Raspberry Pi 3 connected to a LoRaWAN concentrator. The data is transferred to	Microchip ATM90E26 Pycom FiPy sensor nodes Raspberry Pi 3+ LoRaWAN concentrator MQTT software InfluxDB and Grafana	• A LoRaWAN-based power quality monitoring system is feasible for the quasi-real-time operation of sensors. • The LoRaWAN protocol is only useful for lightweight payloads; thus, applications need to be designed for low bit-rate operation.

Paper	Aim of the paper	System Overview	Hardware/ Software Required	Author’s Conclusions
		an online server using MQTT, and the information is made viewable using InfluxDB and Grafana,		
(Mlakic et al., 2019)	An Intelligent Electronic Device (IED) used for gathering automatic meter readings (AMR). Open-source software is used for measuring a low voltage system with a large number of sensors. The IED can be accessed using IoT technologies.	Fringe sensors are used for measuring and obtaining reference information. An Arduino UNO R3 is used as a Wi-Fi switch for the sensors; then a PC gathers the data and performs client-side calculations. The server is written using the Open-source Java programming language.	Current and Voltage Transformers Arduino UNO R3 ENC28J60 Ethernet Module Network Router Personal Computer	• Open-source technology can be used to create tailor-made, low-cost interactions between IEDs and electrical applications. • Using the Java programming language allows the written application to query custom-made IEDs (CMIED) from multiple platforms
(Alonso et al., 2020)	A smart sensor that provides the ability to dynamically adjust settings during real-time operation, in coordination with other smart sensors spread	A smart sensor within a smart grid with distributed generators. The smart sensors publish adjustment parameters using GOOSE messages to change relay settings dynamically.	N/A	• New protective devices and protection schemes must be developed to improve the reliability of smart grids. • Smart sensors able to dynamically optimise settings in real-time can play a vital role in

Paper	Aim of the paper	System Overview	Hardware/ Software Required	Author's Conclusions
	over an extensive network.	Various case studies are used to demonstrate the functionality.		smart grid networks with DGs. • Smart sensors reduce the activation time of breakers, thus improving grid reliability.
(Raja Singh et al., 2020)	A novel IoT-based low-cost smart meter designed to measure and analyse power quality parameters is presented in this paper—a microcontroller using the IEEE 1459-2010 standard to process voltage and current signals.	Current and voltage sensors are connected to a signal conditioning circuit, which is in turn connected to a zero-crossing detection circuit. The setup mentioned above is connected to an ATMEGA4809 microcontroller with an integrated ESP32 for Wi-Fi or Bluetooth communications with a web-based platform.	Voltage and current transducer Signal conditioning circuit Zero-crossing detector Microcontroller ThingSpeak IoT platform	• The IEEE 1459-2010 standard can be easily implemented using commonly found metering and computing hardware. • Wireless communications and IoT technologies have matured to the point where smart meters have become accessible and easily implementable using low-cost hardware and software platforms.

2.11.1. Comparative analysis of the existing literature listed in Table 2-4

Smart meters have become the focal point of the smart grid framework. Their continued deployment has enabled network operators to integrate them into monitoring for purposes beyond consumer billing, as explained in section 2.6. This comparative analysis shall focus on the integration of smart meters and advanced Information and Communication Technology (ICT) in power quality monitoring for increased situational awareness, control and mitigation.

Saari et al. (1996) present an early iteration of an Advanced Metering Infrastructure (AMI) system with integrated power quality monitoring. The proposed system uses remotely readable meters that can produce both electricity billing and power quality data. The data is sent to a computer system, not unlike the one described in section 2.6. It was found that a traditional meter with power quality monitoring capabilities provided a level of flexibility previously unavailable. The ability to perform power quality studies remotely provided the billing and data analysis servers with the necessary data to perform fault analysis, location and mitigation. The work of Leou, Chang and Teng (2001) and Lee, Lai and Tse (2002) follow similar approaches, with the main differences being a focus on web access and the benefits of engineers being able to access data from anywhere. In both conclusions, it is acknowledged that the use of internet technologies poses a cyber-security threat with sensitive data and the power system as a whole being vulnerable to attack.

Montoya *et al.* (2011) presents an open-source platform for power quality monitoring in which the software could be modified according to the user's needs. The use of open-source software enables the development of a license-free tool for energy and power quality monitoring. It was noted that the implemented system made it easy to deploy and modify a system using low cost embedded systems and smart sensors.

The usage of low-cost smart meters and hardware in an IoT (Internet of Things) environment has become a popular topic over the past few years. Ferreira *et al.* (2019) explore the viability of using a LoRaWAN-based communications network to deliver quasi-real-time sensor reading in lightweight data packets in an urban environment. Whilst Raja Singh *et al.* (2020) implements an IoT based IEEE 1459 compliant solution using a microcontroller that communicates with remote servers via Wi-Fi. The system in question monitors an AC motor connected to a DC load; the AC load is controlled using a LabView program. The monitoring system consists of an ATMEGA4809 microcontroller integrated with an ESP32 microcontroller for Wi-Fi and Bluetooth communications. Measurements taken using the algorithms implemented on the ATMEGA4809 were verified using a handheld power quality analyser. It was concluded that wireless technologies coupled with cloud-based IoT platforms such as “ThingSpeak” could potential be used easily remotely monitor large machines such as AC motors from multiple devices.

This flexibility was explicitly useful in a system within the distributed environment, further enforcing the role of wireless IoT technologies in Smart Metering.

Finally, the utilisation of low smart power quality meters to gather data for analytical and optimisation purposes has also been investigated by Brittes *et al.* (2016). They explored the concept of power quality monitoring being a system-wide high-level analytics-based process that can only be successfully executed by connecting thousands of smart meters in a deregulated smart grid. A conventional PQ assessment hierarchal system is replaced by “PQ Smart Assessment” framework to account for the bidirectional power flow producer/consumers (“prosumers”) and the general growth of fluctuating renewable energies within the power grid. This revised framework allows for the participation of all stakeholders via the interconnection of customer premises’ smart meters, substation IEDs, central control centres and business applications. The standout aspect of this approach is the concept of collaboration between stakeholders to improve service delivery and potentially maximise profits by reducing losses. It is found that a standard Customer Information Model (CIM) supporting file formats such as COMTRADE and PQDIF is required. Combined with interoperable IEC 61850 standard-compliant devices, the CIM allows for a flow of information from substations to the Corporate Service Bus. This flow of information provides a bidirectional connection to the Utility functional Centre. The two areas mentioned above contain several zones used for running various analytics and data exchanges between the available applications within the system.

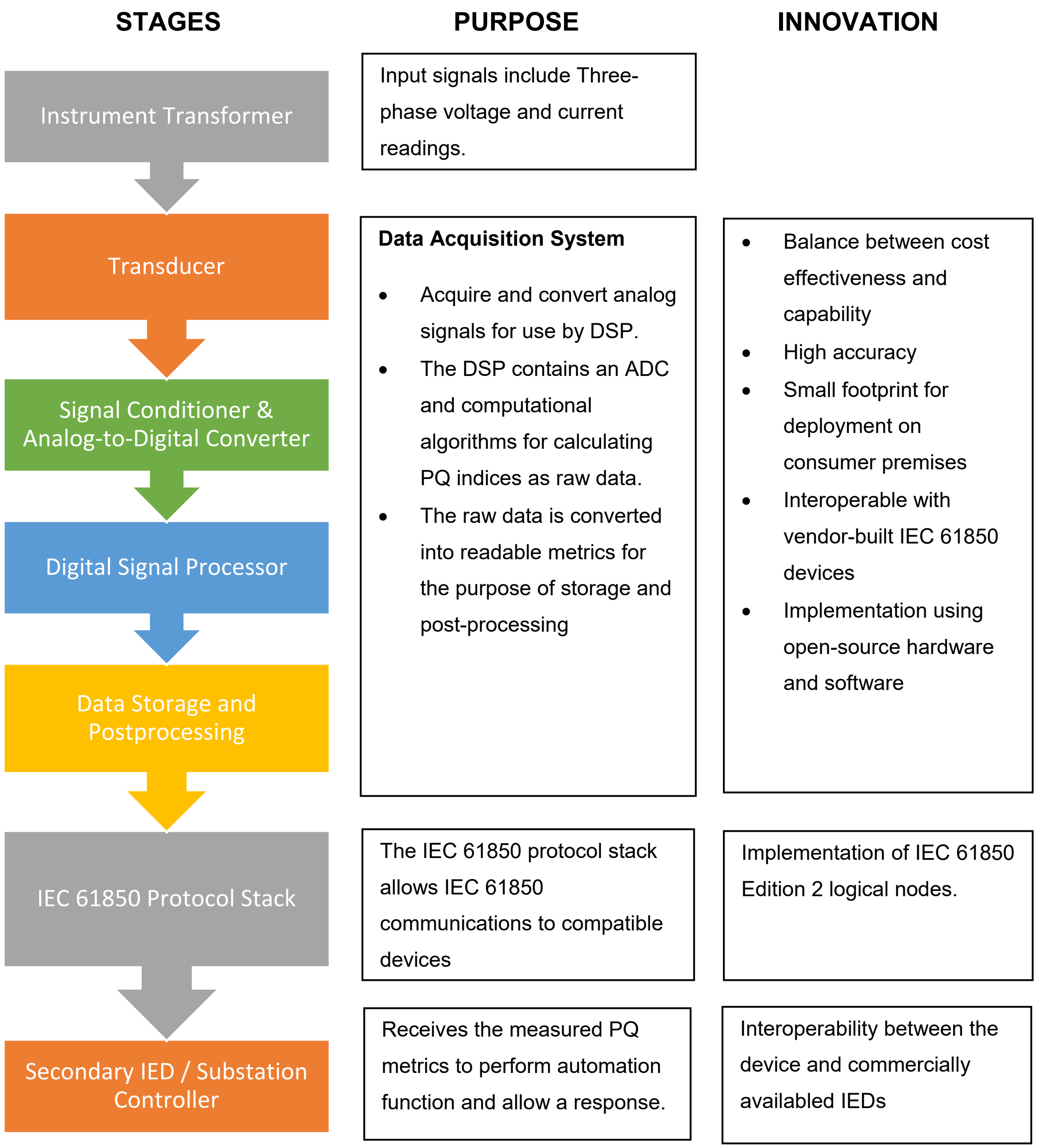
2.12. System architecture based on Literature Review

Performing the literature review reveals some common trends and technologies in the research and development of power quality monitoring. Based on the analysis performed during the literature review for the three research categories on power quality monitoring; this research project aims to develop a power quality monitoring instrument that is an accurate, precise, low cost, secure and able to communicate with other IEDs and utility equipment. According to the papers reviewed, it may be discerned that the optimal way to achieve the abovementioned goals is to:

- Utilise digital signal processors for data acquisition, processing and postprocessing as demonstrated in (Ruiz et al., 1995), (Ananth & Morcos, 1999), (Hong et al., 2002), (Ferrigno et al., 2008) and (Artale et al., 2018).
- Use a low-cost embedded system (Microcontroller, FPGA or ARM processor) for data processing, calculation algorithms and communications stack implementation; this is demonstrated in (Garcez et al., 2001), (Ferrigno et al., 2008), (Artale et al., 2018), (Ferreira et al., 2019), (Mlakic et al., 2019) and (Raja Singh et al., 2020).
- Implement the IEC 61850 standard for data modelling and mapping as demonstrated in (Apostolov, 2009), (Steel, 2016), (Aftab et al., 2018), (Jinhao et al., 2019) and (Della Giustina et al., 2020).
- Implement a high-speed Ethernet-based communications network for data transfers as in (Lee et al., 2002), (Brittes et al., 2016), (Mlakic et al., 2019) and (Auler & D'Amore, 2003).

This approach would more than likely yield a design that makes use of modern, readily available technology, but is forward-thinking in its form-factor and emphasis on flexibility.

Based on the findings in the literature review and the structure of a power quality monitoring system shown in Figure 2-1, a basic proposed system architecture can be modelled. Figure 2-8 shows the proposed system architecture comprised of seven "stages" listed represented by a flow diagram indicating the flow of information. Each stage serves a specific purpose which is specified adjacent to the flow diagram, the possible innovation or unique achievement of the implementation at each stage is also listed.


STAGES
PURPOSE
INNOVATION
Instrument Transformer
Transducer
Signal Conditioner & Analog-to-Digital Converter
Digital Signal Processor
Data Storage and Postprocessing
IEC 61850 Protocol Stack
Secondary IED / Substation Controller

Input signals include Three-phase voltage and current readings.

Data Acquisition System

- Acquire and convert analog signals for use by DSP.
- The DSP contains an ADC and computational algorithms for calculating PQ indices as raw data.
- The raw data is converted into readable metrics for the purpose of storage and post-processing

- Balance between cost effectiveness and capability
- High accuracy
- Small footprint for deployment on consumer premises
- Interoperable with vendor-built IEC 61850 devices
- Implementation using open-source hardware and software

The IEC 61850 protocol stack allows IEC 61850 communications to compatible devices

Implementation of IEC 61850 Edition 2 logical nodes.

Receives the measured PQ metrics to perform automation function and allow a response.

Interoperability between the device and commercially availabled IEDs

Figure 2-10: Proposed System Architecture

Although subject to adjustment, Figure 2-8 serves as the basic architecture for the final implemented system. The implementation has been divided into seven “stages”, these seven stages all serve a particular purpose and can be considered modules in a more extensive system. Although each step performs a specific purpose, the steps can be further combined and grouped into more complex functionalities such as “data acquisition”. With consideration to the data acquisition function, this consists of the transducer, signal conditioning, analogue to digital conversion, DSP, data storage and post-processing phases. These combined phases work in tandem to form a data acquisition system which can be implemented using a wide variety of methods.

The innovations expected to be achieved in this specific implementation are also detailed in figure 2-8; this highlights particular constraints and deliverables to effectively achieve the aims and objectives of this proposed research work. As detailed under "innovations", there shall be a great emphasis on developing a device with a small footprint and an even greater focus on the software algorithms and communications procedures, precisely the IEC 61850 standard.

2.13. Conclusion

In this chapter, a literature review was carried out on power quality monitoring, with an emphasis on the communications and data acquisition methods. A large group of relevant papers on power quality monitoring are identified, grouped, compared, and analysed. The initial group is whittled down to smaller groups for analysis. The evolution of the methods, communications protocols and system architectures were charted from the late 1980s up until the modern-day. Various papers were discussed, grouped and analysed to provide a clear picture of the current research landscape of power quality monitoring. The literature review was divided into three separate categories; general power quality monitoring, IEC 61850-based power quality monitoring and power quality monitoring using smart grid technologies.

The first category dealt with general power quality monitoring topics, including data acquisition methods, digital signal processing and remote communications methods. The second category details the development of the IEC 61850 standard to accommodate for power quality monitoring as well as the modelling considerations when it comes to deploying IEC 61850-based system. Finally, the third category looks at the growing trend of "smartening" (extending the functionality and integrating with other applications within the electric grid) power quality monitoring and energy metering infrastructure according to the smart grid framework.

Within the literature review for general power quality monitoring, it is found that the general developmental trend of power quality monitoring has been the use of digital signal processing. Various communications technologies for remote metering, with Ethernet and wireless technologies, have also become prevalent communications methods of choice. The essential nature of performing long term studies to capture specific problem areas and trends is also seen as a critical driver in modern power quality monitoring.

Next, the role of the IEC 61850 standard in modelling and implementing power quality monitoring systems was analysed. The main takeaways are that the standard allows for the implementation of advanced power quality mitigation algorithms. Advanced algorithms may be modelled with standard power quality indices and then integrated within a wide range of auxiliary functions such as electric vehicle charging and fault localisation, isolation and restoration. This aspect of the IEC 61850 function and the interoperability requirement of compliant devices makes the standard suited for implementation within a distributed “smart grid”.

The adoption of bidirectional smart meters with power quality monitoring capabilities is becoming a popular way of conducting power quality studies on customer premises, and this is the mainline topic of the third literature study which focuses on the integration of smart grid technologies. Some key finding was the increasing use of open-source hardware and software, IoT technologies coming to the fore, and the increased risk of cyber-security vulnerabilities due to the “smartening” of electric power grids.

Chapter 3 presents a theoretical background of the several power quality issues affecting DERs and electric power grids as a whole. The theoretical background forms an essential step in determining the crucial power quality metrics to be monitored by the eventual power quality monitoring system. The power quality metrics will eventually be modelled in the development of the power quality measurement device.

CHAPTER THREE

THEORETICAL BACKGROUND OF POWER QUALITY IN DERS

3.1. Introduction

Decentralised (distributed) generation systems have distinct advantages over centralised systems; these include the potential for lower losses, a higher diversity of fuel types such as photovoltaics, fuel cells and wind; and finally, more options for economic dispatch. There are, however, several concerns that could potentially manifest themselves in terms of the complexity of distributed generation systems, monitoring and most importantly, power quality (Chambers, 2018).

This chapter explores the theoretical background of power quality in low-voltage networks with distributed generation plants. Chapter 2 presents the literature review exploring many papers and fields of research surrounding the topic of this research project. Specific issues are detailed to determine the trajectory and focus areas of the final implementation. Section 3.1 introduces the chapter, while in Section 3.2 different approaches to integrating DERs into the distribution network are examined, this influences the types of power quality issues that are manifested and ~~how~~ the roles of the consumer/producer and the Distribution System Operator (DSO). Thirdly, Section 3.3 explores the quality of supply issues typically experienced when integrating DERs into electrical networks based on the findings in the previous section. This assists in identifying the metrics to be measured by the completed proposed device. Finally, Section 3.4 provides a conclusion to this chapter.

3.2. DER integration approaches

As explained in the Introduction in Chapter 1, the challenge of integrating distributed generators into the traditional grid is a major one. Differing criteria for the capacity of installed power, where the point of connection to the distribution and transmission grids are located, and the types of technology used are some of the typical issues. One of the most significant issues, though, is how to deal with low power quality. There are two common approaches, pro-prosumer and pro-operator. The pro-prosumer system allows the distributed generation producer (the prosumer) to feed into the grid without any prerequisite conditions. According to the US Department of Energy, a “prosumer” is a consumer that is also an electricity generator or producer (EERE, 2017). This term has become more prevalent due to the rise of small-scale solar and wind energy generation. In contrast, the pro-operator approach puts the responsibility of control, regulation, stability and maintenance of grid code standards in the hands of the distribution network operator

(Sikorski & Rezmer, 2018). The following sections define and describe the two DER connection approaches, unconditional connection and network-operator controlled connections.

3.2.1. Unconditional connection

An unconditional connection forms part of the pro-prosumer approach to Distributed Energy Resource (DER) connections. It is based on the criteria for unconditional connection in the references for electromagnetic compatibility emission limits. The standards for connecting to the Low Voltage (LV) distribution network are based on traditional definitions of IEC Electro-Magnetic Compatibility (EMC) emission limits, which require standardised laboratory tests on the generation equipment to ensure they are eligible for specific certifications (IEC TC 77, 2009). This approach requires that the emission limits, as well as the tests and measurements for specific power quality parameters, should be defined (Sikorski & Rezmer, 2018). In essence, this means that distributed generation equipment should be treated like traditional loads without any special considerations. With the aforementioned in mind, the LV Distributed Generation (DG) loads need to adhere to the emission limits found in IEC 61000-3-2 and 3-12 (Harmonic current emissions), IEC 61000-3-3 and 3-11 (Voltage change and voltage fluctuation limits). This approach is preferable in systems where a small number of embedded/distributed generation units are present, as the overall effect of a disturbance in these plants is negligible. Problems could arise with a more considerable uptake as described in the introduction of Chapter 1. In this case, a more pro-operator or controlled approach is preferred.

3.2.2. Network operator-controlled

In a network operator-controlled system, the DSO is responsible for the reliability of the electric grid, this, in essence, gives the DSO authority to control and regulate the generation units (Abapour & Zare, 2018). Steady-state conditions give rise to different considerations compared to transient conditions. In steady-state, active and reactive power control is crucial to voltage regulation, whilst in transient conditions, under, and over-voltage/frequency regulation becomes necessary to preserve the integrity of the system (Hidayatullah et al., 2011). This type of system allows LV DGs to take on a network support role in addition to their generation roles. It also enables the DSO to control the operation of the DG units dynamically, making the grid more flexible, adaptable and offers an increased situational awareness as opposed to the unconditional connection system.

3.3. Quality of Supply in Network Operator controlled systems

With the quickening uptake of DG units, it can be safely concluded that the preferred approach should be the network operator-controlled model of management. As such, the quality of supply issues are examined with the assumption that DSOs have control over the operation of DG units. Such an assumption allows us to more closely look at the types of measurements needed for the regulation functions and studies performed by a DSO. This section details the various considerations that need to be made concerning the gathering of relevant information and monitoring essential aspects of the electric power grid's power quality. Three factors are examined in the following sections; static voltage stability, voltage variations and harmonic distortion.

3.3.1. Static voltage stability

Inverters are commonplace in the integration of DG systems into the LV electric grid. Inverter-based DG systems can assist in the stability of voltage in steady-state using active and reactive power control. They can be used to either maintain a specific power factor or operate under several power factors based on the amount of generated reactive power depending on the application or system characteristics (Sikorski & Rezmer, 2018).

In the case of a system with a constant power factor, the power factor can typically not be less than 0.9 or 0.95 depending on the apparent power of the generator. This type of approach is generally seen in photovoltaic systems designed to work with a power factor equal to 1. Both asynchronous and synchronous generators may also play a role in active and reactive power control. Asynchronous generators via capacitor banks used for reactive power control, while synchronous generators play a similar role in LV distribution networks. The production of active power plays a significant role in the characteristics of the power factor in a given generation unit, specifically those that are inverter-based. Although that is often the case, it is mainly dependant on what kind of power factor characteristics are desired by the DSO, and this is determined by analysing the current operating conditions at the Point of Common Connection (PCC). In reality, a standard power factor is often implemented and is mostly dependant on the maximum apparent power produced by the generation unit (Sikorski & Rezmer, 2018).

Figure 3-1 shows an example of a grid-tied inverter with active and reactive power control. The inverter consists of an IGBT based power converter denoted by the letter A, the output of the typically full-bridge inverter is connected to an LCL filter indicated by the letter B to lower attenuation.

The letter C represents the next phase which is the utility grid, which in practice is the Point of Common Connection (PCC). The output current and voltage readings are fed back into a grid-connected control module (represented by the letter D). Reference active and reactive power signals are compared with the readings provided from the grid to generate an appropriate pulse width modulation signal to control the switching action of the IGBT. This type of control is known as current-mode control.

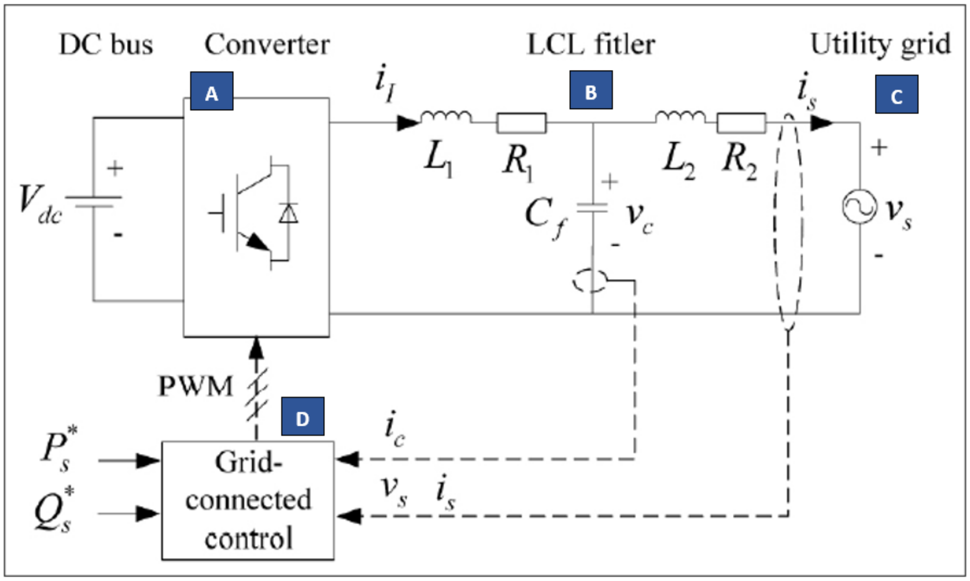


Figure 3-1: Active and Reactive Power Control (Peng et al., 2011)

The following equation can represent reactive power (Q):

$$Q = VI \sin \emptyset \tag{3.1}$$

Where:

V = Voltage

I = Current

θ = Phase angle

Since the formula for active power (P) is:

$$P = VI \tag{3.2}$$

Substituting equation (3.1) into (3.2) gives us:

$$Q = P \sin \emptyset \tag{3.3}$$

Using equation (3.3), the value of θ can be determined using the Q and P values. Furthermore, the θ value may be used in the control algorithm for active and reactive power control. Finally, since $\cos\theta$ represents the power factor, the equation can be further modified to determine the value of Q using the desired power factor:

$$Q = P \sin(\cos^{-1} \emptyset) \tag{3.4}$$

In Equation (3.4), the value of θ represents the power factor. With solar photovoltaic installations being one of the most popular types of DG systems, inverters are gaining prevalence, making the control of active and reactive power to maintain a stable voltage even more crucial.

3.3.2. Voltage variations

One of the most frequent causes of voltage fluctuations, especially rapid ones, are switching operations. The impact of specific operating conditions of the generator that could result in sudden voltage changes at the PCC can be estimated using the following equations:

$$U_{max} = K \frac{S_{Emax}}{S_{kPCC}} 100\% \tag{3.5}$$

$$K = \frac{I_{aE}}{I_{rE}} \tag{3.6}$$

Where:

S_{kPCC} = Short-circuit power at PCC

S_{Emax} = Maximum apparent power of the generator

I_{aE} = Inrush current of generator

I_{rE} = Generator’s rated continuous output current

In the EMC standards IEC 61000-3-11 and 3-12, the short-circuit power coefficient (denoted as R_k) is used to test emission limits; it is usually assumed that the value is not less than 33.3. Equation (3.5) can be re-written using R_k as follows:

$$U_{max} = K\frac{1}{R_k}100\% \tag{3.7}$$

R_k can be determined as the ratio of S_{kPCC} and S_{Emax}:

$$R_k = \frac{S_{kPCC}}{S_{Emax}} \tag{3.8}$$

Using Equations (3.5) to (3.8) allows for the determination of the severity of rapid voltage changes occurring within the system. Whether the DG installation adheres to the limits as specified in the IEC 61000-3-3 and 3-11 parts of the IEC 61000 standard can also be determined, and this can be incorporated into voltage sag and swell condition indices in a PQ monitoring system.

Additionally, long-term and short-term flicker severity indices may be used to evaluate voltage fluctuations. The same standards in the IEC 61000-3-3 and 3-11 parts may be used to investigate the voltage fluctuations, meaning the R_{kPCC} parameter as used in determining the severity of rapid voltage changes (Equation 3.8) may be used for long- and short-term flicker calculations as well.

3.3.3. Harmonic limits

The harmonic current limits are specified by the international standards the IEC 61000-3-2 and 3-12 parts. These EMC standards detail allowed harmonic currents at various input currents and may be used to determine the permitted limits for distributed generators (IEC TC 77, 2009). As with voltage fluctuations, the limits can be recalculated using the short-circuit power at the PCC (S_{kPCC}). The German Verband Deutscher Elektrotechniker VDE-AR-N 4105:2011-8 standard specifies the minimum technical requirements for connecting and operating power generation systems in low-voltage distribution networks. This standard sets the limits for current harmonics which can be expressed as A/MVA of S_{kPCC}. The value of A/MVA can be denoted as I_{vzul} and represent the absolute limits, which can be recalculated using the following formula:

$$i_{vzul} = \frac{I_{vzul}}{S_{kPCC}} \rightarrow I_{vzul} = i_{vzul} \cdot S_{kPCC} \tag{3.9}$$

Where:

I_{vzul} = current harmonic limits (A/MVA)

S_{kPCC} = Short-circuit power at PCC

The value of S_{kPCC} is directly proportional to the value of R_k (short circuit power coefficient), the same value as defined in the EMC standard. In theory, to increase the robustness of a given network, higher levels of S_{kPCC} may be introduced concerning the maximum apparent power at the generator (S_{Emax}). Higher levels of S_{kPCC} manifests as higher levels of R_k, which in practice means that more elevated amounts of current harmonic content may be permitted. The amount of harmonic content is a consideration that must be taken when deploying a DER into a low voltage network, along with the stipulated maximum amount of DC-injected current an inverter-based system may inject into a network as specified in the EN 50438 and IEEE 929 standards (Sikorski & Rezmer, 2018).

With regards to harmonics in voltage signals, the amount of distortion in the sinusoidal signal is dependent on the generation unit. A sinusoidal voltage signal is heavily influenced by the equipment, component sizing and types of loads on the supply side. It is still worth measuring and studying the harmonic quantity present in the voltage signal as this gives a good indication of whether the quality of supply is in adherence to the standards stipulated by the DSO. Measurement and analysis of both current and voltage harmonic content should ideally be done according to the IEC 61000-4-7 standard.

3.3.4. Other considerations

Several other power quality issues should be taken into consideration when integrating Distributed Generators into a low-voltage network. The following is a brief description of other common problems:

a) **Disconnection due to Protection Functions:** Disconnection of a distributed generator can happen for many reasons, but the most concerning should be disconnection due to a protection function. Several protection functions may cause the generation unit to be switched out of the network; Overload protection, over-frequency, under-frequency, over-voltage, under-voltage, anti-islanding and subsystem resynchronisation (Antonova et al., 2012).

b) **Medium/Long Term Voltage Changes:** The short circuit power in an upstream network influences the voltage level at the PCC. Reactive power generation or consumption is a crucial determinant of the voltage level of a system and as such, close attention must be paid to the types of loads being removed or added to the system (Dulău et al., 2014).

c) **Voltage Unbalance:** In Chapter 1 section 1.4.9, voltage unbalance was defined as the ratio of the negative or zero sequence component to the positive sequence components. In the case of distributed generators, single-phase DG units can cause negative sequence currents and asymmetrical voltages by being distributed in an unbalanced manner (Chua et al., 2011).

d) **Voltage Notches:** These are also described in Chapter 1, section 1.4.6. These types of disturbances are common in inverter-based systems. Due to the switching action of the power electronic devices, induced oscillations may be introduced into the voltage signal. They have the potential to heighten voltage peaks and interfere with consumer equipment (Sikorski & Rezmer, 2018).

3.4. Conclusion

DER connection methods often determine the monitoring approach employed by the Distribution System Operator. An unconditional connection approach allows connections to the distribution grid as long as the generation unit adheres to IEC EMC emission limits. Network operator-controlled systems give the DSO the authority to control and regulate generation units. The network operator-controlled system is the preferred approach as it allows the operator to monitor and mitigate power quality issues dynamically. Commonly experienced power quality issues in distributed generation units, specifically at the Point of Common Coupling (PCC), are described and investigated in this chapter.

Upon analysis, it can be concluded that monitoring the stability of the voltage in a steady-state, detecting and mitigating voltage fluctuations and variation events are the minimum requirements for effectively monitoring a distributed generation (DG) system. Moreover, finally monitoring the harmonic content on both the current and voltage systems are vital to ensuring that a DG system is operating within the allowable tolerances according to the IEC 61000-3-2 and 61000-3-12 parts and the DSO's specifications. Determining the necessary measurements allows for the modelling of power quality monitoring functions to develop a useful IEC 61850-based device model.

Based on the theoretical background provided in this chapter, the power quality measurements that shall be implemented as part of the device's calculation algorithms are as follows:

- Phase Current
- Phase to Ground Voltage
- Active Power

- Reactive Power
- Power Factor
- Voltage Dips
- Voltage Swells
- Current Harmonics
- Voltage Harmonics
- Frequency

Chapter 4 details the fundamentals of the IEC 61850 standard, then describes the procedure of selecting an appropriate digital signal processor or analogue front-end for data acquisition purposes. An IEC 61850 device model based on the power quality events and measurements listed in the above list shall be developed.

CHAPTER FOUR

IEC 61850 STANDARD OVERVIEW AND ANALOGUE FRONT-END DEVICE MODELLING

4.1. Introduction

The IEC 61850 standard describes a broad set of object-oriented data models; these are used to describe processes, equipment and functions that are implemented and controlled in a typical substation environment. Service models to describe the interactions between devices commonly found in substations, and to facilitate the transfer of data to and from IEDs are some of the available features. Data objects, known as logical nodes, are used to standardise IED data, and this enables the possibility of readily sharing information in a network, regardless of the implementation. (Ozansoy et al., 2009).

In this chapter, an IEC 61850-Based Power Quality Monitoring device is modelled according to the quality of supply needs identified in Chapter 3. From this point forward, each chapter begins with an overview of the system in block diagram form to indicate which module is dealt with within the chapter's content. The block diagram is based on the proposed system architecture described in Figure 2-8 but simplified to six "stages" instead of seven. Figure 4-1 visualises the specific module that is the focus area of this chapter marked in green. The device modelling detailed in this chapter forms part of the implementation of the IEC 61850 protocol stack, precisely defining the data model mapping aspect.

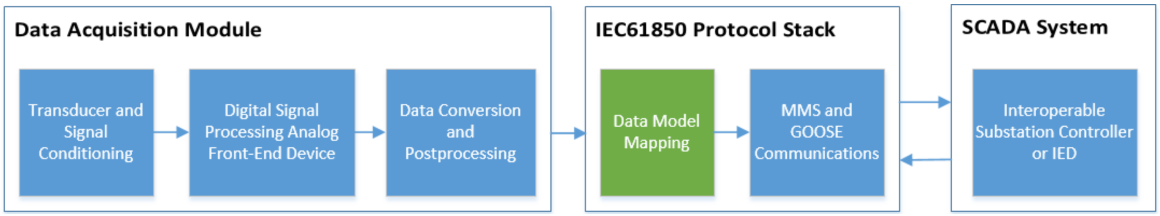

Data Acquisition Module
Transducer and Signal Conditioning
Digital Signal Processing Analog Front-End Device
Data Conversion and Postprocessing
IEC61850 Protocol Stack
Data Model Mapping
MMS and GOOSE Communications
SCADA System
Interoperable Substation Controller or IED

Figure 4-1: Chapter 4 Focus Area

The use of the IEC 61850 standard caters for interoperability between the device and OEM substation controllers as well as SCADA systems. The standard makes use of a configuration description language known as Substation Configuration Language (SCL) for writing configuration files as described in IEC 61850-6. SCL files are available with several file types, each serving a specific purpose in the engineering process, these SCL files promote interoperability by defining specific descriptors. All SCL files are written using the eXtensible Markup Language (XML), and all SCL files may be written and edited using the XML Marker software.

First, the fundamental concepts behind the IEC 61850 standard are introduced in section 4.2. Then the IEC 61850 information model as well as some general ideas contained in the standard, are briefly discussed in section 4.3. Several analogue front-end devices are compared and evaluated in section 4.4, and lastly, section 4.5 details the SCL object model and the types of SCL files along with their functionality.

4.2. The IEC 61850 standard

The IEC 61850 standard was developed primarily for addressing interoperable communication between devices from various vendors within substations. The end goal of its development was defined by several functions in the substation that needed to be addressed. These were: Protection, Control, Automation, Measurements and Power Quality Monitoring and Recording. In addition to the functions, several associated tasks needed to be supported by the standard, such as engineering, operations, commissioning, testing, maintenance, event analysis and security. Its development took place over a decade from the mid-’90s to the early 2000s and consisted of two primary tasks. Firstly, the development of Generic Object Models for Substation and Feeder Equipment (GOMSFE) which were under the auspices of the Utility Communications Architecture (UCA) and secondly, the actual International Electrotechnical Commission (IEC) 61850 project under IEC Technical Committee (TC) 57. The two activities were eventually combined due to the similarity of the work being executed and became integrated into the IEC TC 57 (Apostolov, 2007).

The IEC 61850 standard was developed based on providing technology independence, flexibility, and expandability. Edition one of the IEC 61850 standard consists of 14 documents spanning a wide range of concepts, that not only describes the data that should be communicated in a substation LAN but also what needs to be transmitted. A vital characteristic of the standard is data abstraction of substation equipment and functions. The 14 documents in the IEC 61850 standard are grouped in 10 separate parts:

Part 1: Introduction and Overview

Part 2: Glossary

Part 3: General Requirements

Part 4: System and Project Management

Part 5: Communications Requirements for Functions and Device Models

Part 6: Substation Automation System Configuration Language

Part 7: Basic Communication Structure for Substation and Feeder Equipment

Part 7-1: Principles and Models

Part 7-2: Abstract Communications Service Interface

Part 7-3: Common Data Classes

Part 7-4: Compatible Logical Node Classes and Data Classes

Part 8-1: Specific Communication Service Mapping (SCSM) – Mappings to MMS and to ISO/IEC 8802-3

Part 9: Process Bus Mapping

Part 9-1: Sampled values over serial unidirectional multi-drop point to point link

Part 9-2: Specific Communication Service Mapping (SCSM) – Sampled values over ISO/IEC 8802-3

Part 10: Conformance Testing

The standard allows for the development of distributed functions. Distributed functions require an exchange of data between two or more functional elements (located in a variety of physical locations) or different levels of the functional substation hierarchy. The IEC 61850 standard also defines the communication interfaces used between physical connections and even the functions of substation automation systems (SAS) present in protection, control, monitoring and recording equipment. Substation functions are distributed in three different levels defined within the functional substation hierarchy. Figure 4-2 illustrates the various levels or functional hierarchy and their respective logical interfaces defined in the IEC 61850 standard. In Figure 4-2, the levels are separated alphanumerically. Levels are denoted alphabetically, and the logical interfaces numerically. The following levels are defined in the substation hierarchy:

- A - Station Level
- B - Bay/Unit Level
- C - Process Level

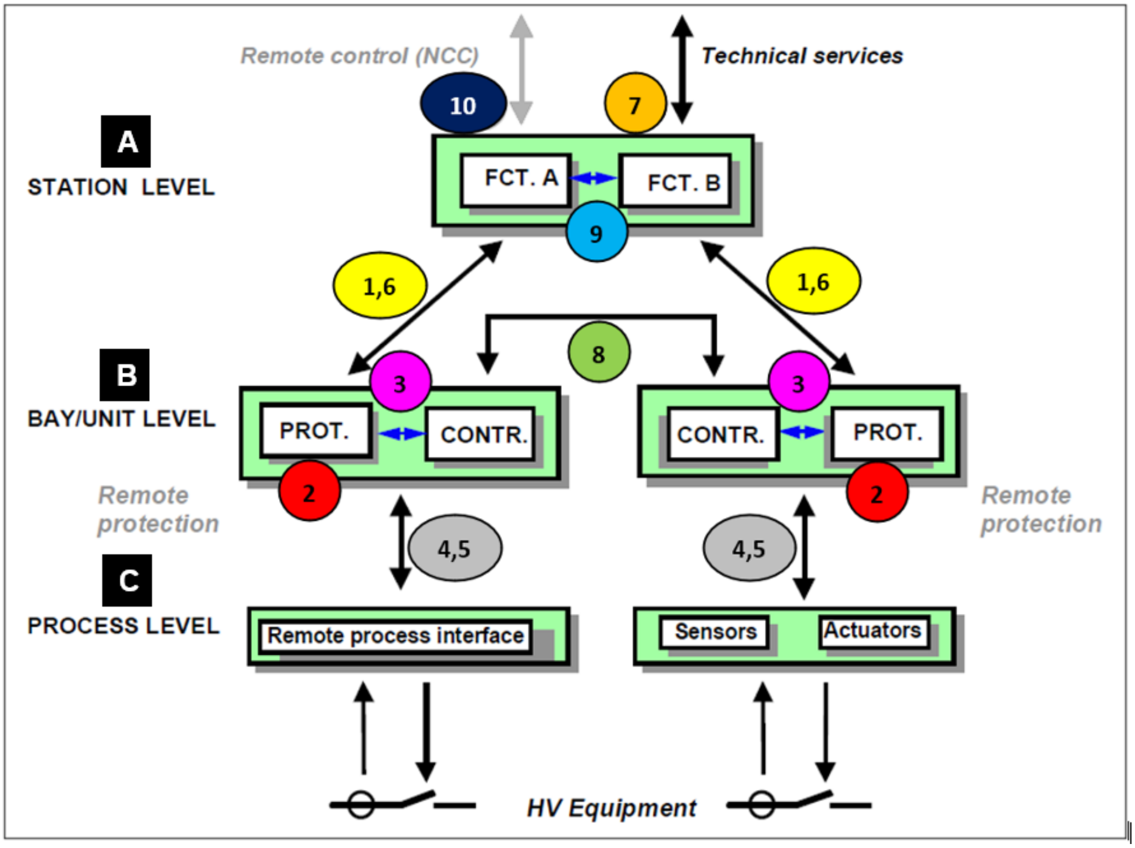


Figure 4-2: Levels and logical interfaces in Substation Automation Systems - IEC 61850-5 Page 15 (IEC, 2003)

The numerically denoted logical interfaces can be described as follows:

1. **Yellow:** Protection and data exchange between the bay and station level
2. **Red:** Protection and data exchange between the bay level and remote protection (outside the scope of this part of IEC 61850)
3. **Magenta:** Data exchange within the bay level
4. **Grey:** CT and VT instantaneous data exchange between the process and bay level
5. **Grey:** Control and data exchange between the process and bay level
6. **Yellow:** Control and data exchange between the bay and station level
7. **Orange:** Data exchange between the substation and a remote engineer
8. **Green:** Data exchange directly between the bays for fast functions such as interlocking
9. **Light Blue:** Data exchange within the station level
10. **Maroon:** Control-data exchange between substation (devices) and a remote-control centre (outside the scope of this part of IEC 61850).

Typically, the three levels, as described in Figure 4-2, are defined as follows:

Process Level: All of the functions that interface with the primary equipment in the substation (binary signals, analogue signals, binary controls and others) reside. Logical interfaces 4 and 5 all communications from the process level to the bay level.

Bay Level: Contains functions comprised primarily of data from a singular bay which mainly deal with the primary equipment of the bay mentioned above. Common examples include the circuit breakers, transformers, switches and various other related switchgear that may reside between the substation's bus and the transmission or distribution line.

Station Level: The station level contains functions that encompass the entirety of the station and covers the functioning of the various equipment contained within. These can be grouped between process-related station-level functions and Interface-related station-level functions. The differences being that process-related functions use the data of multiple bays or the entire substation to perform actions that affect numerous bays or station as the whole. Interface-related functions represent the various interfaces from the SAS to the HMI and finally to the SCADA, logical interfaces 1 and 6 represent bay level communications, and interface 7 illustrates communications with the outside world.

The functional hierarchy, as well as the description of various functions, provides a framework from which a data acquisition and control system can be designed and implemented in a structured manner. The IEC 61850 standard's use of distributed functions makes it a flexible and ideal vehicle for the implementation of remote communications and control systems.

4.3. The IEC 61850 Information Model

In the IEC 61850 standard, the real-world behaviour of primary and secondary substation equipment is virtualized; this concept is depicted in Figure 4-3. This approach is used to model common information found in substation devices; the standard defines all of the information available for exchange with other devices. In the IEC 61850 standard, common information refers to the information that vendors and users have agreed to make available for open exchange between devices. Abstract models are used to define information and information exchange. Abstract models, along with the use to virtualisation, allows engineers to represent aspects of real devices that are of interest in a particular implementation.

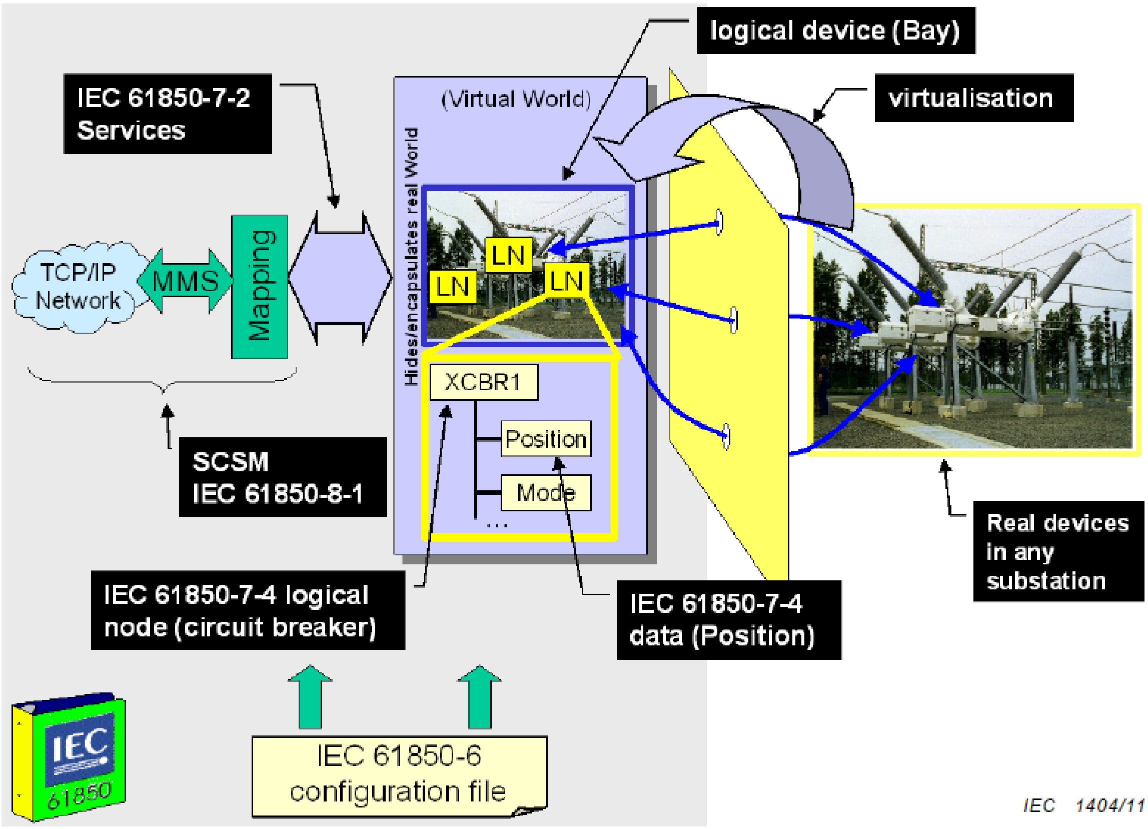


Figure 4-3: IEC 61850 Modelling - IEC 61850-7-1 Ed 2 page 15 (IEC, 2011)

IEC 61850-5 describes the decomposition of an application into its smallest entities. These entities are then used to exchange information and are known as Logical Nodes (LN). Using Figure 4-3 as an example, as defined in IEC 61850-7-4, XCBR1 (in the yellow box) is a logical node that describes the functionality of a circuit breaker with the Position and Mode as some of its data attributes.

A combination of logical nodes builds a logical device, which in the case of Figure 4-3 is the bay unit contained within a blue border. Logical devices may only exist on a specific IED; thus, logical devices may not include logical nodes from other IEDs. Logical Nodes contain a list of available data attributes ("Position" and "Mode" as in the case of XCBR1), which is exchanged using many applications protocols via the Specific Communication Service Mapping (SCSM) and Abstract Communications Service Interface (ACSI). Available application protocols include Manufacturing Message Specification (MMS), Transport Control Protocol/Internet Protocol (TCP/IP) and Ethernet.

Table 4-1: Logical Node Groups contained within the IEC 61850 standard

Group	Logical Node Groups
A	Automatic control
C	Supervisory control
D	DER (Distributed Energy Resources)
F	Functional blocks
G	Generic function references
H	Hydropower
I	Interfacing and archiving
K	Mechanical and non-electrical primary equipment
L	System logical nodes
M	Metering and measurement
P	Protection functions
Q	Power quality events detection related
R	Protection related functions
S	Supervision and monitoring
T	Instrument transformer and sensors
W	Wind power
X	Switchgear
Y	Power transformer and related functions
Z	Further (power system) equipment

During device setup, the specific logical devices and logical nodes need to be configured. The configuration is meant for the selection and assignment of instance-specific values, which in turn reference the instantiated logical nodes with the data exchange mechanisms. Logical node groups were developed to describe the different functions available on typical substation devices. As of the IEC 61850 standard Edition 2, 19 logical node groups are available with over 100 logical nodes available overall. Table 4-1 list the available logical node groups where each group is given a unique letter to quickly identify in which logical node group a specific logical node belongs. For example, XCBR1, the XCBR instance, as shown in Figure 4-2, belongs to the switchgear logical node group as it is prefixed with the letter “X”.

On closer examination of the logical node groups listed in Table 4-1, possible logical node groups appropriate for representing a power quality monitoring device for DERs are: D, M and Q. Logical Node Group D contains functions for Distributed Energy Resources, M handles metering and measurement functions whilst Q is for power quality events.

Logical nodes can provide several specified data objects. IEC 61850-7-4 describes the data objects provided by each specific logical node; each data object contains data attributes. Figure 4-4 is an example of a data model for the MMXU logical node. This data model represents a metering device capable of reading phase voltage (PhV), frequency (Hz), active power (W) and current (A), where these quantities represent the data objects. Data objects are divided into data classes as defined in part IEC 61850-7-3 of the IEC 61850 standard. In the case of the aforementioned data objects, all except Hz are part of the phase-to-ground/neutral related measured values of a three-phase systems (WYE) data class, whilst Hz forms part of the Measured Value (MV) data class.

The data classes determine the types of data attributes the data object contains. Each data object has several data attributes, as shown in Figure 4-4 where A (Data Object in the yellow box in the middle at the bottom) contains the phsA, phsB and phsC sub-data objects which represent the current measurements for each of the three phases. The phsA to phsC sub-data objects form a part of the Complex Measured Values (CMV) data class, which contains a large number of data attributes. It should be noted that Logical Node Zero (LLN0) is mandatory for all logical devices.

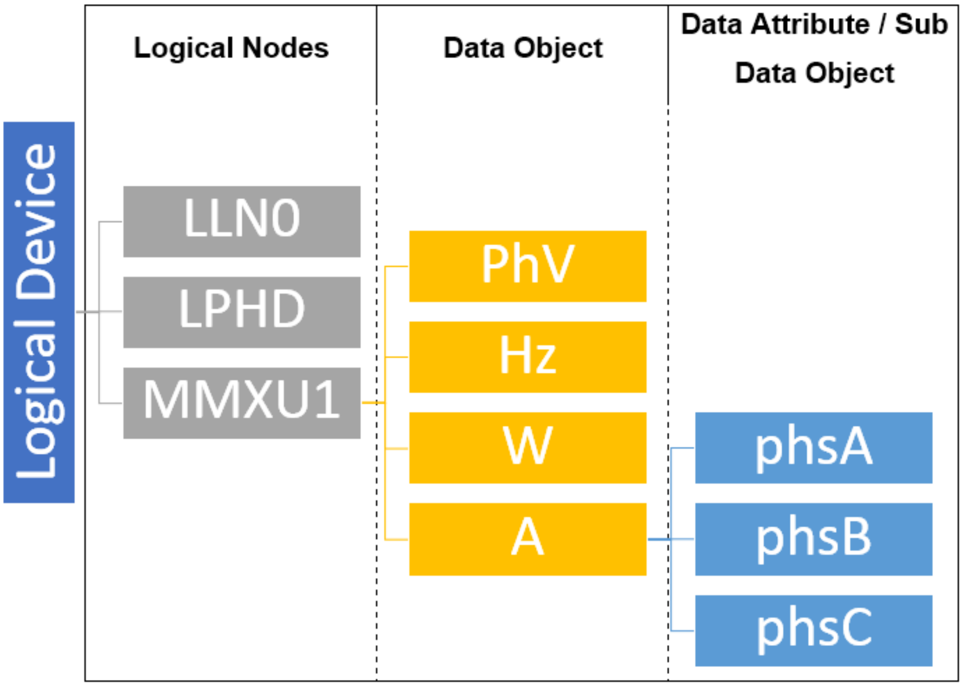


Figure 4-4: Example Data Model for MMXU

An introduction to the IEC 61850 information model was presented in this section. The concept of virtualisation of real-world equipment into logical devices and subsequently, logical nodes containing several data objects were explored. Then the available logical nodes were examined to determine the most suitable logical node groups for power quality monitoring logical nodes. Finally, the data modelling aspect was discussed in detail, with the focus on a logical device containing an instance of the MMXU logical node and its specific data objects and data attributes. In the next section, an analogue front-end device where instances of the relevant logical nodes and their required data objects are modelled according to measurement values available on the device.

4.4. Comparison of Analog Front-End device capabilities

Chapter 3 presents the standard power quality issues related to a distributed generator, while in Section 4.3, the modelling approach of IEC 61850 is described. The previously mentioned information makes it possible to model a device specifically designed for power quality monitoring for DGs. The modelling of functions depends on the capabilities of the device performing the measurements, as stipulated in the IEC 61850 standard. As such, the hardware choice is essential in determining the limitations and available power quality functions in the eventual implementation.

In the literature review study performed in Chapter 2, the use of DSP's as important enabling technology for low cost but highly functional metering devices are presented. As such, several Analogue Front-ends (AFE) and energy metering ICs were identified as potential data acquisition devices. An analogue front-end contains analogue signal conditioning circuitry comprising of analogue amplifiers, operational amplifiers, filters and specially built circuits. This aspect of the AFE provides a configurable interface for a variety of sensors, ADCs or microcontrollers. The AFE is ideal for this type of implementation due to their relatively low cost, low power consumption and flexibility within their specific function. Built-in computational algorithms and available on-chip meter data make analogue front ends ideal for designing and building metering systems.

Figure 4-5 shows a block diagram of a single voltage and current channel of a Microchip MCP39F511 power monitoring IC, which represents a typical AFE for energy metering. The process of acquiring, converting, calculating and transmitting the raw data happens in five distinct stages. Firstly, analogue signals are fed into the operational amplifiers (highlighted in Red), converted to digital signals via a 24-bit delta-sigma analogue-to-digital converter, which is contained within the blue rectangle.

Lastly, digital filtering is performed (shown in orange) before the quantities are calculated by the onboard calculation engine (highlighted in Green) and subsequently transmitted via the Universal Asynchronous Receiver/Transmitter (UART) serial interface or as status bits, demarcated by the yellow rectangle in Figure 4-5. A secondary computing unit may then process the transmitted data.

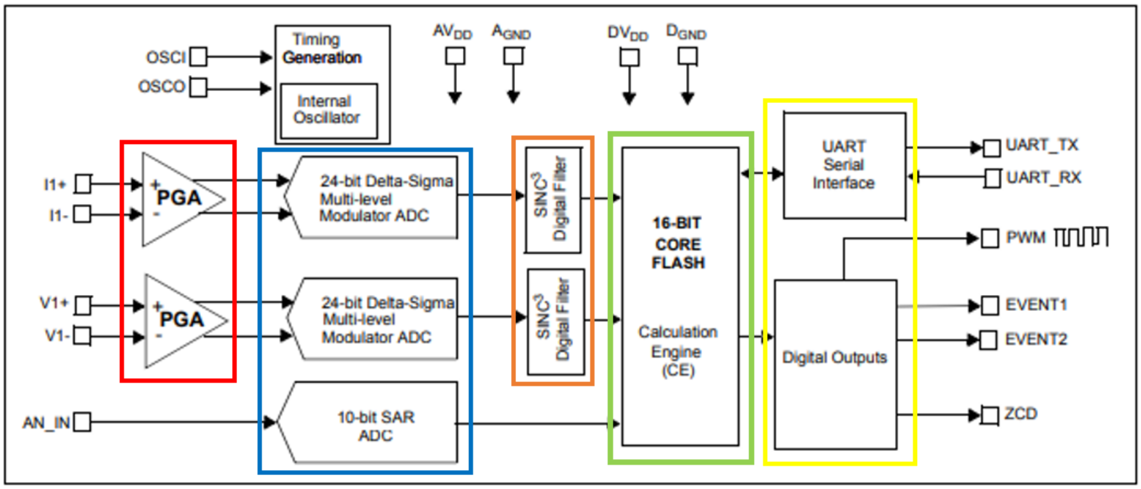


Figure 4-5: Typical Energy Metering AFE Block Diagram (Microchip, 2015)

Several specifications are considered for comparison purposes; these characteristics are based on the research and literature review in chapter 2. The specifications were mainly extracted from manufacturer data sheets and application guides.

a) **Single or Three-Phase Measurements:** The ability to measure three-phase systems is essential, and this provides the metering device with the flexibility to be deployed in various scenarios and network topologies.

b) **Metering Standards:** The metering standards of not only the power quality measurements but also of typically measured quantities need to be considered and eventually validated against the specified standard as defined.

c) **Available Power Quality Metrics:** Device should have predefined power quality measurements. Chapter 3, section 3.3, determined which measurements are of importance when dealing with distributed generation units.

d) **Communications Protocols:** Communications protocols available for extracting data to be processed and transmitted from the device. Most commonly used protocols are Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I2C), Universal Asynchronous Receiver Transceiver (UART) and Universal Synchronous Asynchronous Receiver Transceiver (USART).

e) **Integrated Microcontroller:** The inclusion of an integrated microcontroller is advantageous, but not critical.

Table 4-2 is the comparison chart used to select the appropriate hardware; where seven devices are shortlisted based on their functionality. Upon further analysis, it is noted that a higher presence of metering standards typically resulted in more power quality metrics being available; even when those standards, as mentioned earlier were not related to power quality measurements. This higher presence power quality metrics can be attributed to the presence of more sophisticated DSP algorithms which enable the device to compute more complex data and achieve higher performance levels.

IEC standards 62053-21,22,23 and 24 are most commonly found amongst all six devices, the IEC 62053 series of standards mainly deal with electrical metering along with the EN 50470 series. Compliance with standards mentioned allows for the implementation of power quality event monitoring such as dips, swells, harmonics, and overvoltage. Devices with more complex metrics such as total harmonic distortion were found to be compliant with standards more explicitly dealing with power quality such as IEC 61000-4-30 and IEC 61000-4-7. It should be noted that the IEC 61000 series of standards is the same as that presented in Chapter 3.

Table 4-2: Comparison of 7 Analog Front Ends and Energy Metering ICs

AFE / Metering Device	Phases Supported	Supported Metering Standards	Communications	Integrated MCU	Power Quality Metrics
STPMC1	Multiphase	• IEC 62052-11 • IEC 62053-21 • IEC 62053-23	• SPI	No	• Zero-Crossing Detection
ADE9000	Multiphase	• IEC 62053-21 • IEC 62053-22 • IEC 62053-23 • IEC 62053-24 • EN 50470-3 • OIML R46 • ANSI C12.20 • IEC 61000-4-30 • IEC 61000-4-7	• SPI	No	• Zero-Crossing Detection • Phase Sequence • Phase Angle • Voltage Sag • Voltage Swell • Peak Detection • Overcurrent • Total Harmonic Distortion • Power Factor
ADE9153A	Single-phase	• IEC 62053-21 • IEC 62053-22 • IEC 62053-23 • IEC 62053-24 • EN 50470-3 • OIML R46 • ANSI C12.20	• SPI • UART	No	• Zero-Crossing Detection • Voltage Sag • Voltage Swell • Phase Angle Measurement • Overcurrent • Peak Detection • Power Factor
ADE7880	Multiphase	• IEC 62053-21 • IEC 62053-22 • IEC 62053-23 • IEC 62053-24 • EN 50470-1	• I2C • SPI (with High-Speed Data Capture Interface)	No	• Zero-Crossing Detection • Phase Sequence • Time Interval Between Phases • Phase Voltage Sag • Period Measurement

		• EN 50470-3 • ANSI C12.20 • IEEE1459 • IEC 61000-4-7			• Peak Detection • Overvoltage • Overcurrent • Total Harmonic Distortion • Various harmonic calculations • Power Factor
MCP39F511	Single-phase	N/A	• UART	No	• Overpower • Overcurrent • Voltage Swell • Voltage Dip
ATSAM4CMS32	Multiphase	• ANSI C12.20 • IEC 62053-22	• USART • UART • SPI • I2C	Yes	• None built-in, but can be implemented using the Atmel Metrology Library
ATM90E32AS	Multiphase	• IEC 62052-11 • IEC 62053-22 • IEC 62053-23 • ANSI C12.20 • ANSI C12.21	• SPI	No	• Voltage Dip • Overvoltage • Phase loss • Overcurrent • Frequency • Zero-crossing detection • Phase sequence error

Ultimately, the Analog Devices manufactured ADE9000 (highlighted in green in Table 4-2) was selected as the most appropriate AFE. The deciding factor was the device's ability to implement IEC 61000-4-30 as well as IEC 61000-4-7, as none of the other devices had similar capabilities. IEC 61000-4-30 provides a standard for measurement and interpretation of results, whilst IEC 61000-4-7 provides methods for harmonic analysis. In simple terms, this means that the quantities measured by the ADE9000 are in line with the standards required to determine values of the dip level, swell level, total harmonic distortion accurately. In systems with sensitive equipment, obtaining reliably accurate data with standard measurement techniques is vital for maintaining the quality of supply.

4.5. Modelling of functions in the ADE9000 AFE

The first step of the modelling process is to extract the capabilities of the device. The ADE9000 datasheet and user guide indicate the functions available on the ADE9000 that can be accessed via an internal data register. Data on the ADE9000 can be transmitted to a secondary device via the SPI interface functioning as an SPI slave, meaning it never initiates communications. 16-bit and 32-bit read/write operations are available. Figure 4-6 illustrates the format of the 16-bit command header (CMD_HDR) used for reading or writing data from the device. Each register has a specific address that is specified in the ADDR[11:0] portion of the 16-bit message; this comes with an R/W bit that's set to 1 for read and 0 for write. The data within the register is then be mapped to standard logical nodes as per IEC 61850-7-4, this is an essential step as it determines whether vital events/measurements need to be obtained from a second distributed device or can be extracted directly from the device.

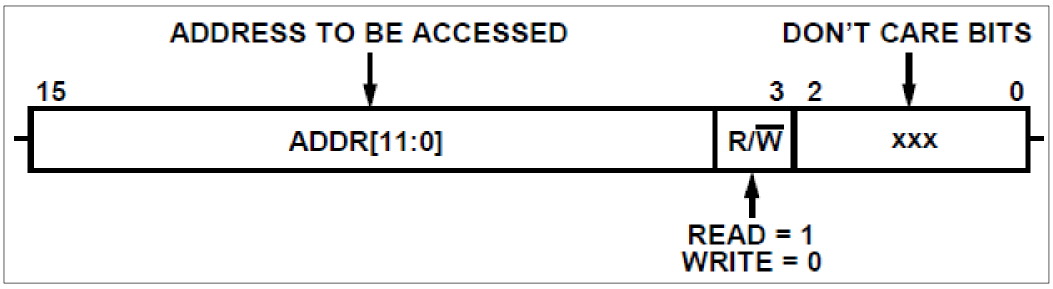


Figure 4-6: Command Header

Table 4-3 shows the data registers available on the ADE9000 along with their respective IEC 61850 logical node mappings. All measurements and events can be extracted per phase on the ADE9000, and each register name is suffixed or prefixed with "x", in place of the phase (A, B or C). Voltage Dip and Swell event starts are represented by bits in register address 0x403 (STATUS1). STATUS1 is a 32-bit field containing status bits, the Dip and

Swell status bits for phases A to C are stored in bits 20 to 25, the bit changing to 1 indicates a voltage that has entered into a dip or swell state, whilst 0 indicates an exit from a dip or swell state. The remainder of the registers carries measured data as 32-bit values instead of status bit fields. Logical nodes QVVR (Voltage Variation Events), MHAI (Harmonics and Interharmonics) and MMXU (Measurement) are mapped to Dip/Swell, Voltage/Current Total Harmonic Distortion (THD) and Voltage/Current/Power Factor measurements respectively, via the appropriate data objects. These logical nodes are now part of the IEC 61850 Edition 2 release of the standard; therefore, no new logical nodes need to be defined in the development of the proposed Power Quality device, as all of the required functions are available.

Table 4-3: Selected ADE9000 Registers and their Equivalent Logical Nodes

Measurement / Event	**ADE9000 Register Name and Address**	**Equivalent Logical Node**	**Data Object**
Voltage Dip Start	STATUS1/DIPx	QVVR	DipStr
Voltage Dip Level	DIP_LVL	QVVR	DipStrVal
Voltage Swell Start	STATUS1/SWELLx	QVVR	SwlStr
Voltage Swell Level	SWL_LVL	QVVR	SwlStrVal
Voltage THD	xVTHD	MHAI	ThdPhv
Current THD	xITHD	MHAI	ThdA
Phase to GND Voltage	xVRMS	MMXU	PhA
Phase Current	xIRMS	MMXU	A
Phase Active Power	xWATT	MMXU	W
Phase Reactive Power	xVAR	MMXU	VAr
Power Factor	xPF	MMXU	TotPF
Frequency	xPERIOD (calculated)	MMXU	Hz

The data registers mapped with their equivalent logical nodes, can be represented as a data model, as shown in Figure 4-7, which illustrates the data model of the New Power Quality Device which is developed in this research, extrapolated from the mappings in Table 4-3. On the input side, a CT and VT provide the raw analogue data to be processed by the ADE9000 to produce status and measurement data. The status and measurement data are extracted via the selected registers to create a logical device containing the QVVR, MMXU and MHAI logical nodes.

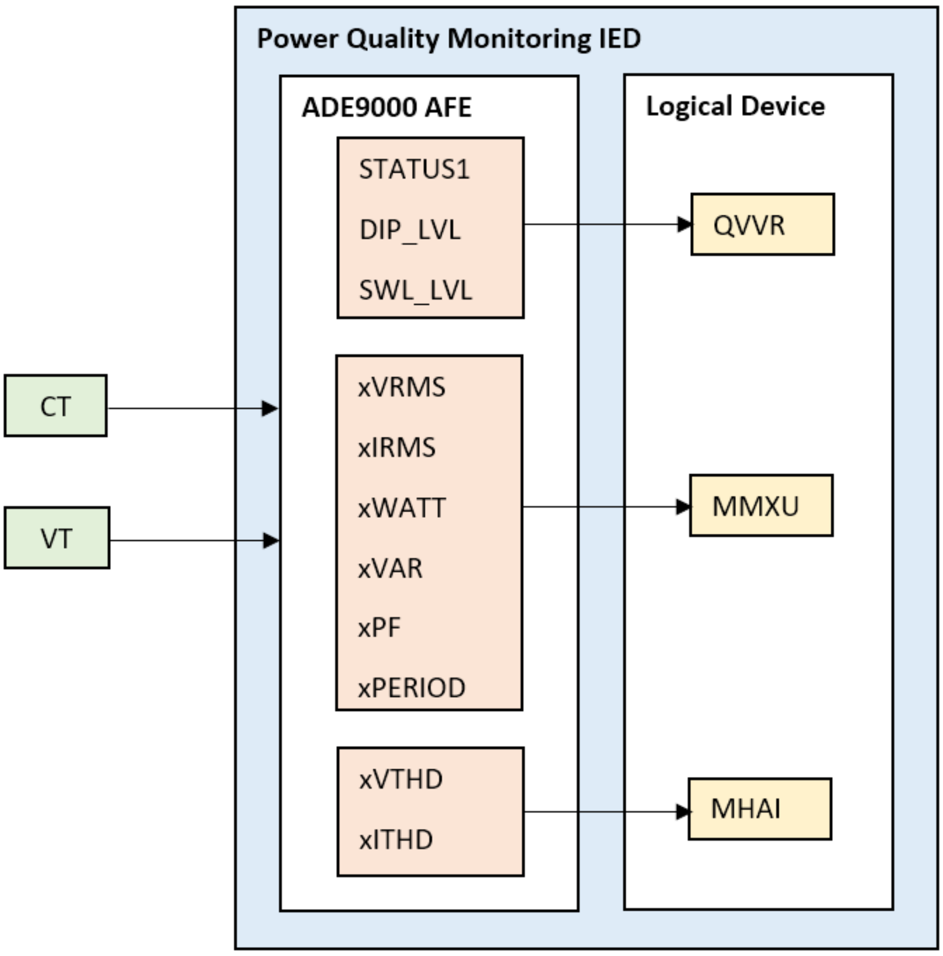


Figure 4-7: Resultant Data Model of the New Power Quality Device

QVVR contains data objects for voltage variations, as stated in Chapter 2, that are the most common power quality events which can significantly impact sensitive loads. The data objects within QVVR are mapped to the ADE9000 data registers. The ADE9000 uses RMS ½ values measured on the voltage channels to determine whether a dip or swell event has taken place. The DIP_LVL register determines the threshold for the dip condition depending on the number of half-cycles configured in the DIP_CYC register. DIP_LVL can be directly mapped to DipStrVal or IntrStrVal data objects present in QVVR. SWELL_LVL performs the same function as DIP_LVL, only for the swell condition instead of dip. This register is directly mapped to SwlStrVal and IntrStrVal. The voltage level dropping below the DIP_LVL register detects the corresponding DIPA, DIPB and DIPC bits set in the STATUS1 register (Analog Devices Inc., 2017a).

Similarly, when the voltage level spikes about the pre-set value in SWELL_LVL, the corresponding SWELLA, SWELLB and SWELLC bits in STATUS1 is set to 1. These register values are mapped to QVVR through the DipStr, SwlStr and VarStr respectively since there are three distinct input values for each voltage variation type. The results from the ADE9000 require an OR logic function to map to the correct data object, as shown in Figure 4-8. Additionally, the value recorded during the variation event is stored in separate registers (DIPA, DIPB, DIPC, SWELLA, SWELLB and SWELLC); these can be extracted for disturbance recording purposes.

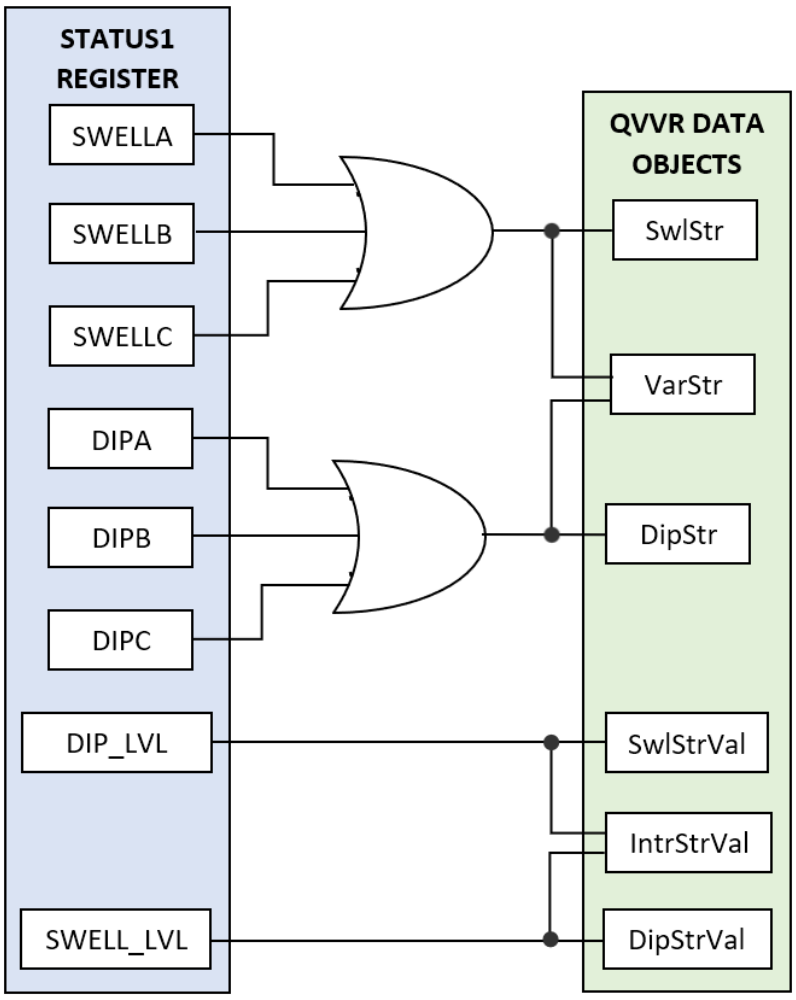


Figure 4-8: STAUS1 Register Mappings to QVVR Data Object

The ADE9000 register mappings to the IEC 61850-7-4 logical nodes form a building block for a complete information model. A fully realised information model requires a communication model, a substation model and an IED model. The mappings, as determined in this chapter, are a part of the IED model, where all three of the models mentioned above, are configured as part of a substation configuration language (SCL) file.

This section briefly described the data transfer functions of the ADE9000 and modelled its registers with equivalent IEC 61850 data objects. The next subsection describes the different configurations files used to model an IEC 61850.

4.6. Substation Configuration Language Object Model

The Substation configuration language (SCL) is a file format that describes the communications related to IED configurations, parameters, communications system configurations, substation functions and the relationships between them. It was developed to simplify the exchange of IED capability descriptions between engineering tools of different manufacturers whilst maintaining compatibility. The SCL is built upon the SCL object model, which consists of three parts: substation model, IED model and the communication model. Figure 4-9 shows the complete SCL data model.

The substation model (green in Figure 4-9) provides descriptions of the equipment in the switchyard or substation and the connections within the substations. A hierarchal structure of objects characterises the substation model, comprising of several elements and sub-elements such as a substation, voltage level, bay, equipment and sub-equipment. In a typical model, a logical node for a switch controller is attached to a switching device, a logical node to perform measurements and links to the bay for measurands and a transformer has its specific logical nodes. This particular model mostly forgoes the substation modelling comprising of bays and voltage levels due to the singular device being utilised.

The IED model (orange rectangle in Figure 4-9), also known as the product model, consists of the hardware and software that implements the functions available in the substation. In terms of scope, the function implementation is limited to the capabilities of the hardware device. Processes may only be modelled based on the functions available on the device. The IED model is the portion of the SCL object model in which the logical device and logical nodes along with their data objects are defined. In addition to the logical device and logical nodes, and IED, IED server is also specified in the IED model. In terms of hierarchy, the IED model follows accordingly:

1. **IED:** A substation automation device that communicates with other IEDs via a communications system.
2. **Server:** The server is an entity within the device that acts as an access point for communications with other devices.
3. **Logical Device:** As defined in 4.2 and IEC 61850-7-2, exists within a server.
4. **Logical Node:** As defined in chapter 2. Contains various DO which is described in the SCL file.
5. **Data Object:** The data within the logical node.

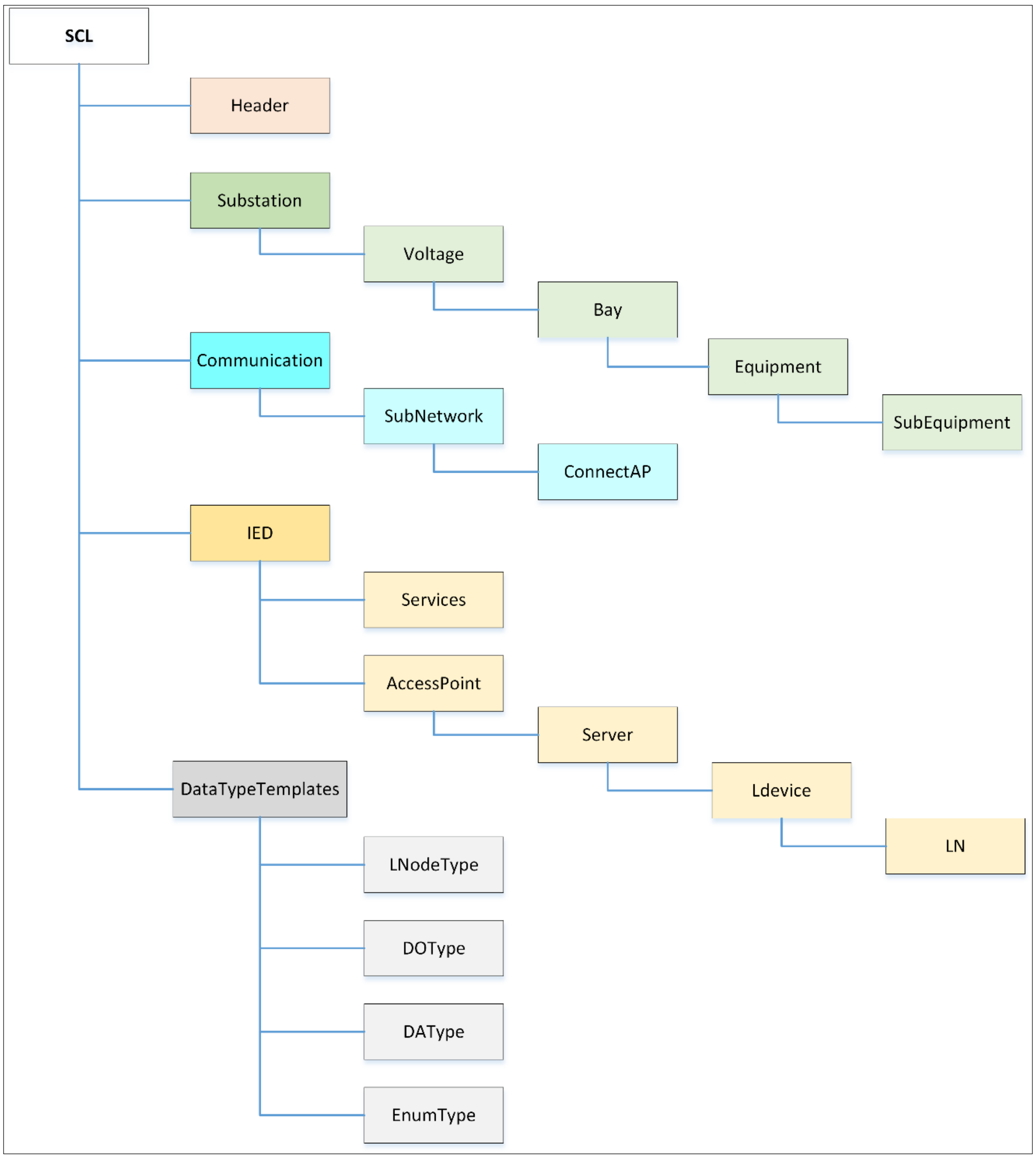


Figure 4-9: The SCL Data Model (Huo & Zhang, 2008)

As opposed to the previous two models, the communications model (blue in Figure 4-9) is not hierarchal. Access points form the logical connections between IEDs across subnetworks; subnetworks act as a connection node between access points and not a physical structure per se. Within the subnetwork, each "ConnectedAP" contains some parameters related to MMS client/server and GSE control block connections (which is discussed in the next chapter). The logical device or clients within an IED connects to subnetworks using an access point, which could be a physical port or server or an IED. Subnetworks are in practice logical connections but can be correlated to physical networks by way of naming subnetworks and access points according to existing networking infrastructure and physical connection points.

Finally, the DataTypeTemplate is not a model on its own but is common among all SCL files. This section typically contains logical node (LNodeType), data object (DOType), data attribute (DAType) and enumerator (EnumType) type definitions and templates.

The SCL model enables the substation automation engineer to design a specific substation automation system (SAS) in a logical and standardised fashion. A variety of SCL file types were developed to meet the detailed engineering needs to achieve flexibility while maintaining conformity within the overall model. The following subsection looks at the differences between various SCL file types to determine the most appropriate SCL file for the project configuration.

4.7. SCL file types and schema

The SCL object model or part thereof can only be implemented in a limited number of SCL file types. As of Edition 2, Six SCL file types are available and are defined in IEC 61850-6, each serving a specific purpose for configuration and parameterisation. Table 4-4 lists the types of SCL files currently in existence as well as their general functions and rules. Each file type serves a specific purpose in the engineering of IEC 61850 compliant substation automation systems. For implementation purposes, an ICD file is generated for the device configuration of the power quality monitor. The implementation is based on the functions and rules of the ICD file type shown in Table 4-4.

Table 4-4: SCL File Types (IEC, 2012)

File Type	**Abbreviation / File Extension**	**Functions and Rules**
IED Capability Description	ICD	• An ICD file describes the capabilities of an IED. • Only one IED section present and described. • IED is named TEMPLATE. • Data type templates and logical node type definitions are present. • Substation section is optional • Optional communications section may be present • File extension: .ICD
Instantiated IED Description	IID	• Meant for single IED preconfigured for a project

		• An IID file contains pre-configured project-specific names, addresses and data model with data set definitions. • They are typically used for IEDs with LN instances dependant on project-specific single line diagrams or other IEDs. • Optional data set and control block definitions identical to ones present in the system tool. • File extension: IID
System Specification Description	SSD	• Description of a single line diagram of a substation as well as the required logical nodes • The substation description section is mandatory as well as data type templates and logical node type definitions. • File extension: SSD
Substation Configuration Description	SCD	• Contains IEDs, a communications configuration section and a substation description. • File extension: SCD
Configured IED Description	CID	• An Instantiated IED is described. • The current address of the IED is contained within the communication section. • The substation section is not mandatory but may contain name values that are specific to the project. • It is essentially a less comprehensive SCD file that specifies what the IED should know. • File extension: CID
System Exchange Description	SED	Describes the interfacing between two projects—one to be used by the other. A subset of the SCD file and as such contains the interfacing parts of the IEDs, and the connection between projects are engineered. Engineering right and ownership of each IED is stated. File extension: SED

SCL files are based on the eXtensible Mark-up Language, more commonly known as XML. The syntax of an XML file is defined by the schema being used, for SCL the W3C XML schema is utilised. The Unified Modelling Language (UML) can be used to represent the relationships between SCL elements, as this only shows the relationships between the elements instead of the objects that are represented by the elements. Figure 4-10 illustrates the relationships between the SCL elements in the Unified Model Language (UML) form. Each rectangular box represents a schema type, which is identified by a lowercase t. In summary, the following can be derived from Figure 4-10:

- The basic SCL element can be derived from the tBaseElement schema type.
- One header element of type tHeader is contained within the SCL element.
- One or several substation elements of type tSubstation.
- A communication section of type tCommunication.
- One or several IED elements of type tIED.
- A DataTypeTemplates section of type tDataTypeTemplates.

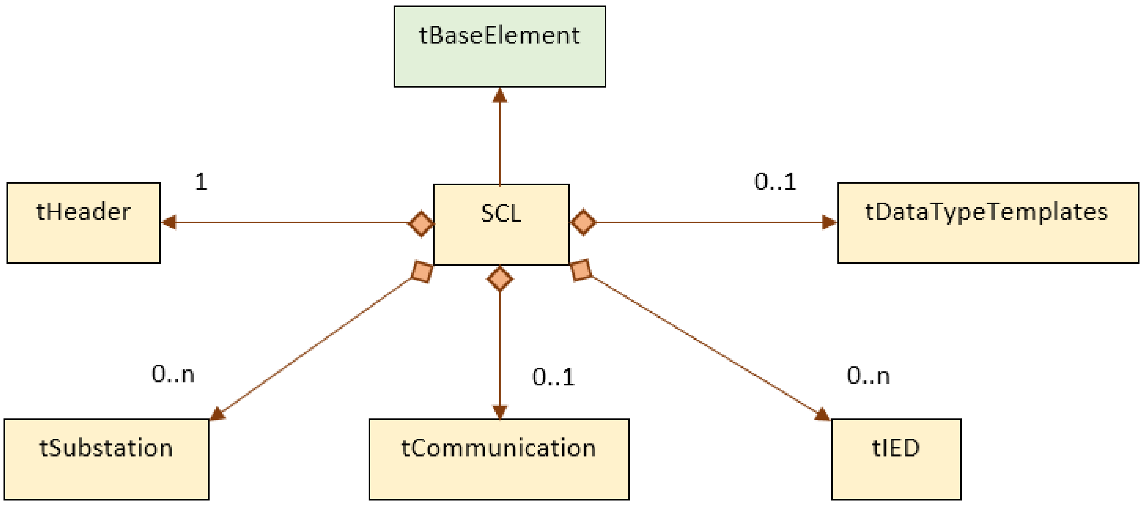


Figure 4-10: UML Diagram of SCL Schema

The SCL elements are crucial for successfully implementing an appropriate SCL file for the project. In this chapter, relevant information regarding the functions of the analogue front end along with how to appropriately map them to IEC 61850 logical nodes was determined. The SCL Object model was explored to determine the procedure for configuring device functionality and communications. Conclusively, it can be stated that an ICD file needs to be generated since the IED capabilities have been determined in this chapter and an appropriate data model formulated.

4.8. Conclusion

This chapter explored in detail, the various modelling requirements of an IEC 61850-based system, then considered the specific design considerations with regards to a Power Quality Monitoring system, including the selection and conditions of a data acquisition Analogue Front-end (AFE). In Chapter 4, the critical power quality metrics and events are identified and subsequently modelled in this chapter according to available IEC 61850 logical nodes as of IEC 618650 Edition 2. Once the required logical nodes are identified and modelled, a data model matching the ADE9000's data registers with the logical nodes' data attributes can be developed. Finally, the object model of Substation Configuration Language (SCL) Files is briefly defined to determine the required SCL file type for the substation configuration.

Determination of the various models provides a necessary framework for implementation. Once modelling is completed, the implementation process may begin. Chapter 5 details the implementation of the IEC 61850 standard on low-power, low-cost and open-source hardware and software platform. An open-source IEC 61850 standard is used for the implementation, and a standard ICD file based on the device model developed in this chapter is designed.

CHAPTER FIVE

IMPLEMENTING AND CONFIGURING THE IEC 61850 STANDARD ON A SINGLE-BOARD COMPUTER

5.1. Introduction

The ADE9000 AFE on its own cannot implement the software needed to allow for IEC61850 communications are well as the successful data acquisition and processing. As such, a secondary piece of hardware needs to be utilised for higher-level processing, data conversion and data transfer within a more comprehensive communications system. As usual, Figure 5-1 illustrates the focus area of the chapter, highlighted in green. The main focus is the implementation of the MMS and GOOSE communications within the IEC 61850 protocol stack. Firstly, this chapter explores the various types of software licensing models and determine which is the most appropriate for this project, and then a software package is selected. Once the software package is chosen, a compatible hardware platform is found, and the software library implemented and tested. With an IEC61850 compliant device obtained, the ADE9000 specific functions can then be modelled by writing an appropriate SCL file, then generating a model for the device functions.

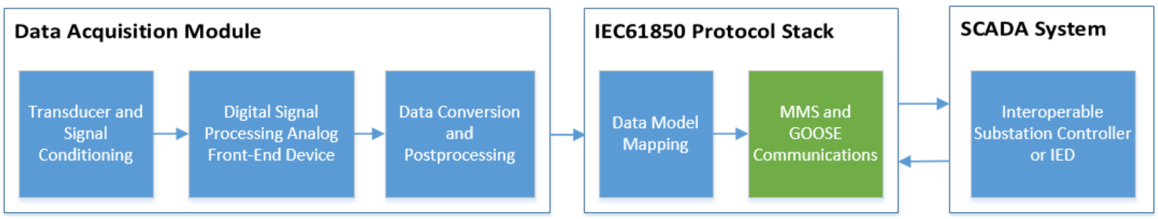

Data Acquisition Module
Transducer and Signal Conditioning
Digital Signal Processing Analog Front-End Device
Data Conversion and Postprocessing
IEC61850 Protocol Stack
Data Model Mapping
MMS and GOOSE Communications
SCADA System
Interoperable Substation Controller or IED

Figure 5-1: Chapter 5 Focus Area

This chapter explores the various considerations, design decisions relating to the implementation and configuration of IEC 61850. Section 5.2 discusses the software and hardware requirements of the computing device, and Section 5.3 provides an overview of the software and hardware to be used in the final implementation. In section 5.4, the building and initial testing of the IEC 61850 library are detailed and finally, sections 5.5 and 5.6 gives an in-depth explanation of the SCL file generated for the project as well as the procedure for converting from XML to C code.

5.2. Software and hardware requirements

Implementing a system based on the IEC 61850 standard requires a device with a pre-loaded license or if that is not available, the purchase of a license for a third party IEC61850 compliant protocol stack or an open-source alternative

The pros and cons of each of the options above are quite elementary. A device with built-in capabilities and pre-loaded license means less engineering work is required, as the software and hardware are very well optimised with high levels of reliability and warranties readily available. The most significant and apparent negative is that the cost is typically a lot higher, which is of concern, especially if the intention is to eventually upscale deployment to hundreds of DG sites. The high prices would make the capital investment prohibitively high. One final factor to consider is that the capabilities of such devices are typically quite wide-ranging, which is not ideal for a project with as narrow a scope as this one.

The purchase of a compliant protocol stack typically means that access to support and documentation is relatively simple and most software with a paid license comes with a list of compatible devices, which is useful for hardware selection and troubleshooting. A massive drawback is the pricing and licensing models often put vital features such as GOOSE messaging and Sampled Values behind paywalls. These are sought after features in protection and other systems with high-performance requirements, so it could end up becoming more expensive for a more fully-featured software package.

Lastly, an open-source software package can be considered. A great advantage of open-source software is the fact that the GNU's General Public License (GPL) allows users to use, share and modify the specified software freely. A user that is well versed in the programming language used to write the software package could make their modifications where they deem appropriate. There is also the community-based approach of most open-source software, where users often assist and advise each other on relevant issues. A massive drawback of open-source software is the security aspect, as open-source software is vulnerable to malicious code if intercepted, so care must be taken to acquire it from trusted sources. Another disadvantage is often the lack of any detailed documentation or warranty, and a large amount of time is usually spent trouble-shooting when software bugs are encountered.

Ultimately, the use of an open-source software license makes the most sense since the device that is interfaced with the ADE9000 needs to be compatible with SPI communications and be low-cost as well. Devices that most commonly have those characteristics are Advanced RISC Machines (ARM)-based computer boards or Microcontroller Units. Between ARM-Based devices and Microcontroller Units (MCU), ARM-based computers most commonly use Linux as an operating system, whereas MCUs are purely embedded software system. As such, a combination of an open-source library with an ARM-based computer is ideal.

5.3. Open-Source software and ARM hardware overview

In terms of open-source IEC61850 software libraries, the two most fully featured and well known are OpenIEC61850 by Beanit GmbH and LibIEC61850 by MZ Automation. Both software libraries are aimed at DER and inverter control, as such, are both similarly featured, containing services necessary for implementing Data set services, buffered and unbuffered reporting, control service, getDataValues and setDataValues.

The key differentiating feature is the programming languages they are written in; OpenIEC61850 is written in Java whilst LibIEC61850 is written in C. The fact that LibIEC61850 is written C means that it is more suited for lower-end, resource-constrained hardware such as mobile computing platforms and ARM System-on-Chips (SoC). The C programming is a low-level, procedural, compiled language as compared to Java which is high-level, object-oriented and interpreted. In terms of speed and code efficiency. The C language has the edge which makes it ideal for this project, as the intention is to utilise low-cost, low power hardware, and therefore the LibIEC61850 software was chosen.

Some of the outstanding features of LibIEC61850 software as obtained from their official website are as follows:

- Full TCP/IP based ISO protocol stack
- Static IED model generation of C code from SCL file
- SCL file to IED model conversion tool
- Read and write services for MMS variables
- Data set services
- IEC 61850 client and server application programming interface (API)
- MMS client API
- Buffered and unbuffered reporting service
- GOOSE publisher and subscriber capabilities
- IEC 61850-9-2 Sampled Values publisher and subscriber
- UCA edition 1 and 2 certifications ready
- Implementable on Linux, WIN32, Mac OS X and FreeBSD
- Hardware abstraction layer (HAL) allows for compilation on x86 and ARM-based systems (Raspberry Pi Foundation, 2019)

With the desired software library chosen, the next task is to select an appropriate hardware platform to implement the software. As previously mentioned, small size, low-cost, Ethernet capabilities and the ability to interface with the ADE9000 via SPI communications are primary requirements, with an added condition of running Linux, WIN32, Mac OS X or

FreeBSD operating systems. With the provisions mentioned above in mind, the primary focus was cast on Advanced RISC Machine (ARM) based development boards with Linux compatibility. These types of development boards form an ideal platform for prototyping, mainly due to the versatile, standardised nature of the ARM hardware architecture and the mature development of open-source Linux distributions compiled to run on ARM-based hardware. ARM's standard instruction set and workflow allow for greater interoperability between implementations of the same system, hence the ease at which various Linux distributions can be ported and compiled on ARM-based systems. Additional advantages to prototyping on an ARM device is the relatively low cost and low power consumption of devices. ARM technology is most commonly used on System-on-Chips (SoC), which often consist of integrated circuitry that houses all of the components found in computers. For example, a single chip may house the central processing unit, random access memory chips, audio DACs, graphical processing unit and communications modules.

Based on the requirements discussed, several ARM-based, Linux compatible development boards were identified as potential candidates. Ultimately, with a plethora of relatively similar development boards available on the market in terms of features, the most significant deciding factors became initial cost, SPI communications and Ethernet capability. The search was further narrowed down based on the tested platforms specified on the LibIEC61850 website, these being the BeagleBone Black and the Raspberry Pi (Pictured in Figure 5-2). On paper, both devices have similar characteristics, with key differences being the existence of a floating-point accelerator and two built-in 32-bit microcontrollers on the BeagleBone. The Raspberry Pi has neither of the mentioned features, but it boasts more RAM (up to 4GB vs 512MB on the BeagleBone), Gigabit Ethernet and a much faster Quad-Core A72 64-bit Cortex SoC (Long, 2019).

Overall, the BeagleBone is more suited for engineering work, whilst the Raspberry Pi has a broader range of applications. Still, in terms of the needs of this project, the additional microcontrollers and floating-point accelerator are not a massive benefit due to its communications-focused nature. As such, the Raspberry Pi was selected, primarily for the inclusion of Gigabit Ethernet and lower price point. Gigabit Ethernet speeds are advantageous for interfacing with industrial-grade, IEC 61850-compliant data network switches that require lower latency and higher speeds for GOOSE messaging.

Figure 5-2 shows the Raspberry Pi printed circuit board; the critical features denoted alphabetically are the following:

A – Gigabit Ethernet Connection: Crucial for the implementation of a high-speed Ethernet network for MMS and GOOSE communications as well as remote access via File

Transfer Protocol (FTP), Virtual Network Connection (VNC) or Secure Shell Hosting (SSH).

B – USB 3.0 Type A: May be used for transferring data such as data logs and new device firmware.

C – USB C 5V 3A Power Input: Low power consumption and the ability to be powered using a standard power adapter means the device can be deployed in areas without the additional power outlets

D – General Purpose Input/Output (GPIO) Pins: Can be used to interface with peripheral devices using several communications method. Could also be utilised for controlling low voltage interfacing relays for control functions.

E – 4GB LPDDR4 SDRAM: Enables the device to more effectively multitask and run multiple applications simultaneously

F – Broadcom 2711 64-bit 4 core Cortex A72 1.5 GHz CPU: High-speed multicore CPU for multitasking and fast data sampling.

G – MicroSD Card Slot: Can support high capacity Micro SD Card for Operating system installs and historical data storage. A database may be implemented in the operating system to store the historical data.

H – Micro HDMI: Device can be connected to an external display or touch screen. The inclusion of HDMI could be used to implement an HMI when coupled with a keyboard and mouse connected via the USB ports

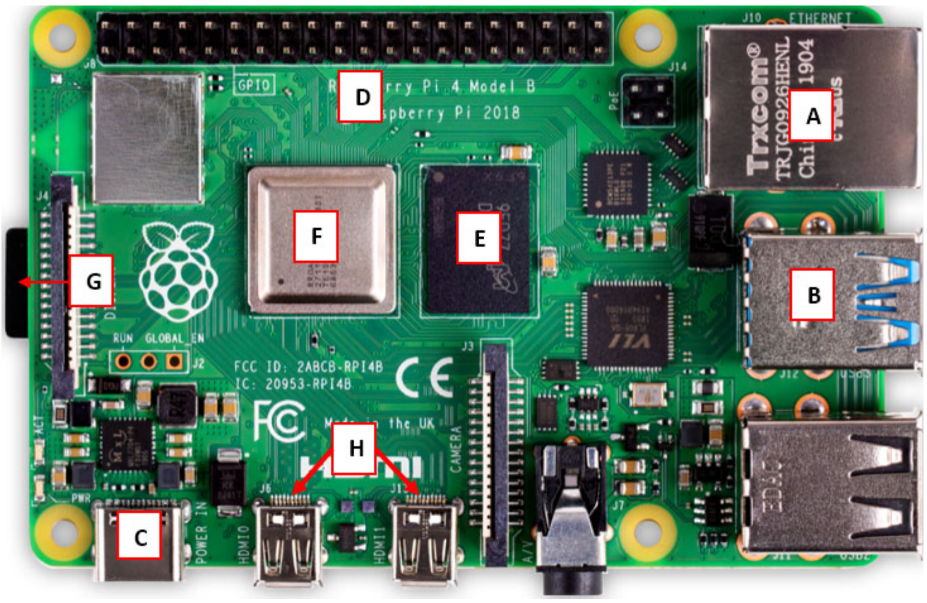


Figure 5-2: The Raspberry Pi hardware

5.4. Building and testing the IEC 61850 library

This section details the configuration and testing of the open-source IEC 61850 library. As this is an initial setup, a Windows PC is required to configure and test the functionality of the library using test code. LibIEC61850 is a server and client library containing services that enable IEC 61850 MMS, GOOSE and Sampled Values communication protocols. Figure 5-3 shows an overview of the communications network with the aforementioned MMS and GOOSE capabilities. The library is compatible with both Windows and Linux on these platforms, the Abstract Communication Services interface (ACSI) and GOOSE are implemented as part of the Application Programming Interface (API) shown as the application domain in Figure 5-3. The ACSI is mapped to the MMS, and TCP/IP in the communications stack then transmitted via ISO/IEC 8802-3 (Ethernet) while GOOSE being a peer-to-peer communications protocol bypasses the TCP/IP stack and maps directly to Ethernet. An unmanaged network switch is suitable for the initial setup and to establish communications.

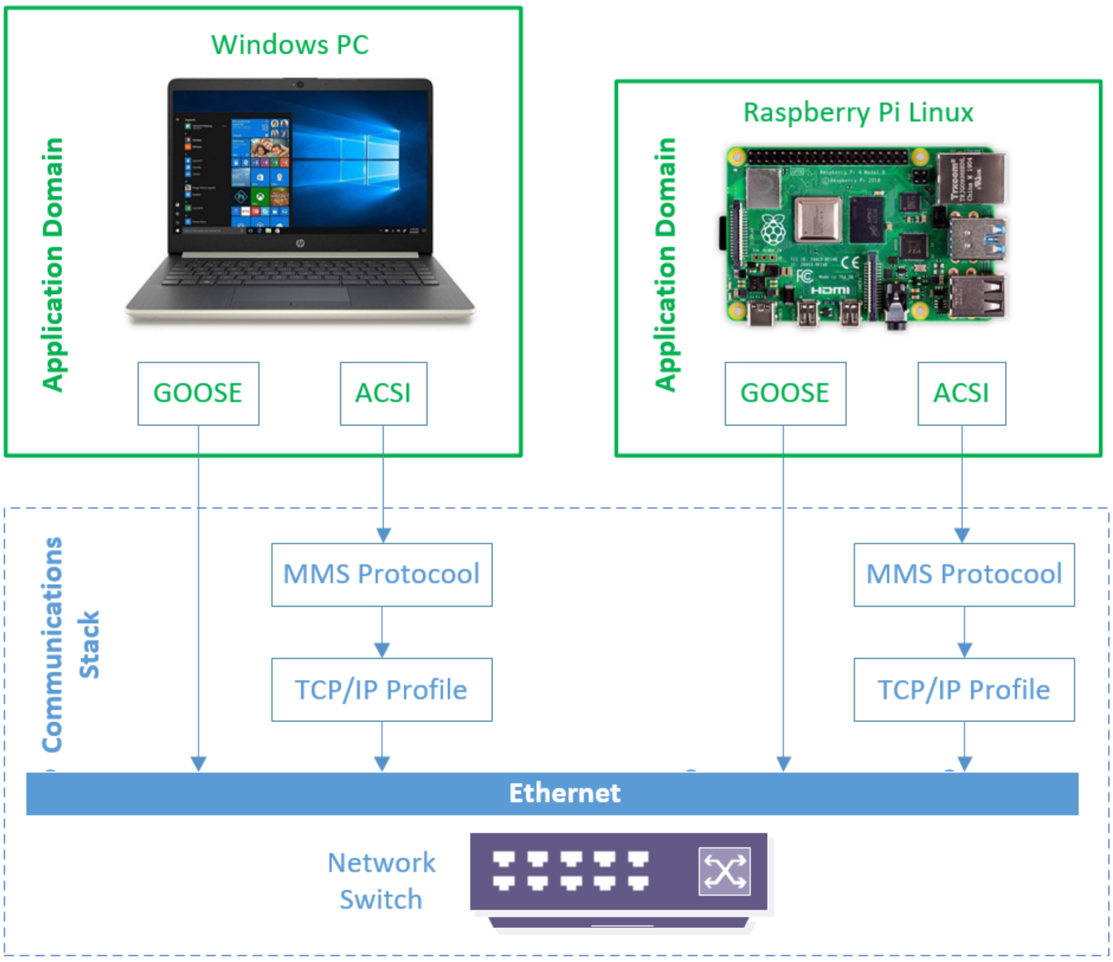


Figure 5-3: Overview of IEC 61850-Based Communications Network

Figure 5-4 presents some steps that need to be executed to configure and implement the communications network, as explained in Figure 5-3 using the open-source IEC 61850 library. In section 5.4.1, the installation of the operating system of Raspberry Pi is described, then the networking environment, FTP client and Virtual Network Computing (VNC) is configured. Finally, section 5.4.2 details the process for building and testing the library on both devices.

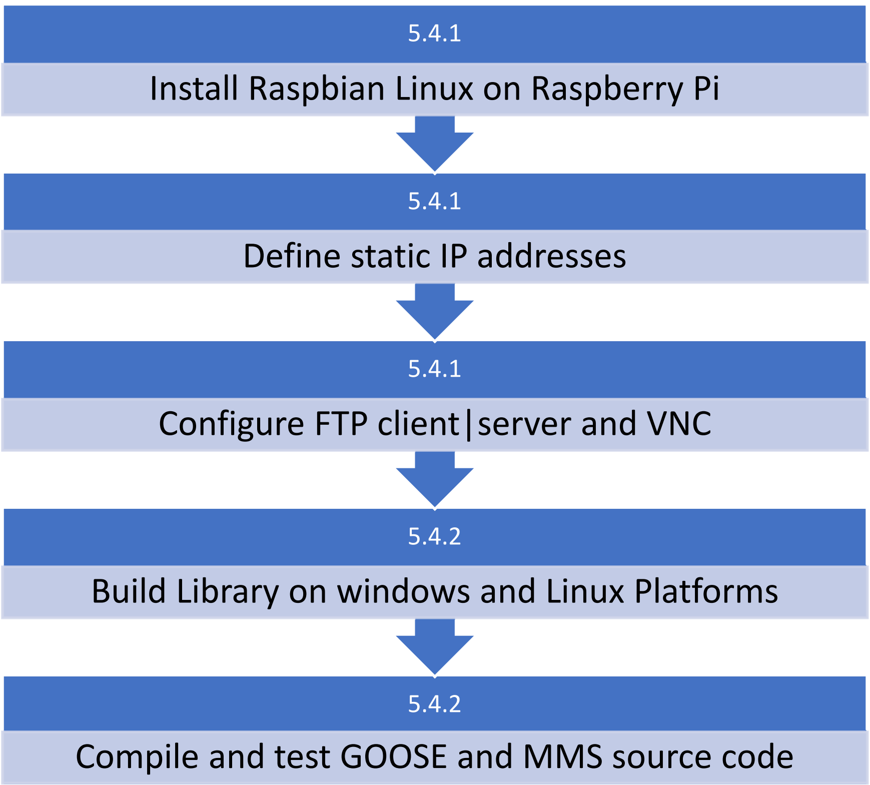
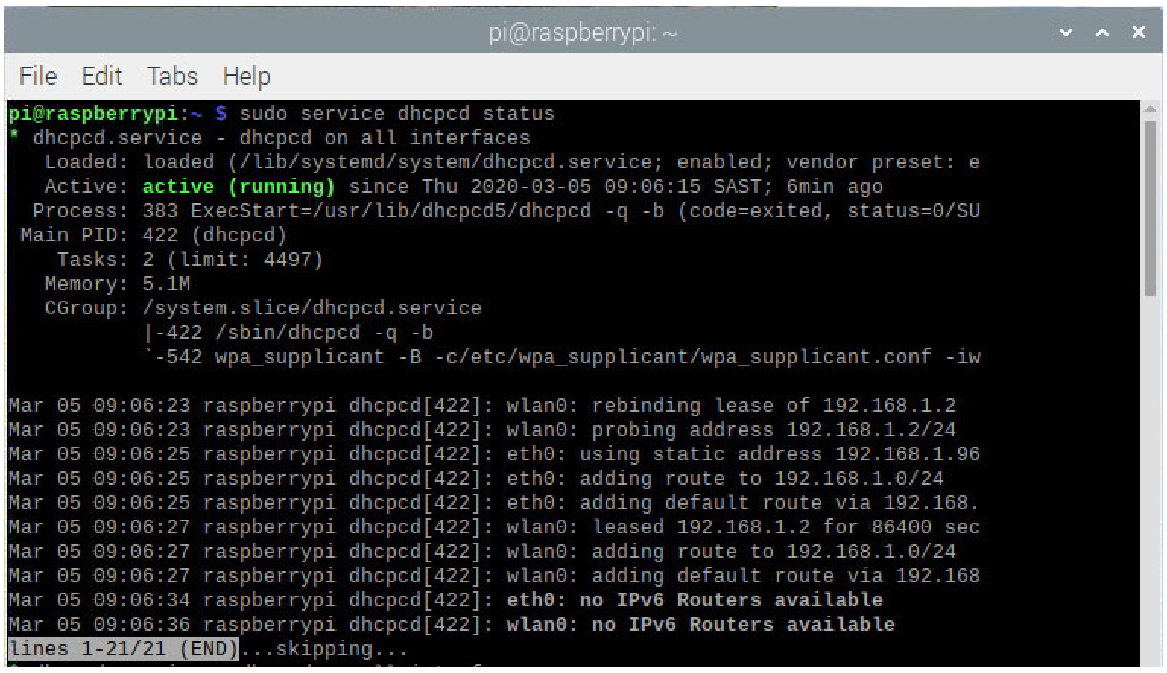

5.4.1
Install Raspbian Linux on Raspberry Pi
5.4.1
Define static IP addresses
5.4.1
Configure FTP client|server and VNC
5.4.2
Build Library on windows and Linux Platforms
5.4.2
Compile and test GOOSE and MMS source code

Figure 5-4: Procedure for the configuration and implementation of IEC 61850 library

5.4.1. Software and networking setup

The installed operating system on the Raspberry Pi can be any of the available Linux distributions that have been developed and recompiled to run on the ARM-based SoC. As such, there is no preference in terms of the operating system, and the Linux specific operations are described in a purely generic manner. The chosen operating system, in this case, is Raspbian. Raspbian is based on the classic Debian Linux distribution, and most of the binaries, software packages and Debian specific functions are available on Raspbian. This particular distribution comes with a built-in desktop environment, making it possible to interact with the system in a Graphical User Interface (GUI) instead of purely through the command line. Installation of Raspbian (Recently renamed to Raspberry Pi OS) is detailed in Appendix B. The operating system is installed on an appropriately sized SD Card which is then inserted into the Raspberry Pi’s SD card slot.

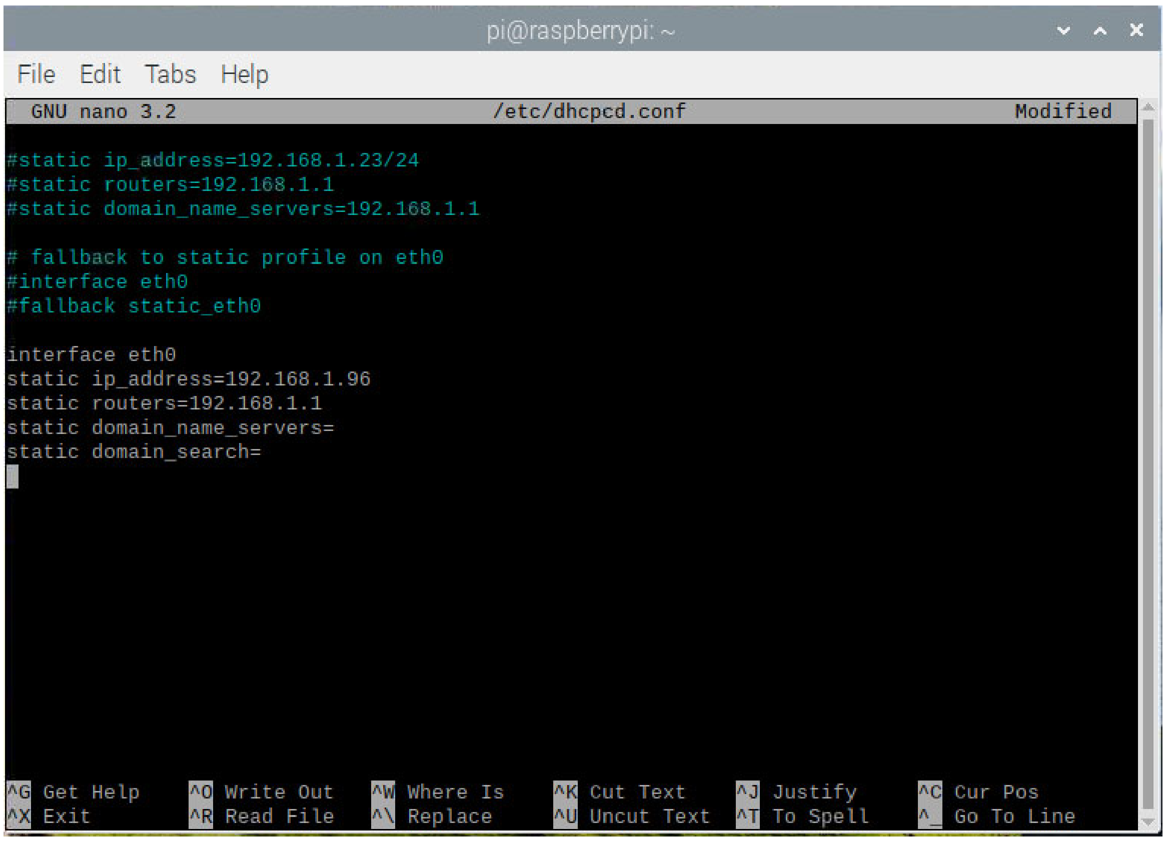
In an industrial networking environment, it is essential to configure devices with static IP addresses. Static IP addresses ensure that when trying to access the device remotely, i.e. via Secure Shell (SSH), FTP or VNC, the specific device addresses are known to the user. Static IP addresses are crucial for engineers to make configuration changes remotely, and specify firewall rules in the substation router to block or allow specified network traffic from untrusted devices and IP addresses. Cybersecurity is critically important, especially when dealing with traffic originating from a Wide Area Network (WAN), especially untrusted or Internet connections. Raspbian OS runs a Dynamic Host Control Protocol (DHCP) client daemon called dhcpcd that can communicate with the router's DHCP server. The available configuration file allows the user to assign a private IPv4 address for the Raspberry Pi, while the status of the dhcpcd client can be checked with the command "sudo service dhcpcd status". Results of the terminal command are shown in Figure 5-5, which include the active statues of the service as well as the current IP addresses being used by the available network interfaces.



```
pi@raspberrypi:~ $ sudo service dhcpcd status
* dhcpcd.service - dhcpcd on all interfaces
   Loaded: loaded (/lib/systemd/system/dhcpcd.service; enabled; vendor preset: e
   Active: active (running) since Thu 2020-03-05 09:06:15 SAST; 6min ago
  Process: 383 ExecStart=/usr/lib/dhcpcd5/dhcpcd -q -b (code=exited, status=0/SU
 Main PID: 422 (dhcpcd)
    Tasks: 2 (limit: 4497)
   Memory: 5.1M
   CGroup: /system.slice/dhcpcd.service
           |-422 /sbin/dhcpcd -q -b
           `-542 wpa_supplicant -B -c/etc/wpa_supplicant/wpa_supplicant.conf -iw

Mar 05 09:06:23 raspberrypi dhcpcd[422]: wlan0: rebinding lease of 192.168.1.2
Mar 05 09:06:23 raspberrypi dhcpcd[422]: wlan0: probing address 192.168.1.2/24
Mar 05 09:06:25 raspberrypi dhcpcd[422]: eth0: using static address 192.168.1.96
Mar 05 09:06:25 raspberrypi dhcpcd[422]: eth0: adding route to 192.168.1.0/24
Mar 05 09:06:25 raspberrypi dhcpcd[422]: eth0: adding default route via 192.168.
Mar 05 09:06:27 raspberrypi dhcpcd[422]: wlan0: leased 192.168.1.2 for 86400 sec
Mar 05 09:06:27 raspberrypi dhcpcd[422]: wlan0: adding route to 192.168.1.0/24
Mar 05 09:06:27 raspberrypi dhcpcd[422]: wlan0: adding default route via 192.168
Mar 05 09:06:34 raspberrypi dhcpcd[422]: eth0: no IPv6 Routers available
Mar 05 09:06:36 raspberrypi dhcpcd[422]: wlan0: no IPv6 Routers available
lines 1-21/21 (END)...skipping...
```

Figure 5-5: Results of the "dhcpcd status" command

Configuring a static IP address requires the modification of the "dhcpcd.conf" file using the terminal command "sudo nano /etc/dhcpcd.conf". The results can be seen in Figure 5-6, where the contents of the configuration file specify the interface and its configuration parameters. The static IP address chosen for the new device is 192.168.1.96, and the static router address is set as 192.168.1.1.



```
  GNU nano 3.2                    /etc/dhcpcd.conf                    Modified

#static ip_address=192.168.1.23/24
#static routers=192.168.1.1
#static domain_name_servers=192.168.1.1

# fallback to static profile on eth0
#interface eth0
#fallback static_eth0

interface eth0
static ip_address=192.168.1.96
static routers=192.168.1.1
static domain_name_servers=
static domain_search=


^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^_ Go To Line
```

Figure 5-6: Dhcpcd.conf file contents

Once a static IP address has been specified, an FTP server and VNC may be configured. FTP servers allow users to transfer files remotely and VNC's enable the use of a remote desktop environment to make device changes. An optional method for connecting to the Raspberry Pi is SSH, but this is not ideal for this specific use case due to SSH being purely command line. The FTP server used on the Raspberry Pi is the Very Secure File Transfer Protocol Daemon (VSFTPD) which is considered one of the most secure FTP daemons. The word "daemon" simply refers to programs that run as background processes on Linux and are typically identified by having names ending with the letter "d". The FTP server allows Windows programs like WinSCP to remotely interact with the Raspberry Pi's file system, which is essential for transferring files such as logs and configuration files. Raspbian comes with a built-in VNC that only requires enabling as it is disabled by default for security reasons. To allow it to, the command "sudo raspi-config" is used to access the Raspbian's software configuration tool. Selecting "Interfacing Options" opens a menu giving you the ability to enable VNC among other functions. Once VNC is enabled on the Raspberry Pi, the "VNC Viewer" software can be used to access the device remotely.

5.4.2. Building and testing the library

With the networking and communications environment configured for the Raspberry Pi, the LibIEC61850 library can now be built and tested. The open-source library needs to be built on both the Windows PC and Raspberry Pi to test MMS and GOOSE functionality, as shown in Figure 5-7 and 5-8.

The procedure for building the library on Linux differs from Windows, as building on Linux uses a traditional "GNU's Not Linux" (GNU) based "make" system, whilst Windows uses "CMake" which in the case of LibIEC61850 requires Microsoft Visual Studio. CMake is open-source software that is used to build, test and package software; the advantage of CMake over other methods is that it is cross-platform. Using CMake allows the user to compile their software in their preferred environment, whether it be Microsoft Visual Studio or Linux. However, the methodology that one uses in Windows as opposed to Linux differs somewhat. Figures 5-7 and 5-8 show the steps for building the library on both Windows (Figure 5-7) and Linux (Figure 5-8). These methods are sufficient to run the MMS examples on both Windows and Linux operating systems as enabling GOOSE messaging requires additional steps on Windows as opposed to Linux.

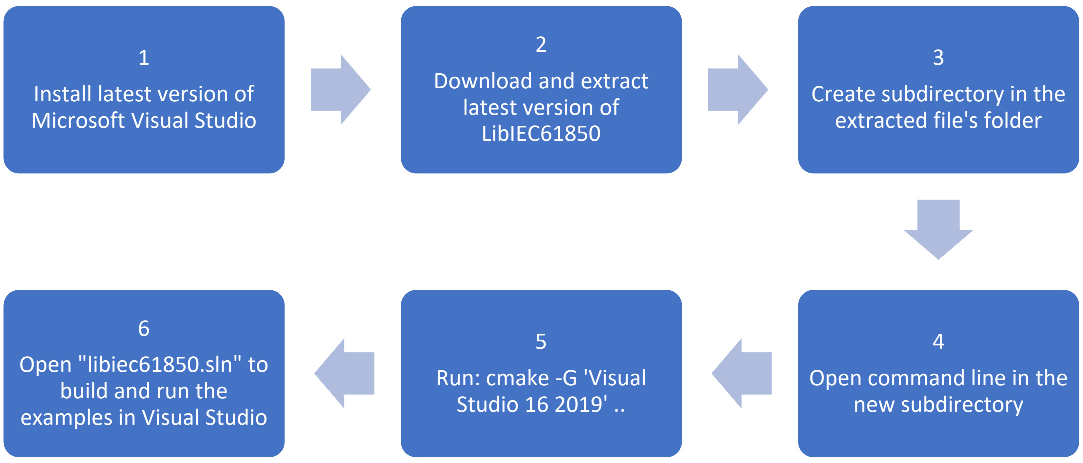

1
Install latest version of Microsoft Visual Studio
2
Download and extract latest version of LibIEC61850
3
Create subdirectory in the extracted file's folder
4
Open command line in the new subdirectory
5
Run: cmake -G 'Visual Studio 16 2019' ..
6
Open "libiec61850.sln" to build and run the examples in Visual Studio

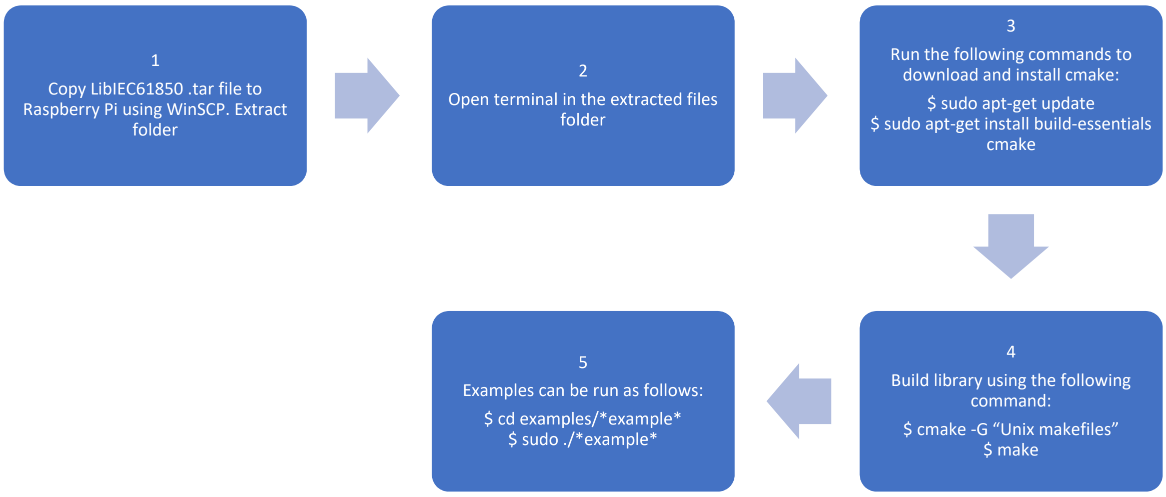
Figure 5-7: Building the Library using the Windows Operating System

The diagram presented in Figure 5-7 shows the steps involved when building the open-source library in windows. The following steps, as shown in the image, are affected:

1) **Install the latest version of Microsoft Visual Studio:** Microsoft Visual Studio is an Integrated Development Environment (IDE), it exists as several versions, community, professional and enterprise. The differences between each version relate to licensing restrictions, with community meant for students and development teams of less than five developers. In contrast, professional and enterprise are intended for larger projects and corporations. The community version is sufficient for this project.

2) **Download and extract the latest version of LibIEC61850:** The newest version of the LibIEC61850 software library may be downloaded from the following URL - https://libiec61850.com/libiec61850/downloads/. It is beneficial to download the newest version of the library due to the constant feature updates and bug fixes since the library is an ongoing project.
3) **Create a subdirectory in the extracted file's folder:** The newly created subdirectory serves as the location of the built library's visual studio project and "make" files.
4) **Open a command line in the new subdirectory:** Holding the shift button, then pressing the right mouse button within the new subdirectory enables the user to open a Powershell window in the folder, negating the need to navigate to the folder via command line.
5) **Run the command "cmake -G 'Visual Studio 16 2019'":** This command invokes the cmake system and build the library within the subdirectory according to the code within the makefile. If new source code files apart from the included examples require creation, then they must be written into the makefile to be included as part of the project solution.
6) **Open "libIEC61850.sln" to build and run the examples in Visual Studio:** Opening the solution file within the folder allows for the editing and compiling of example files.

Enabling GOOSE messaging functionality within the Windows Operating System environment requires the installation of WinPcap software, which is a tool that bypasses the protocol stack, enabling specific applications to capture and transmit network packets. It is not necessary to enable GOOSE messaging on the Windows machine as it not sending or receiving GOOSE messages.



1
Copy LibIEC61850 .tar file to Raspberry Pi using WinSCP. Extract folder

2
Open terminal in the extracted files folder

3
Run the following commands to download and install cmake:
```
$ sudo apt-get update
$ sudo apt-get install build-essentials cmake
```

4
Build library using the following command:
```
$ cmake -G “Unix makefiles”
$ make
```

5
Examples can be run as follows:
```
$ cd examples/*example*
$ sudo ./*example*
```

Figure 5-8:Building the Library using a Linux Operating System

Figure 5.7 shows the steps involve when building LibIEC61850 on a Linux system. The following are steps that require completion to build the software library:

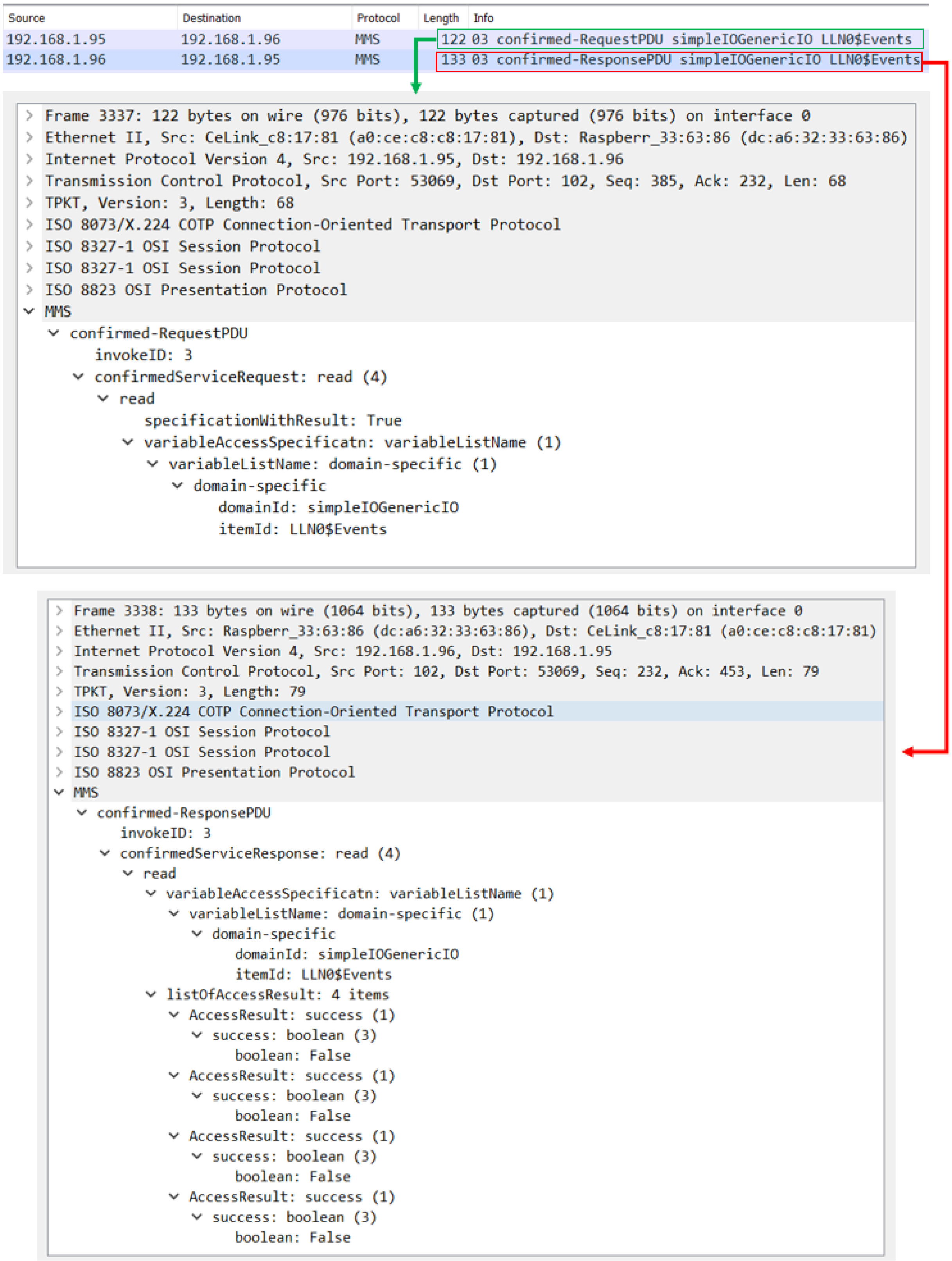
1) **Copy LibIEC61850.tar file to Raspberry Pi using WinSCP and extract folder:** The downloaded archive or .tar type must be copied to the Raspberry Pi using WinSCP. A terminal may be opened within the folder on the Raspberry Pi and the .tar file extracted using the command “tar xvf libiec61850-1.3.3.tar”.
   Where,
   -x: Extract a tar ball
   -v: Verbose output or show progress while extracting files
   -f: Specify an archive or a tarball filename
2) **Open terminal in extracted files folder.**
3) **Download and install Cmake:** On Linux, Cmake may be installed using the following procedure:
   First, the package information from all sources my is downloaded using the “sudo apt-get update”. Then cmake can be downloaded using “sudo apt-get install build-essentials cmake”.
4) **Build the library:** Simply running the command “$ cmake -G “Unix makefiles” $make” within the folder containing the makefile. Similar to windows, new source code files may be written then added to the makefile’s source code. Edited source code files can be built simply by typing “make” in the terminal window, this runs the script within the make file present in the folder.
5) **Run example code:** Example code may be run by navigating to the folder containing the executable file then running the following command: “$ sudo ./example”.

One of the primary methods of transporting data between devices in the IEC 61850 standard is by mapping the services, objects and algorithms described in the previous chapter to the Manufacturing Message Specification (MMS or ISO 9506). MMS allows the exchange of real-time data and supervisory control information within a network or between applications (SISCO Inc., 1995). The main advantages of MMS are that it is not dependent on the specific function being performed and is vendor-agnostic, which allows for greater interoperability, making it ideal for use within the IEC 61850 standard.

The LibIEC61850 software contains many MMS-based example programs for testing the communications functions and library features. Testing the available source code for MMS requires running an MMS server and client. On the Raspberry Pi, server_example_basic_io updates attribute from a predetermined static model, whilst on the Windows PC, iec61850_client_example1 reads specific dataset values. The attribute values can be manually modified in the source code to test whether MMS is functioning correctly.

The functionality of the MMS can be confirmed using protocol analyser software such as Wireshark which is shown in Figure 5-9. The source and destination IP addresses indicate from which device the MMS requests and responses originate. Each data object is read from the specified dataset, in this case, "simpleIOGenericIO/LLN0.Events". Each protocol data unit (PDU) contains necessary information that applies to both MMS requests and responses such as; IPv4 information, Transmission Control Protocol (TCP) information and finally the OSI Protocol information. A detailed explanation of the MMS protocol stack as applied in IEC 61850 may be found in IEC 61850-8-1, Page 21-25 (IEC, 2005c).

Marked in green, the MMS section of the first message in the transaction is a read request sent by the Windows PC for data set “Events” located within Logical Node Zero (LLNO), this forms the “itemID” within the message. This data belongs to an IED “simpleIO”, the logical device “GenericIO”, which forms the “domainID” within the MMS data packet. The response (marked in red) includes the confirmed number of results as 4 (data points within the data set), then lists each data point along with the data type and current state.



```
Source           Destination      Protocol  Length  Info
192.168.1.95     192.168.1.96     MMS       122 03 confirmed-RequestPDU simpleIOGenericIO LLN0$Events
192.168.1.96     192.168.1.95     MMS       133 03 confirmed-ResponsePDU simpleIOGenericIO LLN0$Events
```

```
> Frame 3337: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0
> Ethernet II, Src: CeLink_c8:17:81 (a0:ce:c8:c8:17:81), Dst: Raspberr_33:63:86 (dc:a6:32:33:63:86)
> Internet Protocol Version 4, Src: 192.168.1.95, Dst: 192.168.1.96
> Transmission Control Protocol, Src Port: 53069, Dst Port: 102, Seq: 385, Ack: 232, Len: 68
> TPKT, Version: 3, Length: 68
> ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
> ISO 8327-1 OSI Session Protocol
> ISO 8327-1 OSI Session Protocol
> ISO 8823 OSI Presentation Protocol
v MMS
    v confirmed-RequestPDU
          invokeID: 3
        v confirmedServiceRequest: read (4)
            v read
                  specificationWithResult: True
                v variableAccessSpecificatn: variableListName (1)
                    v variableListName: domain-specific (1)
                        v domain-specific
                              domainId: simpleIOGenericIO
                              itemId: LLN0$Events
```

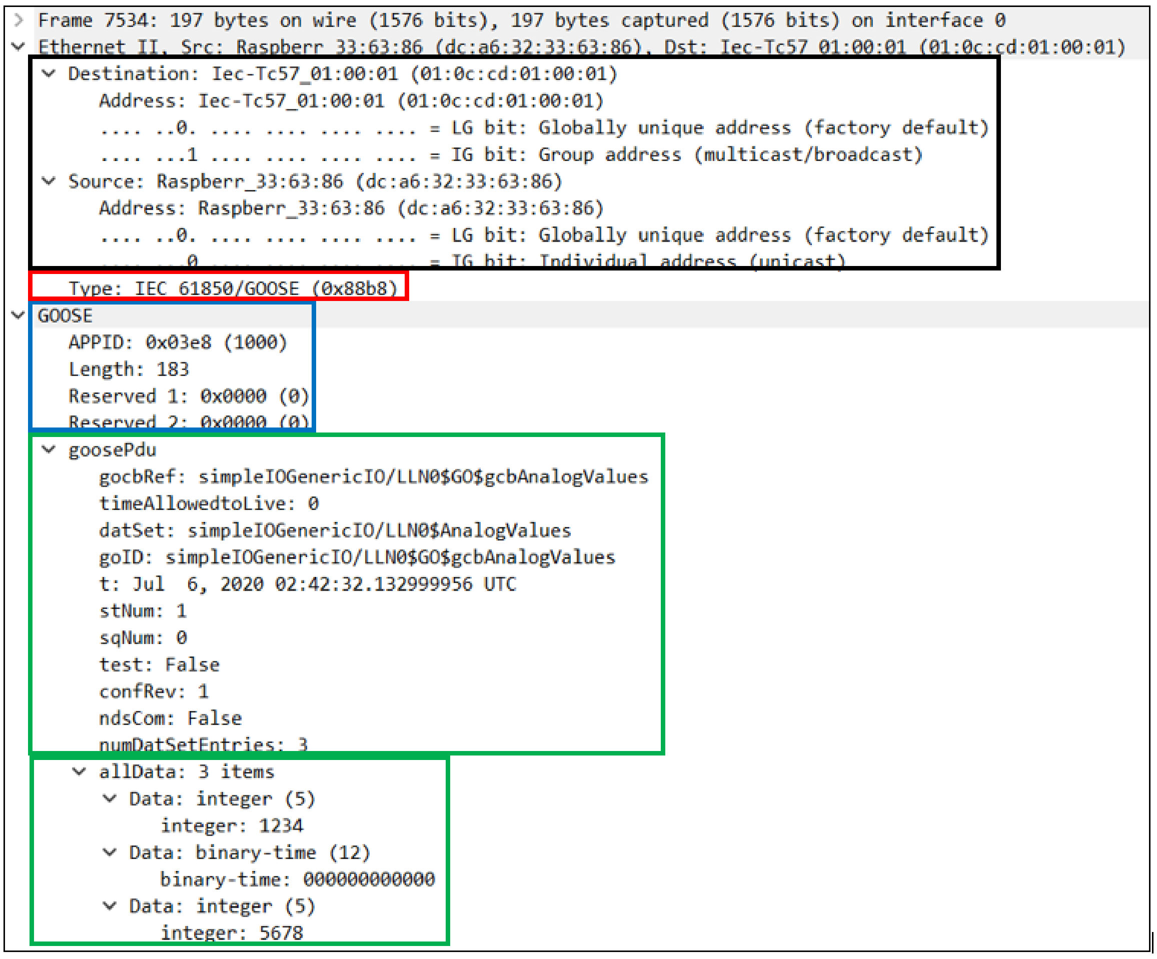
```
> Frame 3338: 133 bytes on wire (1064 bits), 133 bytes captured (1064 bits) on interface 0
> Ethernet II, Src: Raspberr_33:63:86 (dc:a6:32:33:63:86), Dst: CeLink_c8:17:81 (a0:ce:c8:c8:17:81)
> Internet Protocol Version 4, Src: 192.168.1.96, Dst: 192.168.1.95
> Transmission Control Protocol, Src Port: 102, Dst Port: 53069, Seq: 232, Ack: 453, Len: 79
> TPKT, Version: 3, Length: 79
> ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
> ISO 8327-1 OSI Session Protocol
> ISO 8327-1 OSI Session Protocol
> ISO 8823 OSI Presentation Protocol
v MMS
    v confirmed-ResponsePDU
          invokeID: 3
        v confirmedServiceResponse: read (4)
            v read
                v variableAccessSpecificatn: variableListName (1)
                    v variableListName: domain-specific (1)
                        v domain-specific
                              domainId: simpleIOGenericIO
                              itemId: LLN0$Events
                v listOfAccessResult: 4 items
                    v AccessResult: success (1)
                        v success: boolean (3)
                              boolean: False
                    v AccessResult: success (1)
                        v success: boolean (3)
                              boolean: False
                    v AccessResult: success (1)
                        v success: boolean (3)
                              boolean: False
                    v AccessResult: success (1)
                        v success: boolean (3)
                              boolean: False
```

Figure 5-9: Wireshark Snapshot of MMS Traffic

The source code for GOOSE messaging functionality may be tested using the “goose_publisher_example” c source code. An identical GOOSE message is published three times every 1000ms, as shown in Figure 5-10. The GOOSE Protocol Data Unit (PDU) information contains all of the parameters of the GOOSE Control Block, Dataset and Goose ID as well as the specific data set items present in the message. In this instance, it is only necessary to verify that the GOOSE message is being publishing since no control or protection functions are going to be implemented on the PQ Monitoring device.

GOOSE messaging is an important aspect of what makes the IEC 61850 standard unique; as such, it is worth going into a bit more detail regarding what is contained within a GOOSE message. Simplifying the message feature in Figure 5-10 into a logical structure yields the block diagram shown below the Wireshark extract which shows the structure of the GOOSE PDU within the framework of the ISO/IEC 8802-3 frame format. Each part of the data packet is identified using unique colours. The header portion contains information about the MAC address of the device where the GOOSE message originated from, in this case, the source is the Raspberry Pi. Since GOOSE messages are broadcast UDP published messages, they do not have any specific destination; thus, the destination is shown as any device that’s compatible with GOOSE frames as described in the standards developed by the IEC-TC57. The payload is of Ethertype GOOSE, this payload is defined with AppID, frame length and finally a GOOSE Protocol Data Unit (PDU) field. The GOOSE PDU consists of the following parameters:

- gocbRef (GOOSEcontrolblockReference)
- TAL (Time Allowed to Live)
- Datset (Data-set)
- goID (GOOSE ID)
- t (timestamp)
- stNum (state Number)
- sqNum (sequence Number)
- confRev (configuration Revision)
- ndsCom (needs Commissioning)
- numDatsetEntries (number of dataset entries)
- All data – which contains the actual data points contained in the data set.



```
> Frame 7534: 197 bytes on wire (1576 bits), 197 bytes captured (1576 bits) on interface 0
v Ethernet II, Src: Raspberr_33:63:86 (dc:a6:32:33:63:86), Dst: Iec-Tc57_01:00:01 (01:0c:cd:01:00:01)
  v Destination: Iec-Tc57_01:00:01 (01:0c:cd:01:00:01)
      Address: Iec-Tc57_01:00:01 (01:0c:cd:01:00:01)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
  v Source: Raspberr_33:63:86 (dc:a6:32:33:63:86)
      Address: Raspberr_33:63:86 (dc:a6:32:33:63:86)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IEC 61850/GOOSE (0x88b8)
v GOOSE
    APPID: 0x03e8 (1000)
    Length: 183
    Reserved 1: 0x0000 (0)
    Reserved 2: 0x0000 (0)
  v goosePdu
      gocbRef: simpleIOGenericIO/LLN0$GO$gcbAnalogValues
      timeAllowedtoLive: 0
      datSet: simpleIOGenericIO/LLN0$AnalogValues
      goID: simpleIOGenericIO/LLN0$GO$gcbAnalogValues
      t: Jul  6, 2020 02:42:32.132999956 UTC
      stNum: 1
      sqNum: 0
      test: False
      confRev: 1
      ndsCom: False
      numDatSetEntries: 3
    v allData: 3 items
      v Data: integer (5)
          integer: 1234
      v Data: binary-time (12)
          binary-time: 000000000000
      v Data: integer (5)
          integer: 5678
```

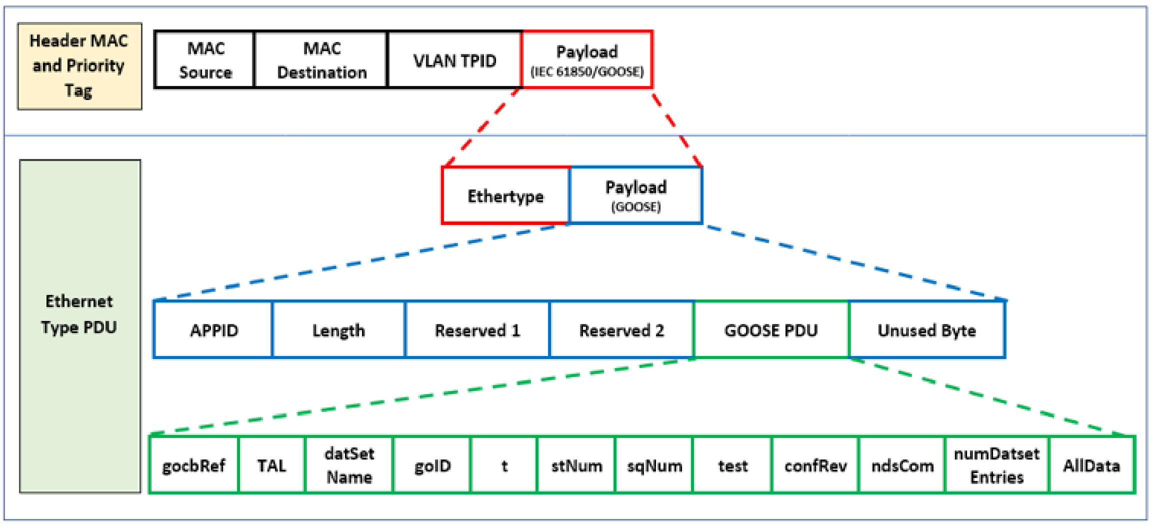
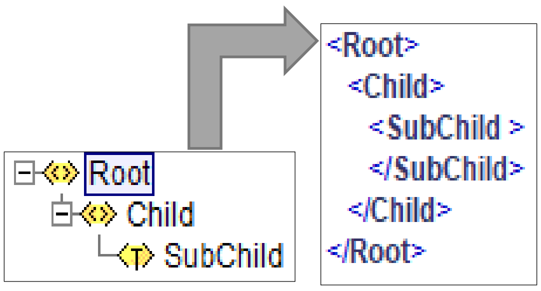


Figure 5-10: Example packet and structure of a GOOSE Message (Ingalalli et al., 2017)

5.5. ICD file parameterisation and overview

The static data model is generated using the code generation tool included as part of the LibIEC61850 software. Functional Constraint Data Attributes (FCDA) can be accessed and updated programmatically using the generated C code as a part of the source code, code generation is described in Section 5.6. The type of SCL file used for generating a static model is the ICD file, which is used to describe the built-in capabilities of a single IED. These capabilities, as determined in Chapter 3, are modelled in the ICD file following the available functions on the ADE9000. The ICD file contains several sections: Header, Communication section, IED section, and DataTypeTemplates.

The basic structure of the XML document is based on the element tree. A single root element forms the highest level of the tree, with child elements forming the branches, sub-child elements may also be present within the child element branches. Figure 5-11 demonstrates the basic XML tree structure, where relationships between the elements are described using the terms parent, child and sibling. In Figure 5-10, the “root” is the parent, “child” and “sub child” are the children. A child can only exist with a parent and siblings (the other children that live on the same level), the XML tree structure forms the basis of the ICD file. The following sections detail the individual child elements as written for the PQ Monitoring device according to the XML tree structure.

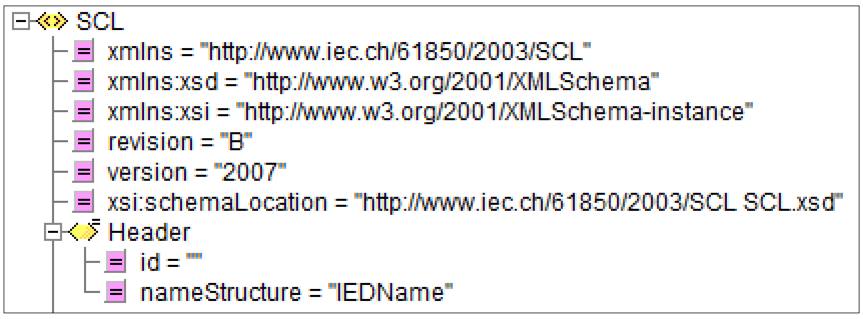


```
<Root>
  <Child>
    < SubChild >
    </SubChild>
  </Child>
</Root>
```

Figure 5-11: Parent and Child structure of XML

5.5.1. XML Schema and Header section

The root element or “parent” of the ICD file is called “SCL”. Shown in Figure 5-12, the SCL section of the XML file contains all of the child elements, namely: Header, Communications, IED and DataTypeTemplates. Schema information including the version, revision and location of the standard all form part of the necessary information for the XML schema. The header child element contains information such as the IED name.



```
SCL
  xmlns = "http://www.iec.ch/61850/2003/SCL"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  revision = "B"
  version = "2007"
  xsi:schemaLocation = "http://www.iec.ch/61850/2003/SCL SCL.xsd"
  Header
    id = ""
    nameStructure = "IEDName"
```

Figure 5-12: XML Header

5.5.2. Communication section

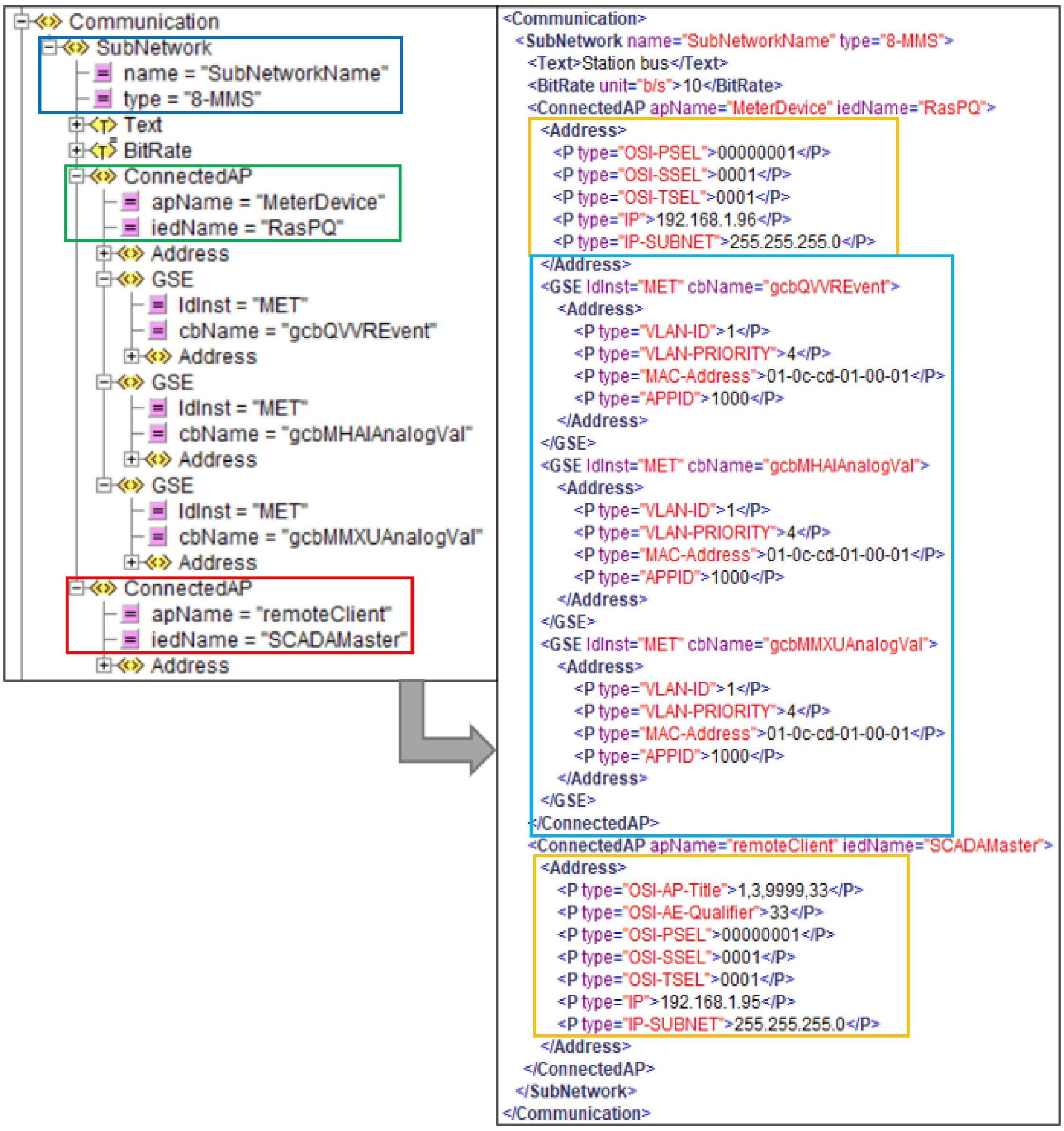
In the Communication section of the XML file, parameters regarding the communications network are present. Various connections to subnetworks and networks are detailed here, as well as their communication access points.

In the ICD file, as shown in Figure 5-13, one subnetwork called “SubNetworkName” (in the blue rectangle) is connected to the access points “MeterDevice” (In green rectangle) belonging to IED “RasPQ” and “remoteClient” (in red rectangle) belonging to IED “SCADAMaster”. The sub-network type 8-MMS refers to the MMS Protocol are specified in IEC 61850-8-1 and IEC 61850-9-2. The GSE control blocks for QVVR (Voltage Variation), MHAI (Harmonics and Interharmonics) and MMXU (Measurement) logical node data objects are also defined in this section.

Within the connectedAPs, two types of addresses are present, ones for MMS (highlighted in orange) and one for each GOOSE Control Block (shown in the light blue rectangle). There are many parameters pertaining to the MMS communications configuration, the specifics for mapping services and objects of the Abstract Communication Service Interface (ACSI, IEC 61850-7-2) to Manufacturing Message Specification (MMS. ISO 9506) and ISO/IEC 8802-3 (Ethernet) frames are detailed in IEC 61850-8-1. The important MMS parameters that appear in connectAP are:

- AP Title: Typically set to 1,3,9999 as standard.
- AE Qualifier: An integer value that is used to identify the application entity (unique identifier for the specific application)
- OSI Presentation Selector (PSEL)
- OSI Session Selector (SSEL)
- OSI Transport Selector (TSEL)

The GOOSE control block’s address information includes four typical parameters: VLAN-ID, VLAN-PRIORITY, MAC-Address and APPID. Of the listed parameters, only the MAC-address and APPID parameters are required, the MAC address entry is typically a multicast mac address.



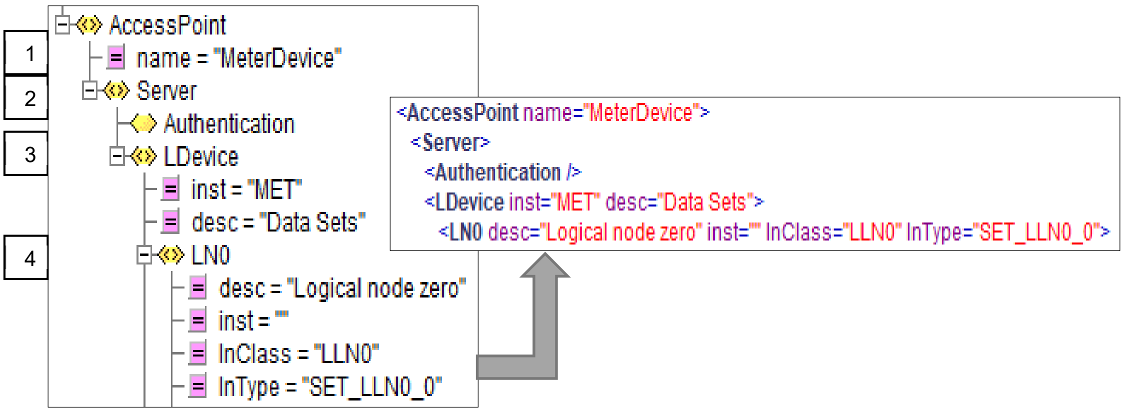
```
Communication
  SubNetwork
    name = "SubNetworkName"
    type = "8-MMS"
    Text
    BitRate
    ConnectedAP
      apName = "MeterDevice"
      iedName = "RasPQ"
      Address
      GSE
        ldInst = "MET"
        cbName = "gcbQVVREvent"
        Address
      GSE
        ldInst = "MET"
        cbName = "gcbMHAIAnalogVal"
        Address
      GSE
        ldInst = "MET"
        cbName = "gcbMMXUAnalogVal"
        Address
    ConnectedAP
      apName = "remoteClient"
      iedName = "SCADAMaster"
      Address
```

```
<Communication>
 <SubNetwork name="SubNetworkName" type="8-MMS">
  <Text>Station bus</Text>
  <BitRate unit="b/s">10</BitRate>
  <ConnectedAP apName="MeterDevice" iedName="RasPQ">
   <Address>
    <P type="OSI-PSEL">00000001</P>
    <P type="OSI-SSEL">0001</P>
    <P type="OSI-TSEL">0001</P>
    <P type="IP">192.168.1.96</P>
    <P type="IP-SUBNET">255.255.255.0</P>
   </Address>
   <GSE ldInst="MET" cbName="gcbQVVREvent">
    <Address>
     <P type="VLAN-ID">1</P>
     <P type="VLAN-PRIORITY">4</P>
     <P type="MAC-Address">01-0c-cd-01-00-01</P>
     <P type="APPID">1000</P>
    </Address>
   </GSE>
   <GSE ldInst="MET" cbName="gcbMHAIAnalogVal">
    <Address>
     <P type="VLAN-ID">1</P>
     <P type="VLAN-PRIORITY">4</P>
     <P type="MAC-Address">01-0c-cd-01-00-01</P>
     <P type="APPID">1000</P>
    </Address>
   </GSE>
   <GSE ldInst="MET" cbName="gcbMMXUAnalogVal">
    <Address>
     <P type="VLAN-ID">1</P>
     <P type="VLAN-PRIORITY">4</P>
     <P type="MAC-Address">01-0c-cd-01-00-01</P>
     <P type="APPID">1000</P>
    </Address>
   </GSE>
  </ConnectedAP>
  <ConnectedAP apName="remoteClient" iedName="SCADAMaster">
   <Address>
    <P type="OSI-AP-Title">1,3,9999,33</P>
    <P type="OSI-AE-Qualifier">33</P>
    <P type="OSI-PSEL">00000001</P>
    <P type="OSI-SSEL">0001</P>
    <P type="OSI-TSEL">0001</P>
    <P type="IP">192.168.1.95</P>
    <P type="IP-SUBNET">255.255.255.0</P>
   </Address>
  </ConnectedAP>
 </SubNetwork>
</Communication>
```

Figure 5-13:Communication Section

5.5.3. IED section

The IED section of the XML file contains information about the “MeterDevice” access point as defined in the communications section. Figure 5-14 shows the MeterDevice (1) access point with the server (2), LDevice (3) and LN0 (4) sub-children. LDevice is the logical device which contains logical nodes LN0, LPHD, MMXU, QVVR and MHAI. LLN0 (Logical Node Zero) is vital as it includes dataset definitions as shown in Figures 5-15 for QVVR definitions, 5-16 for MHAI definitions and 5-17 for MMXU definitions. Additionally, report control block and GSE control block definitions are also found in LLN0, as shown in Figure 5-18 and 5-19.

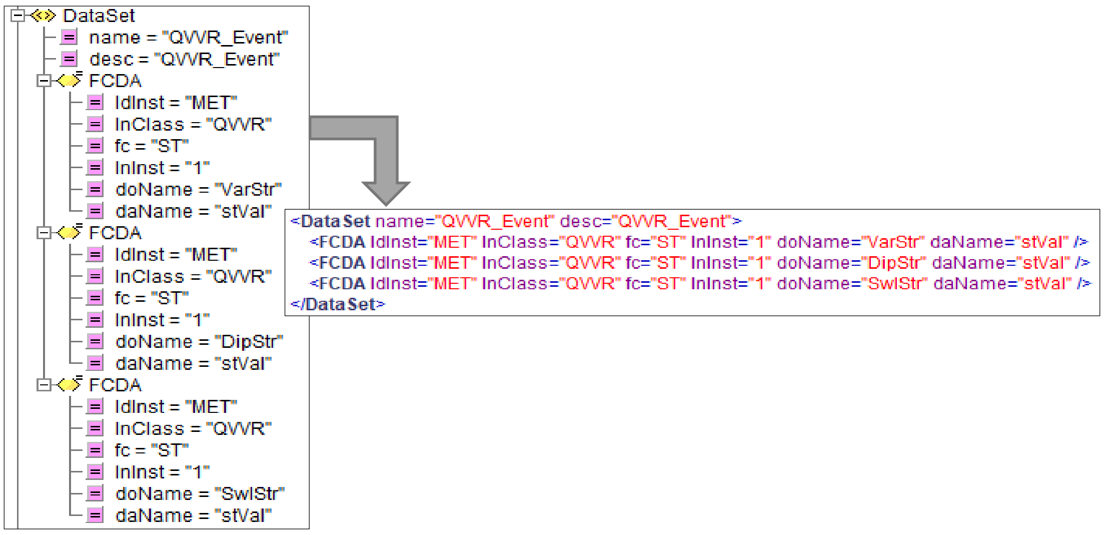


```
AccessPoint
  name = "MeterDevice"
  Server
    Authentication
    LDevice
      inst = "MET"
      desc = "Data Sets"
      LN0
        desc = "Logical node zero"
        inst = ""
        lnClass = "LLN0"
        lnType = "SET_LLN0_0"
```

```
<AccessPoint name="MeterDevice">
  <Server>
    <Authentication />
    <LDevice inst="MET" desc="Data Sets">
      <LN0 desc="Logical node zero" inst="" lnClass="LLN0" lnType="SET_LLN0_0">
```

Figure 5-14: Logical Node Zero

Figure 5-15 illustrates the structure for the QVVR_Event dataset. Three Functional Constraint Data Attributes (FCDA's) are present for data objects VarStr, DipStr and SwlStr. All three belong to the MET logical device instance and QVVR logical node class. The functional constraint is "ST", meaning all three are constrained to only status functions, and finally, the data attribute is "stVal" or status value.

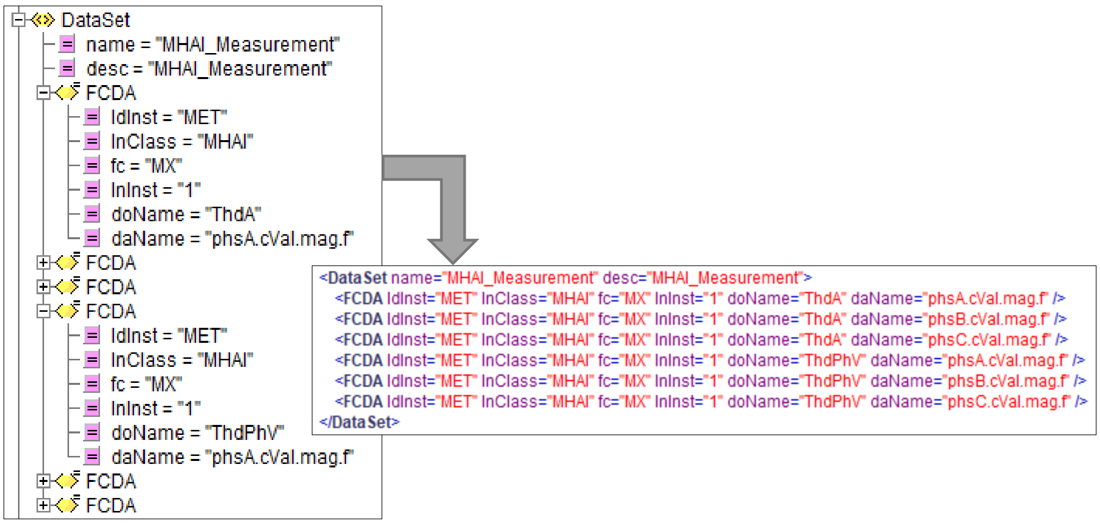


```
DataSet
  name = "QVVR_Event"
  desc = "QVVR_Event"
  FCDA
    ldInst = "MET"
    lnClass = "QVVR"
    fc = "ST"
    lnInst = "1"
    doName = "VarStr"
    daName = "stVal"
  FCDA
    ldInst = "MET"
    lnClass = "QVVR"
    fc = "ST"
    lnInst = "1"
    doName = "DipStr"
    daName = "stVal"
  FCDA
    ldInst = "MET"
    lnClass = "QVVR"
    fc = "ST"
    lnInst = "1"
    doName = "SwlStr"
    daName = "stVal"
```

```
<DataSet name="QVVR_Event" desc="QVVR_Event">
  <FCDA ldInst="MET" lnClass="QVVR" fc="ST" lnInst="1" doName="VarStr" daName="stVal" />
  <FCDA ldInst="MET" lnClass="QVVR" fc="ST" lnInst="1" doName="DipStr" daName="stVal" />
  <FCDA ldInst="MET" lnClass="QVVR" fc="ST" lnInst="1" doName="SwlStr" daName="stVal" />
</DataSet>
```

Figure 5-15: QVVR Dataset containing dip and swell status values and a general VarStr status value

In Figure 5-16, the dataset for MHAI_Measurement is shown to have FCDA's for instances of MET and the logical node class MHAI. The functional constraint (FC) is MX, logical node instance is 1, the data object is ThdA or ThdPhV, and finally, the data attribute is "phsA.cVal.mag.f". The FCDA is replicated for all three phases for both ThdA and ThdPhV.

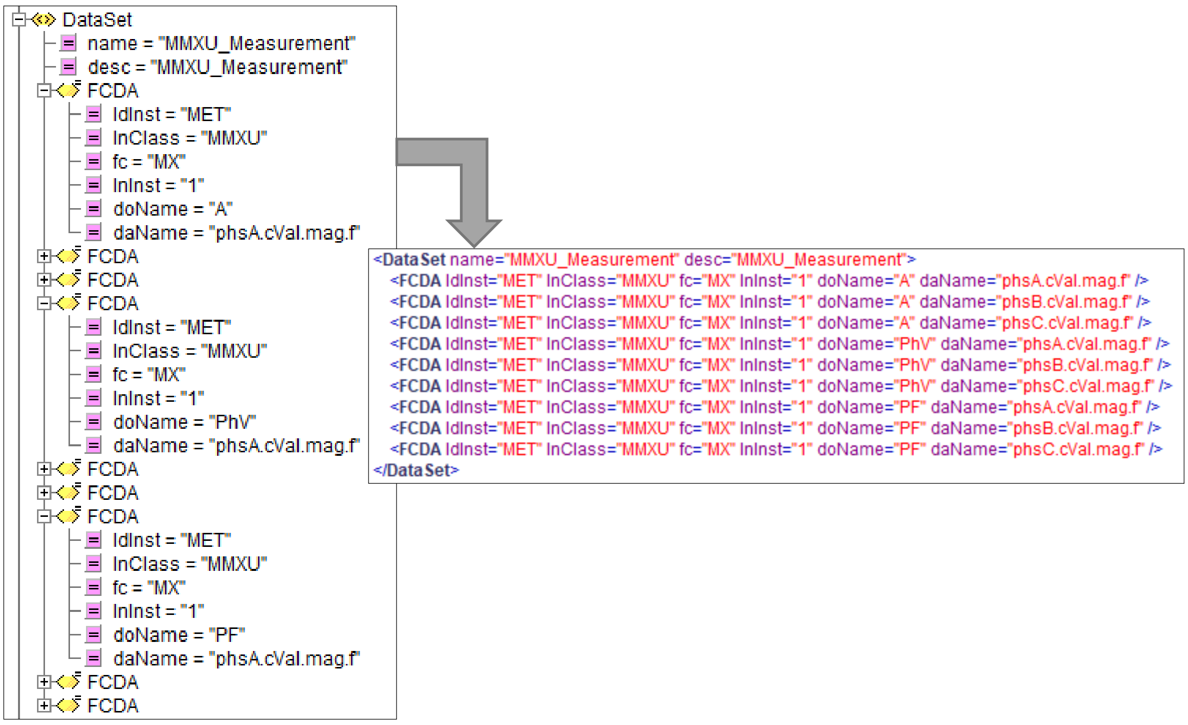


```
DataSet
  name = "MHAI_Measurement"
  desc = "MHAI_Measurement"
  FCDA
    ldInst = "MET"
    lnClass = "MHAI"
    fc = "MX"
    lnInst = "1"
    doName = "ThdA"
    daName = "phsA.cVal.mag.f"
  FCDA
  FCDA
  FCDA
    ldInst = "MET"
    lnClass = "MHAI"
    fc = "MX"
    lnInst = "1"
    doName = "ThdPhV"
    daName = "phsA.cVal.mag.f"
  FCDA
  FCDA
```

```
<DataSet name="MHAI_Measurement" desc="MHAI_Measurement">
  <FCDA ldInst="MET" lnClass="MHAI" fc="MX" lnInst="1" doName="ThdA" daName="phsA.cVal.mag.f" />
  <FCDA ldInst="MET" lnClass="MHAI" fc="MX" lnInst="1" doName="ThdA" daName="phsB.cVal.mag.f" />
  <FCDA ldInst="MET" lnClass="MHAI" fc="MX" lnInst="1" doName="ThdA" daName="phsC.cVal.mag.f" />
  <FCDA ldInst="MET" lnClass="MHAI" fc="MX" lnInst="1" doName="ThdPhV" daName="phsA.cVal.mag.f" />
  <FCDA ldInst="MET" lnClass="MHAI" fc="MX" lnInst="1" doName="ThdPhV" daName="phsB.cVal.mag.f" />
  <FCDA ldInst="MET" lnClass="MHAI" fc="MX" lnInst="1" doName="ThdPhV" daName="phsC.cVal.mag.f" />
</DataSet>
```

Figure 5-16: MHAI Dataset with data objects for ThdA and ThdPhV for all three phases

The dataset for MMXU_Measurement (Figure 5-17) is configured similarly to MHAI_Measurement, only with three data object types (A, PhV and PF) instead of two.

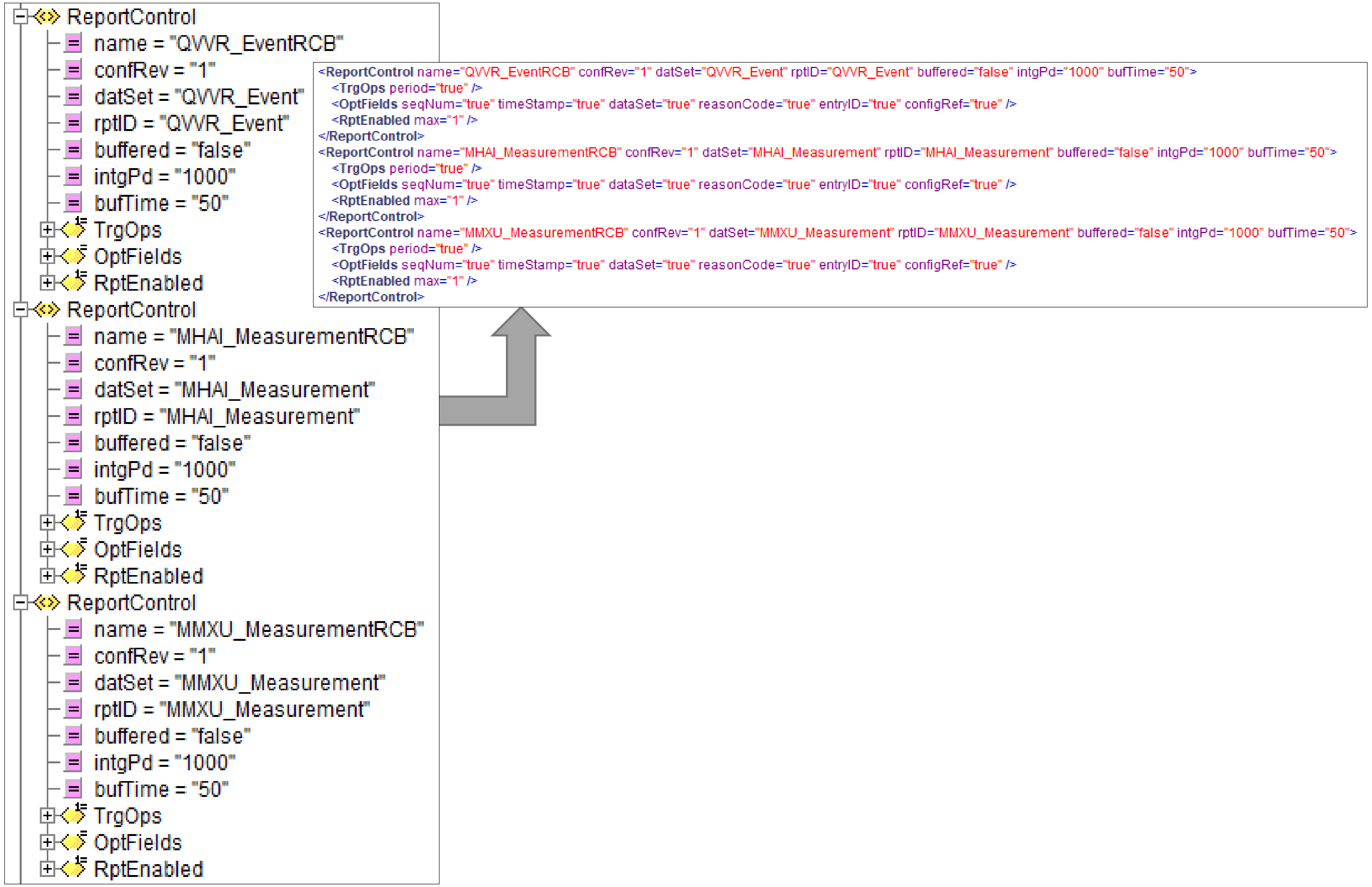


```
DataSet
  name = "MMXU_Measurement"
  desc = "MMXU_Measurement"
  FCDA
    ldInst = "MET"
    lnClass = "MMXU"
    fc = "MX"
    lnInst = "1"
    doName = "A"
    daName = "phsA.cVal.mag.f"
  FCDA
  FCDA
  FCDA
    ldInst = "MET"
    lnClass = "MMXU"
    fc = "MX"
    lnInst = "1"
    doName = "PhV"
    daName = "phsA.cVal.mag.f"
  FCDA
  FCDA
  FCDA
    ldInst = "MET"
    lnClass = "MMXU"
    fc = "MX"
    lnInst = "1"
    doName = "PF"
    daName = "phsA.cVal.mag.f"
  FCDA
  FCDA
```

```
<DataSet name="MMXU_Measurement" desc="MMXU_Measurement">
  <FCDA ldInst="MET" lnClass="MMXU" fc="MX" lnInst="1" doName="A" daName="phsA.cVal.mag.f" />
  <FCDA ldInst="MET" lnClass="MMXU" fc="MX" lnInst="1" doName="A" daName="phsB.cVal.mag.f" />
  <FCDA ldInst="MET" lnClass="MMXU" fc="MX" lnInst="1" doName="A" daName="phsC.cVal.mag.f" />
  <FCDA ldInst="MET" lnClass="MMXU" fc="MX" lnInst="1" doName="PhV" daName="phsA.cVal.mag.f" />
  <FCDA ldInst="MET" lnClass="MMXU" fc="MX" lnInst="1" doName="PhV" daName="phsB.cVal.mag.f" />
  <FCDA ldInst="MET" lnClass="MMXU" fc="MX" lnInst="1" doName="PhV" daName="phsC.cVal.mag.f" />
  <FCDA ldInst="MET" lnClass="MMXU" fc="MX" lnInst="1" doName="PF" daName="phsA.cVal.mag.f" />
  <FCDA ldInst="MET" lnClass="MMXU" fc="MX" lnInst="1" doName="PF" daName="phsB.cVal.mag.f" />
  <FCDA ldInst="MET" lnClass="MMXU" fc="MX" lnInst="1" doName="PF" daName="phsC.cVal.mag.f" />
</DataSet>
```

Figure 5-17: MMXU Data Set

Figure 5-18 shows the report control blocks, which is not mandatory but is configured to enable metering device and to send reports if deemed necessary.

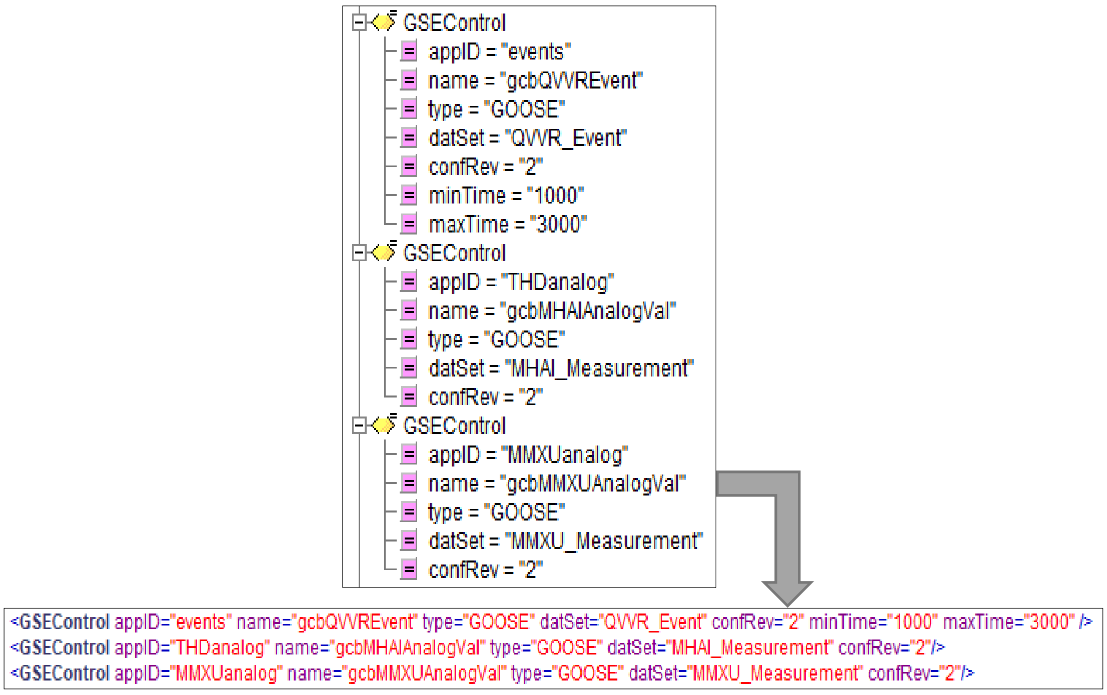


```
ReportControl
  name = "QVVR_EventRCB"
  confRev = "1"
  datSet = "QVVR_Event"
  rptID = "QVVR_Event"
  buffered = "false"
  intgPd = "1000"
  bufTime = "50"
  TrgOps
  OptFields
  RptEnabled
ReportControl
  name = "MHAI_MeasurementRCB"
  confRev = "1"
  datSet = "MHAI_Measurement"
  rptID = "MHAI_Measurement"
  buffered = "false"
  intgPd = "1000"
  bufTime = "50"
  TrgOps
  OptFields
  RptEnabled
ReportControl
  name = "MMXU_MeasurementRCB"
  confRev = "1"
  datSet = "MMXU_Measurement"
  rptID = "MMXU_Measurement"
  buffered = "false"
  intgPd = "1000"
  bufTime = "50"
  TrgOps
  OptFields
  RptEnabled
```

```
<ReportControl name="QVVR_EventRCB" confRev="1" datSet="QVVR_Event" rptID="QVVR_Event" buffered="false" intgPd="1000" bufTime="50">
  <TrgOps period="true" />
  <OptFields seqNum="true" timeStamp="true" dataSet="true" reasonCode="true" entryID="true" configRef="true" />
  <RptEnabled max="1" />
</ReportControl>
<ReportControl name="MHAI_MeasurementRCB" confRev="1" datSet="MHAI_Measurement" rptID="MHAI_Measurement" buffered="false" intgPd="1000" bufTime="50">
  <TrgOps period="true" />
  <OptFields seqNum="true" timeStamp="true" dataSet="true" reasonCode="true" entryID="true" configRef="true" />
  <RptEnabled max="1" />
</ReportControl>
<ReportControl name="MMXU_MeasurementRCB" confRev="1" datSet="MMXU_Measurement" rptID="MMXU_Measurement" buffered="false" intgPd="1000" bufTime="50">
  <TrgOps period="true" />
  <OptFields seqNum="true" timeStamp="true" dataSet="true" reasonCode="true" entryID="true" configRef="true" />
  <RptEnabled max="1" />
</ReportControl>
```

Figure 5-18: Report Control Blocks

The GOOSE Control block (GoCB) is vital for enabling GOOSE communications, and Figure 5-19 shows three GSEControl configurations which contain parameters for each individual GoCB defined in the SCL file. Each GoCB has an appID, a unique name ("name") which describe the specific instance of the GOOSE Control Block, a type specification ("type") which determines the type of data that is allowed to be sent.



```
GSEControl
  appID = "events"
  name = "gcbQVVREvent"
  type = "GOOSE"
  datSet = "QVVR_Event"
  confRev = "2"
  minTime = "1000"
  maxTime = "3000"
GSEControl
  appID = "THDanalog"
  name = "gcbMHAIAnalogVal"
  type = "GOOSE"
  datSet = "MHAI_Measurement"
  confRev = "2"
GSEControl
  appID = "MMXUanalog"
  name = "gcbMMXUAnalogVal"
  type = "GOOSE"
  datSet = "MMXU_Measurement"
  confRev = "2"
```

```
<GSEControl appID="events" name="gcbQVVREvent" type="GOOSE" datSet="QVVR_Event" confRev="2" minTime="1000" maxTime="3000" />
<GSEControl appID="THDanalog" name="gcbMHAIAnalogVal" type="GOOSE" datSet="MHAI_Measurement" confRev="2"/>
<GSEControl appID="MMXUanalog" name="gcbMMXUAnalogVal" type="GOOSE" datSet="MMXU_Measurement" confRev="2"/>
```

Figure 5-19: GSE Control Blocks for All Datasets

Additionally, the name of the dataset ("datSet") to be sent and finally the configuration revision ("confRev") which is a count that represents the number of times a specific dataset referenced by "datSet" has been changed. The GOOSE Control Blocks mentioned above have been defined according to Table 5-1.

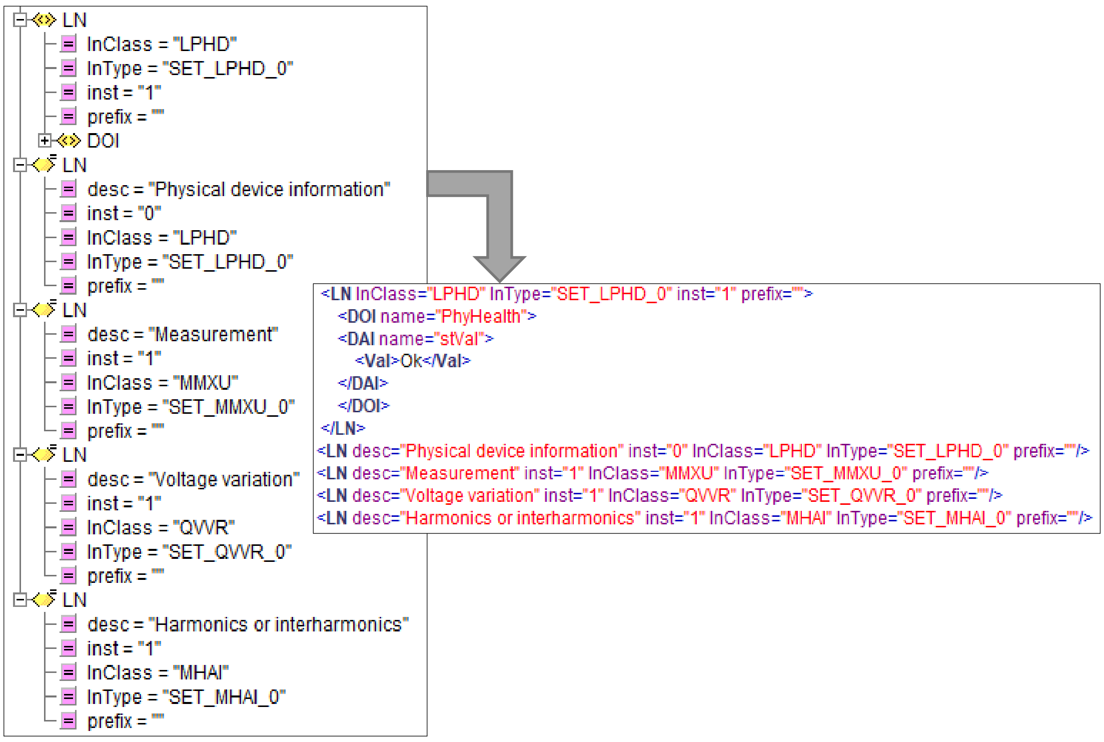
Table 5-1: Goose Control Block Information

appID	Unique Name	Type Spec.	Dataset Name	Config. Rev.
events	gcbQVVREvent	GOOSE	QVVR_Event	2
THDanalog	gcbMHAIAnalogVal	GOOSE	MHAI_Measurement	2
MMXUanalog	gcbMMXUAnalogVal	GOOSE	MMXU_Measurement	2

Figure 5-20 shows the logical node definitions for the logical nodes used for measurement, voltage variation events and harmonics and interharmonics measurements. A logical node definition for Physical device information is also present in this section. There are four logical node definitions in the ICD file:

- Physical device information: LPHD of type SET_LPHD_0,
- Measurements: MMXU of type SET_MMXU_0,
- Voltage variation events: QVVR of type SET_QVVR_0
- Harmonics and interharmonics measurements: MHAI of type SET_MHAI_0.

This section's primary purpose is to configure the number of instances of a logical node there are in the IED configuration, as well as the logical node type used for each logical node. The logical node type (lnType) is defined in the DataTypeTemplate. Figure 5-22 shows the aforementioned logical node types, which are discussed in the next section.



```
LN
├ lnClass = "LPHD"
├ lnType = "SET_LPHD_0"
├ inst = "1"
├ prefix = ""
⊞ DOI
LN
├ desc = "Physical device information"
├ inst = "0"
├ lnClass = "LPHD"
├ lnType = "SET_LPHD_0"
└ prefix = ""
LN
├ desc = "Measurement"
├ inst = "1"
├ lnClass = "MMXU"
├ lnType = "SET_MMXU_0"
└ prefix = ""
LN
├ desc = "Voltage variation"
├ inst = "1"
├ lnClass = "QVVR"
├ lnType = "SET_QVVR_0"
└ prefix = ""
LN
├ desc = "Harmonics or interharmonics"
├ inst = "1"
├ lnClass = "MHAI"
├ lnType = "SET_MHAI_0"
└ prefix = ""
```

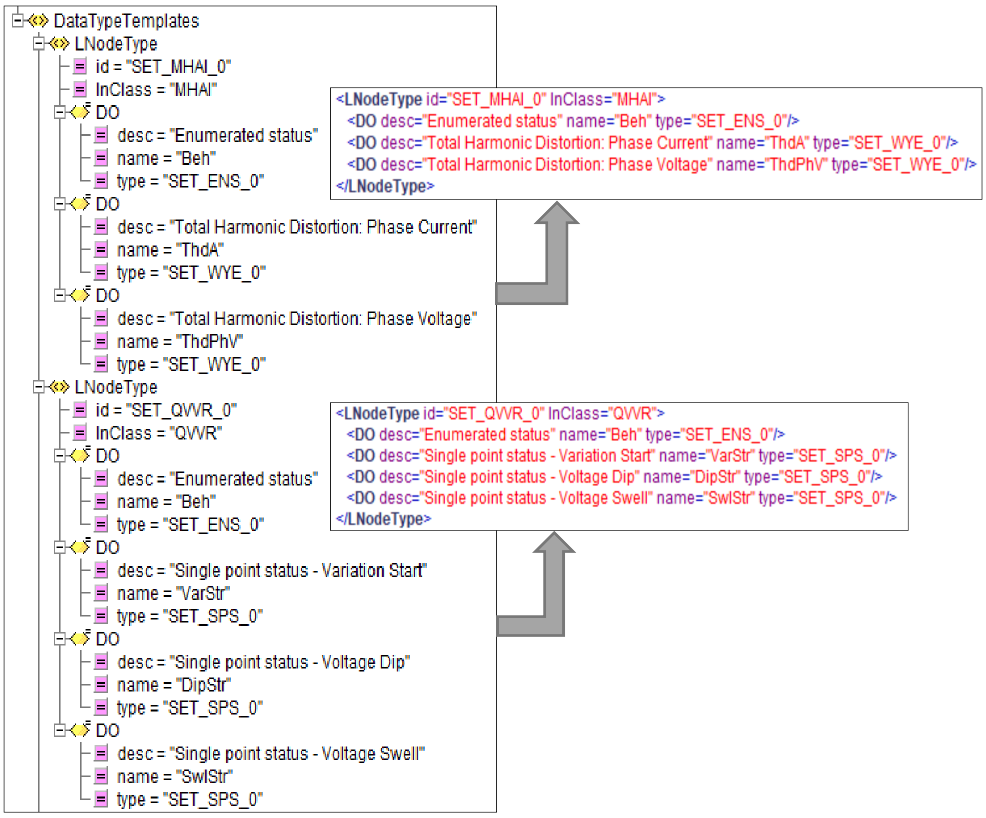
```
<LN lnClass="LPHD" lnType="SET_LPHD_0" inst="1" prefix="">
  <DOI name="PhyHealth">
  <DAI name="stVal">
    <Val>Ok</Val>
  </DAI>
  </DOI>
</LN>
<LN desc="Physical device information" inst="0" lnClass="LPHD" lnType="SET_LPHD_0" prefix=""/>
<LN desc="Measurement" inst="1" lnClass="MMXU" lnType="SET_MMXU_0" prefix=""/>
<LN desc="Voltage variation" inst="1" lnClass="QVVR" lnType="SET_QVVR_0" prefix=""/>
<LN desc="Harmonics or interharmonics" inst="1" lnClass="MHAI" lnType="SET_MHAI_0" prefix=""/>
```

Figure 5-20: LPHD. MMXU, QVVR and MHAI Logical Node Definitions

5.5.4. DataTypeTemplate

This section describes the various data type templates required for datasets. The data type template section contains all of the data type definitions used in the IED sections. Figures 5-22 to 5-26 illustrate the data type templates for both the Logical Nodes and the Data Objects. The LNodeType (Logical Node Type) uses the DOTypes in Figures 5-22 to 5-26 to define the logical node-specific data objects.

Figure 5-21 shows examples of LNodeType definitions for MHAI and QVVR; each LNodeType contains several DO (data objects). In each data object, a description, name and type are defined. The type parameter is defined in DOType, as shown in Figures 5-23 to 5-27.



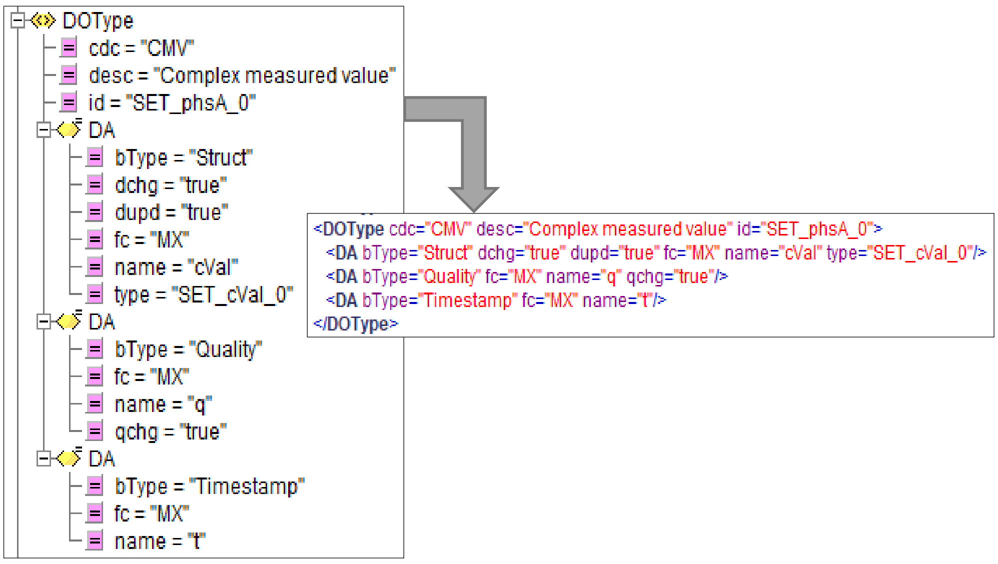
```
DataTypeTemplates
  LNodeType
    id = "SET_MHAI_0"
    lnClass = "MHAI"
    DO
      desc = "Enumerated status"
      name = "Beh"
      type = "SET_ENS_0"
    DO
      desc = "Total Harmonic Distortion: Phase Current"
      name = "ThdA"
      type = "SET_WYE_0"
    DO
      desc = "Total Harmonic Distortion: Phase Voltage"
      name = "ThdPhV"
      type = "SET_WYE_0"
  LNodeType
    id = "SET_QVVR_0"
    lnClass = "QVVR"
    DO
      desc = "Enumerated status"
      name = "Beh"
      type = "SET_ENS_0"
    DO
      desc = "Single point status - Variation Start"
      name = "VarStr"
      type = "SET_SPS_0"
    DO
      desc = "Single point status - Voltage Dip"
      name = "DipStr"
      type = "SET_SPS_0"
    DO
      desc = "Single point status - Voltage Swell"
      name = "SwlStr"
      type = "SET_SPS_0"
```

```xml
<LNodeType id="SET_MHAI_0" lnClass="MHAI">
  <DO desc="Enumerated status" name="Beh" type="SET_ENS_0"/>
  <DO desc="Total Harmonic Distortion: Phase Current" name="ThdA" type="SET_WYE_0"/>
  <DO desc="Total Harmonic Distortion: Phase Voltage" name="ThdPhV" type="SET_WYE_0"/>
</LNodeType>
```

```xml
<LNodeType id="SET_QVVR_0" lnClass="QVVR">
  <DO desc="Enumerated status" name="Beh" type="SET_ENS_0"/>
  <DO desc="Single point status - Variation Start" name="VarStr" type="SET_SPS_0"/>
  <DO desc="Single point status - Voltage Dip" name="DipStr" type="SET_SPS_0"/>
  <DO desc="Single point status - Voltage Swell" name="SwlStr" type="SET_SPS_0"/>
</LNodeType>
```

Figure 5-21: Data Object Definitions for MHAI and QVVR

In Figure 5-22, the DOType "complex measured value" or CMV, this data object contains three data attributes, namely Struct, Quality and Timestamp. A CMV is used for each phase measurement in both the appropriate MHAI and MMXU measurements. The phase-specific DO Types have given an ID "SET_phsx_0", the data attribute type "SET_cVal_0" is as shown in Figure 5-27.

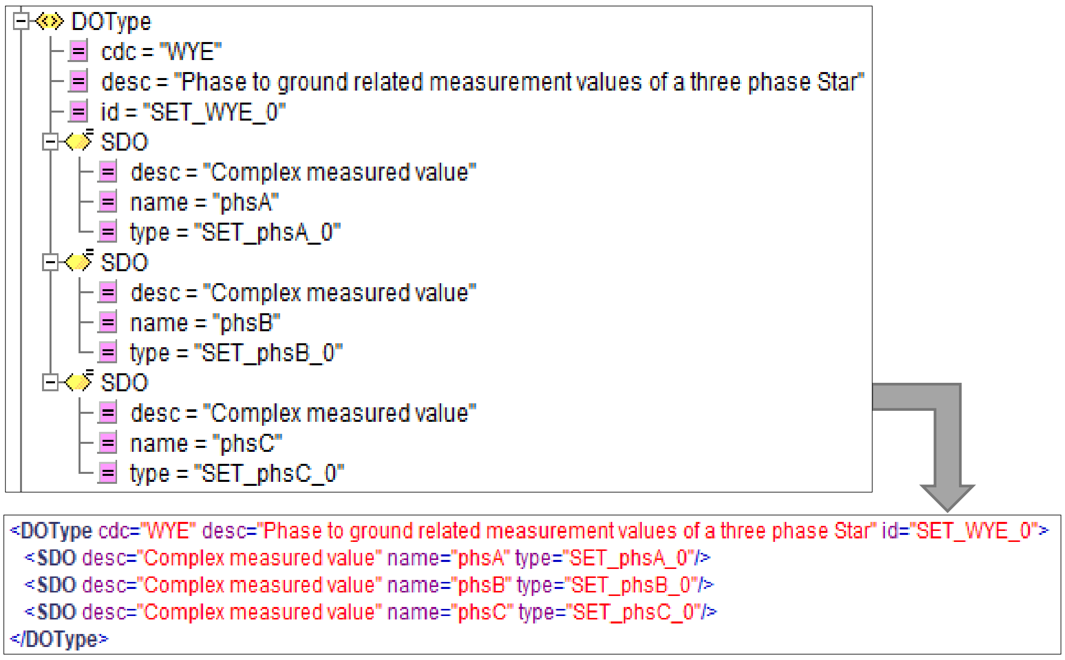


```
DOType
  cdc = "CMV"
  desc = "Complex measured value"
  id = "SET_phsA_0"
  DA
    bType = "Struct"
    dchg = "true"
    dupd = "true"
    fc = "MX"
    name = "cVal"
    type = "SET_cVal_0"
  DA
    bType = "Quality"
    fc = "MX"
    name = "q"
    qchg = "true"
  DA
    bType = "Timestamp"
    fc = "MX"
    name = "t"
```

```xml
<DOType cdc="CMV" desc="Complex measured value" id="SET_phsA_0">
  <DA bType="Struct" dchg="true" dupd="true" fc="MX" name="cVal" type="SET_cVal_0"/>
  <DA bType="Quality" fc="MX" name="q" qchg="true"/>
  <DA bType="Timestamp" fc="MX" name="t"/>
</DOType>
```

Figure 5-22: Complex measured value data object type

Figure 5-23 is an example of the WYE Data Object type. WYE comprises of elements of an already defined data object element, in this case, the aforementioned “SET_phsx_0” CMVs are used in WYE. This way, the DO Type WYE can be used in any phase-to-ground measurement as it is required.

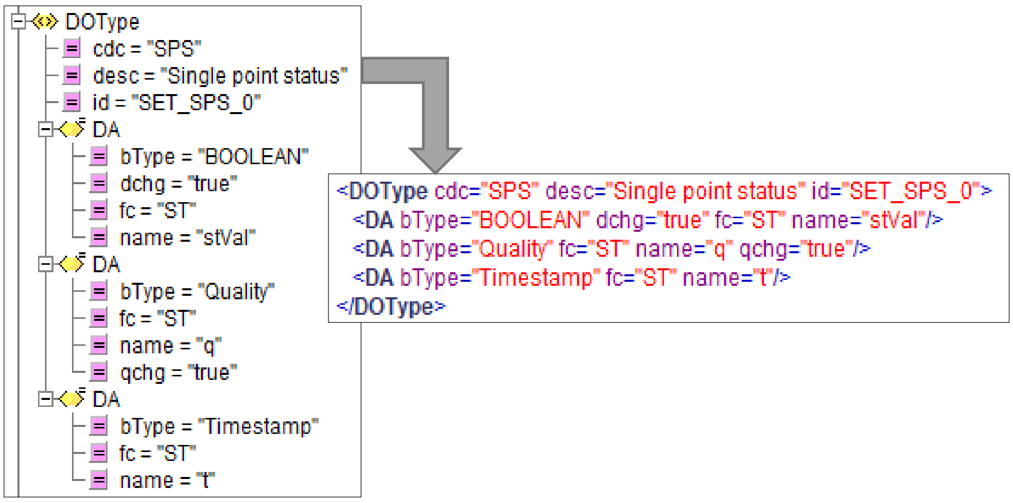


```
DOType
  cdc = "WYE"
  desc = "Phase to ground related measurement values of a three phase Star"
  id = "SET_WYE_0"
  SDO
    desc = "Complex measured value"
    name = "phsA"
    type = "SET_phsA_0"
  SDO
    desc = "Complex measured value"
    name = "phsB"
    type = "SET_phsB_0"
  SDO
    desc = "Complex measured value"
    name = "phsC"
    type = "SET_phsC_0"
```

```
<DOType cdc="WYE" desc="Phase to ground related measurement values of a three phase Star" id="SET_WYE_0">
  <SDO desc="Complex measured value" name="phsA" type="SET_phsA_0"/>
  <SDO desc="Complex measured value" name="phsB" type="SET_phsB_0"/>
  <SDO desc="Complex measured value" name="phsC" type="SET_phsC_0"/>
</DOType>
```

Figure 5-23: Data object type WYE

In Figure 5-24 the DOType “single point status” or SPS, this data object contains three data attributes, namely BOOLEAN, Quality and Timestamp. SPS is generally used for boolean event statuses; in this case, SPS is used in QVVR for the VarStr, DipStr and SwlStr status indications.

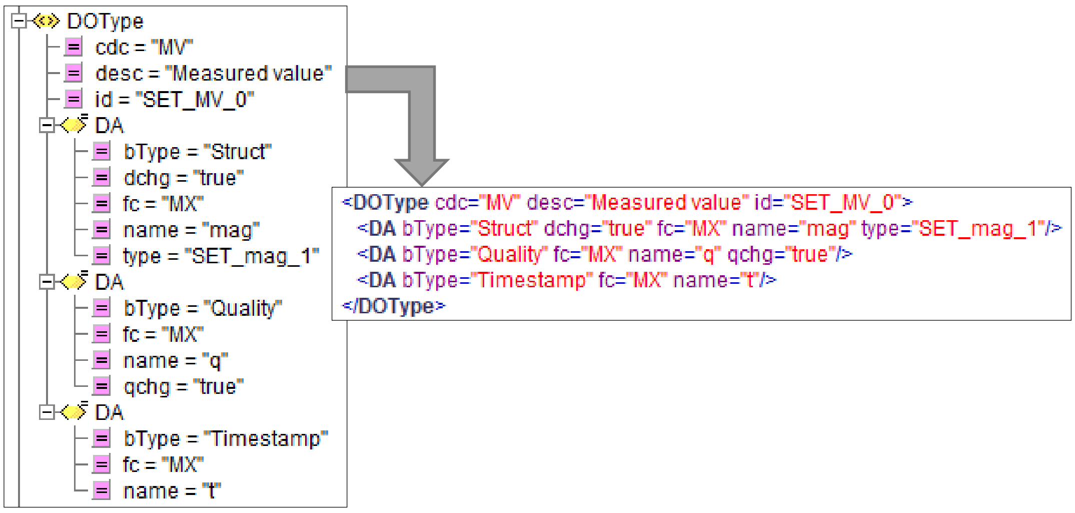


```
DOType
  cdc = "SPS"
  desc = "Single point status"
  id = "SET_SPS_0"
  DA
    bType = "BOOLEAN"
    dchg = "true"
    fc = "ST"
    name = "stVal"
  DA
    bType = "Quality"
    fc = "ST"
    name = "q"
    qchg = "true"
  DA
    bType = "Timestamp"
    fc = "ST"
    name = "t"
```

```
<DOType cdc="SPS" desc="Single point status" id="SET_SPS_0">
  <DA bType="BOOLEAN" dchg="true" fc="ST" name="stVal"/>
  <DA bType="Quality" fc="ST" name="q" qchg="true"/>
  <DA bType="Timestamp" fc="ST" name="t"/>
</DOType>
```

Figure 5-24: Data Object Type Single Point Status

In Figure 5-25, the DOType “Measured value” or MV, this data object contains three data attributes, namely Struct, Quality and Timestamp. Unlike CMV, MV is used for single measured values; as such, the “Struct” data attribute is of type “SET_mag_1” which is a FLOAT32 value.

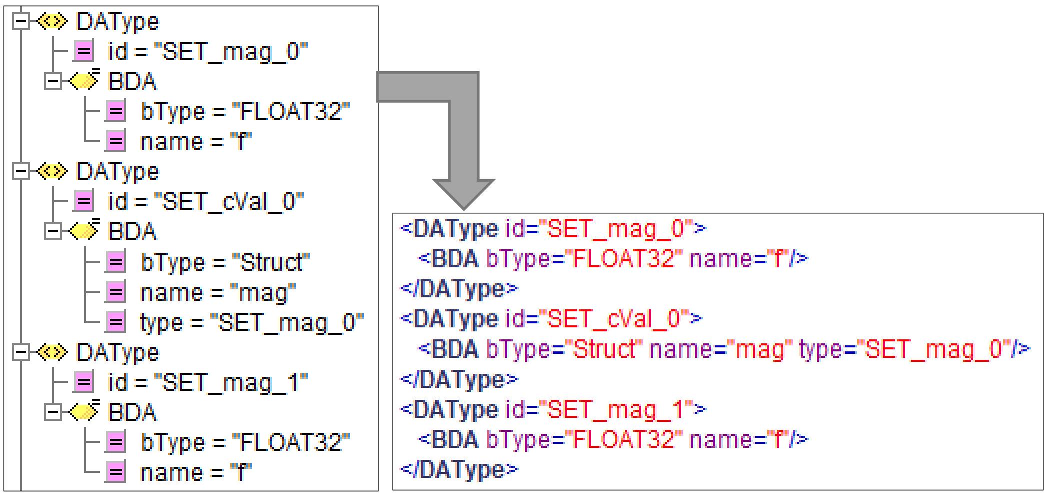


```
DOType
  cdc = "MV"
  desc = "Measured value"
  id = "SET_MV_0"
  DA
    bType = "Struct"
    dchg = "true"
    fc = "MX"
    name = "mag"
    type = "SET_mag_1"
  DA
    bType = "Quality"
    fc = "MX"
    name = "q"
    qchg = "true"
  DA
    bType = "Timestamp"
    fc = "MX"
    name = "t"
```

```
<DOType cdc="MV" desc="Measured value" id="SET_MV_0">
  <DA bType="Struct" dchg="true" fc="MX" name="mag" type="SET_mag_1"/>
  <DA bType="Quality" fc="MX" name="q" qchg="true"/>
  <DA bType="Timestamp" fc="MX" name="t"/>
</DOType>
```

Figure 5-25: Data object type for “Measured Value”

Finally, the data attribute types are also defined in the data type template section. Figure 5-26 shows the three data attribute types defined in the ICD file, namely SET_mag_0, SET_cVal_0 and SET_mag_1. SET_mag_0 and SET_mag_1 uses FLOAT32 as the basic type (bType) whilst “Struct” is the bType of SET_cVal_0. Furthermore, the type of Struct used in SET_cVal_0 is SET_mag_0, which uses bType FLOAT32.



```
DAType
  id = "SET_mag_0"
  BDA
    bType = "FLOAT32"
    name = "f"
DAType
  id = "SET_cVal_0"
  BDA
    bType = "Struct"
    name = "mag"
    type = "SET_mag_0"
DAType
  id = "SET_mag_1"
  BDA
    bType = "FLOAT32"
    name = "f"
```

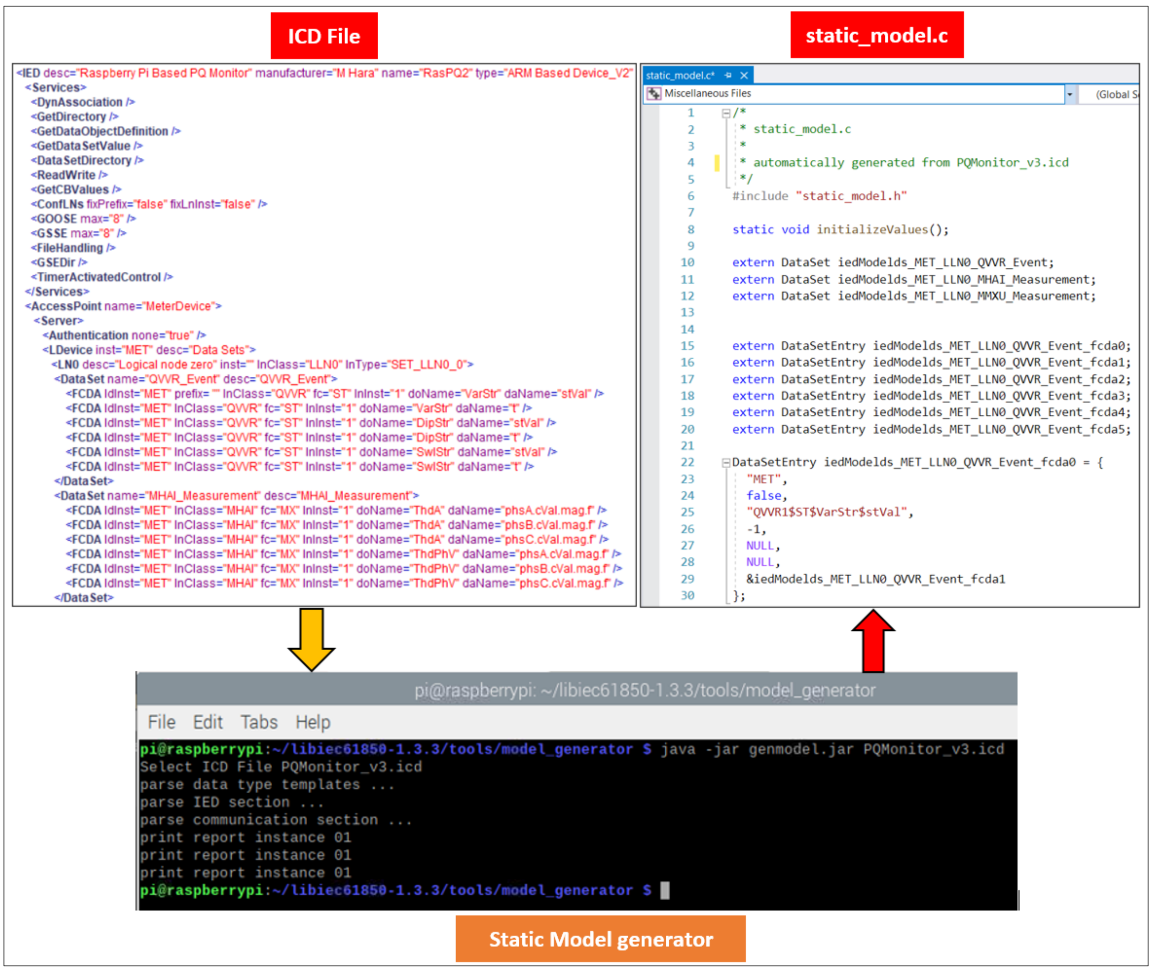
```
<DAType id="SET_mag_0">
  <BDA bType="FLOAT32" name="f"/>
</DAType>
<DAType id="SET_cVal_0">
  <BDA bType="Struct" name="mag" type="SET_mag_0"/>
</DAType>
<DAType id="SET_mag_1">
  <BDA bType="FLOAT32" name="f"/>
</DAType>
```

Figure 5-26: Data Attribute Types

5.6. Static Model generation using an SCL file

The written ICD file is used to generate a static model that may be used as an IEC 61850 configuration on the Raspberry Pi. The static model is a conversion of the XML-based ICD document to standard C code, this allows the device to access the datasets and logical node definitions programmatically, to more effectively integrate the IEC 61850 portion with the Raspberry Pi software and hardware. Appendix C details the entire ICD file in XML format. Static model generation is achieved using the included java program, genmodel.jar.

Figure 5-27 shows the ICD file selected in the static model generator, executing the program results in the generation of the static model, which results in static_model.c, the command used to run the generation procedure is “java -jar genmodel.jar scl_file_name.icd”. The generated C code is available in static_model.c, whose function and macro declarations can be accessed by including the static_model.h header file in the main program. Generation of the static model requires the ICD file and the genmodel.jar program file to be in the same folder.



```
<IED desc="Raspberry Pi Based PQ Monitor" manufacturer="M Hara" name="RasPQ2" type="ARM Based Device_V2"
  <Services>
   <DynAssociation />
   <GetDirectory />
   <GetDataObjectDefinition />
   <GetDataSetValue />
   <DataSetDirectory />
   <ReadWrite />
   <GetCBValues />
   <ConfLNs fixPrefix="false" fixLnInst="false" />
   <GOOSE max="8" />
   <GSSE max="8" />
   <FileHandling />
   <GSEDir />
   <TimerActivatedControl />
  </Services>
  <AccessPoint name="MeterDevice">
   <Server>
     <Authentication none="true" />
     <LDevice inst="MET" desc="Data Sets">
       <LN0 desc="Logical node zero" inst="" lnClass="LLN0" lnType="SET_LLN0_0">
       <DataSet name="QVVR_Event" desc="QVVR_Event">
          <FCDA ldInst="MET" prefix= "" lnClass="QVVR" fc="ST" lnInst="1" doName="VarStr" daName="stVal" />
          <FCDA ldInst="MET" lnClass="QVVR" fc="ST" lnInst="1" doName="VarStr" daName="t" />
          <FCDA ldInst="MET" lnClass="QVVR" fc="ST" lnInst="1" doName="DipStr" daName="stVal" />
          <FCDA ldInst="MET" lnClass="QVVR" fc="ST" lnInst="1" doName="DipStr" daName="t" />
          <FCDA ldInst="MET" lnClass="QVVR" fc="ST" lnInst="1" doName="SwlStr" daName="stVal" />
          <FCDA ldInst="MET" lnClass="QVVR" fc="ST" lnInst="1" doName="SwlStr" daName="t" />
       </DataSet>
       <DataSet name="MHAI_Measurement" desc="MHAI_Measurement">
          <FCDA ldInst="MET" lnClass="MHAI" fc="MX" lnInst="1" doName="ThdA" daName="phsA.cVal.mag.f" />
          <FCDA ldInst="MET" lnClass="MHAI" fc="MX" lnInst="1" doName="ThdA" daName="phsB.cVal.mag.f" />
          <FCDA ldInst="MET" lnClass="MHAI" fc="MX" lnInst="1" doName="ThdA" daName="phsC.cVal.mag.f" />
          <FCDA ldInst="MET" lnClass="MHAI" fc="MX" lnInst="1" doName="ThdPhV" daName="phsA.cVal.mag.f" />
          <FCDA ldInst="MET" lnClass="MHAI" fc="MX" lnInst="1" doName="ThdPhV" daName="phsB.cVal.mag.f" />
          <FCDA ldInst="MET" lnClass="MHAI" fc="MX" lnInst="1" doName="ThdPhV" daName="phsC.cVal.mag.f" />
       </DataSet>
```

```
/*
 * static_model.c
 *
 * automatically generated from PQMonitor_v3.icd
 */
#include "static_model.h"

static void initializeValues();

extern DataSet iedModelds_MET_LLN0_QVVR_Event;
extern DataSet iedModelds_MET_LLN0_MHAI_Measurement;
extern DataSet iedModelds_MET_LLN0_MMXU_Measurement;


extern DataSetEntry iedModelds_MET_LLN0_QVVR_Event_fcda0;
extern DataSetEntry iedModelds_MET_LLN0_QVVR_Event_fcda1;
extern DataSetEntry iedModelds_MET_LLN0_QVVR_Event_fcda2;
extern DataSetEntry iedModelds_MET_LLN0_QVVR_Event_fcda3;
extern DataSetEntry iedModelds_MET_LLN0_QVVR_Event_fcda4;
extern DataSetEntry iedModelds_MET_LLN0_QVVR_Event_fcda5;

DataSetEntry iedModelds_MET_LLN0_QVVR_Event_fcda0 = {
  "MET",
  false,
  "QVVR1$ST$VarStr$stVal",
  -1,
  NULL,
  NULL,
  &iedModelds_MET_LLN0_QVVR_Event_fcda1
};
```

```
pi@raspberrypi:~/libiec61850-1.3.3/tools/model_generator $ java -jar genmodel.jar PQMonitor_v3.icd
Select ICD File PQMonitor_v3.icd
parse data type templates ...
parse IED section ...
parse communication section ...
print report instance 01
print report instance 01
print report instance 01
pi@raspberrypi:~/libiec61850-1.3.3/tools/model_generator $
```

Figure 5-27: Static model generation and result

5.7. Conclusion

In this chapter, a range of hardware and software platforms are examined to determine their suitability to this project. The hardware chosen to run the data acquisition as well as IEC 61850 protocol stack is the Raspberry Pi; selected primarily due to its low price, ease of use, open-source operating system compatibility, the presence of Ethernet communications and generally powerful software. An open-source IEC 61850 library known as LibIEC61850 is also chosen as the protocol provider and source code base due to its suitability for DER projects and flexibility. The open-source IEC 61850 library is configured on both the Raspberry Pi and a Windows PC, and the source code for MMS and GOOSE communications are tested and ICD file created and used to generate a static model. At this point, it is possible to communicate with a remote client using both MMS and GOOSE by initially entering data manually. Once the data acquisition system is implemented, data may also be introduced by injecting current and voltage signals using the OMICRON test set.

Chapter 6 discusses implementing the data acquisition system using the ADE9000 AFE, then transmitting measured physical values using MMS and GOOSE. A schematic for the data acquisition system is created with all of the necessary circuitry. The data registers mapped to the modelled IEC 61850 data objects are configured, and C/C++ source code is written to interface the ADE9000 with the Raspberry Pi.

CHAPTER SIX

DATA ACQUISITION SYSTEM IMPLEMENTATION AND INTEGRATION

6.1. Introduction

The data acquisition system handles the acquisition, conversion, and processing of physical, electrical signals. The ADE9000 is the heart of the system, handling all of acquisition and conversion functions, the serial peripheral interface transfers the converted raw data to the Raspberry Pi. Figure 6-1 shows the focus area of this chapter within the system overview highlighted in green. The data acquisition and IEC 61850 protocol stack form a large bulk of the overall system, with the implemented modules within this chapter comprising of the system modules mentioned earlier. This chapter describes the design, implementation, and integration of the data acquisition system and relevant protocol stack components. Firstly, the design of the hardware, including circuitry and component selection of the PCB, is discussed. Secondly, the ADE9000 registers are configured and calibrated. Then, GOOSE and MMS communications are integrated with the data acquisition and conversion source code to transmit the measured values.

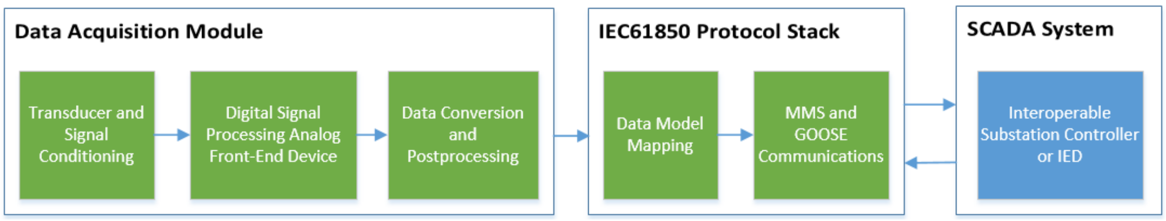

Data Acquisition Module
Transducer and Signal Conditioning
Digital Signal Processing Analog Front-End Device
Data Conversion and Postprocessing
IEC61850 Protocol Stack
Data Model Mapping
MMS and GOOSE Communications
SCADA System
Interoperable Substation Controller or IED

Figure 6-1: Chapter 6 Focus Areas

6.2. Data Acquisition hardware overview

As mentioned in Chapter 4, the data acquisition system has been designed with the ADE9000 analogue front end in mind. Several constraints were also identified, such as the measurement of primarily low voltage installations according to the project scope and the requirement of SPI communications on the Raspberry Pi and ADE9000. As such, the general makeup of the system consists of a large number of components designed to operate within the limitations as mentioned earlier.

The data acquisition system consists of 4 main components or modules, namely:

1. The voltage and current sensor circuits for interfacing between the voltage transformer, current transformer and the ADE9000's integrated analogue-to-digital converters.
2. ADE9000 analogue front end with integrated ADCs, digital signal processing functions and SPI communications functions.

3. Isolation and DC to DC conversion circuit for safe isolation between the ADE9000 and the Raspberry Pi; and interfacing pins for data transmission to and from the Isolation circuits and GPIO pins on the Raspberry Pi.
4. Data modelling and mapping of IEC 61850-7-4 data objects and functions to ADE9000 data registers. Communications protocols for data transmission using the LibIEC61850 open-source software library.

Figure 6-2 illustrates in block form, the four components described above and the resultant system when combined. The voltage and current sensing interface consist of a voltage divider circuit and a burden resistor circuit. The voltage divider resistor circuit is used to lower the AC voltage from the secondary side of the VT or 230V mains to a value that is within the full-scale amount of the analogue to digital converter. The current sensing circuit consists of two centre-tapped burden resistors to generate a differential, antiphase signal from the CT output current and an antialiasing RC filter. In the ADE9000, the ADC receives the converted signals and converts them to digital values according to predefined full-scale codes. The digital values are sent to the digital signal processing functions which handle the calculation and processing of various predefined indices and stores the results in data registers. Finally, the ADUM6404 and ADUM4151 provide digital isolation for serial peripheral interface data transferred to and from the Raspberry Pi. Isolation is essential to protection-sensitive devices as well as to prevent electrocution from the 230 V input.

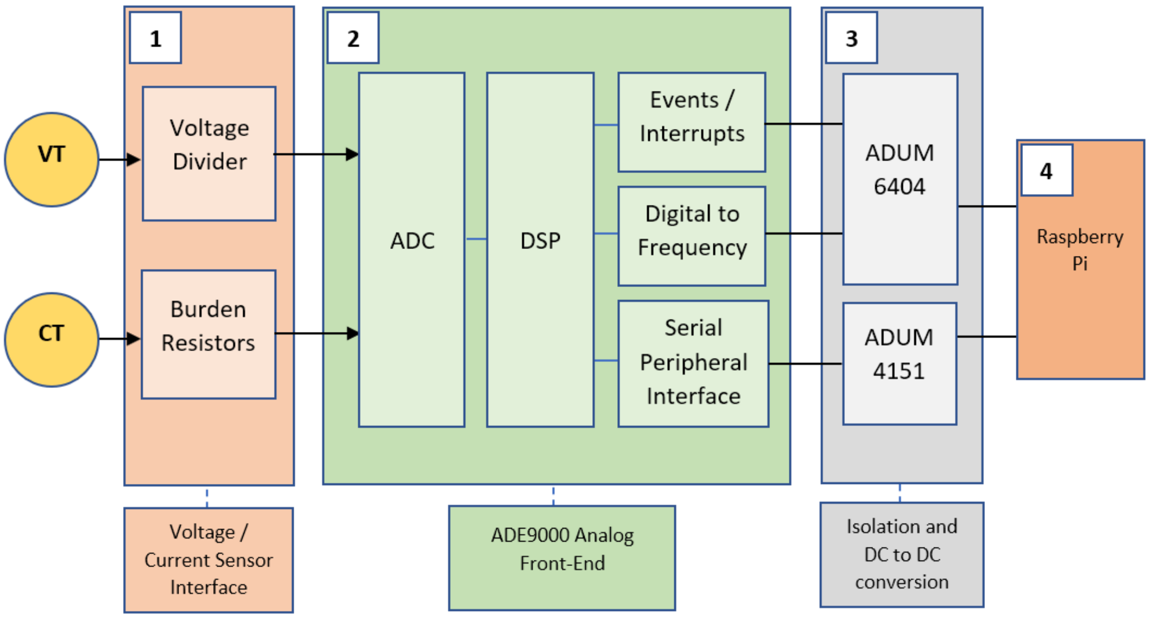


Figure 6-2: Data Acquisition System Block Diagram

6.3. Interfacing hardware design considerations

In the previous sub-section, the basic functionality of the voltage and current sensing circuits as well as the digital isolation circuits was briefly described. The two circuits mentioned above form a significant portion of the individual hardware components present on the data acquisition system. With this consideration in mind, this sub-section explores the specific design considerations of the voltage and current sensing circuits.

Component selection for both the voltage and current sensor circuits is determined by the end user's needs as well as the maximum input ratings of the ADCs. The voltage channels use resistor dividers to lower the input voltage to a value within the full-scale input range of the ADC. Although the ADE9000 ADC has a differential full-scale input range of +-1 V peak and +-0.707 Vrms, the input on the pins may not exceed 0.6 V. Assuming a 230 V input, a voltage divider with a summated resistor rating of 1MΩ would yield an input of 0.230 V using the following equation

$$V_{ADC} = \frac{V_{in}}{R_d \times 0.001} \tag{6.10}$$

Where:

V_{ADC} = Voltage at the ADC input

V_{in} = Voltage from mains or VT

R_d = Resistor divider value

Figure 6-3 shows the abovementioned resistor divider circuit for the A-phase voltage channel as designed for the data acquisition board. An 800kΩ resistor divider is connected to the live, yielding a maximum ADC input value of 0.2875V (VAP) assuming a 230 VAC input (VIN-1) signal. A lower resistor divider value was chosen to maintain a higher resolution at low voltage V_{ADC}. An RC filter is then created by placing a 22nF (C100) capacitor between VAP and ground (GND). This is useful for eliminating phase errors between the current and voltage signals.

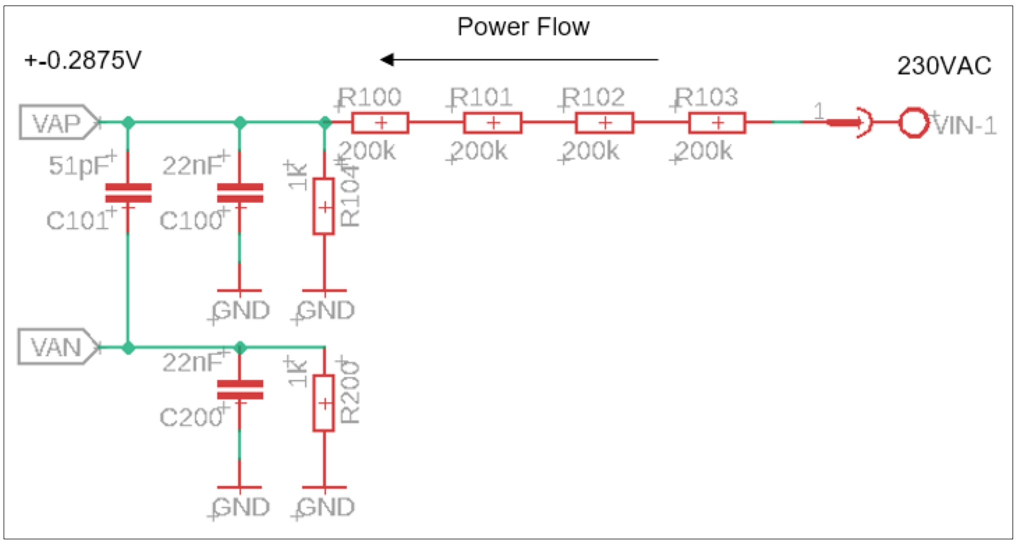


Figure 6-3: Resistor Divider Circuit

The current sensing circuits make use of burden resistors connected across the secondary side to produce an output voltage that is proportional to the resistor value. This value is based on the current following through the current transformer; for example, a CT with a 1:10 turns ratio produces a 10:1 current ratio. Using Ohm's law, if 1A on the primary side of the transformer has 0.1A on the secondary, then a burden resistor of 1Ω produces 0.1 V per amp across it. Increasing the resistor value to 10Ω yields 1 V per amp, and 100Ω have 10V and so forth. It is recommended that the ADE9000's burden resistor is centre tapped, as shown in Figure 6-4.

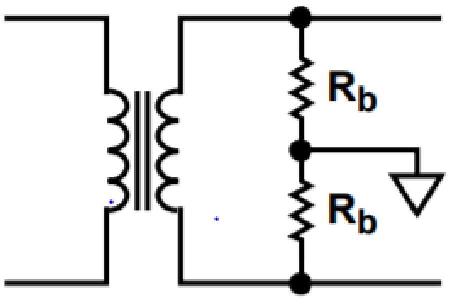


Figure 6-4: Centre Tapped Burden Resistor (Analog Devices Inc., 2017a)

The following equations can be used for determining the value of the burden resistor:

Firstly, two variables, x and y need to be resolved, the following equation defines the y variable:

$$y = \frac{V_{ADC}}{2} \tag{6.2}$$

Where:

V_{ADC} = Maximum ADC reference voltage in Volts

The following equation determines the x variable:

$$x = \frac{I_{RMS}}{CT_{Turns}} \tag{6.3}$$

Where:

I_{RMS} = Maximum RMS current rating of the CT

CT_{Turns} = number of turns in the current transformers

Then finally, the ratio of y over x gives us our burden resistor value:

$$\frac{y}{x} = 2R_b \tag{6.4}$$

Where:

R_b = Burden resistor value

The value of R_b is a summation of the burden resistance. Since the burden resistor is centre-tapped, R_b must be divided by 2 to determine the value of each resistor. Equations (6.2) and (6.3) can be combined to derive the following formula:

$$R_b = \frac{V_{ADC} \times CT_{Turns}}{2 \times I_{RMS}} \tag{6.5}$$

Figure 6-5 shows the current sensing circuit designed for the data acquisition board. The design is based on the SCT013-000 split-core CT which has a maximum current rating of 100A (I_{RMS}) on the primary for a 50mA output on the secondary with a transformer ratio of 2000:1 (CT_{Turns}). With the full-scale input range of the ADE9000 ADC being 0.707V (V_{ADC}), the parameters can be entered into equation (6.5) to yield an ideal burden resistor value of 7.07Ω. A resistor with values of 5.1Ω each was chosen to create a centre tapped burden resistor with a combined value of 10.2Ω. A higher burden resistor value was selected to increase the input voltage at the ADC and improve the signal to noise ratio. Finally, an RC filter consisting of a 1kΩ resistor and a 22nF capacitor are also present to provide attenuation of out of band signal near the clock frequency of the ADC modulator (2.048 MHz).

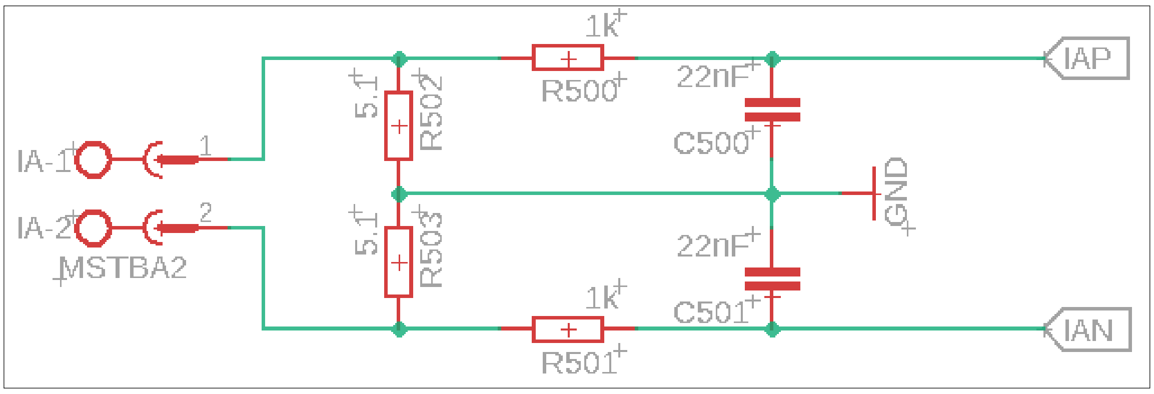


Figure 6-5: Current Sensing Circuit with centre-tapped Burden Resistors

Digital isolation is the last design consideration that is discussed. Digital isolation is used to isolate the signals between two “zones”, “live” (dangerous) zone and “safe” zone. Digital isolator ICs are used to ensure adequate protection of sensitive components as well as to prevent possible electrocution as in the case of the data acquisition board, which interfaces directly with 230 VAC signals. The isolation requirements first need to be identified; this is done by determining the number of signals being transferred between devices along with the different types of signals. In the case of this project, the ADE9000 and its peripheral circuitry are the live zones, whilst the Raspberry Pi creates the “safe” zone. The ADE9000 exchanges data via SPI has particular interrupt and event pins and finally requires a supply voltage. It may be concluded that the aforementioned need to be isolated since they interact with the Raspberry Pi. Two devices were chosen to achieve isolation, the ADUM4151 and the ADUM6404.

Shown in Figure 6-6, the ADUM6404 is a general-purpose digital isolator with an integrated DC to DC converter. All event and interrupt pins with a 0 to 3.3V logic level, and the 5V supply from the Raspberry Pi are connected to this chip.

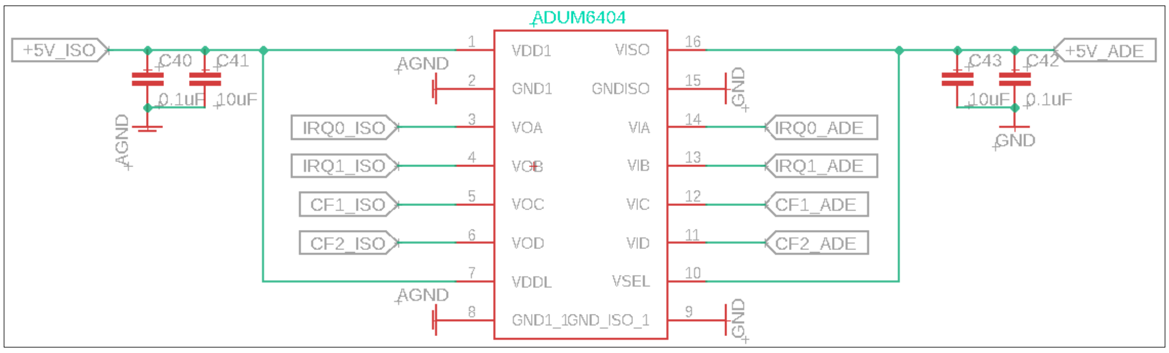


Figure 6-6: ADUM6404 Digital Isolator

In the final design, the 5V supply from the Raspberry Pi (+5V_ISO) is converted to 5V signal for the live side (+5V_ADE) and fed through the ADP122 3.3V linear regulator to provide the supply voltage for the ADE9000 (+3.3V_ADE).

The ADUM4151 digital isolator is shown in Figure 6-7, and it is designed explicitly for SPI signals; these chips are optimised for the timing requirements for SPI and support SPI clock rates of up to 17 MHz. The internal circuitry consists of encoders connected to decoders and separated control blocks.

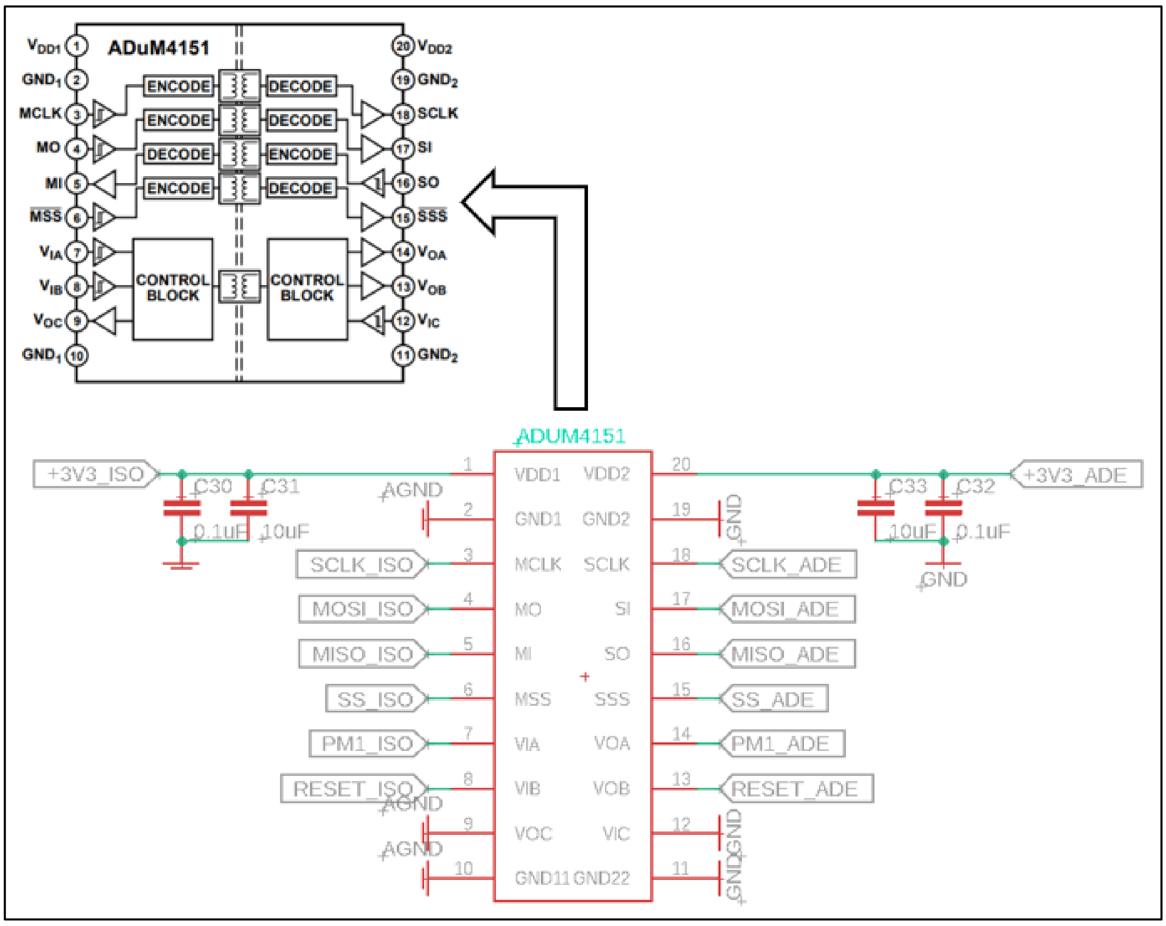


Figure 6-7: ADUM4151 Digital Isolator Schematic and Internal Circuitry

Appendix E is an Eagle CAD drawing of the final schematic of the complete data acquisition circuit include the sensor interfacing, digital isolation, DC to DC conversion and Raspberry Pi connection circuits. Figure 6-8 is a photograph of the final PCB design as manufactured, with all of the stages defined in Figure 6-2 identified by the coloured rectangular numbered outlines. The PCB design is based on the ADE9000 Arduino shield but adapted and modified for the Raspberry Pi (Analog Devices Inc., 2017b). The Eagle PCB Gerber design files are generated sent to a PCB manufacturing firm for production. With the physical components determined and the schematic completed, the last step is to calculate the measurement parameter of the data register values using the component values chosen in this section.

Figure 6-8 shows the final PCB design manufactured using the design in Appendix C. The functional block, as described in Figure 6-1, is highlighted and numbered accordingly.

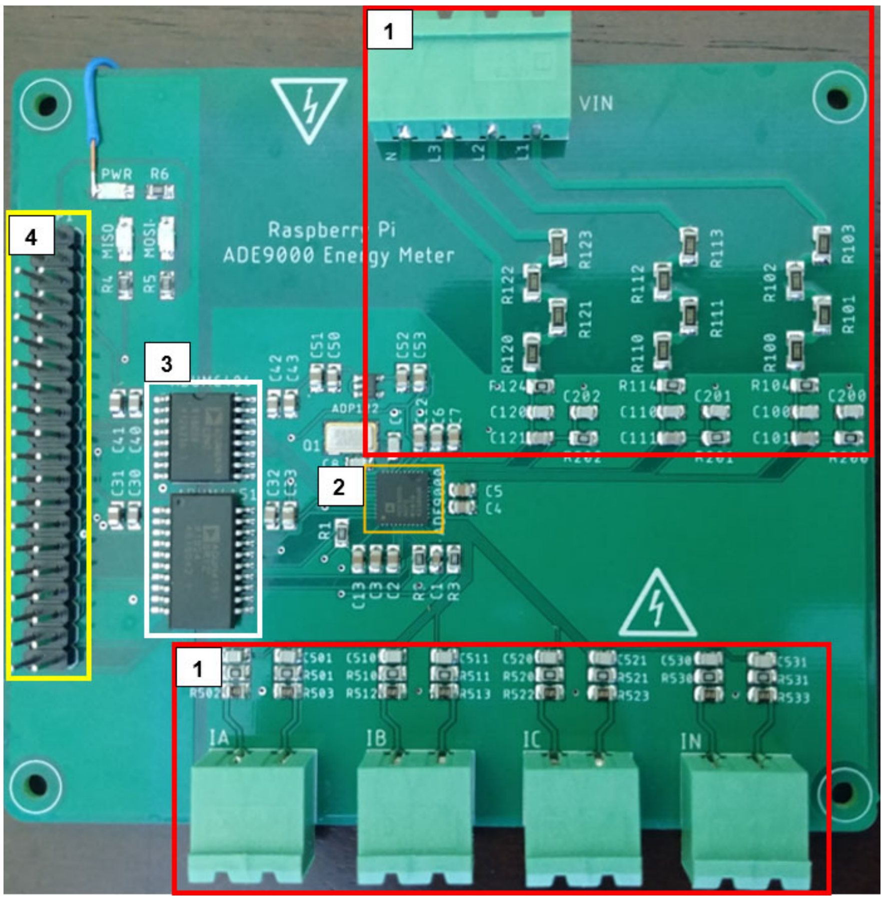


Figure 6-8: Final Data Acquisition System Hardware

6.4. Configuration of ADE9000 measurement registers

This subsection details the configuration procedure of the ADE9000 measurement registers. Hundreds of data registers are available in the ADE9000. Still, for this project, only the ones relevant to the objectives and requirements are configured, this means that functions such as energy accumulation, temperature and the waveform buffer are not configured, since they are not used within the scope of this project.

The following configuration procedures are adapted from the ADE9000 User Guide document (ADE9000-UG-1098) with the example component values substituted with the actual components used in the final design. Methods for configuring specific measurement parameters are examined, and additionally, the SPI communications protocol is configured from both the ADE9000 and Raspberry Pi’s perspective.

6.4.1. Measurement parameters and calibration

Table 6-1: Parameters used for calibration

System Parameter	**Value**
VN_{ominal}	230 V
$I_{Nominal}$	10A
Line Frequency	50Hz
CT Ratio	2000:1
Burden Resistor	10.2Ω
Voltage Divider	800MΩ
R2	1kΩ

Calibration values for ADE9000 measurement are determined using derived transfer functions. Transfer functions represent the relationship between the output signal and the input signal of a given control system; this transfer function should account for all possible input values. The following equation can determine the transfer function for the current sensor:

$$TF_A = R_b/C_{Turns} \tag{6.6}$$

Where:

TF_A = Current transfer function measured in V_{rms}/A_{rms}

R_b = Burden resistor value

CT_{Turns} = number of turns in the current transformers

Substituting with figures in Table 8 yields:

$$10.2/2000 = 0.0051\ V_{rms}/A_{rms}$$

The Voltage Transfer Function can be determined using the following equation:

$$TF_V = 1/(R_{vd} + R2) \tag{6.7}$$

Where:

TF_V = Voltage transfer function measured in V_{rms}/A_{rms}

R_{vd} = Voltage resistor value

R2 = Attenuation resistor

Substituting with figures in Table 8 yields:

$$1/(800+1) = 0.00125\ V_{rms}/A_{rms}$$

With the transfer functions determined, they can now be used to determine the input voltage at the ADC pins for specific nominal current and voltage by multiplying the transfer function with the nominal values given in Table 8.

Input at the current ADC pins:

$$0.0051 \times 10 = 0.051\ V_{rms}$$

Input at the voltage ADC pins:

$$0.00125 \times 230 = 0.2875\ V_{rms}$$

Next, the input voltages at both the ADC pins need to be expressed as a percentage of the full-scale RMS voltage. The ADC full-scale voltage at gain = 1 is 0.707 V_{rms}.

Nominal current as a percentage of full scale (I_{FSP}):

$$0.051/0.707 = 0.07214 = 7.214\%$$

Nominal voltage as a percentage of full scale (V_{FSP}):

$$0.2875/0.707 = 0.407 = 40.7\%$$

I_{FSP} and V_{FSP} can be used to calibrate gain and offset registers for RMS current, voltage and power. These registers are used to acquire the most accurate measurement data possible but require a highly accurate reference meter to achieve this level of accuracy. For this project, gain and offset calibration are not done as the conversion constant provides sufficient levels of accuracy for this implementation. The conversion constants are used to convert the ADE9000 register readings into physical quantities. Several full-scale ADC codes are required to convert the registries; these are shown in Table 6-2.

Table 6-2: Full-Scale ADC Code Parameters and their full-scale codes

Full-Scale ADC Codes Parameter	**Full-Scale Codes (Decimal)**
Total and Fundamental IRMS and VRMS	52,702,092
Total and Fundamental WATT, VAR, and VA	20,694,066
Fast RMS½	52,702,092
10 Cycle RMS/12 Cycle RMS	52,702,092
Resampled Data	18,196

Using the ADC codes in Table 6-2, the conversion constants for RMS current, RMS voltage, Watt, VAR and VA can be determined using some equations. The following equation is used for calculating the current conversion constant:

$$A_{CC} = \frac{I_{nominal} \times 10^6}{I_{FSP} \times FS_{RMSCode}} \tag{6.8}$$

Where:

A_{cc} = Current conversion constant measured in uArms/LSB

$I_{nominal}$ = Nominal current

I_{FSP} = Nominal current as a percentage of full-scale ADC voltage

$FS_{RMSCode}$ = Full Scale ADC Code

Using the previously calculated I_{FSP} value an $I_{nominal}$ of 10A, the A_{cc} can be calculated with an $FS_{RMSCode}$ of 52702092, which is the full-scale code for RMS current. Substituting the values into Equation (9) yields:

$$A_{CC} = \frac{10 \times 10^6}{0.07214 \times 52\,702\,092} = 2.6299943\ \mu A_{rms}/LSB$$

The equation for calculating the voltage conversion constant is nearly identical to the current conversion constant, the only difference being the usage of $V_{nominal}$ instead of $I_{nominal}$:

$$V_{CC} = \frac{V_{nominal} \times 10^6}{V_{FSP} \times FS_{RMSCode}} \tag{6.9}$$

Where:

V_{cc} = Voltage conversion constant

$V_{nominal}$ = Nominal voltage

V_{FSP} = Nominal voltage as a percentage of full-scale ADC voltage

As with the current conversion constant, the previously calculated V_{FSP} value and the $V_{nominal}$ of 230 V as defined in Table 6-1 may be used in the V_{cc} calculation. An $FS_{RMSCode}$ of 52702092 which is the full-scale code for RMS voltage, along with the values mentioned above, can be substituted into Equation (10) to yield:

$$V_{CC} = \frac{230 \times 10^6}{0.407 \times 52\,702\,092} = 10.722735\ \mu V_{rms}/LSB$$

Finally, the equation for the power conversion constant follows the same structure as that of the current and voltage. The only differences being that $V_{nominal}$ and $I_{nominal}$ are multiplied as well as V_{FSP} and I_{FSP}:

$$P_{CC} = \frac{I_{nominal} \times V_{nominal} \times 10^3}{I_{FSP} \times V_{FSP} \times FS_{RMSCode}} \tag{6.10}$$

Where:

P_{cc} = Power conversion constant

The same previously used $I_{nominal}$, $V_{nominal}$, I_{FSP} and V_{FSP} values are substituted into Equation (12), with an $FS_{RMSCode}$ of 20694066 for total and fundamental Watt, VAR and VA, yielding the following result:

$$p_{CC} = \frac{10 \times 230 \times 10^3}{0.07214 \times 0.407 \times 20\ 694\ 066} = 3.7854\ (mWATT/mVAR/mVA)/LSB$$

As shown in the calculation, conversion constant P_{cc} is used for a power calculation; as such, a single function may be written to compute the power readings from all three phases. The same applies to the current and voltage conversion constants.

To summarise, the derived conversion constants are as follows:

Current: 2.6299943 x 10^{-6}

Voltage: 10.722735 x 10^{-6}

Power: 3.7854 x 10^{-3}

In the Raspberry Pi source code, the physical quantities can be obtained by applying the conversion constant to the value read from the register via SPI communications. A basic example would be the following:

If the register value in the AIRMS register (A Phase RMS Current) is 12 000 500 then:

12 000 500 x (2.6299943 x 10^{-6}) = 31.56 A_{rms}

6.4.2. Power Quality Measurement registers

Many registers for power quality measurements and energy metering are available on the ADE9000. In Chapter 3, important power quality measurements are determined, then the actual ADE9000 registers are identified and modelled against IEC 61850 logical nodes and data objects in Chapter 4. This subsection briefly explains how the selected power quality register values are calculated both internally by the ADE9000's DSP and externally via calculations written in the Raspberry Pi source code. Four of the primary power quality measurements are examined; frequency, voltage dip/swell, power factor and total harmonic distortion

a) Frequency

The frequency value does not exist as a unique register value but can be computed using the xPERIOD register. Calculating the frequency requires the following equation:

$$f_L = \frac{800 \times 2^{16}}{xPERIOD + 1}(Hz) \tag{6.11}$$

Where:

F_L = Line frequency

xPERIOD = Period of a specific phase

The value of xPERIOD is calculated using low pass filtered values of voltage and current signal. Then using two values near the transition between negative and positive zero crossings (one on the negative side and one on the positive), the period is calculated using linear interpolation. It is essential first to set the SELFREQ bit in the ACCMODE register to 0 for a 50 Hz system or 1 for 60 Hz system to force the register value to either 50 or 60 Hz when the value is above 70Hz or below 40Hz.

At 50 Hz xPERIOD = 0x00A0_0000 (Hex)

At 60 Hz xPERIOD = 0x0085_5554 (Hex)

b) Voltage Dip and Swell

The dip and swell functions mainly consist of three registers, one to set the threshold, another to set the number of cycles over which the voltage signal is monitored and lastly a register for storing recorded value during the dip/swell event. Registers to set the threshold are the DIP_LVL for dips and SWELL_LVL for swell, whilst DIP_CYC and SWELL_CYC are used for setting the number of cycles to monitor and DIPx stores the recorded values.

Specifying an RMS ½ value in the DIP_LVL and SWELL_LVL register determines the threshold for a dip or swell event, a low value in DIP_LVL and a high value in SWELL_LVL. An RMS ½ value is an RMS measurement done over one cycle that’s updated every half cycle, and the ADE9000 contains registers for RMS ½ value on every phase for both voltage and current written as xVRMSONE and xIRMSONE respectively. The dip and swell level thresholds may be determined by writing the following values to the registers:

$$DIP\_LVL = xVRMSONE \times 2^{-5}$$

$$SWELL\_LVL = xVRMSONE \times 2^{-5}$$

Where:

xVRMSONE = RMS ½ value written to the LVL register.

The formulae can be further extrapolated using the voltage conversion constant as follows:

$$DIP\_LVL = \frac{V_{dip}}{k_v \times 10^{-6}} \times 2^{-5} \tag{6.12}$$

$$SWELL\_LVL = \frac{V_{swl}}{k_v \times 10^{-6}} \times 2^{-5} \tag{6.13}$$

Where:

V_{dip} = Intended RMS voltage dip threshold value

V_{swl} = Intended RMS voltage swell threshold value

K_v = voltage conversion constant

In equations (9) and (10), the actual RMS voltage values to the monitored are converted to RMS ½ value by division the voltage value with the voltage conversion constant multiplied by 10^{-6}.

Once a dip or swell event is recorded, a bit is set in the STATUS1 register. Each phase has a dedicated event bit which is specified in the register description section of the ADE9000 user guide. The mapping of STATUS1 register bits to QVVR logical node data objects is described in Chapter 4.

c) Power Factor

Power Factor is calculated simply by dividing an xWATT register value with xVA for each phase denoted by "x", the active and apparent power values are accumulated over 1.024s. For example, the power factor on the A phase is calculated as follows:

$$APF = \frac{AWATT}{AVA} \tag{6.14}$$

Where:

AWATT = Active power on the A Phase accumulated over 1.024s

AVA = Apparent power on the A Phase accumulated over 1.024s

The sign of AWATT determines the sign of the power factor, which is stored in 5.27 format. The following equation is implemented in the source code to convert the register value to a decimal value between -1 and 1:

$$Power\ Factor = APF \times 2^{-27}$$

A value of 0x07FF_FFFF in xPF corresponds to a power factor of 1 whilst -1 is stored as 0xF800_000.

d) Total Harmonic Distortion

The register value of the total harmonic distortion is calculated using the total and fundamental RMS values, as shown in the following equation:

$$AITHD = \sqrt{\frac{AIRMS^2 - AIFRMS^2}{AIFRMS^2}} \qquad 6.15)$$

The highest possible THD value is 400%, which is stored as 0x2000_0000 in the register. Obtaining the THD value as a percentage can be done by executing the following calculation in the source code:

$$\%THD = xITHD \times 2^{-27} \times 100\%$$

THD values are available for both voltage and current signals on all three phases, i.e. AITHD, AVTHD, BITHD, BVTHD, CITHD and CVTHD.

6.5. SPI communications and accessing register data

The ADE9000, like most Serial Peripheral Interfaces, consists of four pins: SCLK (Serial Clock), MOSI (Master Out, Slave In), MISO (Master In, Slave Out) and SS (Slave Select). The only mode of operation available on the ADE9000 is as an SPI slave, meaning it never initiates communications. Master communications are left up to the Raspberry Pi's BCM2835 ARM peripheral SoC which contains an SPI interface that can run either as master or slave. On the ADE9000, the SPI interface is compatible with both 16 and 32-bit read and write operations, as such, the register information section of the ADE9000 user guide must be referenced before formulating read/write functions on the Raspberry Pi.

Figure 6-9 shows the pin connections between the ADE9000 (the slave device) and the BCM2835 (master device), whilst Figure 6-10 represents signal diagrams for typical read and write transactions. SPI communications are initiated by the CS (Chip Select) pin, the SS pin uses inverted logic, applying a logic level high with keeping the pin low. The SS pin must remain low for the duration of the SPI transaction, and a high logic level aborts the transfer. While SS is low, data may be shifted into the device at the MOSI logic input, on the falling edge of SCLK, the rising edge of SCLK causes the device to sample the input data.

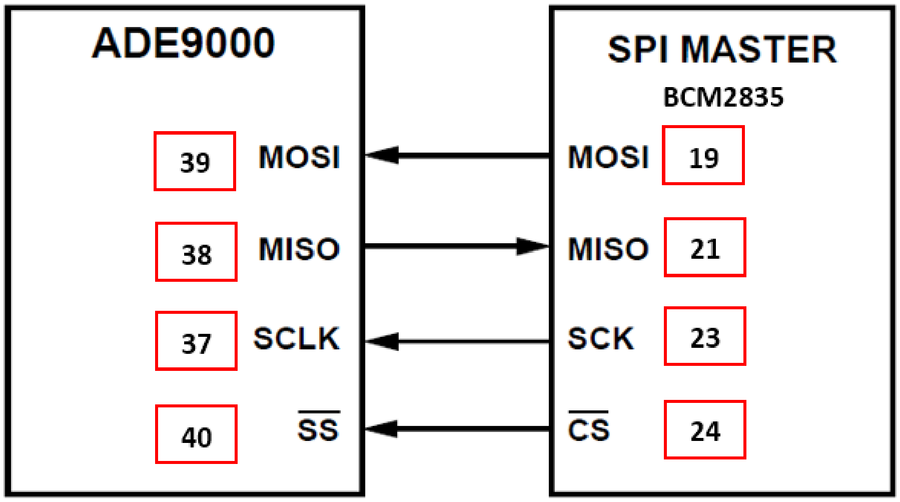


Figure 6-9: SPI Communication Pins (Analog Devices Inc., 2017a)

Data can be shifted from the ADE9000 from MISO also at the falling edge of SCLK and must be sampled by the BCM2835 on the rising edge. The most significant bit is shifted in and out first during transactions. This bit order is manually defined on the Raspberry Pi using the function: bcm2835_spi_setBitOrder() in the BCM2835 library.

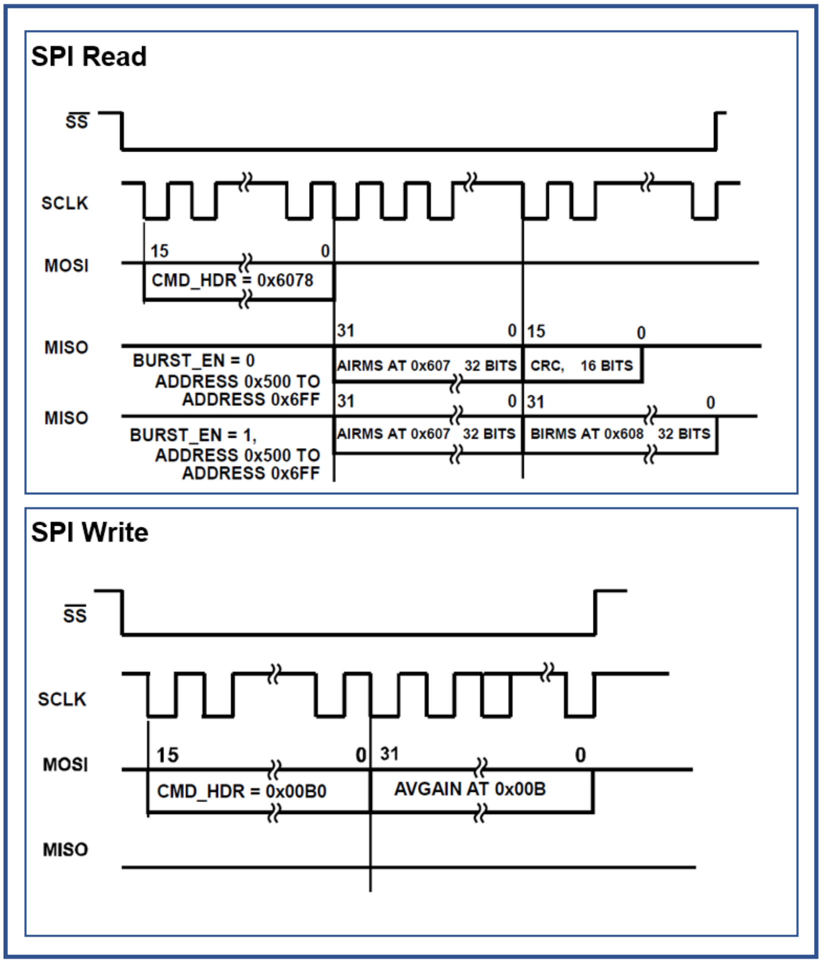


Figure 6-10: SPI Read and Write Signal Diagram (Analog Devices Inc., 2017a)

The SPI protocol on the ADE9000 is separated into three parts; these transaction types are typically comprised of the following:

1. A 16-bit command which indicates whether a read or write operation and which register to access is sent.
2. The command is followed by a 16 or 32-bit string of data to be written in the case of an SPI write, or the data read from the register, in the case of an SPI read.
3. In the last case, SPI read operations contain a cyclic redundancy check (CRC) after the read data.

The abovementioned transaction types need to be translated into source code logic in a way that takes into account the possible data lengths and conditions. All SPI communications are initiated by a command header that contains all of the relevant information mentioned in points 1 and 2. As briefly explained in Chapter 4, section 4.5, the 16-bit command header specifies a register address number and whether a read or write operation is to be executed. Figure 6-11 shows the 16-bit command header (CMD_HDR), using the figure as a reference, it can be observed that the command header contains the following important information:

- Bits 15 to 4, which are the most significant bits of the command header contains the address to be accessed, denoted as ADDR[11:0].
- Bit number "3" specifies whether the current operation is a read or write. "1" indicates a read, whereas "0" indicates a write.
- The final three bits (2 to 0) are reserved for internal chip timing, and their logic level does not matter.

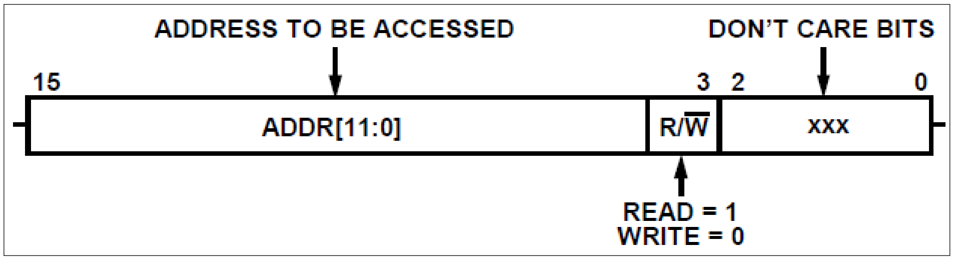


Figure 6-11: ADE9000 Command Header Format (Analog Devices Inc., 2017a)

The following subsection briefly describes the critical functions required to read and write data. Four main function types are available: "resize word", "Write SPI", "Parse Return Value" and "read".

6.6. Raspberry Pi source code design and implementation

This section details the design and implementation of the source code on the Raspberry Pi. The source code is written in C++ to more easily integrate with the LibIEC61850 library, which is written in C. Firstly, the procedure for accessing data from the registers programmatically using the SPI is described. Then finally, the integration of GOOSE and MMS with the acquired ADE9000 data is detailed.

Both programs are described using UML class diagrams and flow charts. The SPI communications source code is based on a few examples from Analog Devices, notably the EV-ADE9000SHIELDZ Arduino Shield as well as an open-source project called "piMeter" which aimed to create a Raspberry Pi based energy meter. At the same time, the source code for the GOOSE and MMS functions is partly based on example code present as part of the library but re-written in C++ to integrate the objects and function from the communications source code.

6.6.1. Startup and initialisation procedure

Figure 6-12 shows a flow diagram representing the ADE9000's start-up procedure. The start-up procedure makes extensive use of the writeSPI and writeSPILong function to write values to the required settings registers; these functions are described in more detail in sections 6.6.2 to 6.6.4. A quick start-up procedure is available in the ADE9000 User Guide, which forms the basis of the flow diagram. Before the registers may be written, the SPI communications are initialised using the BCM2835 functions described in Section 6.5, which are represented as subfunctions in Figure 6-12. Table 6-3 provides detailed descriptions of each step, denoted numerically from 1 to 10. Each execution step or ADE9000 register is defined and described.

Table 6-3: SPI and ADE9000 Startup and Initialisation Procedure Overview

Flow Chart Step	**Execution Step / ADE9000 Register**	**Description**
1 (1.1)	Initialise SPI	The BCM2835 initSPI() function is called. This function goes through the procedure of initialising the SPI protocol on the Raspberry Pi.
2	Check BCM2835	Make sure the BCM2835 chipset is enabled. If the chipset is not enabled, exit program.
3 (3.1)	**RUN:** Disable DSP	The digital signal processor on the ADE9000 is disabled to allow for configuration parameters to be written to registers.
4 (3.1)	**PGA_GAIN:** adjust Voltage and current gain	ADC gain voltage and current channels are adjusted using this register. The default gain on all registers is set to 1.
5 (3.1)	**VLEVEL:** Set the nominal voltage	The system's nominal voltage is set using this register.

6 (3.1)	**ACCMODE:** Set ADE9000 parameters	Contains several essential configuration parameters that are crucial to the operation of the ADE9000. The most critical parameters for this implementation are:
		SELFREQ = 0 (50Hz)
		VCONSEL = 0 (4 wire WYE)

7 (3.1)	**EP_CFG:** Energy Accumulation	It is used to configure energy accumulation for kWh meters. Disabled for this implementation.
8 (3.1)	**EGY_TIME:** Energy Time	Sets the desired number of samples during energy accumulation. Not used but needs to be specified
9 (3.1)	**RUN:** Enable DSP	The DSP is enabled once the configuration of registers is complete.
10 (3.1)	Pause execution	Pause software for 500ms to allow for all operations to complete

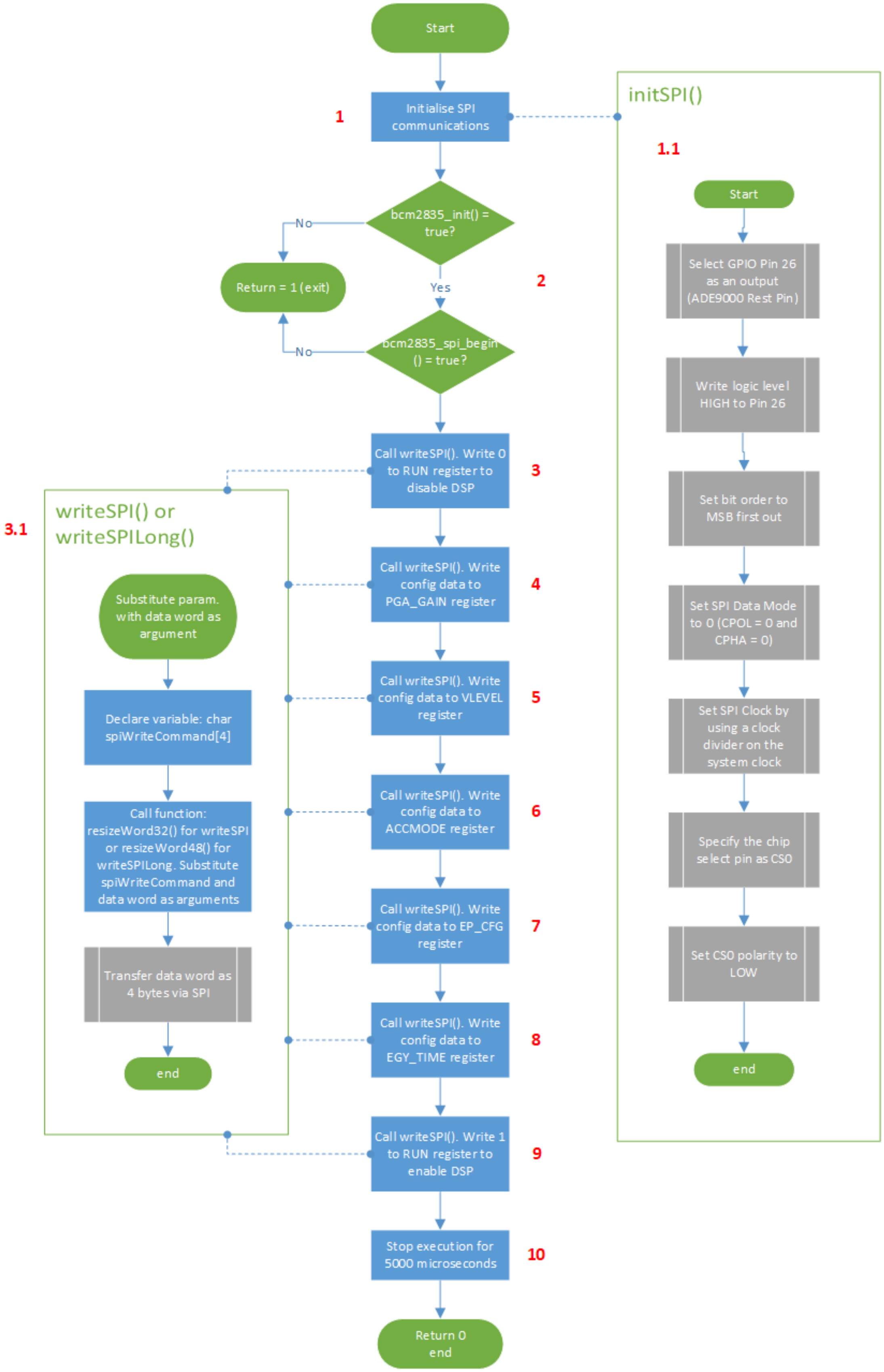
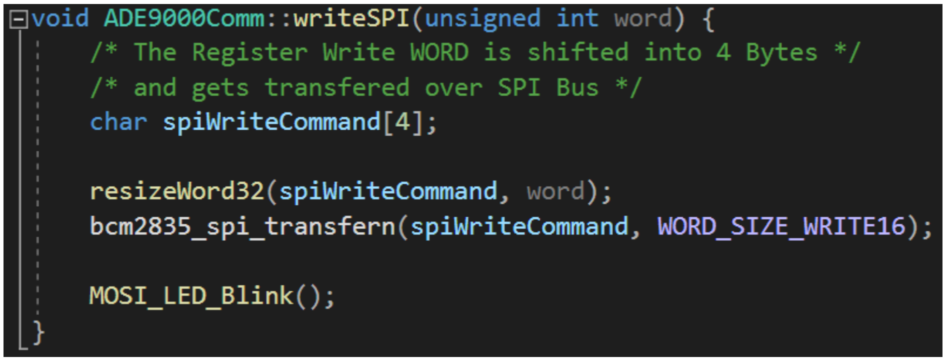

Start
1
Initialise SPI communications
initSPI()
1.1
bcm2835_init() = true?
No
Return = 1 (exit)
Yes
2
bcm2835_spi_begin () = true?
No
Start
Select GPIO Pin 26 as an output (ADE9000 Rest Pin)
Write logic level HIGH to Pin 26
Set bit order to MSB first out
Set SPI Data Mode to 0 (CPOL = 0 and CPHA = 0)
Set SPI Clock by using a clock divider on the system clock
Specify the chip select pin as CS0
Set CS0 polarity to LOW
end
Call writeSPI(). Write 0 to RUN register to disable DSP
3
3.1
writeSPI() or writeSPILong()
Substitute param. with data word as argument
Declare variable: char spiWriteCommand[4]
Call function: resizeWord32() for writeSPI or resizeWord48() for writeSPILong. Substitute spiWriteCommand and data word as arguments
Transfer data word as 4 bytes via SPI
end
Call writeSPI(). Write config data to PGA_GAIN register
4
Call writeSPI(). Write config data to VLEVEL register
5
Call writeSPI(). Write config data to ACCMODE register
6
Call writeSPI(). Write config data to EP_CFG register
7
Call writeSPI(). Write config data to EGY_TIME register
8
Call writeSPI(). Write 1 to RUN register to enable DSP
9
Stop execution for 5000 microseconds
10
Return 0 end

Figure 6-12: SPI Write Function and ADE9000 Start-up Procedure

6.6.2. WriteSPI() and WriteSPILong() functions

The “writeSPI” and “writeSPILong” functions write data to ADE9000 registers. Figure 6-13 shows the “writeSPI” function written in C++; this function contains a parameter named “word” of type “unsigned int”. In Figure 6.12, steps 3 to 10 use the write functions to configure the ADE9000 registers; the steps contained within the function are shown in step 3.1. A character array “spiWriteCommand consisting of four indexes is defined, the value of “word” is shifted into the four predefined bytes of “spiWriteCommand”. The resultant character array is transferred to the ADE9000 using the BCM2835 library’s “bcm2835_spi_transfern()” function. Along with writeSPI(), a writeSPILong() function is also available, this function is used for writing 48-bit (6 byte) values.

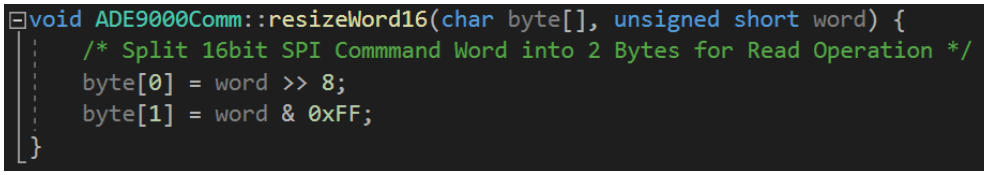


```
void ADE9000Comm::writeSPI(unsigned int word) {
    /* The Register Write WORD is shifted into 4 Bytes */
    /* and gets transfered over SPI Bus */
    char spiWriteCommand[4];

    resizeWord32(spiWriteCommand, word);
    bcm2835_spi_transfern(spiWriteCommand, WORD_SIZE_WRITE16);

    MOSI_LED_Blink();
}
```

Figure 6-13: WriteSPI Function

6.6.3. ResizeWord functions

The resizeWord function is used to convert the register addresses into character arrays. This is done to separate the data into several bytes depending on whether a read or write operation is occurring. Read procedures are exclusively 16-bit since only the command header requires transfer to the ADE9000, write operations can be either 32 or 48-bit depending on the type of data being written to the specified register. Figure 6-14 shows an example of the source code written for resizing a 16-bit word into multiple bytes. In the model, the function resizeWord16() consisting of parameters “byte[]” of type “char” and “word” of type “unsigned short” shown. The word is shifted by 8 bits to the right; the result is stored in index 0 of the character array (byte[0]). The second byte is determined by executing an AND logic function on the word with “FF” (hex). Similar functions are available for 32 and 48-bit values. These bytes would then be transferred to the ADE9000 using the “writeSPI” function.



```
void ADE9000Comm::resizeWord16(char byte[], unsigned short word) {
    /* Split 16bit SPI Commmand Word into 2 Bytes for Read Operation */
    byte[0] = word >> 8;
    byte[1] = word & 0xFF;
}
```

Figure 6-14: ResizeWord Example Code

6.6.4. Reading data From ADE9000 registers

Reading the ADE9000 registers is performed using the “read” function. The read function’s purpose is to accept a register address along with a command header as described in section 6.5 and send the data to the ADE9000 using the SPI bus. The read functions are the amalgamation of the resize word and parsing function (described in section 6.6.5) as well as the conversion constants defined earlier in the chapter. The read functions ultimately simplify the reading of measurement registers by providing easy to use templates. They may be used for all registers with minor modifications to accommodate varying bit counts and conversion constants where applicable.

Figure 6-15 shows the complete source code logic for the read function. The required steps are detailed in Table 6-3, where each step is denoted numerically from 1 to 5, sub-functions are specified with .1, the execution step is given a brief description as well.

Table 6-4: Steps for Reading Data from the ADE9000

Flow Chart Annotation	Execution Step	Description
1	Declare variables	Variables for the read command, message buffer and data storage are declared.
2 (2.1)	resizeWord: Split 16bit SPI command	Uses resizeWord() function as described in section 6.6.3 to create the read command in the formatting described in section 6.5. The substituted value should be hexadecimal or binary number containing the command header and register information. Store in declared “read command” variable.
3 (3.1)	Transfer data and read return data	The built-in bcm2835_spi_transfernb function is called to transfer data from the Raspberry Pi and receive the response from the ADE9000.
4 (4.1)	Move the data into an array and store as variable	Data returned by the ADE9000 is parsed using the parse32bitReturnValue function and stored in a 32-bit array for use by the program. Parsing is necessary due to CRC bits being present in the returned data.
5	Apply conversion constant	Conversion constant is applied to convert the raw data to a physical quantity.

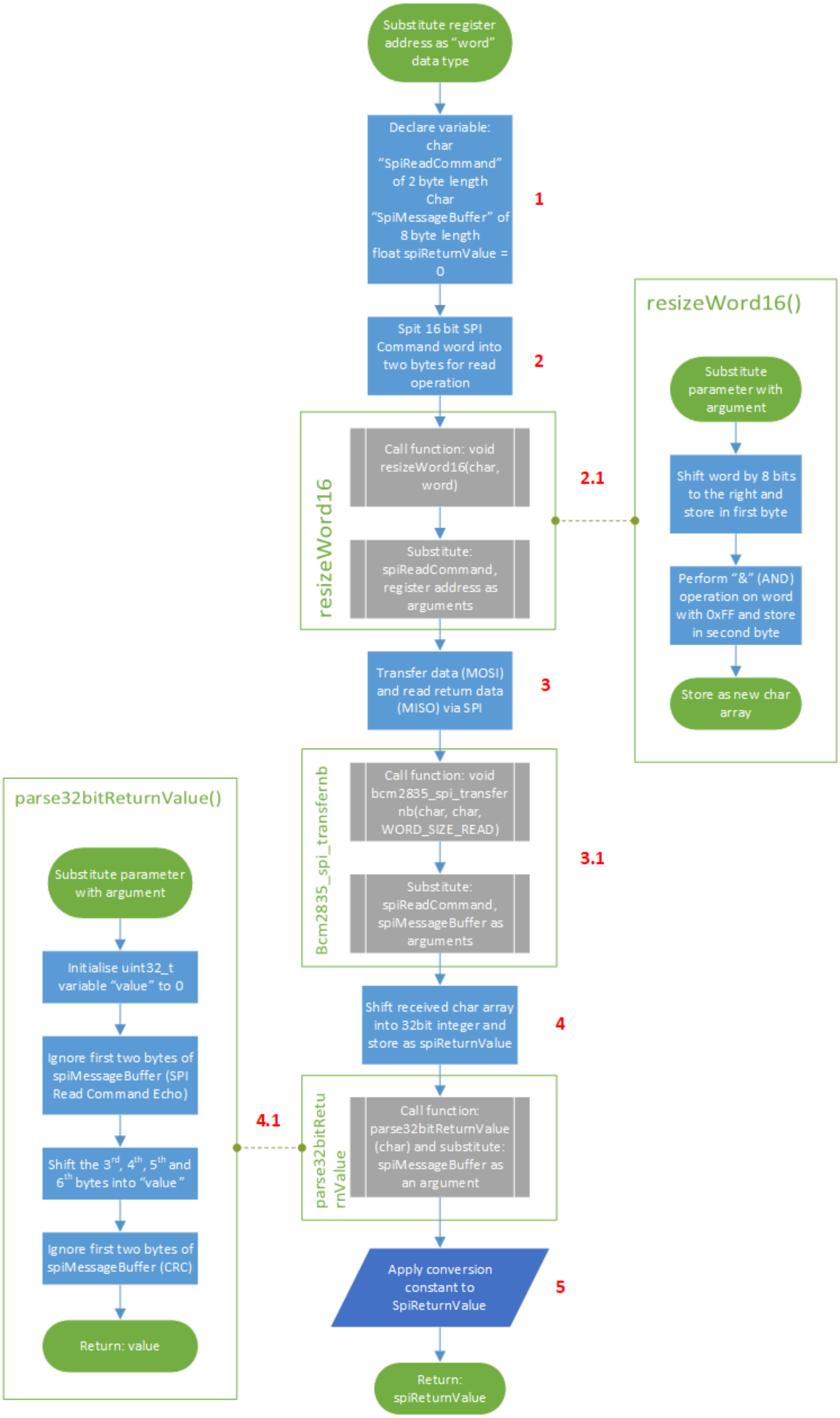
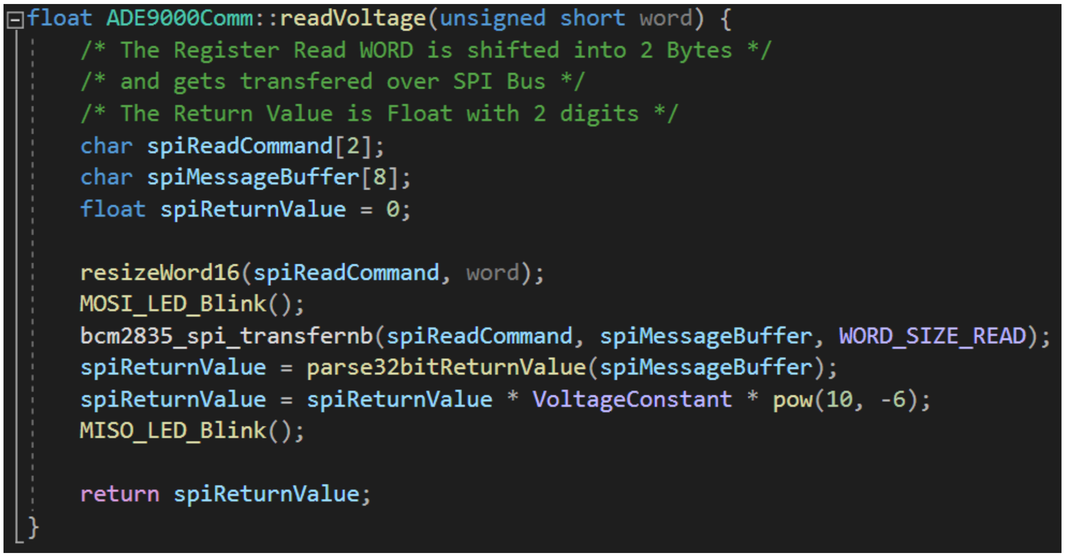

Substitute register address as "word" data type
Declare variable: char "SpiReadCommand" of 2 byte length Char "SpiMessageBuffer" of 8 byte length float spiReturnValue = 0
1
Spit 16 bit SPI Command word into two bytes for read operation
2
resizeWord16
Call function: void resizeWord16(char, word)
Substitute: spiReadCommand, register address as arguments
2.1
resizeWord16()
Substitute parameter with argument
Shift word by 8 bits to the right and store in first byte
Perform "&" (AND) operation on word with 0xFF and store in second byte
Store as new char array
Transfer data (MOSI) and read return data (MISO) via SPI
3
Bcm2835_spi_transfernb
Call function: void bcm2835_spi_transfernb(char, char, WORD_SIZE_READ)
Substitute: spiReadCommand, spiMessageBuffer as arguments
3.1
Shift received char array into 32bit integer and store as spiReturnValue
4
parse32bitReturnValue
Call function: parse32bitReturnValue (char) and substitute: spiMessageBuffer as an argument
4.1
parse32bitReturnValue()
Substitute parameter with argument
Initialise uint32_t variable "value" to 0
Ignore first two bytes of spiMessageBuffer (SPI Read Command Echo)
Shift the 3rd, 4th, 5th and 6th bytes into "value"
Ignore first two bytes of spiMessageBuffer (CRC)
Return: value
Apply conversion constant to SpiReturnValue
5
Return: spiReturnValue

Figure 6-15: SPI Read Function Source Code Logic

6.6.5. Read functions

Finally, the read functions are the amalgamation of both the resize word and parsing function as well as the conversion constants defined earlier in the chapter. The read functions ultimately simplify the reading of measurement registers by providing easy to use templates. They may be used for all registers with minor modifications to accommodate varying bit counts and conversion constants where applicable.

Figure 6-16 shows an example of a read function for phase voltages. As with previous functions, it requires the parameter “word” of type “unsigned short”. Three variables are declared, spiReadCommand[2] and spiMessageBuffer[8] arrays of type “char”, along with spiReturnValue of type “float” which is initiated to 0. The value of “word” is shifted into the two predefined bytes of “spiReadCommand” using the resizeWord16() function; then the read command is transferred via SPI using the bcm2835_spi_transfernb() function. Although similar to the bcm2835_spi_transfern() function introduced in section 6.5.3, a key difference is inclusion on an extra parameter for the return value within the function. The resultant 32-bit return value is stored in spiMessageBuffer, then parsed and written to the variable spiReturnValue as a floating-point value. Finally, the conversion constant is applied, and the spiReturnValue is returned as a result.



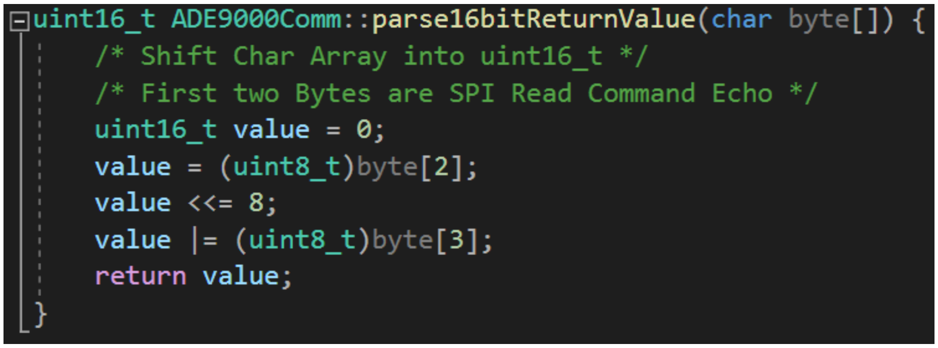
```
float ADE9000Comm::readVoltage(unsigned short word) {
    /* The Register Read WORD is shifted into 2 Bytes */
    /* and gets transfered over SPI Bus */
    /* The Return Value is Float with 2 digits */
    char spiReadCommand[2];
    char spiMessageBuffer[8];
    float spiReturnValue = 0;

    resizeWord16(spiReadCommand, word);
    MOSI_LED_Blink();
    bcm2835_spi_transfernb(spiReadCommand, spiMessageBuffer, WORD_SIZE_READ);
    spiReturnValue = parse32bitReturnValue(spiMessageBuffer);
    spiReturnValue = spiReturnValue * VoltageConstant * pow(10, -6);
    MISO_LED_Blink();

    return spiReturnValue;
}
```

Figure 6-16: Example of an ADE9000 Read Function

6.6.6. Parsing functions

Three parsing functions are available to accommodate for SPI read functions of all lengths. The purpose of the parsing function is only to retrieve the required data from the received data. In Figure 6-10, the SPI Read signal diagram shows the transmission of a command header and the possible responses depending on whether burst read is enabled or not. With burst read disabled, the read response for 32-bit measured value includes a 16-bit CRC. The CRC is only useful for debugging; as such, there is no need to show the result of the CRC unless necessary. As such, a parsing function is required to discard the CRC data and only display the measurand data. In all responses, an echo of the 16-bit read command is sent before the actual data, which also necessitates parsing.

Figure 6-17 is an example of a function that parses a 16-bit return value, and the function contains the parameter “byte[]” of type ”char”. Within the function, the variable “value” of type “uint16_t” is declared and initialised to 0. In the actual source code, the first two bytes are skipped, then “value” is assigned to byte[2], “value” is then shifted to the left by 8 bits. Subsequent bytes are assigned to the newly shifted “value” using a bitwise OR. This way, all of the shifted bits retain their original values while bits 7 to 0 are written with data contained in “byte[3]”.



```
uint16_t ADE9000Comm::parse16bitReturnValue(char byte[]) {
    /* Shift Char Array into uint16_t */
    /* First two Bytes are SPI Read Command Echo */
    uint16_t value = 0;
    value = (uint8_t)byte[2];
    value <<= 8;
    value |= (uint8_t)byte[3];
    return value;
}
```

Figure 6-17: Function for Parsing and a 16-bit value

In Figure 6-17, the aforementioned amalgamation of functions is visually illustrated using a flow chart for the read function. The parsing, bcm2835 and the resize word functions are highlighted and shown as subfunctions. The flow chart provides a detailed look at the sub-processes involved in the source code shown in Figure 6-16 and described in the previous paragraph.

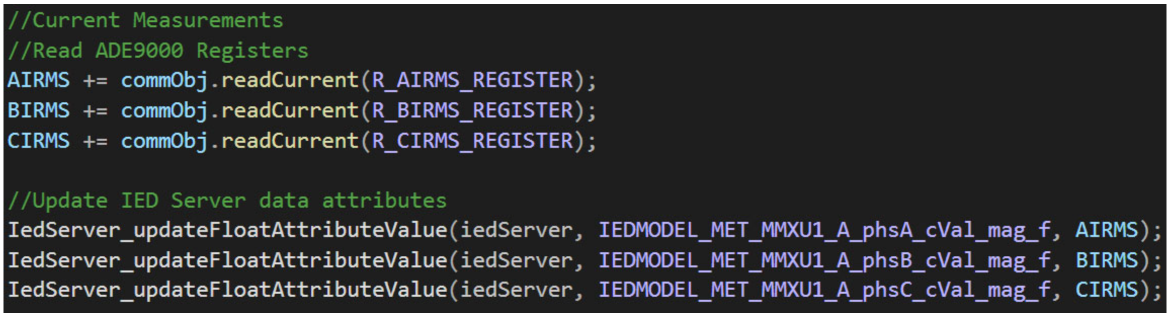
6.6.7. LibIEC61850 GOOSE and MMS functions

The last consideration in terms of software is the source code for the IEC 61850 functions. Although the LibIEC61850 library supports hundreds of functions, only a few are required to implement GOOSE publishing and an MMS server on the Raspberry Pi. The “main loop” of the project’s source code is discussed in this subsection. Figure 6-19 shows a flow chart of the necessary start-up procedure of the LibIEC61850 functions. The process is standard for any source code involving GOOSE and MMS communications.

The following is a summary of the process, according to Figure 6-19, each step is annotated within the flow diagram, numbered 1 to 8:

1. IEC 61850 model is imported from the SCL file
2. Necessary variables are declared (see Appendix D)
3. Create a new server configuration object and set parameters if necessary
4. Check command line input for specific GOOSE interface.
5. If the interface is entered in the command line: GOOSE interface ID = entry. else, GOOSE interface ID = eth0
6. New IEC 61850 server instance is created using a configuration object
7. Start MMS server and begin listening for clients on port 102
8. Enable GOOSE Publishing

The read function forms the basis of the integration between LibIEC61850 functions and a critical component of the “main loop”, the return values from reading functions are used to update data attributes. In figure 6-18, the result of readCurrent() is stored in float variables AIRMS, BIRMS and CIRMS. The value of the variables mentioned above is then written to the IED server a data attribute in the MMXU logical node instance using one of the LibIEC61850 update functions. Data attributes are all read and written in this manner, and then transmitted either using GOOSE or MMS depending on the communications model.



```
//Current Measurements
//Read ADE9000 Registers
AIRMS += commObj.readCurrent(R_AIRMS_REGISTER);
BIRMS += commObj.readCurrent(R_BIRMS_REGISTER);
CIRMS += commObj.readCurrent(R_CIRMS_REGISTER);

//Update IED Server data attributes
IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_MET_MMXU1_A_phsA_cVal_mag_f, AIRMS);
IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_MET_MMXU1_A_phsB_cVal_mag_f, BIRMS);
IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_MET_MMXU1_A_phsC_cVal_mag_f, CIRMS);
```

Figure 6-18: Data attributes are updated using the ADE9000 read the function return value

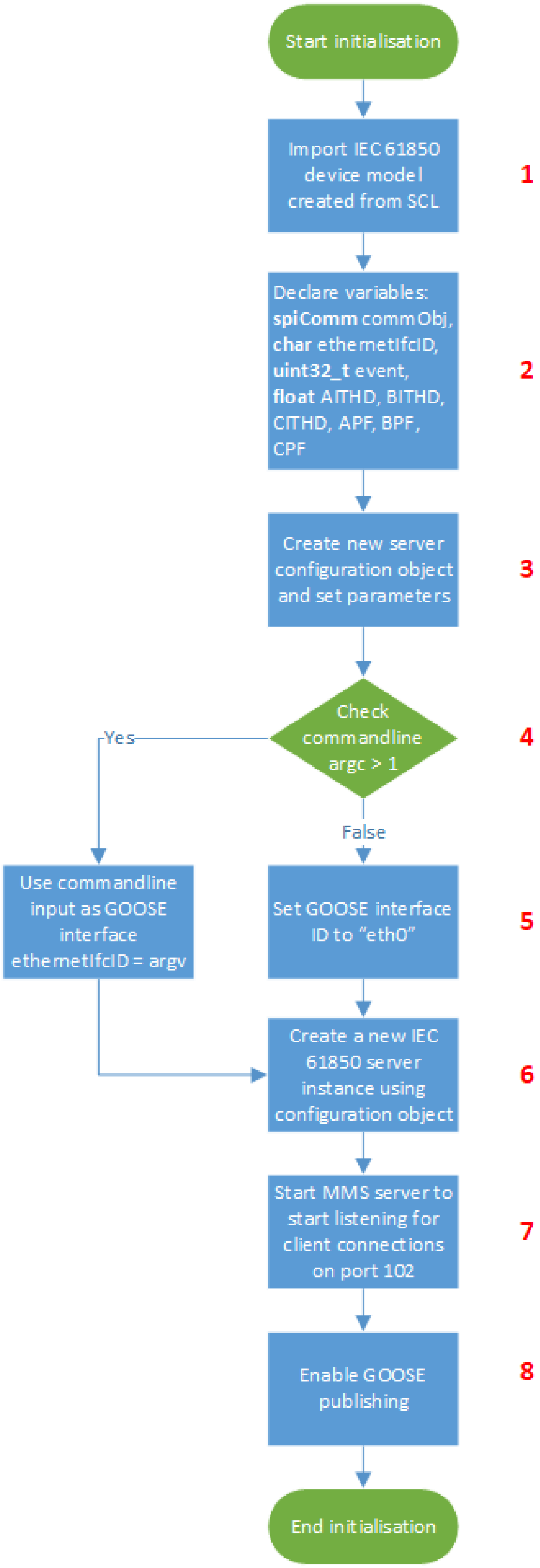

Start initialisation
Import IEC 61850 device model created from SCL
1
Declare variables: spiComm commObj, char ethernetIfcID, uint32_t event, float AITHD, BITHD, CITHD, APF, BPF, CPF
2
Create new server configuration object and set parameters
3
Check commandline argc > 1
4
Yes
False
Use commandline input as GOOSE interface ethernetIfcID = argv
Set GOOSE interface ID to "eth0"
5
Create a new IEC 61850 server instance using configuration object
6
Start MMS server to start listening for client connections on port 102
7
Enable GOOSE publishing
8
End initialisation

Figure 6-19: Initialisation Procedure Flowchart

The overall program loop is presented in Figure 6-20 as a flow chart. Regardless of the measurement or events being read and transmitted, the loop remains the same, and Figure 6-20 represents the generic procedure instead of the actual source code. Based on the flowchart, the procedure can be summarised as 11 steps which are annotated on the flow chart:

1. Lock IED Server Data Model
2. Get system time and set as timestamp variable
3. Define the timestamp object
4. Define maximum time drift variable
5. Set the timestamp object as “IEC timestamp” variable
6. Define IEC timestamp and maximum time drift variables
7. Check if time has drifted above the value of the time drift variable and set time not synchronized flag is valid.
8. Call the read function and substitute register address. The result is stored as an appropriate data type (float or unint32_t) and update timestamp and data attribute on IED server
9. Unlock IED server data model
10. Sleep for 1 second
11. Go back to step 1 and restart loop

The program continues until the process is killed in the command window using Ctrl + C. Modifying the program’s sleep time changes the frequency at which GOOSE messages are published. Optionally, a script may be written to start the program on start-up; this enables the system to function without a monitor connected or VNC client running to initiate the data acquisition software.

Appendix D is an extract if the entire final source code for both the SPI read and write functions and the libeIEC61850 GOOSE and MMS Function mapping.

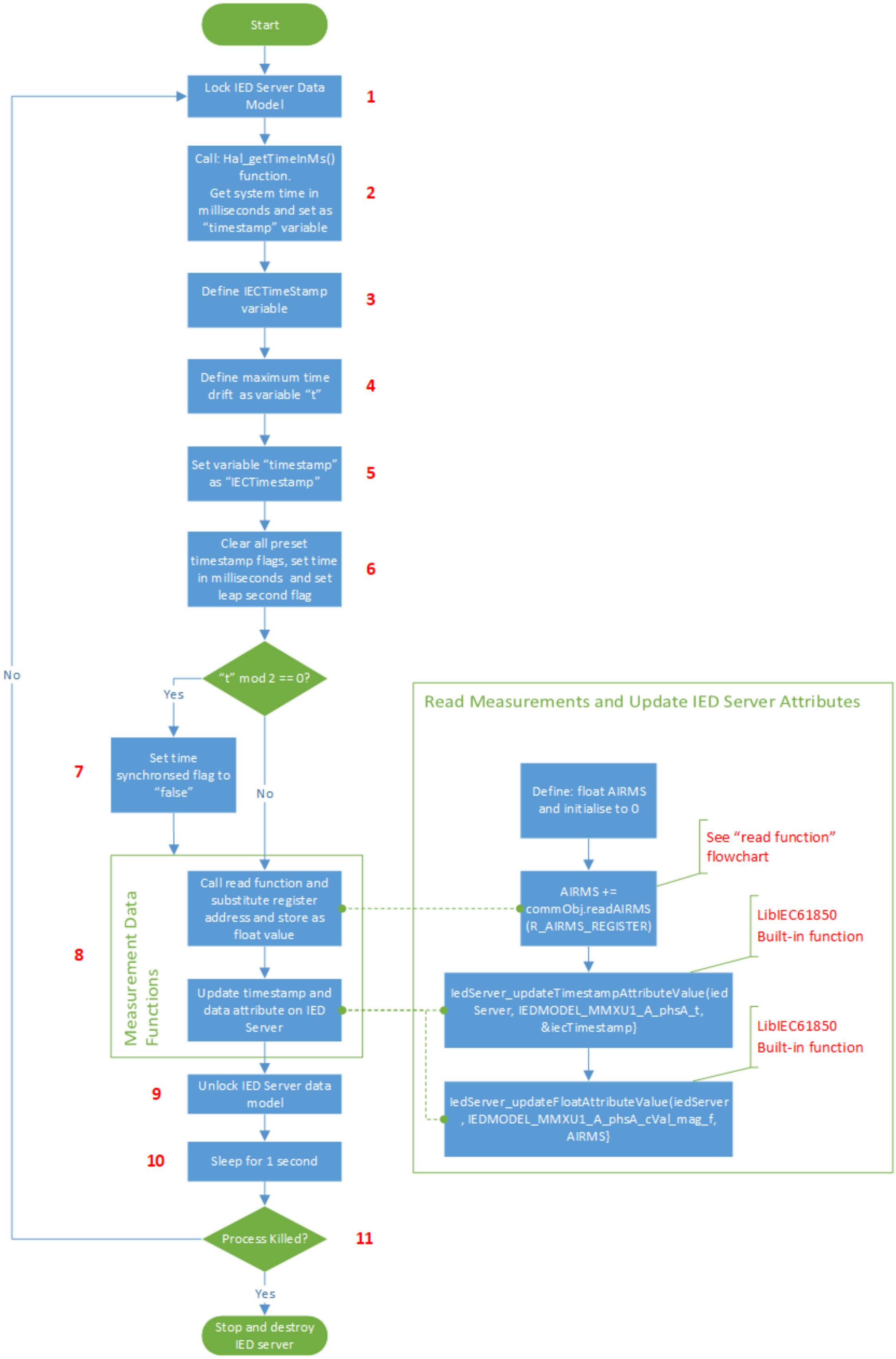

Start
Lock IED Server Data Model
1
Call: Hal_getTimeInMs() function.
Get system time in milliseconds and set as "timestamp" variable
2
Define IECTimeStamp variable
3
Define maximum time drift as variable "t"
4
Set variable "timestamp" as "IECTimestamp"
5
Clear all preset timestamp flags, set time in milliseconds and set leap second flag
6
"t" mod 2 == 0?
Yes
No
Set time synchronsed flag to "false"
7
No
Measurement Data Functions
8
Call read function and substitute register address and store as float value
Update timestamp and data attribute on IED Server
Unlock IED Server data model
9
Sleep for 1 second
10
Process Killed?
11
Yes
Stop and destroy IED server
Read Measurements and Update IED Server Attributes
Define: float AIRMS and initialise to 0
See "read function" flowchart
AIRMS += commObj.readAIRMS (R_AIRMS_REGISTER)
LibIEC61850 Built-in function
IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_MMXU1_A_phsA_t, &iecTimestamp}
LibIEC61850 Built-in function
IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_MMXU1_A_phsA_cVal_mag_f, AIRMS}

Figure 6-20: The Final Program Loop

6.7. Final enclosure assembly

Figure 6-21 shows the power quality monitoring device before and after being placed within an enclosure. The enclosure provides a secure vessel for transport and prototyping, with an open-ended design to accommodate for connections with CTs and voltage signals as well as the ribbon cable. The enclosure is constructed from two pieces of acrylic, with the Raspberry Pi and ADE9000 PCBs mounted on the top and bottom acrylic sheets respectively. Steel standoffs are used to create the space required for mounting the two PCB, and the ribbon cable continues to act as the signal bus between the digital signal processing and computing boards. The construction is secured with end-cap nuts on the top acrylic sheet that forms the roof and rubber feet at the base acrylic.

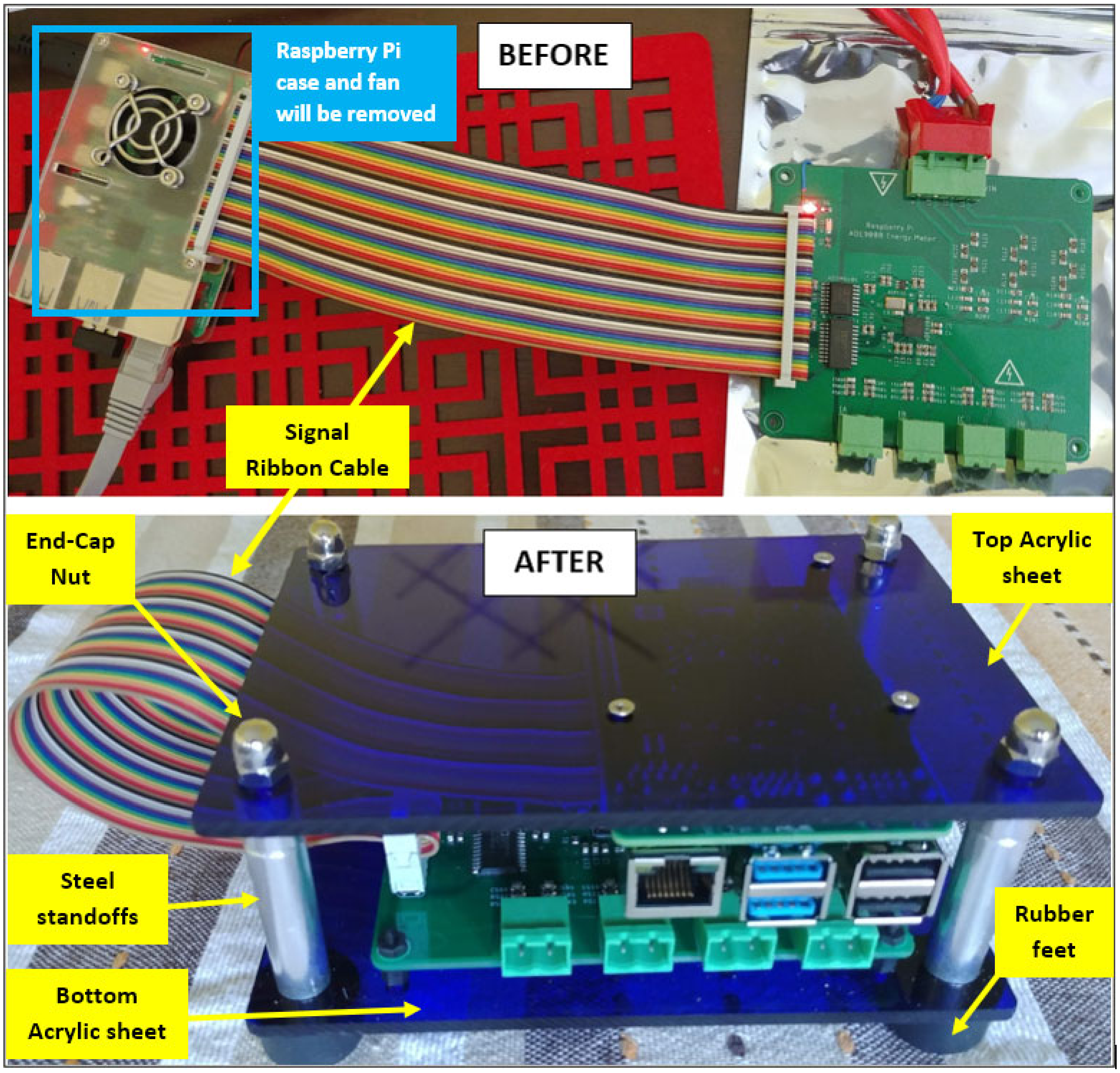


Figure 6-21: Implemented Device Before and After Assembly Enclosure

6.8. Conclusion

Writing the software and assembling the device concludes the implementation phase of the project. In this chapter, a novel low-cost IEC 61850-based data acquisition and metering device is designed and fully implemented.

The overall system design is reviewed, with four primary system modules identified and described. The hardware design is then explored in detail; this comprised of designing and component-sizing the various data acquisition circuits for current and voltage, which serve the basis of all measured and computed values obtained on the device. Implementing the data acquisition circuits allowed for the configuration and calibration of relevant data registers using calculated component sizes along with ADC code parameters; conversion parameters enabled the conversion of register values to physical quantities. Finally, the source code implementation on the Raspberry Pi is discussed for both transferring data from the ADE9000 using the Raspberry Pi's built-in SPI bus and subsequently mapping and transmitting the converted quantities using GOOSE and MMS.

With an understanding of how the necessary processes execute and the various components' roles within the system, Chapter 7 details the evaluation of the system's functionality based on the deliverables identified in Chapter 2 and the required power quality measurements identified in Chapter 3. A testbed is developed for data validation purposes. Signal injection tests are carried out to assess the circuit design, calibration procedures and conversion parameters. The IEC 61850 implementation is also tested using the IEC 61850-10 part as a basis.

CHAPTER SEVEN

EVALUATION AND TESTING OF METERING AND IEC 61850 FUNCTIONS

7.1. Introduction

The previous chapter presented the conclusion of the implementation phase of the projects. All hardware and software components have been assembled and configured, and the resulting device is fully functional, but not yet validated. This chapter aims to demonstrate the testing and validation of the metering and IEC 61850-based communications functionality of the power quality monitoring (PQM) device. As such, the procedure for assessing and evaluating these integral aspects of the implementation are described in this chapter. The formulated tests are designed to determine whether the objectives set out in the original scope were achieved and additionally to test the overall capabilities and the shortcomings of the implementation in its current state.

The required assessment criteria first need to be determined by referencing the desired outcomes of the research project. Then each test procedure is described in detail by providing a procedure overview, the required hardware and software, detailing the testing methodology, then finally compiling and analysing the results.

7.2. Assessment criteria

The following assessment criteria are used to measure and verify the performance of the device under various conditions:

- **Measurement Accuracy:** The accuracy of the data acquisition portion of the device is determined by injecting voltage and current signals in a controlled environment and comparing the readings with a reference meter.
- **Power Quality Metrics:** Power Quality Monitoring features available on the device are tested using simulated power quality disturbances of varying magnitudes and settings to determine the detection accuracy and reliability.
- **IEC 61850 MMS and ACSI Service Conformance:** In this test, the implementation of MMS and ACSI services are tested for conformance with IEC 61850-8-1 and tests are based on the guidelines tabled in the IEC 61850-10 part of the standard.

7.3. Test 1: Measurement accuracy validation

Measurement accuracy tests are designed to determine, and if required, calibrate the accuracy of the device's measurement capabilities. The three-phase voltage and current readings are integral to the operation of the ADE9000's DSP calculations. The tests in this section are designed to evaluate the entire data acquisition module, as presented by the green blocks in Figure 7-1, this is achieved by injecting signals and determining whether the DSP and data conversion is functioning as expected according to the configurations done in chapter 6.

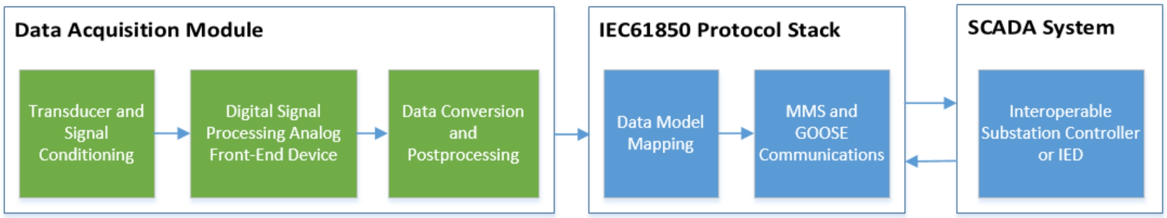

Data Acquisition Module
Transducer and Signal Conditioning
Digital Signal Processing Analog Front-End Device
Data Conversion and Postprocessing
IEC61850 Protocol Stack
Data Model Mapping
MMS and GOOSE Communications
SCADA System
Interoperable Substation Controller or IED

Figure 7-1: Section 7.3 Focus Areas

Although the device would typically be utilised in a three-phase environment, the measurement accuracy tests are run exclusively in a single-phase configuration, and this is to simplify the testing and avoid duplication of similar tests on different phases.

7.3.1. Procedure overview

In Test 1, the accuracy of the meter is examined and further calibrated if necessary. In this experiment/calibration procedure, the variables being assessed are current and voltage on the A Phase.

Firstly, control measurements are required to obtain references to evaluate the implemented meter against a reference meter. A control measurement may be attained by injecting stable voltage and current signals of a set range using the OMICRON CMC 356 and measured using a calibrated multimeter, in this case, an Agilent U1232A True RMS multimeter.

The same signals are subsequently injected into the measurement inputs of the implemented hardware, and a percentage error (deviation from the control value) is determined. Suppose the readings diverge too far outside the accepted levels of accuracy. In that case, the voltage and current constants can then be adjusted accordingly to achieve an acceptable level of accuracy within the given range.

7.3.2. Required hardware and software

The following hardware and software are required for the measurement accuracy test:

Hardware

- Omicron CMC 356
- Agilent U1232A Multimeter
- Windows PC
- Raspberry Pi
- ADE 9000

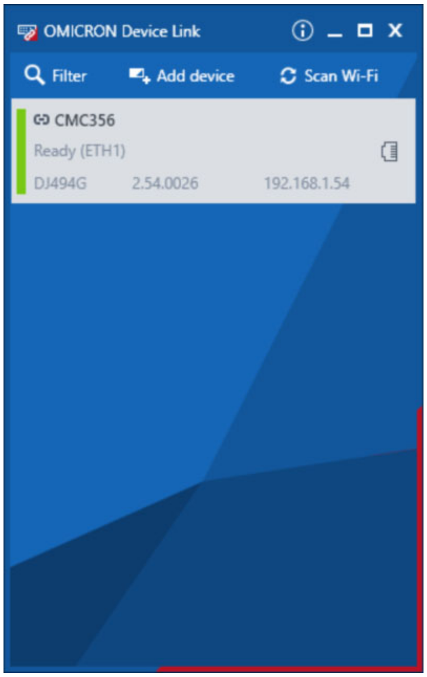
Software

- Microsoft Word
- VNC Viewer and FTP Client
- Raspberry Pi Data Acquisition Software
- Omicron Test Universe 4.00
    - Quick CMC Test Module

7.3.3. Test methodology

The procedure for testing the accuracy of the voltage and current in a controlled environment is as follows:

1. The OMICRON Test Universe is associated with the CMC 356 using the test set association dialogue, as shown in Figure 7-2. Once an association is completed, the Quick CMC may be opened and configured.



CMC356
Ready (ETH1)
DJ494G 2.54.0026 192.168.1.54

Figure 7-2: OMICRON Device Link Dialogue Window

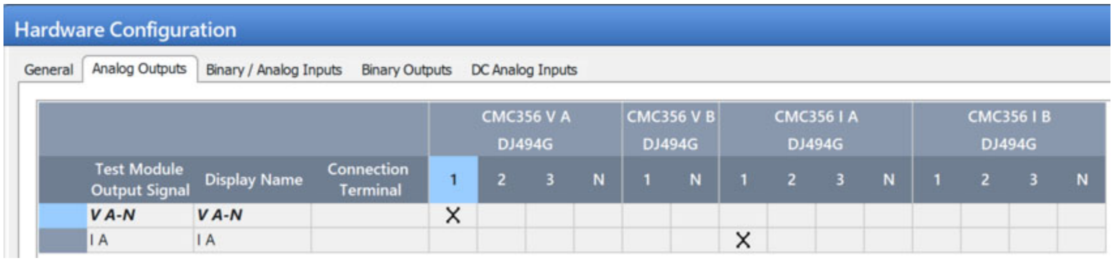
2. Once Quick CMC is opened, the Test Object may be configured with the settings shown in Table 7-1.

Table 7-1: Parameters for Measurement Test

Test Parameter	Setting
F nom (Nominal Frequency)	50 Hz
V nom (secondary)	240 V (L-N)
V primary	240 V (L-N)
I nom (secondary)	10 A
I primary	10 A
V max	420 V (L-L)
I max	50 A

The nominal frequency is set to the standard 50 Hz, as shown in Table 7-1. The secondary nominal voltage is set to match the primary voltage since the signals are injected directly into the voltage channels of the data acquisition hardware and will thus not require a voltage transformer. Voltage (both nominal and primary) are set 10 V above real nominal (240 V vs 230 V) to allow values slightly above 230 V to be injected. In the case of the current, the SCT13-000 split-core CT, as described in Chapter 6, is used, thus expected primary current values may also be injected. An amount of 10 A is chosen as the nominal current due to the current limitations of the multimeter. Still, this limit provides an acceptable range of possible current values to test. V max and I max bear no relevance to this test; as such, they may be set to any value above nominal.

3. The hardware configuration only requires the correct configuration of the analogue output connection terminals, as shown in Figure 7-3.

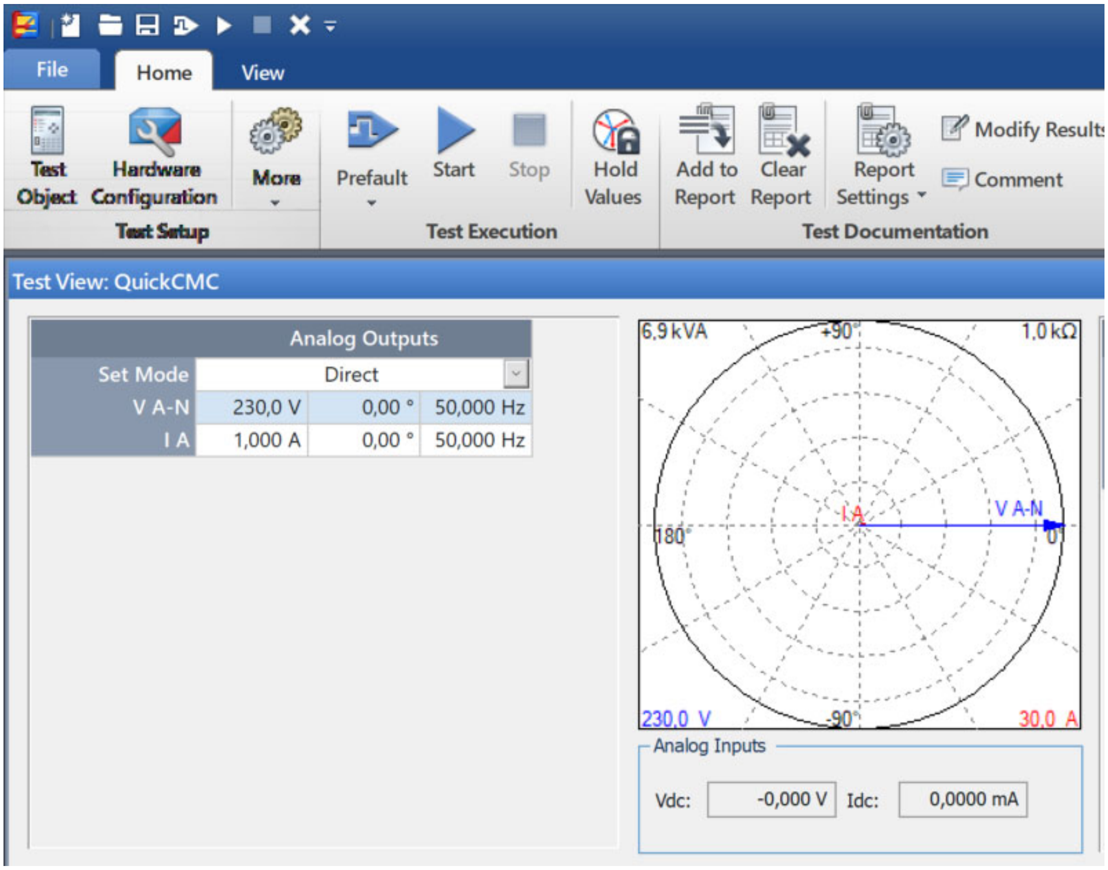


			CMC356 V A DJ494G				CMC356 V B DJ494G		CMC356 I A DJ494G				CMC356 I B DJ494G			
Test Module Output Signal	Display Name	Connection Terminal	1	2	3	N	1	N	1	2	3	N	1	2	3	N
V A-N	V A-N		X													
I A	I A								X							

Figure 7-3: Hardware Configuration for Measurement Test

Since a single phase is tested, only terminal 1 of CMC356 V A (voltage output signal V A-N) and terminal 1 of CMC356 I A (current output signal I A) require configuration.

Once the setup is completed, the QuickCMC test view contains the outputs as shown in Figure 7-4, the display shown allows for the modification of the voltage and current values as well as their respective phase angles and signal frequencies.



Test View: QuickCMC

Analog Outputs			
Set Mode	Direct		
V A-N	230,0 V	0,00 °	50,000 Hz
I A	1,000 A	0,00 °	50,000 Hz

Analog Inputs

Vdc: -0,000 V Idc: 0,0000 mA

Figure 7-4: QuickCMC Test View

4. The multimeter is connected to the CMC connection terminals, as shown in Figure 7-5 for the specific test. The voltage output terminals are connected in parallel to the multimeter input terminals, whilst the current output terminals are connected in series. With the multimeter is connected, the desired voltage and current signal values may be set, and the test initiated. Once the test is activated, the specified amount is applied and measured by the multimeter.
   - For voltage, values ranging from 150 V to 240 V are applied with 10 V increments between values. The injected signal is held for 10 seconds to allow the multimeter reading to stabilize and meter reading noted down. This procedure is repeated until all required readings have been attained.
   - Current is measured similarly to voltage, with values ranging from 0,5 A to 8 A applied at an increment of 500mA.

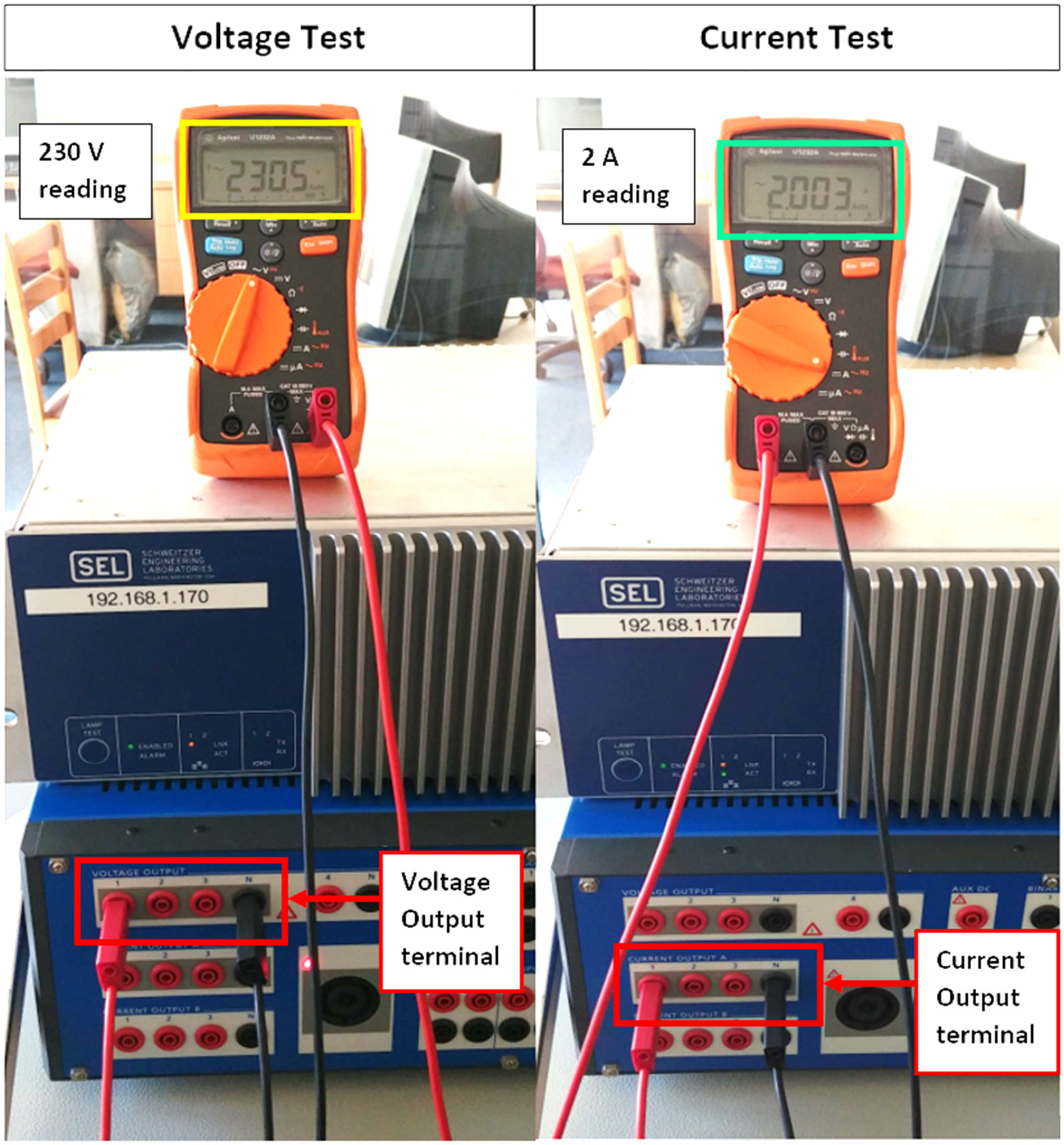


Figure 7-5: Voltage and Current Measurements with a multimeter

5. With the baseline readings from the multimeter obtained, the same readings may be performed using the PQM device connected, as shown in Figure 7-6. In the figure, a simple testbed is created by connecting the CMC voltage terminals (shown in the red rectangle) to the device's input terminals (yellow rectangle). The current input of the device is also connected to the clamped CT. The CT (Indicated by the orange rectangle) is clamped to the wire connected between current output terminal 1 and the neutral terminal, whilst the voltage terminals are connected directly to the A-phase and Neutral voltage channels on the PQ monitoring device, which is indicated by the yellow rectangle.

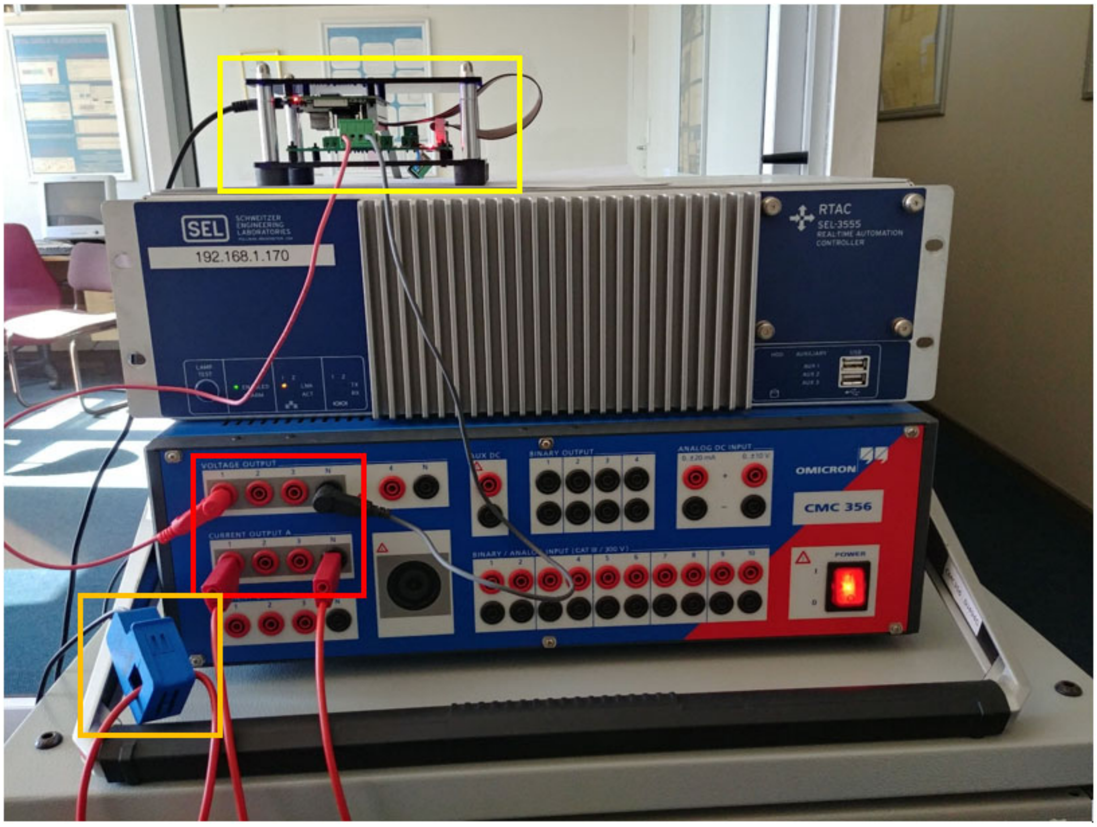
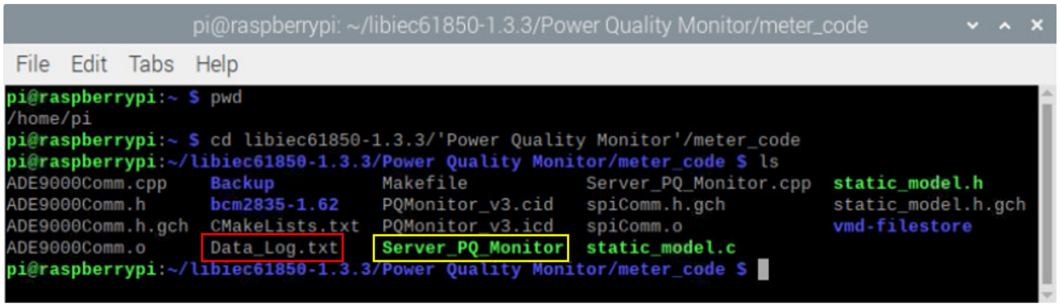


Figure 7-6: CMC 356 outputs connected to the device

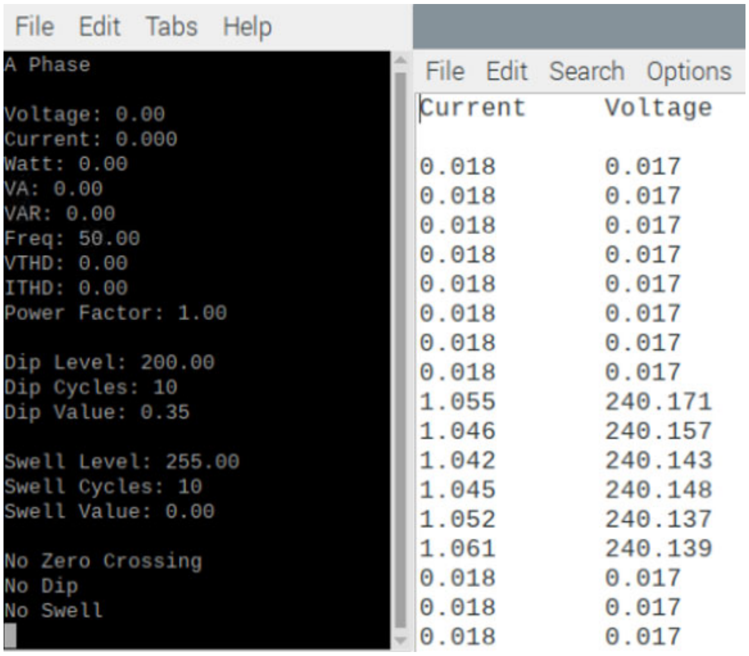
6. Once the terminals have been connected, data acquisition may begin. The following procedure applies to initiate the data acquisition process:
   A. Login to Raspberry Pi OS using the VNC Viewer
   B. Navigate to the directory where the source code and executable file is stored using Linux commands shown in Figure 7-7. In the image, the file named “Server PQ Monitor” (yellow) is the executable file that starts data acquisition and “Data_Log.txt” (red) stores the data samples.
   C. Finally, the program may be started by entering “sudo ./Server_PQ_Monitor”.


pi@raspberrypi: ~/libiec61850-1.3.3/Power Quality Monitor/meter_code
File
Edit
Tabs
Help

```
pi@raspberrypi:~ $ pwd
/home/pi
pi@raspberrypi:~ $ cd libiec61850-1.3.3/'Power Quality Monitor'/meter_code
pi@raspberrypi:~/libiec61850-1.3.3/Power Quality Monitor/meter_code $ ls
ADE9000Comm.cpp    Backup          Makefile           Server_PQ_Monitor.cpp  static_model.h
ADE9000Comm.h      bcm2835-1.62    PQMonitor_v3.cid   spiComm.h.gch          static_model.h.gch
ADE9000Comm.h.gch  CMakeLists.txt  PQMonitor_v3.icd   spiComm.o              vmd-filestore
ADE9000Comm.o      Data_Log.txt    Server_PQ_Monitor  static_model.c
pi@raspberrypi:~/libiec61850-1.3.3/Power Quality Monitor/meter_code $
```

Figure 7-7: Navigation to source code and executable file folder

7. Once the program has been started, the output window shown in Figure 7-8 shows the current measured values as well as the log file. The output is updated approximately every 2 seconds and the data recorded in Data_Log.txt. At this point, the signal injection may be initiated from QuickCMC; the same signals generated for the multimeter test are generated for the PQM device. Each signal is injected for +-10 seconds to obtain 4 to 5 samples in the log file.



```
A Phase

Voltage: 0.00
Current: 0.000
Watt: 0.00
VA: 0.00
VAR: 0.00
Freq: 50.00
VTHD: 0.00
ITHD: 0.00
Power Factor: 1.00

Dip Level: 200.00
Dip Cycles: 10
Dip Value: 0.35

Swell Level: 255.00
Swell Cycles: 10
Swell Value: 0.00

No Zero Crossing
No Dip
No Swell
```

```
Current      Voltage

0.018        0.017
0.018        0.017
0.018        0.017
0.018        0.017
0.018        0.017
0.018        0.017
0.018        0.017
0.018        0.017
1.055        240.171
1.046        240.157
1.042        240.143
1.045        240.148
1.052        240.137
1.061        240.139
0.018        0.017
0.018        0.017
0.018        0.017
```

Figure 7-8: Left: Data output window Right: logfile

8. Once the test has been completed, the data log file may be copied from the Raspberry Pi to the user PC using an FTP client such as WinSCP. Then the file may be deleted by navigating back into the folder and using the “rm Data_Log.txt” command. The actual value is obtained by calculating an average of the samples obtained in the log file.

7.3.4. Results and analysis

Table 7-2 details the results obtained from the voltage test; the desired value specified in the test module is presented in the first column as the variable that is adjusted between each test. Voltage measurements from both the multimeter and PQM (power quality monitoring) device are presented in the next two columns to illustrate the difference between the two readings. Finally, the last column shows the deviation of the measurement of the PQM device from the multimeter reading.

This value is obtained using the following Equation (7.1):

$$\%d = 100 - ((V_b \div V_a) \times 100) \tag{7.1}$$

Where:

%d = Percentage deviation

V_b = PQM device voltage measurement

V_a = Multimeter voltage measurement

Table 7-2: Voltage Measurement Accuracy

Injected Value (V)	**Voltage Measurements (V)**		**Device Deviation from Multimeter (%)**
	Multimeter	**PQM Device**	
150	150.4	150.009	- 0.26
160	160.4	160.003	-0.248
170	170.4	169.996	-0.237
180	180.4	179.994	-0.225
190	190.5	189.985	-0.27
200	200.5	199.985	-0.257
210	210.5	209.98	-0.247
220	220.5	219.97	-0.241
230	230.5	229.964	-0.233
240	240.6	239.959	-0.267

The following observations can be made according to the results obtained:

- The multimeter consistently recorded values 0.4 V higher than the value specified in Test Universe, confirming a stable voltage source and multimeter readings.
- Readings obtained from the device are consistently lower than the multimeter readings; the Voltage Constant value derived in the previous chapter may be modified to calibrate readings according to the reference meter.
- The average deviation from the multimeter reading by the power quality monitoring device is -0.2482%, indicating that readings are close to the reference meter’s measured value, and the difference is negligible.
- A constant deviation can be observed, with only a 0.045% difference between the highest (0.27%) and the lowest (0.225%) recorded deviation.

Overall, the results indicate a high level of consistency and an acceptable level of accuracy across several voltage levels. Figure 7-9 plots the two voltage readings in a single graph to illustrate the lack of any noticeable deviation between the measurements taken on the multimeter vs the PQM device.

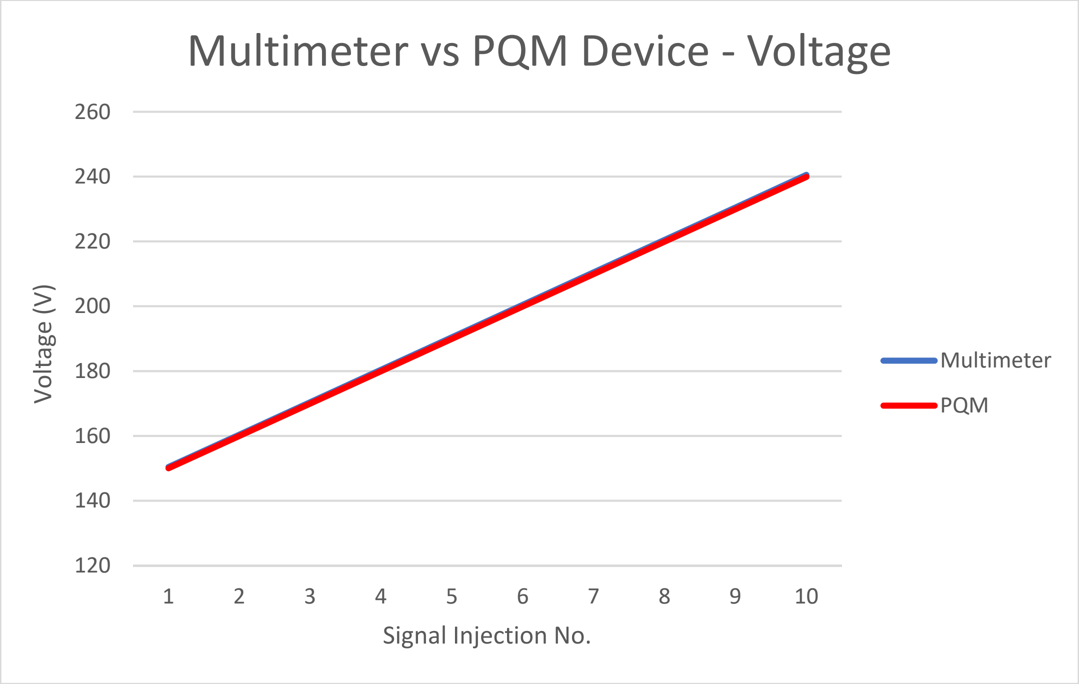


Figure 7-9: Chart showing the deviation between the multimeter and PQM readings

Table 7-3 shows the current measurements taken by both the multimeter and the PQM device. The test methodology is identical to the voltage accuracy test; as such, the measurement deviation may be calculated using the following Equation (7.2):

$$\%d = 100 - ((I_b \div I_a) \times 100) \tag{7.2}$$

Where:

%d = Percentage deviation

V_b = PQM device voltage measurement

V_a = Multimeter voltage measurement

The results which are displayed in Table 7-3 may be summarized as follows:

- Reference multimeter readings indicate a stable current signal supplied by the CMC, with the multimeter's measured value slowly inching downwards as the supply current is increased towards the 10A maximum rating.
- An average deviation of +0.604% between the multimeter and the PQM device readings is recorded.
- The deviation value may be observed as being unstable with the difference between the highest (+2.718%) and lowest (-0.1%) deviation value being 2.818%.
- It should be noted that the value of the highest recorded deviation is measured at 500 mA, considering the CT utilized for the test is has a ratio 100:0.05, a variation of 2.818% at 500mA on the primary side is acceptable.
- The low output current, coupled with the non-linear properties of the current transformer may lead to some anomalous readings at lower primary current values.

Table 7-3: Current Measurement Accuracy

Injected Value (A)	**Current Measurements (A)**		**Device Deviation from Multimeter (%)**
	Multimeter	**PQM Device**	
0.5	0.501	0.515	+2.718
1	1.002	1.014	+1.183
1.5	1.503	1.505	+0.132
2	2.004	2.006	+0.0998
2.5	2.504	2.510	+0.24
3	3.004	3.001	-0.1
3.5	3.502	3.505	+0.087
4	4.002	4.021	+0.475
4.5	4.501	4.525	+0.533
5	4.999	5.013	+0.28
5.5	5.497	5.527	+0.545
6	5.994	6.035	+0.684
6.5	6.491	6.54	+0.75
7	6.99	7.037	+0.672
7.5	7.49	7.543	+0.708
8	7.99	8.042	+0.651

Based on the results, it can be concluded that the meter can achieve an above-average level of accuracy. The accuracy can be further improved using the ADE9000's phase calibration capabilities, which can correct the nonlinearities of current transformers to achieve Class 0.2 accuracy. Selection of a current transformer with a smaller range may also yield improved accuracy for the current readings.

Figure 7-10 plots the multimeter and PQM device readings as two series to further illustrate the lack of deviation between the readings of the two devices.

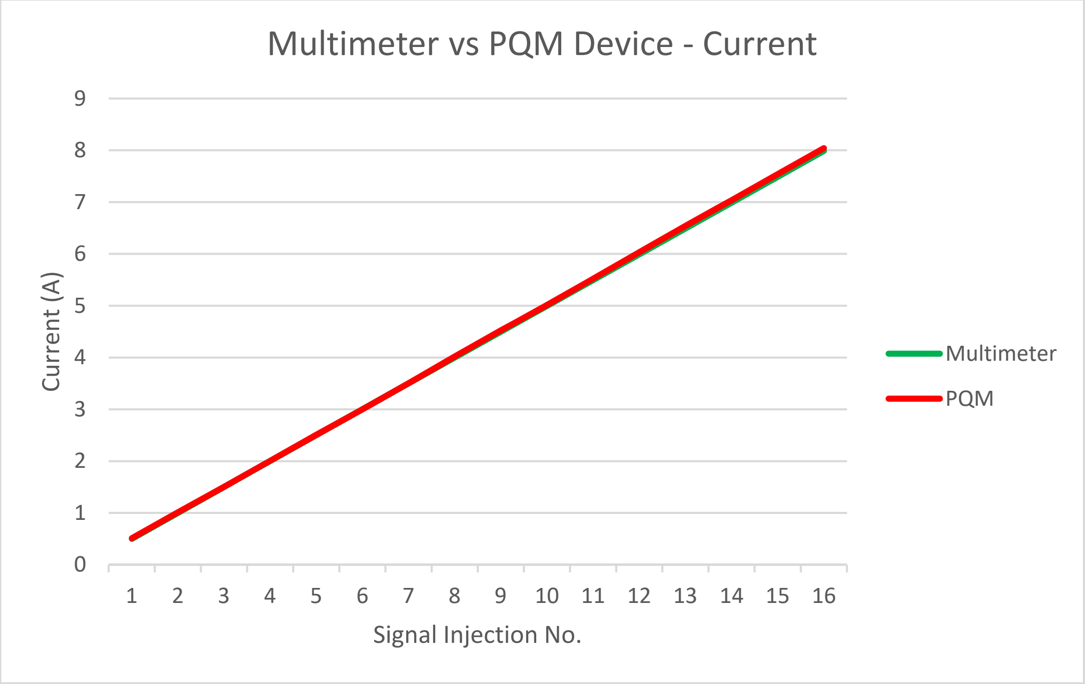


Figure 7-10: Chart showing the deviation between the multimeter and PQM readings

7.4. Test 2: Harmonics measurement validation

The power quality tests are intended to test the efficacy and accuracy of implemented power quality metrics. The harmonics test's objective is to verify the total harmonic distortion (THD) calculation algorithm on the ADE9000. As previously mentioned, harmonics are a crucial part of power quality measurements in distributed generators due to the repetitive switching action of power electronic converters that may inject harmonic signals. Even ignoring DERs, power electronic loads such as variable speed drives (VSDs) and switching power supplies are some of the most common sources of harmonics (Grady, 2012).

7.4.1. Required hardware and software

The required hardware and software remain the same as in the previous test. With the only difference being the usage of the Harmonics Test Tool instead of the QuickCMC test module. The following hardware and software are required for the test:

Hardware

- Omicron CMC 356
- Agilent U1232A Multimeter
- Windows PC
- Raspberry Pi
- ADE 9000

Software

- Microsoft Word
- VNC Viewer and FTP Client
- Raspberry Pi Data Acquisition Software
- Omicron Test Universe 4.00
    - Harmonics Test Tool

7.4.2. Test methodology

1. The OMICRON CMC 356, hardware connections and test universe module may be configured as described in the measurement accuracy test. Refer to points 1 to 3 in section 7.3.3 for details. For the Harmonic Test Tool, a test object does not need to be defined; as such, only the CMC hardware must be configured.
2. Once the hardware configuration is complete, the signals may be defined as shown in Figure 7-6. Similar to the measurement accuracy test, only a single phase is tested, since the computational algorithms on each phase are identical. The voltage signal is configured to 230 VAC at 0 degrees and the current signal at 1 A at 0 degrees.

   Harmonic signals ranging from the 2nd to 10th harmonic order at a magnitude of 10% of the fundamental phase shifted at 0% is injected. Each subsequent signal injection includes an additional harmonic component, and the THD determined using Equation (7.3), as introduced in Chapter 1:

$$THD = \sqrt{\sum_{n=2}^{H} \left(\frac{G_n}{G_1}\right)^2} \tag{7.3}$$

   Where:

   G_n is the Harmonic component

   n is the harmonic order: running a number

   G_1 is the Fundamental component

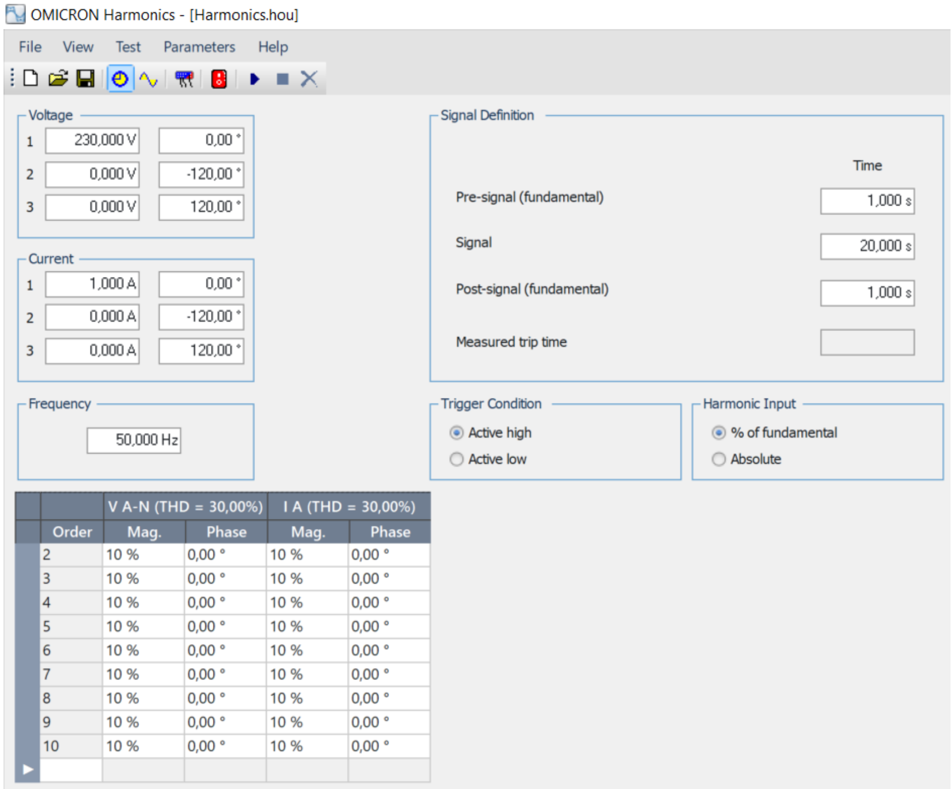
H is the Harmonic order

In practice, the total harmonic distortion for the 2nd harmonic component as well as the 2nd and third combined may be determined as follows:

$$THD_2 = \sqrt{\left(\frac{G_2}{G_1}\right)^2} = \sqrt{\left(\frac{230 \times 0.1}{230}\right)^2} = 0.1 = 10\% \qquad (7.4)$$

$$THD_{2+3} = \sqrt{\left(\frac{G_2}{G_1}\right)^2 + \left(\frac{G_3}{G_1}\right)^2} = \sqrt{\left(\frac{230 \times 0.1}{230}\right)^2 + \left(\frac{230 \times 0.1}{230}\right)^2} = 0.1414 = 14{,}14\% \qquad (7.5)$$

3. The harmonic test is done up to 30% THD, which is achievable by injecting harmonic components up to the 10th order at a magnitude of 10% fundamental. Figure 7-11 shows the signal configuration for the final test at 30%, and preceding tests are performed up to the specified harmonic order. The signal is injected for 20 seconds to obtain as many samples as possible. Each signal is set at a standard 50 Hz frequency.



Voltage

1	230,000 V	0,00 °
2	0,000 V	-120,00 °
3	0,000 V	120,00 °

Current

1	1,000 A	0,00 °
2	0,000 A	-120,00 °
3	0,000 A	120,00 °

Frequency: 50,000 Hz

Signal Definition

	Time
Pre-signal (fundamental)	1,000 s
Signal	20,000 s
Post-signal (fundamental)	1,000 s
Measured trip time	

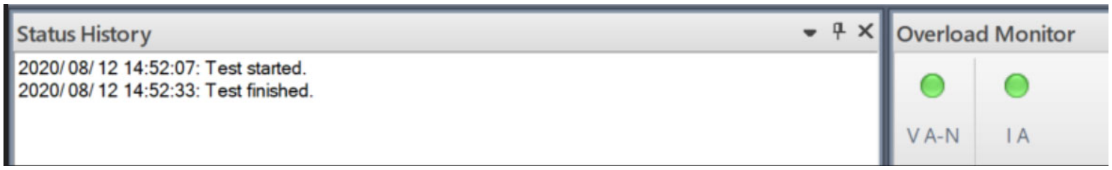
Trigger Condition: ● Active high ○ Active low

Harmonic Input: ● % of fundamental ○ Absolute

	V A-N (THD = 30,00%)		I A (THD = 30,00%)	
Order	Mag.	Phase	Mag.	Phase
2	10 %	0,00 °	10 %	0,00 °
3	10 %	0,00 °	10 %	0,00 °
4	10 %	0,00 °	10 %	0,00 °
5	10 %	0,00 °	10 %	0,00 °
6	10 %	0,00 °	10 %	0,00 °
7	10 %	0,00 °	10 %	0,00 °
8	10 %	0,00 °	10 %	0,00 °
9	10 %	0,00 °	10 %	0,00 °
10	10 %	0,00 °	10 %	0,00 °

Figure 7-11: Harmonics Test Tool Configuration

4. Finally, when the test tool has been configured, the signal injection is performed by selecting the "start" button. Figure 7-12 shows the test and overload monitor status; a message appears indicating whether the test has started or finished.



2020/ 08/ 12 14:52:07: Test started.
2020/ 08/ 12 14:52:33: Test finished.

Figure 7-12; Harmonic test status and overload monitor

7.4.3. Results and analysis

It is worth noting that the THD is computed using the following Equation (7.6) on the ADE9000:

$$AxTHD = \sqrt{\frac{AxRMS^2 - AxFRMS^2}{AxFRMS^2}} \tag{7.6}$$

This formula is different from the one described in the methodology section but should produce the same result. The above formula is described in more detail in chapter 6 section 5 (6.5) as well as the equation for converting to percentage.

Table 7-4 shows the results of the signal injection tests. The first three columns represent the test universe settings along with the expected THD. The first column indicates the harmonic orders of the harmonic components being injected as part of the signal; the second column shows the magnitude of the distortion as a percentage of the fundamental, and finally, the third column specifies the expected THD reading. Actual measurements made by the PQM device are shown in the final four columns, with the first two indicating the voltage measurement and THD on the voltage signal respectively and the last two the current measurement and THD on the current signal respectively.

Table 7-4: Total Harmonic Distortion Measurements up to the 10th Harmonic Order

Test Universe Injected Signals			Actual Measurements			
Harmonic Order	**Mag. Per Order (%)**	**Expected THD (%)**	**Voltage meas. (V)**	**Voltage THD (%)**	**Current meas. (A)**	**Current THD (%)**
2	10	10	231.206	10.003	1.009	10.077
2, 3	10	14.14	232.343	14.142	1.014	14.28
2, 3, 4	10	17.32	233.481	17.316	1.019	17.482
2, 3, 4, 5,	10	20	234.624	19.992	1.024	20.19
2, 3, 4, 5, 6,	10	22.36	235.755	22.348	1.029	22.575
2, 3, 4, 5, 6, 7	10	24.49	236.835	24.475	0.985	26.06
2, 3, 4, 5, 6, 7, 8	10	26.46	237.932	26.429	0.995	27.963
2, 3, 4, 5, 6, 7, 8, 9	10	28.28	239.03	28.246	0.981	30.158
2, 3, 4, 5, 6, 7, 8, 9, 10	10	30	240.112	29.95	0.990	32.058

On analysis, the following may be determined from the results:

- The distortion on the measured signal affects the instantaneous voltage and current readings as expected with the voltage reading increasing proportionally to the THD and the current inversely proportional; demonstrating how distorted signals could potentially damage sensitive electrical equipment.
- Actual THD readings for the voltage very closely match the expected THD value as specified by the test universe harmonics tool. A minimal deviation from the expected value is observed across all THD readings
- The current THD readings exhibit a much higher variation in the differences between the expected signal value vs the measured. As previously mentioned, the non-linearity of the current transformer as well as the low secondary output current (maximum 50mA) affects the accuracy of current readings on the lower end of the scale.

In conclusion, the total harmonic distortion measurements show an excellent level of accuracy for the voltage THD and an acceptable level of accuracy for current THD. The selection of a current transformer with a smaller range, of perhaps 0 to 10 A, may potentially yield better results. However, this test still proves the usefulness of the current harmonic if the intention is to measure network with high power draw or rotating machines.

7.5. Test 3: Additional Power Quality verification

The additional power quality measurements available on the ADE9000 are based on calculations derived from the voltage and current signals. With that in mind, minimal samples are required to determine whether the calculation is correctly computed.

7.5.1. Frequency measurements

The frequency is computed on the ADE9000 using an adapted version of the standard formula for calculating frequency. Traditionally, frequency is measured using the following formula:

$$f_L = \frac{1}{T}(Hz) \tag{7.7}$$

Where

f = frequency

T = period of the signal

On the ADE9000 a similar calculation is made, only with the addition of conversion factors. The following equation, as described in section 6.4.2 of Chapter 6, is used for calculating the frequency:

$$f_L = \frac{8000 \times 2^{16}}{xPERIOD + 1}(Hz) \tag{7.8}$$

To test the frequency, QuickCMC is used to inject frequency values from 47Hz to 50Hz using the same procedure as described in section 7.3.3. Each frequency value is injected with standard voltage and current values of 230V and 5 A, respectively.

Table 7-5: Calculated Frequency Results

Expected Frequency (HZ)	Period Measurement		Calculated Frequency (HZ)
	Raw Value	Converted (Sec)	
47	11155063	0.021277	47
47.5	11037641	0.021053	47.5
48	10922666	0.020833	48
48.5	10810060	0.020619	48.5
49	10699754	0.020408	49
49.5	10591675	0.020202	49.5
50	10485759	0.02	50

The results as shown in Table 7-5 indicate a very accurate period computation algorithm on the ADE9000 over the seven samples obtained, with the raw decimal values acquired from the device converting to seconds using the equation:

$$T_L = \frac{xPERIOD + 1}{8000 \times 2^{16}} (Sec) \tag{7.9}$$

Then these values may then be converted to frequency using equation (7.7) or directly using the raw values using equation (7.8).

7.5.2. Power measurements

The power measurements are computed internally, with the result being stored in the xWATT/xVA/xVAR registers, with the letter “x” denoting the from which specific phase the measurement originates. For this implementation, there is not a large focus on power measurements due to the inability to accurately calibrate the registers without a reference meter. Also, the fact that the available OMICRON CMCs do not contain the required license to use the “Power” test module used to verify power readings.

As such, all injections done using QuickCMC are performed at a 0-phase angle between and current and voltage signal and thus unity power factor, and this invalidates any controlled tests conducted since the expected WATT value is always equal VA.

Confirmation of the statement mentioned above was done by injecting several current values at 230V using the method for injecting and acquiring data from the Raspberry Pi described in section 7.3.3. Table 7-6 shows the test results obtained from the signal injections.

Table 7-6: Power and Power Factor Measurements

Injected Signals (A / V)	Power Measurements		Power Factor
	Watts	VA	
1 / 230	231.504	231.697	0.9992
2 / 230	460.025	459.756	1.0006
5 / 230	1150.205	1152.817	0.9977

The results show the expected equal real and apparent power measurements which result in a virtually unity power factor. The power factor value of 1 means that cosθ = 1, this results in real and apparent power values that can be derived as follows:

$$P_{Watt} = VI \cos\theta = VI = P_{VA} \tag{7.10}$$

To conclude, at the minimum, this test proves the calculation algorithm in the ADE9000 can compute real and apparent power values but does not confirm that it can accurately account for phase angle differences between the voltage and current signals and produce the expected results.

7.6. Test 4: IEC 61850 MMS and ACSI Service mapping verification

This section aims to test and prove IEC 61850 conformance of the PQM device by examining the Protocol Data Unit (PDU) requests and responses during MMS transactions. IEC 61850-10 describes a few conformance tests that may be performed to prove conformance of the IED being assessed to specific parts and subclauses of the IEC 61850 standard. Figure 7-13 shows the system overview with the modules being evaluated and verified as green rectangles. The MMS and ACSI services form part of the IEC 61850 protocol stack running on Raspberry Pi and provided by LibIEC61850.

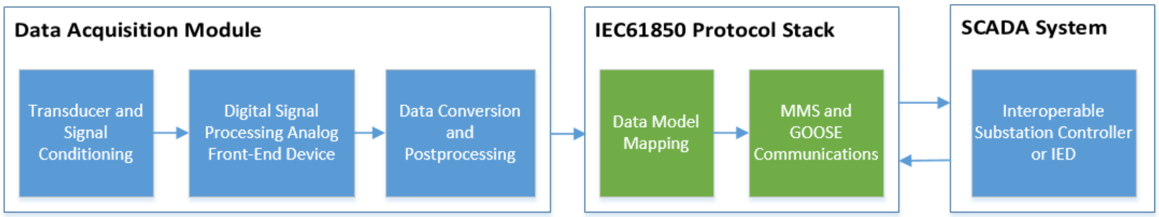

Data Acquisition Module
Transducer and Signal Conditioning
Digital Signal Processing Analog Front-End Device
Data Conversion and Postprocessing
IEC61850 Protocol Stack
Data Model Mapping
MMS and GOOSE Communications
SCADA System
Interoperable Substation Controller or IED

Figure 7-13: System module being assessed

7.6.1. Required hardware and software

The following hardware and software are required for the IEC 61850 MMS and GOOSE verification test:

Hardware

- ADE9000
- Windows PC
- Raspberry Pi

Software

- LibIEC61850 Software Library
- Raspberry Pi Data Acquisition Software
- IEDExplorer Software

7.6.2. IEC 61850-10 Conformance Testing and procedure overview

IEC 61850-10 includes a section for the Abstract Communication Service Interface (ACSI). This section is designed to test services specified in the information models defined in IEC 61850-7-2 (IEC, 2005a), these tests are designed to assess the PQM device and mapping of the services as mentioned earlier to their equivalent MMS service. The following ACSI service types have tests associated with them within IEC 61850-10 (IEC, 2005b):

- Application association (Ass);
- Server, Logical device, Logical node, Data, and Data Attribute model (Srv);
- Data set model (Dset);
- Setting group model (Sg);

- Report control model (Rpt);
- Log control model (Log);
- Control model (Ctl);
- Substitution model (Sub);
- Transmission of sampled values model (Sv);
- Time and time synchronisation model (Tm);
- File transfer model (Ft).
- Generic object-oriented system-wide events (Goo);

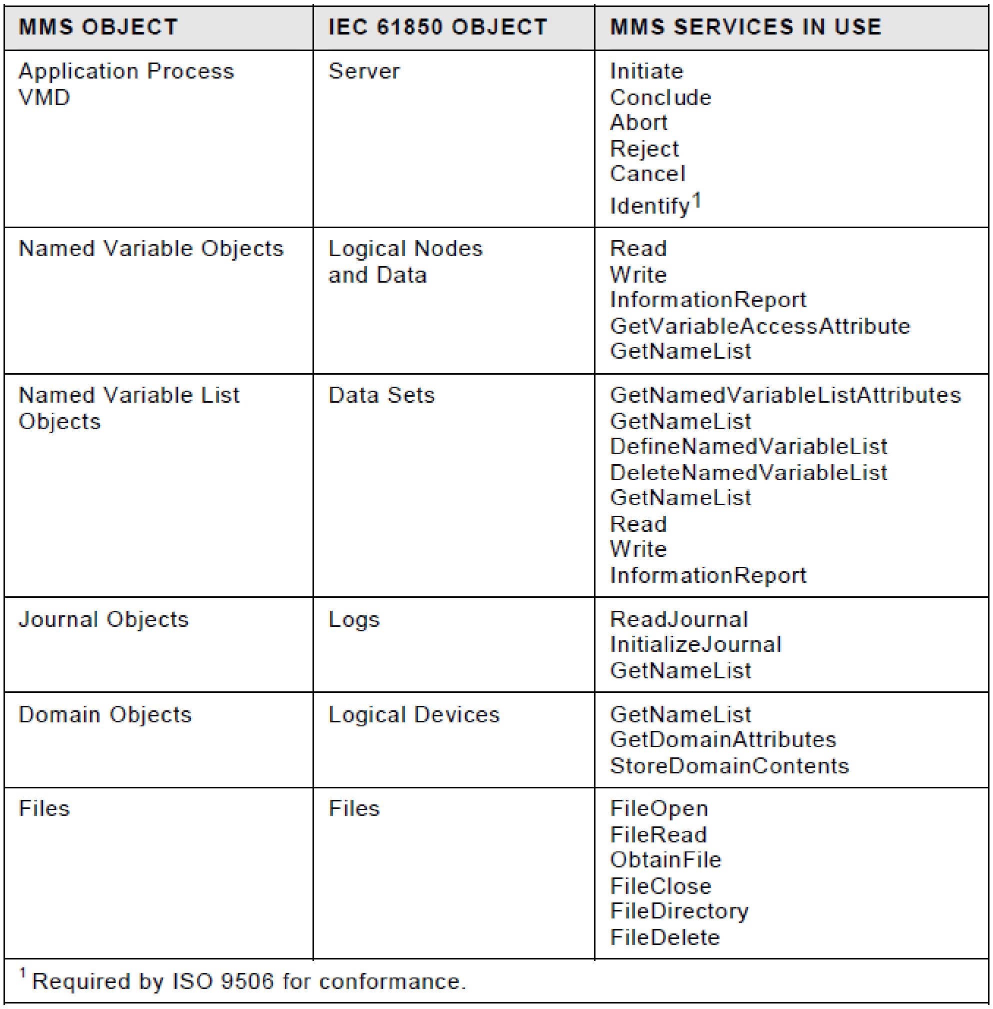
The services are associated with many test cases, and these test cases are tabled and identified in IEC 61850-10 by a unique acronym, along with a test number and an optional "N" denotation if the pass condition requires a negative response from the device being assessed. Test cases are expected to generate positive results unless the aforementioned "N" is present. For example, Table 7-7 shows several positive test cases for the application association service. Application association has a unique acronym of "Ass", and each separate test case is identified by an appending number (e.g. test case 1 is "Ass1"). Negative test cases are named similarly, only with the letter "N" before the test number (e.g. AssN1). Each test's description also specifies the IEC 61850 part as well as the subclause that is being proven.

Table 7-7: Application Association Positive Test Cases

Test	**Description**
Ass1	Associate and release a TPAA association (IEC 61850-7-2 Subclause 7.4)
Ass2	Associate and client-abort TPAA association (IEC 61850-7-2 Subclause 7.4)
Ass3	Associate with the maximum number of clients simultaneously (PIXIT)

ACSI services are mapped to MMS services following IEC 61850-8-1; this is known as Specific Communication Services Mapping or SCSM. Table 7-8 shows MMS objects along with the IEC 61850 Objects they are mapped to, the specific services being utilised within the MMS Object to implement the IEC 61850 Object are also specified. With this in mind, it is possible to test the implementation of the ACSI services by using an MMS server to request data from the relevant ACSI services on the PQM device and monitoring the responses.

Table 7-8: MMS Objects and Services used within SCSM (IEC, 2005c)



MMS OBJECT	IEC 61850 OBJECT	MMS SERVICES IN USE
Application Process VMD	Server	Initiate Conclude Abort Reject Cancel Identify[1]
Named Variable Objects	Logical Nodes and Data	Read Write InformationReport GetVariableAccessAttribute GetNameList
Named Variable List Objects	Data Sets	GetNamedVariableListAttributes GetNameList DefineNamedVariableList DeleteNamedVariableList GetNameList Read Write InformationReport
Journal Objects	Logs	ReadJournal InitializeJournal GetNameList
Domain Objects	Logical Devices	GetNameList GetDomainAttributes StoreDomainContents
Files	Files	FileOpen FileRead ObtainFile FileClose FileDirectory FileDelete

[1] Required by ISO 9506 for conformance.

The IED Explorer software can be used as a testing platform for proving the conformance of various MMS and ACSI service of the PQM device. IED Explorer is a free, open-source software package that allows the end-user to connect to an IED, inspect MMS variables and capture MMS packets, among other features (Pavel-ch, 2020).

Figure 7-14 is a diagram that details the structure of the communications network for testing MMS and ACSI services using IED Explorer. The software is installed on a Windows PC, which is networked with the Raspberry Pi via an Ethernet Network. IED Explorer mainly functions as an MMS client, sending MMS service requests, but also can send Goose messages (this function is not used). The MMS service requests are sent using the MMS protocol via TCP/IP, the messages are received by the Raspberry Pi and mapped with ACSI services according to IEC 61850-8-1 SCSM. The response from the Raspberry Pi is sent using LibIEC61850 functions, which are mapped back from ACSI to MMS services.

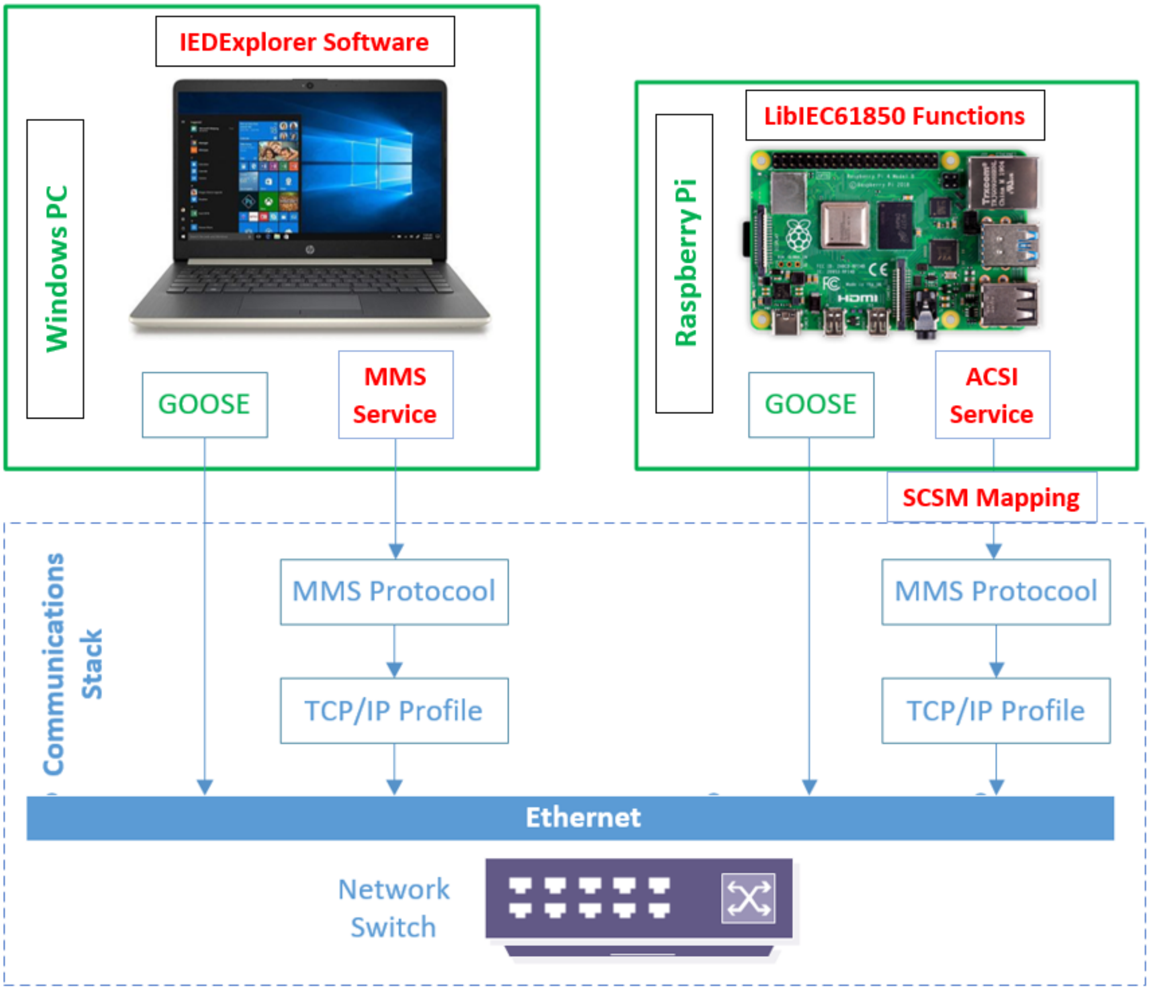
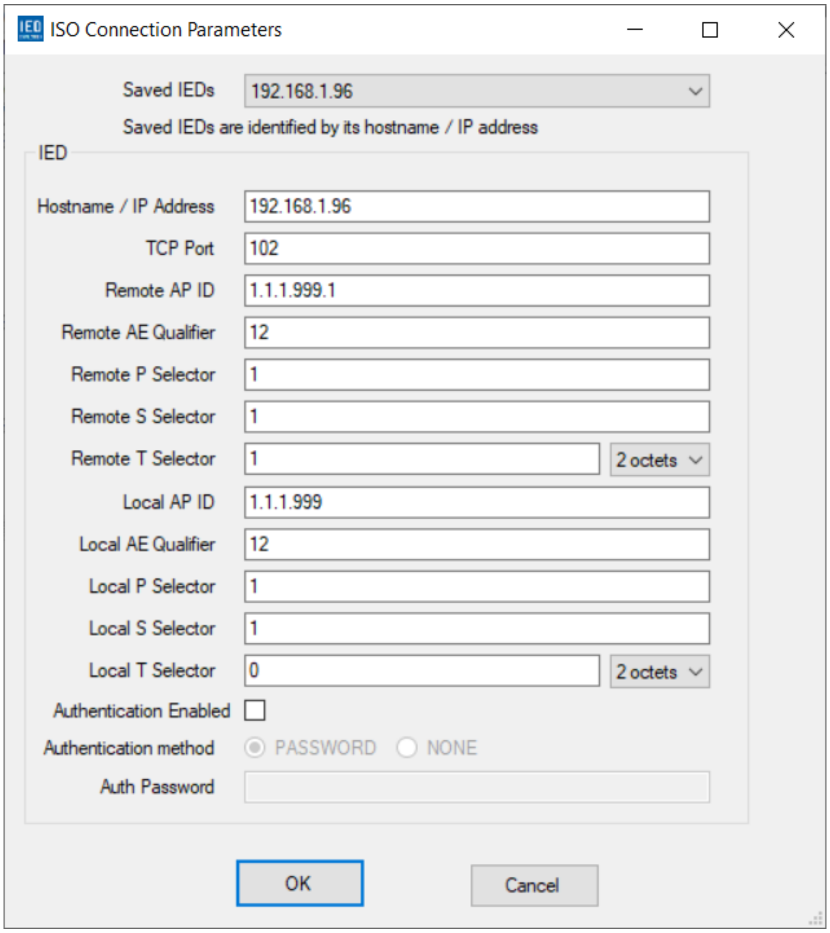


Figure 7-14: Using IED Explorer to Connect with an IEC 61850 Compliant Device

7.6.3. MMS and GOOSE performance test procedure and results

Figure 7-15 shows the ISO (MMS) connection parameters for the IED; only the IP address and TCP port numbers are mandatory to connect to an IED successfully. Once the remote IED has been configured, selecting “CaptureView” and “start capture” on the toolbar starts the capturing MMS packets. Finally selecting the “run” button on the taskbar initiates the MMS transactions, once communications have been initiated, MMS packets appear in the “CaptureView” window as shown in Figure 7-15.



Saved IEDs: 192.168.1.96

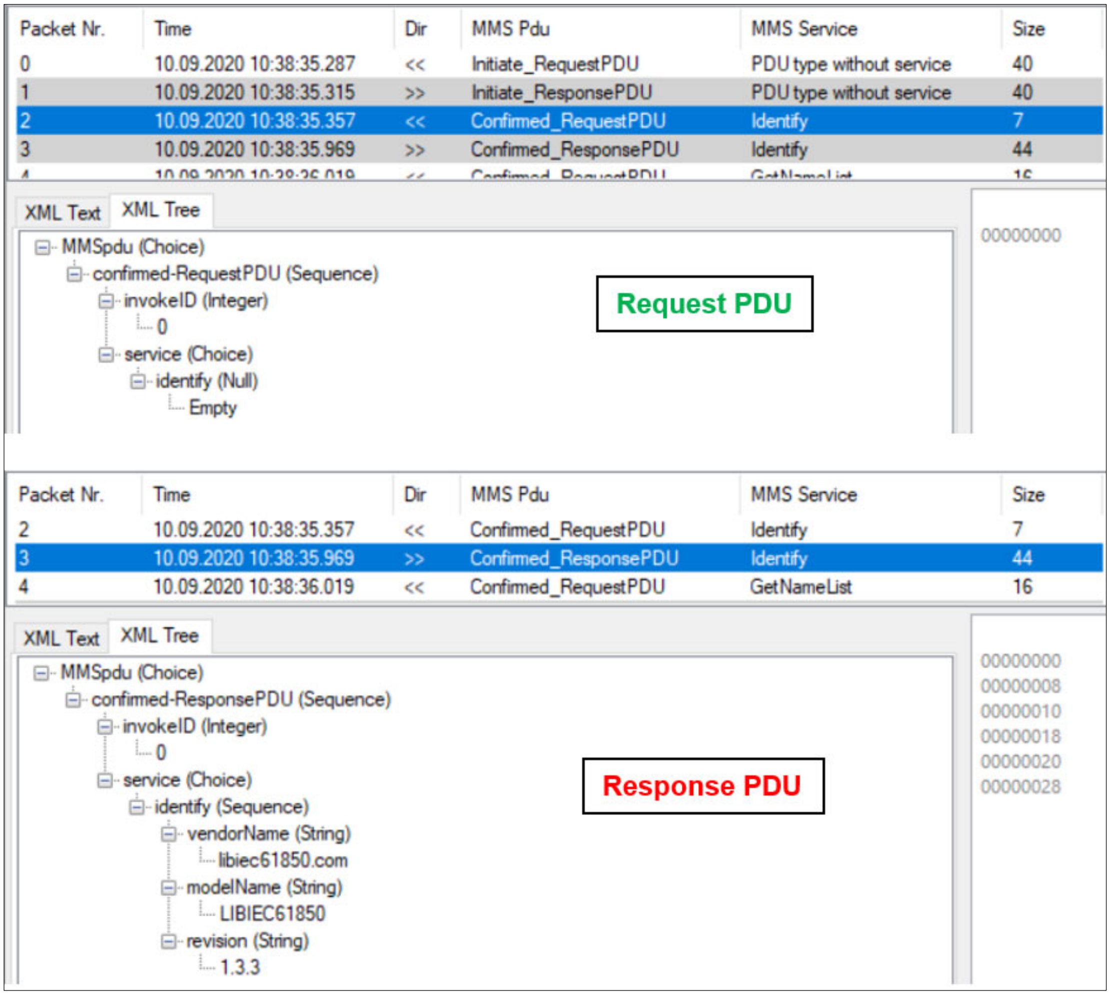
Saved IEDs are identified by its hostname / IP address

IED

Hostname / IP Address	192.168.1.96
TCP Port	102
Remote AP ID	1.1.1.999.1
Remote AE Qualifier	12
Remote P Selector	1
Remote S Selector	1
Remote T Selector	1 (2 octets)
Local AP ID	1.1.1.999
Local AE Qualifier	12
Local P Selector	1
Local S Selector	1
Local T Selector	0 (2 octets)
Authentication Enabled	□
Authentication method	● PASSWORD ○ NONE
Auth Password	

Figure 7-15:: IED Explorer ISO Connection Parameters

Figure 7-16 shows the transaction for the identify service captured within the “CaptureView” window, “Identify” is considered a mandatory service for the ISO 9506 standard (IEC, 2005c). In this case, the PQM device is successfully identified at an MMS level. The “Identification” service forms a part of the initial start-up of MMS communications, after successful identification, MMS transactions may proceed.



Packet Nr.	Time	Dir	MMS Pdu	MMS Service	Size
0	10.09.2020 10:38:35.287	<<	Initiate_RequestPDU	PDU type without service	40
1	10.09.2020 10:38:35.315	>>	Initiate_ResponsePDU	PDU type without service	40
2	10.09.2020 10:38:35.357	<<	Confirmed_RequestPDU	Identify	7
3	10.09.2020 10:38:35.969	>>	Confirmed_ResponsePDU	Identify	44
4	10.09.2020 10:38:36.019	<<	Confirmed_RequestPDU	GetNameList	16

XML Text XML Tree

```
MMSpdu (Choice)
    confirmed-RequestPDU (Sequence)
        invokeID (Integer)
            0
        service (Choice)
            identify (Null)
                Empty
```

00000000

Request PDU

Packet Nr.	Time	Dir	MMS Pdu	MMS Service	Size
2	10.09.2020 10:38:35.357	<<	Confirmed_RequestPDU	Identify	7
3	10.09.2020 10:38:35.969	>>	Confirmed_ResponsePDU	Identify	44
4	10.09.2020 10:38:36.019	<<	Confirmed_RequestPDU	GetNameList	16

XML Text XML Tree

```
MMSpdu (Choice)
    confirmed-ResponsePDU (Sequence)
        invokeID (Integer)
            0
        service (Choice)
            identify (Sequence)
                vendorName (String)
                    libiec61850.com
                modelName (String)
                    LIBIEC61850
                revision (String)
                    1.3.3
```

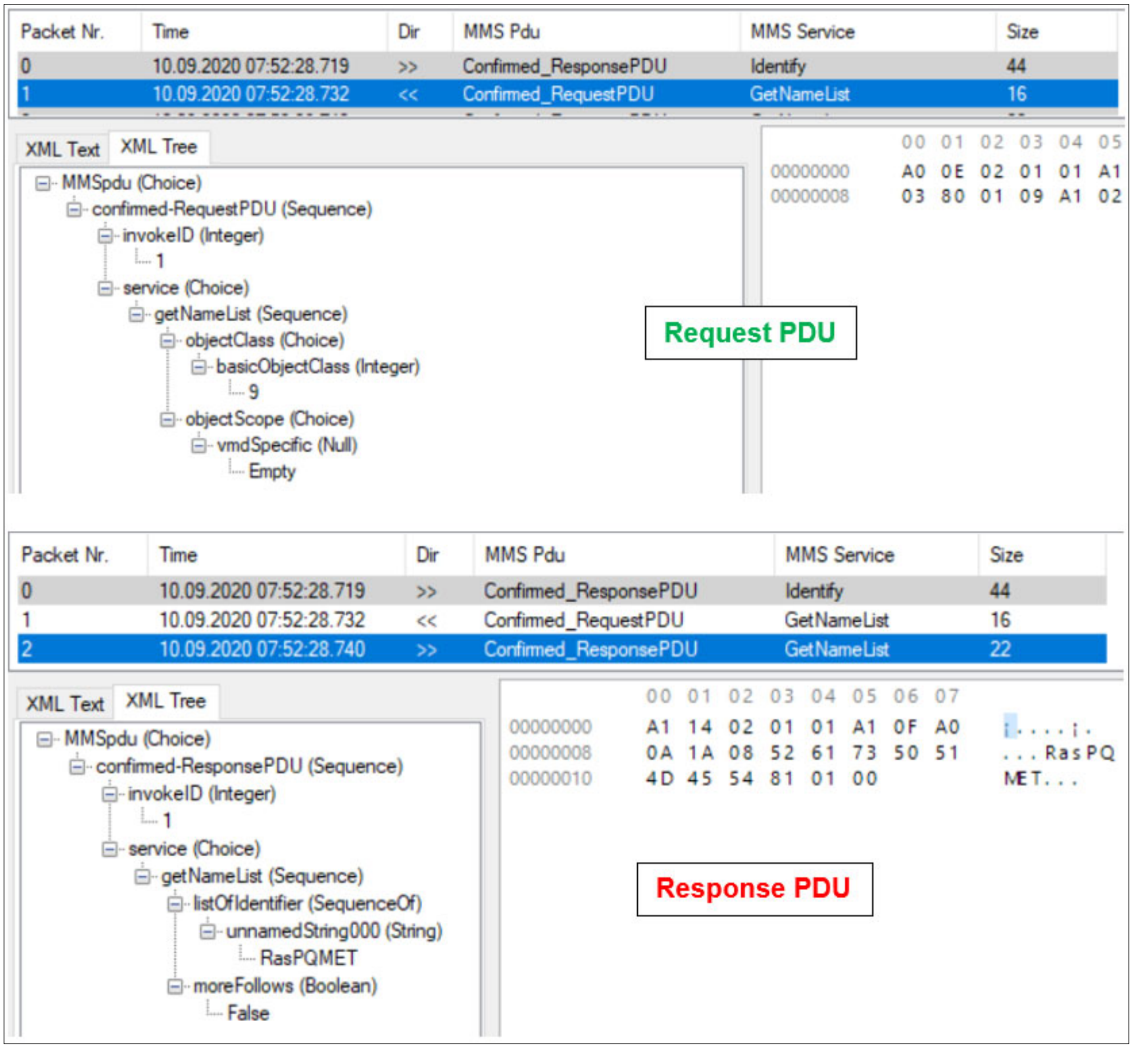
```
00000000
00000008
00000010
00000018
00000020
00000028
```

Response PDU

Figure 7-16: Initiate and Identify MMS Services

Next, Figure 7-17 shows the GetNameList request PDU sent by the IED Explorer software and response PDU sent by the PQM device. The XML tree shows the contents for both the request and the response MMS PDUs. GetNameList of ObjectClass 9 is requested, the prescribed response for the GetNameList service contains a mandatory “list of identifier” sequence, according to the conformance statement table for GetNameList in IEC 61850-8-1 Page 100 (IEC, 2005d). By examining the screenshot in Figure 7-17, the following conclusions can be made:

- The example featured in Figure 7-17 returns RasPQMET, which is the name of the IED (RasPQ) and the Logical Device (MET) within the IED.
- The second GetNameList transaction observed in IED Explorer returns a list of the logical nodes configured within the logical device, MET.
- It thus may be concluded that the PQM device conforms with the standard GetNameList service by responding with a “list of identifier” sequence.



Packet Nr.	Time	Dir	MMS Pdu	MMS Service	Size
0	10.09.2020 07:52:28.719	>>	Confirmed_ResponsePDU	Identify	44
1	10.09.2020 07:52:28.732	<<	Confirmed_RequestPDU	GetNameList	16

```
MMSpdu (Choice)
    confirmed-RequestPDU (Sequence)
        invokeID (Integer)
            1
        service (Choice)
            getNameList (Sequence)
                objectClass (Choice)
                    basicObjectClass (Integer)
                        9
                objectScope (Choice)
                    vmdSpecific (Null)
                        Empty
```

```
            00 01 02 03 04 05
00000000    A0 0E 02 01 01 A1
00000008    03 80 01 09 A1 02
```

Request PDU

Packet Nr.	Time	Dir	MMS Pdu	MMS Service	Size
0	10.09.2020 07:52:28.719	>>	Confirmed_ResponsePDU	Identify	44
1	10.09.2020 07:52:28.732	<<	Confirmed_RequestPDU	GetNameList	16
2	10.09.2020 07:52:28.740	>>	Confirmed_ResponsePDU	GetNameList	22

```
MMSpdu (Choice)
    confirmed-ResponsePDU (Sequence)
        invokeID (Integer)
            1
        service (Choice)
            getNameList (Sequence)
                listOfIdentifier (SequenceOf)
                    unnamedString000 (String)
                        RasPQMET
                moreFollows (Boolean)
                    False
```

```
            00 01 02 03 04 05 06 07
00000000    A1 14 02 01 01 A1 0F A0    ¡....¡.
00000008    0A 1A 08 52 61 73 50 51    ...RasPQ
00000010    4D 45 54 81 01 00          MET...
```

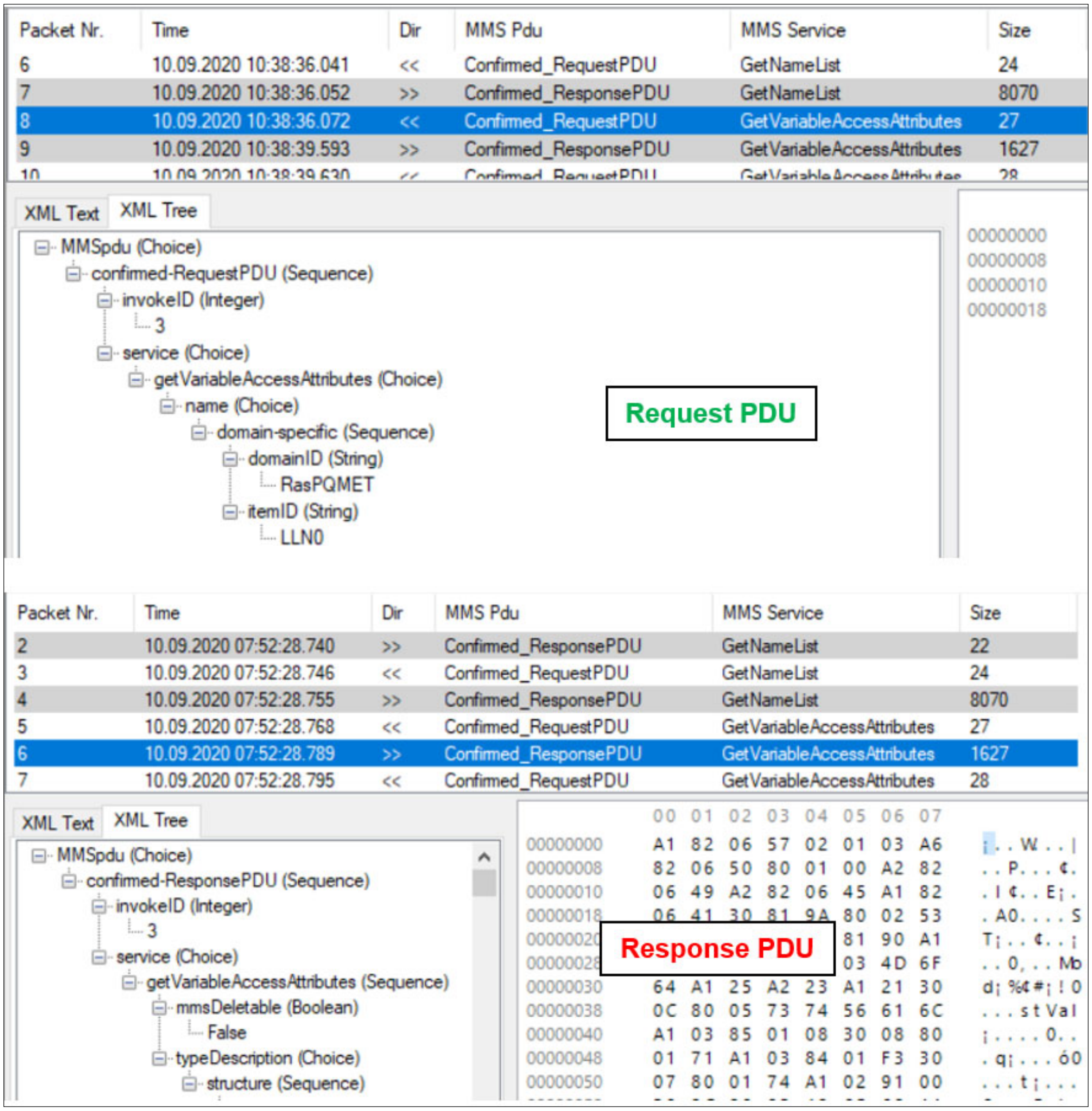
Response PDU

Figure 7-17: GetNameList Request and Response PDUs

In Figure 7-18, the transaction that IED Explorer initiates once the IED and logical device have been identified is illustrated. GetVariableAccessAttributes is an MMS service that is mapped to the GetDataDirectory ACSI service, and the transaction returns typeDescription at an MMS level, which is mapped to DataAttributeName at the ACSI level. Each request returns data about logical node instances available on the IED. In Figure 7-14, the following can be concluded from the response:

- A request is made for the information contained in LLN0 (Logical Node Zero), an itemID within a domainID named RasPQMET.(IED and Logical Device name).
- The response returns the data attribute information of the data objects within the logical node instance LLN0.
- The GetVariableAccessAttributes request is repeated for each subsequent logical node instance: LPH0, LPHD1, MHAI1, MMXU1 and QVVR1.

- It may be concluded that the PQM device conforms with the standard GetVariableAccessAttributes MMS service as stipulated in IEC 61850-8-1.



Packet Nr.	Time	Dir	MMS Pdu	MMS Service	Size
6	10.09.2020 10:38:36.041	<<	Confirmed_RequestPDU	GetNameList	24
7	10.09.2020 10:38:36.052	>>	Confirmed_ResponsePDU	GetNameList	8070
8	10.09.2020 10:38:36.072	<<	Confirmed_RequestPDU	GetVariableAccessAttributes	27
9	10.09.2020 10:38:39.593	>>	Confirmed_ResponsePDU	GetVariableAccessAttributes	1627
10	10.09.2020 10:38:39.630	<<	Confirmed_RequestPDU	GetVariableAccessAttributes	28

```
MMSpdu (Choice)
  confirmed-RequestPDU (Sequence)
    invokeID (Integer)
      3
    service (Choice)
      getVariableAccessAttributes (Choice)
        name (Choice)
          domain-specific (Sequence)
            domainID (String)
              RasPQMET
            itemID (String)
              LLN0
```

```
00000000
00000008
00000010
00000018
```

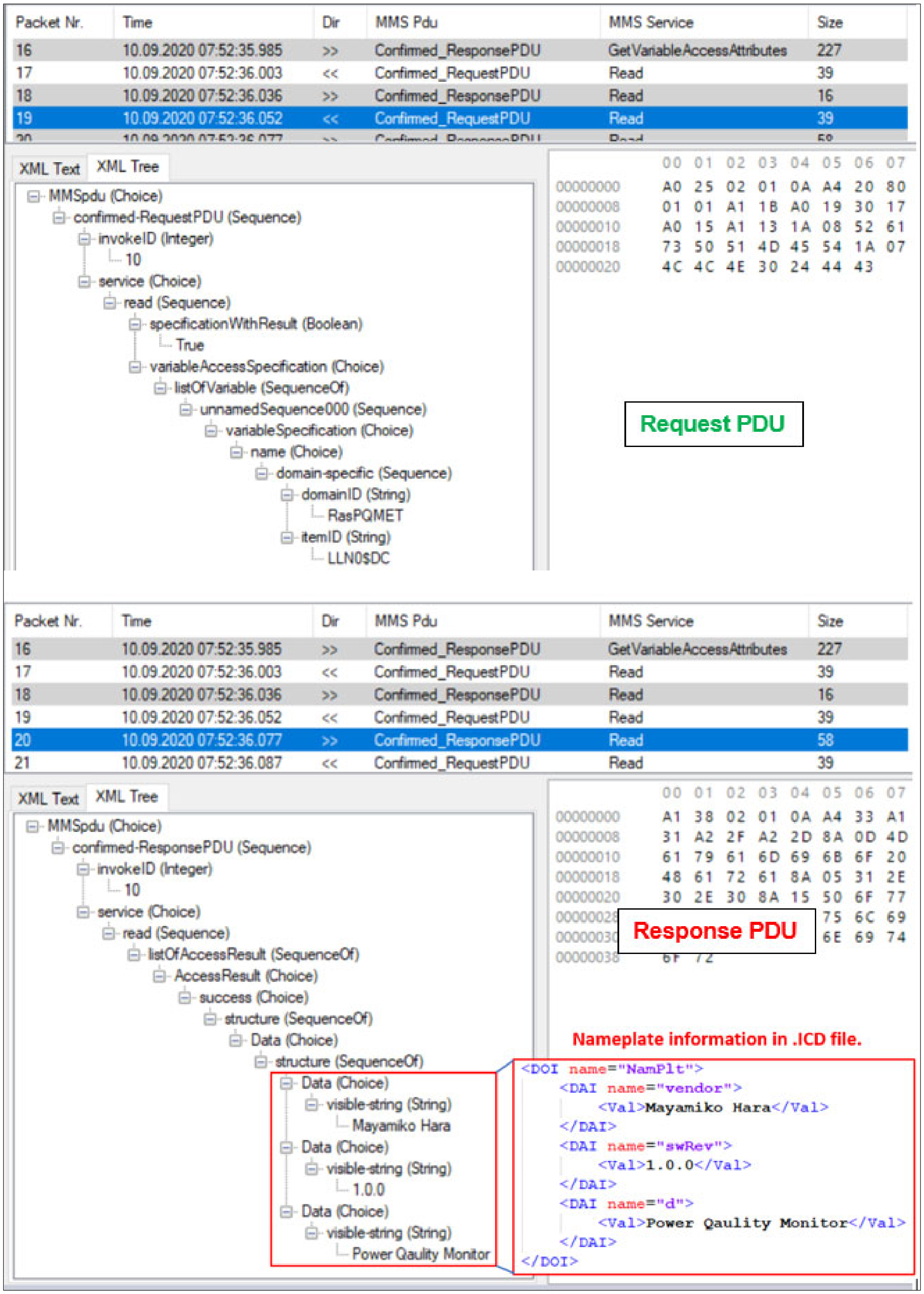
Packet Nr.	Time	Dir	MMS Pdu	MMS Service	Size
2	10.09.2020 07:52:28.740	>>	Confirmed_ResponsePDU	GetNameList	22
3	10.09.2020 07:52:28.746	<<	Confirmed_RequestPDU	GetNameList	24
4	10.09.2020 07:52:28.755	>>	Confirmed_ResponsePDU	GetNameList	8070
5	10.09.2020 07:52:28.768	<<	Confirmed_RequestPDU	GetVariableAccessAttributes	27
6	10.09.2020 07:52:28.789	>>	Confirmed_ResponsePDU	GetVariableAccessAttributes	1627
7	10.09.2020 07:52:28.795	<<	Confirmed_RequestPDU	GetVariableAccessAttributes	28

```
MMSpdu (Choice)
  confirmed-ResponsePDU (Sequence)
    invokeID (Integer)
      3
    service (Choice)
      getVariableAccessAttributes (Sequence)
        mmsDeletable (Boolean)
          False
        typeDescription (Choice)
          structure (Sequence)
```

```
            00 01 02 03 04 05 06 07
00000000    A1 82 06 57 02 01 03 A6   ¡..W...|
00000008    82 06 50 80 01 00 A2 82   ..P...¢.
00000010    06 49 A2 82 06 45 A1 82   .I¢..E¡.
00000018    06 41 30 81 9A 80 02 53   .A0....S
00000020                81 90 A1   T¡..¢..¡
00000028                03 4D 6F   ..0,..Mo
00000030    64 A1 25 A2 23 A1 21 30   d¡%¢#¡!0
00000038    0C 80 05 73 74 56 61 6C   ...stVal
00000040    A1 03 85 01 08 30 08 80   ¡....0..
00000048    01 71 A1 03 84 01 F3 30   .q¡...ó0
00000050    07 80 01 74 A1 02 91 00   ...t¡...
```

Figure 7-18: GetVariableAccessAttributes request and response PDUs

The MMS read service can be mapped to various ACSI services depending on the type of data being read. Figure 7-19 shows a PDU for requesting data from the MET logical device along with the subsequent response from the PQM device; the read MMS service is mapped with the GetAllDataValues ACSI service. The data being requested is of the functional constraint DC (Description), which encompasses the device information (nameplate, vendor, swRev and d). The response provided by the device contains unique nameplate information found only in the ICD file (vendor name "vendor", software revision "swRev", and device "d"). Subsequent read requests are sent to acquire the remaining data values, such as GOOSE control blocks, data sets and the functional constraints of the remaining logical node instances. The packets as mentioned above, prove the conformance of the device as per the MMS read standard described in IEC 61850-8-1 Page 102 (IEC, 2005e) as well as its mapping with the GetAllDataValues ACSI service.



Packet Nr.	Time	Dir	MMS Pdu	MMS Service	Size
16	10.09.2020 07:52:35.985	>>	Confirmed_ResponsePDU	GetVariableAccessAttributes	227
17	10.09.2020 07:52:36.003	<<	Confirmed_RequestPDU	Read	39
18	10.09.2020 07:52:36.036	>>	Confirmed_ResponsePDU	Read	16
19	10.09.2020 07:52:36.052	<<	Confirmed_RequestPDU	Read	39
20	10.09.2020 07:52:36.077	>>	Confirmed_ResponsePDU	Read	58

```
MMSpdu (Choice)
  confirmed-RequestPDU (Sequence)
    invokeID (Integer)
      10
    service (Choice)
      read (Sequence)
        specificationWithResult (Boolean)
          True
        variableAccessSpecification (Choice)
          listOfVariable (SequenceOf)
            unnamedSequence000 (Sequence)
              variableSpecification (Choice)
                name (Choice)
                  domain-specific (Sequence)
                    domainID (String)
                      RasPQMET
                    itemID (String)
                      LLN0$DC
```

```
          00 01 02 03 04 05 06 07
00000000  A0 25 02 01 0A A4 20 80
00000008  01 01 A1 1B A0 19 30 17
00000010  A0 15 A1 13 1A 08 52 61
00000018  73 50 51 4D 45 54 1A 07
00000020  4C 4C 4E 30 24 44 43
```

Request PDU

Packet Nr.	Time	Dir	MMS Pdu	MMS Service	Size
16	10.09.2020 07:52:35.985	>>	Confirmed_ResponsePDU	GetVariableAccessAttributes	227
17	10.09.2020 07:52:36.003	<<	Confirmed_RequestPDU	Read	39
18	10.09.2020 07:52:36.036	>>	Confirmed_ResponsePDU	Read	16
19	10.09.2020 07:52:36.052	<<	Confirmed_RequestPDU	Read	39
20	10.09.2020 07:52:36.077	>>	Confirmed_ResponsePDU	Read	58
21	10.09.2020 07:52:36.087	<<	Confirmed_RequestPDU	Read	39

```
MMSpdu (Choice)
  confirmed-ResponsePDU (Sequence)
    invokeID (Integer)
      10
    service (Choice)
      read (Sequence)
        listOfAccessResult (SequenceOf)
          AccessResult (Choice)
            success (Choice)
              structure (SequenceOf)
                Data (Choice)
                  structure (SequenceOf)
                    Data (Choice)
                      visible-string (String)
                        Mayamiko Hara
                    Data (Choice)
                      visible-string (String)
                        1.0.0
                    Data (Choice)
                      visible-string (String)
                        Power Qaulity Monitor
```

```
          00 01 02 03 04 05 06 07
00000000  A1 38 02 01 0A A4 33 A1
00000008  31 A2 2F A2 2D 8A 0D 4D
00000010  61 79 61 6D 69 6B 6F 20
00000018  48 61 72 61 8A 05 31 2E
00000020  30 2E 30 8A 15 50 6F 77
00000028                 75 6C 69
00000030                 6E 69 74
00000038  6F 72
```

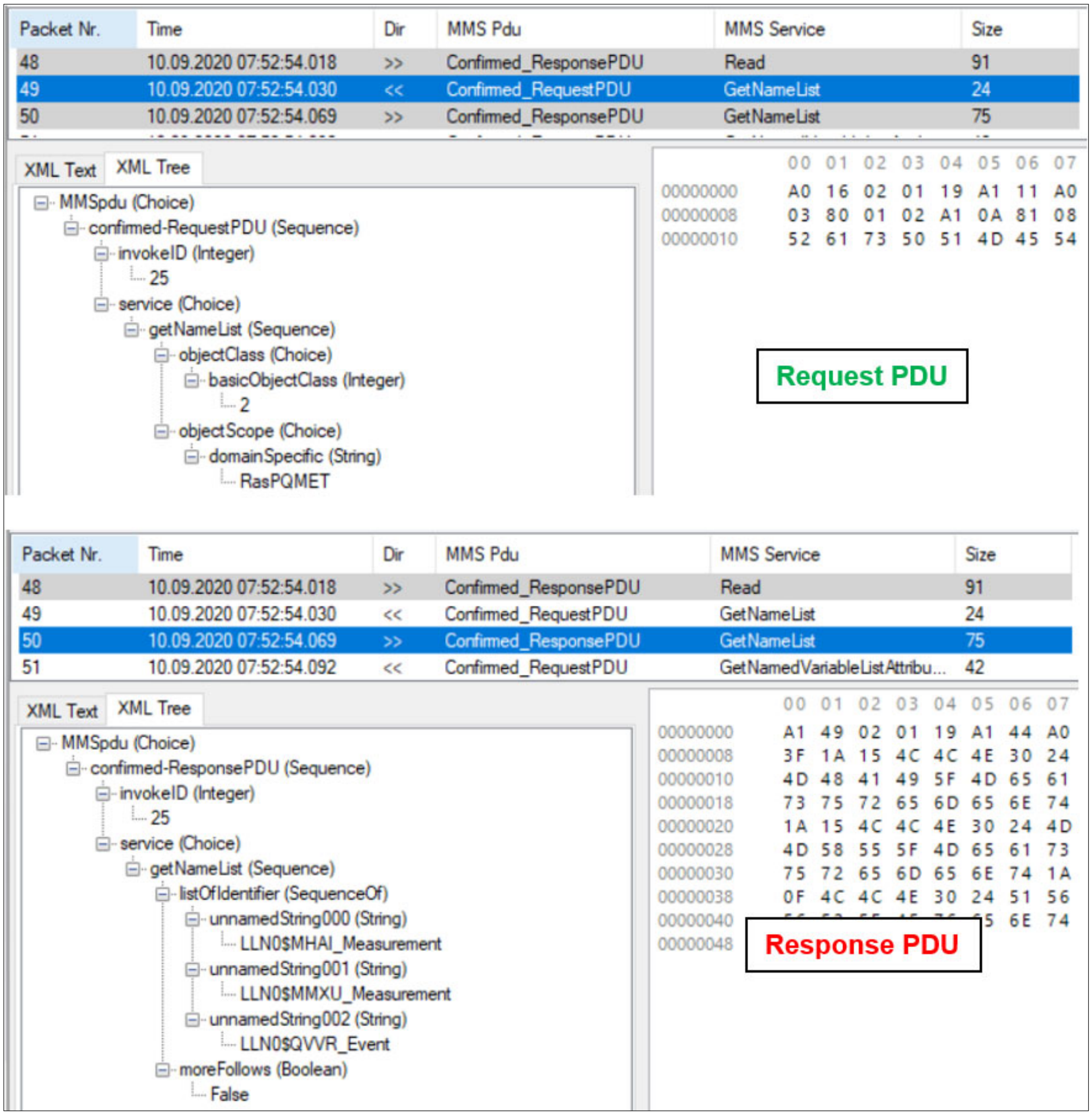
Response PDU

Nameplate information in .ICD file.

```
<DOI name="NamPlt">
    <DAI name="vendor">
        <Val>Mayamiko Hara</Val>
    </DAI>
    <DAI name="swRev">
        <Val>1.0.0</Val>
    </DAI>
    <DAI name="d">
        <Val>Power Qaulity Monitor</Val>
    </DAI>
</DOI>
```

Figure 7-19: Read Service PDU Request and Response

The MMS GetNameList service may also be used to request different types of data based on the value of basicObjectClass. Figure 7-20 shows another example of a GetNameList service request, this time for object class 2 instead of object class 9. The request is initiated by selecting the “IED View” window, which is shown in Figure 7-22 on the IED Explorer toolbar, which lists the Data, DataSets and Unbuffered Reports configurations. As such, a request for object class 2 data within RasPQMET returns a list of the data sets defined within the ICD file, and this is shown in the response PDU which lists the three data sets defined within the SCL file; these are MHAI_Measurement, MMXU_Measurement and QVVR_Event.



```
Packet Nr.  Time                     Dir  MMS Pdu               MMS Service   Size
48          10.09.2020 07:52:54.018  >>   Confirmed_ResponsePDU  Read          91
49          10.09.2020 07:52:54.030  <<   Confirmed_RequestPDU   GetNameList   24
50          10.09.2020 07:52:54.069  >>   Confirmed_ResponsePDU  GetNameList   75
```

```
MMSpdu (Choice)
  confirmed-RequestPDU (Sequence)
    invokeID (Integer)
      25
    service (Choice)
      getNameList (Sequence)
        objectClass (Choice)
          basicObjectClass (Integer)
            2
        objectScope (Choice)
          domainSpecific (String)
            RasPQMET
```

```
            00 01 02 03 04 05 06 07
00000000    A0 16 02 01 19 A1 11 A0
00000008    03 80 01 02 A1 0A 81 08
00000010    52 61 73 50 51 4D 45 54
```

Request PDU

```
Packet Nr.  Time                     Dir  MMS Pdu               MMS Service                    Size
48          10.09.2020 07:52:54.018  >>   Confirmed_ResponsePDU  Read                           91
49          10.09.2020 07:52:54.030  <<   Confirmed_RequestPDU   GetNameList                    24
50          10.09.2020 07:52:54.069  >>   Confirmed_ResponsePDU  GetNameList                    75
51          10.09.2020 07:52:54.092  <<   Confirmed_RequestPDU   GetNamedVariableListAttribu...  42
```

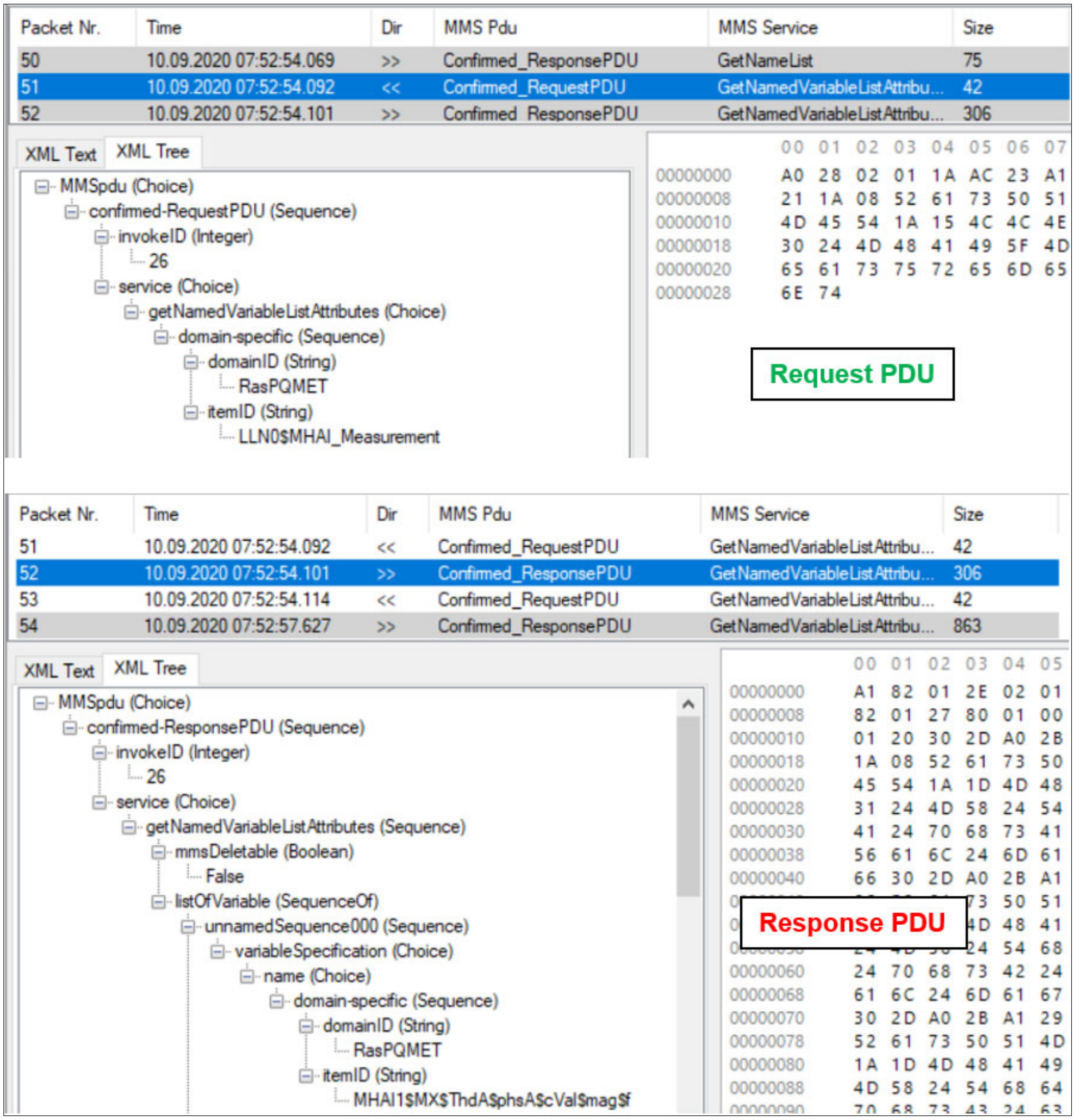
```
MMSpdu (Choice)
  confirmed-ResponsePDU (Sequence)
    invokeID (Integer)
      25
    service (Choice)
      getNameList (Sequence)
        listOfIdentifier (SequenceOf)
          unnamedString000 (String)
            LLN0$MHAI_Measurement
          unnamedString001 (String)
            LLN0$MMXU_Measurement
          unnamedString002 (String)
            LLN0$QVVR_Event
        moreFollows (Boolean)
          False
```

```
            00 01 02 03 04 05 06 07
00000000    A1 49 02 01 19 A1 44 A0
00000008    3F 1A 15 4C 4C 4E 30 24
00000010    4D 48 41 49 5F 4D 65 61
00000018    73 75 72 65 6D 65 6E 74
00000020    1A 15 4C 4C 4E 30 24 4D
00000028    4D 58 55 5F 4D 65 61 73
00000030    75 72 65 6D 65 6E 74 1A
00000038    0F 4C 4C 4E 30 24 51 56
00000040                      5 6E 74
00000048
```

Response PDU

Figure 7-20: GetNameList PDU for Data Sets

Once the list of data sets has been obtained using the MMS GetNameList service, a GetNamedVariableListAttribute service request is sent by IED Explorer; this MMS service is mapped to the ACSI GetDataSetDirectory service. The GetDataSetDirectory service returns the contents of the data set in question. Figure 7-21 shows an extract of the GetNamedVariableListAttribute request and response PDUs as well as their contents. In this example, a request for the contents of the MHAI_Measurement dataset is made, and the response includes the data attributes of the three complex phase A, B and C magnitude measurements.



Packet Nr.	Time	Dir	MMS Pdu	MMS Service	Size
50	10.09.2020 07:52:54.069	>>	Confirmed_ResponsePDU	GetNameList	75
51	10.09.2020 07:52:54.092	<<	Confirmed_RequestPDU	GetNamedVariableListAttribu...	42
52	10.09.2020 07:52:54.101	>>	Confirmed_ResponsePDU	GetNamedVariableListAttribu...	306

```
MMSpdu (Choice)
  confirmed-RequestPDU (Sequence)
    invokeID (Integer)
      26
    service (Choice)
      getNamedVariableListAttributes (Choice)
        domain-specific (Sequence)
          domainID (String)
            RasPQMET
          itemID (String)
            LLN0$MHAI_Measurement
```

```
          00 01 02 03 04 05 06 07
00000000  A0 28 02 01 1A AC 23 A1
00000008  21 1A 08 52 61 73 50 51
00000010  4D 45 54 1A 15 4C 4C 4E
00000018  30 24 4D 48 41 49 5F 4D
00000020  65 61 73 75 72 65 6D 65
00000028  6E 74
```

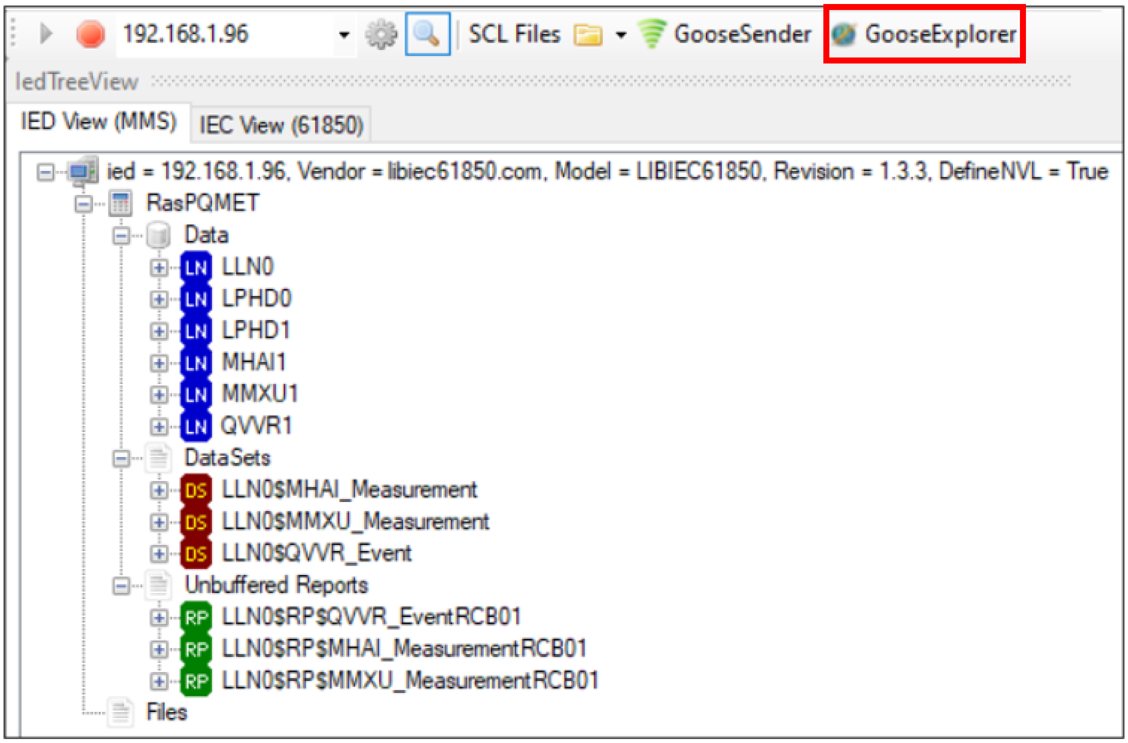
Packet Nr.	Time	Dir	MMS Pdu	MMS Service	Size
51	10.09.2020 07:52:54.092	<<	Confirmed_RequestPDU	GetNamedVariableListAttribu...	42
52	10.09.2020 07:52:54.101	>>	Confirmed_ResponsePDU	GetNamedVariableListAttribu...	306
53	10.09.2020 07:52:54.114	<<	Confirmed_RequestPDU	GetNamedVariableListAttribu...	42
54	10.09.2020 07:52:57.627	>>	Confirmed_ResponsePDU	GetNamedVariableListAttribu...	863

```
MMSpdu (Choice)
  confirmed-ResponsePDU (Sequence)
    invokeID (Integer)
      26
    service (Choice)
      getNamedVariableListAttributes (Sequence)
        mmsDeletable (Boolean)
          False
        listOfVariable (SequenceOf)
          unnamedSequence000 (Sequence)
            variableSpecification (Choice)
              name (Choice)
                domain-specific (Sequence)
                  domainID (String)
                    RasPQMET
                  itemID (String)
                    MHAI1$MX$ThdA$phsA$cVal$mag$f
```

```
          00 01 02 03 04 05
00000000  A1 82 01 2E 02 01
00000008  82 01 27 80 01 00
00000010  01 20 30 2D A0 2B
00000018  1A 08 52 61 73 50
00000020  45 54 1A 1D 4D 48
00000028  31 24 4D 58 24 54
00000030  41 24 70 68 73 41
00000038  56 61 6C 24 6D 61
00000040  66 30 2D A0 2B A1
0                  73 50 51
0                  4D 48 41
0                  24 54 68
00000060  24 70 68 73 42 24
00000068  61 6C 24 6D 61 67
00000070  30 2D A0 2B A1 29
00000078  52 61 73 50 51 4D
00000080  1A 1D 4D 48 41 49
00000088  4D 58 24 54 68 64
00000090  70 68 73 43 24 63
```

Figure 7-21: GetNamedVariableListAttribute PDU returning data set items

The GetNamedVariableListAttribute request is sent for each of the datasets available on the IED, along with the responses from the MMS GetVariableAccessAttributes and read service requests, an overview of an existing IED's configuration and structure can be built within the IED Explorer software. Figure 7-22 shows a snapshot of the IED structure once all of the required MMS responses have been received from the PQM device. All logical node instances, data sets and reports configured on the device are reported along with their contents.

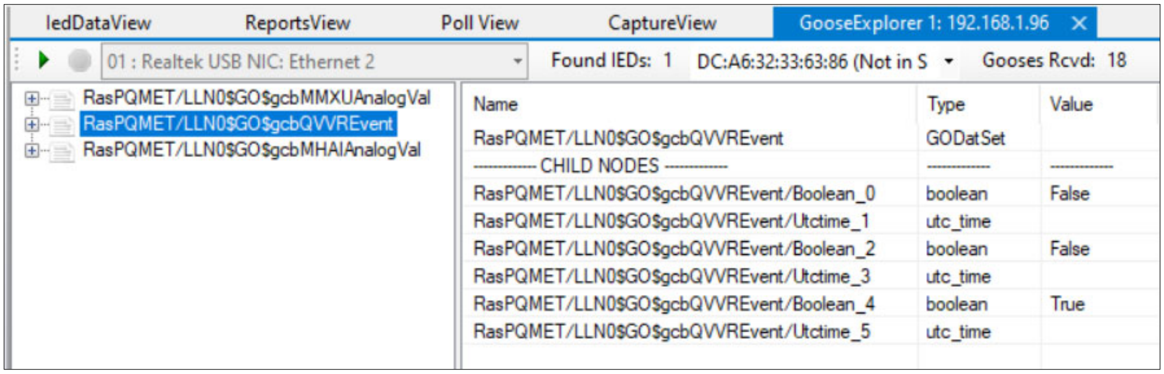


```
ied = 192.168.1.96, Vendor = libiec61850.com, Model = LIBIEC61850, Revision = 1.3.3, DefineNVL = True
  RasPQMET
    Data
      LN LLN0
      LN LPHD0
      LN LPHD1
      LN MHAI1
      LN MMXU1
      LN QVVR1
    DataSets
      DS LLN0$MHAI_Measurement
      DS LLN0$MMXU_Measurement
      DS LLN0$QVVR_Event
    Unbuffered Reports
      RP LLN0$RP$QVVR_EventRCB01
      RP LLN0$RP$MHAI_MeasurementRCB01
      RP LLN0$RP$MMXU_MeasurementRCB01
  Files
```

Figure 7-22: IED View Confirming the Details of the PQM Device

Finally, GOOSE publishing may be confirmed by only capturing the published GOOSE messages on IED Explorer, and this may be done by selecting “GooseExplorer” on the toolbar (highlighted in red in Figure 7.22). Figure 7-23 shows the Goose messages being published on the Ethernet 2 interface. All discovered Goose control blocks are listed on the left, whilst the data types and current data value of each data type found in the child nodes of the Goose control block are shown in the capture window on the right-hand side.

In Figure 7-23, The number of IEDs found on the interface, the PQM device's MAC address and the number of Goose messages received are also presented in the information bar above the captured data. The information bar is useful for confirming where the GOOSE messages are originating. Confirmation may be done by cross-referencing the MAC address using the address resolution protocol "arp -a" command in the command line. The "Goose Rcvd" statistic (on the top-right of Figure 7-18) provides a live update of the number of Goose messages currently published.



01 : Realtek USB NIC: Ethernet 2 Found IEDs: 1 DC:A6:32:33:63:86 (Not in S Gooses Rcvd: 18

- RasPQMET/LLN0$GO$gcbMMXUAnalogVal
- RasPQMET/LLN0$GO$gcbQVVREvent
- RasPQMET/LLN0$GO$gcbMHAIAnalogVal

Name	Type	Value
RasPQMET/LLN0$GO$gcbQVVREvent	GODatSet	
------------ CHILD NODES ------------	------------	------------
RasPQMET/LLN0$GO$gcbQVVREvent/Boolean_0	boolean	False
RasPQMET/LLN0$GO$gcbQVVREvent/Utctime_1	utc_time	
RasPQMET/LLN0$GO$gcbQVVREvent/Boolean_2	boolean	False
RasPQMET/LLN0$GO$gcbQVVREvent/Utctime_3	utc_time	
RasPQMET/LLN0$GO$gcbQVVREvent/Boolean_4	boolean	True
RasPQMET/LLN0$GO$gcbQVVREvent/Utctime_5	utc_time	

Figure 7-23: GooseExplorer capture window showing published Goose Messages

In conclusion, this section has proven that the PQM device's functions have been successfully mapped and conform with MMS and ACSI services. First, it was established that the device was discoverable at an MMS level. That basic ISO 9506 services could be recognised and responded to by the device, and finally that MMS services mapped to ACSI services such as GeAllDataValues, GetDataDirectory and GetDataSetDirectory can be requested and responded to appropriately.

Further conformance tests may be performed according to IEC 61850-10. Still, for this implementation, it is not necessary, as further testing of ACSI services would only prove useful for IEC 61850 compliance and verification testing. A complete compliance and verification test is only be required if the device is intended for commercial sale, which is not an objective of this research project. Proving basic/partial compliance with the necessary MMS and ACSI services is sufficient to test interoperability with fully compliant vendor manufactured devices. The LibIEC61850 software library supports all of the essential ACSI services required for full IEC 61850 conformance. A complete list of the services supported by the library can be found at Zillgith (2014).

7.7. Conclusion

In this chapter, several tests are performed to verify the various functional parts of the device, namely the measurement accuracy and the verification IEC 61850-based communications procedures. First, the accuracy of the general measurement quantities such as voltage and current were assessed by injecting signals using the Omicron CMC and verifying them using a reference meter. Then the power quality measurement accuracy and detection were tested by injecting the appropriate power quality signals using available Omicron Test Universe test modules. Finally, IEC 61850 MMS and ACSI Service mapping and conformance are verified using the IED Explorer software.

Overall, the developed power quality device is found to be highly accurate with a low level of deviation from the Fluke meter readings over a range of input values for both voltage and current. Harmonic, Frequency and power measurements were verified by injecting specifically selected values from the Test Universe Software but could not be compared with an existing meter due to the lack of availability of a power quality meter for reference readings. The device is successfully discovered and modelled on the IED Explorer software, thus proving conformance with vital MMS functions and ACSI mappings required for IEC 61850-based communications. GOOSE messaging is also verified and identified by the software.

With the various functions of the device verified, its feasibility and interoperability level as an IED may be tested by implementing a simple SCADA system with a vendor-built, IEC 61850 compatible remote terminal unit (RTU) or automation controller. Chapter 8 explores the procedure of testing interoperability in a practical setting. A widely available industrial-grade automation controller is used to simulate a typical substation setting containing devices from other vendors. Practical applications of the implemented device are also briefly described, with an emphasis on distributed generation systems and IEC 61580 modelling.

CHAPTER EIGHT

INTEROPERABILITY TEST AND PRACTICAL APPLICATIONS OVERVIEW

8.1. Introduction

Interoperability is a vital aspect of the IEC 61850 standard, the ability for all devices compliant with the standard to communicate with each other is an essential feature of the standard since its inception. Mguzulwa (2018) performed an investigation into the interoperability of various IED from different vendors and their specific interpretation of the IEC 61850 standard. It was found that several items/situations such as GOOSE Message data format, differences in engineering tools, IED name length differences, GOOSE identifies, configuration revision and differences in SCL file structures could impede interoperability.

This chapter aims to demonstrate interoperability and the use of the device in a real-world scenario as part of substation automation and SCADA system. In the previous chapter, the effectiveness and accuracy of the measurement algorithms, and the conformance of the device with ACSI and MMS services, were proven. The device may now be used and evaluated in a scenario close to real-world implementation. Figure 8-1 shows the modules assessed in this Chapter.

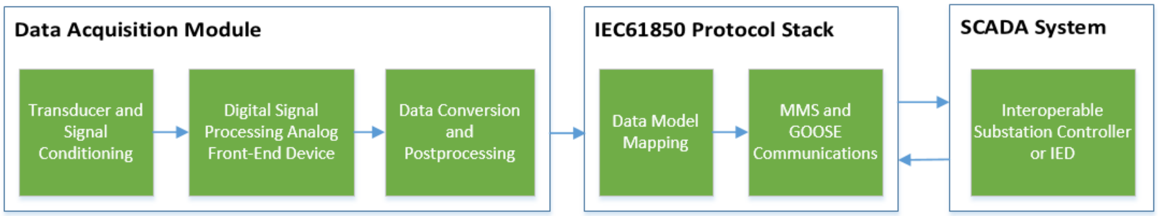

Data Acquisition Module
Transducer and Signal Conditioning
Digital Signal Processing Analog Front-End Device
Data Conversion and Postprocessing
IEC61850 Protocol Stack
Data Model Mapping
MMS and GOOSE Communications
SCADA System
Interoperable Substation Controller or IED

Figure 8-1: Modules Assessed in Chapter 8

Proving the PQM device's usefulness in real-world scenarios required interoperability tests with various vendor devices. A procedure has been designed to simulate the implementation of a substation automation or supervisory control and data acquisition (SCADA) system that includes the PQM device in an IEC 61850-based environment along with an IEC 61850 compliant commercially available RTU (Remote Terminal Unit). In this experiment, interoperability is tested with specifically the Schweitzer Engineering Laboratories Real-Time Automation Controller (SEL RTAC). Testing other devices may require a modified test procedure depending on the engineering software. Once interoperability is proven, the suitability of the developed device in a network with

Distributed Energy Resources (DERs) and some real-world practical applications are explored.

8.2. Interoperability and SCADA functionality demonstration

The interoperability and SCADA functionality demonstration focuses on the practical aspects of the device's operation as an IEC 61850-based data acquisition device. The interoperability of the device with a vendor-built remote terminal unit or automation controller, and its functionality within a SCADA environment, is tested.

An SEL RTAC (Real-Time Automation Controller) 3555 is selected as the substation controller for this test. SEL uses the AcSELerator Architect software to configure the IEC61850-based communications on the RTAC along with a wide range IEDs. The software also allows for the importation of IEDs from other vendors using the relevant SCL file, which provides SEL devices with the ability to interact with dataSets originating from devices from other vendors in the form MMS client inputs and GOOSE subscriptions. Then finally, an SCD file may be created and sent to the RTAC using the AcSELerator RTAC software.

The interoperability test aims to verify whether the ICD file used to create the static IED model on the Raspberry Pi can be imported and configured using SEL's software platform using the same methods as with any other third-party vendor IED.

8.2.1. Procedure overview

The implementation procedure features several prerequisite tests that must be passed to move onto the next one. All the tests listed in order are as follows:

1) Importing custom ICD file into the SEL AcSELerator Architect software IED Palette
2) RTAC Goose Subscription and MMS Client configuration
3) Generate and import A valid SCD file from the Architect Project file to AcSELerator RTAC software.
4) Human-Machine Interface (HMI) display creation and configuration
5) Signals are injected using the CMC 356 and the Quick CMC Test Module
6) Observe MMS and GOOSE network traffic on Wireshark and confirm acquired data updates on HMI diagram.

Each of the abovementioned tests shall be executed in the order as shown above to accurately simulate the procedure of configuring an external vendor IED within the SEL software environment.

8.2.2. Required hardware and software

The hardware and software required for the interoperability and SCADA communications tests mainly comprises of products belonging to the assessed vendor's range of software products, along with the implemented system and its components:

Hardware

- Raspberry Pi
- ADE9000 data acquisition board
- SEL RTAC 3555
- Windows PC
- Omicron CMC 356

Software

- SEL-5032 AcSELerator Architect
- SEL-5033 AcSELerator RTAC
- SEL-5055 AcSELerator Diagram Builder
- VNC Viewer and FTP Client
- Raspberry Pi Data Acquisition Software
- Wireshark
- Omicron Test Universe 4.00
    - Quick CMC Test Module

8.2.3. Import custom ICD File to IED Palette

The IEC61850 configuration process of a specific SEL device begins by defining the IEDs and automation controllers present in the substation's IEC 61850-based communications network. SEL-5032 AcSELerator Architect software is the primary software package used for IEC61580 configuration; it serves the primary purpose of generating and exporting SCL files within the SEL software ecosystem.

The software also allows for the importation of ICD files originating from non-SEL devices as long as they are written according to the SCL file standard stipulated in IEC 61850-6. Having the ability to import user-generated SCL files allows SEL devices to communicate with IEC61850 compliant devices from other vendors. As such, the first test to determine interoperability with SEL devices is the ability to import the written ICD file into the AcSELerator Architect software's IED database.

Figure 8-2 shows the Architect software's IED palette, and this window presents the devices available for use in an IEC61850-based project, represented by SEL's signature blue faceplate. A new IED may be imported by right-clicking and selecting the "Import IED" button in the dialogue box. Once the Windows explorer window appears, a valid ICD file may be selected.

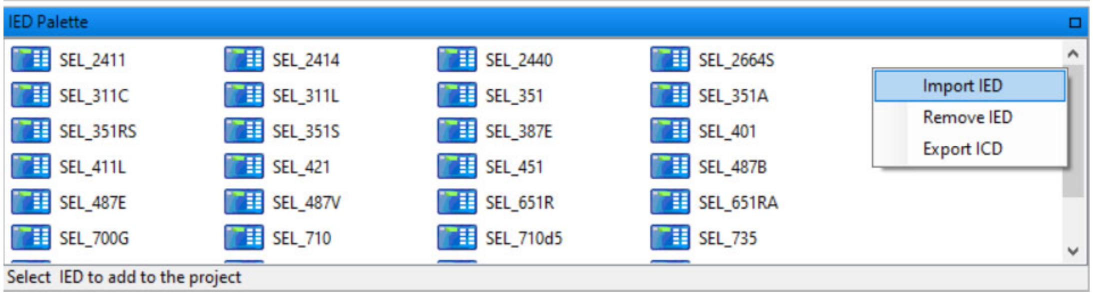


Figure 8-2: Importation of IED into AcSELerator Architect's IED Palette

Once the IED has been imported, it appears as a green faceplate, indicating it is a manually imported/non-SEL device. Figure 8-3 shows two different versions of the PQMonitor.icd file, which is the ICD file for the PQM device. These devices may be used in any Architect project, with SEL devices able to subscribe to Goose Messages and create client inputs for MMS servers.

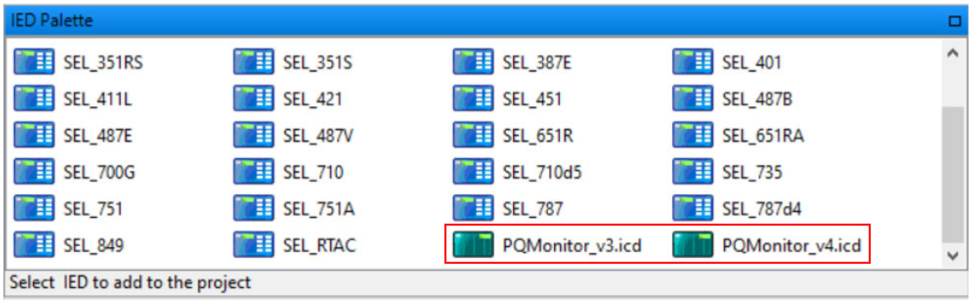
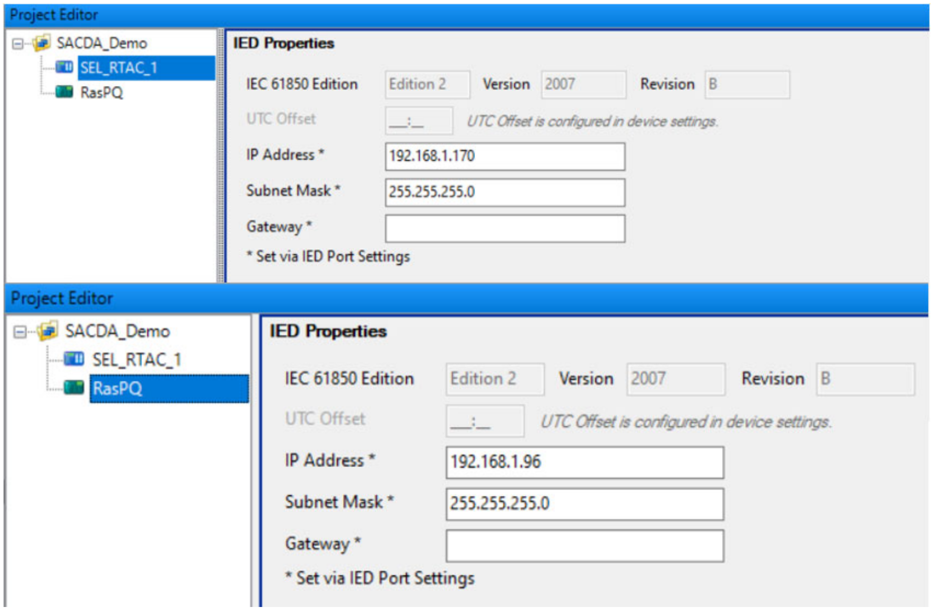


Figure 8-3: Manually Imported IED's within the IED Palette

With the device imported into the IED palette, it may now be used in an IEC61850 project. At this point, the device may be added similarly other IEDs in the IED palette, in this case, PQMonitor_v4.icd is added to the project, with the SEL RTAC used as the other device in the test.

Figure 8-4 shows the overview of the project, confirming that the PQM device may be imported into AcSELerator Architects' IED palette and used in an IEC61580 project. It should be noted that the PQM device's IED name (RasPQ) is imported directly from the SCL file's "IEDName" field as defined in the communications – connectedAP section of the ICD file. The IP address of 192.168.1.96 is also imported directly from the information contained in the ICD file (See Figure 5-13 for more detail), thus confirming that the IED information is successfully loaded to the SEL AcSELerator Architect Project Editor.



IED Properties

IEC 61850 Edition: Edition 2 — Version: 2007 — Revision: B
UTC Offset: __:__ *UTC Offset is configured in device settings.*
IP Address *: 192.168.1.170
Subnet Mask *: 255.255.255.0
Gateway *:
* Set via IED Port Settings

IED Properties

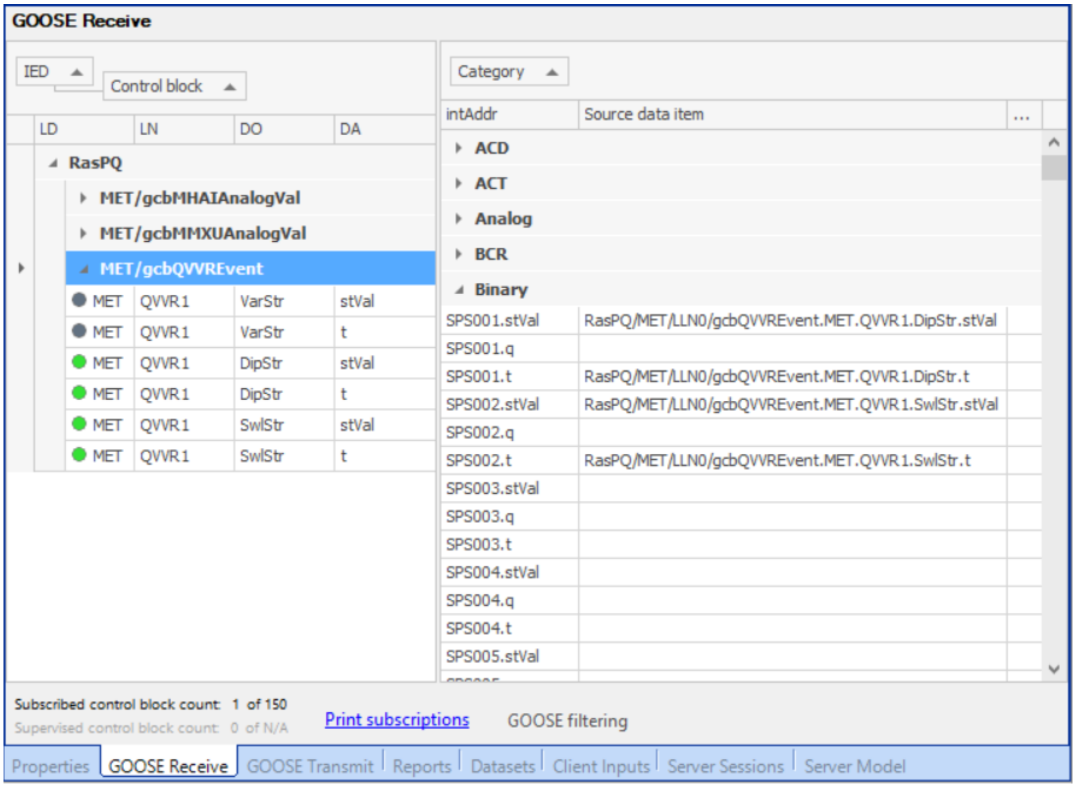
IEC 61850 Edition: Edition 2 — Version: 2007 — Revision: B
UTC Offset: __:__ *UTC Offset is configured in device settings.*
IP Address *: 192.168.1.96
Subnet Mask *: 255.255.255.0
Gateway *:
* Set via IED Port Settings

Figure 8-4: IEC61850 Project Overview and IED Properties

8.2.4. GOOSE Subscription and MMS Client

Importing the IED and creating a project including the SEL RTAC 3555 allows for the subscription of GOOSE messages and the creation Client inputs by the substation mentioned above controller. Selecting the RTAC and navigating to the relevant configuration tabs allows for the mappings of relevant IEC61850 functions to the selected device, in this case, the RTAC 3550 maps inputs from the RasPQ device to binary and analogue inputs. This test is designed to demonstrate that the GOOSE control blocks and MMS dataSets configurated within PQMonitor_v4.icd are available for mapping to the RTAC configuration.

Figure 8-5 shows the contents of the "GOOSE Receive" tab, on the left-hand window, GOOSE control blocks available in the project and their relevant data attributes are listed. It can be observed that the GOOSE control blocks shown are only those available in the ICD file implemented in Chapter 5 and subsequently imported for this test. The right window shows the binary inputs that are mapped to the status value data attributes in QVVR1. For verification purposes, the QVVR1 dip and swell data attributes for both status value and time are mapped to binary input SPS001 and SPS002 in the RTAC, respectively.



GOOSE Receive

IED ▲ Control block ▲

LD	LN	DO	DA
◢ RasPQ			
▸ MET/gcbMHAIAnalogVal			
▸ MET/gcbMMXUAnalogVal			
◢ MET/gcbQVVREvent			
● MET	QVVR1	VarStr	stVal
● MET	QVVR1	VarStr	t
● MET	QVVR1	DipStr	stVal
● MET	QVVR1	DipStr	t
● MET	QVVR1	SwlStr	stVal
● MET	QVVR1	SwlStr	t

Category ▲

intAddr	Source data item
▸ ACD	
▸ ACT	
▸ Analog	
▸ BCR	
◢ Binary	
SPS001.stVal	RasPQ/MET/LLN0/gcbQVVREvent.MET.QVVR1.DipStr.stVal
SPS001.q	
SPS001.t	RasPQ/MET/LLN0/gcbQVVREvent.MET.QVVR1.DipStr.t
SPS002.stVal	RasPQ/MET/LLN0/gcbQVVREvent.MET.QVVR1.SwlStr.stVal
SPS002.q	
SPS002.t	RasPQ/MET/LLN0/gcbQVVREvent.MET.QVVR1.SwlStr.t
SPS003.stVal	
SPS003.q	
SPS003.t	
SPS004.stVal	
SPS004.q	
SPS004.t	
SPS005.stVal	

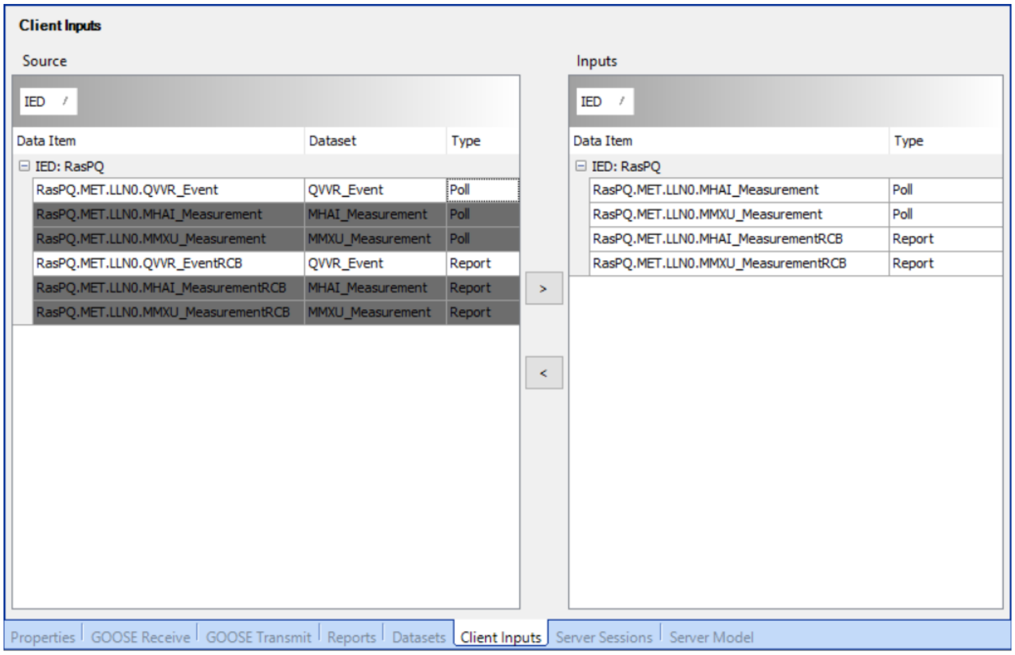
Subscribed control block count: 1 of 150
Supervised control block count: 0 of N/A

Print subscriptions GOOSE filtering

Figure 8-5: Subscribing to GOOSE Control Block QVVREvent

Next, the client inputs may be configured to poll for the relevant datasets and reports. The client inputs require a source originating from another IED within the project to create the necessary inputs in the RTAC successfully.

Figure 8-6 shows the client inputs chosen from source datasets available on the PQM device's ICD file. On the left-hand window is a list of available data items (sources), for each data item, the data Item name, dataset name and dataset types are all presented. Three dataset types are available in the IED configurations, for demonstration purposes, only the datasets "MHAI_Measurement" and "MMXU_Measurement" have client inputs. To preserve bandwidth, a report may be used instead of a polled dataset. Reports are periodically sent by the MMS server (unsolicited communications) instead of waiting for a request PDU from the client as in the case of datasets (solicited communications). To demonstrate both communication methods, two SCD files are generated, one including polled datasets and another with reports. Once selected, the client inputs appear in the list on the right side of Figure 8-6 (Inputs).



Client Inputs

Source

IED: RasPQ

Data Item	Dataset	Type
RasPQ.MET.LLN0.QVVR_Event	QVVR_Event	Poll
RasPQ.MET.LLN0.MHAI_Measurement	MHAI_Measurement	Poll
RasPQ.MET.LLN0.MMXU_Measurement	MMXU_Measurement	Poll
RasPQ.MET.LLN0.QVVR_EventRCB	QVVR_Event	Report
RasPQ.MET.LLN0.MHAI_MeasurementRCB	MHAI_Measurement	Report
RasPQ.MET.LLN0.MMXU_MeasurementRCB	MMXU_Measurement	Report

Inputs

IED: RasPQ

Data Item	Type
RasPQ.MET.LLN0.MHAI_Measurement	Poll
RasPQ.MET.LLN0.MMXU_Measurement	Poll
RasPQ.MET.LLN0.MHAI_MeasurementRCB	Report
RasPQ.MET.LLN0.MMXU_MeasurementRCB	Report

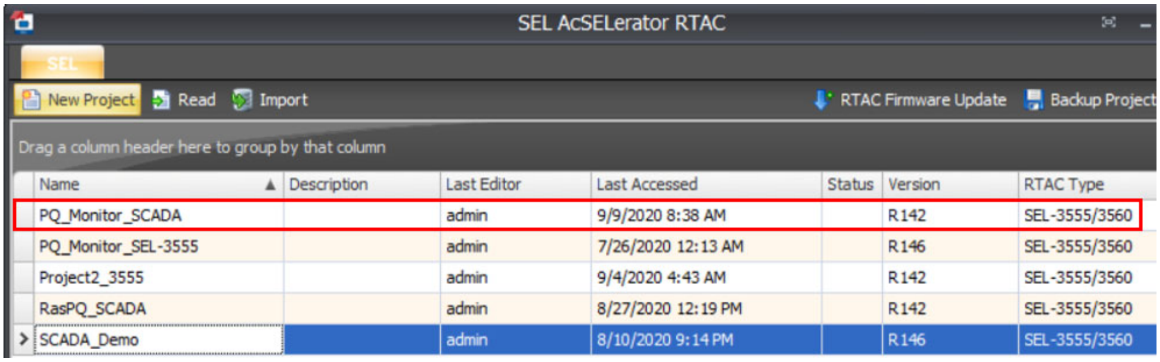
Figure 8-6: Client Source and Inputs

With GOOSE subscription and client input definition successfully implemented, an SCD (Substation Configuration Description) file may be created using the RTAC project. As previously stated in Chapter 4, the SCD file contains information regarding the IEDs, communications setup and the overall IEC 61850 substation topology; the SCL file can then be uploaded to the RTAC using the AcSELerator RTAC software. On AcSELerator Architect, the SCD file is generated by navigating to file – save as… and selecting the file type as “Architect project” which is saved as an SCD file. Two SCD files are generated for the demonstration of the two data acquisition methods; “SCADA_Demo_R135” including reports and “SCADA_Demo_No_Reports” which is polled datasets only.

8.2.5. Import SCD file and upload configuration

Importing the SCD files and uploading the configuration to the RTAC is the final step of the IEC 61850 configuration procedure. Importation and uploading are performed using the AcSELerator RTAC software. Importing the SCD file requires an RTAC project to be created in the RTAC Database, either by creating a new configuration file or reading an existing configuration from an RTAC device and modifying it. In this case, reading and changing the device’s existing configuration is preferred as it lessens the engineering requirements for other parts of the configuration file. The configuration is read on the RTAC software’s database screen, once it has been read, a new configuration file appears as part of the RTAC database, in this case, the downloaded configuration is renamed to

“PQ_Monitor_SCADA”. Figure 8-7 shows the project window for AcSELerator RTAC, the project renamed from the default configuration is highlighted in red.



Name	Description	Last Editor	Last Accessed	Status	Version	RTAC Type
PQ_Monitor_SCADA		admin	9/9/2020 8:38 AM		R142	SEL-3555/3560
PQ_Monitor_SEL-3555		admin	7/26/2020 12:13 AM		R146	SEL-3555/3560
Project2_3555		admin	9/4/2020 4:43 AM		R142	SEL-3555/3560
RasPQ_SCADA		admin	8/27/2020 12:19 PM		R142	SEL-3555/3560
SCADA_Demo		admin	8/10/2020 9:14 PM		R146	SEL-3555/3560

Figure 8-7: RTAC Project Window

Opening the “PQ_Monitor_SCADA” configuration file allows for the insertion of an IEC 61850 configuration, and this is done by navigating to the toolbar and selecting “Insert” as shown in Figure 8-8, under insert, selecting IEC 61850 and “Set IEC 61850 Configuration” in the drop-down menu allows for the importation of the SCD file. Once the dialogue box is opened, the “SCADA_Demo_R135.SCD” file report based data acquisition generated by AcSELerator Architect may be selected.

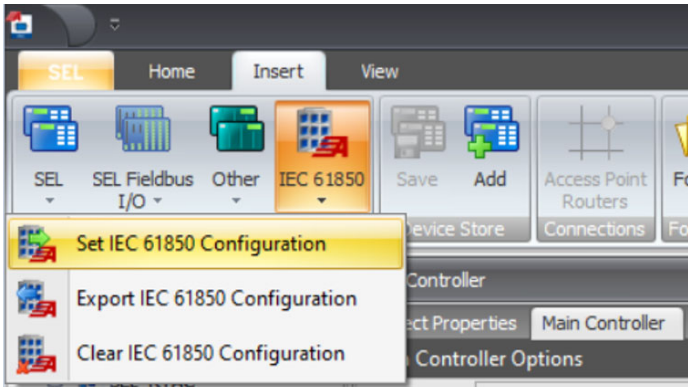


Figure 8-8: Setting the IEC 61850 Configuration Using the Generated SCD file

When the SCD file is successfully imported, this results in the appearance of two additional green selectable menu items in the Project window, as shown in Figure 8-9. These two items represent the MMS and the GOOSE portions of the IEC 61850 configuration, with “RasPQ_850” representing the MMS Client and ‘RasPQ_MET_gcbQVVREvent_GRX” representing the GOOSE control block.

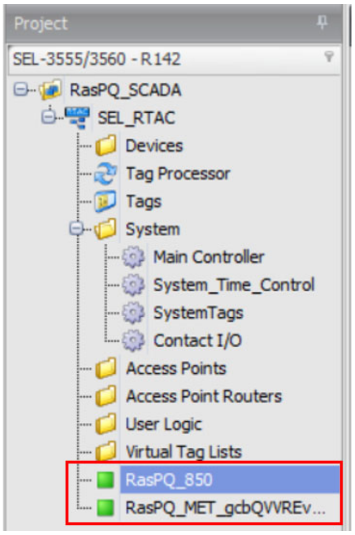
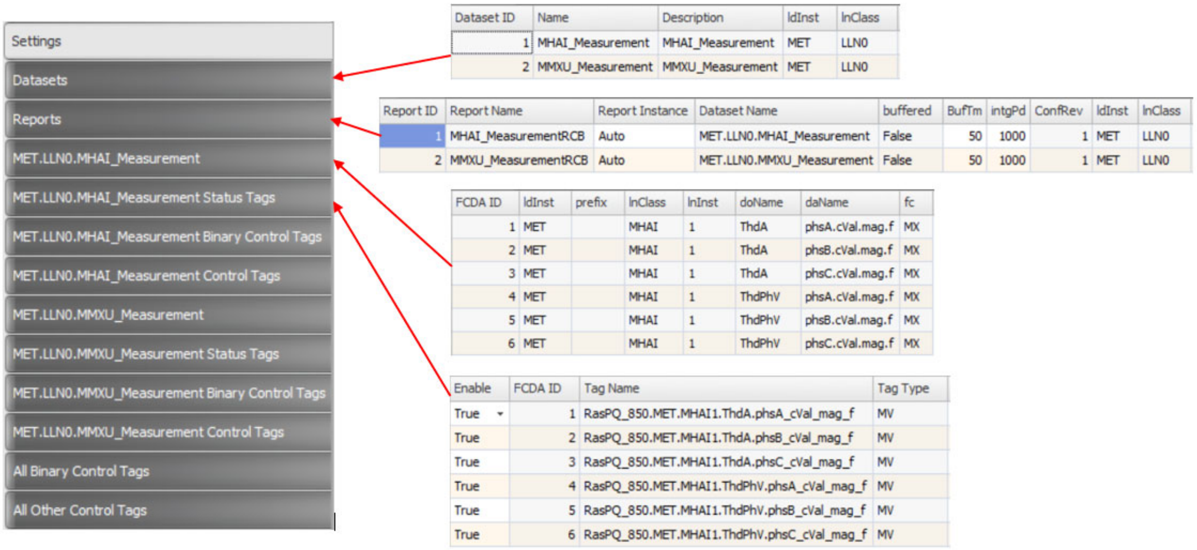


Figure 8-9: IEC 61850 Configurations Imported int Project

The contents of RasPQ_850 may be selected using the project sidebar. The full configuration of the MMS Client is shown in Figure 8-10 with the following is details:

- **Datasets** showing “MMXU_Measurement” and “MHAI_Measurement” datasets.
- **Reports:** Only populated in “SCADA_Demo_R135.SCD” which includes reports, which in this case are MHAI_MeasurementRCB and MMXU_MeasurementRCB.
- **MET.LLN0.MHAI_Measurement:** Contains information about the logical nodes’ data objects, data attributes and functional constraints within MHAI_Measurements
- **MET.LLN0.MHAI_Measurement Status Tags:** The RTAC tags associated with the data attributes within the MHAI_Measurements dataset.


Settings
Datasets
Reports
MET.LLN0.MHAI_Measurement
MET.LLN0.MHAI_Measurement Status Tags
MET.LLN0.MHAI_Measurement Binary Control Tags
MET.LLN0.MHAI_Measurement Control Tags
MET.LLN0.MMXU_Measurement
MET.LLN0.MMXU_Measurement Status Tags
MET.LLN0.MMXU_Measurement Binary Control Tags
MET.LLN0.MMXU_Measurement Control Tags
All Binary Control Tags
All Other Control Tags

Dataset ID	Name	Description	ldInst	lnClass
1	MHAI_Measurement	MHAI_Measurement	MET	LLN0
2	MMXU_Measurement	MMXU_Measurement	MET	LLN0

Report ID	Report Name	Report Instance	Dataset Name	buffered	BufTm	intgPd	ConfRev	ldInst	lnClass
1	MHAI_MeasurementRCB	Auto	MET.LLN0.MHAI_Measurement	False	50	1000	1	MET	LLN0
2	MMXU_MeasurementRCB	Auto	MET.LLN0.MMXU_Measurement	False	50	1000	1	MET	LLN0

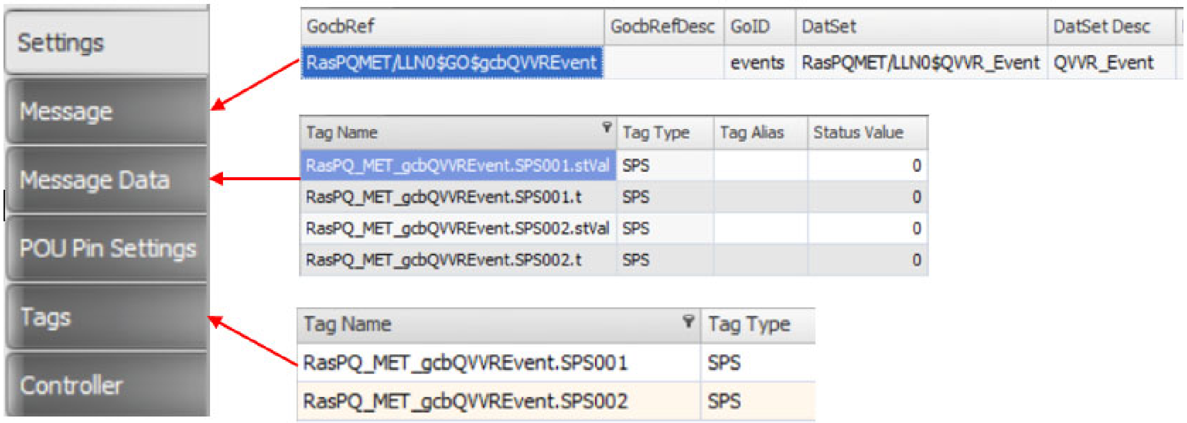
FCDA ID	ldInst	prefix	lnClass	lnInst	doName	daName	fc
1	MET		MHAI	1	ThdA	phsA.cVal.mag.f	MX
2	MET		MHAI	1	ThdA	phsB.cVal.mag.f	MX
3	MET		MHAI	1	ThdA	phsC.cVal.mag.f	MX
4	MET		MHAI	1	ThdPhV	phsA.cVal.mag.f	MX
5	MET		MHAI	1	ThdPhV	phsB.cVal.mag.f	MX
6	MET		MHAI	1	ThdPhV	phsC.cVal.mag.f	MX

Enable	FCDA ID	Tag Name	Tag Type
True	1	RasPQ_850.MET.MHAI1.ThdA.phsA_cVal_mag_f	MV
True	2	RasPQ_850.MET.MHAI1.ThdA.phsB_cVal_mag_f	MV
True	3	RasPQ_850.MET.MHAI1.ThdA.phsC_cVal_mag_f	MV
True	4	RasPQ_850.MET.MHAI1.ThdPhV.phsA_cVal_mag_f	MV
True	5	RasPQ_850.MET.MHAI1.ThdPhV.phsB_cVal_mag_f	MV
True	6	RasPQ_850.MET.MHAI1.ThdPhV.phsC_cVal_mag_f	MV

Figure 8-10: Contents of RasPQ_850

Similarly, the contents of RasPQ_MET_gcbQVVREvent_GRX can be seen when selected, as presented in Figure 8-11. Here the settings, message details, message data, controller logic associated with the GOOSE message and the tags assigned by acSELerator Architect may be verified. The essential items include:

- **Message:** which presents the GOOSE control block reference and data set,
- **Message Data:** contains information relating to the data values contained within the GOOSE message
- **Tags:** displays the RTAC tags associated with the relevant data values.



GocbRef	GocbRefDesc	GoID	DatSet	DatSet Desc
RasPQMET/LLN0$GO$gcbQVVREvent		events	RasPQMET/LLN0$QVVR_Event	QVVR_Event

Tag Name	Tag Type	Tag Alias	Status Value
RasPQ_MET_gcbQVVREvent.SPS001.stVal	SPS		0
RasPQ_MET_gcbQVVREvent.SPS001.t	SPS		0
RasPQ_MET_gcbQVVREvent.SPS002.stVal	SPS		0
RasPQ_MET_gcbQVVREvent.SPS002.t	SPS		0

Tag Name	Tag Type
RasPQ_MET_gcbQVVREvent.SPS001	SPS
RasPQ_MET_gcbQVVREvent.SPS002	SPS

Figure 8-11: Contents of RasPQ_MET_gcbQVVREvent_GRX

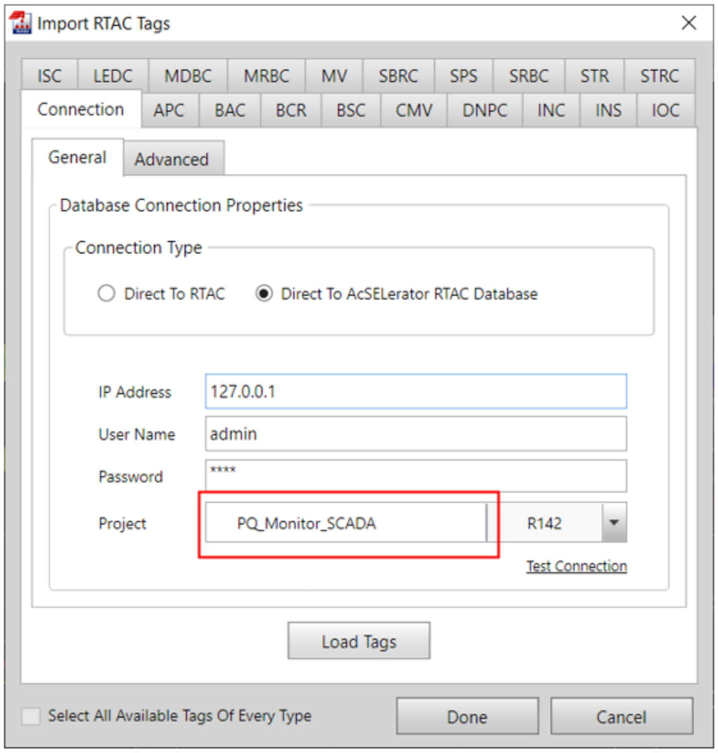
The SCD file configuration can be verified by ensuring all required datasets, reports and goose control blocks are present and mapped to RTAC tags, and these specified menu items are shown in Figure 8-10 and 8-11. Successful verification of the configuration, according to the SCD file generated in AcSELerator Architect, confirms that the power quality monitoring device can be successfully and fully configured to communicate with the RTAC device using SEL engineering software.

At this point, the written configuration file may be sent to the RTAC to test the MMS Client and GOOSE subscription, and this is achieved by navigating to “SEL” on the toolbar and selecting “Send”. The dialogue box for uploading the generated RTAC configuration file appears once clicked, the configuration may then be uploaded by logging into the connected RTAC and sending the file. Once the configuration is uploaded, the software configuration of the RTAC is completed. The second SCD configuration file may be tested by selecting the “Clear IEC 61850 Configuration” as shown in Figure 8-7 to clear the current configuration, then clicking the “Set IEC 61850 Configuration” option in the same dialogue and selecting “SCADA_Demo_No_Reports.SCD”.

8.2.6. Human Machine Interface display creation

To elegantly visualise the data and more accurately simulate a functioning SCADA system, an HMI diagram is designed and made functional using the RTAC tags from the IEC 61850 configuration; the HMI is built using the AcSELerator Diagram Builder software. The Diagram Builder enables users to add a diverse number of controls such as gauges, visual representations of substation equipment and measurand trends.

Once the Diagram Builder software is opened, the RTAC tags may be imported by selecting the “Import” button on the toolbar. The dialogue box shown in Figure 8-12 appears when “Import” is selected; in this case, the configuration file was obtained directly from the RTAC database. With the project created earlier (PQ_Monitor_SCADA) selected for tag importation once the correct database is selected, clicking the “Load Tags” button loads all of the tags configured in the RTAC configuration file.



○ Direct To RTAC ● Direct To AcSELerator RTAC Database

IP Address: 127.0.0.1

User Name: admin

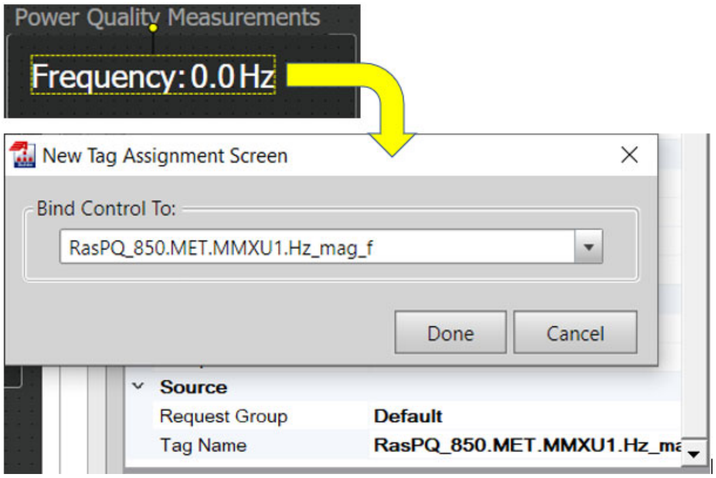
Password: ****

Project: PQ_Monitor_SCADA R142

□ Select All Available Tags Of Every Type

Figure 8-12: RTAC Tag Import Dialogue

With the tags loaded, an HMI display may be built using available controls to represent the IEC 61850 data. Each control included in the HMI display is associated with an appropriate RTAC tag. Figure 8-13 shows an example of the tag assignment. The frequency status label is associated with the RTAC tag “RasPQ_850.MET.MMXU1.Hz_mag_f”, which represents the magnitude data attribute (measured as a floating-point value) for the frequency data object within MMXU. All data objects present on the HMI display are connected to their respective controls in this manner.



Frequency: 0.0 Hz

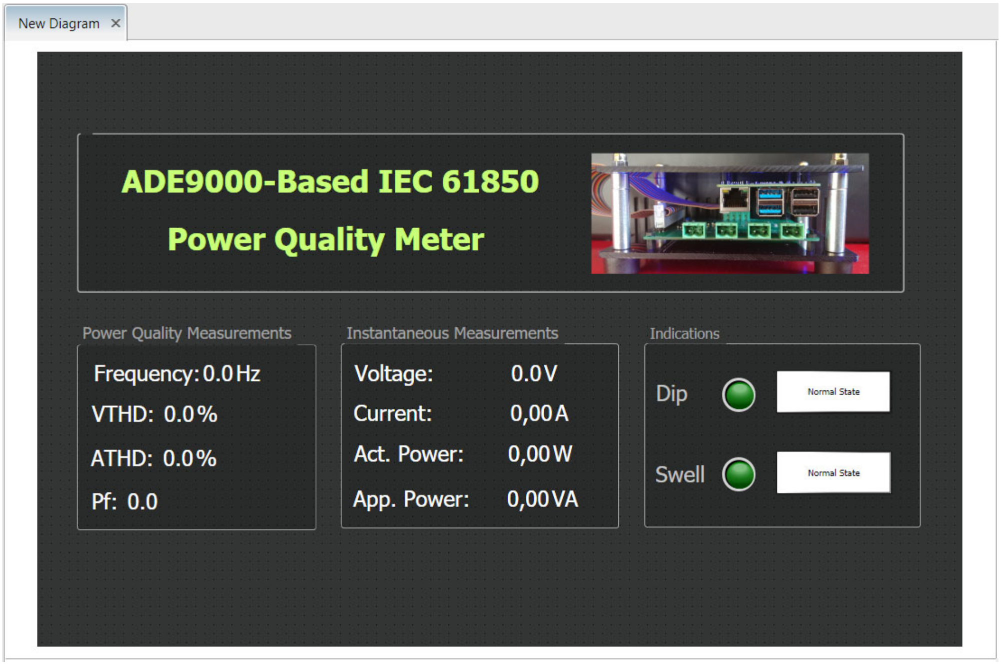
Bind Control To: RasPQ_850.MET.MMXU1.Hz_mag_f

Source

Request Group	Default
Tag Name	RasPQ_850.MET.MMXU1.Hz_ma

Figure 8-13: Frequency Measurement Associated with an RTAC Tag

Figure 8-14 shows the final layout of the HMI as designed using the AcSELrator RTAC Diagram Builder software that includes all controls needed to display the received data. The measured values for frequency, THD for voltage, THD for current and power factors are grouped under the Power Quality Measurements group, while voltage, current, active power and apparent power are in a separate group. The dip and swell indications are displayed using input LEDs and status labels.



Frequency: 0.0 Hz
VTHD: 0.0%
ATHD: 0.0%
Pf: 0.0

Voltage:	0.0 V
Current:	0,00 A
Act. Power:	0,00 W
App. Power:	0,00 VA

Dip	●	Normal State
Swell	●	Normal State

Figure 8-14: AcSELerator Diagram Builder Constructed HMI Display

Finally, the HMI display may be loaded by navigating to “File” on the toolbar and selecting “Send Current Project”. The presented dialogue box can be used to connect to the RTAC device and send the created HMI display. Successfully loading the HMI confirms that the tags created from the IEC 61850 configuration can be used to visually display the data transmitted by the Power Quality Monitoring device.

8.2.7. Data transmission monitoring and SCADA demo

The final test involves confirming the functionality of the IEC 61850 client and GOOSE Receive configured on the SEL RTAC’s ethernet interface. The PQM device is considered a connected IED on the RTAC configuration; connected IEDs may be viewed using the RTAC webpage. The RTAC webpage may be accessed by entering the RTAC’s configured IP address into a chosen web browsers address bar; in this case, 192.168.1.170 is entered. Pressing the enter button opens the login screen, entering the administrator username and password grants access to the RTAC’s configuration parameters via the webpage. The RTAC webpage consists of several links, Figure 8-15 shows the links under the “Reports” and “HMI” categories, for this test, the “Connected IED” and “SCADA_HMI” links (highlighted in red) are the only required page links.

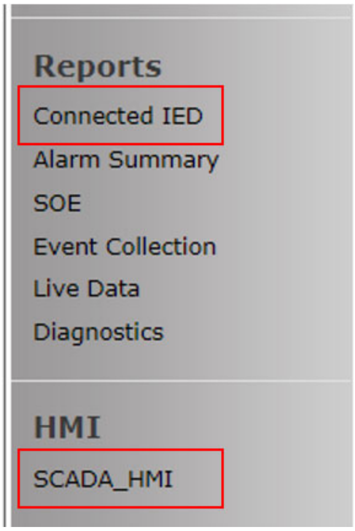
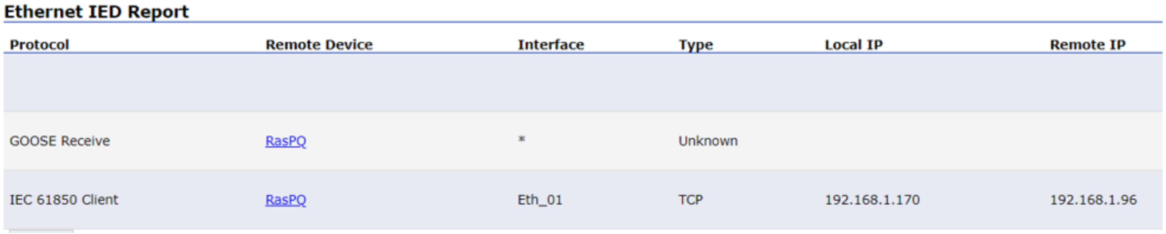


Figure 8-15: Reports and HMI RTAC webpage links

First, selecting “Connected IED” generates a report of the configured IEDs based on the interface chosen on the user interface, selecting “Eth_01” shows the IED’s connected to that specific interface as shown in Figure 8-16. In Figure 8-16, the two methods of receiving data from the PQM device are shown as two separate communications protocols, GOOSE Receive and IEC 61850 Client. The “Remote Device” column indicates the name of the

IED, in this case, “RasPQ” (The IED name as specified in the .ICD file) indicates that the two protocols originate from the PQM device. Along with the device name, the Interface name, type, Local IP and Remote IP are indicated in the table where applicable. The information presented in this table confirms the presence of the IEDs connected to the specified interface.

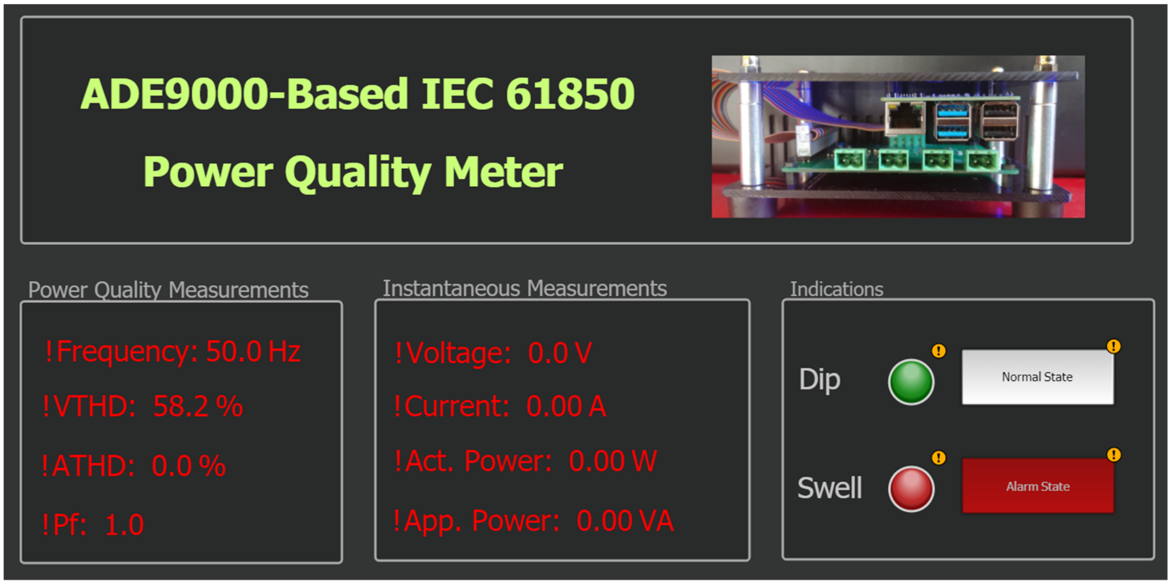


Ethernet IED Report

Protocol	Remote Device	Interface	Type	Local IP	Remote IP
GOOSE Receive	RasPQ	*	Unknown		
IEC 61850 Client	RasPQ	Eth_01	TCP	192.168.1.170	192.168.1.96

Figure 8-16: IEC 61850 IED Report

With the IED interfaces confirmed via the webpage interface, data acquisition may be initialised by injecting voltage and current values as described in measurement accuracy-test methodology (Section 7.3.3). Once the PQM device is connected to the network, and signal injection started using the CMC, the SCADA HMI may be opened using the selection, as shown in Figure 8-15. Once the HMI is opened, the HMI configured using the SEL Diagram Builder software is presented in the webpage interface. Figure 8-17 shows the HMI diagram with values that have not been updated; exclamation marks represent these values; at this point, the MMS server on the Raspberry Pi has not been started.



ADE9000-Based IEC 61850

Power Quality Meter

Power Quality Measurements

!Frequency: 50.0 Hz

!VTHD: 58.2 %

!ATHD: 0.0 %

!Pf: 1.0

Instantaneous Measurements

!Voltage: 0.0 V

!Current: 0.00 A

!Act. Power: 0.00 W

!App. Power: 0.00 VA

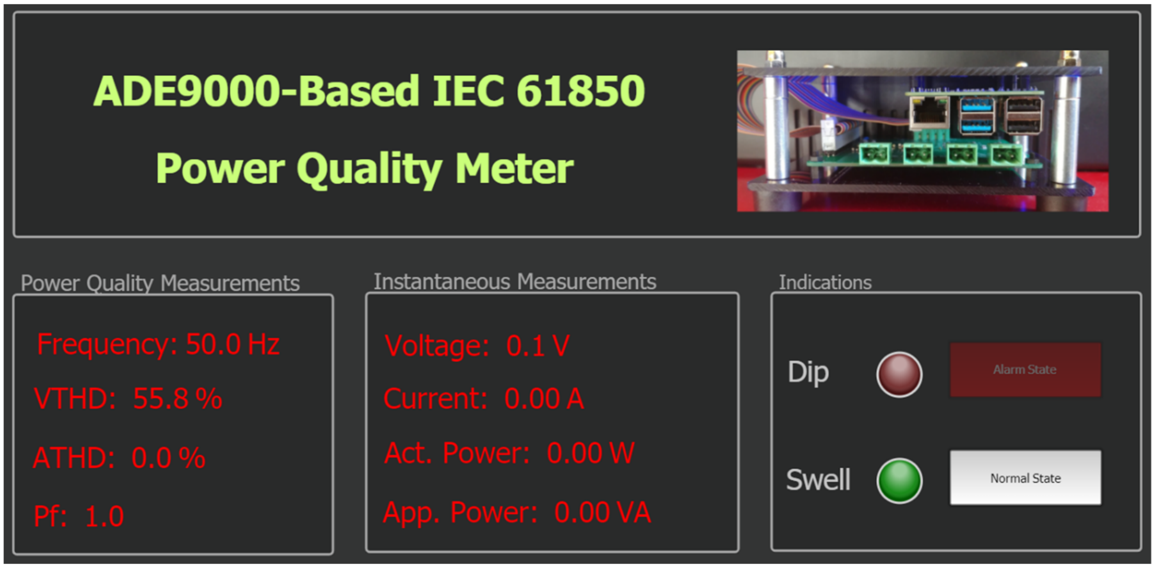
Indications

Dip ● Normal State

Swell ● Alarm State

Figure 8-17: HMI Display with no updated values

Starting the MMS server as detailed in Section 7.33 opens a connection between the MMS client and server as well as beginning the GOOSE publishing. Figure 8-18 shows valid data acquired from the PQM device via the MMS Client and the GOOSE subscription. Still, no current and voltage injection is occurring, as such, all power quality and instantaneous measurements are either zero or base values as is the case with the Frequency and Power Factor. The Voltage THD records a valid 56.4% due to the 0.1V floating voltage value from the CMC output terminal.



ADE9000-Based IEC 61850

Power Quality Meter

Power Quality Measurements

Frequency: 50.0 Hz

VTHD: 55.8 %

ATHD: 0.0 %

Pf: 1.0

Instantaneous Measurements

Voltage: 0.1 V

Current: 0.00 A

Act. Power: 0.00 W

App. Power: 0.00 VA

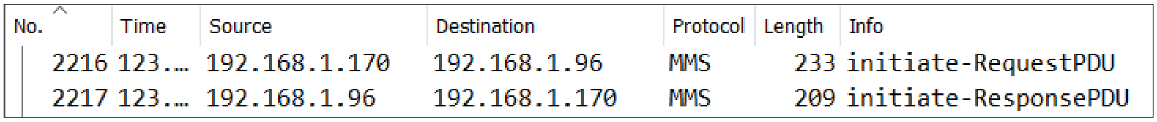
Indications

Dip ● Alarm State

Swell ● Normal State

Figure 8-18: The SCADA HMI with MMS Client Connected and GOOSE Publishing Enabled but no Current or Voltage Injected

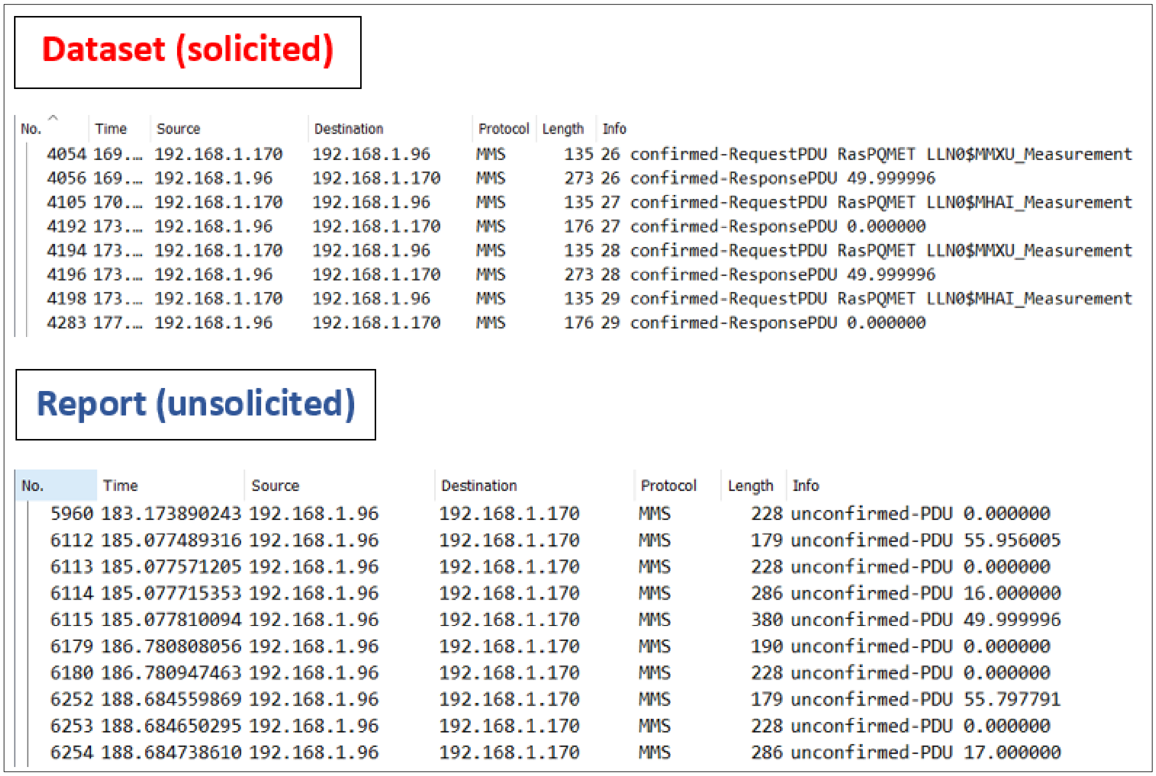
Communications may be monitored by starting a Wireshark capture before starting the MMS server on the Raspberry Pi; the resulting capture may be used to confirm the data acquisition method as specified in the SCD file. Figure 8-19 shows a Wireshark capture of the mandatory initiate MMS service as described in section 7.6; in this case, the SEL RTAC initiates communications with the Raspberry Pi as the MMS client. Being a mandatory service, the “initiate-RequestPDU” is present in MMS transactions of both data acquisition methods.



```
No.    Time     Source          Destination     Protocol  Length  Info
  2216 123.…  192.168.1.170   192.168.1.96    MMS          233  initiate-RequestPDU
  2217 123.…  192.168.1.96    192.168.1.170   MMS          209  initiate-ResponsePDU
```

Figure 8-19: Initiation MMS service sent by the RTAC, response from the Raspberry Pi

Once initialisation between the client and server is complete, the MMS transactions differ based on the data acquisition method. Figure 8-20 shows a screenshot of the MMS requests and responses for the polled dataset configuration as well as report or unsolicited report communications method present in the second SCD configuration file. In polled dataset configuration, the MMS client (IP address 192.168.1.170) requests datasets “MMXU_Measurement” and “MHAI_Measurement” from the MMS server (IP address 192.168.1.96). The report configuration transmits the datasets without the need of request PDUs from the MMS client, and reports are sent periodically from the server without the need for client requests or confirmations.



Dataset (solicited)

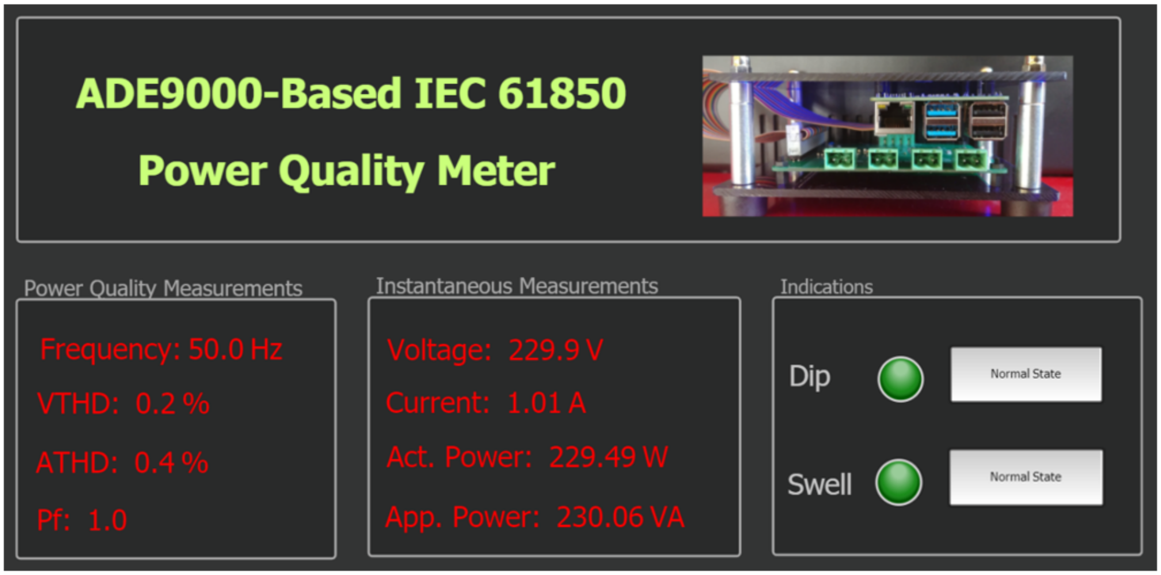
```
No.   Time     Source          Destination     Protocol Length Info
4054 169.… 192.168.1.170  192.168.1.96   MMS    135 26 confirmed-RequestPDU RasPQMET LLN0$MMXU_Measurement
4056 169.… 192.168.1.96   192.168.1.170  MMS    273 26 confirmed-ResponsePDU 49.999996
4105 170.… 192.168.1.170  192.168.1.96   MMS    135 27 confirmed-RequestPDU RasPQMET LLN0$MHAI_Measurement
4192 173.… 192.168.1.96   192.168.1.170  MMS    176 27 confirmed-ResponsePDU 0.000000
4194 173.… 192.168.1.170  192.168.1.96   MMS    135 28 confirmed-RequestPDU RasPQMET LLN0$MMXU_Measurement
4196 173.… 192.168.1.96   192.168.1.170  MMS    273 28 confirmed-ResponsePDU 49.999996
4198 173.… 192.168.1.170  192.168.1.96   MMS    135 29 confirmed-RequestPDU RasPQMET LLN0$MHAI_Measurement
4283 177.… 192.168.1.96   192.168.1.170  MMS    176 29 confirmed-ResponsePDU 0.000000
```

Report (unsolicited)

```
No.  Time           Source        Destination    Protocol Length Info
5960 183.173890243 192.168.1.96  192.168.1.170  MMS      228 unconfirmed-PDU 0.000000
6112 185.077489316 192.168.1.96  192.168.1.170  MMS      179 unconfirmed-PDU 55.956005
6113 185.077571205 192.168.1.96  192.168.1.170  MMS      228 unconfirmed-PDU 0.000000
6114 185.077715353 192.168.1.96  192.168.1.170  MMS      286 unconfirmed-PDU 16.000000
6115 185.077810094 192.168.1.96  192.168.1.170  MMS      380 unconfirmed-PDU 49.999996
6179 186.780808056 192.168.1.96  192.168.1.170  MMS      190 unconfirmed-PDU 0.000000
6180 186.780947463 192.168.1.96  192.168.1.170  MMS      228 unconfirmed-PDU 0.000000
6252 188.684559869 192.168.1.96  192.168.1.170  MMS      179 unconfirmed-PDU 55.797791
6253 188.684650295 192.168.1.96  192.168.1.170  MMS      228 unconfirmed-PDU 0.000000
6254 188.684738610 192.168.1.96  192.168.1.170  MMS      286 unconfirmed-PDU 17.000000
```

Figure 8-20: Solicited and Unsolicited Data Acquisition Methods

With communications between the client and server confirmed, analogue quantities may now be changed to verify the correctness of the acquired data and the timeous transmission of data values. Initiating injection from the CMC according to the procedure described in section 7.33 alters the received information as shown in Figure 8-21. In the example, 230 V and 1 A are being injected using the Quick CMC test module. The present voltage and current signals prompt the DSP engine on the ADE9000 to calculate the power quality and power measurements along with the power quality measurements. In typical SCADA systems, a dead band may be applied to analogue measurements to better manage limited bandwidth in systems with high numbers of analogue values, in this implementation, no dead band is used on the SEL RTAC.



ADE9000-Based IEC 61850
Power Quality Meter

Power Quality Measurements

Frequency: 50.0 Hz
VTHD: 0.2 %
ATHD: 0.4 %
Pf: 1.0

Instantaneous Measurements

Voltage: 229.9 V
Current: 1.01 A
Act. Power: 229.49 W
App. Power: 230.06 VA

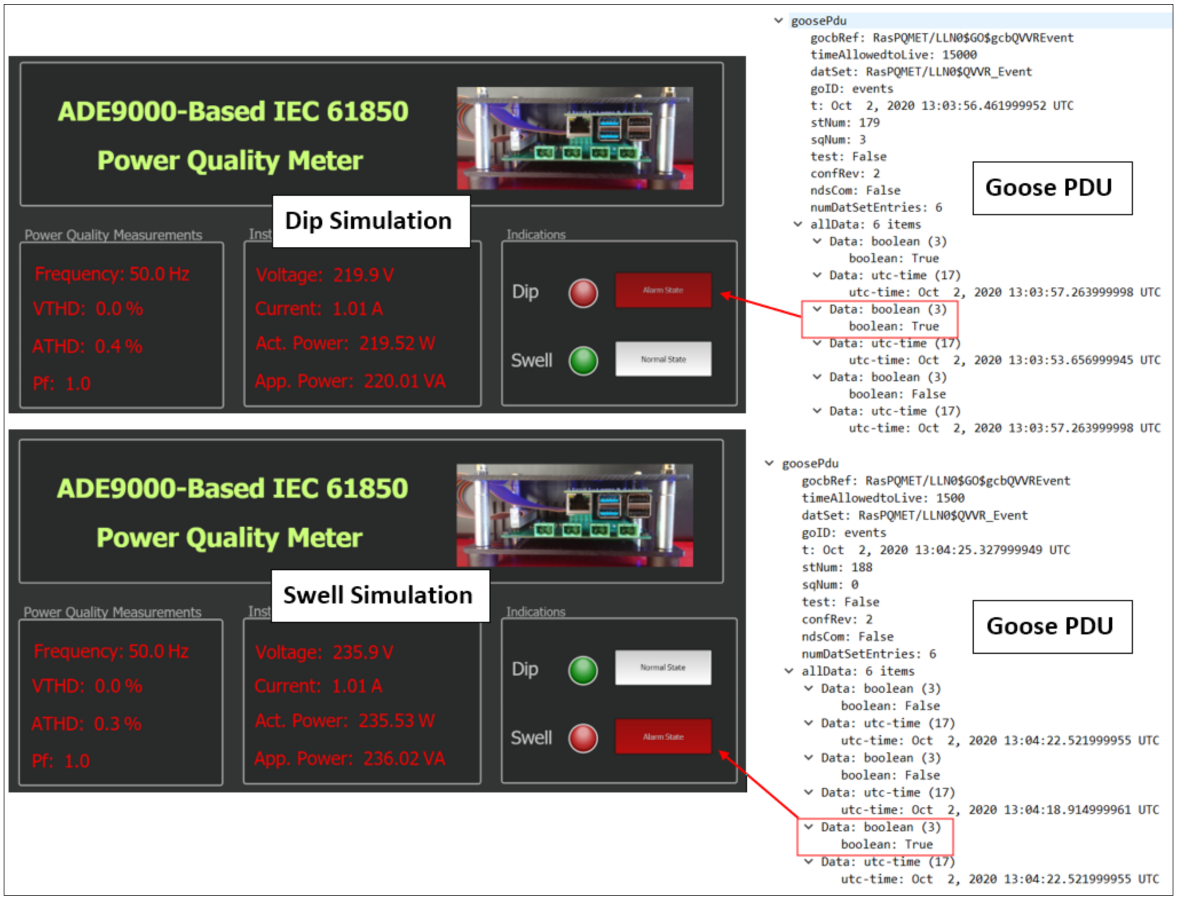
Indications

Dip ● Normal State
Swell ● Normal State

Figure 8-21: HMI Display with Live Data

Beside the MMS-based communications implemented for the analogue measurements, two GOOSE subscriptions for voltage variation events have been included in the RTAC's configuration, for voltage dip and swell respectively. GOOSE messages are typically used for protection and control functions that require fast response times. An example of how quick dip and swell detection could be used in a power system is by implementing a dynamic voltage restorer (DVR). Chaudray and Gangil (2016) demonstrate a system of power electronic devices that are used to quickly mitigate voltage dips within the system to restore the voltage to pre-fault levels. In this application, the usage of GOOSE messaging would be useful for sending the dip and swell level for comparison with the reference voltage and initiate control action.

Figure 8-22 shows the published GOOSE messages along with the states of the associated RTAC inputs. A simple algorithm is written in the server source code to prove the state changes of the published messages. The algorithm attaches the voltage level with the voltage dip and swell indications, with a voltage equal to or above 235 V triggering a "TRUE" boolean state on DipStr.StVal and "FALSE" boolean state SwlStr.StVal whilst voltage equal to or below 220 V triggers a "TRUE" on SwlStr.StVal and "FALSE" on DipStr.StVal. Values between 220 and 235V write a "FALSE" to both status value data attributes. In Figure 8-22 presents examples of both the dip and swell simulation conditions, with each condition's associated GOOSE PDU shown alongside the HMI diagram display.



Frequency: 50.0 Hz
VTHD: 0.0 %
ATHD: 0.4 %
Pf: 1.0

Voltage: 219.9 V
Current: 1.01 A
Act. Power: 219.52 W
App. Power: 220.01 VA

```
goosePdu
    gocbRef: RasPQMET/LLN0$GO$gcbQVVREvent
    timeAllowedtoLive: 15000
    datSet: RasPQMET/LLN0$QVVR_Event
    goID: events
    t: Oct  2, 2020 13:03:56.461999952 UTC
    stNum: 179
    sqNum: 3
    test: False
    confRev: 2
    ndsCom: False
    numDatSetEntries: 6
  allData: 6 items
    Data: boolean (3)
        boolean: True
    Data: utc-time (17)
        utc-time: Oct  2, 2020 13:03:57.263999998 UTC
    Data: boolean (3)
        boolean: True
    Data: utc-time (17)
        utc-time: Oct  2, 2020 13:03:53.656999945 UTC
    Data: boolean (3)
        boolean: False
    Data: utc-time (17)
        utc-time: Oct  2, 2020 13:03:57.263999998 UTC
```

Frequency: 50.0 Hz
VTHD: 0.0 %
ATHD: 0.3 %
Pf: 1.0

Voltage: 235.9 V
Current: 1.01 A
Act. Power: 235.53 W
App. Power: 236.02 VA

```
goosePdu
    gocbRef: RasPQMET/LLN0$GO$gcbQVVREvent
    timeAllowedtoLive: 1500
    datSet: RasPQMET/LLN0$QVVR_Event
    goID: events
    t: Oct  2, 2020 13:04:25.327999949 UTC
    stNum: 188
    sqNum: 0
    test: False
    confRev: 2
    ndsCom: False
    numDatSetEntries: 6
  allData: 6 items
    Data: boolean (3)
        boolean: False
    Data: utc-time (17)
        utc-time: Oct  2, 2020 13:04:22.521999955 UTC
    Data: boolean (3)
        boolean: False
    Data: utc-time (17)
        utc-time: Oct  2, 2020 13:04:18.914999961 UTC
    Data: boolean (3)
        boolean: True
    Data: utc-time (17)
        utc-time: Oct  2, 2020 13:04:22.521999955 UTC
```

Figure 8-22: Published GOOSE Messages and their mapped inputs

8.2.8. Final Testbed overview

Finally, with the interoperability validation tests completed, the overall testbed used for the evaluation of the implemented device is as shown in Figure 8-23. The pictured testbed simulates a typical IEC 61850-based communications network and substation automation system. As a starting point, the testbed forms the based of a potentially more complex that may be implemented to test advanced applications on the implemented device. The testbed could be expanded to with additional IEDs, HMI display, substation controllers and metering equipment to perform different kinds of research.

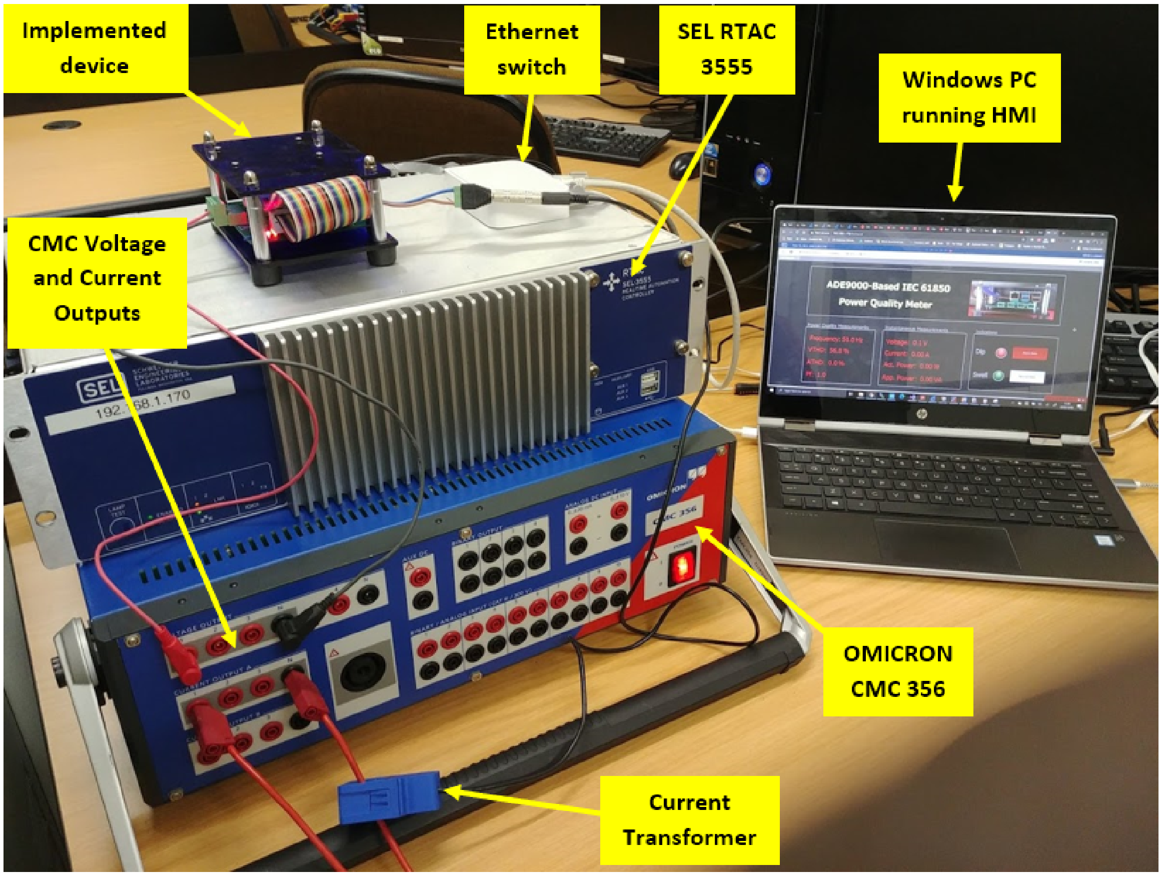


Figure 8-23: Test Bed for Evaluating the Implemented Device

8.3. Possible applications

Confirmation of IEC 61850 compliance and Interoperability with vendor devices opens the possibility of utilizing the PQM device in several diverse applications. Power Quality monitoring remains the focus of the device. Still, due to the versatility of the Raspberry Pi and energy metering functionality of the ADE9000, the device may also be used in additional applications.

This section briefly discusses some potential applications of the power quality monitoring device based on the findings in this and Chapter 7. Control and monitoring of DERs at utility-scale is the primary focus as that is the primary driver of the solution developed in this project. A few other possible applications are discussed as part of the analysis of the applications.

8.3.1. Utility-Scale control and monitoring

With the continuing interest and growing penetration of DERs within the electric grid, there is a growing need to monitor and control the various energy generation and storage plants. A 2017 report by the North American Electric Reliability Council (NERC) recognised that the shift from conventional synchronous to asynchronous source such as inverter-based systems necessitated the need for proper modelling of DERs (North American Electric Reliability Corporation, 2017). One of the findings within the NERC report was the usage of the IEC 61850 standard as a basis for the information model used for defining data exchange (North American Electric Reliability Corporation, 2017b).

In IEC 61850-7-420, logical nodes specifically developed for DERs are described, it is possible to model and implement the logical nodes on the PQM device. Along with the existing logical nodes from the power quality (Q) and metering (M) logical node classes, the functionality of the device could feasibly be expanded to control and monitor DER with an extensive utility-scale network.

Figure 8-23, sourced from the IEC 61850-7-420 page 11, shows an example of a communications network containing a diverse set of DERs. The problem that part 7-420 tries to solve is the unification of all DER controllers and meters into a set of logical devices and logical nodes to simplify the adoption and intake of DERs into the electric power grid.

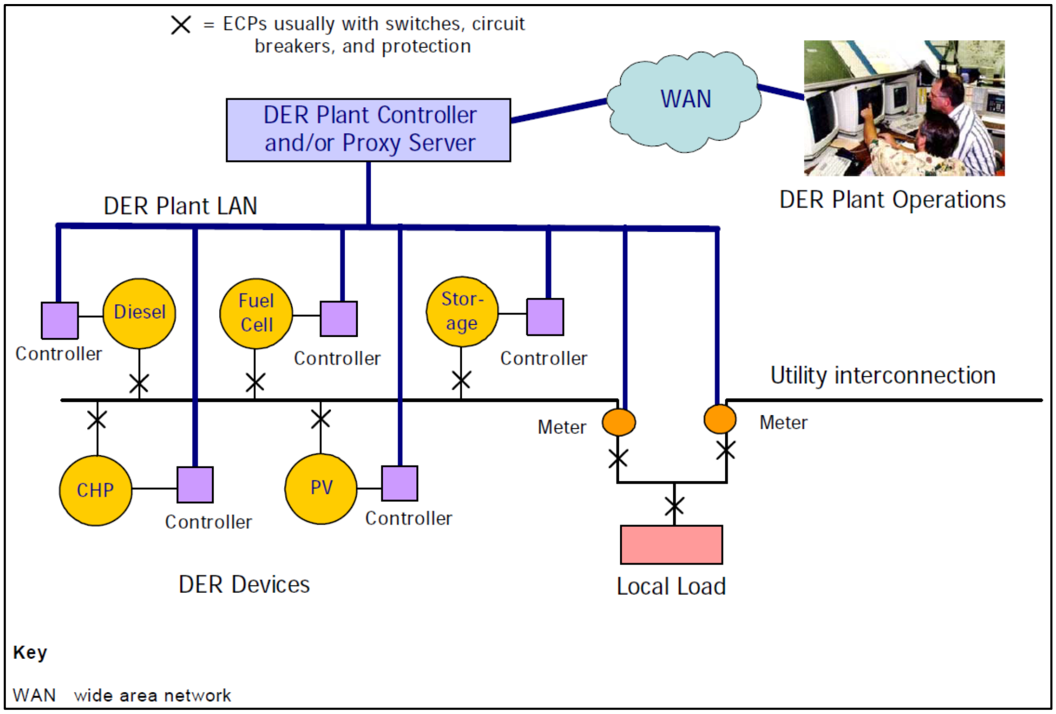


Figure 8-24: Example of a DERs Communications Network (IEC, 2008)

8.3.2. Integration of the PQM device within the network

In Figure 8-25, a detailed information model for a small-scale embedded generator or DER plant is presented. Using the logical nodes defined in IEC 61850-7-420, the components and equipment related to the monitoring and control of a photovoltaic plant (shown in the dashed blue rectangle) may be accurately modelled.

The several components that photovoltaic plants are comprised of such as the Maximum Power Point Tracking (MPPT) controller, inverter unit and battery storage bank can be modelled using IEC 61850-7-420 logical nodes. To interface with the photovoltaic plant devices, the RasPQ Meter can be modelled as a DER controller also according to the IEC 61850-7-420 part of the standard.

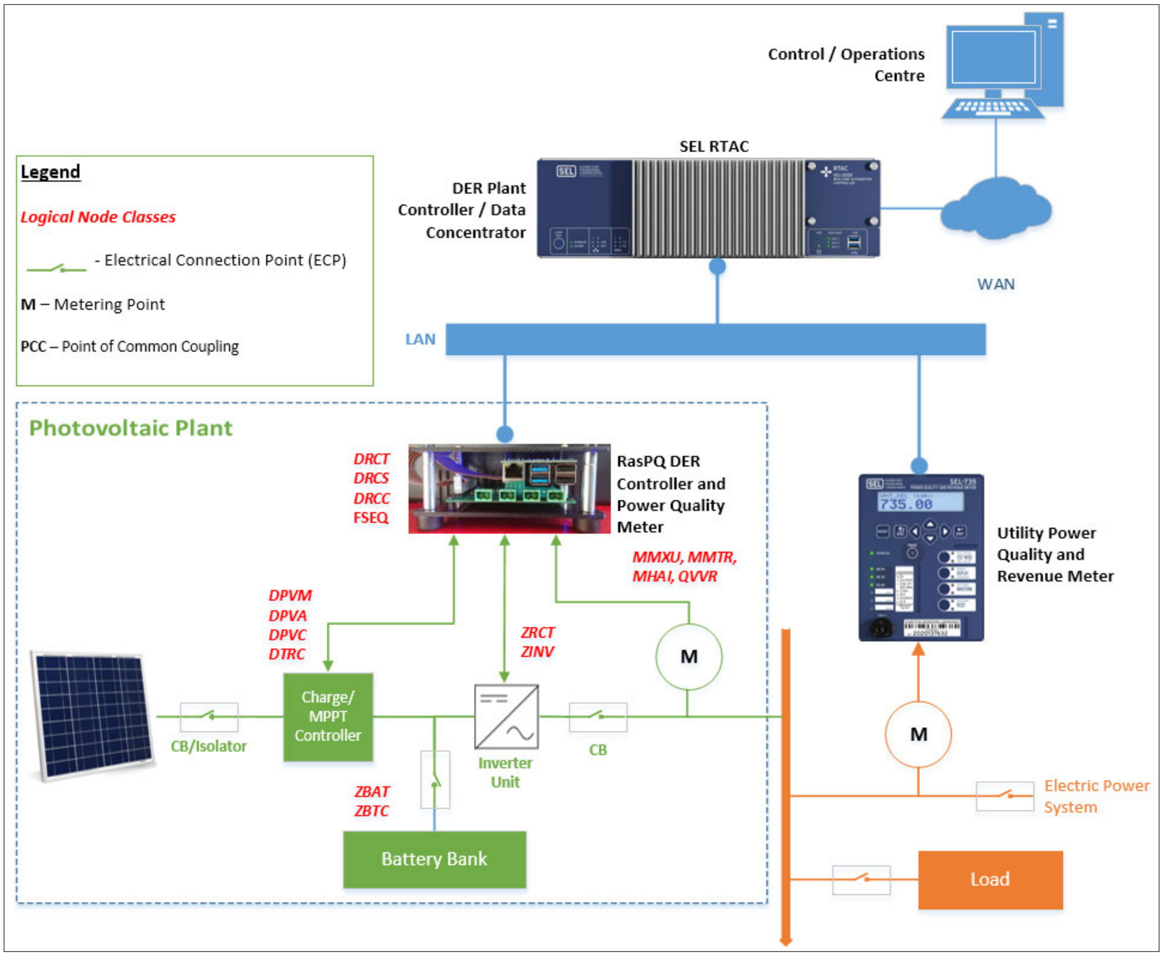


Figure 8-25: Photovoltaic plant monitored and controlled using the Implemented device

It is possible to model and generate a device model that may then be compiled and loaded onto the RasPQ Meter, essentially expanding the capabilities of the device beyond its primary purpose of power quality monitoring. Along with monitoring and controlling the

DER unit, the device may also be used for check metering alongside a utility revenue meter, as shown in Figure 8-25. Check metering is only possible due to the inclusion of kWh energy readings within the ADE9000's DSP computations. The data can then be transmitted to a substation controller such as the RTAC and then a central SCADA control system. A bidirectional energy meter may also be implemented, as demonstrated by Benitez *et al.* (2020). Whose prototype of a bidirectional meter explicitly designed for DER's utilises the ADE9000 as the data acquisition device. Bidirectional meters can monitor the energy being drawn from the system (Import), and well as power being fed back into the system (export), these types of meters are useful in areas with feed-in tariffs and DER's generating excess energy.

Functionally, the RasPQ is at its heart a Raspberry Pi single-board computer, meaning that theoretically, the PQM device could be configured as a meter, power quality monitor and communications gateway simultaneously with the appropriate software packages and interfacing hardware. The Raspberry Pi can be interfaced with the DER controller using MODBUS via a commercially available RS422/RS485 expansion board, IEC 60870-5-104 by using the open-source Lib60870-5 library or DNP3.0 using openDNP3 software. The acquired data may then be transmitted over a local area network (LAN) to a DER Plant Controller/data concentrator such as SEL RTAC using MMS or GOOSE depending on the application. The plant controller would transmit the data to a control centre over a Wide Area Network (WAN).

Using IEC 61850-7-420 and the flexibility afforded with the Raspberry Pi, DER's may be modelled based on the functionality and capability of the plant owner or power generator's DER controller. As long as the plant meets minimum specifications of the distribution network operator (DNO), the requirement for the plant owner to purchase a particular type of controller is negated. Configuration templates depending on the kind of DER and controller could be developed to simplify the adoption of IEC 61850 compliant devices. A key aspect to note is the requirement to harden the Raspberry Pi to make it suitable for industrial use cases and 24/7 usage. An alternative is to replace the computer board with one more suited for industrial applications such as a DIN-rail mounted Raspberry Pi Compute Module based design, Revolution Pi (RevPi), the Siemens Simatic IOT2020/2040 industrial gateway.

8.3.3. Additional applications

Besides the control and monitoring of DERs, several potential applications may also be made available for implementation by extending the functionality of the implemented device. Examples of additional functions may be the following:

- File transfers of metering and disturbance data via the implementation of the COMTRADE (Common format for Transient Data Exchange) or PQDIF (Power Quality Data Interchange Format) file formats.
- Power quality mitigation and compensation systems like photovoltaic ramp-rates and dynamic voltage restoration.
- Optimisation functions such as multi power point tracking (MPPT) control may also be monitored.

8.4. Conclusion

In this Chapter, the interoperability of the implemented device with SEL Real-Time Automation Controllers (RTAC) was proven; this was achieved by creating a rudimentary but realistic SCADA system with an RTAC 3550 device. Implementation was achieved by including the written ICD file in the AcSELerator Architect software's IED palette and generating an SCD file as part of a project including the SEL RTAC 3550. Then the SCD file was loaded into an AcSELerator RTAC project as an IEC 61850 configuration and uploaded to the RTAC device. This procedure demonstrated the simplicity at which the implemented device may be configured as an IED within an IEC 61850-based system as well as the compatibility of the Raspberry Pi's implemented logical nodes with an automation controller, this was a vital outcome of the research project and opened the door to various applications.

To theoretically demonstrate a possible application, a short case study was performed to determine the suitability of the device in a utility-scale electric power grid containing DERs and other IEC 61850-compliant devices. Coupled with the new logical nodes described in IEC 61850-7-420, it is found that the functionality of the device can be extended to far exceed the original scope of a power quality meter but also play the role of an energy meter and DER controller. To summarise, the device is found to be reliably capable of interacting with automation controllers built by SEL, thus proving interoperability with at least that vendor's devices.

Chapter 9 contains the deliverables as stated in Chapter 1, findings and forms conclusions as well as recommendations for future research.

CHAPTER NINE

DELIVERABLES, CONCLUSION AND RECOMMENDATIONS

9.1. Introduction

The continued introduction of distributed generation units into the electric grid has resulted in an ever-increasing number of points of supply. This increase along with the usage of power electronic converters for interfacing of solar, wind and open-cycle gas turbine plants, has created an environment where power quality monitoring becomes crucial. The bidirectional nature of DER's may also adversely affect the power quality of a given network, by injecting distorted or unbalanced signals into the grid, making it essential to both monitor and control the generation plant. Common power quality phenomena such as voltage dips, swells, and harmonics can potentially damage sensitive equipment and cause inefficiencies that could result in economic losses. It therefore becomes vital to develop a system for reliably monitoring DERs at the Point of Common Coupling to ensure adequate power quality is maintained throughout the system. Written when electric power grids were more centralised, power quality standards form part of regulatory frameworks in most countries. These frameworks still apply in the age of distribution generations. New technologies such as digital signal processors, low-cost single board computers and open-source communications libraries can be used to quicken the uptake of DERs and develop modern metering and monitoring infrastructure.

In this thesis, a literature review of existing power quality monitoring techniques is performed, then common power quality issues affecting DERs are identified. The identified power quality metrics are used to develop an information model based on the required logical nodes, and then an IED model is developed and implemented using LibIEC61850. Finally, the modelled IED is represented physically using the ADE9000's data register values and the SPI bus. Tests performed on the implemented device reveals a few potential real-world applications for the device, which is accurate, scalable, configurable, and interoperable with IEC 61850-compliant devices.

Section 9.2 states the aim of the research project, and section 9.3 defines the objectives. Section 9.4 briefly discusses the deliverables, results, and key findings of this research project. Potential applications are further discussed in 9.5 as well as potential future research projects based on the implemented device or similar. Section 9.6 presents an overview of the possible future work as a result of this research; section 9.7 lists the publications related to this project, and finally, section 9.8 is the conclusion to this chapter and the research project as a whole.

9.2. Aim:

This project aims to design and implement an IEC 61850-based power quality and DER monitoring system by developing new logical nodes or implementing existing ones on a selected hardware platform. This new monitoring system is aimed at providing an interoperable, cost-effective and scalable device to increase the deployment of power quality monitoring devices in a distributed environment to make the monitoring of DER supply points easier.

9.3. Objectives:

Maintaining a good quality of supply shall become more complicated in power systems with multiple points of supply, power electronic converters and bidirectional DER's. With this in mind, power quality monitoring needs to become more widespread and smarter. Deployment of IEC 61850-compliant devices at the DER's supply points allows for more robust monitoring by enabling them to interoperate with existing IEC 61850-based substation automation devices. To achieve the stated research project aim, the following objectives have been set out:

- To perform a thorough literature review of existing papers for power quality monitoring systems to determine the best practice, procedures and technologies for measurement and data transmission.
- To define the indices, algorithms and logical nodes necessary for monitoring the power quality of DERs at points of supply within the electric power grid.
- To select the appropriate hardware platform for the implementation of the software algorithms and the development of the IEC 61850 logical nodes.
- To implement and calibrate the power quality detection algorithms on the selected hardware platform/device.
- To implement IEC 61850 logical nodes on the selected hardware platform of choice using an open-source protocol stack.
- To establish, transmit and receive IEC 61850-based communications from devices in a substation automation environment.
- To test and evaluate the implemented logical nodes and power quality detection algorithms using test and simulation software and hardware components

- To prove interoperability by configuring the device to communicate with an industrial-grade IEC 61850 compliant IED or automation controller.

9.4. Deliverables

Power quality monitoring has evolved through the years, from a small number of expensive disturbance recorders requiring analysis post fault, to modern, connected power quality meters connected using advanced communications technologies. The advent of distributed generation and continued adoption of the IEC 61850 has made it necessary to develop a cost-effective, interoperable, scalable, and accurate solution. A device satisfying these requirements has been designed and implemented using the IEC 61850 standard, modern computing hardware and DSP technology. The deliverables are as follows:

9.4.1. Literature Review

A literature review was carried out on power quality monitoring, with an emphasis on the communications and data acquisition methods. A large group of relevant papers on power quality monitoring are identified, grouped, compared, and analysed. The initial group is whittled down to smaller groups for analysis. The evolution of the methods, communications protocols and system architectures were charted from the late 1980s up until the modern-day. Various papers were discussed, grouped and analysed to provide a clear picture of the current research landscape of power quality monitoring. The literature review was divided into three separate categories; general power quality monitoring, IEC 61850-based power quality monitoring and power quality monitoring using smart grid technologies.

The first category deals with general power quality monitoring topics, including data acquisition methods, digital signal processing and remote communications methods. The second category details the development of the IEC 61850 standard to accommodate for power quality monitoring as well as the modelling considerations when it comes to deploying IEC 61850-based system. Finally, the third category looks at the growing trend of “smartening” (extending the functionality and integrating with other applications within the electric grid) power quality monitoring and energy metering infrastructure according to to the smart grid framework.

Within the literature review for general power quality monitoring, it was found that the general developmental trend of power quality monitoring has been the use of digital signal processing.

Various communications technologies for remote metering, with Ethernet and wireless technologies, have also become prevalent communications methods of choice. The essential nature of performing long term studies to capture specific problem areas and trends is also seen as a critical driver in modern power quality monitoring.

Next, the role of the IEC 61850 standard in modelling and implementing power quality monitoring systems was analysed. The main takeaways are that the standard allows for the implementation of advanced power quality mitigation algorithms, the modelling of standard power quality indices and the integration of power quality monitoring within a wide-range of auxiliary functions. Electric vehicle charging and fault localisation, isolation and restoration are examples of possible applications. This aspect of the IEC 61850 function and the interoperability requirement of compliant devices makes the standard suited for implementation within a distributed “smart grid”.

The adoption of bidirectional smart meters with power quality monitoring capabilities is becoming a popular way of conducting power quality studies on customer premises, and this is the mainline topic of the third literature study which focuses on the integration of smart grid technologies. Some key finding was the increasing use of open-source hardware and software, IoT technologies coming to the fore, and the increased risk of cyber-security vulnerabilities due to the “smartening” of electric power grids.

9.4.2. Theoretical aspects of Power Quality in DERs

Distributed Energy Resource (DER) connection methods often determine the monitoring approach employed by the Distribution System Operator (DSO). An unconditional connection approach allows connections to the distribution grid as long as the generation unit adheres to IEC EMC emission limits. Network operator-controlled systems give the DSO the authority to control and regulate generation units. The network operator-controlled system is the preferred approach as it allows the operator to monitor and mitigate power quality issues dynamically. Commonly experienced power quality issues in distributed generation units, specifically at the Point of Common Coupling (PCC), are described and investigated in this chapter.

Upon analysis, it can be concluded that monitoring the stability of the voltage in a steady-state, detecting and mitigating voltage fluctuations and variation events are the minimum requirements for effectively monitoring a distributed generation (DG) system. Moreover, finally monitoring the harmonic content on both the current and voltage systems are vital to ensuring that a DG system is operating within the allowable tolerances according to the

IEC 61000-3-2 and 61000-3-12 parts and the DSO’s specifications. Determining the necessary measurements allows for the modelling of power quality monitoring functions to develop a useful IEC 61850-based device model.

9.4.3. IEC 61850 Standard Device Model development

The following practical contribution involves the development of a device model for power quality monitoring based on the requirements identified in the theoretical background. An understanding of the IEC 61850-part 7-4 data and device modelling is required to create an appropriate device model. The theoretical study is performed on the IEC 61850 standard, covering logical node classes, data modelling, device modelling and the Substation Configuration Language (SCL) object model. An appropriate analogue front-end device is selected; the selected device contains the necessary metering functions as determined in Chapter 3. The ADE9000 is chosen as the DSP and analogue front-end of choice; a device model is developed by mapping the available measurement registers to IEC 61850-7-4 logical nodes and data attributes. Finally, the SCL object model is briefly explained to gain an understanding of how the device is eventually configured.

9.4.4. Implementation of Open-Source IEC 61850 software library

The next practical contribution is the implementation of the open-source library for IEC 61850. Analysis of available single-board computers resulted in the popular Raspberry Pi being chosen as a hardware platform. The Raspberry Pi proves to be a cost-effective, scalable, and configurable data processing and communications platform. The LibIEC61850 open-source software library is configured, tested, and loaded onto a Raspberry Pi. The procedure for building and testing the library on both Windows and Linux platforms is explained and tested. A fixed IP network is configured, and the Raspberry Pi device is given a static IP address of 192.168.1.96 and the Windows PC 192.168.1.95.

MMS communications are successfully tested between the device and a host PC on the configured network. The process of configuring an ICD file and all of the critical SCL file sections is detailed, the ICD file is written based on the device model developed in Chapter 4. Finally, a static model is generated using the static model generator included with the open-source library. The static model may be used to programmatically apply the developed device model to the ADE9000’s functions.

9.4.5. Design and implementation of Data Acquisition hardware

Next, the data acquisition hardware is designed, manufactured, and calibrated for interfacing the ADE9000 with the physical world using voltage dividers and burden resistors. A circuit with the necessary isolation and interfacing circuitry is designed, the schematic is converted to a PCB design and manufactured by a PCB fabricator. The Raspberry Pi is configured to communicate via the serial peripheral interface. Calibration of the inputs is performed using calculated conversion factors based on the component selection. Finally, the source code for communications procedures (SPI read and write) between the ADE9000 and Raspberry Pi, along with the main program's source code for MMS and GOOSE communications is written and described in detail.

9.4.6. Evaluate Power Quality Monitoring and IEC 61850 functionality

A testbed is developed for both testing the measurement and IEC 61850 functionality. Chapter 7 details the testing and evaluation of the device as a power quality meter, with several tests performed to verify the accuracy of readings performed and computed by the ADE9000, where possible, a reference meter is used to confirm accuracy. Signals are injected using the OMICRON CMC 356 Test set and the Test Universe software. The readings are first taken with a calibrated meter, and then the same readings are taken with the implemented power quality monitoring device. The device is found to be accurate and suitable for power quality measurements. The device's compatibility with MMS and ACSI services is also tested using IEDExplorer and select IEC 61850-10 conformance tests; the necessary services were found to conform with the requirements of the standard, thus theoretically proving interoperability.

9.4.7. IEC 61850 Interoperability test

Finally, In Chapter 8, a SEL RTAC is used to prove interoperability; this is achieved by creating a project featuring the implemented device's ICD using only SEL software. The ICD file created earlier is loaded into a SEL Architect Project to generate an SCD file. The SCD file is then loaded onto the RTAC device as a configuration file, and an HMI is created and loaded onto the RTAC for data visualisation. Voltage and current signal are injected using the CMC 356.

Upon starting the MMS server and GOOSE publisher on the implemented device, the SEL RTAC 3550 successfully reads and receives MMS and GOOSE messages respectively, thus proving interoperability between the implemented device and the SEL manufactured device. All traffic is monitored and verified using WireShark while the HMI display provides visual confirmation of the changes to measurement values. Both MMS dataset polling and reports are tested and working successfully, GOOSE messaging publishes periodically, and the subscribed messages change on the HMI display as expected. Proving interoperability and IEC 61850-based communications open the door for the device to be tested in a practical utility-scale application. An application that involves both the implemented device and the SEL RTAC is demonstrated theoretically.

The thesis deliverables provide a novel proof of concept for a device that may be deployed and modified depending on the use case and specific application. The following research contributions are identified:

1) Modelling and implementation of a viable, interoperable and low-cost IEC 61850-based power quality monitoring device using off-the-shelf components and an open-source software library. The implementation of this device provides a platform that is easily modifiable for use in a range of applications.
2) The method of modelling IEC 61850 logical nodes and functions to analogue front-end measurement registers presented in this research demonstrates how to design and map information to equivalent logical nodes. The knowledge demonstrated shows how device modelling techniques can be used to extract additional information for mapping to existing logical nodes or to develop new ones.

9.4.8. Software development

Several pieces of source code were modified and developed to build the new system. Table 9-1 describes the software modules developed in this project. The various software modules are shown, including how and where they are applied and also the programming language used for the implementation.

Table 9-1: Software Developed for the Research Project

Software Module	Application	Environment/Language
MMS Server	The MMS Server provides the MMS client with datasets for polling or reports. Integrated into the main loop of the program	C and C++ (Genie and Visual Studio)
GOOSE Publisher	Used for periodically publishing GOOSE messages. Integrated into the main loop of the program	C and C++ (Genie and Visual Studio)
MMS Client	They are used for testing the MMS server implementation.	C (Genie and Visual Studio)

ADE9000 SPI Comms	Read and write commands for accessing ADE9000 data registers. Conversion constant application	C++ (Genie)
ICD File	Use to define the IED model of the “RasPQ” power quality monitor, and additionally to generate the static IED model.	Notepad++ and XML Marker
SCD File	IEDs configured within an IEC 61850 project are described. MMS Client -RTAC 3550 MMS Server - RasPQ	AcSELerator Architect
RTAC Configuration File	RTAC Tags are connected to data objects contains with the SCD file	AcSELerator RTAC
RTAC HMI Display	RTAC tags are displayed visually to simulate a typical SCADA display.	AcSELerator Diagram Builder

9.5. Application of developed Power Quality Monitoring device

The implemented power quality monitoring device can be used for both industrial and academic applications. This section, in brief details some of the possible applications.

9.5.1. Industrial application

The implemented device can be utilised in power utilities and for several current and future applications with minimal modifications.

- The device may be used for utility-scale monitoring at hundreds of distributed generators’ Points of Common Coupling.
- Check-metering on customer premises to verify utility energy consumption.
- Power Quality mitigation algorithms can be implemented using IEC 61850-based control functions.
- Short term power quality studies may be performed due to the portability and small overall footprint of the device.
- The device may provide a low-cost alternative for smaller municipalities trying to implement an IEC 61850-based communications system.
- It could be used as a training and lab testing tool for protection technicians and engineers.

9.5.2. Academic application

The implemented device can be utilised in power utilities and

- May be used as an educational tool for undergraduate university students learning about the IEC 61850 standard, Linux Operating Systems, networking, software engineering, embedded systems and substation automation.
- The implemented device may also be used for developing new logical nodes for power quality monitoring and other applications at the postgraduate level.

9.6. Future work

This research was performed as a proof of concept for the monitoring of power quality in a distributed environment and as a starting point for the development of cheaper and more accessible monitoring devices for distributed generators.

- Multiple vendors can be tested for further interoperability tests.
- The device may be placed within a simulated environment to test its effectiveness at monitoring DERs, a system like the Real-Time Digital Simulator (RTDS) or similar could be used to build a network and performed "Hardware-in-the-Loop" testing. Virtually placing the device within the system would allow for a performance assessment of its performance in a true-to-life simulation of real-world scenarios.
- Once hardware-in-the-loop simulations have been completed, a field test within the industry could be feasibly arranged and run over several months or as a pilot project.
- This research project mainly focused on the power quality monitoring aspect of the ADE9000. The device is primarily a polyphase energy meter which supports active energy standards: IEC 62053-21 and IEC 62053-22; EN50470-3; OIML R46; and ANSI C12.20 and reactive energy standards: IEC 62053-23, IEC 62053-24, this should make the device ideal as a replacement for existing older energy meters.
- The implemented device could be tested as an IEC 61850-compliant alternative to existing Smart Meters using the Intelligent Logical Node (iLN) Architecture described in Vyatkin *et al.*, (2012). Using the IEC 61499 standard for the Architecture for Distributed Automation along with the IEC 61850 standard for the communications architecture, a similar system may be designed and modelled using a number devices to create a network with distributed intelligence.

9.7. Publication

Publications Related to the Thesis

Journal Publications

- Hara M., Kriger C. 2020. “Implementation of an IEC 61850-based metering device using open-source software”, European Journal of Engineering Research and Science (EJERS) (submitted for publication)
- Hara M., Kriger C. 2020. “Data Modelling of Analog Front-End functions to IEC 61850 Logical Nodes”, International Journal of Computers Communication and Control (IJCCC) (submitted for publication)

9.8. Conclusion

This chapter presents a conclusion to this research work. It starts with the initially proposed aim and stated objectives. The project deliverables and theoretical and practical achievements are also presented. The implemented testbeds, together with the validation tests that were conducted, are described. The possible applications of the implemented device are highlighted and discussed. Potential future research work that may be of interest is proposed. Finally, a list of publications related to this research is presented at the end of this thesis.

REFERENCES

Abapour, S. & Zare, K. 2018. Reliability-Based Scheduling of Active Distribution System With the Integration of Wind Power Generation. *Operation of Distributed Energy Resources in Smart Distribution Networks*, (March): 203–230.

Aftab, M.A., Hussain, S.M.S., Ali, I. & Ustun, T.S. 2018. IEC 61850 and XMPP communication based energy management in microgrids considering electric vehicles. *IEEE Access*, 6: 35657–35668.

Alonso, M., Amaris, H., Alcala, D. & Florez, D.M.R. 2020. Smart sensors for smart grid reliability. *Sensors (Switzerland)*, 20(8).

Analog Devices. 2019. A Beginner's Guide to Digital Signal Processing (DSP) | Design Center | Analog Devices. http://www.analog.com/en/design-center/landing-pages/001/beginners-guide-to-dsp.html 23 August 2019.

Analog Devices Inc. 2017a. ADE9000 Technical Reference Manual Functionality and Features of the ADE9000 High Performance, Multiphase Energy and Power Quality Monitoring IC. : 86. www.analog.com.

Analog Devices Inc. 2017b. EVAL-ADE9000-Shield. https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/EVAL-ADE9000-Shield.html#eb-overview.

Ananth, I. & Morcos, M.. 1999. Power quality monitoring system: A case study in DSP-Based Solutions for Electric Power Industry. *1999 IEEE Power Engineering Society Summer Meeting, PES 1999 - Conference Proceedings*, 1(April 1990): 325–327.

Antonova, G., Nardi, M., Scott, A. & Pesin, M. 2012. Distributed generation and its impact on power grids and microgrids protection. *2012 65th Annual Conference for Protective Relay Engineers*: 152–161.

Apostolov, A.P. 2007. IEC 61850 and Power-quality Monitoring and Recording. In *Power Quality: Mitigation Technologies in a Distributed Environment*. 381–415.

Apostolov, A.P. 2009. Impact of IEC 61850 on Power Quality Monitoring and Recording A. Apostolov OMICRON electronics, USA. In *CIRED 2009 - 20th International Conference and*

Exhibition on Electricity Distribution - Part 1. Prague, Czech Republic: IET. https://ieeexplore.ieee.org/document/5255670.

Artale, G., Cataliotti, A., Cosentino, V., Guaiana, S., Di Cara, D., Panzavecchia, N., Tine, G., Dipaola, N. & Sambataro, M.G. 2018. PQ Metrics Implementation on Low-Cost Smart Metering Platforms. A Case Study Analysis. *9th IEEE International Workshop on Applied Measurements for Power Systems, AMPS 2018 - Proceedings*, (4000494849): 1–6.

Auler, L.F. & D'Amore, R. 2003. Power Quality and Control Using Ethernet Networks. In *10th International Conference on Harmonics and Quality of Power. Proceedings (Cat. No.02EX630)*. Rio de Janeiro, Brazil: IEEE. https://ieeexplore.ieee.org/document/1221434.

Benitez, M., Isasi, C., Galeano, R., Gregor, D., Arzamendia, M., Gregor, R., Rodas, J. & Kali, Y. 2020. Design of a Bidirectional Electrical Energy Measurement System for Applications in Distributed Generation. In *2020 5th International Conference on Renewable Energies for Developing Countries (REDEC)*. Marrakech, Morocco: IEEE: 1–6. https://ieeexplore.ieee.org/abstract/document/9163605.

Bracale, A., Caramia, P., Carpinelli, G., Russo, A. & Verde, P. 2011. Site and System Indices for Power-Quality Characterization of Distribution Networks With Distributed Generation. *IEEE Transactions on Power Delivery*, 26(3): 1304–1316.

Brittes, J.L.P., Nunes, E., Jardini, J.A., Magrini, L.C., Hokama, W.S. & Fernández Silva, L.G. 2016. 'Smartizing' power quality assessment based on IEC smart substation automation. *2015 IEEE PES Innovative Smart Grid Technologies Latin America, ISGT LATAM 2015*: 225–230.

Chambers, T. 2018. Distributed energy resources and power quality. *Papers Presented at the Annual Conference - Rural Electric Power Conference*, 2018-May: 37–41.

Chandrasekaran, A., Kollipara, R. & Sundaram, A. 1994. Development of Customized Power Electronics System Analysis Software for Use with a Power Quality Workstation. In 1–9.

Chaudray, S.H. & Gangil, G. 2016. Mitigation of voltage sag / swell using Dynamic voltage restorer. *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) - 2016 OPTIMIZATION*, (June): 4046–4051.

Chen, S. 2004. Open Design of Networked Power Quality Monitoring Systems. *IEEE Transactions on Instrumentation and Measurement*, 53(2): 597–601.

Chua, K.H., Wong, J., Lim, Y.S., Taylor, P., Morris, E. & Morris, S. 2011. Mitigation of voltage unbalance in low voltage distribution network with high level of photovoltaic system. *Energy Procedia*, 12(603): 495–501. http://dx.doi.org/10.1016/j.egypro.2011.10.066.

City of Cape Town. 2017. REQUIREMENTS FOR SMALL-SCALE EMBEDDED GENERATION Application process to become a small-scale embedded generator in the City of Cape Town Record of Significant Recent Guideline Revisions. : 1–45. www.capetown.gov.za/elecserviceforms.

Clemmensen, J.M. 1990. Power quality site survey instrumentation and measurement techniques. In *IEEE Conference Record of Industrial and Commercial Power Systems Technical Conference*. 126–135.

Council of the European Union. 1985. *Council Directive of 25 July 1985 on the approximation of the laws, regulations and administrative provisions of the Member States concerning liability for defective products*.

Dong, X., Xiong, G. & Hou, J. 2012. Construction of Artificial Grid Systems Based on ACP Approach. *Service Science, Management, and Engineering*, (January 2019): 285–304.

Dulău, L.I., Abrudean, M. & Bică, D. 2014. Effects of Distributed Generation on Electric Power Systems. *Procedia Technology*, 12: 681–686. https://linkinghub.elsevier.com/retrieve/pii/S2212017313007342.

EBay. 2019. Dranetz 646-3 Disturbance Analyzer 6463 _ eBay. https://www.ebay.com/itm/Dranetz-646-3-Disturbance-Analyzer-6463-/401621712464 22 August 2019.

EERE. 2017. Consumer vs Prosumer: What's the Difference? *U.S. Department of Energy*. https://www.energy.gov/eere/articles/consumer-vs-prosumer-whats-difference.

Ferreira, P., Miranda, R.N., Cruz, P.M. & Mendonca, H.S. 2019. Multi-Protocol LoRaWAN/Wi-Fi Sensor Node Performance Assessment for Industry 4.0 Energy Monitoring. *Proceedings of the 2019 9th IEEE-APS Topical Conference on Antennas and Propagation in Wireless Communications, APWC 2019*: 403–407.

Ferrigno, L., Landi, C. & Laracca, M. 2008. FPGA-based measurement instrument for power quality monitoring according to IEC standards. In *IEEE Instrumentation and Measurement Technology Conference*. 906–911.

Filipova, A. & Morris, M. 2018. Small-Scale Embedded Generation in South African municipalities. *Sustainable Energy Africa*, March(May): 1–9. http://samsetproject.site11.com/wp-content/uploads/2014/10/Embedded-Generation-Case-Study-South-Africa-FINAL.pdf.

Garcez, J.N., Bezerra, U.H., Castro, A.L.S., Alves, R.N.C., Lima, W.J.F., Castro, A.R.G., MacIel, J.H., Moscoso, M.N., Tenorio, K., Tupiassú, A.A. & Mesquita, J.E. 2001. Integrating a power quality monitoring system in a distribution control center. In *2001 IEEE Porto Power Tech Proceedings*. Porto, Portugal: 331–335.

Della Giustina, D., De Sotomayor, A.A., Dedè, A. & Ramos, F. 2020. A model-based design of distributed automation systems for the smart grid: Implementation and validation. *Energies*, 13(14): 1–19.

Grady, M. 2012. *Understanding Power System Harmonics*. Austin: University of Texas at Austin. www.ece.utexas.edu/~grady.

Grigorescu, S.D., Ghita, O.M., Cepisca, C. & Vintea, A.S. 2013. Power quality monitoring systems for smart grid networks. *2013 - 8th International Symposium on Advanced Topics in Electrical Engineering, ATEE 2013*.

Hairabedian, B. 1989. System for Surveying Power Quality. In *6th IEEE Conference Record., Instrumentation and Measurement Technology Conference*. Washington, DC. https://ieeexplore.ieee.org/document/36893?denied=.

Hariri, A. & Faruque, M.O. 2014. Impacts of distributed generation on power quality. *2014 North American Power Symposium, NAPS 2014*, (September 2014).

Hassan, S.R., Rehman, A., Shabbir, N. & Unbreen, A. 2019. Comparative Analysis of Power Quality Monitoring Systems. *NFC IEFR Journal of Engineering and Scientific Research*, 7(1): 19–23.

Hidayatullah, N.A., Paracha, Z.J. & Kalam, A. 2011. Impact of Distributed Generation on Smart Grid Transient Stability. *Smart Grid and Renewable Energy*, 02(02): 99–109.

Hong, D.H.D., Lee, J.L.J. & Choi, J.C.J. 2002. Power Quality Monitoring System Using Power Line Communication. *2005 5th International Conference on Information Communications Signal Processing*, 1: 18–21. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1178253.

Huo, Z. & Zhang, L. 2008. Research on Graphics Model Design for IEC61850 SCL Visual Configuration. In *IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*. 1–5.

IEC. 2011. *Communication networks and systems for power utility automation – Part 7-1: Basic communication structure – Principles and models*. 2nd ed.

IEC. 2003. Communication networks and systems in substations Part 5: Communication requirements for functions and device models. *Iec*, (2): 306. https://webstore.iec.ch/publication/6012.

IEC. 2005a. IEC 61850 - Communication networks and systems for power utility automation - Part 10: Conformance testing. *IEC 61850*, 2: 1–170. https://webstore.iec.ch/publication/6008.

IEC. 2005b. IEC 61850 - Communication networks and systems for power utility automation - Part 10: Conformance testing. *IEC 61850*: 21.

IEC. 2008. IEC 61850 Part 7-420 DER Logical Nodes: Communication Networks and Systems for Power Utility Automation for Distributed Energy Resources (DERs). : 11.

IEC. 2012. IEC61850 - Communication networks and systems for power utility automation – Part 6: Configuration description language for communication in electrical substations related to IEDs.

IEC. 2005c. IEC61850 - Communication Networks and Systems in Substations: Part 08-1: Specific Communication Service Mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3. , 1: 21.

IEC. 2005d. IEC61850 - Communication Networks and Systems in Substations: Part 08-1: Specific Communication Service Mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3. , 1: 100.

IEC. 2005e. IEC61850 - Communication Networks and Systems in Substations: Part 08-1: Specific Communication Service Mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3. : 102.

IEC TC 77. 2009. *Electromagnetic compatibility (EMC) Part 4-7: Testing and measurement techniques - General guide on harmonics and interharmonics measurements and* *instrumentation, for power supply systems and equipment connected thereto*.

IEEE. 1995. IEEE Recommended practice for monitoring electric power quality. *IEEE Standards*: 1–70.

Ingalalli, A., Silpa, K.S. & Gore, R. 2017. SCD based IEC 61850 traffic estimation for substation automation networks. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, (September 2017): 1–8.

Inverter Drive Systems LTD. 2019. Harmonic Distortion 3 phase with harmonics. https://www.inverterdrivesystems.com/wp-content/uploads/2016/01/Harmonic-Distortion-3-phase-with-harmonics.png 29 September 2019.

Jinhao, W., Da, L., Xiaoyang, M. & Honggeng, Y. 2019. Research on the test of the communication protocol of power quality monitoring platform. *2019 IEEE PES Innovative Smart Grid Technologies Asia, ISGT 2019*: 3351–3356.

Lee, P.K. & Lai, L.L. 2008. A practical approach to wireless Power Quality, Energy and Facilities Monitoring System. *IEEE Power and Energy Society 2008 General Meeting: Conversion and Delivery of Electrical Energy in the 21st Century, PES*: 1–3.

Lee, R.P.K., Lai, L.L. & Tse, N. 2002. A web-based multi-channel power quality monitoring system for a large network. *2002 Fifth International Conference on Power System Management and Control Conf. Publ. No. 488)*, (488): 112–117. https://ieeexplore.ieee.org/document/1032148.

Leou, R.C., Chang, Y.C. & Teng, J.H. 2001. A web-based power quality monitoring system. *Proceedings of the IEEE Power Engineering Society Transmission and Distribution Conference*, 3(SUMMER): 1504–1508.

Long, M. 2019. BeagleBone Black vs Raspberry Pi. https://www.electromaker.io/blog/article/beaglebone-black-vs-raspberry-pi 18 July 2020.

Mguzulwa, N.R. 2018. Investigation of Interoperability of Iec 61850 Protection Functions.

Microchip. 2015. MCP39F511 Power-Monitoring IC with Calculation and Energy Accumulation. : 62.

Mlakic, D., Reza Baghaee, H., Nikolovski, S., Vukobratovic, M. & Balkic, Z. 2019. Conceptual

Design of IoT-Based AMR Systems Based. *Energies*, 12(4281). https://www.mdpi.com/1996-1073/12/22/4281/htm.

Montoya, F.G., Alcayde, A., Sánchez, P., Gómez, J. & Martín, F. 2011. zEnergy: An open-source project for power quality assessment and monitoring. *International Conference on Power Engineering, Energy and Electrical Drives*, (May).

Moreno-Munoz, A. 2007. *Power Quality : Mitigation Technologies in a Distributed Environment*. Springer. https://www.springer.com/gp/book/9781846287718.

North American Electric Reliability Corporation. 2017a. *Distributed Energy Resources: Connection Modeling and Reliability Considerations*. https://www.nerc.com/comm/Other/essntlrlbltysrvcstskfrcDL/Distributed_Energy_Resources_Report.pdf.

North American Electric Reliability Corporation. 2017b. *Distributed Energy Resources: Connection Modeling and Reliability Considerations*.

Ozansoy, C.R., Zayegh, A. & Kalam, A. 2009. The application-view model of the international standard IEC 61850. *IEEE Transactions on Power Delivery*, 24(3): 1132–1139.

Pavel-ch. 2020. IEDExplorer: IEC61850 IED Explorer in .net. *Sourceforge*. https://sourceforge.net/projects/iedexplorer/ 5 September 2020.

Peng, S., Luo, A., Chen, Y. & Lv, Z. 2011. Dual-loop power control for single-phase grid-connected converters with LCL filter. *Journal of Power Electronics*, 11(4): 456–463.

Power Quality World. 2011. Harmonic Distortion - Inverter Drive Systems an ABB AVP. *Power Quality World*. https://www.inverterdrivesystems.com/harmonic-distortion/ 11 October 2019.

Putrus, G., Wijayakulasooriya, J. & Minns, P. 2007. Power quality: Overview and monitoring. *ICIIS 2007 - 2nd International Conference on Industrial and Information Systems 2007, Conference Proceedings*, (August): 551–558.

Raja Singh, R., Yash, S.M., Shubham, S.C., V., I., V., V., P., S. & V., S. 2020. IoT embedded cloud-based intelligent power quality monitoring system for industrial drive application. *Future Generation Computer Systems*, 112: 884–898. https://doi.org/10.1016/j.future.2020.06.032.

Raspberry Pi Foundation. 2019. Raspberry Pi 4 Tech Specs. https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/ 16 July 2020.

Rauch, G.B., Sabin, D.D., McGranaghan, M. & Sundaram, A. 1996. Implementing system-wide power quality monitoring plans. *Proceedings of the IEEE Power Engineering Society Transmission and Distribution Conference*: 37–44.

Regula, M., Otcenasova, A., Roch, M., Bodnar, R. & Repak, M. 2016. SCADA system with power quality monitoring in Smart Grid model. *EEEIC 2016 - International Conference on Environment and Electrical Engineering*: 1–5.

Ruiz, J., Ortuondo, J., Palacios, N., Izquierdo, J., Leturiondo, L.A., Aramendi, E. & Amantegui, J. 1995. Real-time power supply quality measurement and monitoring multichannel system. *IEEE Transactions on Power Delivery*, 10(3): 1190–1199.

Saari, H., Koponen, P., Tahvanainen, E. & Lindholm, T. 1996. Remote Reading and Data Management System for kWh-Meters with Power Quality Monitoring. In *Eighth International Conference on Metering and Tariffs for Energy Supply (Conf. Publ. No. 426)*. Brighton, UK: 3–5. https://ieeexplore.ieee.org/document/637404?denied=.

Sakis Meliopoulos, A.P., Fardanesh, B., Zelingher, S. & Cokkinides, G.J. 2000. Harmonic measurement system via synchronized measurements. *Proceedings of the IEEE Power Engineering Society Transmission and Distribution Conference*, 2: 1094–1100.

SANS. 2002. SANS 10142-1-2 : 201X DRAFT : SOUTH AFRICAN NATIONAL STANDARD The wiring of premises Part 1-2 : Specific requirements for embedded generation installations connected to the low voltage distribution network in South Africa.

Shadare, A.E., Sadiku, M.N.O. & Musa, S.M. 2017. Electromagnetic compatibility issues in critical smart grid infrastructure. *IEEE Electromagnetic Compatibility Magazine*, 6(4): 63–70.

Sikorski, T. & Rezmer, J. 2018. Distributed Generation and Its Impact on Power Quality in Low-Voltage Distribution Networks. In *Power Quality Issues in Distributed Generation*. 13.

SISCO Inc. 1995. Overview and Introduction to the Manufacturing Message Specification (MMS). *ReVision*, (11/95). http://www.sisconet.com.

Steel, M.D.W. 2016. Using IEC 61850 GOOSE in wide-area solar-energy storage to speed-up power quality smoothing. *Proceedings of the 2016 Australasian Universities Power*

Engineering Conference, AUPEC 2016: 1–5.

Topscomm. 2016. AMI System, Meter Data Collection System, Meter Data Management System. https://www.topsmetering.com/topscomm-ami-solution_n17 26 August 2019.

Volkwyn, C. 2017. The second wave of smart meter rollouts begin in Italy and Sweden. https://www.smart-energy.com/regional-news/europe-uk/second-wave-smart-meter-rollouts-begins-italy-sweden/#:~:text=Italy and Sweden were the,meters in the last decade.&text=In Italy%2C Enel’s distribution arm,million in the following decade. 20 November 2020.

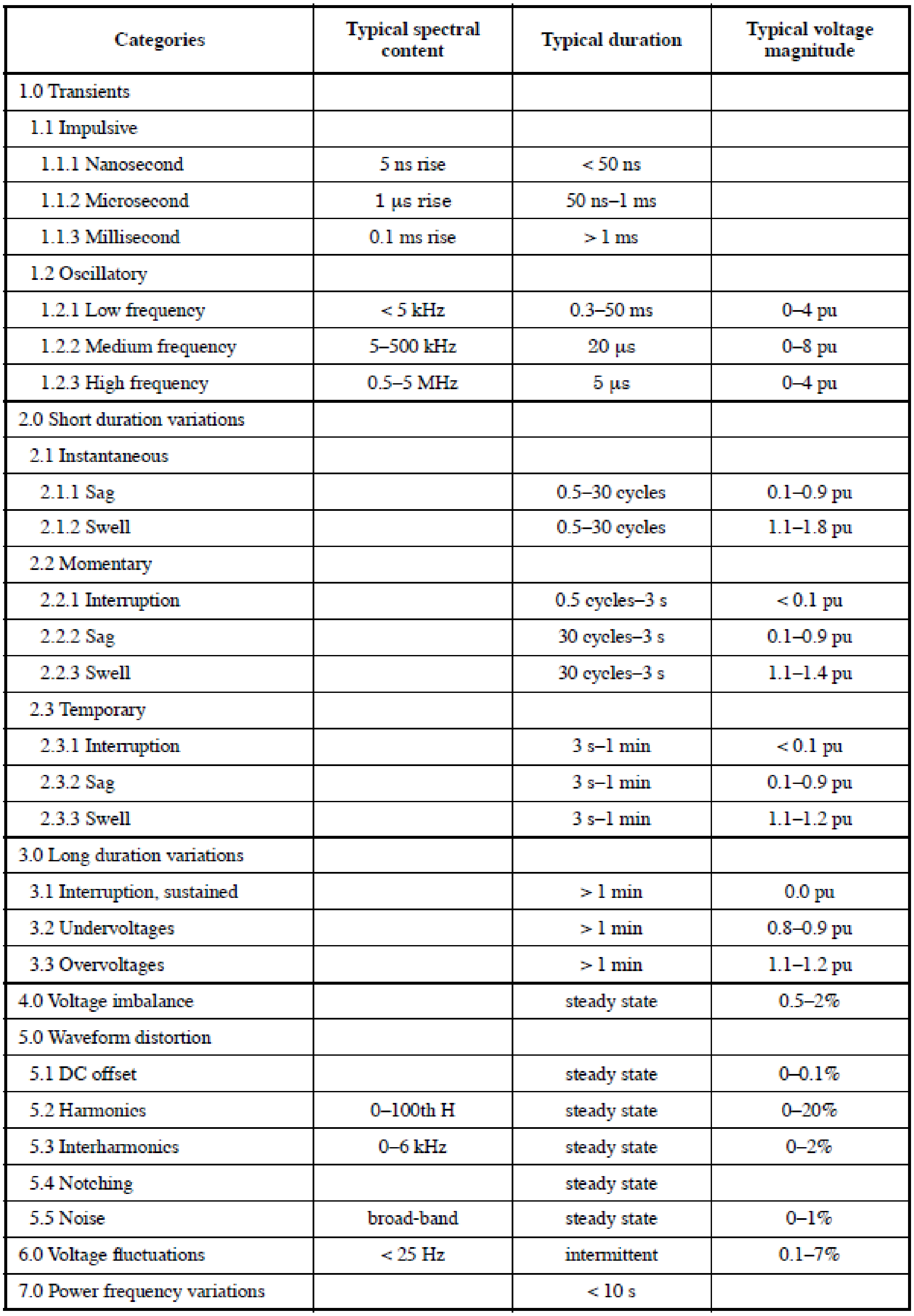
Vyatkin, V., Zhabelova, G., Yang, C.W., McComas, D. & Chouinard, J. 2012. Intelligent IEC 61850/61499 logical nodes for smart metering. *2012 IEEE Energy Conversion Congress and Exposition, ECCE 2012*: 1220–1227.

Zillgith, M. 2014. IEC 61850 ACSI services supported by the library. *libiec61850.com*. https://libiec61850.com/libiec61850/documentation/library-features/.

APPENDICES

Appendix A: Duration and Magnitude of Power Quality Interruptions

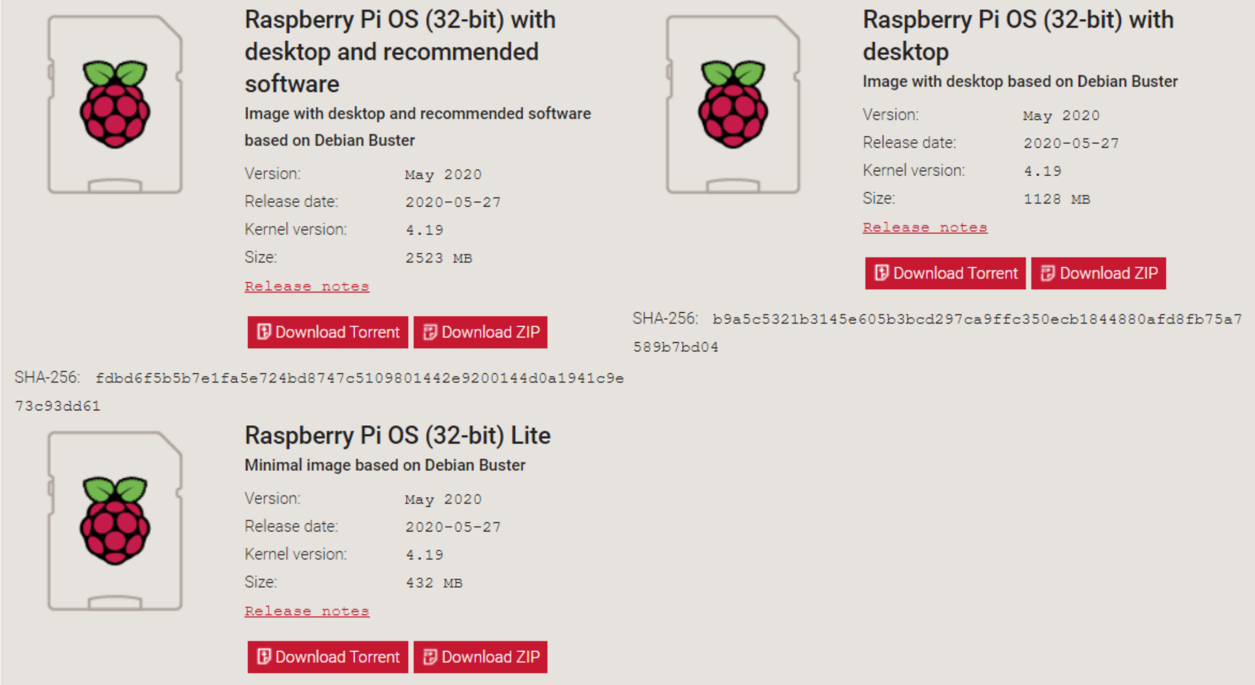
Categories of power quality events along with their duration and magnitudes are listed



Categories	Typical spectral content	Typical duration	Typical voltage magnitude
1.0 Transients			
1.1 Impulsive			
1.1.1 Nanosecond	5 ns rise	< 50 ns	
1.1.2 Microsecond	1 μs rise	50 ns–1 ms	
1.1.3 Millisecond	0.1 ms rise	> 1 ms	
1.2 Oscillatory			
1.2.1 Low frequency	< 5 kHz	0.3–50 ms	0–4 pu
1.2.2 Medium frequency	5–500 kHz	20 μs	0–8 pu
1.2.3 High frequency	0.5–5 MHz	5 μs	0–4 pu
2.0 Short duration variations			
2.1 Instantaneous			
2.1.1 Sag		0.5–30 cycles	0.1–0.9 pu
2.1.2 Swell		0.5–30 cycles	1.1–1.8 pu
2.2 Momentary			
2.2.1 Interruption		0.5 cycles–3 s	< 0.1 pu
2.2.2 Sag		30 cycles–3 s	0.1–0.9 pu
2.2.3 Swell		30 cycles–3 s	1.1–1.4 pu
2.3 Temporary			
2.3.1 Interruption		3 s–1 min	< 0.1 pu
2.3.2 Sag		3 s–1 min	0.1–0.9 pu
2.3.3 Swell		3 s–1 min	1.1–1.2 pu
3.0 Long duration variations			
3.1 Interruption, sustained		> 1 min	0.0 pu
3.2 Undervoltages		> 1 min	0.8–0.9 pu
3.3 Overvoltages		> 1 min	1.1–1.2 pu
4.0 Voltage imbalance		steady state	0.5–2%
5.0 Waveform distortion			
5.1 DC offset		steady state	0–0.1%
5.2 Harmonics	0–100th H	steady state	0–20%
5.3 Interharmonics	0–6 kHz	steady state	0–2%
5.4 Notching		steady state	
5.5 Noise	broad-band	steady state	0–1%
6.0 Voltage fluctuations	< 25 Hz	intermittent	0.1–7%
7.0 Power frequency variations		< 10 s	

Appendix B: Procedure for Installing Raspberry Pi OS on the Raspberry Pi 4

Step 1: Download Raspberry Pi OS (Formerly Raspbian)



Raspberry Pi OS (32-bit) with desktop and recommended software

Image with desktop and recommended software based on Debian Buster

Version:	May 2020
Release date:	2020-05-27
Kernel version:	4.19
Size:	2523 MB

Release notes

SHA-256: fdbd6f5b5b7e1fa5e724bd8747c5109801442e9200144d0a1941c9e73c93dd61

Raspberry Pi OS (32-bit) with desktop

Image with desktop based on Debian Buster

Version:	May 2020
Release date:	2020-05-27
Kernel version:	4.19
Size:	1128 MB

Release notes

SHA-256: b9a5c5321b3145e605b3bcd297ca9ffc350ecb1844880afd8fb75a7589b7bd04

Raspberry Pi OS (32-bit) Lite

Minimal image based on Debian Buster

Version:	May 2020
Release date:	2020-05-27
Kernel version:	4.19
Size:	432 MB

Release notes

Official images for recommended operating systems are available to download from the Raspberry Pi website downloads page:

https://www.raspberrypi.org/downloads/raspberry-pi-os/

Alternative distributions are available from third-party vendors.

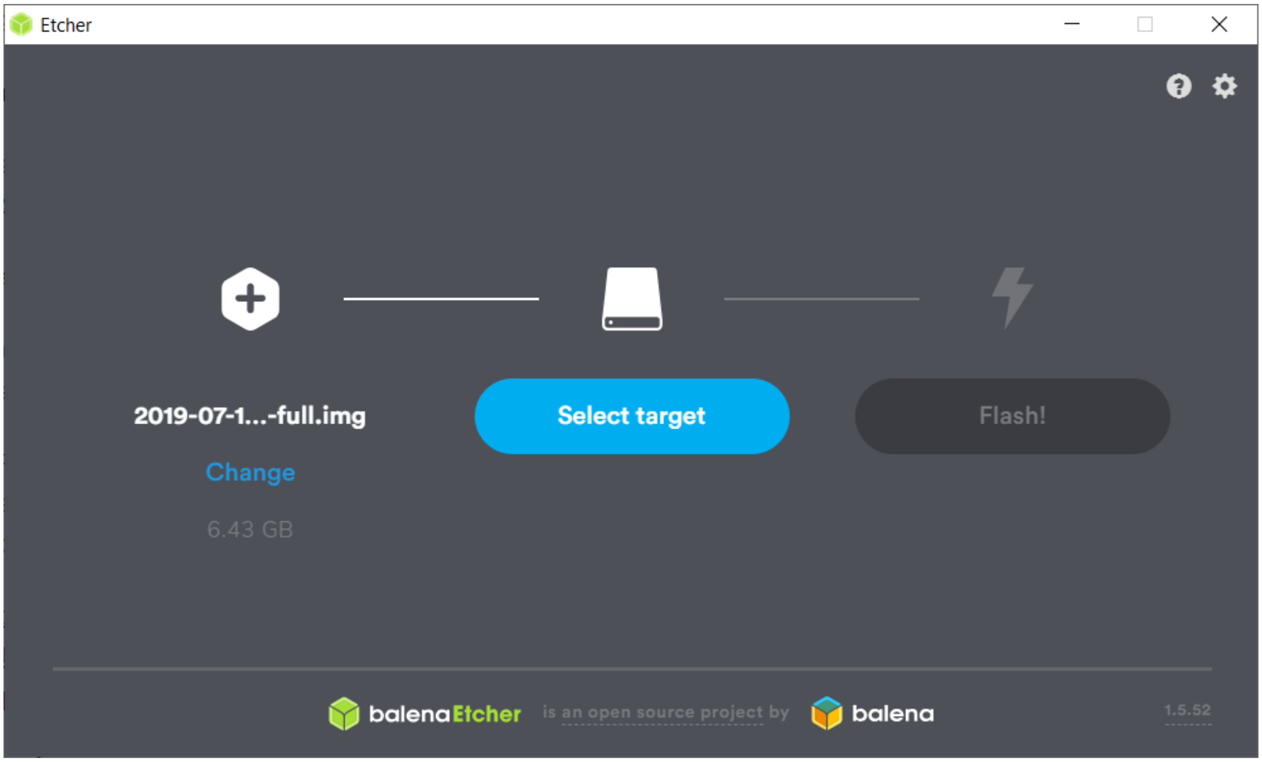
Step 2: Unzip the file

The Raspbian disc image is compressed, so unzipping it is required. The file uses the ZIP64 format, so specific programs may need to be used to unzip it. A few recommended programs depending on the platform are:

- Windows-based systems: 7-Zip
- Mac OS: The Unarchiver
- Linux: Unzip

Step 3: Write the disc image to your microSD card

The disc image may be written using BalenaEtcher and needs to be written on a MicroSD card with a minimum size of 16 GB. A screenshot of BalenaEtcher is shown below:



The installation steps are as follows:

1) Download the Windows installer from https://www.balena.io
2) Run balenaEtcher and select the unzipped Raspberry Pi OS image file
3) Select the SD card drive the image shall be written to
4) Finally, click “Flash!” to write the Raspberry Pi OS image to the SD card
5) A progress bar appears. Once complete, the utility automatically unmounts the SD card, so it is safe to remove it from the computer.

Step 4: Insert the microSD card in the Raspberry Pi and begin booting process

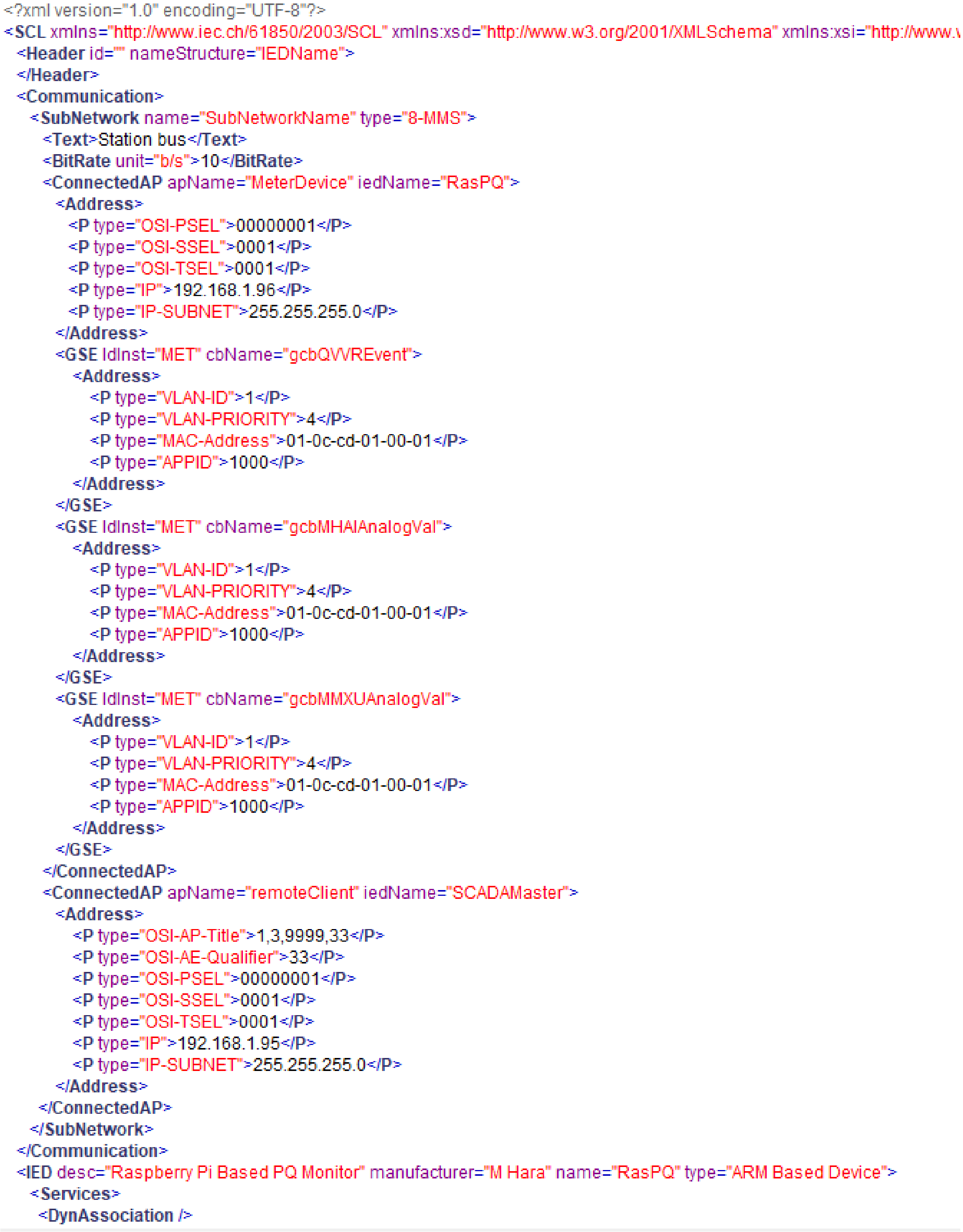
The SD Card may now be inserted into the Raspberry Pi and be powered up.

For the official Raspberry Pi OS, if manual login is required, the default user name is “pi”, with password “raspberry”.

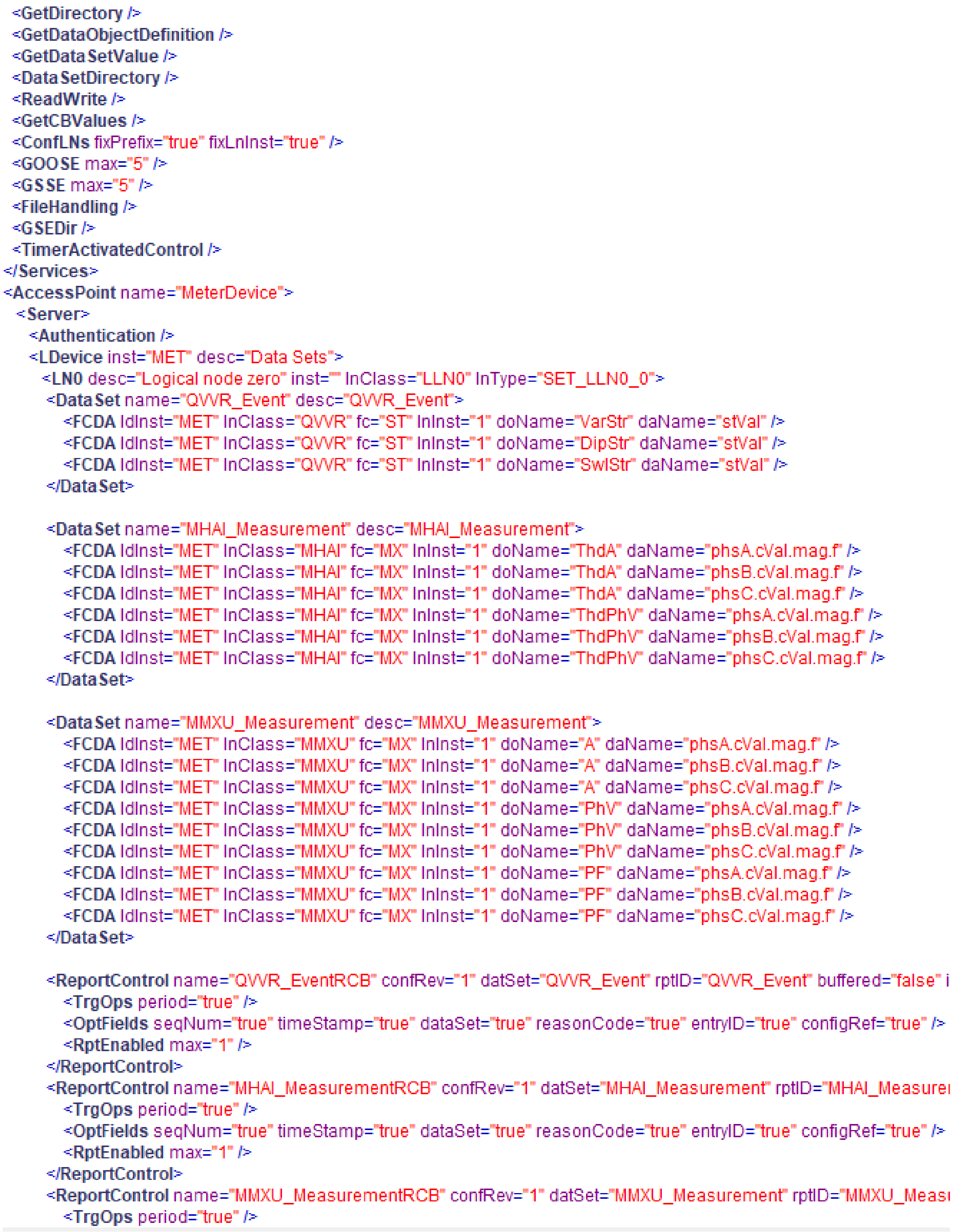
It is advisable to change the default password to ensure security.

Appendix C: Complete ICD File Configuration in .XML Format

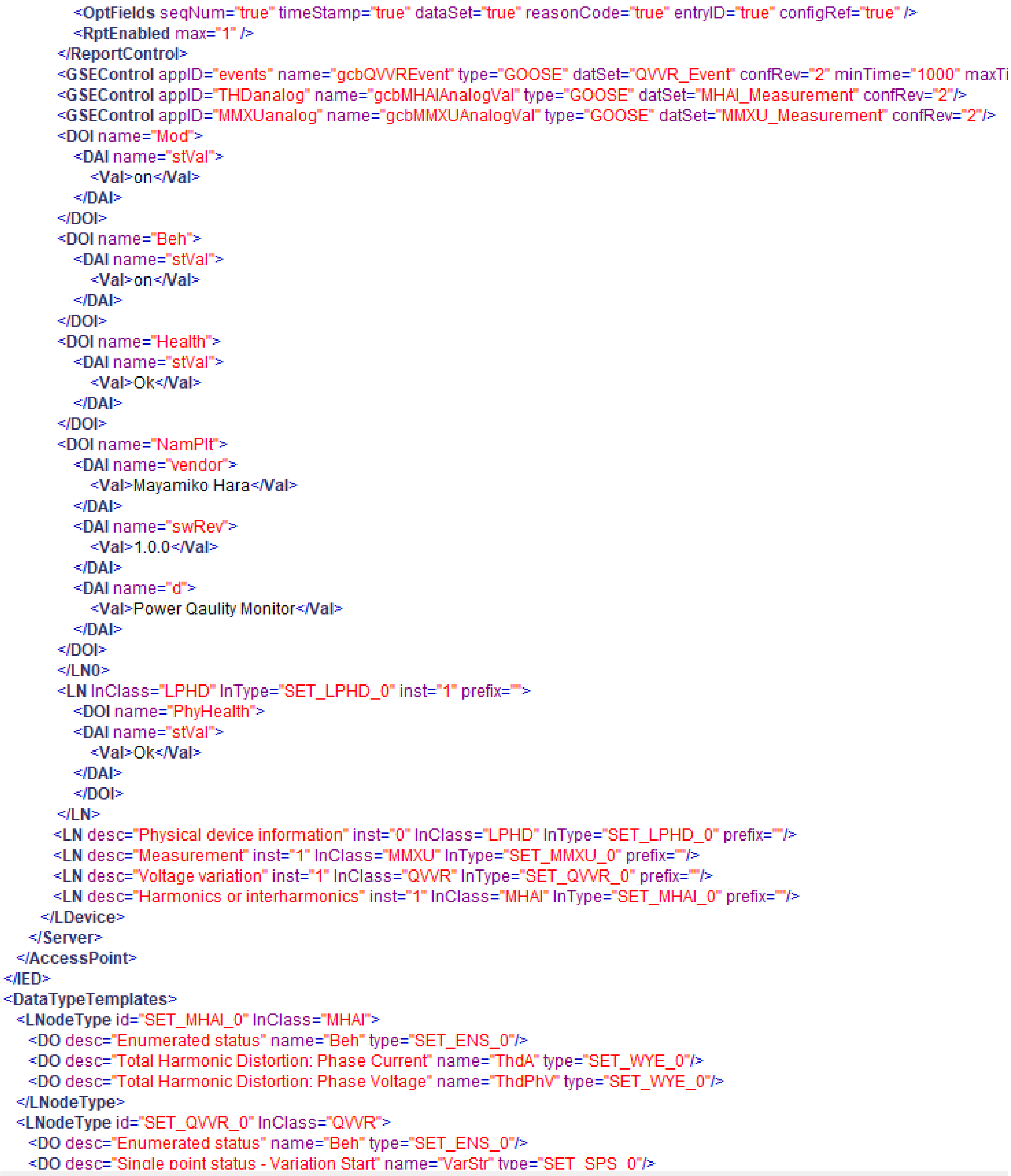
The following is an extract of the SCL file used to configure the implemented device.



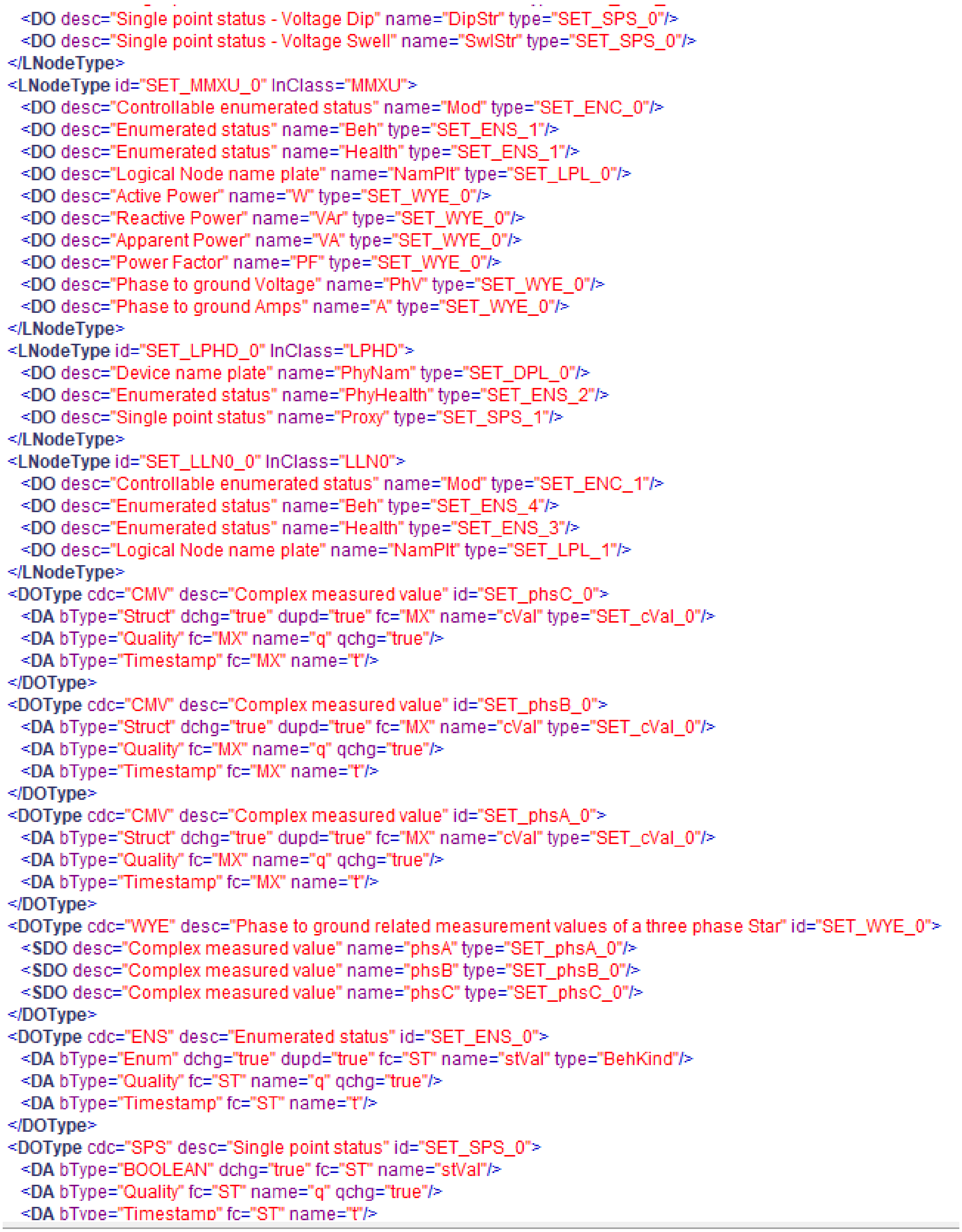
```
<?xml version="1.0" encoding="UTF-8"?>
<SCL xmlns="http://www.iec.ch/61850/2003/SCL" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.
  <Header id="" nameStructure="IEDName">
  </Header>
  <Communication>
    <SubNetwork name="SubNetworkName" type="8-MMS">
      <Text>Station bus</Text>
      <BitRate unit="b/s">10</BitRate>
      <ConnectedAP apName="MeterDevice" iedName="RasPQ">
        <Address>
          <P type="OSI-PSEL">00000001</P>
          <P type="OSI-SSEL">0001</P>
          <P type="OSI-TSEL">0001</P>
          <P type="IP">192.168.1.96</P>
          <P type="IP-SUBNET">255.255.255.0</P>
        </Address>
        <GSE ldInst="MET" cbName="gcbQVVREvent">
          <Address>
            <P type="VLAN-ID">1</P>
            <P type="VLAN-PRIORITY">4</P>
            <P type="MAC-Address">01-0c-cd-01-00-01</P>
            <P type="APPID">1000</P>
          </Address>
        </GSE>
        <GSE ldInst="MET" cbName="gcbMHAIAnalogVal">
          <Address>
            <P type="VLAN-ID">1</P>
            <P type="VLAN-PRIORITY">4</P>
            <P type="MAC-Address">01-0c-cd-01-00-01</P>
            <P type="APPID">1000</P>
          </Address>
        </GSE>
        <GSE ldInst="MET" cbName="gcbMMXUAnalogVal">
          <Address>
            <P type="VLAN-ID">1</P>
            <P type="VLAN-PRIORITY">4</P>
            <P type="MAC-Address">01-0c-cd-01-00-01</P>
            <P type="APPID">1000</P>
          </Address>
        </GSE>
      </ConnectedAP>
      <ConnectedAP apName="remoteClient" iedName="SCADAMaster">
        <Address>
          <P type="OSI-AP-Title">1,3,9999,33</P>
          <P type="OSI-AE-Qualifier">33</P>
          <P type="OSI-PSEL">00000001</P>
          <P type="OSI-SSEL">0001</P>
          <P type="OSI-TSEL">0001</P>
          <P type="IP">192.168.1.95</P>
          <P type="IP-SUBNET">255.255.255.0</P>
        </Address>
      </ConnectedAP>
    </SubNetwork>
  </Communication>
  <IED desc="Raspberry Pi Based PQ Monitor" manufacturer="M Hara" name="RasPQ" type="ARM Based Device">
    <Services>
      <DynAssociation />
```



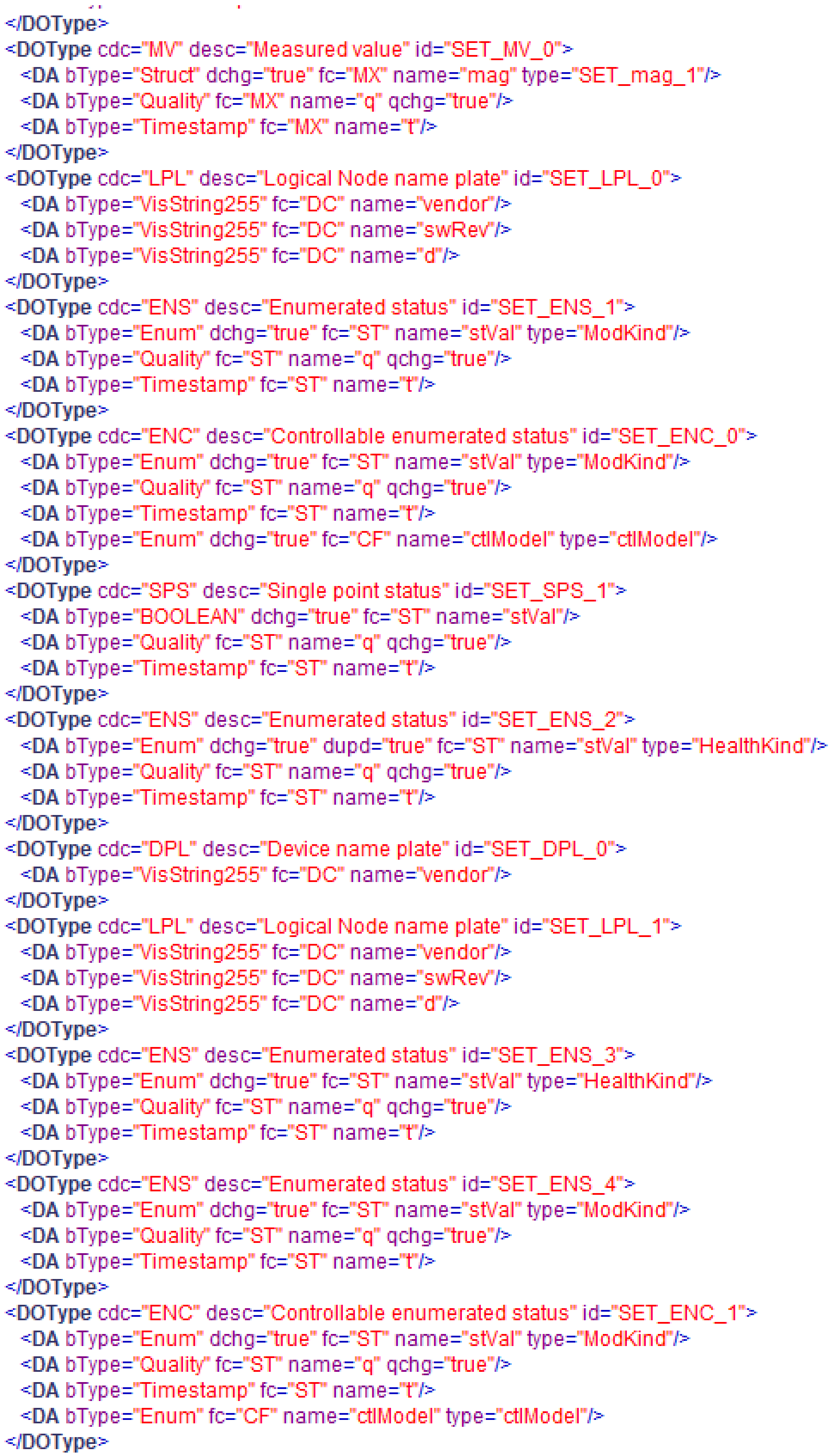
```
  <GetDirectory />
  <GetDataObjectDefinition />
  <GetDataSetValue />
  <DataSetDirectory />
  <ReadWrite />
  <GetCBValues />
  <ConfLNs fixPrefix="true" fixLnInst="true" />
  <GOOSE max="5" />
  <GSSE max="5" />
  <FileHandling />
  <GSEDir />
  <TimerActivatedControl />
</Services>
<AccessPoint name="MeterDevice">
  <Server>
    <Authentication />
    <LDevice inst="MET" desc="Data Sets">
      <LN0 desc="Logical node zero" inst="" lnClass="LLN0" lnType="SET_LLN0_0">
      <DataSet name="QVVR_Event" desc="QVVR_Event">
         <FCDA ldInst="MET" lnClass="QVVR" fc="ST" lnInst="1" doName="VarStr" daName="stVal" />
         <FCDA ldInst="MET" lnClass="QVVR" fc="ST" lnInst="1" doName="DipStr" daName="stVal" />
         <FCDA ldInst="MET" lnClass="QVVR" fc="ST" lnInst="1" doName="SwlStr" daName="stVal" />
      </DataSet>

      <DataSet name="MHAI_Measurement" desc="MHAI_Measurement">
         <FCDA ldInst="MET" lnClass="MHAI" fc="MX" lnInst="1" doName="ThdA" daName="phsA.cVal.mag.f" />
         <FCDA ldInst="MET" lnClass="MHAI" fc="MX" lnInst="1" doName="ThdA" daName="phsB.cVal.mag.f" />
         <FCDA ldInst="MET" lnClass="MHAI" fc="MX" lnInst="1" doName="ThdA" daName="phsC.cVal.mag.f" />
         <FCDA ldInst="MET" lnClass="MHAI" fc="MX" lnInst="1" doName="ThdPhV" daName="phsA.cVal.mag.f" />
         <FCDA ldInst="MET" lnClass="MHAI" fc="MX" lnInst="1" doName="ThdPhV" daName="phsB.cVal.mag.f" />
         <FCDA ldInst="MET" lnClass="MHAI" fc="MX" lnInst="1" doName="ThdPhV" daName="phsC.cVal.mag.f" />
      </DataSet>

      <DataSet name="MMXU_Measurement" desc="MMXU_Measurement">
         <FCDA ldInst="MET" lnClass="MMXU" fc="MX" lnInst="1" doName="A" daName="phsA.cVal.mag.f" />
         <FCDA ldInst="MET" lnClass="MMXU" fc="MX" lnInst="1" doName="A" daName="phsB.cVal.mag.f" />
         <FCDA ldInst="MET" lnClass="MMXU" fc="MX" lnInst="1" doName="A" daName="phsC.cVal.mag.f" />
         <FCDA ldInst="MET" lnClass="MMXU" fc="MX" lnInst="1" doName="PhV" daName="phsA.cVal.mag.f" />
         <FCDA ldInst="MET" lnClass="MMXU" fc="MX" lnInst="1" doName="PhV" daName="phsB.cVal.mag.f" />
         <FCDA ldInst="MET" lnClass="MMXU" fc="MX" lnInst="1" doName="PhV" daName="phsC.cVal.mag.f" />
         <FCDA ldInst="MET" lnClass="MMXU" fc="MX" lnInst="1" doName="PF" daName="phsA.cVal.mag.f" />
         <FCDA ldInst="MET" lnClass="MMXU" fc="MX" lnInst="1" doName="PF" daName="phsB.cVal.mag.f" />
         <FCDA ldInst="MET" lnClass="MMXU" fc="MX" lnInst="1" doName="PF" daName="phsC.cVal.mag.f" />
      </DataSet>

      <ReportControl name="QVVR_EventRCB" confRev="1" datSet="QVVR_Event" rptID="QVVR_Event" buffered="false" i
         <TrgOps period="true" />
         <OptFields seqNum="true" timeStamp="true" dataSet="true" reasonCode="true" entryID="true" configRef="true" />
         <RptEnabled max="1" />
      </ReportControl>
      <ReportControl name="MHAI_MeasurementRCB" confRev="1" datSet="MHAI_Measurement" rptID="MHAI_Measure
         <TrgOps period="true" />
         <OptFields seqNum="true" timeStamp="true" dataSet="true" reasonCode="true" entryID="true" configRef="true" />
         <RptEnabled max="1" />
      </ReportControl>
      <ReportControl name="MMXU_MeasurementRCB" confRev="1" datSet="MMXU_Measurement" rptID="MMXU_Meas
         <TrgOps period="true" />
```



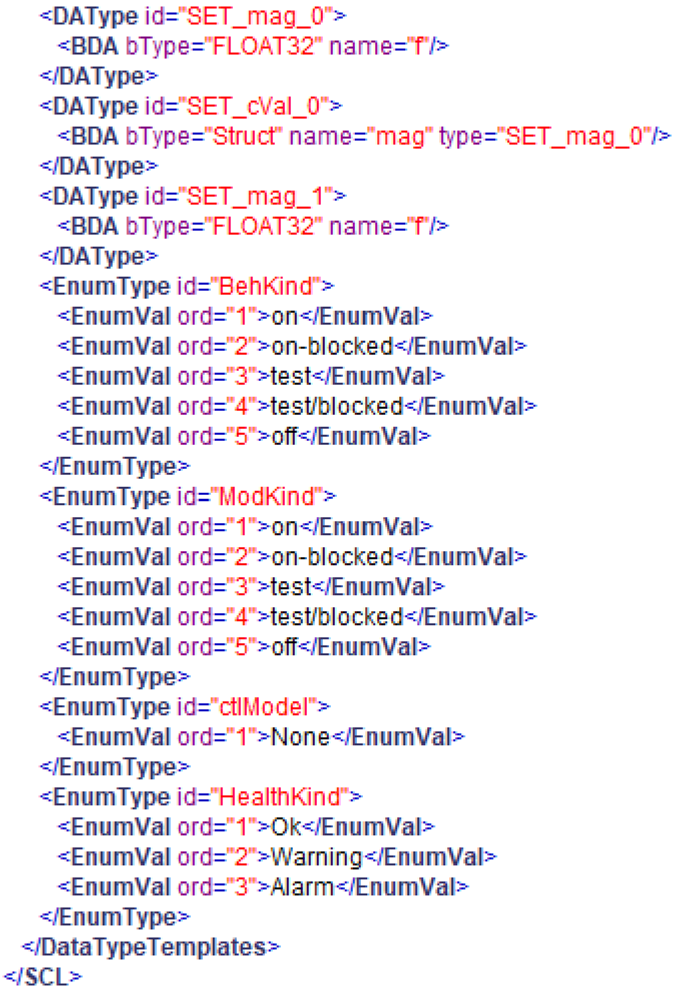
```
          <OptFields seqNum="true" timeStamp="true" dataSet="true" reasonCode="true" entryID="true" configRef="true" />
          <RptEnabled max="1" />
        </ReportControl>
        <GSEControl appID="events" name="gcbQVVREvent" type="GOOSE" datSet="QVVR_Event" confRev="2" minTime="1000" maxTi
        <GSEControl appID="THDanalog" name="gcbMHAIAnalogVal" type="GOOSE" datSet="MHAI_Measurement" confRev="2"/>
        <GSEControl appID="MMXUanalog" name="gcbMMXUAnalogVal" type="GOOSE" datSet="MMXU_Measurement" confRev="2"/>
        <DOI name="Mod">
          <DAI name="stVal">
            <Val>on</Val>
          </DAI>
        </DOI>
        <DOI name="Beh">
          <DAI name="stVal">
            <Val>on</Val>
          </DAI>
        </DOI>
        <DOI name="Health">
          <DAI name="stVal">
            <Val>Ok</Val>
          </DAI>
        </DOI>
        <DOI name="NamPlt">
          <DAI name="vendor">
            <Val>Mayamiko Hara</Val>
          </DAI>
          <DAI name="swRev">
            <Val>1.0.0</Val>
          </DAI>
          <DAI name="d">
            <Val>Power Qaulity Monitor</Val>
          </DAI>
        </DOI>
        </LN0>
        <LN lnClass="LPHD" lnType="SET_LPHD_0" inst="1" prefix="">
          <DOI name="PhyHealth">
          <DAI name="stVal">
            <Val>Ok</Val>
          </DAI>
          </DOI>
        </LN>
        <LN desc="Physical device information" inst="0" lnClass="LPHD" lnType="SET_LPHD_0" prefix=""/>
        <LN desc="Measurement" inst="1" lnClass="MMXU" lnType="SET_MMXU_0" prefix=""/>
        <LN desc="Voltage variation" inst="1" lnClass="QVVR" lnType="SET_QVVR_0" prefix=""/>
        <LN desc="Harmonics or interharmonics" inst="1" lnClass="MHAI" lnType="SET_MHAI_0" prefix=""/>
      </LDevice>
    </Server>
  </AccessPoint>
</IED>
<DataTypeTemplates>
  <LNodeType id="SET_MHAI_0" lnClass="MHAI">
    <DO desc="Enumerated status" name="Beh" type="SET_ENS_0"/>
    <DO desc="Total Harmonic Distortion: Phase Current" name="ThdA" type="SET_WYE_0"/>
    <DO desc="Total Harmonic Distortion: Phase Voltage" name="ThdPhV" type="SET_WYE_0"/>
  </LNodeType>
  <LNodeType id="SET_QVVR_0" lnClass="QVVR">
    <DO desc="Enumerated status" name="Beh" type="SET_ENS_0"/>
    <DO desc="Single point status - Variation Start" name="VarStr" type="SET_SPS_0"/>
```



```
  <DO desc="Single point status - Voltage Dip" name="DipStr" type="SET_SPS_0"/>
  <DO desc="Single point status - Voltage Swell" name="SwlStr" type="SET_SPS_0"/>
</LNodeType>
<LNodeType id="SET_MMXU_0" lnClass="MMXU">
  <DO desc="Controllable enumerated status" name="Mod" type="SET_ENC_0"/>
  <DO desc="Enumerated status" name="Beh" type="SET_ENS_1"/>
  <DO desc="Enumerated status" name="Health" type="SET_ENS_1"/>
  <DO desc="Logical Node name plate" name="NamPlt" type="SET_LPL_0"/>
  <DO desc="Active Power" name="W" type="SET_WYE_0"/>
  <DO desc="Reactive Power" name="VAr" type="SET_WYE_0"/>
  <DO desc="Apparent Power" name="VA" type="SET_WYE_0"/>
  <DO desc="Power Factor" name="PF" type="SET_WYE_0"/>
  <DO desc="Phase to ground Voltage" name="PhV" type="SET_WYE_0"/>
  <DO desc="Phase to ground Amps" name="A" type="SET_WYE_0"/>
</LNodeType>
<LNodeType id="SET_LPHD_0" lnClass="LPHD">
  <DO desc="Device name plate" name="PhyNam" type="SET_DPL_0"/>
  <DO desc="Enumerated status" name="PhyHealth" type="SET_ENS_2"/>
  <DO desc="Single point status" name="Proxy" type="SET_SPS_1"/>
</LNodeType>
<LNodeType id="SET_LLN0_0" lnClass="LLN0">
  <DO desc="Controllable enumerated status" name="Mod" type="SET_ENC_1"/>
  <DO desc="Enumerated status" name="Beh" type="SET_ENS_4"/>
  <DO desc="Enumerated status" name="Health" type="SET_ENS_3"/>
  <DO desc="Logical Node name plate" name="NamPlt" type="SET_LPL_1"/>
</LNodeType>
<DOType cdc="CMV" desc="Complex measured value" id="SET_phsC_0">
  <DA bType="Struct" dchg="true" dupd="true" fc="MX" name="cVal" type="SET_cVal_0"/>
  <DA bType="Quality" fc="MX" name="q" qchg="true"/>
  <DA bType="Timestamp" fc="MX" name="t"/>
</DOType>
<DOType cdc="CMV" desc="Complex measured value" id="SET_phsB_0">
  <DA bType="Struct" dchg="true" dupd="true" fc="MX" name="cVal" type="SET_cVal_0"/>
  <DA bType="Quality" fc="MX" name="q" qchg="true"/>
  <DA bType="Timestamp" fc="MX" name="t"/>
</DOType>
<DOType cdc="CMV" desc="Complex measured value" id="SET_phsA_0">
  <DA bType="Struct" dchg="true" dupd="true" fc="MX" name="cVal" type="SET_cVal_0"/>
  <DA bType="Quality" fc="MX" name="q" qchg="true"/>
  <DA bType="Timestamp" fc="MX" name="t"/>
</DOType>
<DOType cdc="WYE" desc="Phase to ground related measurement values of a three phase Star" id="SET_WYE_0">
  <SDO desc="Complex measured value" name="phsA" type="SET_phsA_0"/>
  <SDO desc="Complex measured value" name="phsB" type="SET_phsB_0"/>
  <SDO desc="Complex measured value" name="phsC" type="SET_phsC_0"/>
</DOType>
<DOType cdc="ENS" desc="Enumerated status" id="SET_ENS_0">
  <DA bType="Enum" dchg="true" dupd="true" fc="ST" name="stVal" type="BehKind"/>
  <DA bType="Quality" fc="ST" name="q" qchg="true"/>
  <DA bType="Timestamp" fc="ST" name="t"/>
</DOType>
<DOType cdc="SPS" desc="Single point status" id="SET_SPS_0">
  <DA bType="BOOLEAN" dchg="true" fc="ST" name="stVal"/>
  <DA bType="Quality" fc="ST" name="q" qchg="true"/>
  <DA bType="Timestamp" fc="ST" name="t"/>
```



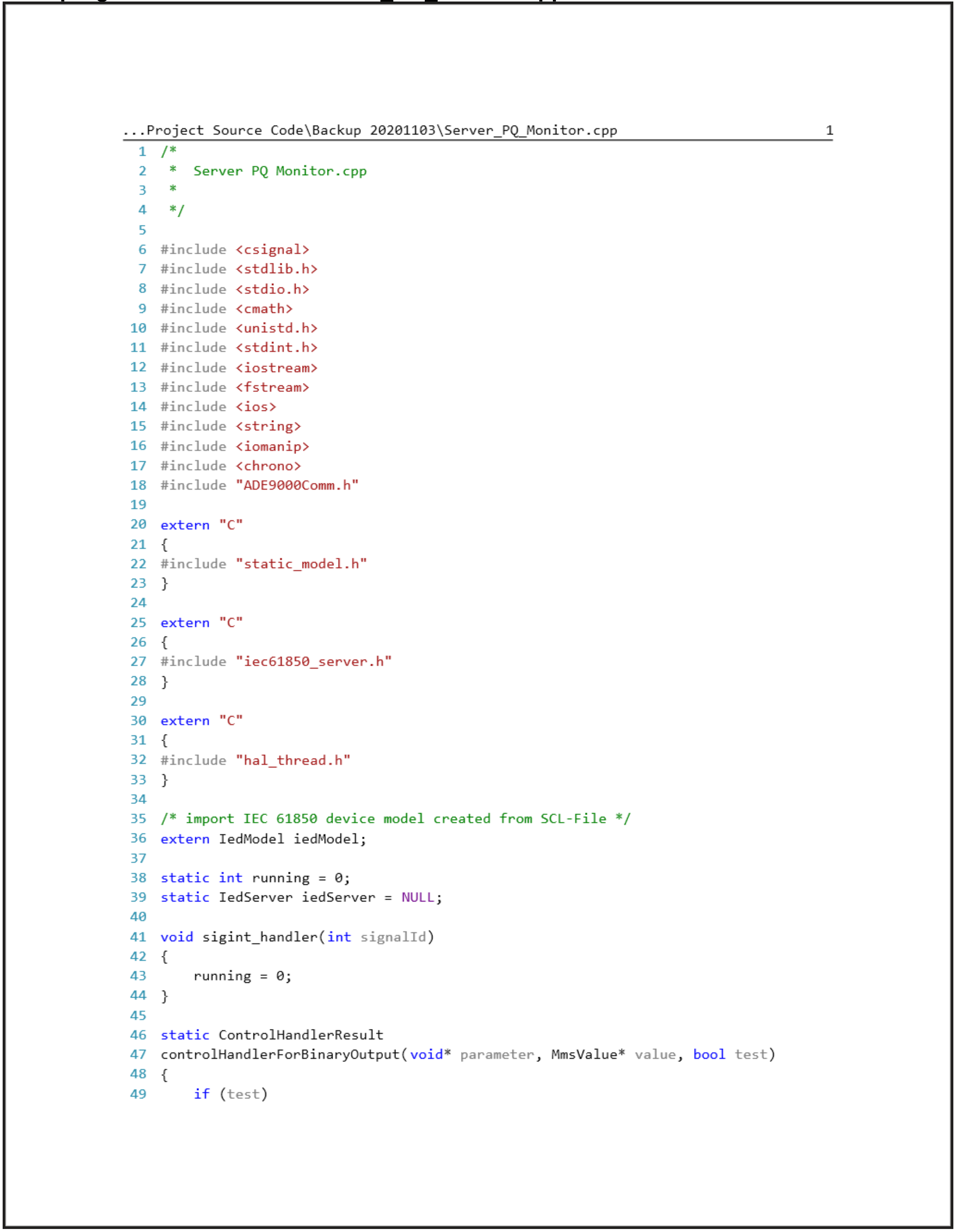
```
</DOType>
<DOType cdc="MV" desc="Measured value" id="SET_MV_0">
  <DA bType="Struct" dchg="true" fc="MX" name="mag" type="SET_mag_1"/>
  <DA bType="Quality" fc="MX" name="q" qchg="true"/>
  <DA bType="Timestamp" fc="MX" name="t"/>
</DOType>
<DOType cdc="LPL" desc="Logical Node name plate" id="SET_LPL_0">
  <DA bType="VisString255" fc="DC" name="vendor"/>
  <DA bType="VisString255" fc="DC" name="swRev"/>
  <DA bType="VisString255" fc="DC" name="d"/>
</DOType>
<DOType cdc="ENS" desc="Enumerated status" id="SET_ENS_1">
  <DA bType="Enum" dchg="true" fc="ST" name="stVal" type="ModKind"/>
  <DA bType="Quality" fc="ST" name="q" qchg="true"/>
  <DA bType="Timestamp" fc="ST" name="t"/>
</DOType>
<DOType cdc="ENC" desc="Controllable enumerated status" id="SET_ENC_0">
  <DA bType="Enum" dchg="true" fc="ST" name="stVal" type="ModKind"/>
  <DA bType="Quality" fc="ST" name="q" qchg="true"/>
  <DA bType="Timestamp" fc="ST" name="t"/>
  <DA bType="Enum" dchg="true" fc="CF" name="ctlModel" type="ctlModel"/>
</DOType>
<DOType cdc="SPS" desc="Single point status" id="SET_SPS_1">
  <DA bType="BOOLEAN" dchg="true" fc="ST" name="stVal"/>
  <DA bType="Quality" fc="ST" name="q" qchg="true"/>
  <DA bType="Timestamp" fc="ST" name="t"/>
</DOType>
<DOType cdc="ENS" desc="Enumerated status" id="SET_ENS_2">
  <DA bType="Enum" dchg="true" dupd="true" fc="ST" name="stVal" type="HealthKind"/>
  <DA bType="Quality" fc="ST" name="q" qchg="true"/>
  <DA bType="Timestamp" fc="ST" name="t"/>
</DOType>
<DOType cdc="DPL" desc="Device name plate" id="SET_DPL_0">
  <DA bType="VisString255" fc="DC" name="vendor"/>
</DOType>
<DOType cdc="LPL" desc="Logical Node name plate" id="SET_LPL_1">
  <DA bType="VisString255" fc="DC" name="vendor"/>
  <DA bType="VisString255" fc="DC" name="swRev"/>
  <DA bType="VisString255" fc="DC" name="d"/>
</DOType>
<DOType cdc="ENS" desc="Enumerated status" id="SET_ENS_3">
  <DA bType="Enum" dchg="true" fc="ST" name="stVal" type="HealthKind"/>
  <DA bType="Quality" fc="ST" name="q" qchg="true"/>
  <DA bType="Timestamp" fc="ST" name="t"/>
</DOType>
<DOType cdc="ENS" desc="Enumerated status" id="SET_ENS_4">
  <DA bType="Enum" dchg="true" fc="ST" name="stVal" type="ModKind"/>
  <DA bType="Quality" fc="ST" name="q" qchg="true"/>
  <DA bType="Timestamp" fc="ST" name="t"/>
</DOType>
<DOType cdc="ENC" desc="Controllable enumerated status" id="SET_ENC_1">
  <DA bType="Enum" dchg="true" fc="ST" name="stVal" type="ModKind"/>
  <DA bType="Quality" fc="ST" name="q" qchg="true"/>
  <DA bType="Timestamp" fc="ST" name="t"/>
  <DA bType="Enum" fc="CF" name="ctlModel" type="ctlModel"/>
</DOType>
```



```
    <DAType id="SET_mag_0">
      <BDA bType="FLOAT32" name="f"/>
    </DAType>
    <DAType id="SET_cVal_0">
      <BDA bType="Struct" name="mag" type="SET_mag_0"/>
    </DAType>
    <DAType id="SET_mag_1">
      <BDA bType="FLOAT32" name="f"/>
    </DAType>
    <EnumType id="BehKind">
      <EnumVal ord="1">on</EnumVal>
      <EnumVal ord="2">on-blocked</EnumVal>
      <EnumVal ord="3">test</EnumVal>
      <EnumVal ord="4">test/blocked</EnumVal>
      <EnumVal ord="5">off</EnumVal>
    </EnumType>
    <EnumType id="ModKind">
      <EnumVal ord="1">on</EnumVal>
      <EnumVal ord="2">on-blocked</EnumVal>
      <EnumVal ord="3">test</EnumVal>
      <EnumVal ord="4">test/blocked</EnumVal>
      <EnumVal ord="5">off</EnumVal>
    </EnumType>
    <EnumType id="ctlModel">
      <EnumVal ord="1">None</EnumVal>
    </EnumType>
    <EnumType id="HealthKind">
      <EnumVal ord="1">Ok</EnumVal>
      <EnumVal ord="2">Warning</EnumVal>
      <EnumVal ord="3">Alarm</EnumVal>
    </EnumType>
  </DataTypeTemplates>
</SCL>
```

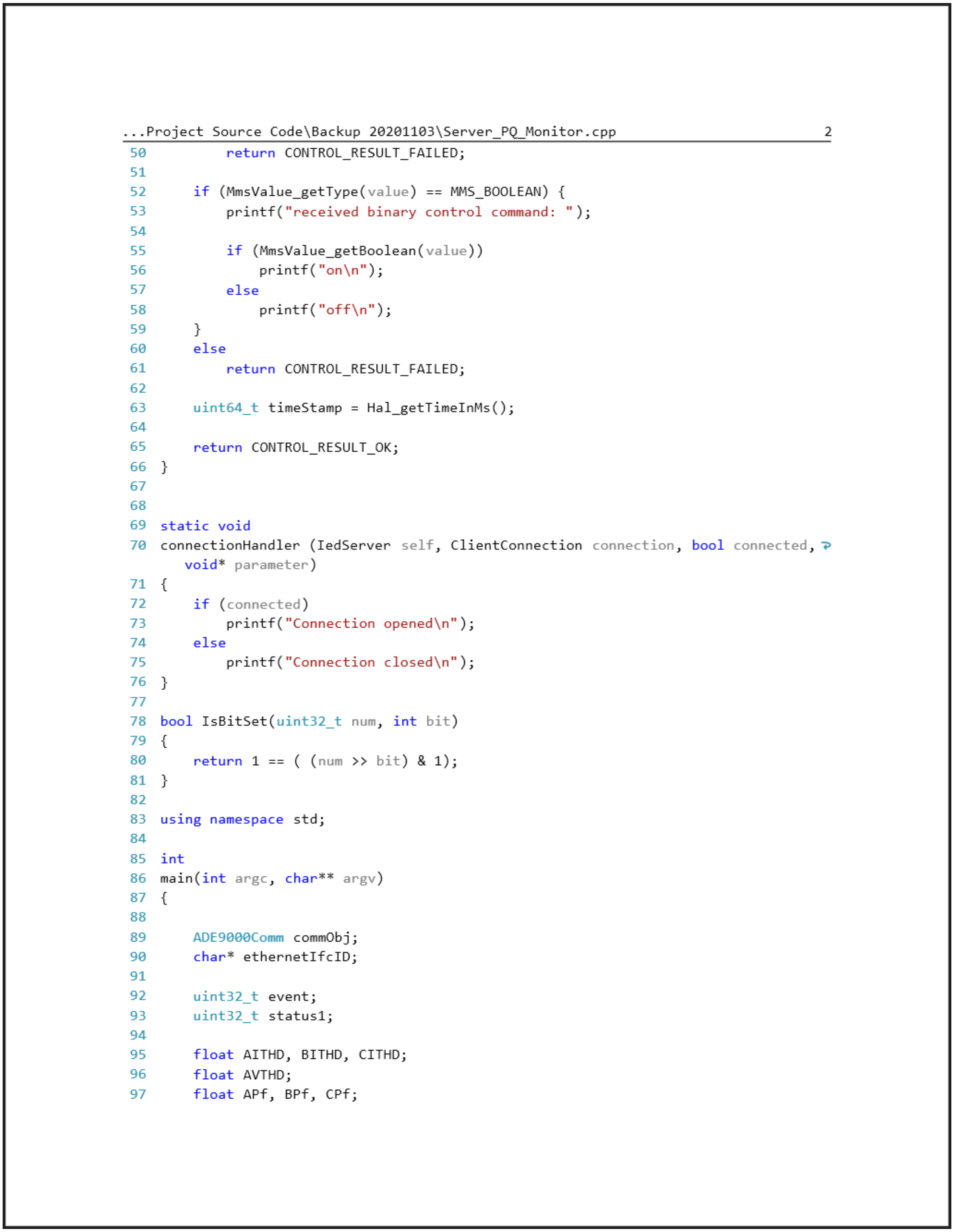
Appendix D: Raspberry Pi data acquisition and IEC 61850 standard communications source code

Main program source code – Server_PQ_Monitor.cpp



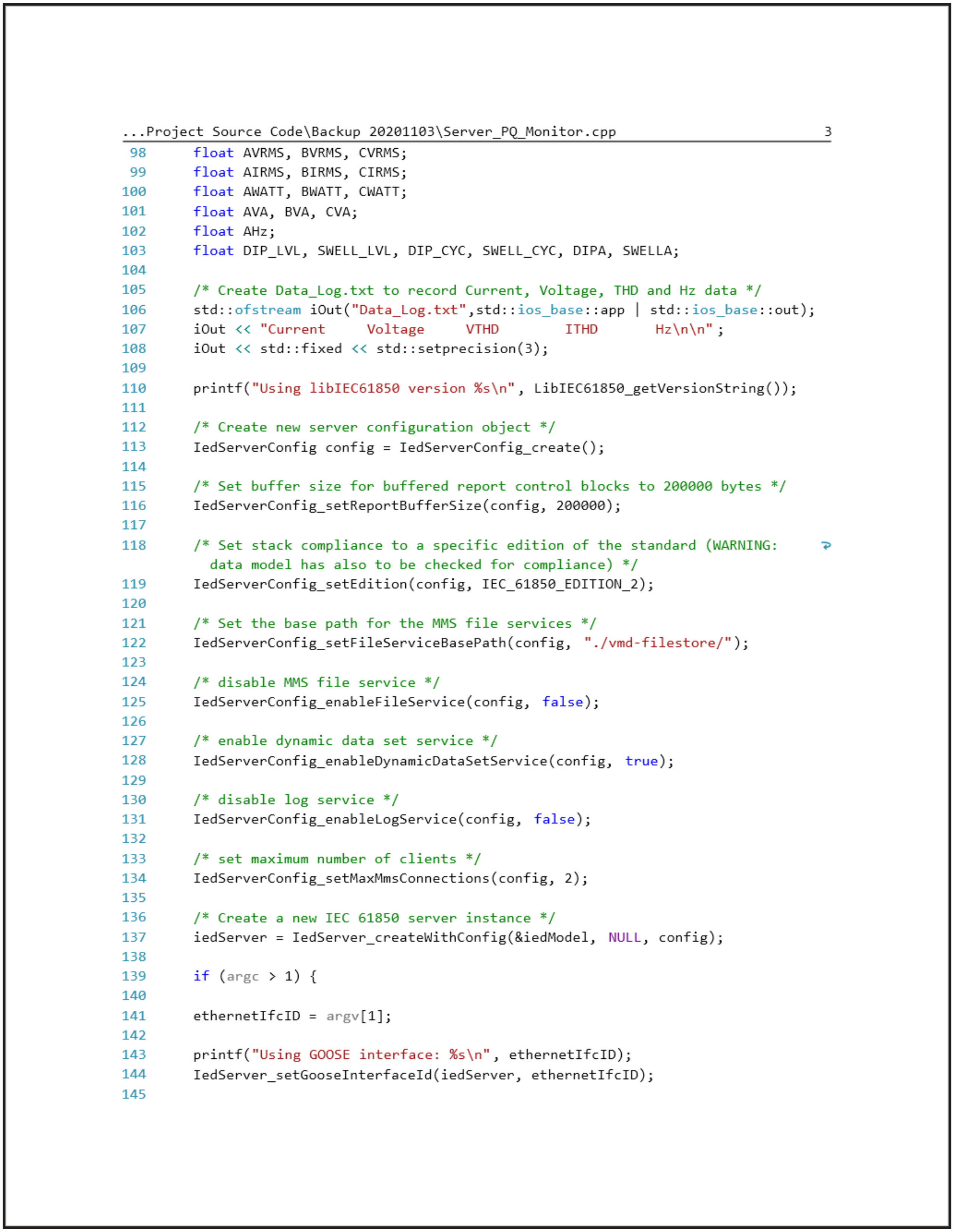
...Project Source Code\Backup 20201103\Server_PQ_Monitor.cpp 1

```
/*
 *  Server PQ Monitor.cpp
 *
 */

#include <csignal>
#include <stdlib.h>
#include <stdio.h>
#include <cmath>
#include <unistd.h>
#include <stdint.h>
#include <iostream>
#include <fstream>
#include <ios>
#include <string>
#include <iomanip>
#include <chrono>
#include "ADE9000Comm.h"

extern "C"
{
#include "static_model.h"
}

extern "C"
{
#include "iec61850_server.h"
}

extern "C"
{
#include "hal_thread.h"
}

/* import IEC 61850 device model created from SCL-File */
extern IedModel iedModel;

static int running = 0;
static IedServer iedServer = NULL;

void sigint_handler(int signalId)
{
    running = 0;
}

static ControlHandlerResult
controlHandlerForBinaryOutput(void* parameter, MmsValue* value, bool test)
{
    if (test)
```



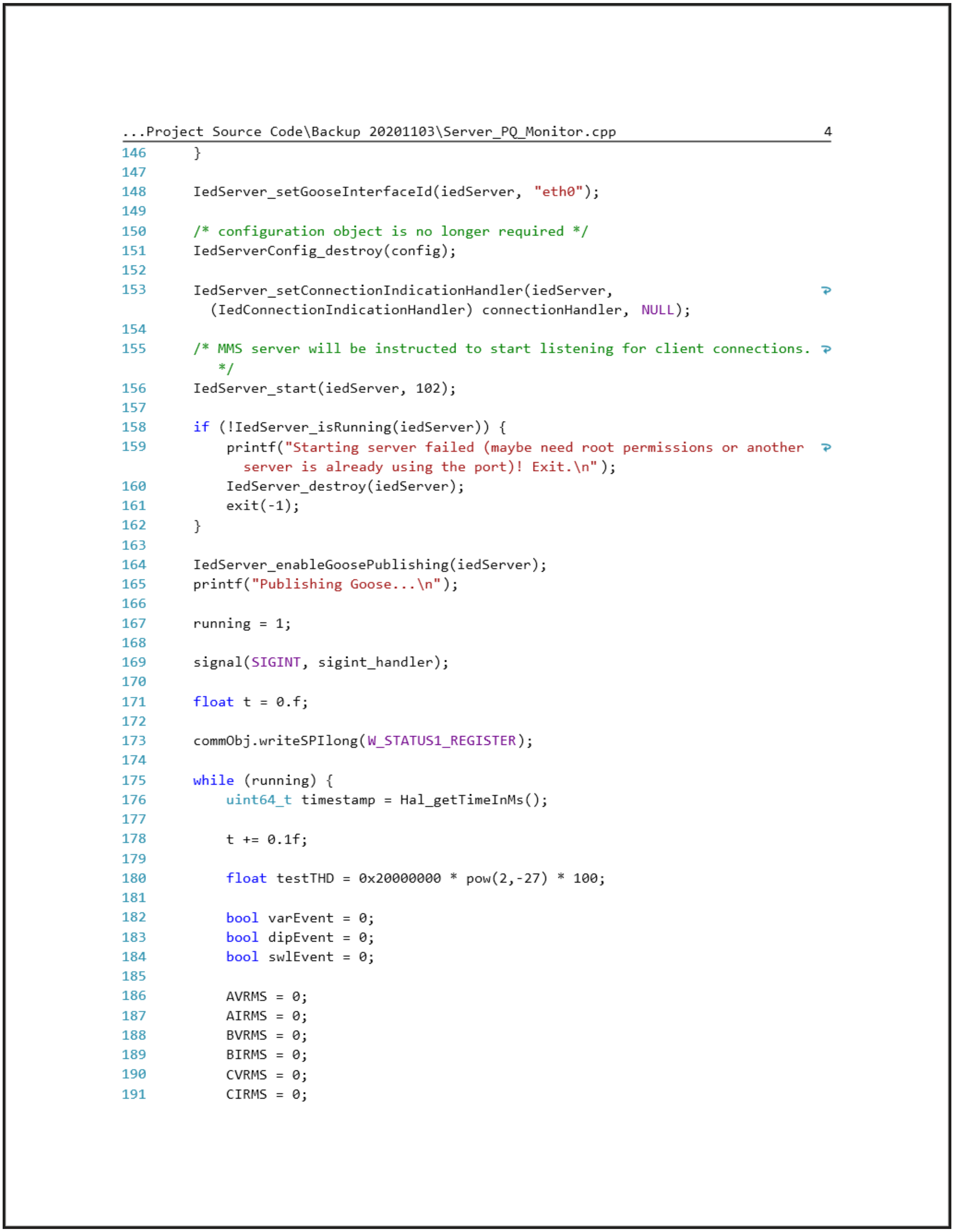
...Project Source Code\Backup 20201103\Server_PQ_Monitor.cpp 2

```cpp
        return CONTROL_RESULT_FAILED;

    if (MmsValue_getType(value) == MMS_BOOLEAN) {
        printf("received binary control command: ");

        if (MmsValue_getBoolean(value))
            printf("on\n");
        else
            printf("off\n");
    }
    else
        return CONTROL_RESULT_FAILED;

    uint64_t timeStamp = Hal_getTimeInMs();

    return CONTROL_RESULT_OK;
}


static void
connectionHandler (IedServer self, ClientConnection connection, bool connected, void* parameter)
{
    if (connected)
        printf("Connection opened\n");
    else
        printf("Connection closed\n");
}

bool IsBitSet(uint32_t num, int bit)
{
    return 1 == ( (num >> bit) & 1);
}

using namespace std;

int
main(int argc, char** argv)
{

    ADE9000Comm commObj;
    char* ethernetIfcID;

    uint32_t event;
    uint32_t status1;

    float AITHD, BITHD, CITHD;
    float AVTHD;
    float APf, BPf, CPf;
```

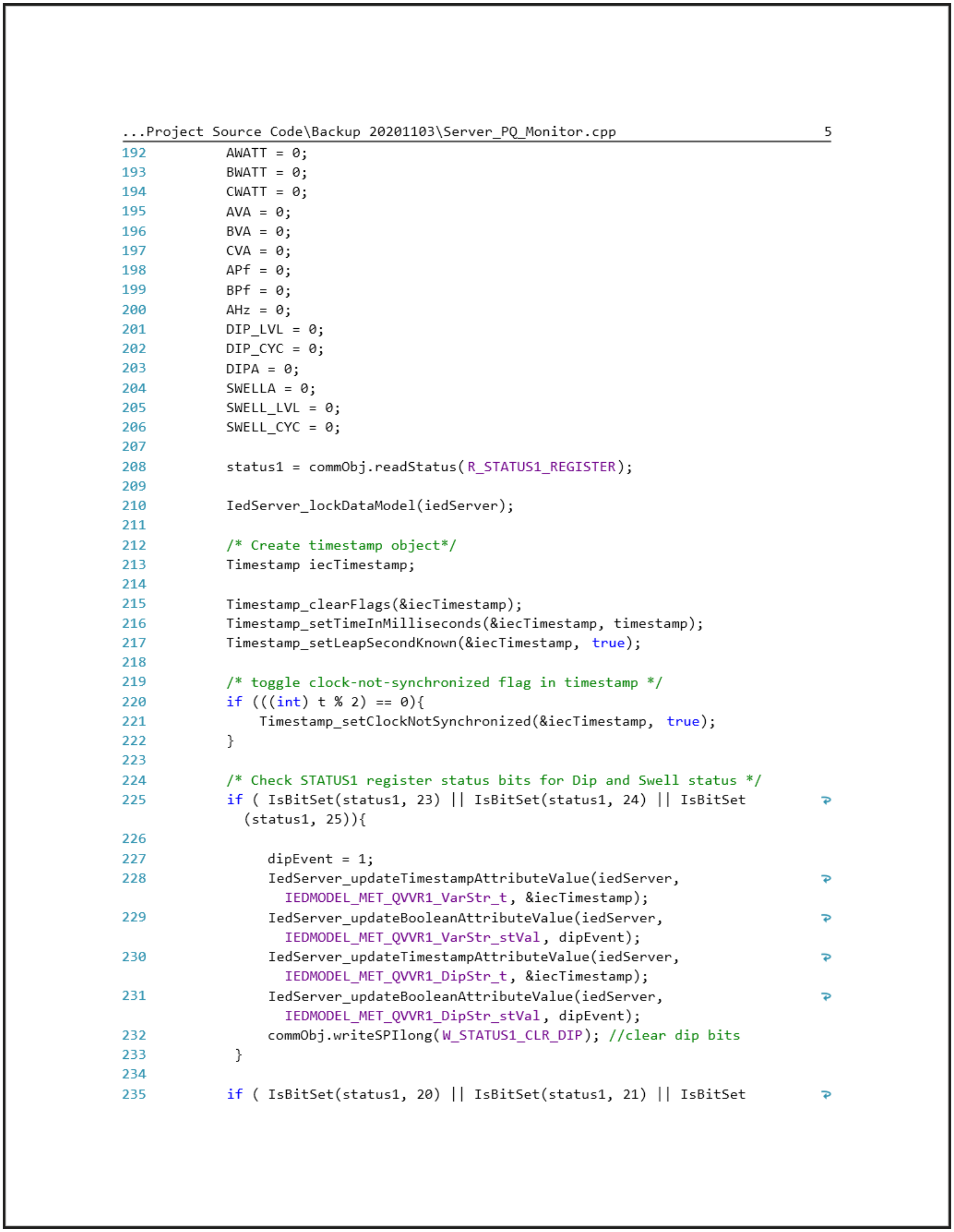


...Project Source Code\Backup 20201103\Server_PQ_Monitor.cpp 3

```cpp
    float AVRMS, BVRMS, CVRMS;
    float AIRMS, BIRMS, CIRMS;
    float AWATT, BWATT, CWATT;
    float AVA, BVA, CVA;
    float AHz;
    float DIP_LVL, SWELL_LVL, DIP_CYC, SWELL_CYC, DIPA, SWELLA;

    /* Create Data_Log.txt to record Current, Voltage, THD and Hz data */
    std::ofstream iOut("Data_Log.txt",std::ios_base::app | std::ios_base::out);
    iOut << "Current     Voltage     VTHD        ITHD        Hz\n\n" ;
    iOut << std::fixed << std::setprecision(3);

    printf("Using libIEC61850 version %s\n", LibIEC61850_getVersionString());

    /* Create new server configuration object */
    IedServerConfig config = IedServerConfig_create();

    /* Set buffer size for buffered report control blocks to 200000 bytes */
    IedServerConfig_setReportBufferSize(config, 200000);

    /* Set stack compliance to a specific edition of the standard (WARNING: data model has also to be checked for compliance) */
    IedServerConfig_setEdition(config, IEC_61850_EDITION_2);

    /* Set the base path for the MMS file services */
    IedServerConfig_setFileServiceBasePath(config, "./vmd-filestore/");

    /* disable MMS file service */
    IedServerConfig_enableFileService(config, false);

    /* enable dynamic data set service */
    IedServerConfig_enableDynamicDataSetService(config, true);

    /* disable log service */
    IedServerConfig_enableLogService(config, false);

    /* set maximum number of clients */
    IedServerConfig_setMaxMmsConnections(config, 2);

    /* Create a new IEC 61850 server instance */
    iedServer = IedServer_createWithConfig(&iedModel, NULL, config);

    if (argc > 1) {

    ethernetIfcID = argv[1];

    printf("Using GOOSE interface: %s\n", ethernetIfcID);
    IedServer_setGooseInterfaceId(iedServer, ethernetIfcID);

```

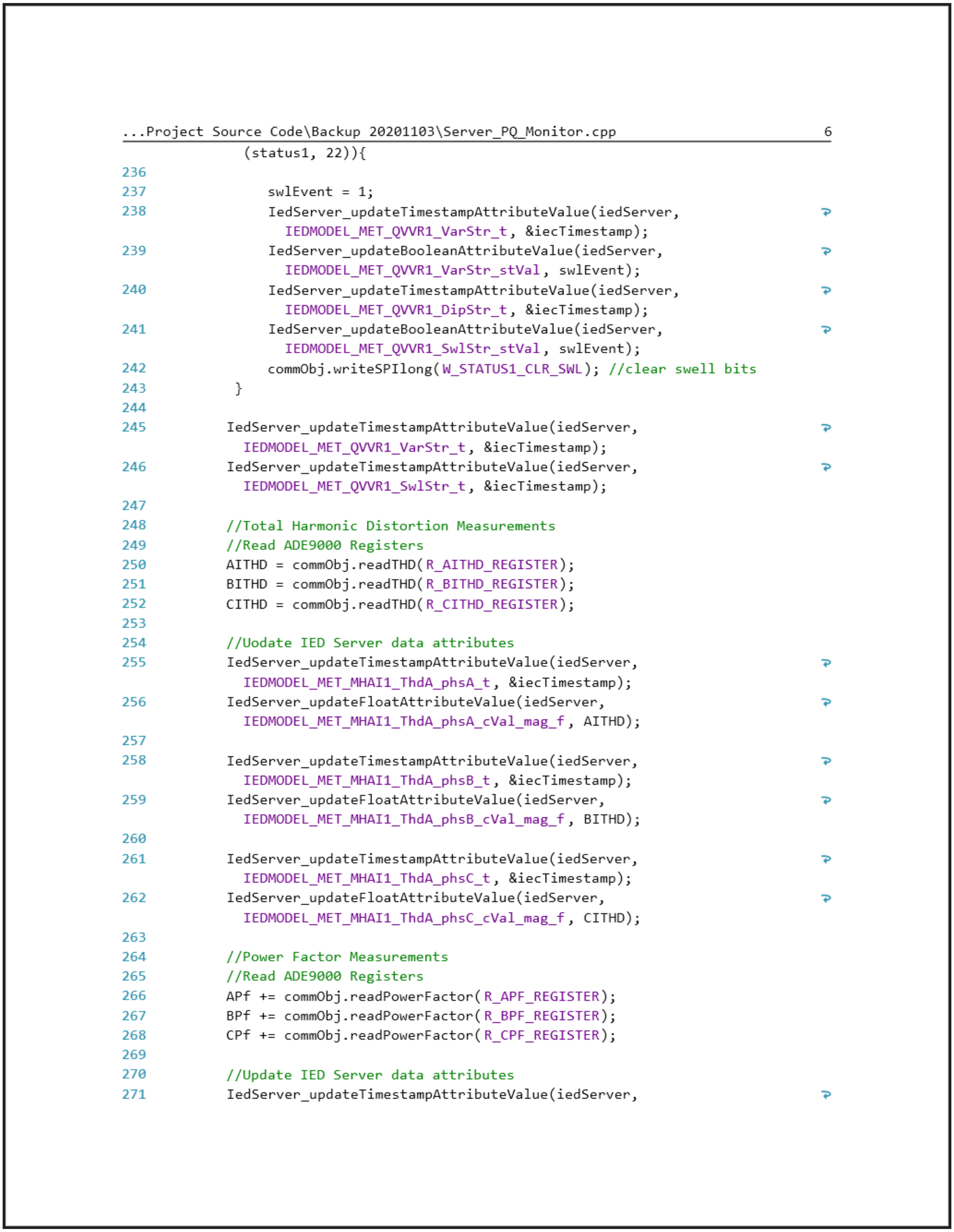


```
    }

    IedServer_setGooseInterfaceId(iedServer, "eth0");

    /* configuration object is no longer required */
    IedServerConfig_destroy(config);

    IedServer_setConnectionIndicationHandler(iedServer, (IedConnectionIndicationHandler) connectionHandler, NULL);

    /* MMS server will be instructed to start listening for client connections. */
    IedServer_start(iedServer, 102);

    if (!IedServer_isRunning(iedServer)) {
        printf("Starting server failed (maybe need root permissions or another server is already using the port)! Exit.\n");
        IedServer_destroy(iedServer);
        exit(-1);
    }

    IedServer_enableGoosePublishing(iedServer);
    printf("Publishing Goose...\n");

    running = 1;

    signal(SIGINT, sigint_handler);

    float t = 0.f;

    commObj.writeSPIlong(W_STATUS1_REGISTER);

    while (running) {
        uint64_t timestamp = Hal_getTimeInMs();

        t += 0.1f;

        float testTHD = 0x20000000 * pow(2,-27) * 100;

        bool varEvent = 0;
        bool dipEvent = 0;
        bool swlEvent = 0;

        AVRMS = 0;
        AIRMS = 0;
        BVRMS = 0;
        BIRMS = 0;
        CVRMS = 0;
        CIRMS = 0;
```

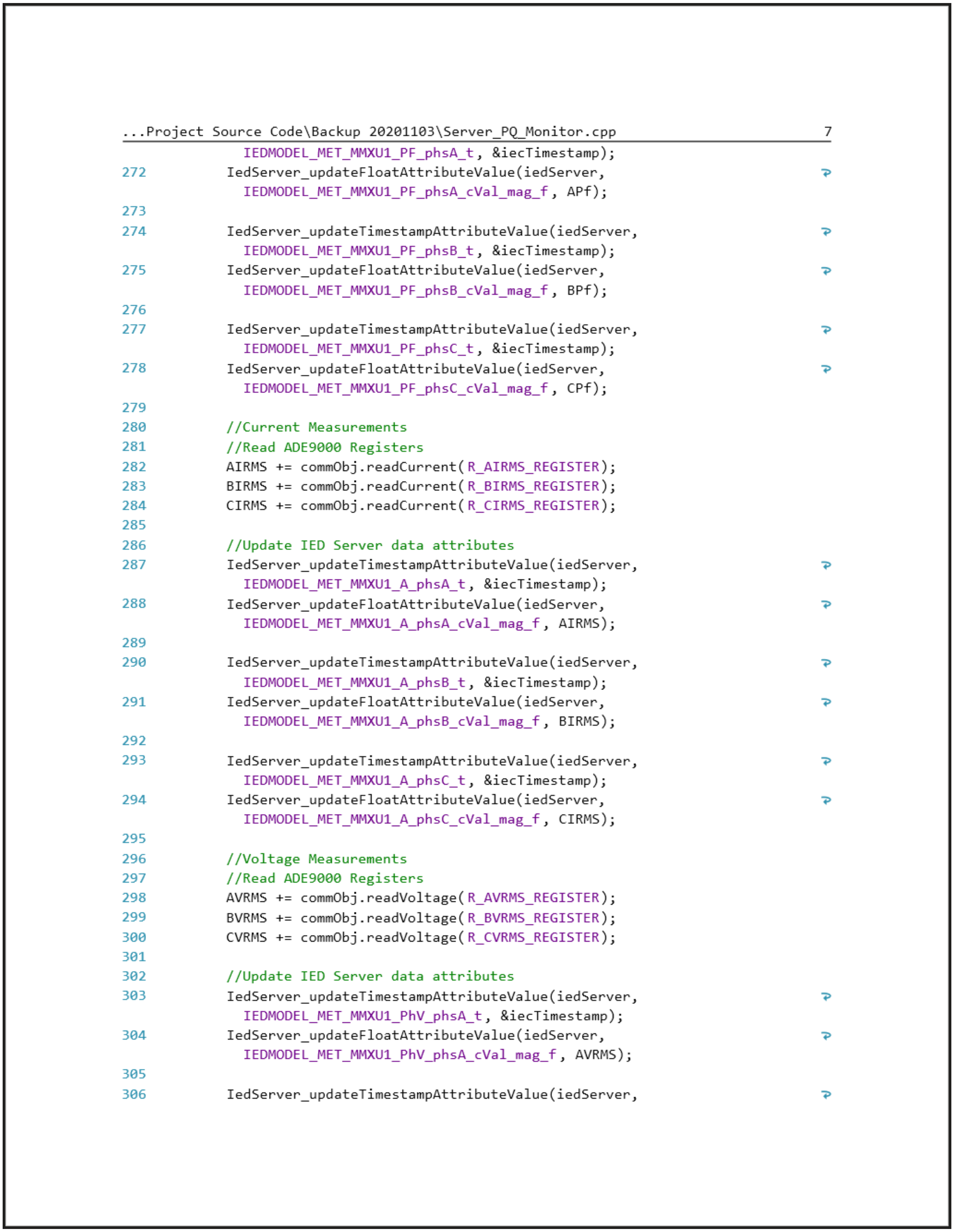


...Project Source Code\Backup 20201103\Server_PQ_Monitor.cpp 5

```
    AWATT = 0;
    BWATT = 0;
    CWATT = 0;
    AVA = 0;
    BVA = 0;
    CVA = 0;
    APf = 0;
    BPf = 0;
    AHz = 0;
    DIP_LVL = 0;
    DIP_CYC = 0;
    DIPA = 0;
    SWELLA = 0;
    SWELL_LVL = 0;
    SWELL_CYC = 0;

    status1 = commObj.readStatus(R_STATUS1_REGISTER);

    IedServer_lockDataModel(iedServer);

    /* Create timestamp object*/
    Timestamp iecTimestamp;

    Timestamp_clearFlags(&iecTimestamp);
    Timestamp_setTimeInMilliseconds(&iecTimestamp, timestamp);
    Timestamp_setLeapSecondKnown(&iecTimestamp, true);

    /* toggle clock-not-synchronized flag in timestamp */
    if (((int) t % 2) == 0){
        Timestamp_setClockNotSynchronized(&iecTimestamp, true);
    }

    /* Check STATUS1 register status bits for Dip and Swell status */
    if ( IsBitSet(status1, 23) || IsBitSet(status1, 24) || IsBitSet(status1, 25)){

         dipEvent = 1;
         IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_MET_QVVR1_VarStr_t, &iecTimestamp);
         IedServer_updateBooleanAttributeValue(iedServer, IEDMODEL_MET_QVVR1_VarStr_stVal, dipEvent);
         IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_MET_QVVR1_DipStr_t, &iecTimestamp);
         IedServer_updateBooleanAttributeValue(iedServer, IEDMODEL_MET_QVVR1_DipStr_stVal, dipEvent);
         commObj.writeSPIlong(W_STATUS1_CLR_DIP); //clear dip bits
     }

    if ( IsBitSet(status1, 20) || IsBitSet(status1, 21) || IsBitSet
```

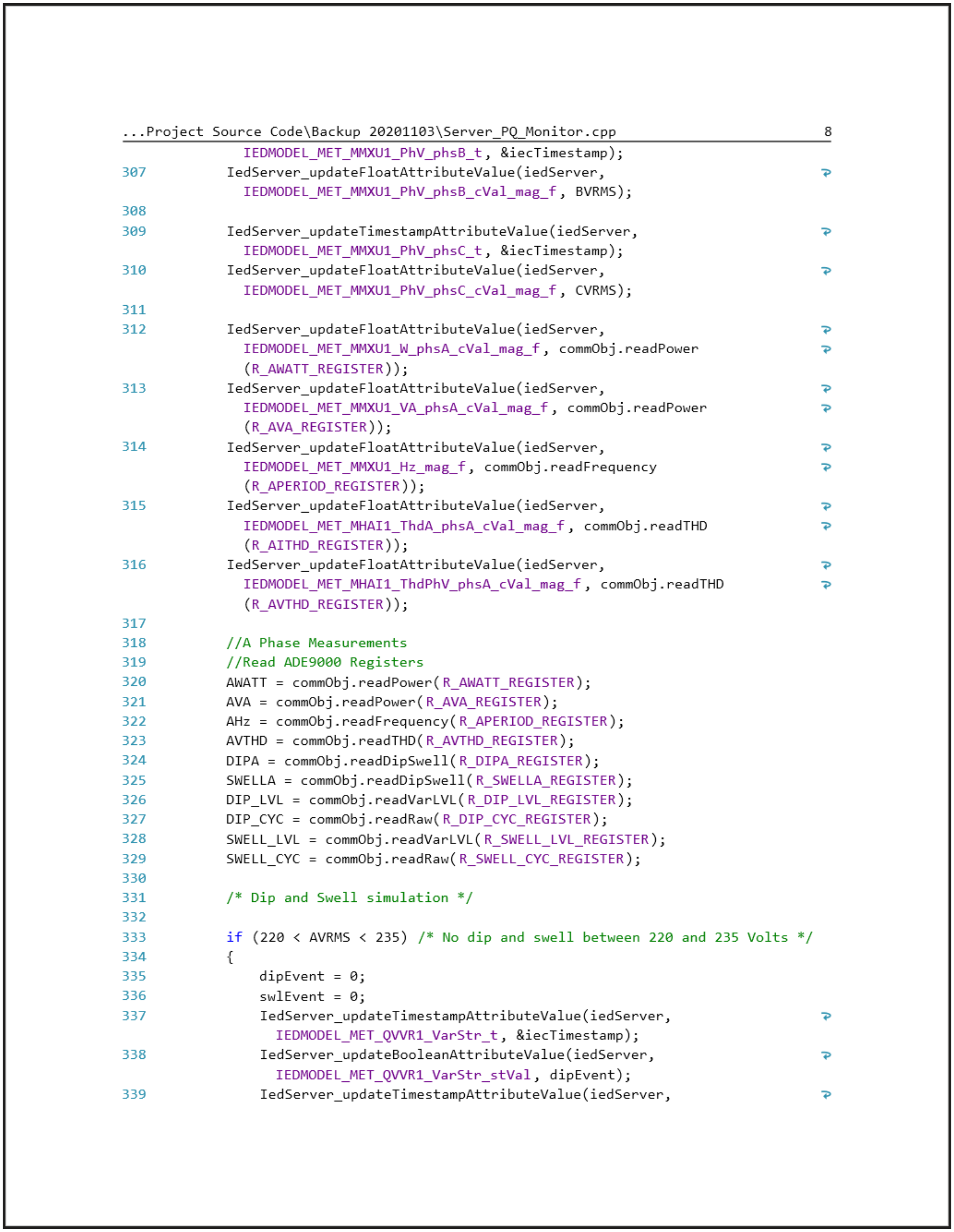


```
          (status1, 22)){

            swlEvent = 1;
            IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_MET_QVVR1_VarStr_t, &iecTimestamp);
            IedServer_updateBooleanAttributeValue(iedServer, IEDMODEL_MET_QVVR1_VarStr_stVal, swlEvent);
            IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_MET_QVVR1_DipStr_t, &iecTimestamp);
            IedServer_updateBooleanAttributeValue(iedServer, IEDMODEL_MET_QVVR1_SwlStr_stVal, swlEvent);
            commObj.writeSPIlong(W_STATUS1_CLR_SWL); //clear swell bits
         }

        IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_MET_QVVR1_VarStr_t, &iecTimestamp);
        IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_MET_QVVR1_SwlStr_t, &iecTimestamp);

        //Total Harmonic Distortion Measurements
        //Read ADE9000 Registers
        AITHD = commObj.readTHD(R_AITHD_REGISTER);
        BITHD = commObj.readTHD(R_BITHD_REGISTER);
        CITHD = commObj.readTHD(R_CITHD_REGISTER);

        //Uodate IED Server data attributes
        IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_MET_MHAI1_ThdA_phsA_t, &iecTimestamp);
        IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_MET_MHAI1_ThdA_phsA_cVal_mag_f, AITHD);

        IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_MET_MHAI1_ThdA_phsB_t, &iecTimestamp);
        IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_MET_MHAI1_ThdA_phsB_cVal_mag_f, BITHD);

        IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_MET_MHAI1_ThdA_phsC_t, &iecTimestamp);
        IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_MET_MHAI1_ThdA_phsC_cVal_mag_f, CITHD);

        //Power Factor Measurements
        //Read ADE9000 Registers
        APf += commObj.readPowerFactor(R_APF_REGISTER);
        BPf += commObj.readPowerFactor(R_BPF_REGISTER);
        CPf += commObj.readPowerFactor(R_CPF_REGISTER);

        //Update IED Server data attributes
        IedServer_updateTimestampAttributeValue(iedServer,
```

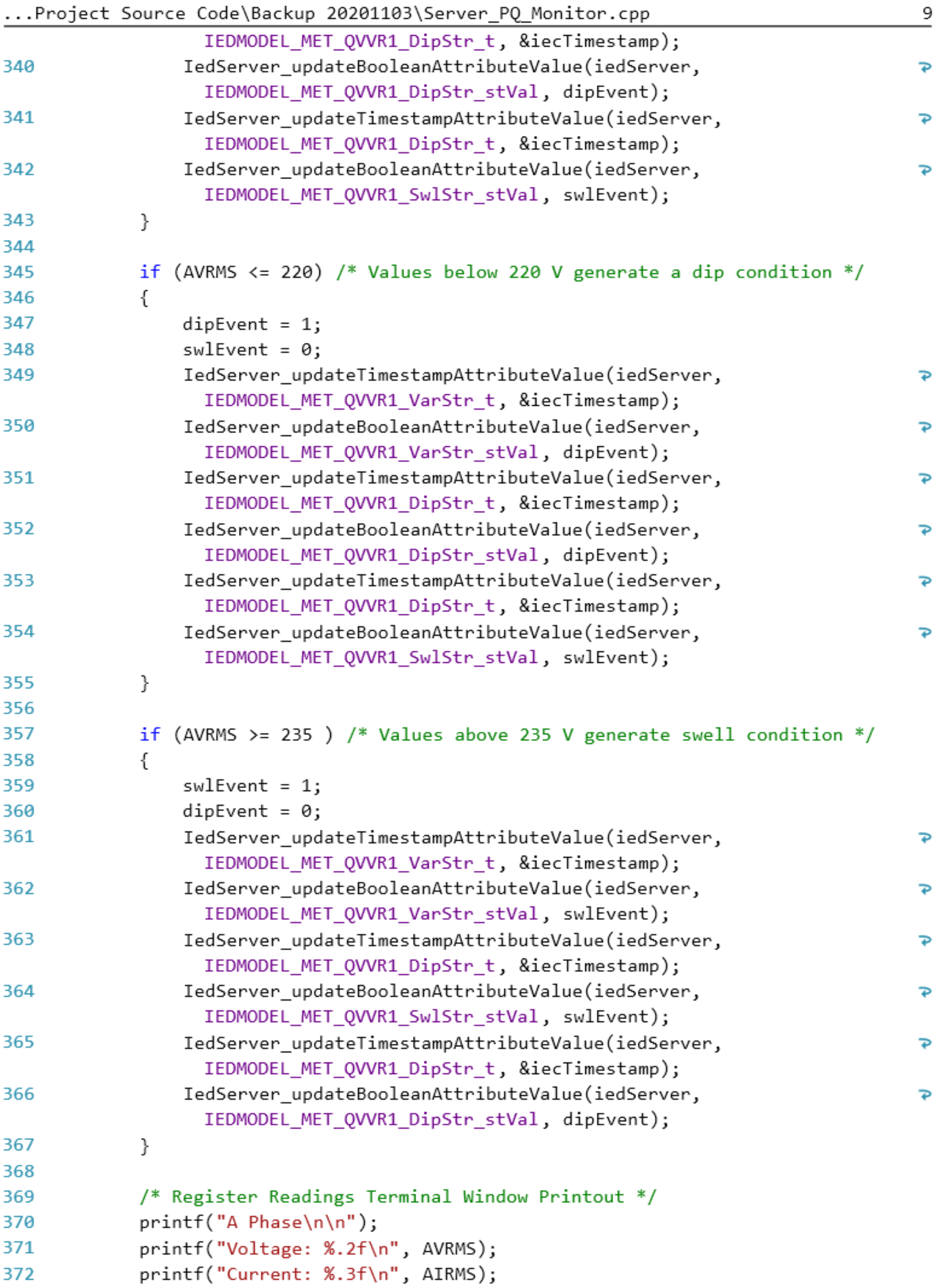


...Project Source Code\Backup 20201103\Server_PQ_Monitor.cpp 7

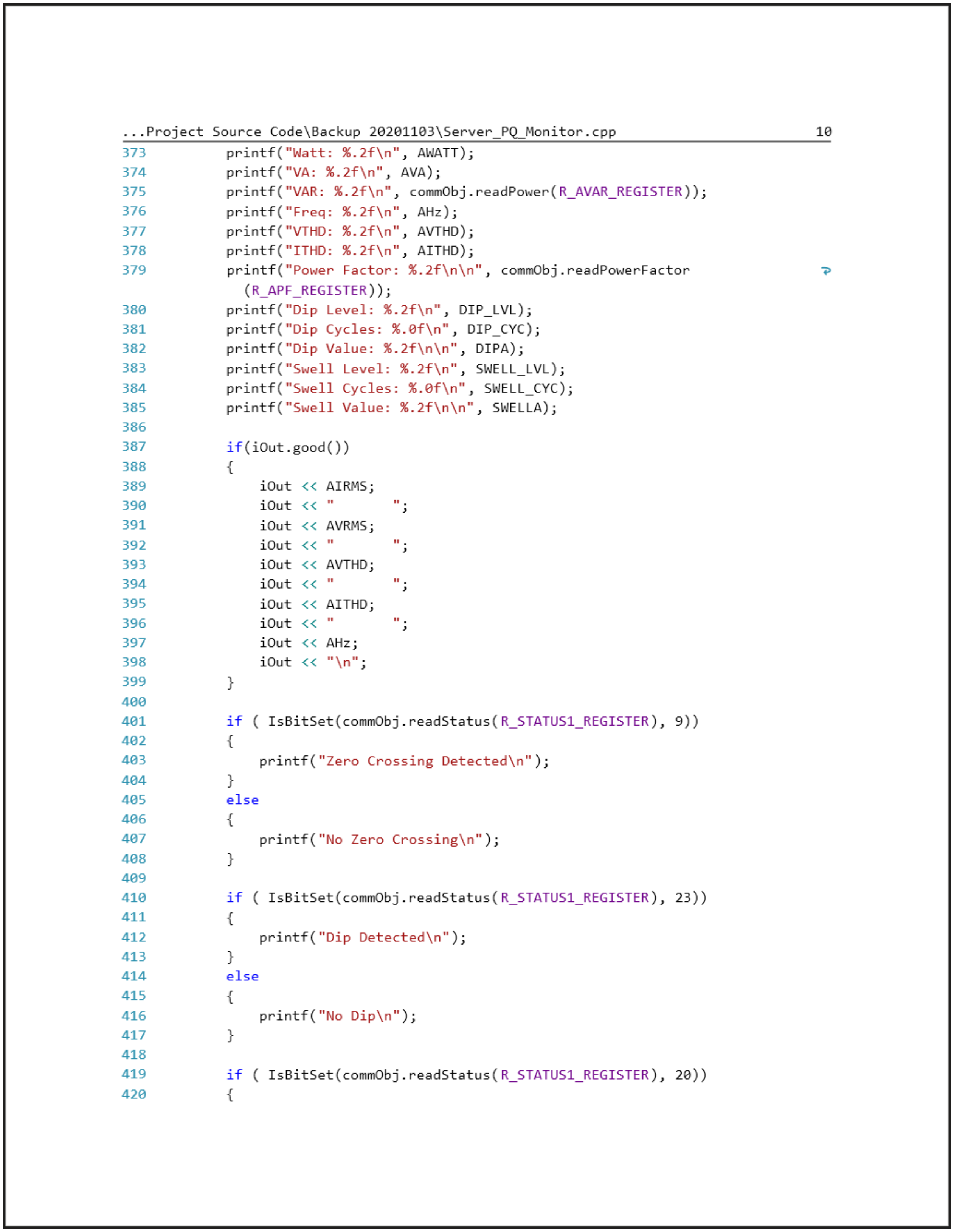
```
  IEDMODEL_MET_MMXU1_PF_phsA_t, &iecTimestamp);
IedServer_updateFloatAttributeValue(iedServer,
  IEDMODEL_MET_MMXU1_PF_phsA_cVal_mag_f, APf);

IedServer_updateTimestampAttributeValue(iedServer,
  IEDMODEL_MET_MMXU1_PF_phsB_t, &iecTimestamp);
IedServer_updateFloatAttributeValue(iedServer,
  IEDMODEL_MET_MMXU1_PF_phsB_cVal_mag_f, BPf);

IedServer_updateTimestampAttributeValue(iedServer,
  IEDMODEL_MET_MMXU1_PF_phsC_t, &iecTimestamp);
IedServer_updateFloatAttributeValue(iedServer,
  IEDMODEL_MET_MMXU1_PF_phsC_cVal_mag_f, CPf);

//Current Measurements
//Read ADE9000 Registers
AIRMS += commObj.readCurrent(R_AIRMS_REGISTER);
BIRMS += commObj.readCurrent(R_BIRMS_REGISTER);
CIRMS += commObj.readCurrent(R_CIRMS_REGISTER);

//Update IED Server data attributes
IedServer_updateTimestampAttributeValue(iedServer,
  IEDMODEL_MET_MMXU1_A_phsA_t, &iecTimestamp);
IedServer_updateFloatAttributeValue(iedServer,
  IEDMODEL_MET_MMXU1_A_phsA_cVal_mag_f, AIRMS);

IedServer_updateTimestampAttributeValue(iedServer,
  IEDMODEL_MET_MMXU1_A_phsB_t, &iecTimestamp);
IedServer_updateFloatAttributeValue(iedServer,
  IEDMODEL_MET_MMXU1_A_phsB_cVal_mag_f, BIRMS);

IedServer_updateTimestampAttributeValue(iedServer,
  IEDMODEL_MET_MMXU1_A_phsC_t, &iecTimestamp);
IedServer_updateFloatAttributeValue(iedServer,
  IEDMODEL_MET_MMXU1_A_phsC_cVal_mag_f, CIRMS);

//Voltage Measurements
//Read ADE9000 Registers
AVRMS += commObj.readVoltage(R_AVRMS_REGISTER);
BVRMS += commObj.readVoltage(R_BVRMS_REGISTER);
CVRMS += commObj.readVoltage(R_CVRMS_REGISTER);

//Update IED Server data attributes
IedServer_updateTimestampAttributeValue(iedServer,
  IEDMODEL_MET_MMXU1_PhV_phsA_t, &iecTimestamp);
IedServer_updateFloatAttributeValue(iedServer,
  IEDMODEL_MET_MMXU1_PhV_phsA_cVal_mag_f, AVRMS);

IedServer_updateTimestampAttributeValue(iedServer,
```



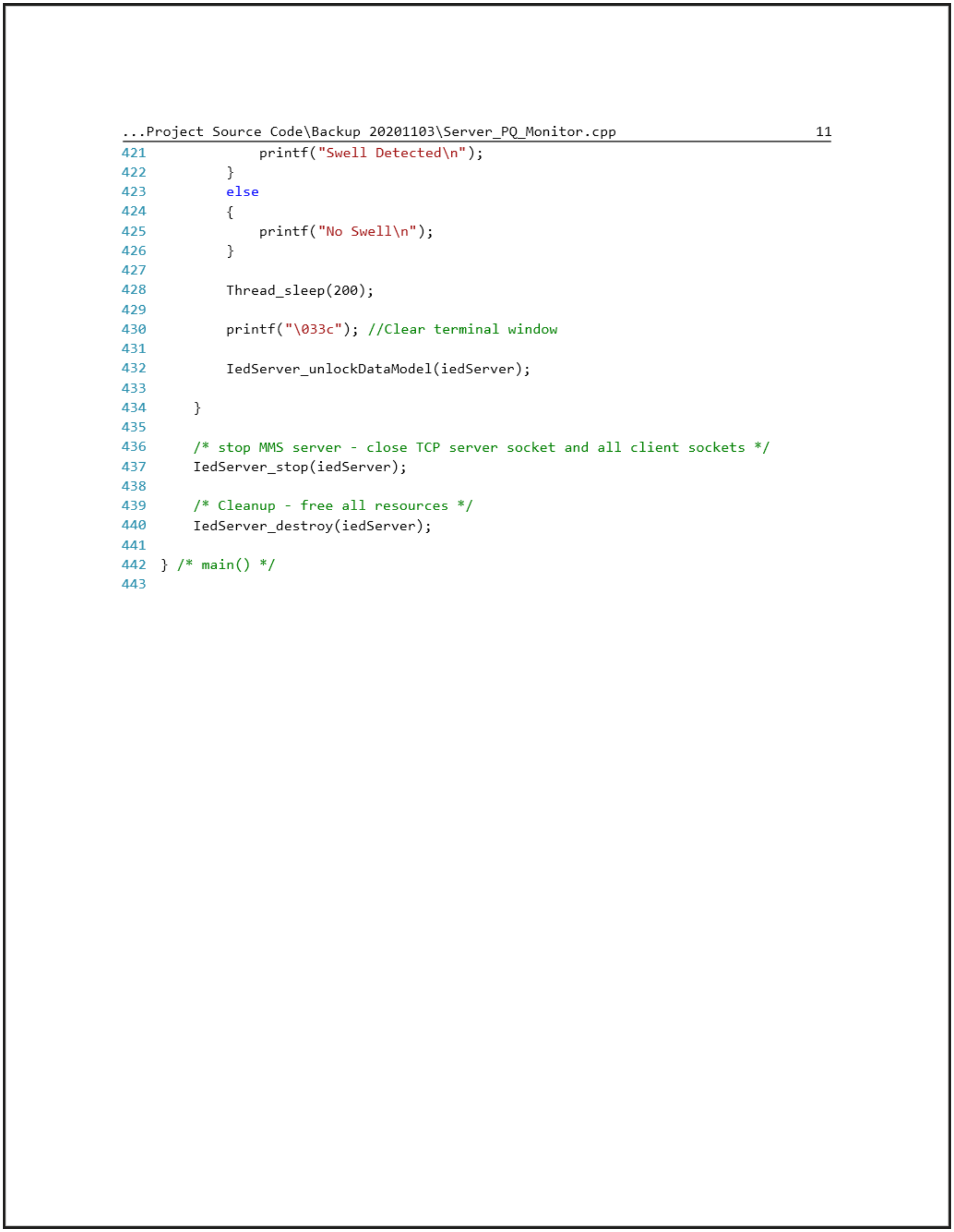
```
          IEDMODEL_MET_MMXU1_PhV_phsB_t, &iecTimestamp);
        IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_MET_MMXU1_PhV_phsB_cVal_mag_f, BVRMS);

        IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_MET_MMXU1_PhV_phsC_t, &iecTimestamp);
        IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_MET_MMXU1_PhV_phsC_cVal_mag_f, CVRMS);

        IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_MET_MMXU1_W_phsA_cVal_mag_f, commObj.readPower(R_AWATT_REGISTER));
        IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_MET_MMXU1_VA_phsA_cVal_mag_f, commObj.readPower(R_AVA_REGISTER));
        IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_MET_MMXU1_Hz_mag_f, commObj.readFrequency(R_APERIOD_REGISTER));
        IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_MET_MHAI1_ThdA_phsA_cVal_mag_f, commObj.readTHD(R_AITHD_REGISTER));
        IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_MET_MHAI1_ThdPhV_phsA_cVal_mag_f, commObj.readTHD(R_AVTHD_REGISTER));

        //A Phase Measurements
        //Read ADE9000 Registers
        AWATT = commObj.readPower(R_AWATT_REGISTER);
        AVA = commObj.readPower(R_AVA_REGISTER);
        AHz = commObj.readFrequency(R_APERIOD_REGISTER);
        AVTHD = commObj.readTHD(R_AVTHD_REGISTER);
        DIPA = commObj.readDipSwell(R_DIPA_REGISTER);
        SWELLA = commObj.readDipSwell(R_SWELLA_REGISTER);
        DIP_LVL = commObj.readVarLVL(R_DIP_LVL_REGISTER);
        DIP_CYC = commObj.readRaw(R_DIP_CYC_REGISTER);
        SWELL_LVL = commObj.readVarLVL(R_SWELL_LVL_REGISTER);
        SWELL_CYC = commObj.readRaw(R_SWELL_CYC_REGISTER);

        /* Dip and Swell simulation */

        if (220 < AVRMS < 235) /* No dip and swell between 220 and 235 Volts */
        {
            dipEvent = 0;
            swlEvent = 0;
            IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_MET_QVVR1_VarStr_t, &iecTimestamp);
            IedServer_updateBooleanAttributeValue(iedServer, IEDMODEL_MET_QVVR1_VarStr_stVal, dipEvent);
            IedServer_updateTimestampAttributeValue(iedServer,
```



```
              IEDMODEL_MET_QVVR1_DipStr_t, &iecTimestamp);
            IedServer_updateBooleanAttributeValue(iedServer, IEDMODEL_MET_QVVR1_DipStr_stVal, dipEvent);
            IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_MET_QVVR1_DipStr_t, &iecTimestamp);
            IedServer_updateBooleanAttributeValue(iedServer, IEDMODEL_MET_QVVR1_SwlStr_stVal, swlEvent);
        }

        if (AVRMS <= 220) /* Values below 220 V generate a dip condition */
        {
            dipEvent = 1;
            swlEvent = 0;
            IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_MET_QVVR1_VarStr_t, &iecTimestamp);
            IedServer_updateBooleanAttributeValue(iedServer, IEDMODEL_MET_QVVR1_VarStr_stVal, dipEvent);
            IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_MET_QVVR1_DipStr_t, &iecTimestamp);
            IedServer_updateBooleanAttributeValue(iedServer, IEDMODEL_MET_QVVR1_DipStr_stVal, dipEvent);
            IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_MET_QVVR1_DipStr_t, &iecTimestamp);
            IedServer_updateBooleanAttributeValue(iedServer, IEDMODEL_MET_QVVR1_SwlStr_stVal, swlEvent);
        }

        if (AVRMS >= 235 ) /* Values above 235 V generate swell condition */
        {
            swlEvent = 1;
            dipEvent = 0;
            IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_MET_QVVR1_VarStr_t, &iecTimestamp);
            IedServer_updateBooleanAttributeValue(iedServer, IEDMODEL_MET_QVVR1_VarStr_stVal, swlEvent);
            IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_MET_QVVR1_DipStr_t, &iecTimestamp);
            IedServer_updateBooleanAttributeValue(iedServer, IEDMODEL_MET_QVVR1_SwlStr_stVal, swlEvent);
            IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_MET_QVVR1_DipStr_t, &iecTimestamp);
            IedServer_updateBooleanAttributeValue(iedServer, IEDMODEL_MET_QVVR1_DipStr_stVal, dipEvent);
        }

        /* Register Readings Terminal Window Printout */
        printf("A Phase\n\n");
        printf("Voltage: %.2f\n", AVRMS);
        printf("Current: %.3f\n", AIRMS);
```



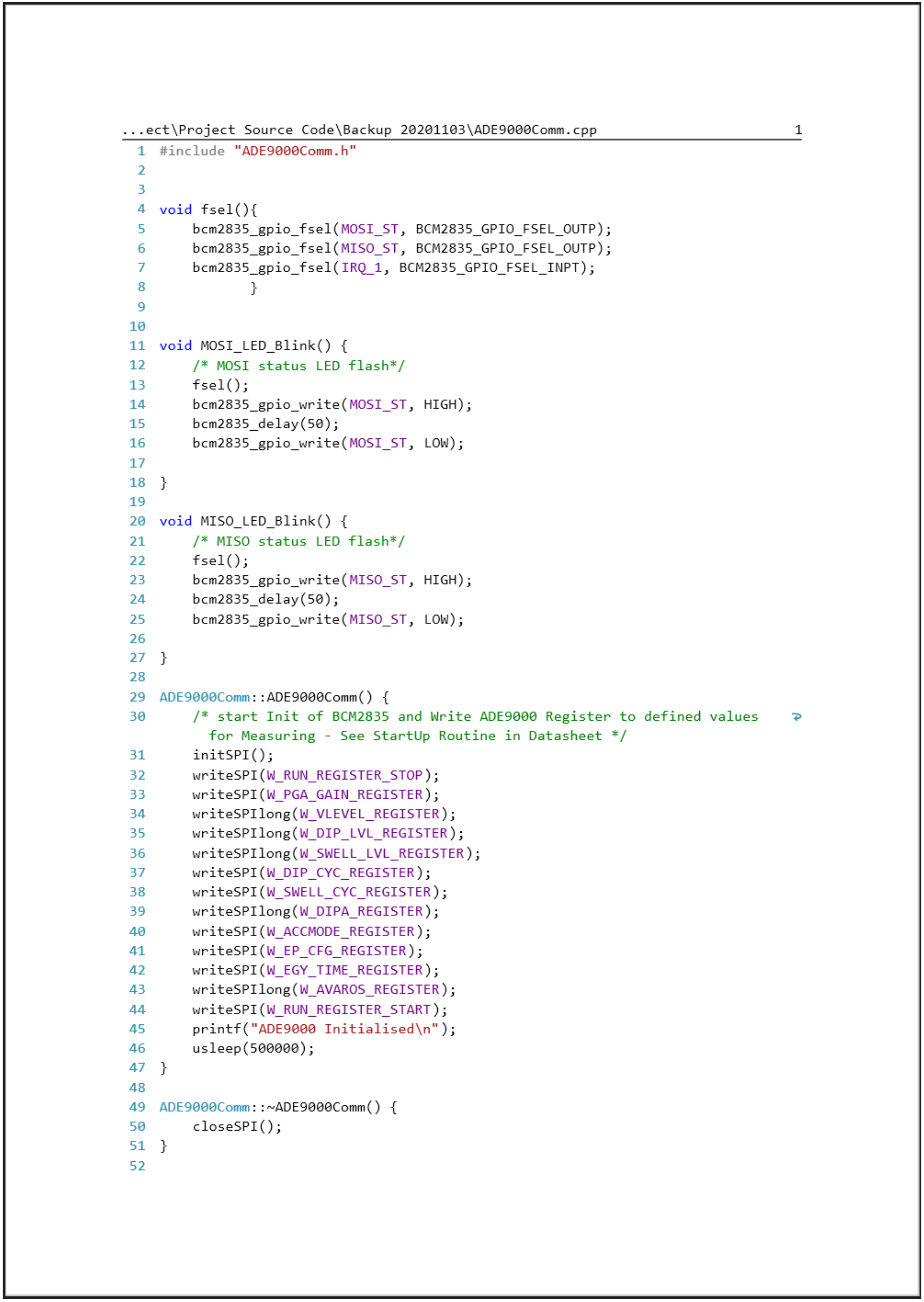
```cpp
        printf("Watt: %.2f\n", AWATT);
        printf("VA: %.2f\n", AVA);
        printf("VAR: %.2f\n", commObj.readPower(R_AVAR_REGISTER));
        printf("Freq: %.2f\n", AHz);
        printf("VTHD: %.2f\n", AVTHD);
        printf("ITHD: %.2f\n", AITHD);
        printf("Power Factor: %.2f\n\n", commObj.readPowerFactor(R_APF_REGISTER));
        printf("Dip Level: %.2f\n", DIP_LVL);
        printf("Dip Cycles: %.0f\n", DIP_CYC);
        printf("Dip Value: %.2f\n\n", DIPA);
        printf("Swell Level: %.2f\n", SWELL_LVL);
        printf("Swell Cycles: %.0f\n", SWELL_CYC);
        printf("Swell Value: %.2f\n\n", SWELLA);

        if(iOut.good())
        {
            iOut << AIRMS;
            iOut << "       ";
            iOut << AVRMS;
            iOut << "       ";
            iOut << AVTHD;
            iOut << "       ";
            iOut << AITHD;
            iOut << "       ";
            iOut << AHz;
            iOut << "\n";
        }

        if ( IsBitSet(commObj.readStatus(R_STATUS1_REGISTER), 9))
        {
            printf("Zero Crossing Detected\n");
        }
        else
        {
            printf("No Zero Crossing\n");
        }

        if ( IsBitSet(commObj.readStatus(R_STATUS1_REGISTER), 23))
        {
            printf("Dip Detected\n");
        }
        else
        {
            printf("No Dip\n");
        }

        if ( IsBitSet(commObj.readStatus(R_STATUS1_REGISTER), 20))
        {
```



...Project Source Code\Backup 20201103\Server_PQ_Monitor.cpp 11

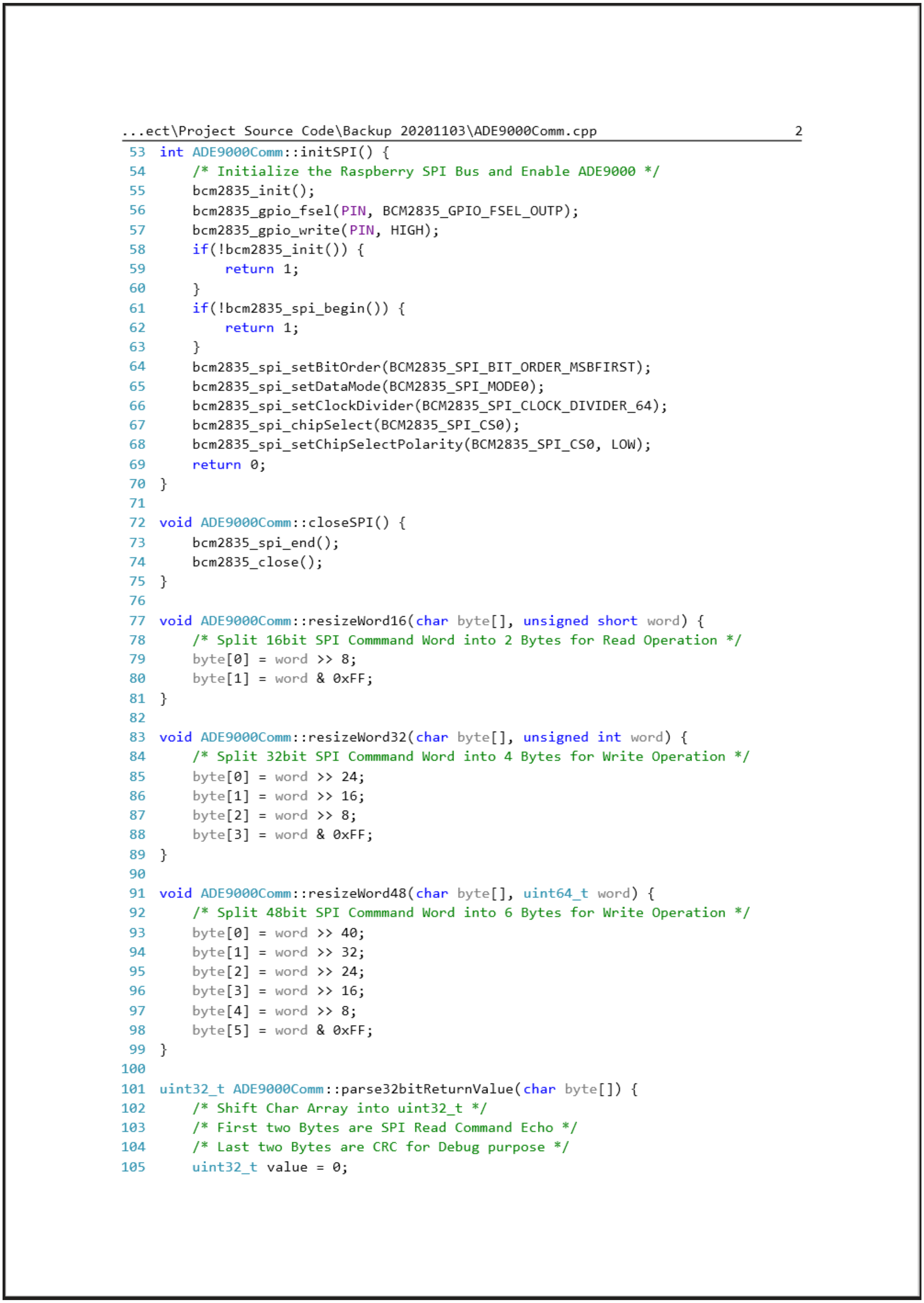
```
            printf("Swell Detected\n");
        }
        else
        {
            printf("No Swell\n");
        }

        Thread_sleep(200);

        printf("\033c"); //Clear terminal window

        IedServer_unlockDataModel(iedServer);

    }

    /* stop MMS server - close TCP server socket and all client sockets */
    IedServer_stop(iedServer);

    /* Cleanup - free all resources */
    IedServer_destroy(iedServer);

} /* main() */

```

ADE9000 and SPI Configuration – ADE9000Com.cpp



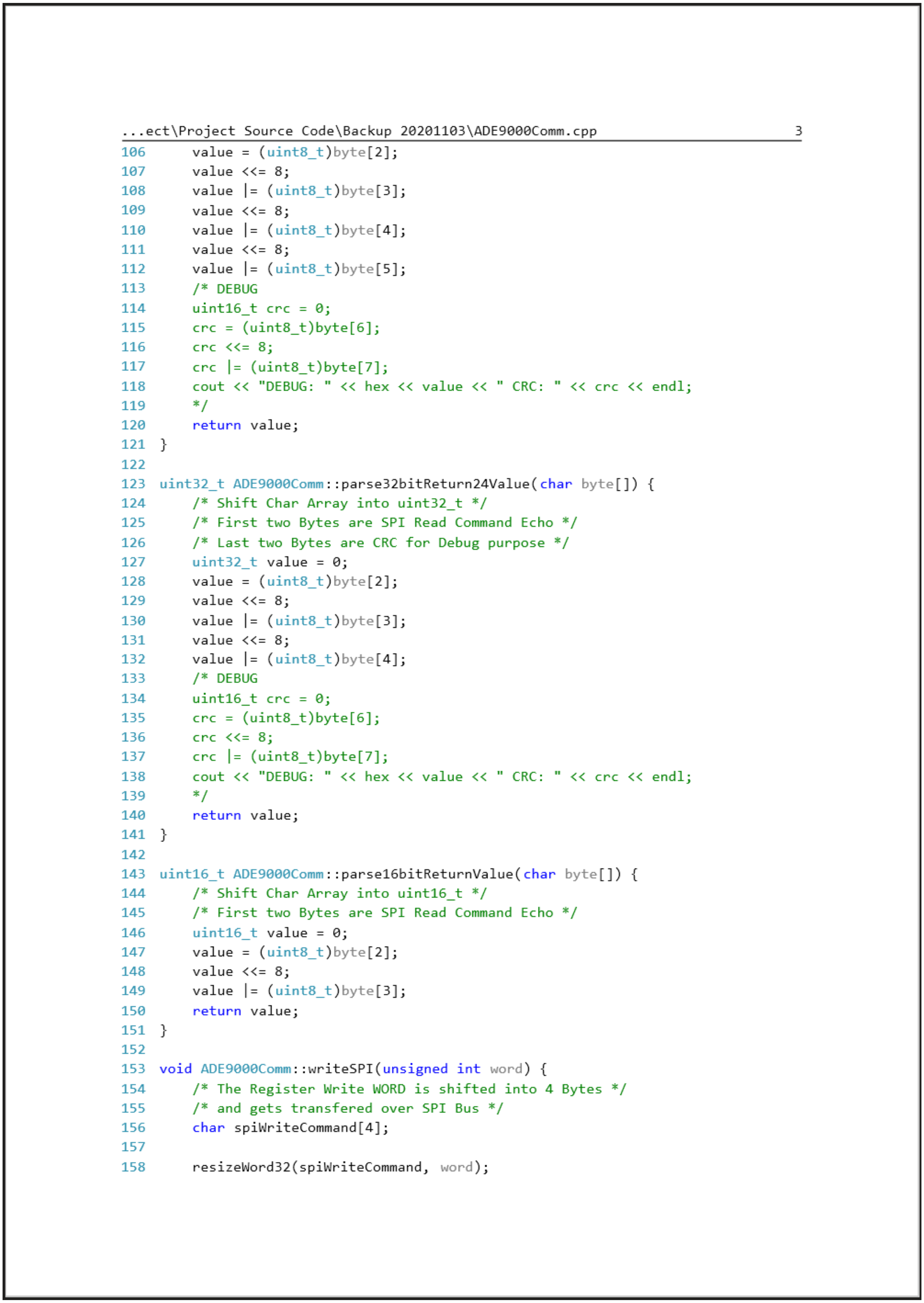
...ect\Project Source Code\Backup 20201103\ADE9000Comm.cpp 1

```
#include "ADE9000Comm.h"


void fsel(){
    bcm2835_gpio_fsel(MOSI_ST, BCM2835_GPIO_FSEL_OUTP);
    bcm2835_gpio_fsel(MISO_ST, BCM2835_GPIO_FSEL_OUTP);
    bcm2835_gpio_fsel(IRQ_1, BCM2835_GPIO_FSEL_INPT);
           }


void MOSI_LED_Blink() {
    /* MOSI status LED flash*/
    fsel();
    bcm2835_gpio_write(MOSI_ST, HIGH);
    bcm2835_delay(50);
    bcm2835_gpio_write(MOSI_ST, LOW);

}

void MISO_LED_Blink() {
    /* MISO status LED flash*/
    fsel();
    bcm2835_gpio_write(MISO_ST, HIGH);
    bcm2835_delay(50);
    bcm2835_gpio_write(MISO_ST, LOW);

}

ADE9000Comm::ADE9000Comm() {
    /* start Init of BCM2835 and Write ADE9000 Register to defined values for Measuring - See StartUp Routine in Datasheet */
    initSPI();
    writeSPI(W_RUN_REGISTER_STOP);
    writeSPI(W_PGA_GAIN_REGISTER);
    writeSPIlong(W_VLEVEL_REGISTER);
    writeSPIlong(W_DIP_LVL_REGISTER);
    writeSPIlong(W_SWELL_LVL_REGISTER);
    writeSPI(W_DIP_CYC_REGISTER);
    writeSPI(W_SWELL_CYC_REGISTER);
    writeSPIlong(W_DIPA_REGISTER);
    writeSPI(W_ACCMODE_REGISTER);
    writeSPI(W_EP_CFG_REGISTER);
    writeSPI(W_EGY_TIME_REGISTER);
    writeSPIlong(W_AVAROS_REGISTER);
    writeSPI(W_RUN_REGISTER_START);
    printf("ADE9000 Initialised\n");
    usleep(500000);
}

ADE9000Comm::~ADE9000Comm() {
    closeSPI();
}

```



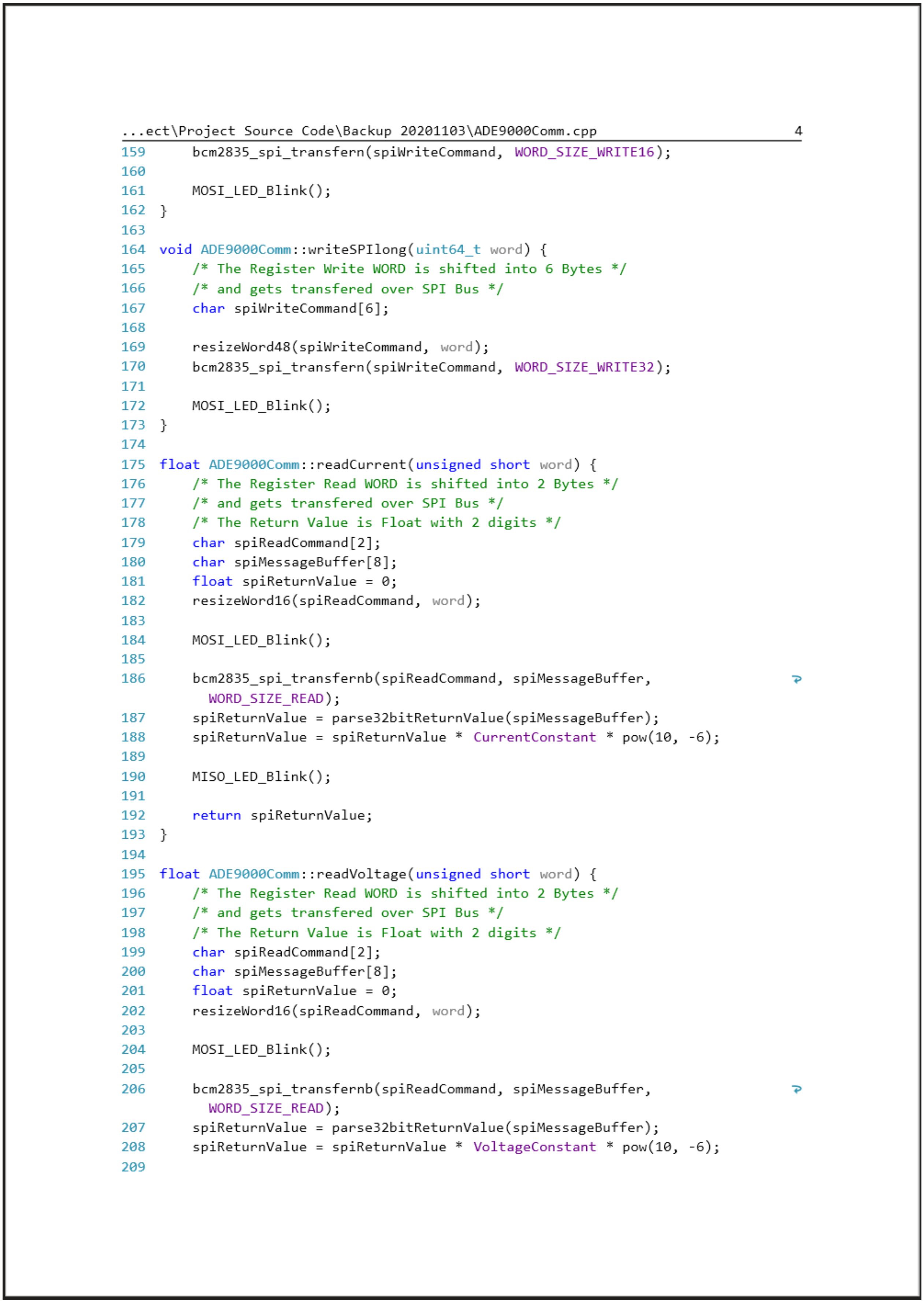
...ect\Project Source Code\Backup 20201103\ADE9000Comm.cpp 2

```cpp
int ADE9000Comm::initSPI() {
    /* Initialize the Raspberry SPI Bus and Enable ADE9000 */
    bcm2835_init();
    bcm2835_gpio_fsel(PIN, BCM2835_GPIO_FSEL_OUTP);
    bcm2835_gpio_write(PIN, HIGH);
    if(!bcm2835_init()) {
        return 1;
    }
    if(!bcm2835_spi_begin()) {
        return 1;
    }
    bcm2835_spi_setBitOrder(BCM2835_SPI_BIT_ORDER_MSBFIRST);
    bcm2835_spi_setDataMode(BCM2835_SPI_MODE0);
    bcm2835_spi_setClockDivider(BCM2835_SPI_CLOCK_DIVIDER_64);
    bcm2835_spi_chipSelect(BCM2835_SPI_CS0);
    bcm2835_spi_setChipSelectPolarity(BCM2835_SPI_CS0, LOW);
    return 0;
}

void ADE9000Comm::closeSPI() {
    bcm2835_spi_end();
    bcm2835_close();
}

void ADE9000Comm::resizeWord16(char byte[], unsigned short word) {
    /* Split 16bit SPI Commmand Word into 2 Bytes for Read Operation */
    byte[0] = word >> 8;
    byte[1] = word & 0xFF;
}

void ADE9000Comm::resizeWord32(char byte[], unsigned int word) {
    /* Split 32bit SPI Commmand Word into 4 Bytes for Write Operation */
    byte[0] = word >> 24;
    byte[1] = word >> 16;
    byte[2] = word >> 8;
    byte[3] = word & 0xFF;
}

void ADE9000Comm::resizeWord48(char byte[], uint64_t word) {
    /* Split 48bit SPI Commmand Word into 6 Bytes for Write Operation */
    byte[0] = word >> 40;
    byte[1] = word >> 32;
    byte[2] = word >> 24;
    byte[3] = word >> 16;
    byte[4] = word >> 8;
    byte[5] = word & 0xFF;
}

uint32_t ADE9000Comm::parse32bitReturnValue(char byte[]) {
    /* Shift Char Array into uint32_t */
    /* First two Bytes are SPI Read Command Echo */
    /* Last two Bytes are CRC for Debug purpose */
    uint32_t value = 0;
```



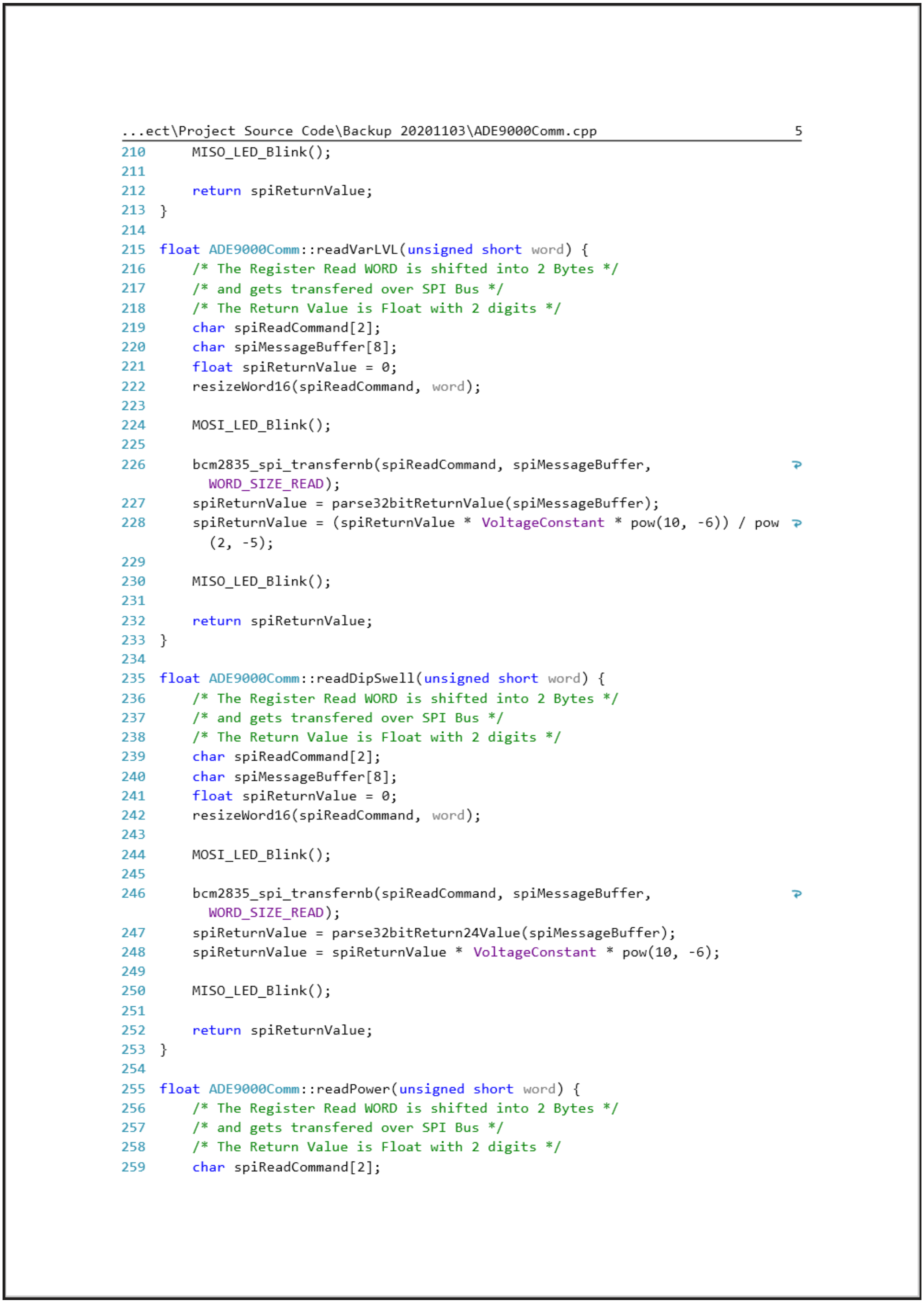
...ect\Project Source Code\Backup 20201103\ADE9000Comm.cpp 3

```
    value = (uint8_t)byte[2];
    value <<= 8;
    value |= (uint8_t)byte[3];
    value <<= 8;
    value |= (uint8_t)byte[4];
    value <<= 8;
    value |= (uint8_t)byte[5];
    /* DEBUG
    uint16_t crc = 0;
    crc = (uint8_t)byte[6];
    crc <<= 8;
    crc |= (uint8_t)byte[7];
    cout << "DEBUG: " << hex << value << " CRC: " << crc << endl;
    */
    return value;
}

uint32_t ADE9000Comm::parse32bitReturn24Value(char byte[]) {
    /* Shift Char Array into uint32_t */
    /* First two Bytes are SPI Read Command Echo */
    /* Last two Bytes are CRC for Debug purpose */
    uint32_t value = 0;
    value = (uint8_t)byte[2];
    value <<= 8;
    value |= (uint8_t)byte[3];
    value <<= 8;
    value |= (uint8_t)byte[4];
    /* DEBUG
    uint16_t crc = 0;
    crc = (uint8_t)byte[6];
    crc <<= 8;
    crc |= (uint8_t)byte[7];
    cout << "DEBUG: " << hex << value << " CRC: " << crc << endl;
    */
    return value;
}

uint16_t ADE9000Comm::parse16bitReturnValue(char byte[]) {
    /* Shift Char Array into uint16_t */
    /* First two Bytes are SPI Read Command Echo */
    uint16_t value = 0;
    value = (uint8_t)byte[2];
    value <<= 8;
    value |= (uint8_t)byte[3];
    return value;
}

void ADE9000Comm::writeSPI(unsigned int word) {
    /* The Register Write WORD is shifted into 4 Bytes */
    /* and gets transfered over SPI Bus */
    char spiWriteCommand[4];

    resizeWord32(spiWriteCommand, word);
```

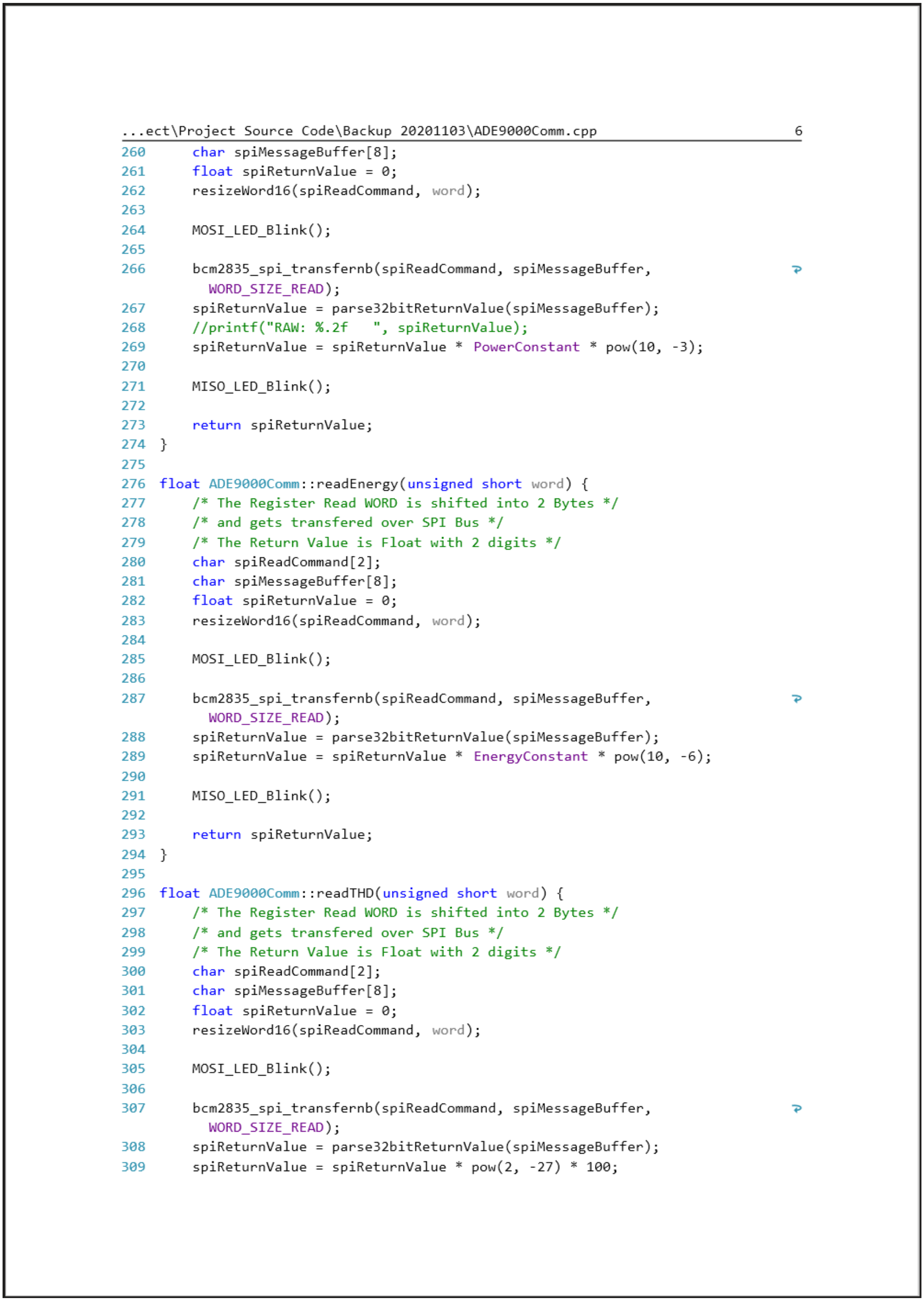


...ect\Project Source Code\Backup 20201103\ADE9000Comm.cpp 4

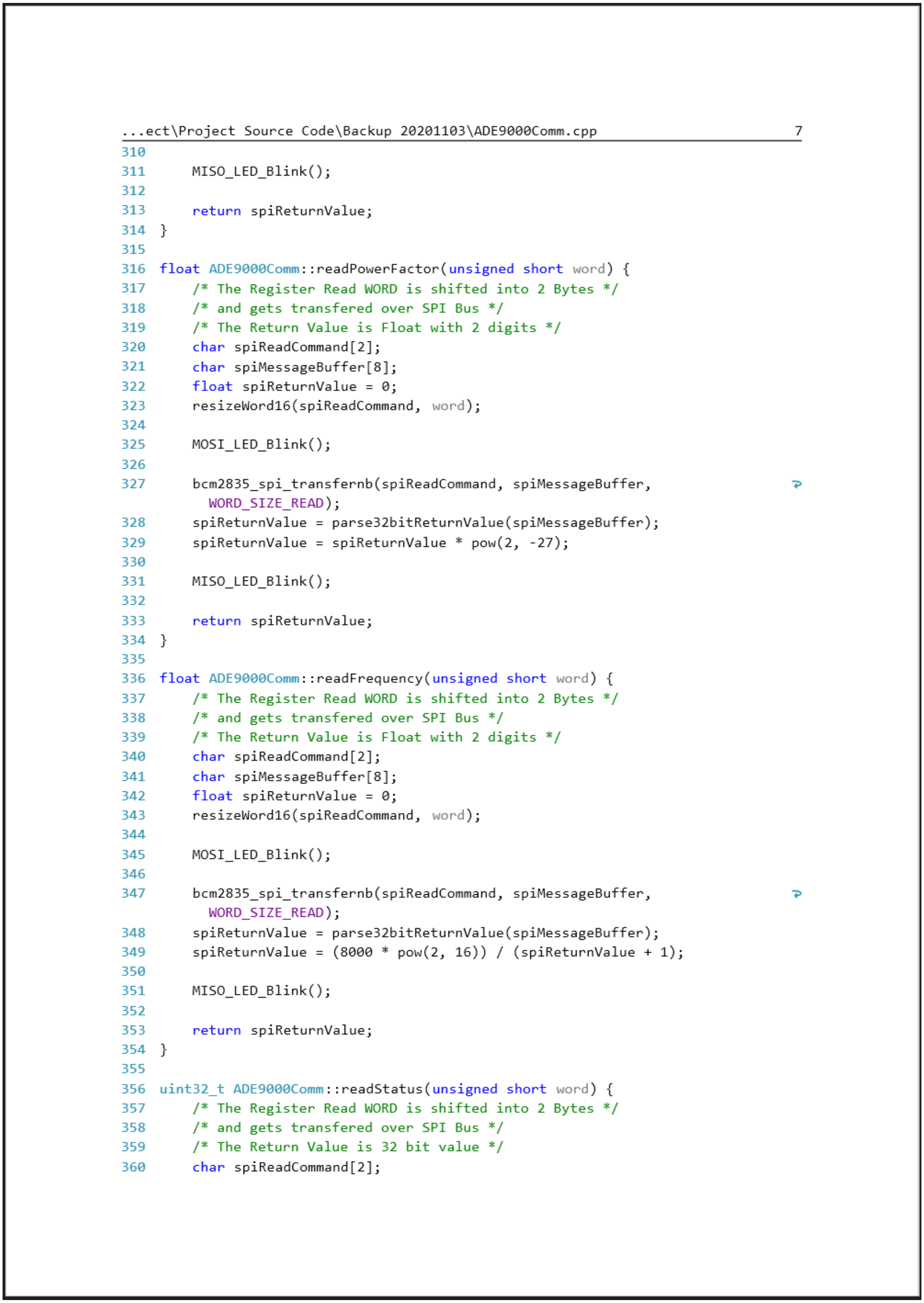
```cpp
    bcm2835_spi_transfern(spiWriteCommand, WORD_SIZE_WRITE16);

    MOSI_LED_Blink();
}

void ADE9000Comm::writeSPIlong(uint64_t word) {
    /* The Register Write WORD is shifted into 6 Bytes */
    /* and gets transfered over SPI Bus */
    char spiWriteCommand[6];

    resizeWord48(spiWriteCommand, word);
    bcm2835_spi_transfern(spiWriteCommand, WORD_SIZE_WRITE32);

    MOSI_LED_Blink();
}

float ADE9000Comm::readCurrent(unsigned short word) {
    /* The Register Read WORD is shifted into 2 Bytes */
    /* and gets transfered over SPI Bus */
    /* The Return Value is Float with 2 digits */
    char spiReadCommand[2];
    char spiMessageBuffer[8];
    float spiReturnValue = 0;
    resizeWord16(spiReadCommand, word);

    MOSI_LED_Blink();

    bcm2835_spi_transfernb(spiReadCommand, spiMessageBuffer,
      WORD_SIZE_READ);
    spiReturnValue = parse32bitReturnValue(spiMessageBuffer);
    spiReturnValue = spiReturnValue * CurrentConstant * pow(10, -6);

    MISO_LED_Blink();

    return spiReturnValue;
}

float ADE9000Comm::readVoltage(unsigned short word) {
    /* The Register Read WORD is shifted into 2 Bytes */
    /* and gets transfered over SPI Bus */
    /* The Return Value is Float with 2 digits */
    char spiReadCommand[2];
    char spiMessageBuffer[8];
    float spiReturnValue = 0;
    resizeWord16(spiReadCommand, word);

    MOSI_LED_Blink();

    bcm2835_spi_transfernb(spiReadCommand, spiMessageBuffer,
      WORD_SIZE_READ);
    spiReturnValue = parse32bitReturnValue(spiMessageBuffer);
    spiReturnValue = spiReturnValue * VoltageConstant * pow(10, -6);

```



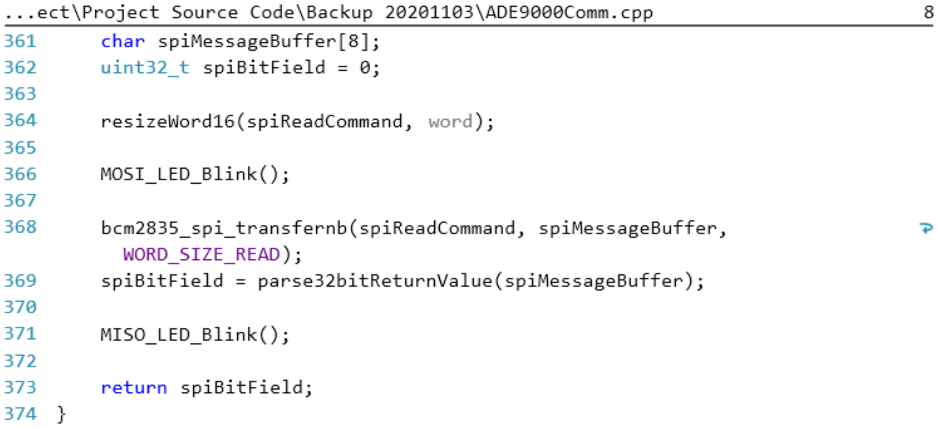
```cpp
    MISO_LED_Blink();

    return spiReturnValue;
}

float ADE9000Comm::readVarLVL(unsigned short word) {
    /* The Register Read WORD is shifted into 2 Bytes */
    /* and gets transfered over SPI Bus */
    /* The Return Value is Float with 2 digits */
    char spiReadCommand[2];
    char spiMessageBuffer[8];
    float spiReturnValue = 0;
    resizeWord16(spiReadCommand, word);

    MOSI_LED_Blink();

    bcm2835_spi_transfernb(spiReadCommand, spiMessageBuffer, WORD_SIZE_READ);
    spiReturnValue = parse32bitReturnValue(spiMessageBuffer);
    spiReturnValue = (spiReturnValue * VoltageConstant * pow(10, -6)) / pow(2, -5);

    MISO_LED_Blink();

    return spiReturnValue;
}

float ADE9000Comm::readDipSwell(unsigned short word) {
    /* The Register Read WORD is shifted into 2 Bytes */
    /* and gets transfered over SPI Bus */
    /* The Return Value is Float with 2 digits */
    char spiReadCommand[2];
    char spiMessageBuffer[8];
    float spiReturnValue = 0;
    resizeWord16(spiReadCommand, word);

    MOSI_LED_Blink();

    bcm2835_spi_transfernb(spiReadCommand, spiMessageBuffer, WORD_SIZE_READ);
    spiReturnValue = parse32bitReturn24Value(spiMessageBuffer);
    spiReturnValue = spiReturnValue * VoltageConstant * pow(10, -6);

    MISO_LED_Blink();

    return spiReturnValue;
}

float ADE9000Comm::readPower(unsigned short word) {
    /* The Register Read WORD is shifted into 2 Bytes */
    /* and gets transfered over SPI Bus */
    /* The Return Value is Float with 2 digits */
    char spiReadCommand[2];
```



```
    char spiMessageBuffer[8];
    float spiReturnValue = 0;
    resizeWord16(spiReadCommand, word);

    MOSI_LED_Blink();

    bcm2835_spi_transfernb(spiReadCommand, spiMessageBuffer, WORD_SIZE_READ);
    spiReturnValue = parse32bitReturnValue(spiMessageBuffer);
    //printf("RAW: %.2f   ", spiReturnValue);
    spiReturnValue = spiReturnValue * PowerConstant * pow(10, -3);

    MISO_LED_Blink();

    return spiReturnValue;
}

float ADE9000Comm::readEnergy(unsigned short word) {
    /* The Register Read WORD is shifted into 2 Bytes */
    /* and gets transfered over SPI Bus */
    /* The Return Value is Float with 2 digits */
    char spiReadCommand[2];
    char spiMessageBuffer[8];
    float spiReturnValue = 0;
    resizeWord16(spiReadCommand, word);

    MOSI_LED_Blink();

    bcm2835_spi_transfernb(spiReadCommand, spiMessageBuffer, WORD_SIZE_READ);
    spiReturnValue = parse32bitReturnValue(spiMessageBuffer);
    spiReturnValue = spiReturnValue * EnergyConstant * pow(10, -6);

    MISO_LED_Blink();

    return spiReturnValue;
}

float ADE9000Comm::readTHD(unsigned short word) {
    /* The Register Read WORD is shifted into 2 Bytes */
    /* and gets transfered over SPI Bus */
    /* The Return Value is Float with 2 digits */
    char spiReadCommand[2];
    char spiMessageBuffer[8];
    float spiReturnValue = 0;
    resizeWord16(spiReadCommand, word);

    MOSI_LED_Blink();

    bcm2835_spi_transfernb(spiReadCommand, spiMessageBuffer, WORD_SIZE_READ);
    spiReturnValue = parse32bitReturnValue(spiMessageBuffer);
    spiReturnValue = spiReturnValue * pow(2, -27) * 100;
```



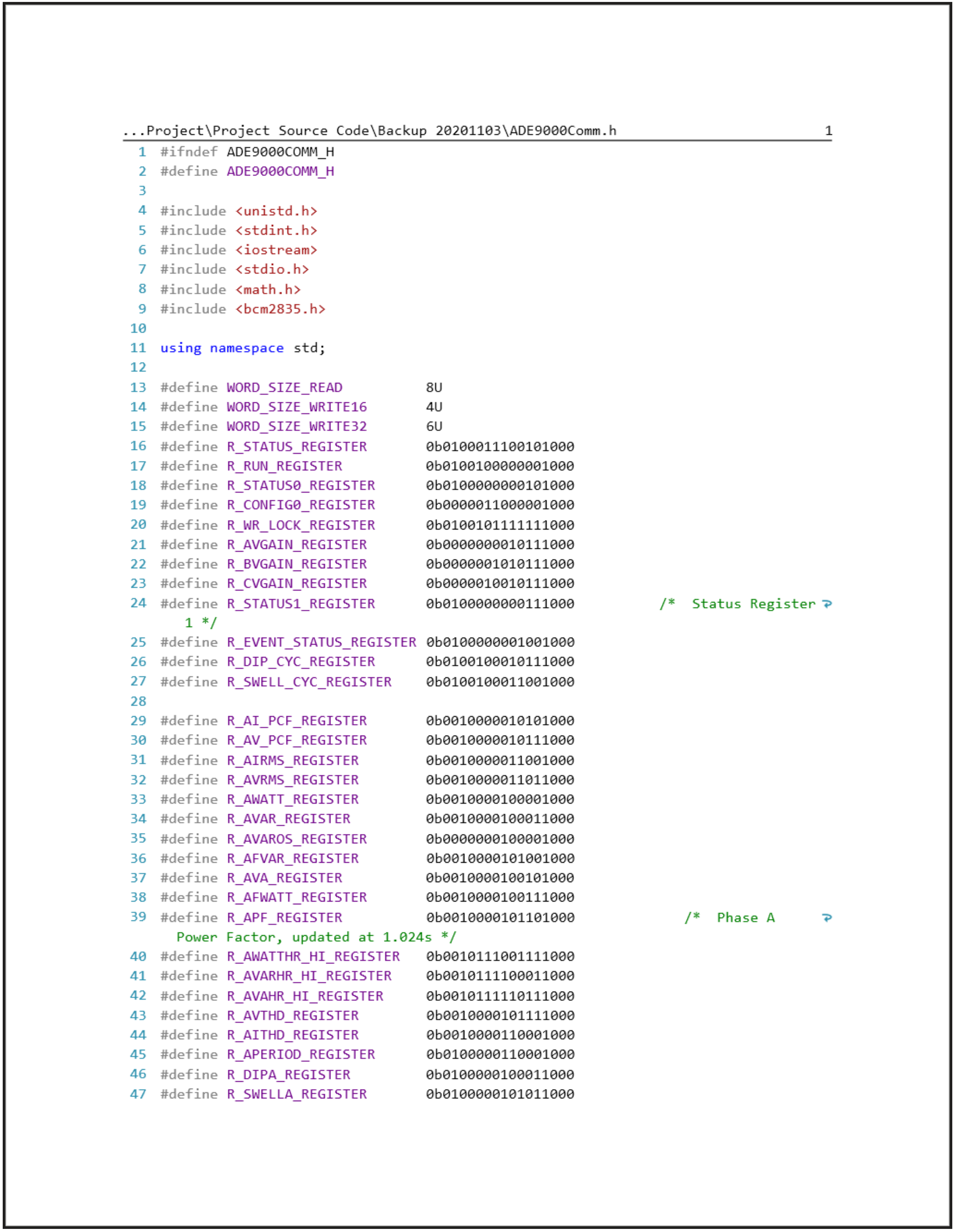
```cpp

    MISO_LED_Blink();

    return spiReturnValue;
}

float ADE9000Comm::readPowerFactor(unsigned short word) {
    /* The Register Read WORD is shifted into 2 Bytes */
    /* and gets transfered over SPI Bus */
    /* The Return Value is Float with 2 digits */
    char spiReadCommand[2];
    char spiMessageBuffer[8];
    float spiReturnValue = 0;
    resizeWord16(spiReadCommand, word);

    MOSI_LED_Blink();

    bcm2835_spi_transfernb(spiReadCommand, spiMessageBuffer,
      WORD_SIZE_READ);
    spiReturnValue = parse32bitReturnValue(spiMessageBuffer);
    spiReturnValue = spiReturnValue * pow(2, -27);

    MISO_LED_Blink();

    return spiReturnValue;
}

float ADE9000Comm::readFrequency(unsigned short word) {
    /* The Register Read WORD is shifted into 2 Bytes */
    /* and gets transfered over SPI Bus */
    /* The Return Value is Float with 2 digits */
    char spiReadCommand[2];
    char spiMessageBuffer[8];
    float spiReturnValue = 0;
    resizeWord16(spiReadCommand, word);

    MOSI_LED_Blink();

    bcm2835_spi_transfernb(spiReadCommand, spiMessageBuffer,
      WORD_SIZE_READ);
    spiReturnValue = parse32bitReturnValue(spiMessageBuffer);
    spiReturnValue = (8000 * pow(2, 16)) / (spiReturnValue + 1);

    MISO_LED_Blink();

    return spiReturnValue;
}

uint32_t ADE9000Comm::readStatus(unsigned short word) {
    /* The Register Read WORD is shifted into 2 Bytes */
    /* and gets transfered over SPI Bus */
    /* The Return Value is 32 bit value */
    char spiReadCommand[2];
```



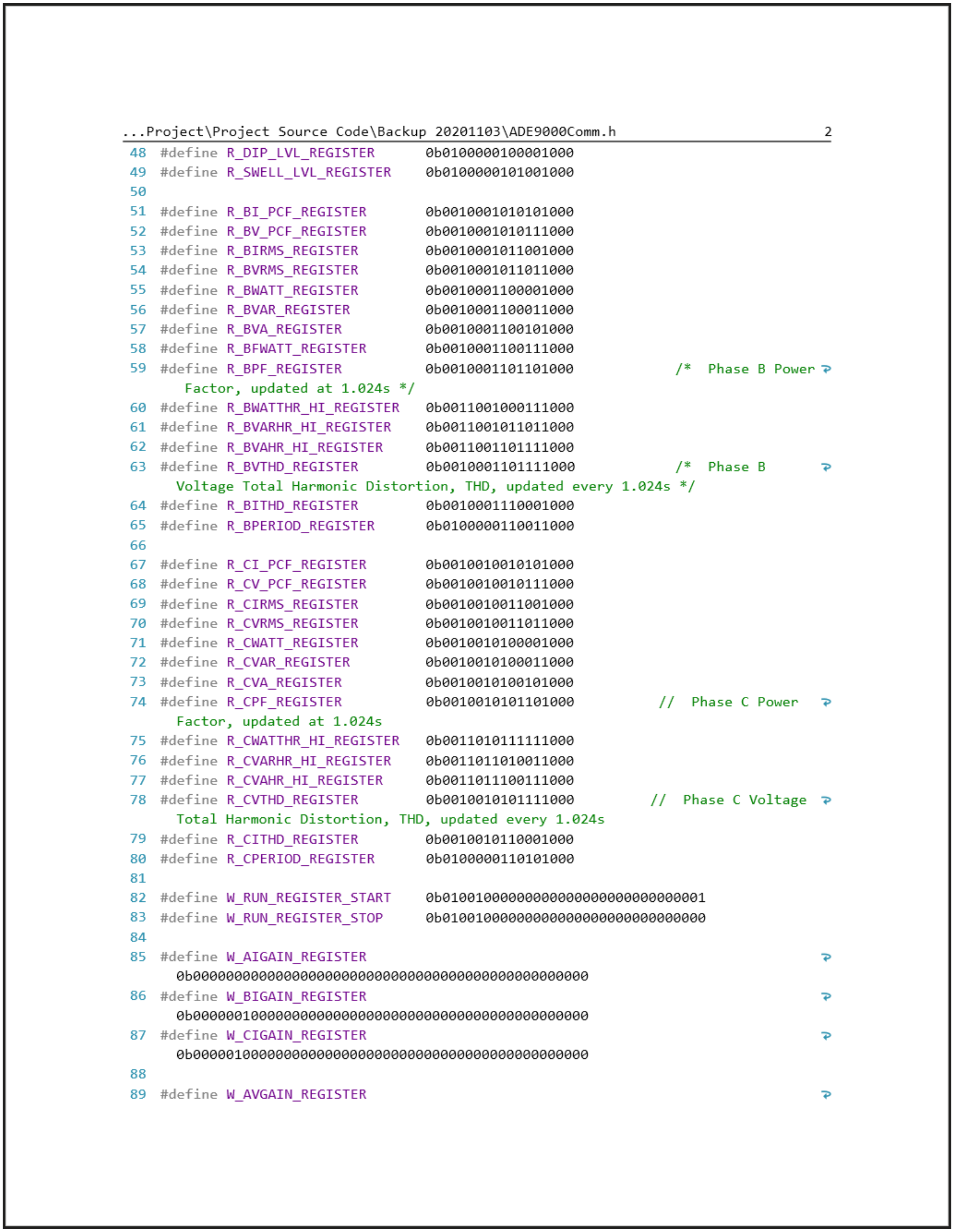
...ect\Project Source Code\Backup 20201103\ADE9000Comm.cpp 8

```cpp
    char spiMessageBuffer[8];
    uint32_t spiBitField = 0;

    resizeWord16(spiReadCommand, word);

    MOSI_LED_Blink();

    bcm2835_spi_transfernb(spiReadCommand, spiMessageBuffer, WORD_SIZE_READ);
    spiBitField = parse32bitReturnValue(spiMessageBuffer);

    MISO_LED_Blink();

    return spiBitField;
}
```

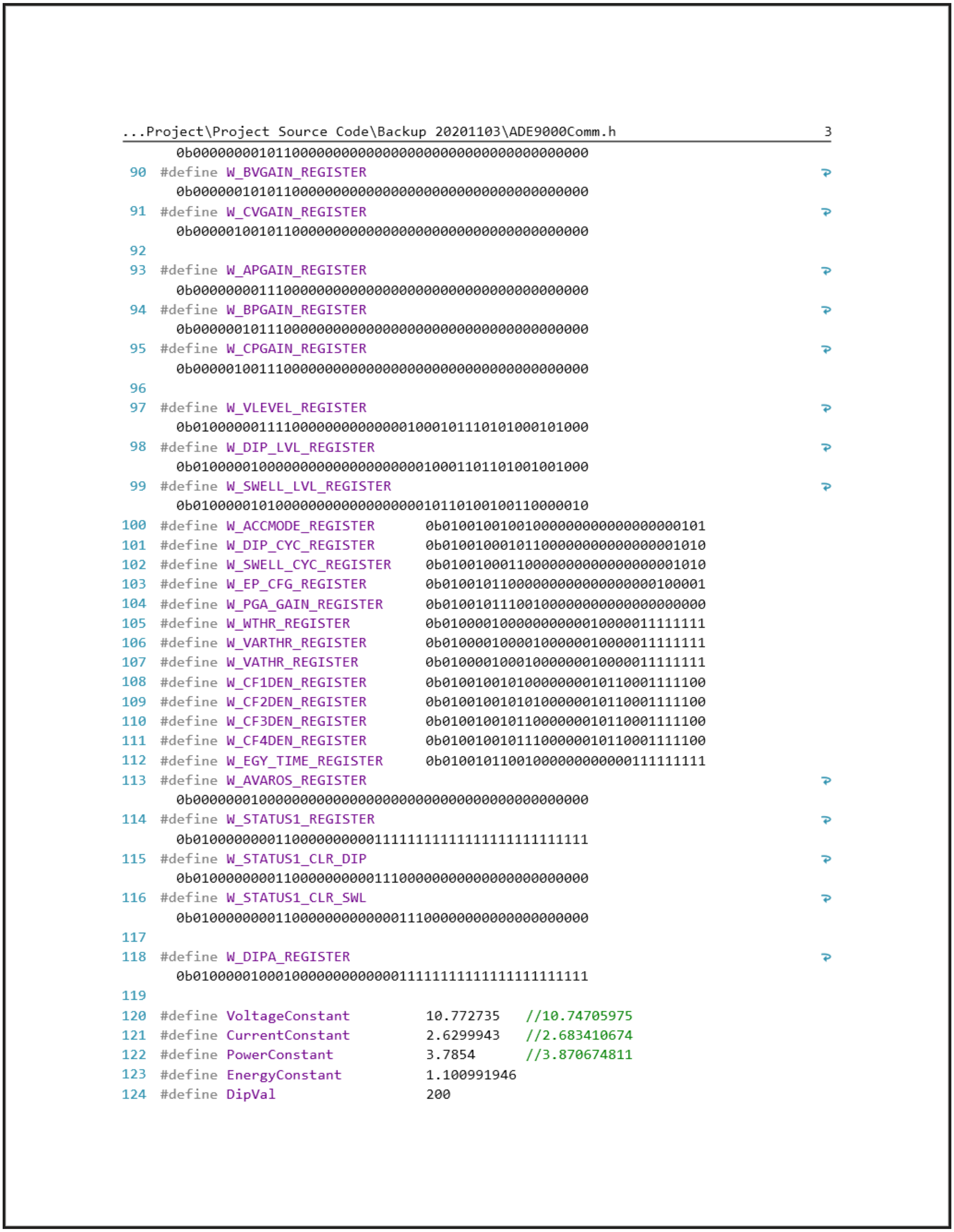
Header file definitions for functions and ADE9000 data registers – ADE9000Comm.h



```
...Project\Project Source Code\Backup 20201103\ADE9000Comm.h                    1
#ifndef ADE9000COMM_H
#define ADE9000COMM_H

#include <unistd.h>
#include <stdint.h>
#include <iostream>
#include <stdio.h>
#include <math.h>
#include <bcm2835.h>

using namespace std;

#define WORD_SIZE_READ          8U
#define WORD_SIZE_WRITE16       4U
#define WORD_SIZE_WRITE32       6U
#define R_STATUS_REGISTER       0b0100011100101000
#define R_RUN_REGISTER          0b0100100000001000
#define R_STATUS0_REGISTER      0b0100000000101000
#define R_CONFIG0_REGISTER      0b0000011000001000
#define R_WR_LOCK_REGISTER      0b0100101111111000
#define R_AVGAIN_REGISTER       0b0000000010111000
#define R_BVGAIN_REGISTER       0b0000001010111000
#define R_CVGAIN_REGISTER       0b0000010010111000
#define R_STATUS1_REGISTER      0b0100000000111000          /*  Status Register
   1 */
#define R_EVENT_STATUS_REGISTER 0b0100000001001000
#define R_DIP_CYC_REGISTER      0b0100100010111000
#define R_SWELL_CYC_REGISTER    0b0100100011001000

#define R_AI_PCF_REGISTER       0b0010000010101000
#define R_AV_PCF_REGISTER       0b0010000010111000
#define R_AIRMS_REGISTER        0b0010000011001000
#define R_AVRMS_REGISTER        0b0010000011011000
#define R_AWATT_REGISTER        0b0010000100001000
#define R_AVAR_REGISTER         0b0010000100011000
#define R_AVAROS_REGISTER       0b0000000100001000
#define R_AFVAR_REGISTER        0b0010000101001000
#define R_AVA_REGISTER          0b0010000100101000
#define R_AFWATT_REGISTER       0b0010000100111000
#define R_APF_REGISTER          0b0010000101101000             /*  Phase A
  Power Factor, updated at 1.024s */
#define R_AWATTHR_HI_REGISTER   0b0010111001111000
#define R_AVARHR_HI_REGISTER    0b0010111100011000
#define R_AVAHR_HI_REGISTER     0b0010111110111000
#define R_AVTHD_REGISTER        0b0010000101111000
#define R_AITHD_REGISTER        0b0010000110001000
#define R_APERIOD_REGISTER      0b0100000110001000
#define R_DIPA_REGISTER         0b0100000100011000
#define R_SWELLA_REGISTER       0b0100000101011000
```

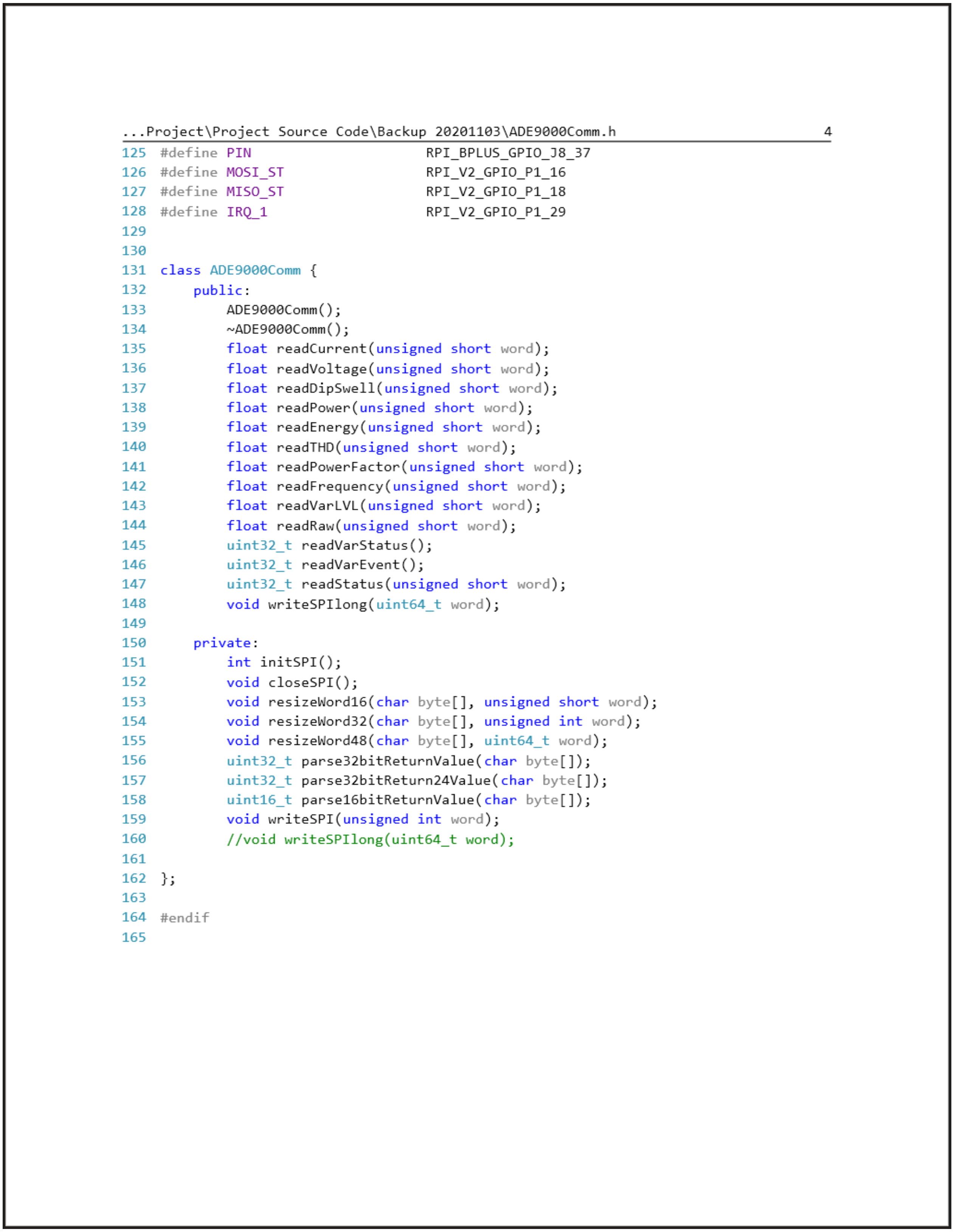


```
#define R_DIP_LVL_REGISTER      0b0100000100001000
#define R_SWELL_LVL_REGISTER    0b0100000101001000

#define R_BI_PCF_REGISTER       0b0010001010101000
#define R_BV_PCF_REGISTER       0b0010001010111000
#define R_BIRMS_REGISTER        0b0010001011001000
#define R_BVRMS_REGISTER        0b0010001011011000
#define R_BWATT_REGISTER        0b0010001100001000
#define R_BVAR_REGISTER         0b0010001100011000
#define R_BVA_REGISTER          0b0010001100101000
#define R_BFWATT_REGISTER       0b0010001100111000
#define R_BPF_REGISTER          0b0010001101101000              /*  Phase B Power Factor, updated at 1.024s */
#define R_BWATTHR_HI_REGISTER   0b0011001000111000
#define R_BVARHR_HI_REGISTER    0b0011001011011000
#define R_BVAHR_HI_REGISTER     0b0011001101111000
#define R_BVTHD_REGISTER        0b0010001101111000              /*  Phase B Voltage Total Harmonic Distortion, THD, updated every 1.024s */
#define R_BITHD_REGISTER        0b0010001110001000
#define R_BPERIOD_REGISTER      0b0100000110011000

#define R_CI_PCF_REGISTER       0b0010010010101000
#define R_CV_PCF_REGISTER       0b0010010010111000
#define R_CIRMS_REGISTER        0b0010010011001000
#define R_CVRMS_REGISTER        0b0010010011011000
#define R_CWATT_REGISTER        0b0010010100001000
#define R_CVAR_REGISTER         0b0010010100011000
#define R_CVA_REGISTER          0b0010010100101000
#define R_CPF_REGISTER          0b0010010101101000            //  Phase C Power Factor, updated at 1.024
#define R_CWATTHR_HI_REGISTER   0b0011010111111000
#define R_CVARHR_HI_REGISTER    0b0011011010011000
#define R_CVAHR_HI_REGISTER     0b0011011100111000
#define R_CVTHD_REGISTER        0b0010010101111000           //  Phase C Voltage Total Harmonic Distortion, THD, updated every 1.024s
#define R_CITHD_REGISTER        0b0010010110001000
#define R_CPERIOD_REGISTER      0b0100000110101000

#define W_RUN_REGISTER_START    0b01001000000000000000000000000001
#define W_RUN_REGISTER_STOP     0b01001000000000000000000000000000

#define W_AIGAIN_REGISTER 0b000000000000000000000000000000000000000000000000
#define W_BIGAIN_REGISTER 0b000000100000000000000000000000000000000000000000
#define W_CIGAIN_REGISTER 0b000001000000000000000000000000000000000000000000

#define W_AVGAIN_REGISTER
```



...Project\Project Source Code\Backup 20201103\ADE9000Comm.h 3

```
    0b000000001011000000000000000000000000000000000000
#define W_BVGAIN_REGISTER
    0b000000101011000000000000000000000000000000000000
#define W_CVGAIN_REGISTER
    0b000001001011000000000000000000000000000000000000

#define W_APGAIN_REGISTER
    0b000000001110000000000000000000000000000000000000
#define W_BPGAIN_REGISTER
    0b000000101110000000000000000000000000000000000000
#define W_CPGAIN_REGISTER
    0b000001001110000000000000000000000000000000000000

#define W_VLEVEL_REGISTER
    0b010000001111000000000000001000101110101000101000
#define W_DIP_LVL_REGISTER
    0b010000010000000000000000000010001101101001001000
#define W_SWELL_LVL_REGISTER
    0b010000010100000000000000000010110100100110000010
#define W_ACCMODE_REGISTER      0b01001001001000000000000000000101
#define W_DIP_CYC_REGISTER      0b01001000101100000000000000001010
#define W_SWELL_CYC_REGISTER    0b01001000110000000000000000001010
#define W_EP_CFG_REGISTER       0b01001011000000000000000000100001
#define W_PGA_GAIN_REGISTER     0b01001011100100000000000000000000
#define W_WTHR_REGISTER         0b01000010000000000010000011111111
#define W_VARTHR_REGISTER       0b01000010000100000010000011111111
#define W_VATHR_REGISTER        0b01000010001000000010000011111111
#define W_CF1DEN_REGISTER       0b01001001010000000010110001111100
#define W_CF2DEN_REGISTER       0b01001001010100000010110001111100
#define W_CF3DEN_REGISTER       0b01001001011000000010110001111100
#define W_CF4DEN_REGISTER       0b01001001011100000010110001111100
#define W_EGY_TIME_REGISTER     0b01001011001000000000000111111111
#define W_AVAROS_REGISTER
    0b000000010000000000000000000000000000000000000000
#define W_STATUS1_REGISTER
    0b010000000011000000000011111111111111111111111111
#define W_STATUS1_CLR_DIP
    0b010000000011000000000011100000000000000000000000
#define W_STATUS1_CLR_SWL
    0b010000000011000000000000011100000000000000000000

#define W_DIPA_REGISTER
    0b010000010001000000000000011111111111111111111111

#define VoltageConstant         10.772735   //10.74705975
#define CurrentConstant         2.6299943   //2.683410674
#define PowerConstant           3.7854      //3.870674811
#define EnergyConstant          1.100991946
#define DipVal                  200
```



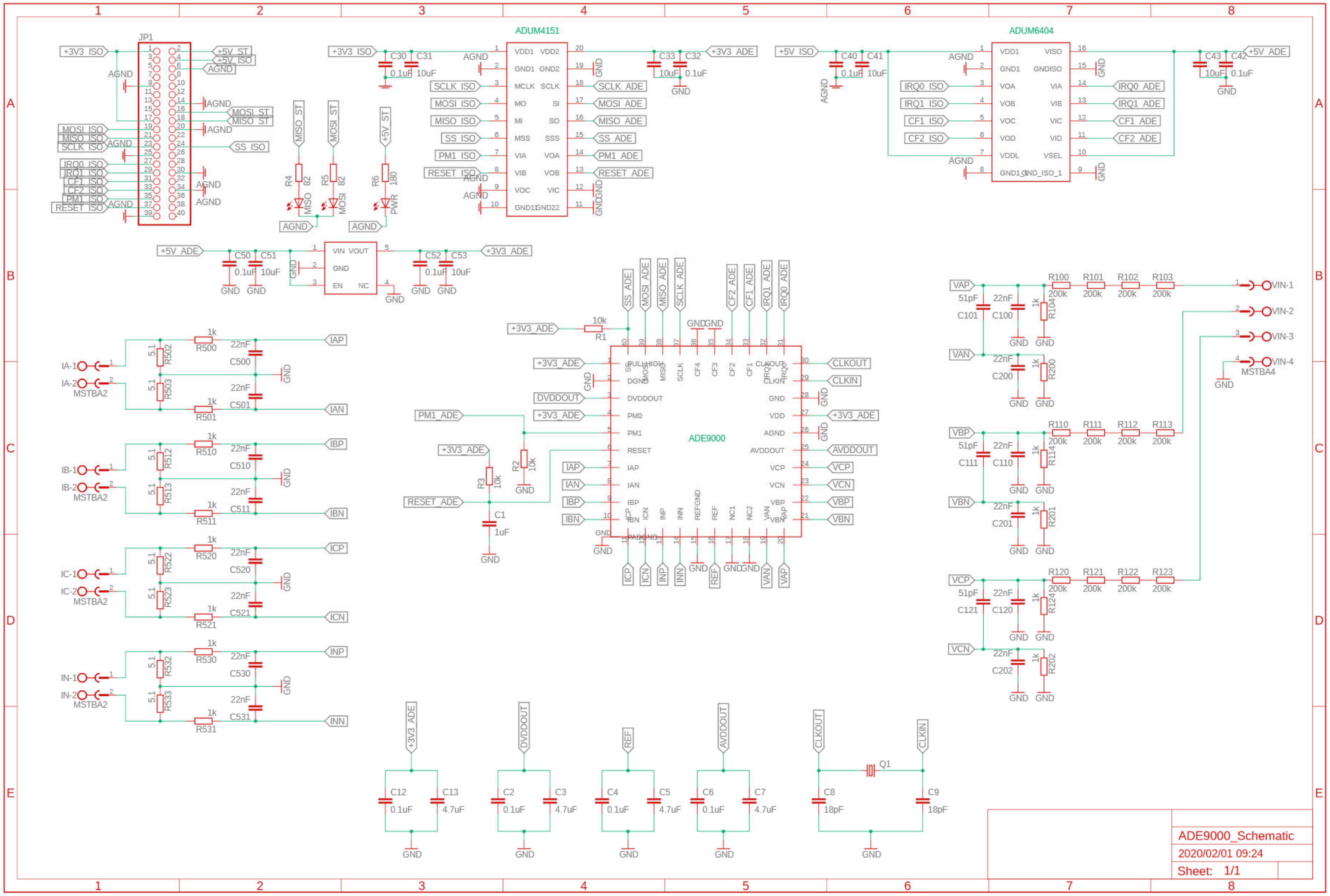
...Project\Project Source Code\Backup 20201103\ADE9000Comm.h 4

```cpp
#define PIN                         RPI_BPLUS_GPIO_J8_37
#define MOSI_ST                     RPI_V2_GPIO_P1_16
#define MISO_ST                     RPI_V2_GPIO_P1_18
#define IRQ_1                       RPI_V2_GPIO_P1_29


class ADE9000Comm {
    public:
        ADE9000Comm();
        ~ADE9000Comm();
        float readCurrent(unsigned short word);
        float readVoltage(unsigned short word);
        float readDipSwell(unsigned short word);
        float readPower(unsigned short word);
        float readEnergy(unsigned short word);
        float readTHD(unsigned short word);
        float readPowerFactor(unsigned short word);
        float readFrequency(unsigned short word);
        float readVarLVL(unsigned short word);
        float readRaw(unsigned short word);
        uint32_t readVarStatus();
        uint32_t readVarEvent();
        uint32_t readStatus(unsigned short word);
        void writeSPIlong(uint64_t word);

    private:
        int initSPI();
        void closeSPI();
        void resizeWord16(char byte[], unsigned short word);
        void resizeWord32(char byte[], unsigned int word);
        void resizeWord48(char byte[], uint64_t word);
        uint32_t parse32bitReturnValue(char byte[]);
        uint32_t parse32bitReturn24Value(char byte[]);
        uint16_t parse16bitReturnValue(char byte[]);
        void writeSPI(unsigned int word);
        //void writeSPIlong(uint64_t word);

};

#endif

```

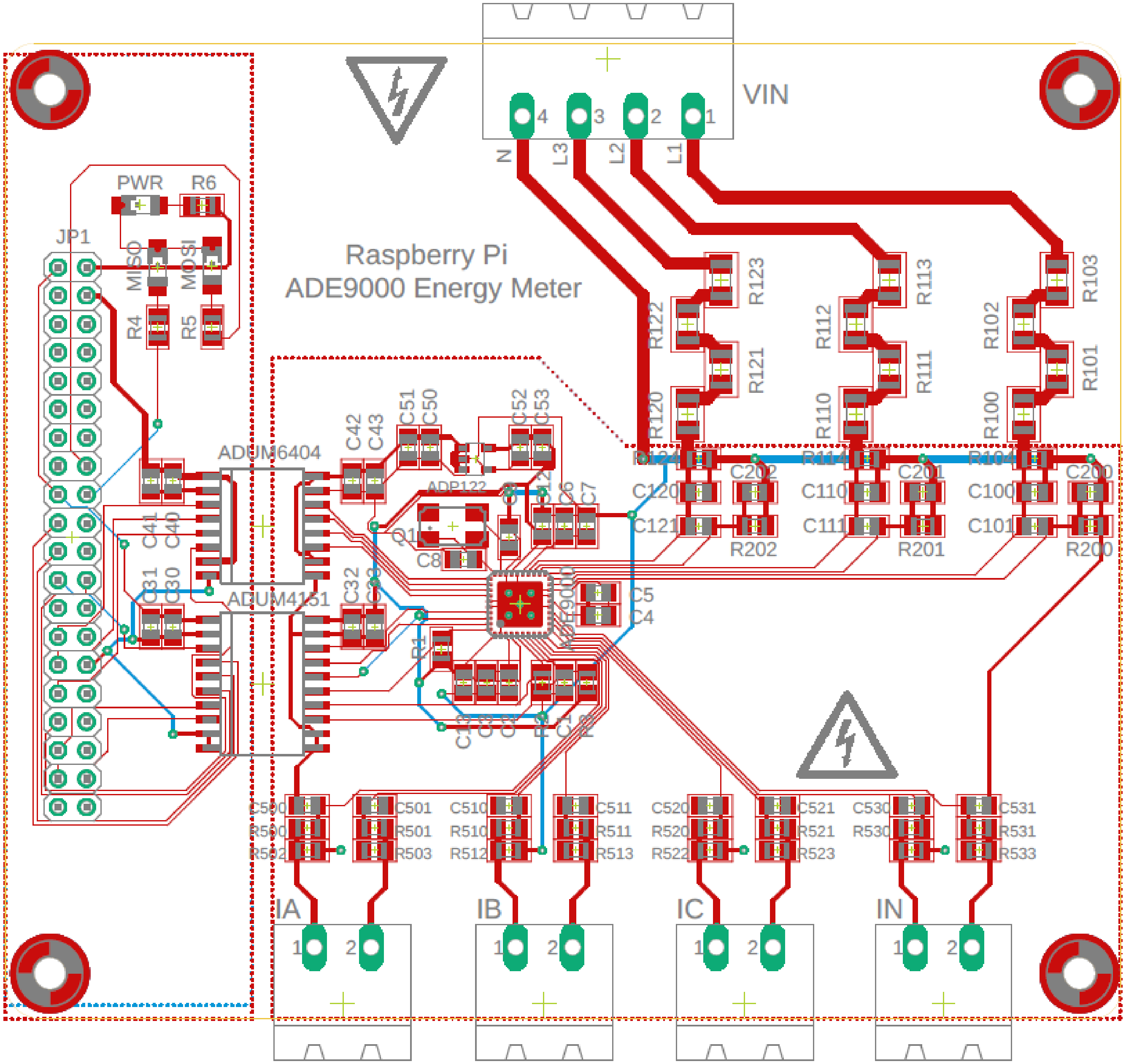
:

Appendix E: PCB Design Files for ADE9000 Data Acquisition Board

Appendix E1: Schematic Diagram (.sch file)



Appendix E2: PCB Design (.BRD file)


VIN
4
3
2
1
N
L3
L2
L1
PWR
R6
JP1
MISO
MOSI
R4
R5
Raspberry Pi
ADE9000 Energy Meter
R123
R122
R121
R120
R113
R112
R111
R110
R103
R102
R101
R100
R124
R114
R104
C202
C201
C200
C120
C110
C100
C121
C111
C101
R202
R201
R200
C42
C43
C51
C50
C52
C53
ADUM6404
ADP122
C41
C40
C31
C30
Q1
C8
C7
C6
ADE9000
C5
C4
ADUM4151
C32
C33
R1
C500
C501
R500
R501
R502
R503
C510
C511
R510
R511
R512
R513
C520
C521
R520
R521
R522
R523
C530
C531
R530
R531
R533
IA
IB
IC
IN
1
2