People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University M'Hamed BOUGARA – Boumerdès

Institute of Electrical and Electronic Engineering
Department of Power and Control

Final Year Project Report Presented in Partial Fulfilment of

the Requirements of the Degree of

'MASTER'

In Electrical and Electronic Engineering

Option: **Control Engineering**

Title:

Real-Time Monitoring of Smart Grid Using Synchronized Measurement Technology

Presented By:

- **MAOUCHE Nadine**
- **TAMACHE Rania**

Supervisor:

Dr. OUADI Abderrahmane

Registration Number:……../2024

Abstract

The aim of this project is to create a monitoring system employing a Phasor Data Concentrator (PDC) integrated with LabVIEW, in accordance with IEEE standards. The objective is to incorporate various PDC functionalities,illustrating the transfer of data streams from multiple Phasor Measurement Units (PMUs) to the PDC and subsequent storage.This system enhances grid performance and reliability through real-time monitoring, and precise, synchronized data management.Adhering to IEEE standards, the project ensures compliance for synchrophasor measurements and data handling. The PDC application is constructed using LabVIEW, harnessing its graphical programming features for efficient implementation and visualization. Essential toolkits are integrated to facilitate data handling and transmission. Furthermore, visualization tools within LabVIEW are created to exhibit real-time data and system status.The PDC effectively processes incoming data streams, aggregating and synchronizing data according to system requirements.In conclusion, the developed monitoring system utilizing PDC offers substantial advantages for grid monitoring and control, signifying improvements in grid reliability, stability, and efficiency within smart grid environments.

Dedication

" I first thank God for all the blessings then I would thank my family the ones who supported me along all these years. Thanks to my sisters Lynda and Zora, and a special thanks to my twin Nada who I love so much, and my mother, who raised me and brought me up and was there for me from the first time I stepped to school to the last one. I could never thank you enough. Thanks to my partner Nadine for the great time we spent together and the efforts she made to make this project works also thanks to my friends, each by their name and to the big family, to the ones who got happy for my success. And finally, I dedicate this work to myself , my mom and the soul of my dad. May Allah grant him paradise. Without you and your sacrifices, I wouldn't be here. "

TAMACHE Rania

Dedication

"First and foremost , I thank ALLAH for illuminating my path and giving me the courage to fulfill this project.I would like to dedicate my work to my beloved parents, whose unlimited support and boundless love have been the cornerstone of my journey. Their encouragement has propelled me forward, shaping every success I've achieved. To my brothers Haythem, his family, Rassim , my friends Rania and Omaima, who have all played significant roles in my life. I want to express special gratitude to my partner, Rania, for her constant encouragement during the challenging moments we faced while completing this project. Her dedication, hard work, and patience have been invaluable. This work is a tribute to the love, and support of each and every one of you. Thank you for being the pillars of my success."

MAOUCHE Nadine

Acknowledgements

Above all, we extend our heartfelt thanks to Allah, the Almighty, for His countless blessings. We are deeply grateful for the good health that sustained us, the courage that carried us through challenging moments, and the inspiration that guided us from the beginning to the end of this endeavor.

We would like to thank our project supervisor, Dr. OUADI whose guidance and support have been invaluable throughout the course of this research We extend our sincere thanks to all those who have supported me in this endeavor . Your encouragement and assistance have been instrumental to this project's completion. We would also like to express our special appreciation to the entire IGEE staff, particularly Dr. BENTARZI Hamid, for his invaluable support.

Contents

List of Figures

List of Tables

List of Abbreviations

AC Alternating Current

AGC Automatic Generation Control

CAM Computer-Aided Manufacturing

CHP Combined Heat and Power

CSV Comma-Separated Values

DBMS Database Management System

DGS-1210-28 D-Link Gigabit Smart Switch

EHL Event Handler Loop

EMS Energy Management System

FTP File Transfer Protocol

GPS Global Positioning System

HMI Human-Machine Interface

IEEE Institute of Electrical and Electronics Engineers

IEEEC37 IEEE Standard C37

IP Internet Protocol

LabVIEW Laboratory Virtual Instrument Engineering Workbench

LAN Local Area Network

MAC Media Access Control

MHL Message Handler Loop

MS Microsoft

NetBIOS Network Basic Input/Output System

NIST National Institute of Standards and Technology

NOS Network Operating System

ODBC Open Database Connectivity

OS Operating System

OSI Open Systems Interconnection

PDC Phasor Data Concentrator

PDC Programmable Logic Controller

PMU Phasor Measurement Unit

QMH Queued Message Handler

RAS Remote Access Service

RMS Root Mean Square

ROCOF Rate of Change of Frequency

SCADA Supervisory Control and Data Acquisition

SER Sequence of Event Recorder

SMTP Simple Mail Transfer Protocol

SQL Structured Query Language

TCP Transmission Control Protocol

Telnet Telecommunication Network

UDP User Datagram Protocol

UTC Coordinated Universal Time

WAMS Wide Area Monitoring System

Introduction

The integration of modern technologies into power systems has given rise to the concept of the Smart Grid, an advanced electrical grid that enhances the efficiency, reliability, and sustainability of electricity services. At the core of this evolution are Wide Area Monitoring Systems (WAMS), which utilize Phasor Measurement Units (PMUs) to provide real-time monitoring and control of electrical grids. PMUs provide high-resolution data on voltage, current, and frequency; this time-synchronized data collection enables precise monitoring and rapid response to grid disturbances. A critical component in this infrastructure is the Phasor Data Concentrator (PDC), which aggregates data from multiple PMUs, aligns the time-stamped data, and processes it for further analysis and storage. The PDC is responsible for ensuring that data from different PMUs, potentially spread over large geographical areas, is synchronized and accurately combined. This aggregated data is then used for real-time monitoring, analysis, and decision-making processes that are vital for the effective management of power systems. The PDC is used to enhance grid stability, improve efficiency and reduce losses, optimize renewable integration and also for better fault management.

This report explores the design and implementation of a PDC using LabVIEW.The primary objective of this research is to design a real-time monitoring system that ensures accurate data aggregation, reliable transmission, and secure storage of phasor data, while maintaining data integrity throughout the process. The research involves extensive testing of the PDC system to verify its performance under various conditions and scenarios. By achieving these goals, the system aims to enhance the monitoring capabilities and overall stability of the Smart Grid. The implementation of such a system is expected to provide significant benefits, including improved fault detection, enhanced grid stability, and better resource management.

The report is organized into four chapters. Chapter 1 provides the theoretical background on smart grids, WAMS, and PMUs in the context of modern grid management. Chapter 2 reviews existing studies, technologies, and methodologies related to PDC design and its applications. Chapter 3 outlines the procedures and methods used, including the design and implementation of the PDC and data handling processes. Chapter 4 describes the testing protocols and results for the PDC system, covering data frame construction, transmission accuracy, time alignment, data aggregation, decoding, visualization, and storage, with each test accompanied by a discussion of the findings and their implications for the Smart Grid and the effectiveness of the PDC system.

Chapter I

Theoritical Background

I.1 Introduction

This chapter explores how the Smart Grid, Wide Area Monitoring Systems (WAMS), and Phasor Measurement Units (PMUs) collaborate in modern grid management. The Smart Grid utilizes digital technology to improve grid performance, while WAMS enhances visibility and stability over large areas. PMUs play a crucial role in WAMS, offering synchronized measurements for real-time monitoring. By examining these components, the chapter aims to clarify their interconnected roles and contributions to contemporary grid operations.

I.2 The Smart Grid

I.2.1 Concept of smart grid

The Smart Grid didn't have a definite beginning; its evolution coincided with the development of electrical distribution systems. As time progressed, various needs arose, including the necessity for controlling, monitoring, pricing, and enhancing services related to power transmission and distribution. Typically, the implementation of Smart Grid technology aligns with the installation of smart meters. During the 1970s and 80s, these meters were utilized to relay consumer information back to the grid, illustrating a pivotal phase in the advancement of energy management systems[1].Even with the latest advancements, the top priority remains ensuring the reliability and efficiency of energy transmission and distribution via the electric power grid. Current research is also exploring expanding grid functions beyond transmission and distribution to include generating clean and sustainable energy, aiming to reduce greenhouse gas emissions and carbon footprints[2].

I.2.2 Definition

The idea of a Smart Grid involves various technologies, solutions that benefit consumers and responds to government regulations. It lacks a simple clear-cut definition. The European Technology Platform describes the Smart Grid as an electricity network that can integrate the actions of all users connected to it intelligently to deliver sustainable, secure, and cost-effective electricity supplies[3]. The US Department of Energy defines the Smart Grid as an electric power system leveraging digital technology to enhance efficiency, reliability, and security across the supply chain.

It encompasses large generation, energy delivery systems, storage resources, and electricity consumers[3]. Additionally, depicted in Figure I.1 [4] , the Smart Grid is described as a framework utilizing digital information and control technologies to facilitate the integration of renewable and distributed resources, electricity storage, peak-shaving technologies, smart consumer devices, and automated systems[5].

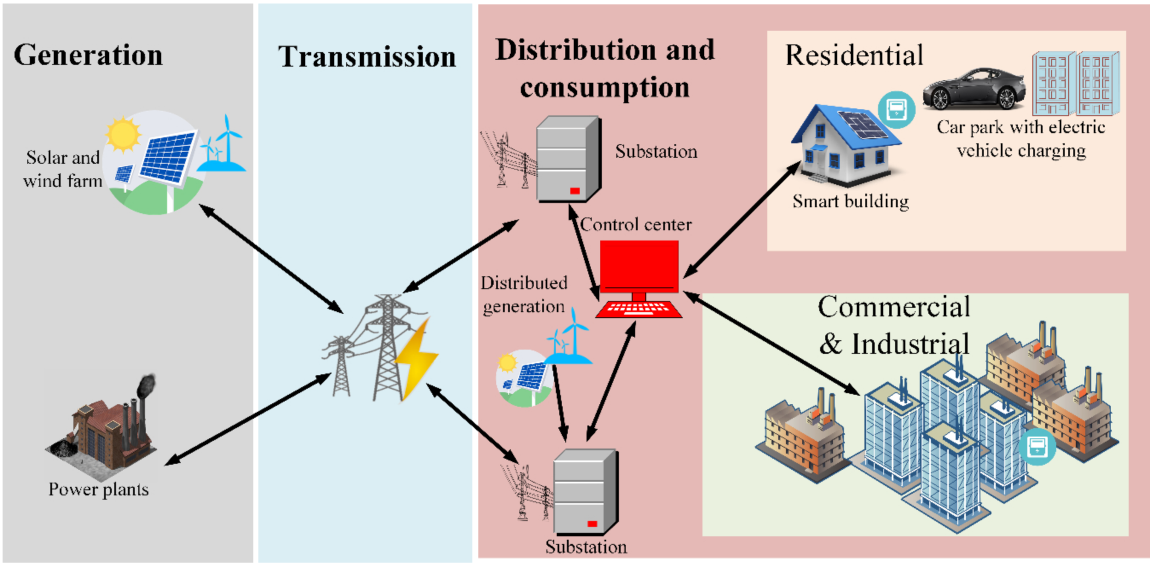


Figure I.1: The overall smart grid concept

I.2.3 Design of smart grid

To comprehend the design and idea behind the smart grid, it's important to see how it differs from the traditional power grid. The differentiation between the smart grid and the traditional power grid is depicted in TableI.3 [6].The design of the smart grid offers flexibility in terms of its usage and associated goals. The National Institute of Standards and Technology (NIST) presented a conceptual model of the smart grid, outlining the planning, development requirements, interconnected stakeholders, and necessary equipment[7]. NIST categorizes these stakeholders into seven domains for modeling, as depicted in Table I.2

Table I.1: Difference between smart and traditional power Grid.

Characteristics	**Power Grid**	**Smart Grid**
Customer participation	Static policies are deployed irrespective of real-time energy consumption from customers.	Dynamic policies are expected to be deployed while considering the real-time energy consumption from customers.
Communication facility	Only power line communication (PLC) is present.	Bi-directional communication facility is present, in which licensed and unlicensed frequency bands are in use.
Distributed energy generation	Few distributed generations such as solar and wind energy are considered.	All types of distributed generations such as solar and wind energy, and combined heat power (CHP) are taken into consideration.
Inclusion of storage devices	Not present.	Different distributed energy storage devices, which can be used in different situations are considered.
Real-time consumption monitoring	Not present.	Using bi-directional communication facility, real-time energy consumption can be monitored.
Security	No proper implementation is present to prevent energy theft and other types of security breach.	Adequate policies are taken into consideration to prevent energy theft and other types of security breach.

Table I.2: Stakeholders of Smart Grid.

Stakeholder	Description
Customer	Electricity is consumed by consumer. It may be domestic, commercial or industrial
Operations	Operations related to power systems are performed. It comprises of regulatory authorities or management responsible for the movement of electricity
Markets	Grid assets are used by stakeholders. Both operators and consumers are play role as market
Generation	Electricity is generated. Generation companies in bulk quantity of electricity are involved as player
Transmission	Electricity is transmitted. Companies or player responsible for transmission of generated electricity for distribution
Distribution	Electricity is distributed to end consumer and monitored. They include distribution of electricity from and to customers
Service Provider	Provide support services to all the stakeholders involved in generation, transmission and distribution of electric power

I.2.4 Architecture of Smart grid

A smart grid has a network that's spread out and organized in layers. It uses smart sensors to gather data about each part of the grid, helping to make smart decisions about how it operates [6]. The main parts of a smart grid include:

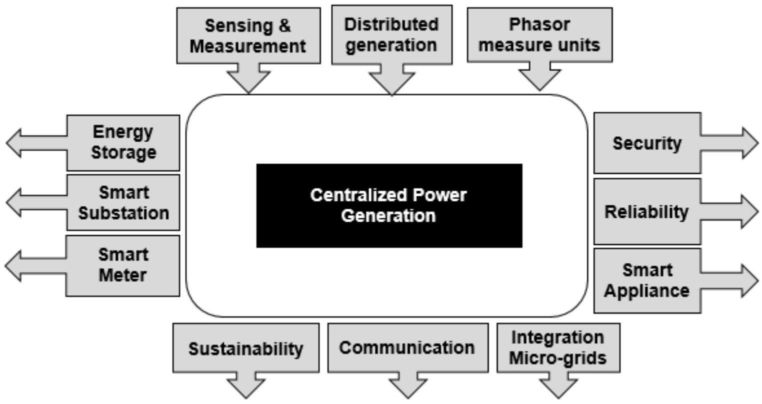


Figure I.2: Overview of Smart Grid architecture.

Energy Storage: Crucial for managing peak power demand and integrating renewable energy, with diverse storage systems like electrical, mechanical, chemical, and thermal.

Smart Meter: Facilitates accurate energy usage tracking and cost display through secure two-way communication networks.

Smart Substation: Supports real-time grid control, analysis, and monitoring, along with information collection and protection.

Distributed Generation: Includes small-scale energy production at consumer levels, improving affordability, reliability, and environmental impact.

Phasor Measurement Units (PMUs): Enhance grid stability and security by measuring voltages and currents with precision.

Sensing and Measurement: Utilizes advanced technologies for data collection, improving system management and reliability.

Smart Appliances: Incorporate cutting-edge technology to minimize peak energy demands, enhancing efficiency in household appliances.

Security: Focuses on proactive anomaly detection and data protection to safeguard against cyber threats and ensure grid reliability.

Reliability: Smart grids feature self-healing capabilities to address faults and enhance service reliability as grids expand and become more complex.

I.2.5 Advantages

- Assisting businesses in minimizing their environmental impact by decreasing carbon emissions.
- Opening doors to novel prospects for technology companies.
- Minimizing the expenses associated with power outages.

I.3 Wide area monitoring systems

I.3.1 Introduction

Electricity grids face challenges such as rising power demands, aging infrastructure, complex power transfers, and integrating renewable energy sources, all of which threaten grid reliability and stability. Traditional SCADA/EMS systems provide only a static view of the grid. Wide Area Measurement Systems (WAMS) enhance visibility and situational awareness using synchrophasor technology for real-time, accurate measurements, surpassing traditional SCADA methods. WAMS enables real-time analysis, allowing operators to address issues proactively, predict risks, and initiate preventive measures. This system supports reliability, real-time analysis, security calculations, threat detection, emergency state prediction, and restorative actions. WAMS also aids in post-disturbance investigations, offering a comprehensive understanding essential for maintaining grid stability amid increasing demands and renewable energy integration[8].

I.3.2 Definition

In a typical Wide Area Measurement System (WAMS), synchronized measurements from Phasor Measurement Units (PMUs) are collected and transmitted to a central Phasor Data Concentrator (PDC). The PDC analyzes this data to determine necessary preventive, corrective, and protective actions, aiding operators in making informed decisions for effective power system management. PMUs and PDCs are essential for WAMS, providing synchrophasor measurements and additional data like real and reactive powers. As the number of PMUs increases, efficient data collection and management become crucial for enhancing WAMS performance, reliability, monitoring, and decision-making[9],[10],[11],[12].

I.3.3 WAMS architecture

WAMS architectures can be categorized into centralized, distributed, and decentralized types[13]. These differ in the flow of information or data between where data is acquired, where decisions are made, and where actions are executed. The following sections provide detailed descriptions of each WAMS architecture type.

I.3.3.1 Centralized WAMS

In a centralized WAMS framework, PMU data collection, analysis, and execution of corrective measures occur at a single location. Phasor data from multiple substations is sent to a central PDC, where it is time-synchronized and consolidated. This data is then used for analytics and visualization, with the resulting remedial actions communicated back to the primary devices[14].

I.3.3.2 Decentralized WAMS

In a decentralized WAMS architecture, the wide area is divided into smaller sections, each controlled by a local PDC. PMUs send phasor data to their respective local PDCs for processing and local decision-making. These local PDCs can communicate with each other to address larger area issues. While this setup allows for localized control and protection, it can be inefficient for larger area monitoring. Coordinating and centralizing data acquisition from multiple local PDCs for comprehensive analysis poses challenges and often does not meet the desired objectives for extensive area monitoring[14].

I.3.3.3 Distributed WAMS

The Distributed WAMS architecture combines aspects of both centralized and decentralized architectures, featuring both local and central controllers. It operates as centralized control with decentralized execution, incorporating local PDCs at the substation or regional level, along with a master PDC at the central control station. PMUs in local areas send phasor data to their corresponding local PDCs, which are then connected to the master PDC at the central control station. The main distinction lies in the information flow: local PDCs may process PMU data locally while being supervised and controlled by the master PDC[14].

I.3.3.4 Comparison of SCADA and WAMS

Table I.3: Comparison SCADA and WAMS

SCADA	**WAMS**
Provides only steady-state information with low sampling density and non-synchronous data.	Allows synchronous observation of the power system on a more detailed time scale.
The control center cannot determine the dynamic operational states of the system. Immediate actions cannot be taken in case of failures.	Requires data to be transmitted and captured at a very high speed . [15]

I.3.3.5 Advantages of WAMS

- Enhanced Situational Awareness.
- Improved System Reliability.
- Accurate and Synchronized Data.
- Cost Savings[16].

I.4 Phasor Measurement Unit

I.4.1 A Short History Of The PMU

The concept of synchronized sampling emerged in the 1980s for designing protection systems, particularly for data samples from widely spaced substations. This led to the development of symmetrical component distance relays,from which the concept of PMUs originated. In 1988, PMUs were introduced, with the first industrial use at Macrodyne Co. PMUs have since become integral in transmission level projects and are now widely used in smart electronic devices in electric substations. A process of standardization by IEEE began, culminating in the first PMU standard in 1995 and the latest version in 2014. The IEEE standard allows manufacturers to choose design solutions, specifying only steady-state and dynamic test conditions and accuracy indices like total vector error. IEEE C37.118.1 introduces two performance classes: P-class for fast- response protection applications and M-class for high-accuracy measurement applications. Another major standard, IEEE C37.242, released in 2013, provides guidelines for PMU calibration, testing, and installation[17].

I.4.2 Definition

Phasor measurement unit(PMU) ,shown in Figure I.3 [18], is a device that produces phasors and measures frequency and ROCOF of an electrical signal, namely voltage and/or current in power networks , these measurements are tagged with the corresponding(analogous) instant of time using a synchronization time source which is GPS .

Figure I.3: Phasor Measurement Unit

It's conspicuous that the functional name of this device depends on the operations it performs so it varies from one to another. The main output of PMU is phasors so a description of this quantity is necessary for forward understanding of this concept [17], [9].

I.4.2.1 Phasor Definition

We can represent a sinusoidal signal in phasor representation which means magnitude and phase. This form is generally used in AC power system analysis. The equation of the AC sinusoidal waveform is defined in this equation I.1:

$$x(t) = X_m \ cos(t + \phi) \tag{I.1}$$

and phasor representation is defined in equation I.2:

$$X = (X_m \ /\sqrt{2})e^{j\phi} = (X_m/\sqrt{2})(cos\phi + jsin\phi) \tag{I.2}$$

$$= X_r + jX_i \tag{I.3}$$

where $X_m \ /\sqrt{2}$ is the RMS value which describes the magnitude and ϕ is the phase angle which depends on time scale at specific instant $t = 0$. From this equation we can substitute the rectangular form defined in equation I.3. It is crucial to recognize that this phasor is defined for the angular frequency ω. Evaluating it with other phasors requires using the same time scale and frequency[9].

I.4.2.2 Synchrophasor definition

The synchrophasor representation of the signal $x(t)$ in Equation I.1 is denoted by X in EquationI.2, where ϕ represents the instantaneous phase angle relative to a cosine function at the system's nominal frequency synchronized to Coordinated Universal Time (UTC).

According to this definition, ϕ is the deviation from a cosine function at the nominal system frequency synchronized to UTC. Since a cosine function reaches its maximum at $t = 0$, the synchrophasor angle is 0 degrees when the maximum of $x(t)$ aligns with the UTC second rollover (1 Pulse Per Second time signal), and -90 degrees when the positive zero crossing aligns with the UTC second rollover (sinusoidal waveform). Figure I.4 illustrates the relationship between the phase angle and UTC time[9].

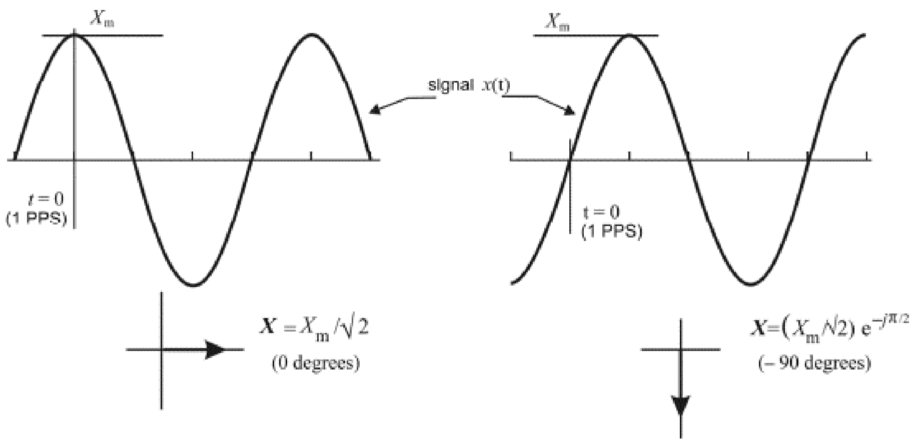


Figure I.4: Synchrophasor Representation

The sinusoidal waveform is represented by Equation I.4:

$$x(t) = X_m\ cos(\omega_0 t + \phi) = X_m\ cos(2\pi f_0 t + \phi) \tag{I.4}$$

where f_0 is the nominal system frequency (50 Hz or 60 Hz) directly indicated by the phasor in Equation I.2. In a more general scenario where the amplitude is a function of time $X_m(t)$ and the sinusoid frequency is also a function of time $f(t)$, we define the function $g(t) = f(t) – f_0$ where f_0 is the nominal frequency and $g(t)$ is the difference between the actual and nominal frequencies (note that g will also be a function of time, e.g., $g(t) = f(t) - f_0$).The sinusoid can be expressed as in Equation I.5:

$$\begin{aligned} x(t) = X_m(t)\ cos(2\pi \int f dt + \phi) = X_m(t)\ cos(2\pi \int (f_0 + g)dt + \phi) \\ = X_m(t)\ cos(2\pi f_0 t + (2\pi \int g dt + \phi)) \end{aligned} \tag{I.5}$$

The synchrophasor representation for this waveform is shown in Equation I.6:

$$X(t) = (X_m(t)/\sqrt{2})e^{j(2\pi \int g dt + \phi)} \tag{I.6}$$

For the special case where $X_m(t) = X_m$ is constant and $g = \Delta f$ is a constant offset from the nominal frequency, $\int g(t)dt = \int \Delta f dt = \Delta f t$ so the synchrophasor is simply as shown in Equation I.7:

$$X(t) = (X_m/\sqrt{2})e^{j(2\pi \Delta f t + \phi)} \tag{I.7}$$

that will rotate at the uniform rate Δf, the difference between the actual and off-nominal frequency[9].

I.4.3 PMU Architecture

The diagram shown in Figure I.5 represents the architecture of a Phasor Measurement Unit (PMU). Here is a brief explanation of each component and their roles in the architecture:

ure.

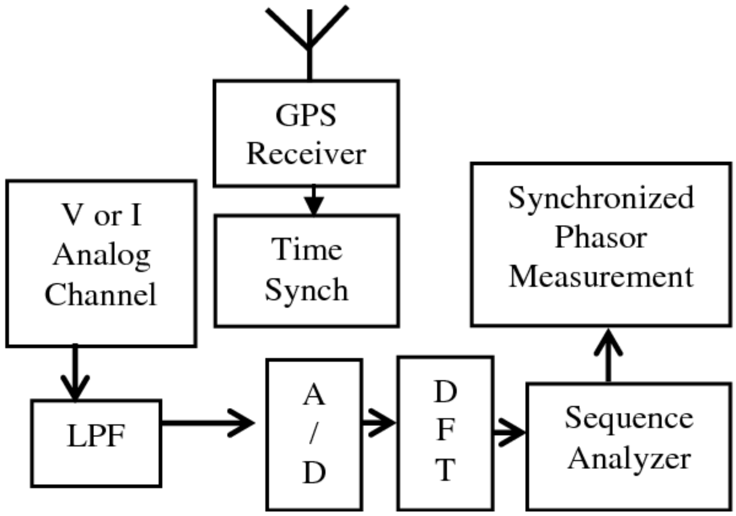


Figure I.5: PMU Architecture

- **V or I Analog Channel:** This component represents the input analog signals, which could be voltage (V) or current (I) measurements from the power system.

- **LPF (Low Pass Filter):** The analog signals are passed through a low pass filter to remove high-frequency noise and ensure that only the relevant signal frequencies are processed.

- **A/D (Analog to Digital Converter):** The filtered analog signals are then converted into digital form by an analog to digital converter, which allows further digital processing.

- **DFT (Discrete Fourier Transform):** The digital signals are processed using the Discrete Fourier Transform to convert the time-domain signal into its frequency components. This is crucial for phasor calculation.

- **Sequence Analyzer:** The frequency components obtained from the DFT are analyzed to determine the sequence components of the signal. This includes calculating the positive, negative, and zero sequence components, which are essential for understanding the power system's state.

- **Synchronized Phasor Measurement**: The analyzed data is then used to calculate the synchronized phasor measurements. These measurements include the magnitude and phase angle of the voltage and current waveforms, synchronized to a common time reference.

- **GPS Receiver:** The GPS receiver provides a precise time reference which is critical for synchronization. The GPS receiver ensures that the time stamps for the phasor measurements are accurate.

- **Time Sync:** This component synchronizes the phasor measurements with the GPS time reference, ensuring that all measurements are accurately time-stamped and can be compared across different PMUs in the network.

In summary, this architecture allows for accurate, time-synchronized measurements of electrical quantities in a power system, which is essential for monitoring, protection, and control purposes.

I.4.4 PMU Application

Phasor Measurement Units (PMUs) are widely used in North America, Europe, China, and Russia for various purposes such as post-disturbance analysis, stability monitoring, thermal overload detection, power system restoration, and model validation. These regions, along with India and Brazil, are also testing or planning to use PMUs for state estimation, real-time control, adaptive protection, and wide-area stabilization. Key potential applications of PMUs in power systems include enhancing static state estimation, fault location in transmission lines, emergency control during major a short disturbances, voltage control, and synchronized event recording[19]. The main applications of PMU are the following : state estimation, event identification and fault location, enhance Situation Awareness.

I.4.4.1 State estimation

The result of state estimation is crucial for various power system applications, including Automatic Generation Control (AGC), load forecasting, optimal power flow, corrective real and reactive power dispatch, stability analysis, and security assessment. Accurate and rapid determination of the system state is essential for secure power system operation. While SCADA provides different measurements (voltage magnitude, real and reactive power injection, etc.), PMUs are increasingly used due to their higher accuracy from synchronized measurements and high data transmission speed. PMUs provide precise measurements of magnitude and phasors for both bus voltage and branch current, which enhances state estimation. Using a sufficient number of PMUs ensures the observability of the entire system, simplifying the state estimation problem.

I.4.4.2 Event identification and fault location

Phasor Measurement Units (PMUs) are essential for event identification and fault location in power systems. They provide voltage/current phasor and frequency data that help identify disturbances using logic-based techniques like Sequence of Event Recorder (SER) or decision trees, which are suitable for real-time decision-making due to their fast execution. PMUs are also critical for quickly and accurately locating faults in transmission lines to minimize economic and social impacts. Effective fault location requires installing PMUs on every bus in the transmission network, making it important to determine the minimum number of PMUs needed to ensure complete network observability while considering installation costs.

I.4.4.3 Enhance Situation Awareness

In power system operations, managing large and complex data for various control operations is essential to maintaining system reliability and stability within specific constraints. As power systems become more interconnected, grid operators face the challenge of analyzing vast amounts of data and making critical, timely decisions under pressure. They must handle complex, dynamic data streams while staying aware of evolving operational conditions. Effective analysis and visualization tools based on PMUs can help operators maintain situational awareness and perform time-critical decision-making tasks in control centers.

I.4.5 IEEE standardization

The IEEE standard 037.118, established in 2005 and updated in 2011, facilitates the storage and transmission of time-tagged data in the power industry while ensuring compatibility across networks. It introduces two types of PMUs: M-class for steady-state measurements and P-class for protection functionality with rapid response. These PMUs seamlessly integrate with existing PDCs and visualization software, ensuring efficient and reliable operation within the power grid[20].

I.5 Conclusion

This chapter introduced the Smart Grid, WAMS, and PMUs in modern grid management. It defined each component and outlines their roles in improving grid reliability and efficiency. The next chapter will provide an in-depth exploration of the PDC, a key component of WAMS, to enhance understanding of grid management.

Chapter II

Phasor Data Concentrator

II.1 Introduction

The Phasor Data Concentrator (PDC) is vital in modern power systems for efficiently collecting, aggregating, and analyzing synchrophasor data, which provides real-time measurements of electrical quantities across the grid.This chapter explores the key functions, architecture, data transfer mechanisms, communication protocols, and data storage solutions of PDCs, highlighting their role in enhancing grid reliability and performance.

II.2 PDC Overview

II.2.1 The concept of PDC

The Phasor Data Concentrator (PDC) consists of three essential components. The first component is a central unit, which is a server-based computer that receives and gathers data from the Phasor Measurement Unit (PMU) through TCP/IP or UDP/IP communication.The PDC software application is responsible for managing, displaying, and recording the data received from the PMU. The second component is data storage, where the central unit collects data from the PMU and stores it in a database. Finally, the Human Machine Interface (HMI) serves as a display unit and controls the entire system, as depicted in Figure II.1 [21] .

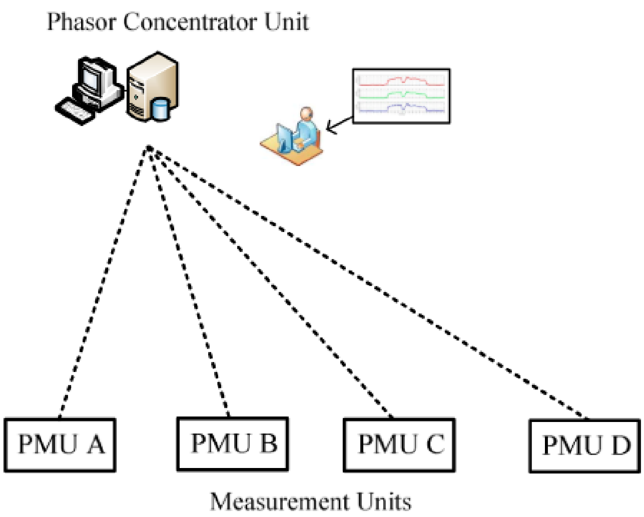


Figure II.1: Phasor Monitoring System

II.2.2 Definition

A Phasor Data Concentrator (PDC) functions as a node within a communication network, processing synchrophasor data from multiple PMUs or PDCs and transmitting it as a unified stream to higher-level PDCs or applications.The PDC organizes synchrophasor data based on timestamps to create a comprehensive set of measurements for the entire system[22].

The PDC also performs various additional functions, including:

- Conducting quality checks on phasor data and flagging any inconsistencies in the data stream.
- Monitoring disturbance flags and recording data files for further analysis.
- Monitoring the overall performance of the measurement system and displaying the results and performance recordings.
- Offering specialized outputs, such as direct interfaces to supervisory control and data acquisition (SCADA) systems or energy management systems (EMS).

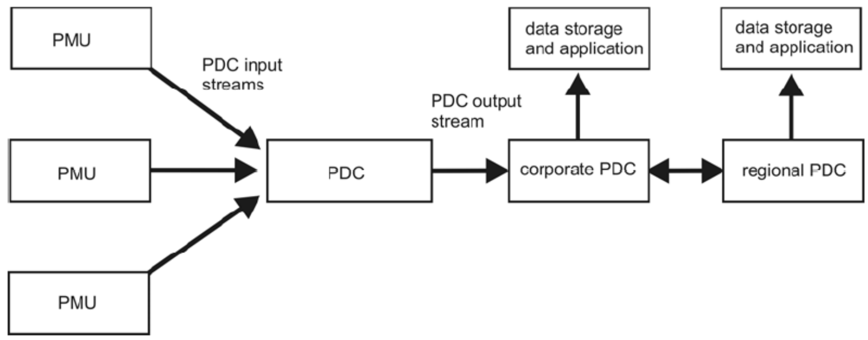


Figure II.2: Synchrophasor data collection network

Local PDCs shown on Figure II.2 collect and synchronize data from multiple PMUs and send it to applications. Mid- and higher-level PDCs gather data from these local PDCs, perform quality checks, and also forward the data to applications.PDCs can be integrated into other systems rather than functioning as standalone units.A hierarchical structure of PDCs supports different system levels and allows for peer-to-peer communication among utilities and control areas.Each layer addresses specific data needs, such as latency and quality.Backup and bypass mechanisms are necessary to mitigate potential failures in the data stream[22].

II.2.3 PDC functions

A PDC can perform various functions on synchrophasor data streams, depending on the requirements of the applications utilizing this data and the design of the synchrophasor systems.This section outlines the functional requirements for the following PDC functions:

- Data aggregation
- Data forwarding
- Data communications
- Data validation

- Data transfer protocol support
- Data transfer protocols conversion
- Data format and coordinate conversion
- Data latency calculation
- Reporting rate conversion
- Output data buffering
- Configuration
- Phase and magnitude adjustment
- Performance monitoring
- Redundant data handling
- Duplicate data handling
- Data re-transmission request
- Cyber security

The next section provides an in-depth analysis of the specific functions implemented in this project, which include the following:

- Data Aggregation
- Data Communication
- Data transfer protocol support

II.2.3.1 Data Aggregation

The aggregation of data can be performed with or without time alignment. It is crucial to maintain the quality of data, as well as indications of time quality and time synchronization for each signal.The output data frames should also include information regarding data quality assigned by the individual devices that transmit the data.

II.2.3.1.1 Data Aggregation with time alignment

Data aggregation with time alignment involves waiting for data with a specific timestamp from all sources, grouping that data into a packet, and forwarding it. The PDC aligns the received PMU/PDC data based on their timestamps, not their order of arrival or arrival time.Time alignment to absolute time entails waiting no longer than a specified absolute wait time after a timestamp for data with that timestamp, necessitating synchronization of the PDC to UTC.

II.2.3.1.2 Data Aggregation without time alignment

To balance the need for minimal latency and prevent data loss due to delayed arrival, a PDC can aggregate synchrophasor measurements without time alignment and transmit them periodically.This approach allows for user-defined transmission intervals or based on data size.In these transmissions, the PDC should identify each signal, despite the absence of specific requirements for time alignment, timestamps, data content, or sequence.

II.2.3.2 Data Communication

Data communication is crucial for a PDC, enabling connections with other devices via serial and Ethernet networks. It typically receives synchrophasor data, performs data alignment or other processing, and transmits the processed data to a data storage system or an application.Therefore, the ability to receive synchrophasor data through these networks is essential for a PDC's functions.

II.2.3.3 Data transfer protocol support

PMU data can be transmitted using various protocols such as IEEE Std C37.118.2-2011, IEEE Std C37.118-2005 and future protocols.A PDC should support receiving, interpreting, and transmitting data in at least one of these protocols.This will be detailed in the following subsection.

II.2.4 PDC Architecture

A synchrophasor data collection network is designed to capture and process high-speed measurements of the power system in real-time, offering improved insights into the system's dynamic behavior.In this network, PMUs are installed at various substations to gather measurements.These PMUs collect data at a very high frequency, usually 50–60 samples per second, and transmit it to the PDC in real-time[23]. The PDC is tasked with gathering all PMU data and aggregating it in a time-synchronized manner.This time-aligned data, processed at the local PDC, is then transmitted to other applications for purposes such as local monitoring, archiving, visualization, control, and protection [24].The basic layout of the PDC has been described in Figure II.3 [24] .

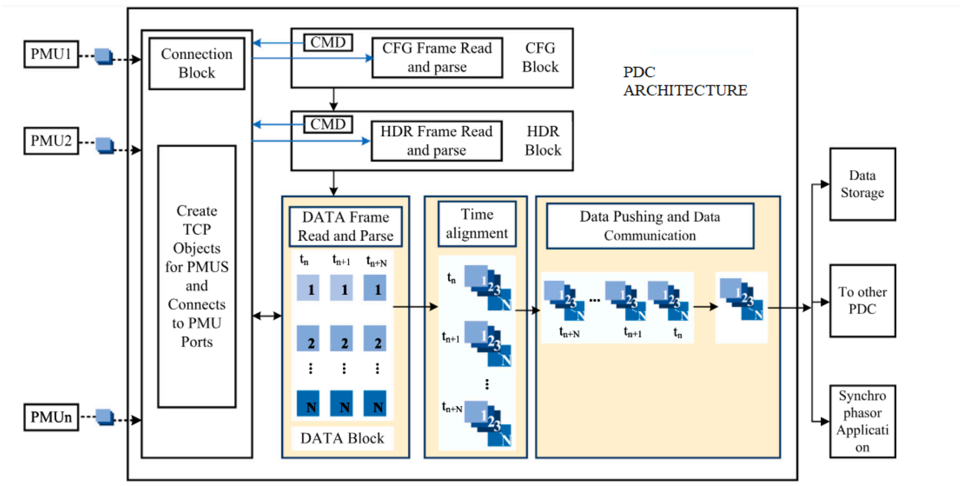


Figure II.3: PDC Architecture

The synchronized phasor data collected can be utilized by these applications to conduct real-time analysis and monitoring of the power system. Moreover, the local PDC shares its precisely synchronized data with the control center PDC. At the control center, the PDC combines data from multiple local PDCs, ensuring a consistent time reference, and forwards it to higher-level PDCs for additional analysis, storage, and monitoring purposes.These PDCs, situated at various hierarchical levels and geographic locations, establish communication over a wide area network to facilitate data exchange[23].

II.3 PDC Data Transfer

II.3.1 Definition of Data Stream

A PMU or PDC can transmit multiple data streams, each with different content, rate, and format, and each stream will have its own IDCODE for proper identification.Each stream operates independently, including the execution of commands and the transmission of data, header, and configuration messages.Information may be stored in any convenient form within the PMU/PDC, but when transmitted, it must be formatted as described in the specified frames.Any commands or other messages received that are not understood (due to unimplemented features, incorrect IDCODE, or bad CRC) will be silently discarded[25],[26].

II.3.2 Synchrophasor Message Format

II.3.2.1 Message Framework

The message framework of a Phasor Data Concentrator (PDC) is essential for the effective communication and processing of synchrophasor data in synchrophasor systems.It specifies the structure and format of the data messages exchanged by the PDC, which include crucial information such as timestamps, device identification, and data quality indicators, along with the synchrophasor data itself.The IEEE Standard C37.118.2-2011 outlines four types of message frames: configuration, command, data, and header.These frames are sent over the communication channel from the transmitter to the receiver and consist of several fields, including SYNC, FRAMESIZE, IDCODE, SOC, FRACSEC, and CHK, which are present in all message frames[25].

The four types of message frames are as follows:

- **Data messages:** Contain measurements made by a Phasor Measurement Unit (PMU).
- **Configuration messages:** Machine-readable messages describing data types, calibration factors, and other metadata for the data sent by the PMU/PDC.
- **Header messages:** Human-readable descriptive information sent from the PMU/PDC, provided by the user.
- **Command messages:** Machine-readable codes sent to the PMU/PDC for control or configuration.

II.3.2.2 Overall message

All message frames begin with a 2-byte SYNC word, followed by a 2-byte FRAMESIZE word, a 2-byte IDCODE, and a timestamp that includes a 4-byte second-of-century (SOC) and a 4-byte FRACSEC, which consists of a 24-bit FRACSEC integer and an 8-bit Time Quality flag. The SYNC word ensures synchronization and frame identification.The IDCODE uniquely identifies the source of data, header, or configuration messages, or the destination of a command message.The IDCODE links the data frames with their corresponding configuration and header information.Each frame concludes with a check word (CHK), which is a CRC-CCITT.

Frames are transmitted exactly as specified, without delimiters. Figure II.4 illustrates the transmission order of the frames: the SYNC word is sent first, and the CHECK word is sent last.All frame types follow this same order and format(Shown in Figure II.4). Table II.1 defines the words common to all frame types[25].

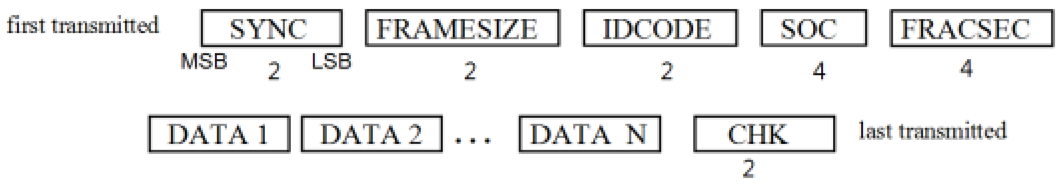


Figure II.4: Example of frame transmission order

II.3.3 Data Frame Format

A data frame contains measured data and is identified by having bits 4–6 in the SYNC word set to zero, as shown in Table II.2.The real-time phasor data frame consists of binary data ordered as outlined in Table II.1 and detailed in Table IV.1.All fields have fixed lengths, and no delimiters are used.As previously shown, the frame starts with SYNC, FRAMESIZE, IDCODE, SOC, and FRACSEC, and it terminates with a CRC-CCITT[25].

Table II.1: Word definitions common to all frame types

Field	**Size (bytes)**	**Comments**
SYNC	2	Frame synchronization word. Leading byte: AA hex Second byte: Frame type and version, divided as follows: Bit 7: Reserved for future definition, must be 0 for this standard version. Bits 6-4: 000: Data Frame 001: Header Frame 010: Configuration Frame 1 011: Configuration Frame 2 100: Configuration Frame 3 101: Command Frame (received message) Bits 3-0: Version number, in binary (1-15) Version 1 (0001) for messages defined in IEEE Std C37.118-2005. Version 2 (0010) for messages added in this revision, IEEE Std C37.118.2-2011.
FRAMESIZE	2	Total number of bytes in the frame, including CHK. 16-bit unsigned number. Range = maximum 65535
IDCODE	2	Data stream ID number, 16-bit integer, assigned by user, 1–65534 (0 and 65535 are reserved). Identifies destination data stream for commands and source data stream for other messages. A stream will be hosted by a device that can be physical or virtual. If a device only hosts one data stream, the IDCODE identifies the device as well as the stream. If the device hosts more than one data stream, there shall be a different IDCODE for each stream.
SOC	4	Time stamp, 32-bit unsigned number, SOC count starting at midnight 01-Jan-1970 (UNIX time base). Range is 136 years, rolls over 2106 AD. Leap seconds are not included in count, so each year has the same number of seconds except leap years, which have an extra day (86400s).
FRACSEC	4	Fraction of second and Time Quality, time of measurement for data frames or time of frame transmission for non-data frames. Bits 31–24: Message Time Quality as defined in 6.2.2. Bits 23–00: FRACSEC, 24-bit integer number. When divided by TIME_BASE yields the actual fractional second. FRACSEC used in all messages to and from a given PMU shall use the same TIME_BASE that is provided in the configuration message from that PMU.
CHK	2	CRC-CCITT, 16-bit unsigned integer.

Table II.2: Data Frame Organization

No.	Field	Size (bytes)	Comment
1	SYNC	2	Sync byte and frame type/version number
2	FRAMESIZE	2	Number of bytes in frame
3	IDCODE	2	Stream source ID number
4	SOC	4	SOC time stamp
5	FRACSEC	4	Fraction of Second and Time Quality
6	STAT	2	Bit-mapped flags
7	PHASORS	$4 \times$ PHNMR or $8 \times$ PHNMR	Phasor estimates
8	FREQ	2 / 4	Frequency
9	DFREQ	2 / 4	ROCOF
10	ANALOG	$2 \times$ ANNMR or $4 \times$ ANNMR	Analog data
11	DIGITAL	$2 \times$ DGNMR	Digital status points
Repeat 6-11			Repeated for each PMU
12+	CHK	2	CRC-CCITT

II.3.4 Data stream decoding

PMU data is typically transmitted in binary format, requiring researchers to design algorithms to decode and extract relevant information from the encoded TCP packet. This involves handling various IEEE Std. C37.118 compliant data frames and accurately interpreting voltage, current, frequency, and phase angle measurements[26].

II.3.4.1 Data Frame Decoding

Figure II.5 illustrates the decoding of data frames.The received data frames are decoded using the configuration frames, which provide information about the PMUs/PDCs and the data formats.The decoder function identifies the SYNC field value of the data frame, which is AA01 hex, and then extracts the relevant data fields, including phasor values, frequency, and ROCOF, among others.

The parsed data frames are stored in a separate data buffer for each PMU/PDC.The buffer depth is a user-configurable parameter, and all PMU data buffers have the same depth.The depth of the buffer determines the number of data frames that can be stored.The data buffer stores the parsed data frames based on the arrival timestamp[26].

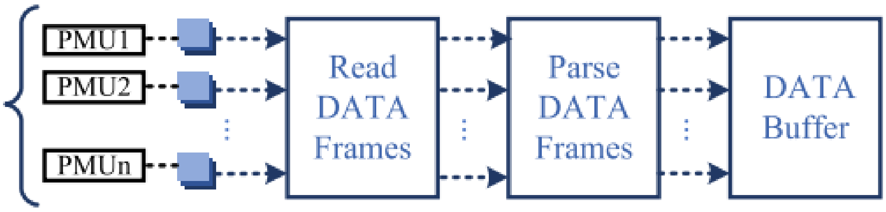


Figure II.5: Multiple PMUs data frame processing

II.3.4.2 Combined data frame

When the Phasor Data Concentrator (PDC) transmits data streams from different PMUs, it combines them into a single frame encoded in a general format, which includes data from all PMUs connected to that PDC.The STAT field is used to indicate the status of the data.For example, if PMU frequencies are stable and phase differences remain unchanged, the PDC modifies the corresponding PMUs STAT from (00) to (90), excluding the measured data when generating the combined frame for transmission.As shown in Figure II.6, the combined frame only includes synchrophasor data from nodes classified as abnormal (STAT: 80).If a data frame is lost or delayed, the node status is classified as unknown (STAT: A0) until two consecutive data frames are available in the PDC buffers.Alternatively, with additional buffers, classification can use data from the two nearest time slots stored in the buffers.Once the next-level PDC receives the combined frame from regional PDCs, it can identify abnormal nodes and the lines or feeders connecting them.Consequently, the regional PDC can initiate necessary actions such as islanding, injecting reactive/real power, or other protective measures. Estimating the power factor (PF) at the local PDC can also help control power[27].

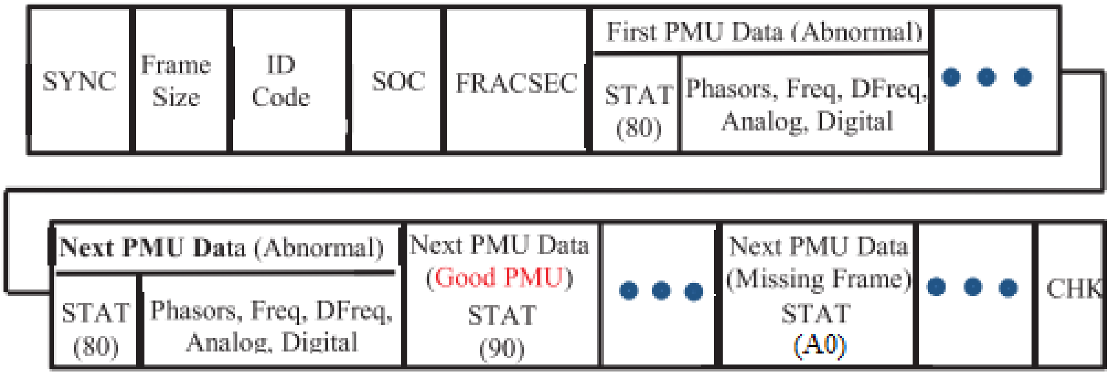


Figure II.6: Structure of a combined data frame for upper PDC

II.4 PDC Communication Protocols

II.4.1 What's communication networking

Communication networks facilitate the exchange of data between end-systems, supporting various services and applications. They enable seamless connectivity for tasks like messaging, streaming, video calls, file sharing, and accessing online services.These networks are vital for modern society, enabling businesses, individuals, and organizations to collaborate and stay connected globally[28].

II.4.2 OSI Model

The OSI model consists of seven distinct layers as illustrated in Figure II.7 [29], each containing different protocols within various protocol suites, with TCP/IP being the most common.This model helps categorize the processes involved in TCP/IP transactions.It is extremely useful for network installation, configuration, maintenance, and especially for troubleshooting network issues[30].

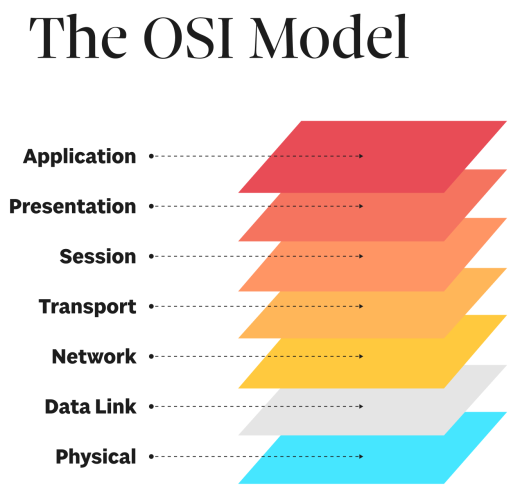


Figure II.7: OSI model

II.4.2.1 OSI model layers

- **Layer 1 (Physical Layer):**The Physical Layer is the physical and electrical medium for data transfer, including cables, jacks, hubs, and more.It is also known as the physical plant.Concepts include topologies, analog vs. digital encoding, bit synchronization, baseband vs. broadband, multiplexing, and serial data transfer.If you can touch it, it's part of this layer.The unit of measurement is bits.

- **Layer 2 (Data Link Layer):** The Data Link Layer establishes, maintains, and decides how transfer is accomplished over the Physical Layer.Devices on this layer include network interface cards and bridges. It ensures error-free transmission under LAN transmissions using MAC addresses.Any device that makes a physical connection to the network is on this layer.The unit of measurement is frames.

- **Layer 3 (Network Layer):** The Network Layer routes and switches information between different networks, LANs, or internetworks. Devices on this layer include routers and IP switches.This layer deals with logical addressing (IP addresses) and ensures compatibility between physical (MAC) and logical addresses.The unit of measurement is packets.

- **Layer 4 (Transport Layer):** The Transport Layer ensures error-free transmission between hosts through logical addressing.It manages message transmission through Layers 1-3, breaking up messages, sending them, and ensuring correct reassembly.It contains both connection-oriented and connectionless systems.The unit of measurement is segments or messages.

- **Layer 5 (Session Layer):** The Session Layer governs the establishment, termination, and synchronization of sessions within the OS over the network and between hosts.It controls the name and address database for the OS or NOS. NetBIOS works on this layer.

- **Layer 6 (Presentation Layer):** The Presentation Layer translates data format from sender to receiver in various operating systems.It includes code conversion, data compression, and file encryption. Redirectors, like mapped network drives, work on this layer.

- **Layer 7 (Application Layer):** The Application Layer is where message creation and packet creation begin.It includes end-user protocols like FTP, SMTP, Telnet, and RAS. It's not the application itself, but the protocols initiated by this layer [30].

II.4.2.2 Understand Layer 2 Switching

The Data Link layer is also where Layer 2 switches reside.A Layer 2 switch is the most common type of switch that is used on a LAN.They are hardware-based as shown in Figure II.8 [31] and they use the MAC address of each host computer's network adapter when deciding where to direct frames of data; every port on the switch is mapped to the specific MAC address of the computer that physically connects to it.Layer 2 switches do not normally modify frames as they pass through the switch on their way from one computer to another. Switches have memory that is set aside to store the MAC address to a port translation table, known as the MAC table or Content Addressable Memory table (CAM table). This table can be compromised with a MAC flood attack.This sends numerous packets to the switch, each of which has a different source MAC address, in an attempt to fill up the memory space on the switch.If this is successful, the switch changes state to what is known as fail open mode .At this point, the switch broadcasts data on all ports the way a hub does[30].

Figure II.8: D-Link DGS-1210-28 swicth

II.4.3 Define UDP communication

The User Datagram Protocol (UDP) is a lightweight data transport protocol operating on top of IP, designed for straightforward data transfer between two computers in a network. UDP sends packets, called datagrams, directly to a target computer without establishing a connection, specifying packet order, or verifying arrival. This makes UDP faster but less reliable compared to TCP, which requires a connection handshake, specifies packet order, and confirms packet arrival. Due to the lack of these mechanisms, UDP can transfer data more quickly than TCP, which can be advantageous in scenarios where speed is prioritized over reliability[32].

II.4.3.1 Communication framework of WAMS

Figure II.9 illustrates the architecture of WAMS communication systems. In this setup, PMUs are strategically placed within the power system to acquire synchronized phasor measurements of voltage, current, and phase angle. The PMUs transmit these synchronized measurements to PDCs over a dedicated communication network. The communication network of a synchrophasor system typically follows a client–server model, where the server is the data source, and the client receives the data. The server can be a PDC or a PMU, while the client can be a PDC or an application. In this model, the client initiates the connection and sends commands to regulate the data flow. The server responds to these commands and sends the requested data. The actual connections between the client and server can be established using TCP, UDP, or a combination of both. TCP connections are the easiest to set up, with the client sending a connection request to the server, after which commands and data are transmitted and received across the connection. TCP connections have the advantage of being state-aware, allowing for quick resolution of communication issues. UDP connections, on the other hand, are faster and have lower overhead compared to TCP. However, UDP connections do not maintain the connection state and do not provide error correction or retransmission of lost messages [26].

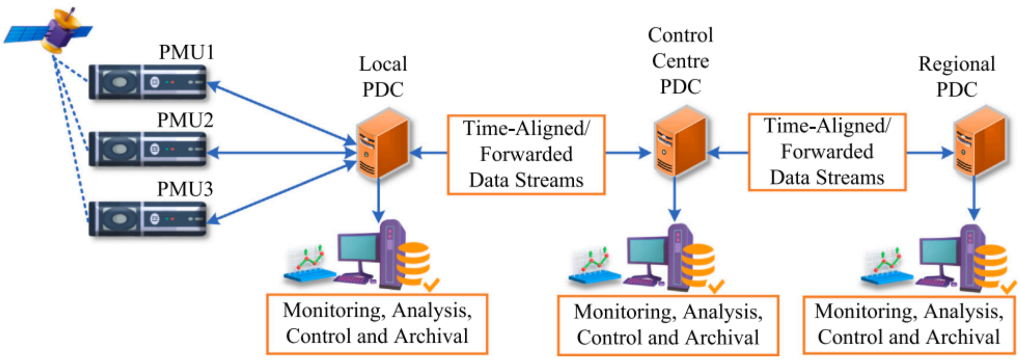


Figure II.9: WAMS system communication architecture

II.5 PDC Data storage

II.5.1 Introduction

Upon receipt of data or configuration frames, the PDC forwards them to a database server, which may be located either locally or remotely. The database server process involves the use of an IEEEC37.118 parser to parse configuration and data frames, creating objects in memory. This section aims to explore the nature of a database and its constituent elements.

II.5.2 What is a Database

A database is a collection of data stored electronically, usually organized and structured in some form. Databases can be very simple, such as a text file or a CSV file. When they are complex, formal methods of design and engineering principles are used to build, update, and maintain these systems. A Database Management System (DBMS) is the software that facilitates interaction between the database and an application that uses the database. Data in databases is typically stored as tables[33].

II.5.3 MS SQL Server

Microsoft SQL Server is a database management system (DBMS) created by Microsoft, which follows the principles of the relational model. In this model, data is structured in tables (known as relations), where these tables establish relationships with one another. Each table consists of rows and columns, representing respective attributes. MS SQL Server is a software tool employed for the administration and retrieval of information from the database[34].

II.5.4 What is an ODBC

The Microsoft Open Database Connectivity (ODBC) interface is a programming language interface in C enabling applications to retrieve data from various database management systems (DBMSs). ODBC serves as a performance-oriented, low-level interface tailored for relational data stores[35].

II.5.5 Database content

The database content encompasses several key elements essential for efficient data management shown in Figure II.10.

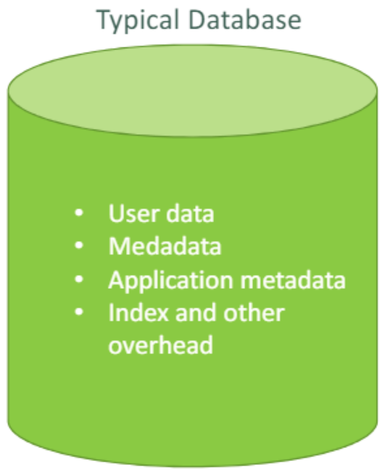


Figure II.10: Database Content

- User data stored in dedicated tables for structured organization and access.
- Metadata critical for maintaining data structure integrity, containing details like table and column names, types, and constraints.
- Application metadata further refines user data with application-specific details, customizing the user experience.
- Utilization of indexes and other overhead data to enhance system performance and streamline maintenance processes.

II.6 Conclusion

Phasor Data Concentrators are crucial for modern electrical grids, offering key functions in data aggregation, communication, and storage. This chapter examined PDC concepts, functions, architecture, and communication protocols, emphasizing the role of robust data storage in managing synchrophasor data. These insights underline PDCs' importance in maintaining efficient and reliable power systems. The next chapter will focus on designing a PDC using LabVIEW, showing practical implementations of these concepts.

Chapter III

PDC design using LABVIEW

III.1 Introduction

This chapter covers the use of LabVIEW software to design and implement a Phasor Data Concentrator (PDC). It highlights LabVIEW's graphical programming capabilities and essential tools for developing complex data systems.Key topics include LabVIEW's dataflow programming model, add-on toolkits, and the overall PDC design, focusing on PMU data transmission, data aggregation, time alignment, and applications like data storage and phasor display. By the end, readers will understand how to effectively use LabVIEW for creating efficient and reliable PDC systems.

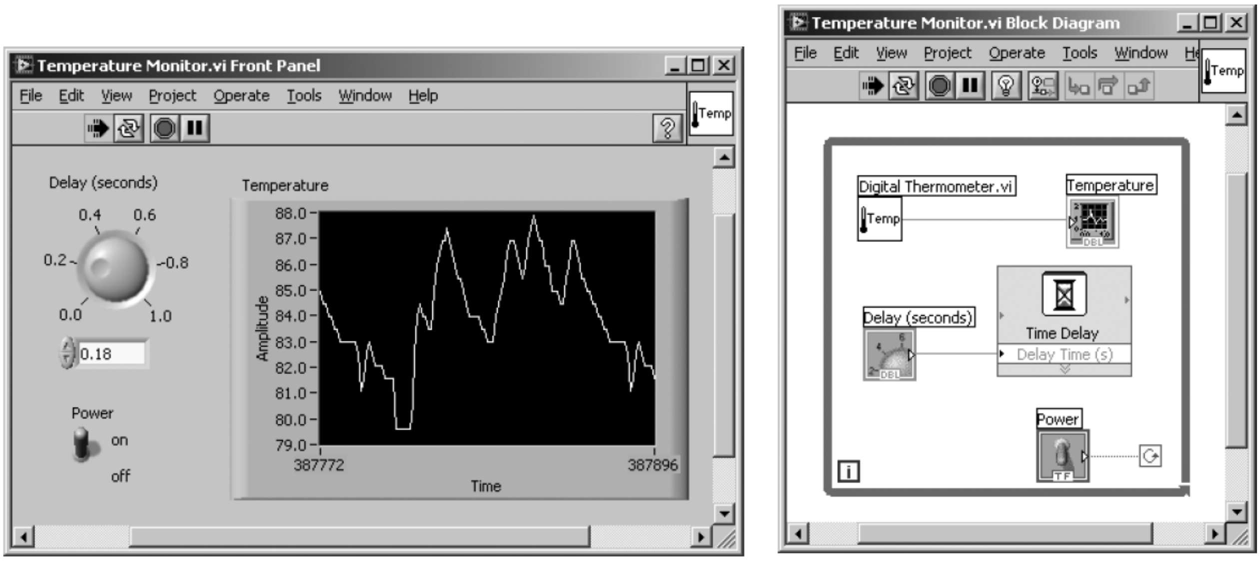
III.2 LABVIEW Software

III.2.1 Definition

Laboratory Virtual Instrument Engineering Workbench(LabVIEW), offers a graphical programming environment distinct from text-based languages like C or Java.It serves as an interactive system for developing and executing programs, particularly suited for professionals like scientists and engineers.Compatible with Windows, Mac OS X, and Linux systems, LabVIEW facilitates the development of programs through graphical notation, enhancing data flow via functional node connections[36].

III.2.2 Dataflow and Graphical programming

LabVIEW utilizes a dataflow programming language called G, characterized by graphical block diagrams where function nodes are interconnected by wires to facilitate variable flow. This structure enables parallel execution of multiple nodes, leveraging built-in schedulers to optimize hardware resources for efficient program execution. Graphical programming removes many of the syntactical concerns found in text-based languages, such as the placement of semicolons and curly braces. This approach allows you to focus on the data flow within your application, as the straightforward syntax makes the program's operations clear. Figure III.1 illustrates a simple LabVIEW user interface and the underlying code[36].



(a) User interface

(b) Graphical code

Figure III.1: LabVIEW Example

- **The Queued Message Handler(QMH):** combines aspects of both producer/consumer and event handler architectures. It consists of two loops: the Event Handler Loop (EHL) serves as the producer, dispatching messages through an event structure, while the Message Handler Loop (MHL) acts as the consumer, processing the received messages. User interface events trigger message queuing, and the QMH can facilitate feedback from the consumer to the producer using User Events.

III.2.3 LabVIEW Add-on Toolkits

LabVIEW offers a wide array of toolkits that simplify and expedite various tasks, making implementation easier for users. Some of the notable toolkits include:

- Application Builder
- Enterprise Connectivity Toolkit
- Internet Toolkit
- Database Connectivity Toolkit
- SPC (Statistical Process Control) Toolkit

In this project, several toolkits were utilized: data communication toolkit, SQL toolkit, electrical power toolkit.

III.2.3.1 Data communication toolkit

The Data Communication Protocol Toolkit includes various protocols, such as UDP/IP, TCP/IP, for facilitating communication between LabVIEW applications and external devices or systems. In this project, UDP communication was utilized. The specific blocks used for this protocol are shown in Figure III.2 and Figure III.3.

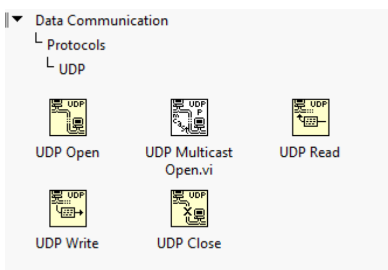
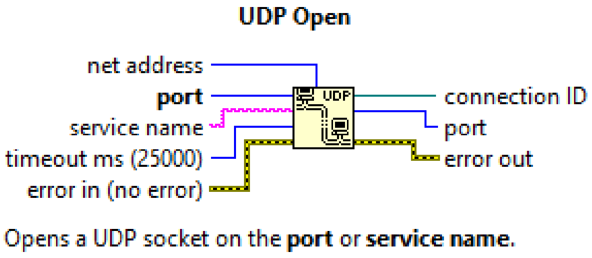
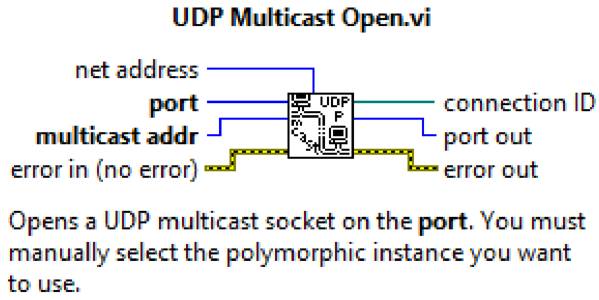


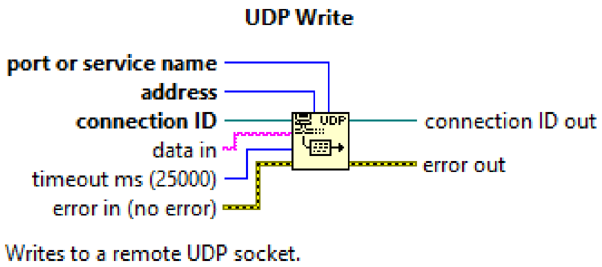
Figure III.2: UDP blocks in labVIEW



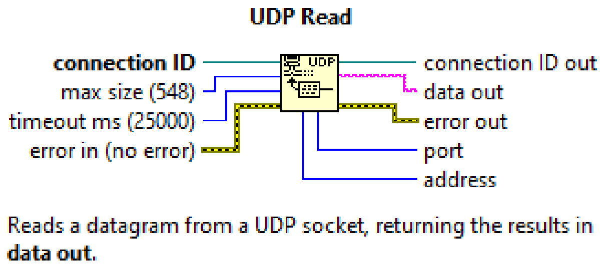
(a) UDP open Block



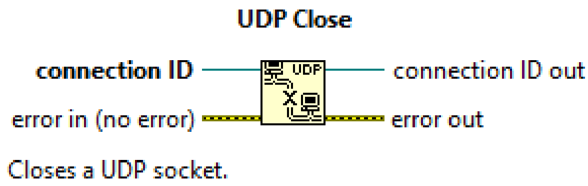
(b) UDP multicast open Block



(c) UDP write Block



(d) UDP read Block



(e) UDP close Block

Figure III.3: UDP block

- **UDP Open:** Initializes a UDP socket for communication.
- **UDP Multicast Open:** Initializes a UDP socket for multicast communication, allowing data to be sent to multiple receivers.
- **UDP Write:** Sends data over the UDP network to a specified IP address and port.
- **UDP Read:** Receives data from the UDP network on a specified port.

- **UDP Close:** Closes the UDP socket, ending the communication session.

These blocks enable efficient implementation of UDP communication within the LabVIEW environment, allowing real-time data exchange for the project.

III.2.3.2 SQL toolkit

The SQL Toolkit in LabVIEW provides functionality for interacting with SQL databases, enabling database operations within the LabVIEW environment.The specific blocks used for this protocol are shown in Figure III.4 and Figure III.5.

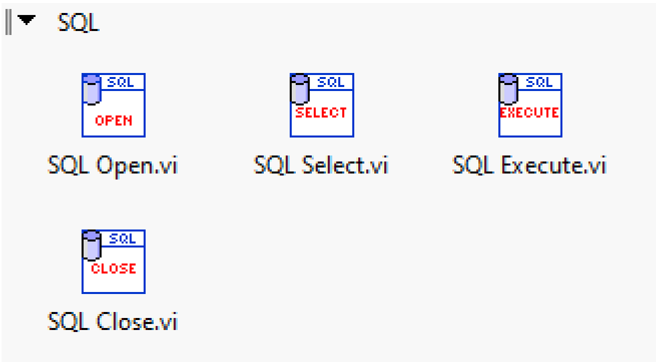
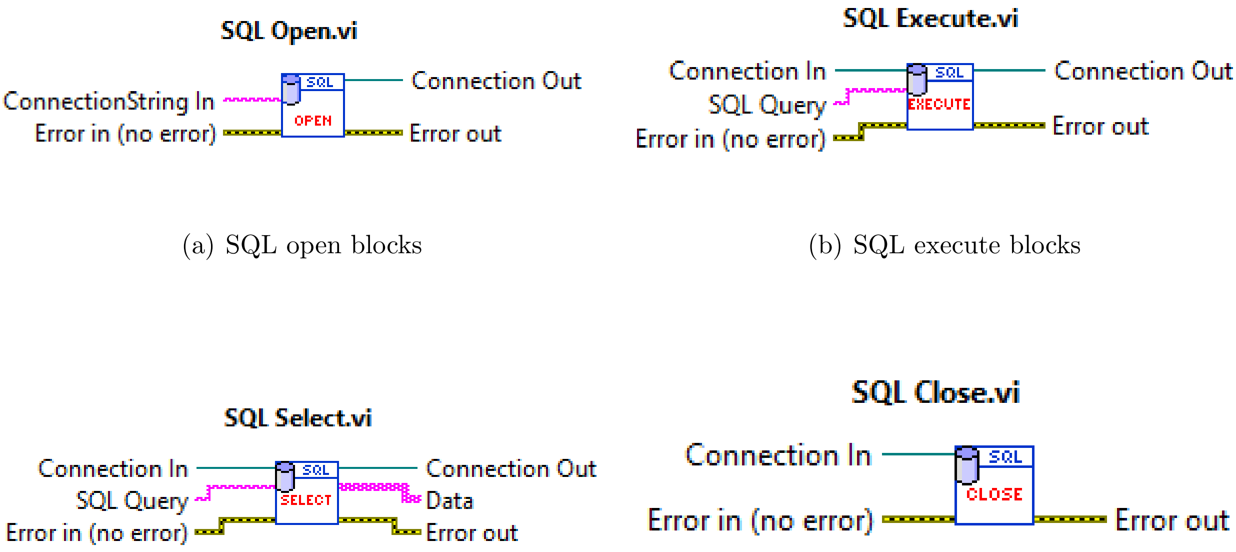


Figure III.4: SQL toolkit blocks

- **SQL Open:** Establishes a connection to a specified SQL database.
- **SQL Execute:** Executes SQL queries to insert, update, or delete data in the database.
- **SQL Select:** Retrieves data from the database based on the executed query.
- **SQL Close:** Closes the connection to the SQL database, releasing the resources.



(a) SQL open blocks

(b) SQL execute blocks

(c) SQL select blocks

(d) SQL close blocks

Figure III.5: SQL blocks

These blocks make it easy to handle database tasks in LabVIEW.

III.2.3.3 Electric power toolkit

In the Electric Power Toolkit, our application was confined to utilizing the vector diagram feature solely for display purposes shown in Figure III.6.

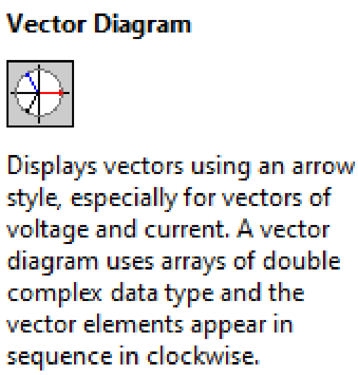


Figure III.6: Vector diagram block

III.3 Overall design of the PDC

The flowchart provided (See Figure III.7)illustrates the comprehensive process of data handling and analysis involving Phasor Measurement Units (PMUs) and Phasor Data Concentrator (PDC) and their integration into a system for real-time monitoring and power calculations. We first start by reading the data from an excel file to construct a binary data frame of PMU1 and PMU2, then send it to PDC via UDP communication. The PDC recieves the data and decode it to align the PMU2 data according to the SOC differnce that will be added as a phase shift to PMU2 phases, also the data is used to compare phases of both PMUs to determine the STAT and format of the combined data frame which will be then sent to differnt applications to be decoded.Two servers are used one to store and retrieve the decoded data and visualize it in waveform charts of differents periods and another one to display the data in phasor diagrams and calculate real and reactive power.

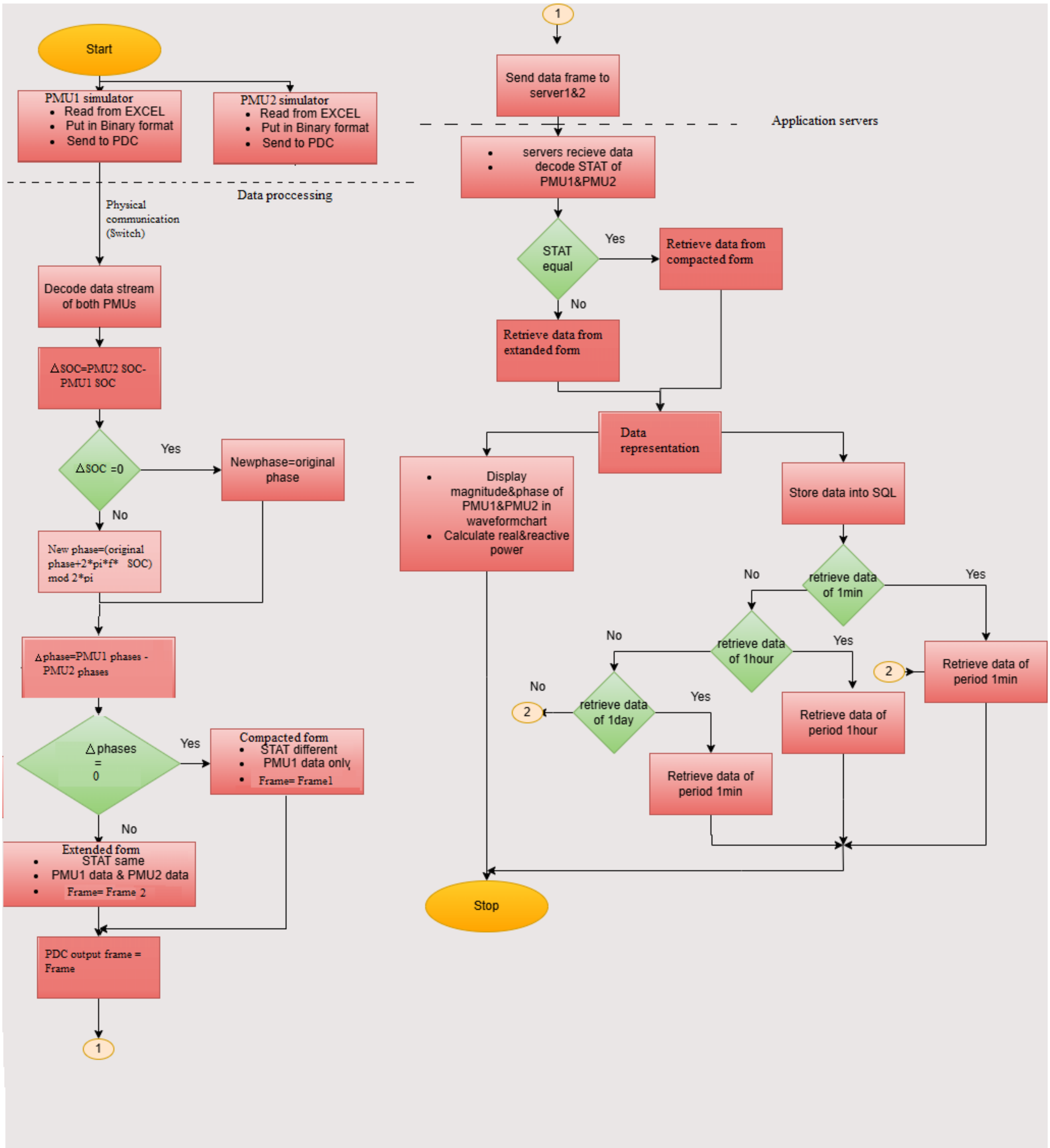
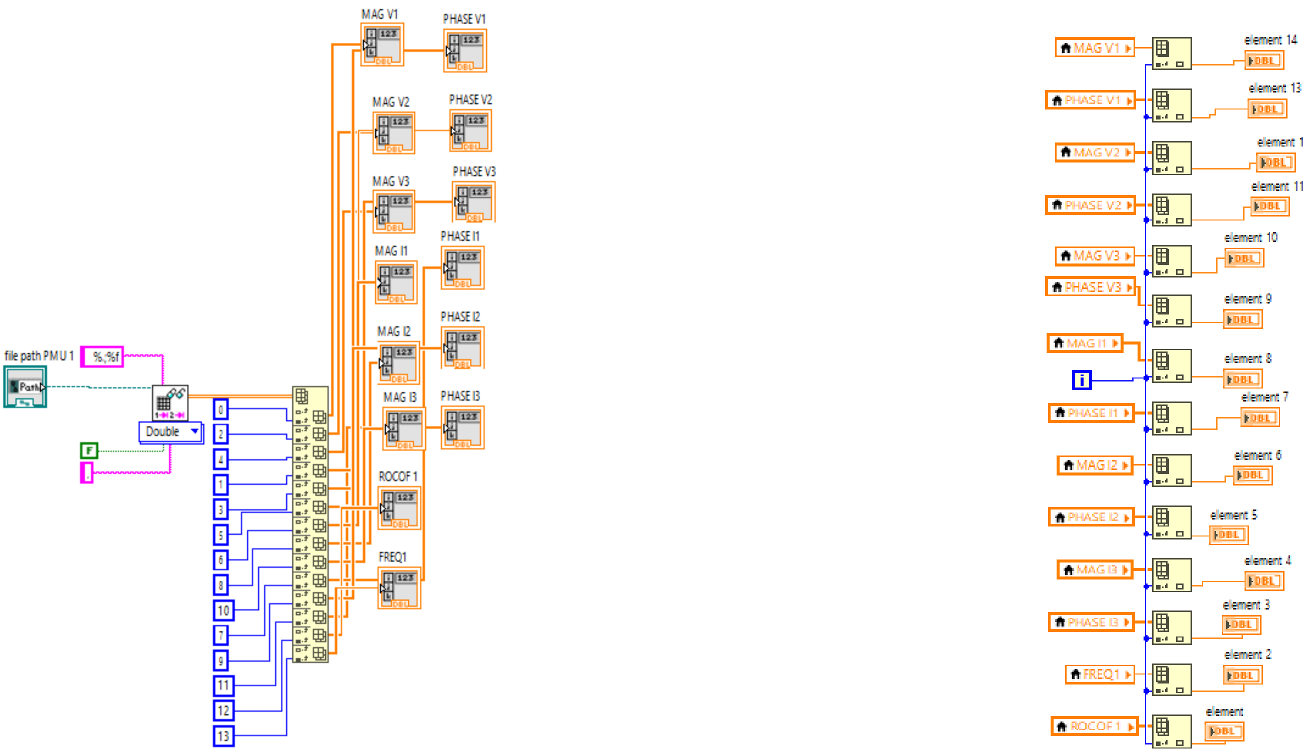


Figure III.7: Flowchart of the whole system

III.4 PMU data sending

III.4.1 Extracting excel elements

In this procedure, PMU data was generated and subsequently organized in an Excel file, as depicted in the accompanying figure. Following this, we employed the “Read Delimited Spreadsheet” block in LabVIEW to parse the Excel file.The output from this block yielded an NxM array. However, for our specific requirement, it was imperative to extract the elements from each column to convert them into binary format and arrange them as desired.To accomplish this task, an “Index Array” was implemented, ensuring the correct positioning of indexes corresponding to the column order in the Excel file(See FigureIII.8).



(a) Extracting columns (b) Extracting elements

Figure III.8: From Excel file to elements

Subsequently, a “For Loop” was employed to iterate over each element within the columns, utilizing local variables and another “Index Array” for efficient data extraction.

III.4.2 Convert to data frame

After extracting elements, the next step involves merging magnitudes with their corresponding phases for both 3-phase current and voltage.Phases are initially converted from degrees to radians multiplied by 1000(shown in FigureIII.9), aligning with the IEEE PDC standard. Afterward, the “Format Into String” function is utilized with the binary format %016b%016b to merge the values, resulting in a 32-bit string combining both magnitude and phase values in binary format.

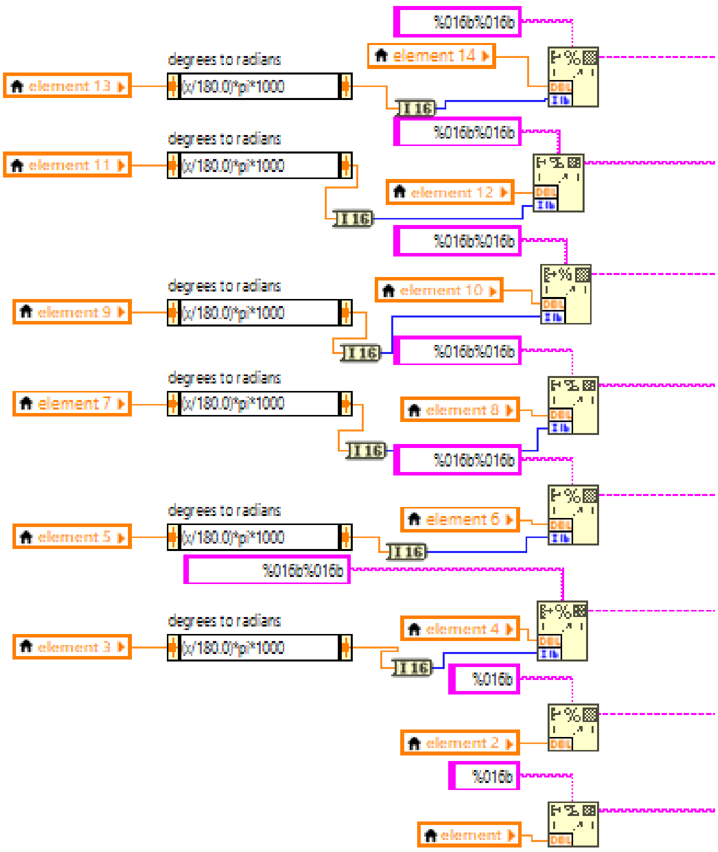


Figure III.9: Conversion to binary format

After converting all magnitudes and phases to binary, a data frame is constructed, respecting the order and number of bits for each word. This is achieved using the "Format Into String" block again(See FigureIII.10), ensuring that the data is formatted correctly for transmission to the PDC via the UDP/IP protocol(See Figure III.11).

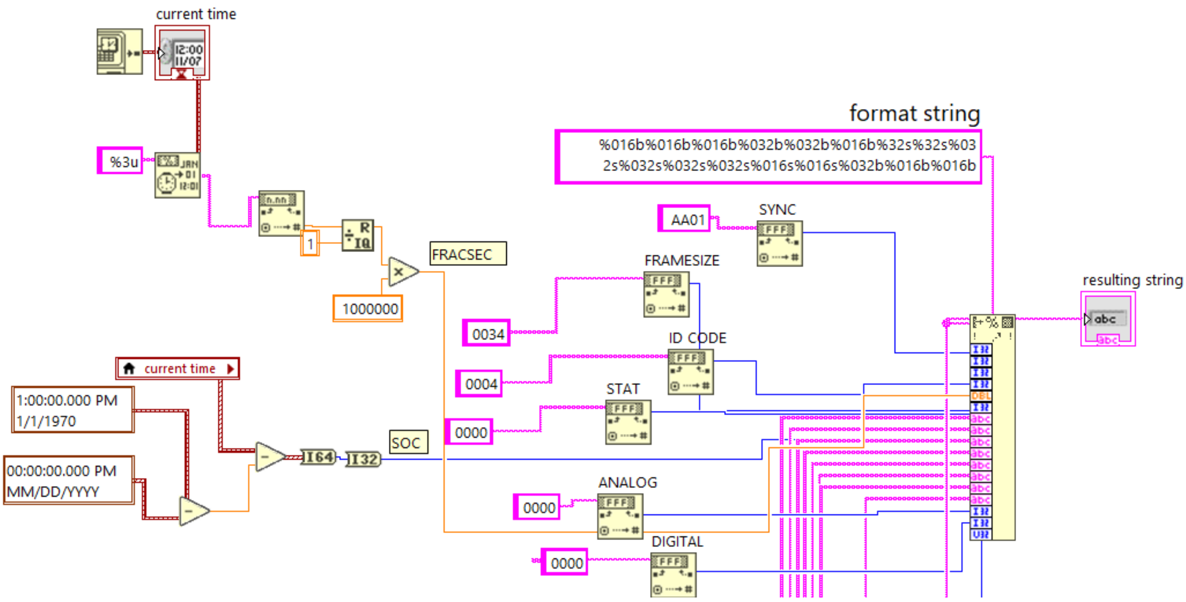


Figure III.10: Converting the data frame

The SOC and fracsec were implemented using straightforward logic blocks. The SOC represents the number of seconds elapsed since January 1, 1970, up to the present moment,The GET DATE/TIME IN SECONDS block in LabVIEW provides the number of seconds elapsed since 12:00 AM on 01/01/1904. To find the Unix time, subtract the timestamp of 01/01/1970 00:00:00.000 from the timestamp of 00:00:00.000 DD/MM/YYYY. while the fracsec denotes the fractional part of a second.SYNC is AA01 hex, and the frame size is 0034 hex. The ID code can be any decimal number. The resulting value is then subtracted from the output of GET DATE/TIME IN SECONDS function. This value is connected to a double block to change the value type, and then to the To signed 64-bit integer function. The output is the Unix timestamp.CHK is D4 3F hex.

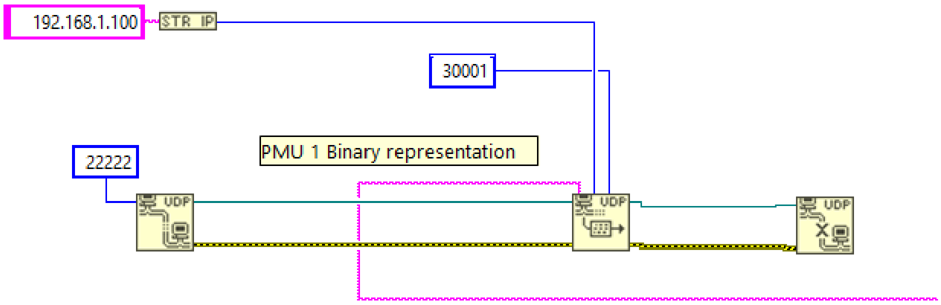


Figure III.11: Sending with UDP

Note : the loop mentioned above was duplicated on another PC. This duplication was necessary because the project involves monitoring data from two separate PMUs, each PMU has its own unique Excel file and operates through different ports.

III.5 Data aggregation and time alignment

III.5.1 The whole loop

This loop represents the PDC software where aggregation and time alignment functions are applied. The PDC will receive data frames from both PMU1 and PMU2 through UDP port numbers 30001 and 30002, respectively. The data for each PMU will be decoded from binary format to integer format using the SCAN FROM STRING block. The input is the binary representation of a specific PMU data frame. The format string depends on the data to be retrieved (refer to Table II.2), and the initial scan location depends on which element is to be retrieved.

III.5.2 Decoding received binary format

The data received from UDP is shown in the Figure III.12 .

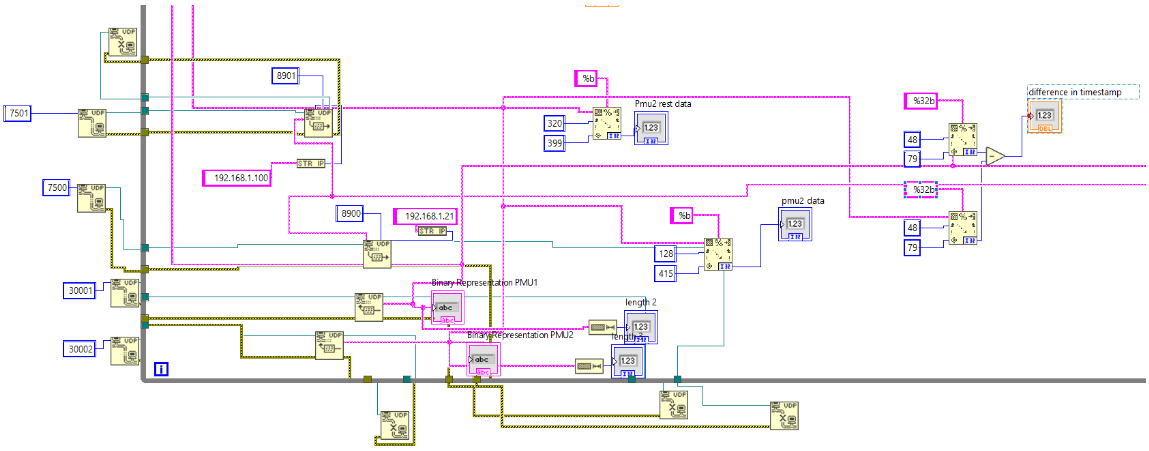


Figure III.12: Data recieved from UDP

It is then split using the SCAN FROM STRING block based on its position in the binary format and converted to decimal(See Figure III.13).This conversion enables us to carry out specific applications, which will be detailed in the following sections.

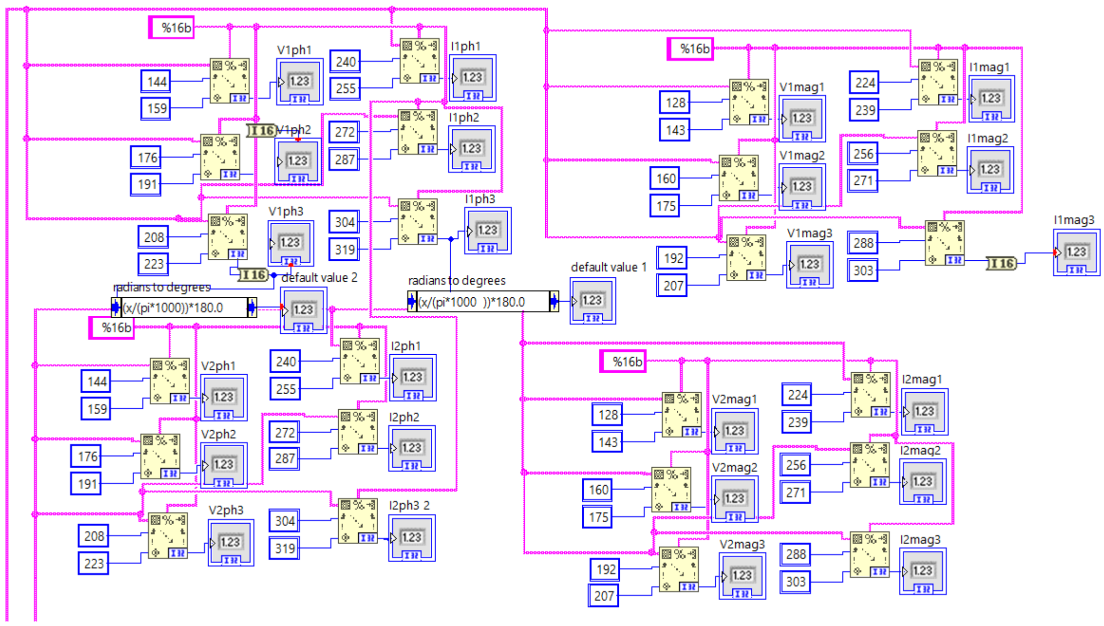


Figure III.13: Decoding binary format

To determine the STAT of the new aggregated data frame, phase decoding is necessary.Since the frequency is stable at 50Hz, only the phases of PMU1 and PMU2 are compared.Vphase1 of PMU1 is compared to Vphase1 of PMU2, and so on.The resulting boolean value shown in Figure III.14 is connected to the selector of the event structure.

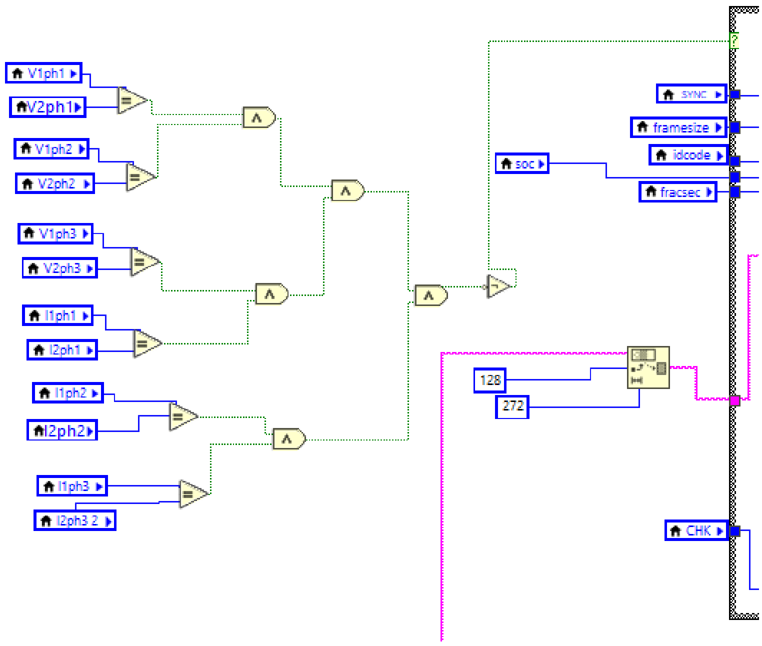


Figure III.14: Condition of case structure

III.5.3 Time alignment

For the alignment function, this occurs when there is a timestamp difference in the SOC of PMU1 and PMU2(See Figure III.15). When the PDC receives this information (SOC), the SOC of PMU1 is set as the time reference for PMU2 by default in our implementation. The resulting difference is added as a phase shift to PMU2 using the following equation:

$$NewPhase = (OriginalPhase + Additional\ Phase\ Shift) * mod360. \qquad \text{(III.1)}$$

Since the additional phase shift angles are periodic with 2π, the modulo operation is performed with 2π. We worked with degrees, so angles are periodic with 360 deg. An addition block is used to sum the additional phase shift with the original phase to obtain the new phase for PMU2 current and voltage phases(See Figure III.16).

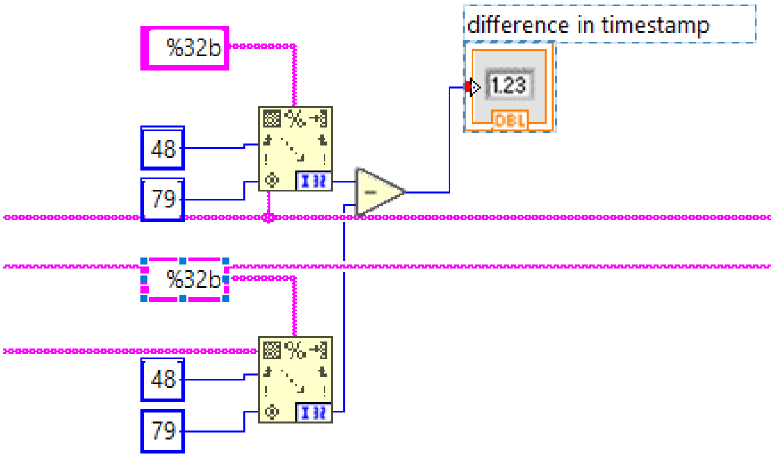


Figure III.15: Timestamp difference

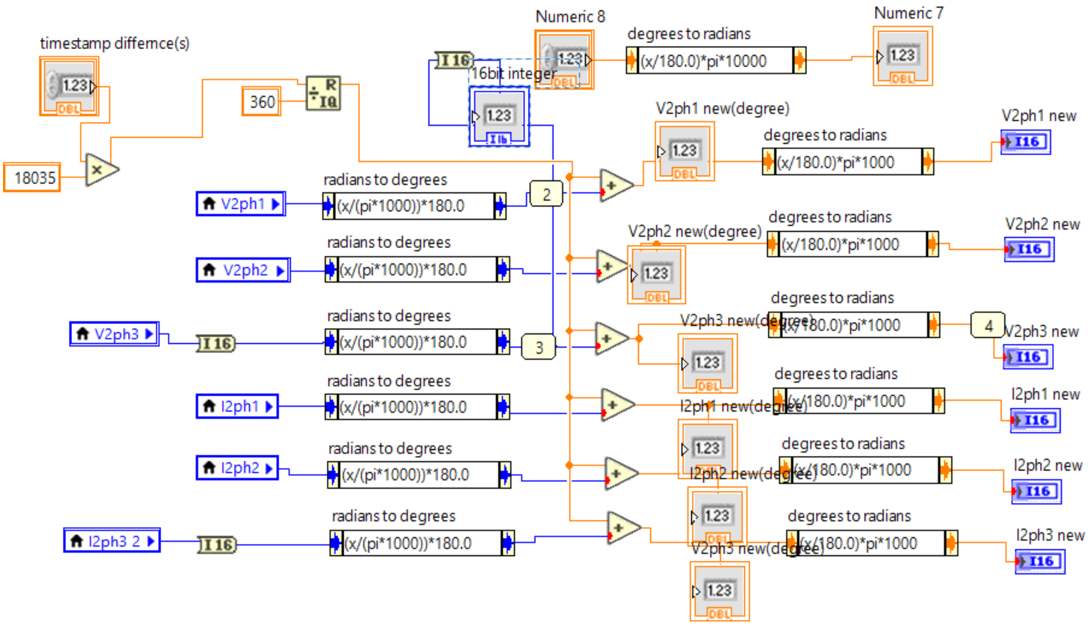


Figure III.16: Phase shifting

III.5.4 Aggregate PMUs data

This case structure includes two cases: TRUE and FALSE, each with a different structure. All data (SYNC, framesize, IDcode, SOC, fracsec, stat1, chk, rest data PMU1) is common for both cases.

III.5.4.1 False case

The FALSE case indicates that the phase difference between PMU1 and PMU2 is the same ,indicating that the STAT of PMU2 is 90 as illustrated in Figure III.17. The inputs are then connected to a FORMAT INTO STRING block with a binary format string of different sizes. Data coming from PMU1 is a string with a length of 272 bits.

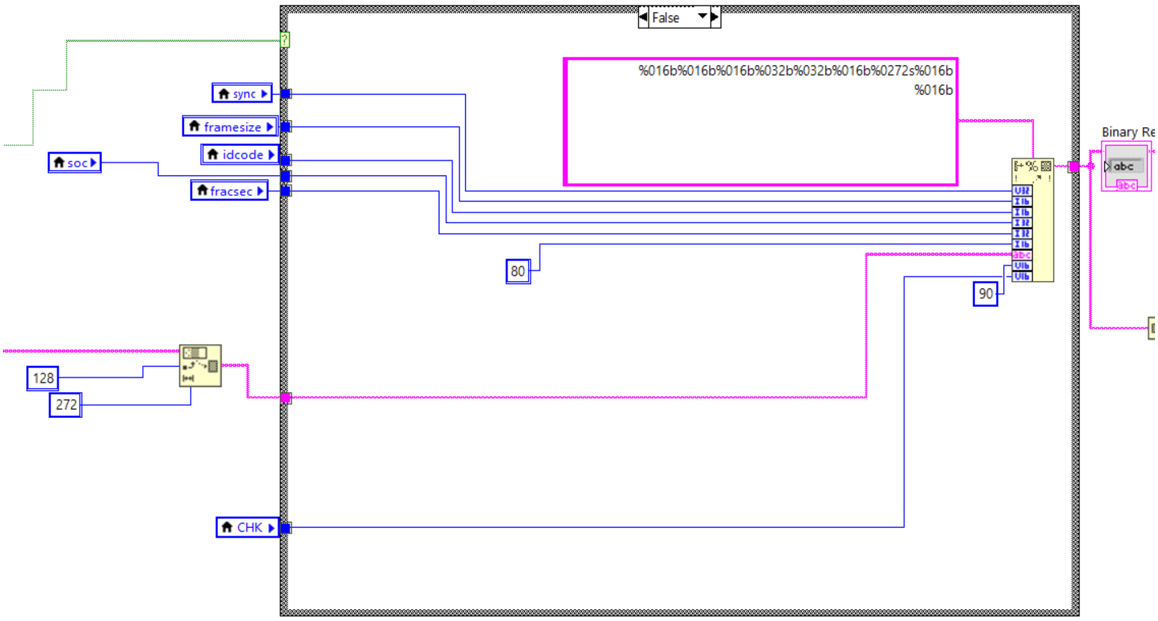


Figure III.17: False case

III.5.4.2 True case

The TRUE case indicates that the phase difference between PMU1 and PMU2 is different, indicating that the STAT of PMU2 is 80. Similar to the FALSE case, all inputs are connected to a FORMAT INTO STRING blockas shown in Figure III.18. Additionally, the decoded data of PMU2, including magnitude and new phase for all voltage and current phases, is added and PMU2 rest data too, which includes digital, analog, frequency, and ROCOF data. The output of both FORMAT INTO STRING blocks is the binary representation of the aggregated aligned data. This binary representation is then sent via UDP protocol through port number 8900 to other devices carrying an IP address of 192.168.1.100 and 192.168.1.21.

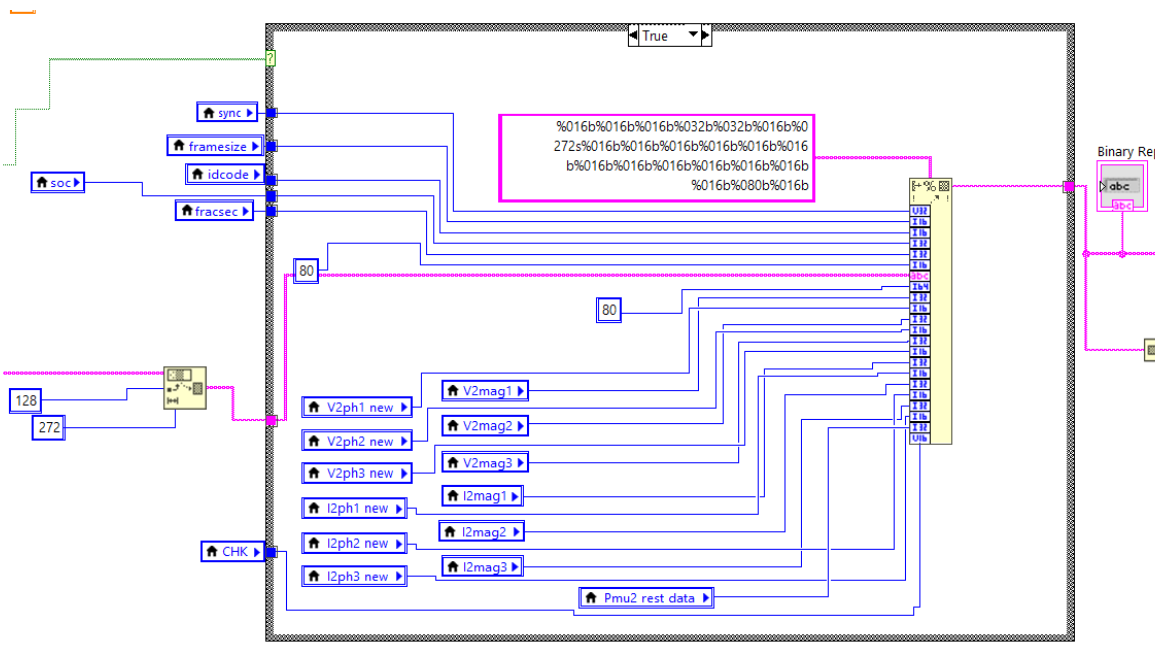


Figure III.18: True case

III.6 PDC performed Applications

III.6.1 Data Storage and Retrieval

The server/application receives a binary-encoded data stream from the PDC via the UDP protocol on port 8900 of the PC. This data is read by the application through a UDP read operation. It is crucial that the size of the transferred data is constant and at least as large as data frame to avoid Buffer size insufficient errors. Upon receiving the data, which is in binary representation, the application decodes it to extract necessary information, such as phasors for voltage and current. Since the data format is known, the application can locate various parameters within the stream. The server/application understands that the combined data frame contains data from two PMUs. To retrieve this data, the server compares the STAT values of the two PMUs, which are each 2 bytes long. The STAT value for PMU1 starts from the 112th bit, while for PMU2, it starts from the

400th bit. To extract these STAT values, a string subset block is used to isolate the binary numbers representing STAT1 and STAT2. These binary numbers are then converted into numerical values using a scan value block, enabling comparison as illustrated in Figure III.19. This comparison yields a boolean value, which is then connected to a case structure with two cases.

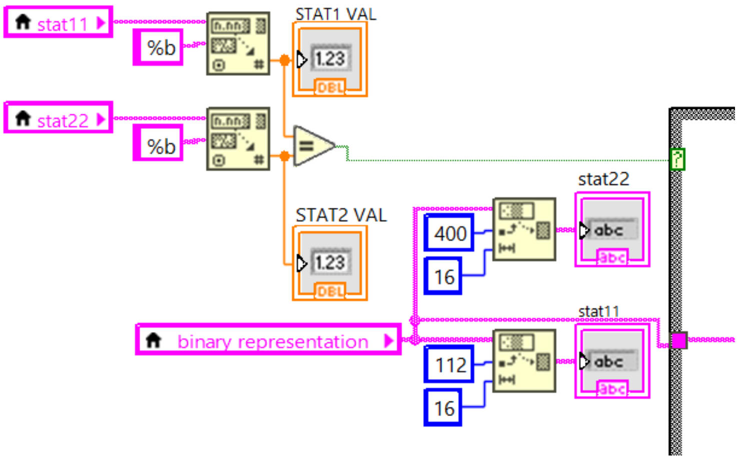


Figure III.19: STAT comparison

False case: This case (shown in Figure III.20)occurs when the boolean is false, indicating that the STAT of PMU1 and PMU2 are different. In LabVIEW, the SCAN FROM STRING block requires two main parameters: the format string and the initial scan location. The format string defines the format of the input string to ensure the output is correctly interpreted. For example, using $\%s$ for a string or $\%b$ for binary will yield different outputs; 10 will be interpreted as 10 (double) with $\%s$ and as 2 (double) with $\%b$. Multiple types can be included in the same format string. The initial scan location defaults to 0 but can be adjusted as needed. In our implementation, the FALSE case implies that the data stream contains only data from PMU1 by default (voltage, current, ROCOF, frequency, etc.). Therefore, the data of PMU2 will be the same as that of PMU1, as discussed before. To extract each data piece individually, we segmented our input string into 16-bit segments, which aligns with the logic in Table II.2.

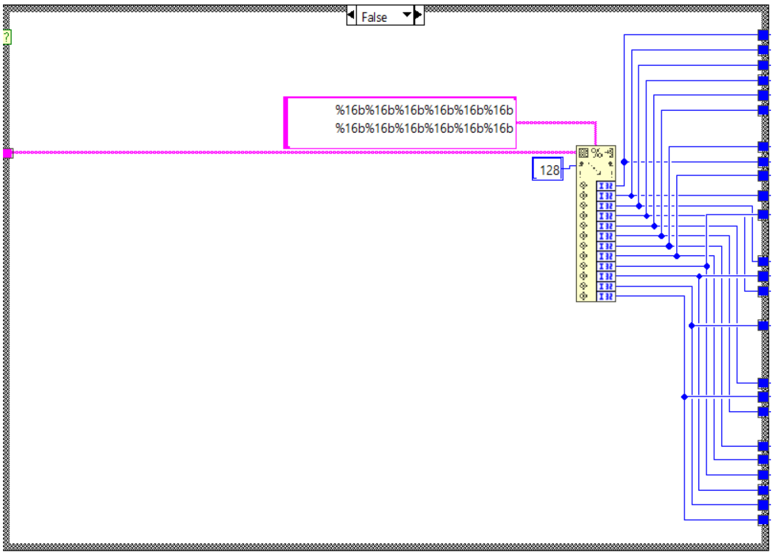


Figure III.20: Different STAT

True case: Two SCAN FROM STRING blocks are used because the STAT values are the same (80) , indicating that the data stream contains data from two PMUs. The structure of the stream data in this case is clearly depicted in Figure III.18 of the previous section. The initial scan starts from bit 128 for PMU1 and bit 416 for PMU2. Both blocks use the same format as in the first case, but they produce different outputs, each corresponding to a specific PMU(See Figure III.21) .

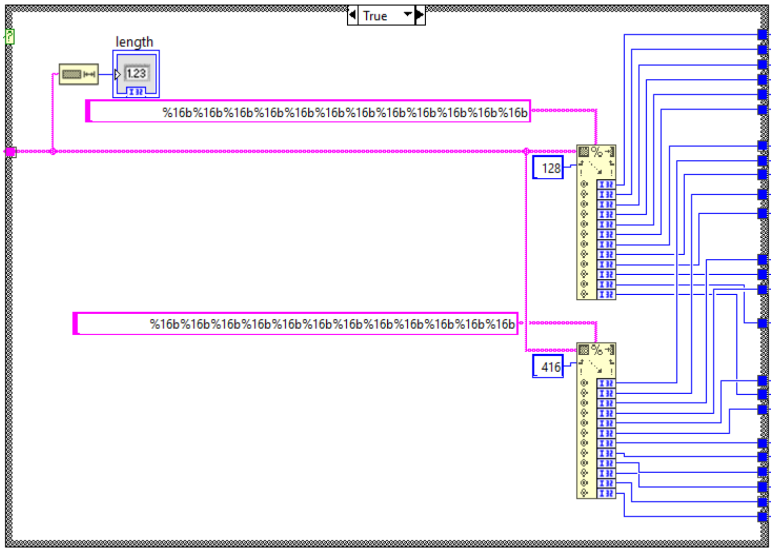


Figure III.21: Same STAT

The output of this loop is the decoded frame sent by the PDC. In our implementation, we were primarily interested in the magnitude and phase of both voltage and current of PMU1 and PMU2. Note that the phase was sent in radian form, so a radian-to-degree block was necessary for further use of this parameter.

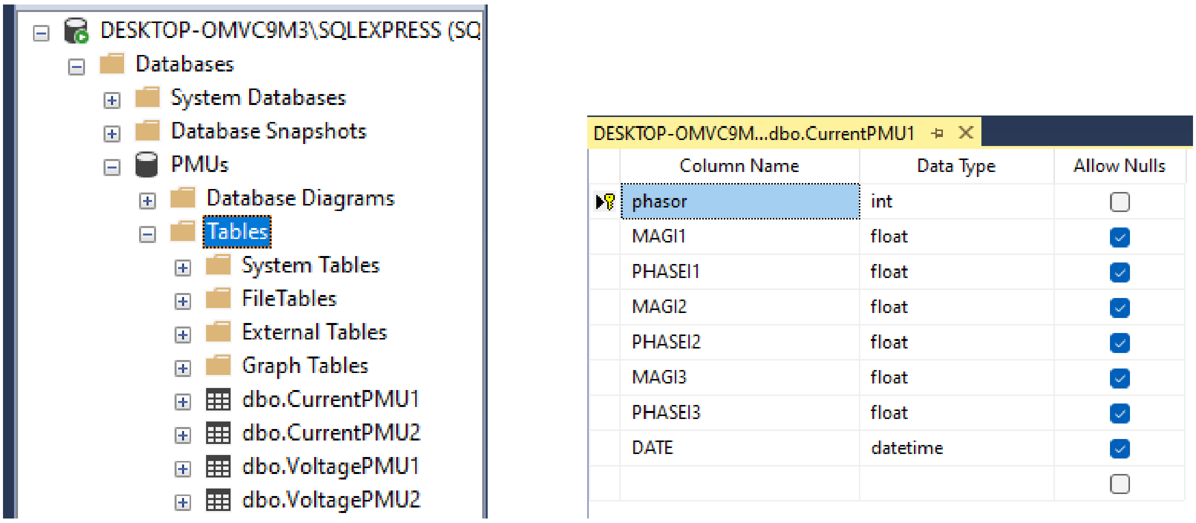
- **Storage loop:**

The structure consists of a for loop inside a while loop with 50 iterations. Local variables are created for the different decoded data parameters that are stored in an SQL database using SQL blocks from LabVIEW. There are primarily four tables:

- **Table 1:** Voltage PMU1 - This table includes the magnitude and phase for the 3-phase voltage.
- **Table 2:** Voltage PMU2 - This table includes the magnitude and phase for the 3-phase voltage.
- **Table 3:** Current PMU1 - This table includes the magnitude and phase for the 3-phase current.

- **Table 4:** Current PMU2 - This table includes the magnitude and phase for the 3-phase current.

Each table contains the following columns (Figure III.22).The index was inserted as a query in order to track the order



(a) Database tables (b) Table columns

Figure III.22: Database configuration

The process starts with an SQL open block that requires one parameter: the ODBC connection, named 'store' in this case. An SQL close block is also used. The main block is the SQL execute block, which is connected to the output and input of SQL open and close, respectively. A FORMAT INTO STRING block is used to assign the data to the parameters of the query. The query's general form is structured so that each value connected to the input of the FORMAT INTO STRING block is assigned to the table's parameters. The output of this block is then connected to SQL execute, and the data is inserted into the database. The Figure III.23 displays how the PMU1 data is stored. Similarly, the PMU2 data is stored in a block directly below this one in the for loop.

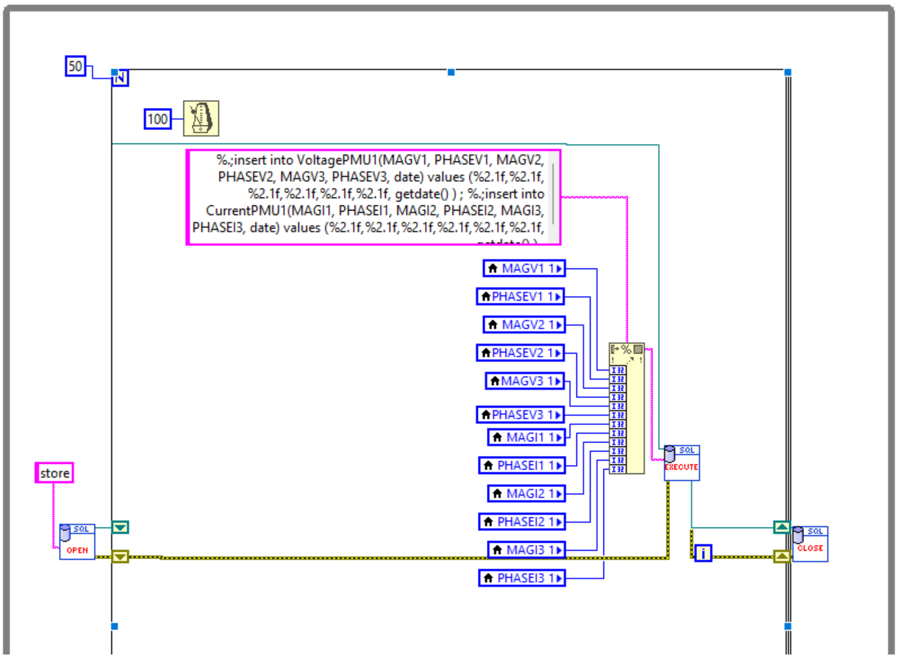


Figure III.23: Storing PMU 1 data in database Tables

- **Retrieve loop :**

Having explained the concept of event structures and the Queued Message Handler (QMH), let's now discuss how they are linked together in our implementation. When a specific case is executed within the event structure, it runs its contents, including the queued element block, which enqueues a specific input. After execution, the enqueued element is sent through the QMH process to other loops for specific usage. In our implementation, we have three cases in the event structure: 1min, 1day, and 1hour, each corresponding to a push button with a similar name. Pressing a button triggers the execution of the corresponding case. The difference between these cases lies in the element to be queued. As we have three cases, each is utilized for a specific operation, necessitating the use of three different enqueued elements.To determine when the user wants to retrieve data, we included a GET DATE/TIME IN SECONDS block, which provides a timestamp of the current time. This timestamp is then processed using the FORMAT DATE/TIME STRING block, where it is formatted according to a specified time format string to obtain the date/time string in the desired general format. Therefore, whenever the 1 min push button is pressed, the date is updated to the current date as shoxn in Figure III.24.

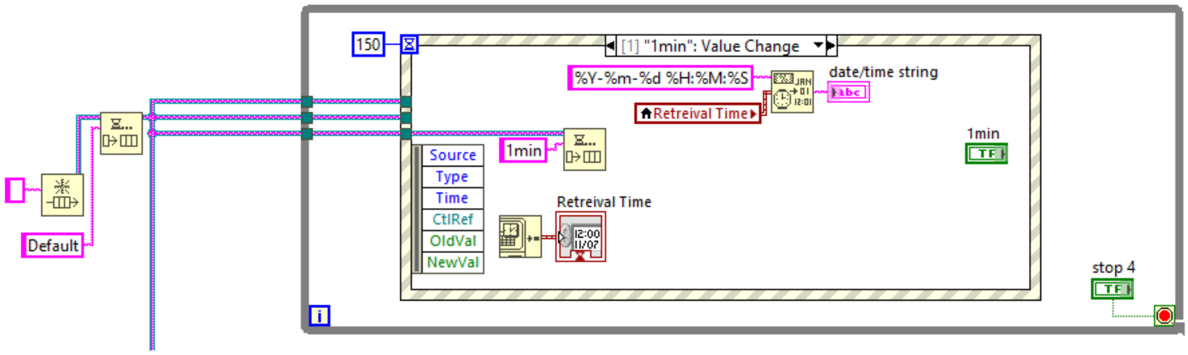


Figure III.24: Event structure case

The retrieving loop is connected to an event structure, which is essentially a case structure inside a while loop. After executing a specific case in the case structure, the enqueued element is then dequeued in this loop. It is connected to the selector of the case structure to execute the case corresponding to the user input. The cases in the loop mirror those in the event structure. Each case includes two elements: the SQL query for retrieving data and the iteration number for the display loop. The query varies based on the table being read from and the period of time for data retrieval. With four tables, there are four corresponding queries, each tailored for specific data display. The format of the queries are shown in Figure III.25.

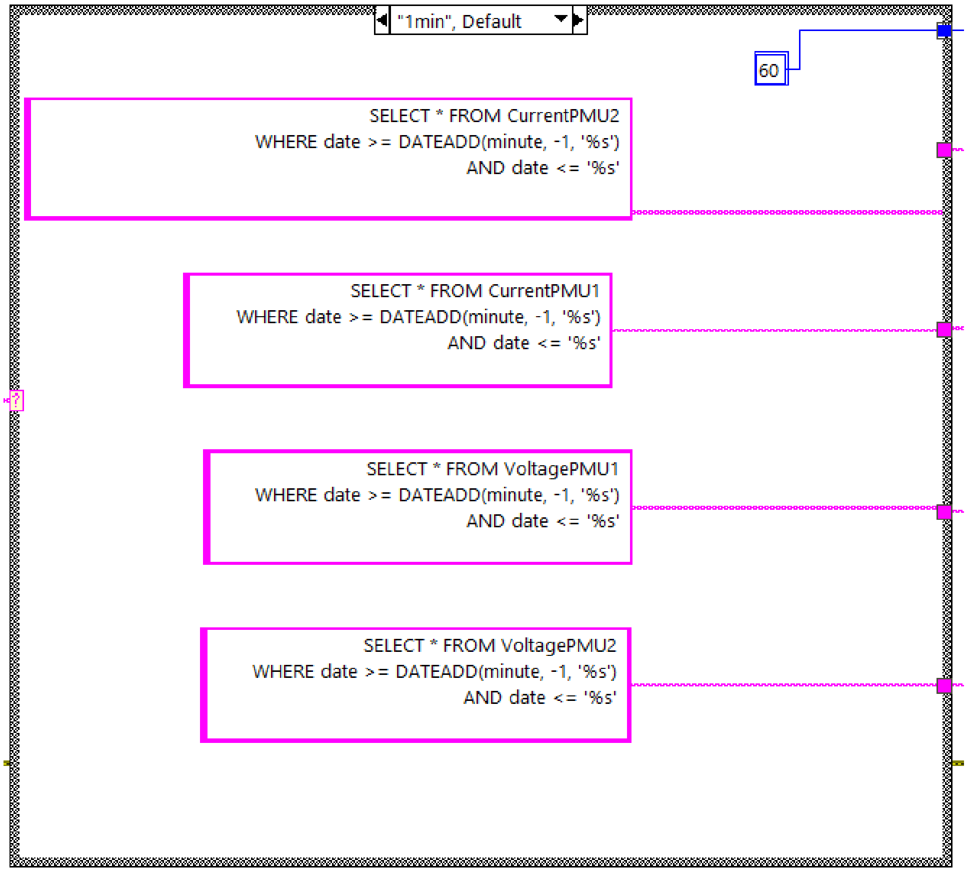


Figure III.25: Retrieve query for 1min

Where DATEADD is a function to retrieve data based on the provided argument, such as 1 minute, 1 hour, or 1 day. The loop illustrated in Figure III.26 retrieves data from a specific time within the last 1 minute, 1 hour, or 1 day, depending on when the user pressed the 1 MIN button. The symbols AND specify the time interval for data retrieval.

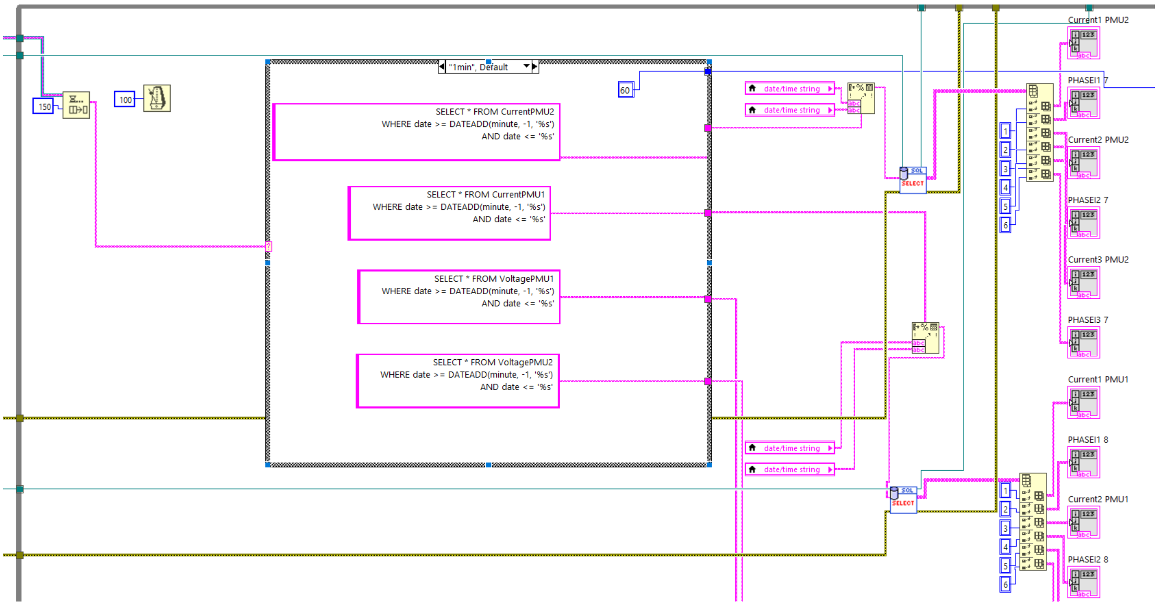


Figure III.26: Retrieve loop

Once a specific case is selected and a specific query is run, the date/time identified by the user needs to be assigned in the query. For this, a FORMAT INTO STRING block is used, with the same variable for the two inputs of the block. The resulting string will be connected to a query select. Now that data is retrieved from the database. An index array is used to extract each column from the array, facilitating the separation and easy reading of data. The size of the array depends on the selected time period: if 1 minute, then 60 elements; if 1 hour, then 3600 elements; if 1 day, then 86400 elements.

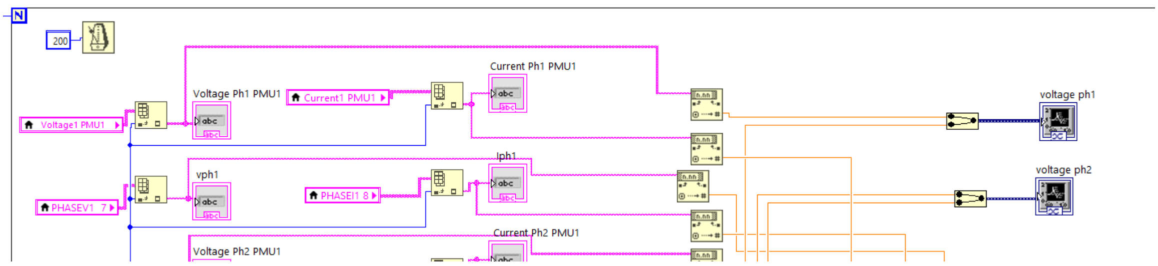


Figure III.27: Display retrieved data

The retrieved data is in the form of a 1D array, each element of this array needs to be read for display purposes(See Figure III.27). A for loop is required, with the number of iterations corresponding to the specific case and an index array block is also needed. Since the elements are in string format, a string-to-number conversion is needed to obtain numeric data. These numeric elements will be connected to a MERGE block, which will then be connected to a waveform block to visualize the data changes. There are seven waveform charts, each with two inputs: Voltage Phase 1 PMU1 and Voltage Phase 1 PMU2. This setup will be extended for current and voltage of all phases. The last chart will include 12 elements, representing the phases of current and voltage of PMU1 and PMU2.

III.6.2 Phasors Display

This section is managed on a separate server. Data received from UDP port 8901 is decoded similarly to the store and retrieve loop. After extracting elements, sine wave signals are generated using the magnitude and phase for each phasor. Voltages and currents from PMU1 and PMU2 are merged and displayed on a waveform chart. A vector graph, created using the polar to complex block, visualizes phase shifts and changes in magnitude and phase. Finally, real and reactive power measurements are calculated to assess the power system's performance as illustrated in Figure III.28.

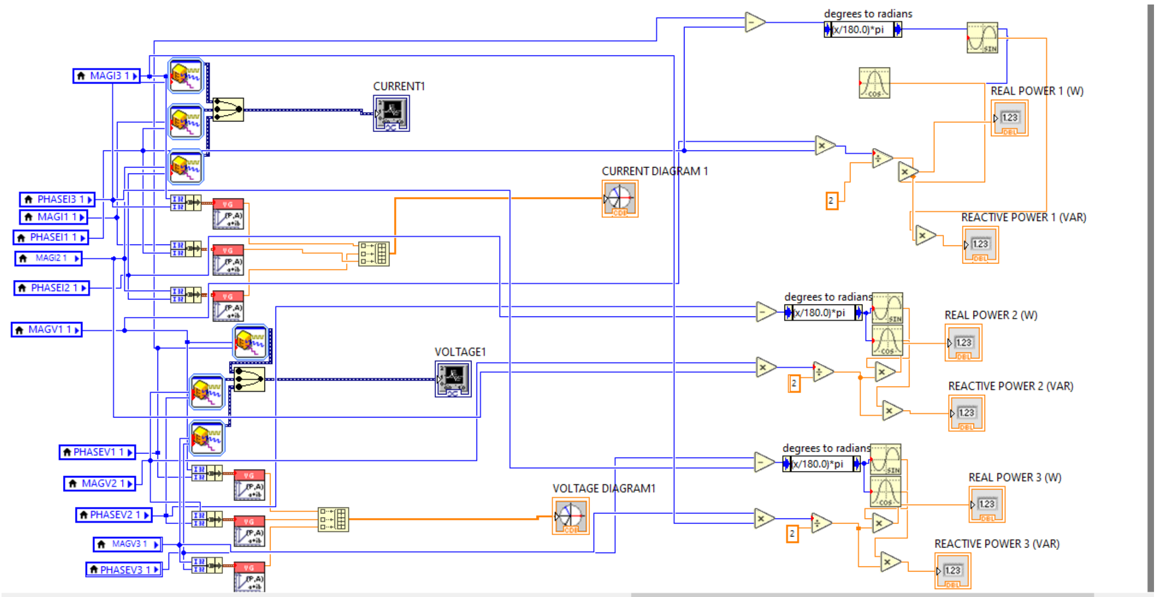


Figure III.28: Phasors Display

III.7 Conclusion

In this chapter, we investigated how LabVIEW software can be utilized to create a PDC. The chapter provided a thorough overview of PDC design, including the transmission of PMU data, data aggregation, time synchronization, and various applications such as data storage and phasor display. This examination underscores LabVIEW's efficacy in developing sophisticated systems for monitoring electrical power. The subsequent chapter will delve into the outcomes and evaluations of our project.

Chapter IV

Results and Discussion

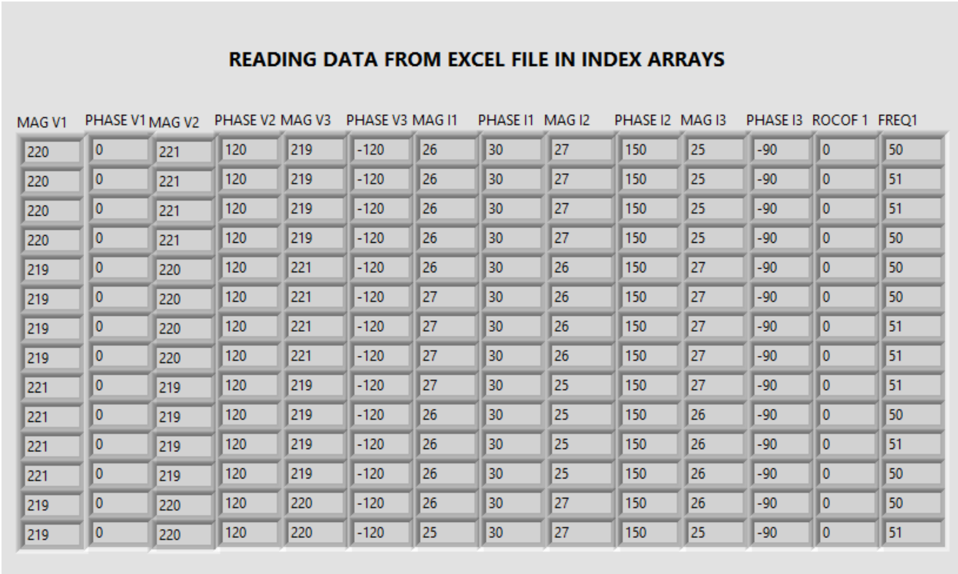
IV.1 Introduction

This chapter presents the results and discussions of the implemented PDC using LabVIEW. It provides detailed insights into the PDC's performance and capabilities, covering components such as PMU data encoding, data aggregation, phasor data visualization, and database operations, where two cases are demonstrated.

IV.2 LabVIEW Results

IV.2.1 PMU encode loop

As previously explained in Chapter III, columns are extracted from the Excel file. The resulting index columns from the spreadsheet are displayed in Figure IV.1.



READING DATA FROM EXCEL FILE IN INDEX ARRAYS

MAG V1	PHASE V1	MAG V2	PHASE V2	MAG V3	PHASE V3	MAG I1	PHASE I1	MAG I2	PHASE I2	MAG I3	PHASE I3	ROCOF 1	FREQ1
220	0	221	120	219	-120	26	30	27	150	25	-90	0	50
220	0	221	120	219	-120	26	30	27	150	25	-90	0	51
220	0	221	120	219	-120	26	30	27	150	25	-90	0	51
220	0	221	120	219	-120	26	30	27	150	25	-90	0	50
219	0	220	120	221	-120	26	30	26	150	27	-90	0	50
219	0	220	120	221	-120	27	30	26	150	27	-90	0	50
219	0	220	120	221	-120	27	30	26	150	27	-90	0	51
219	0	220	120	221	-120	27	30	26	150	27	-90	0	51
221	0	219	120	219	-120	27	30	25	150	27	-90	0	51
221	0	219	120	219	-120	26	30	25	150	26	-90	0	50
221	0	219	120	219	-120	26	30	25	150	26	-90	0	51
221	0	219	120	219	-120	26	30	25	150	26	-90	0	50
219	0	220	120	220	-120	26	30	27	150	26	-90	0	50
219	0	220	120	220	-120	25	30	27	150	25	-90	0	51

Figure IV.1: Resulting columns

The resulting binary string, after completing the conversion and formatting process, is as follows (See Figure IV.2 :

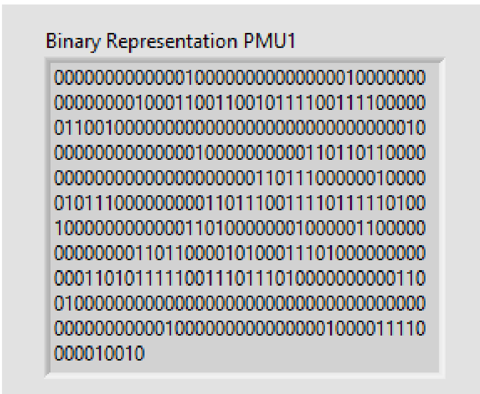


Figure IV.2: Resulting binary representation

The resulting binary string is then sent to the PDC via the UDP/IP protocol to another PC with the IP address 192.168.1.100 on port 30001. The following example, shown in Figure IV.3 , illustrates the process of correctly sending data using UDP on the same PC.

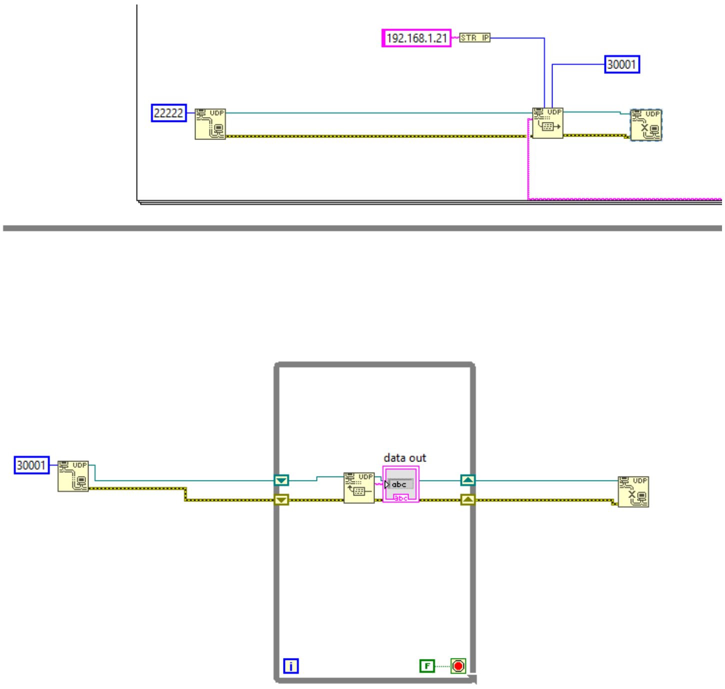


Figure IV.3: Sending data via UDP/IP protocol

The outputs of this process, which confirm the successful transmission and reception of the data, are represented in Figure IV.4. These outputs validate that the data has been accurately sent and received, maintaining the integrity and structure of the information as intended.

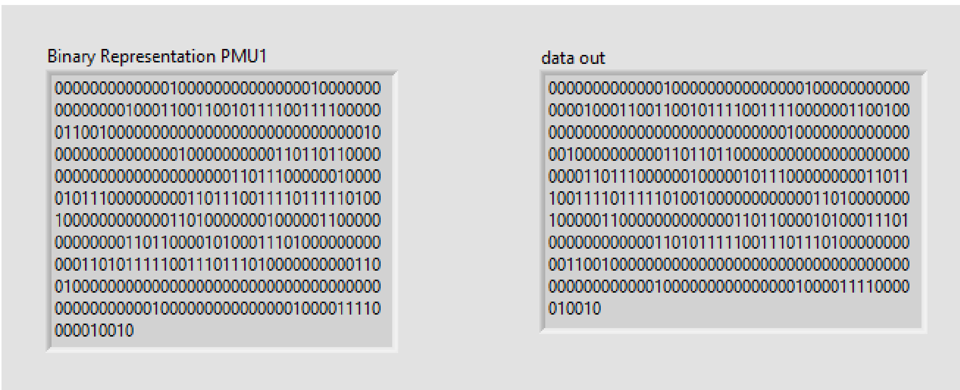


Figure IV.4: UDP/IP data out

After sending the binary string via UDP, the reception process involves decoding all phases and magnitudes to compare the phases and select the appropriate format for the aggregated data. In this part, we compared the resulting values with the sent ones to ensure accuracy and integrity of the transmitted data.

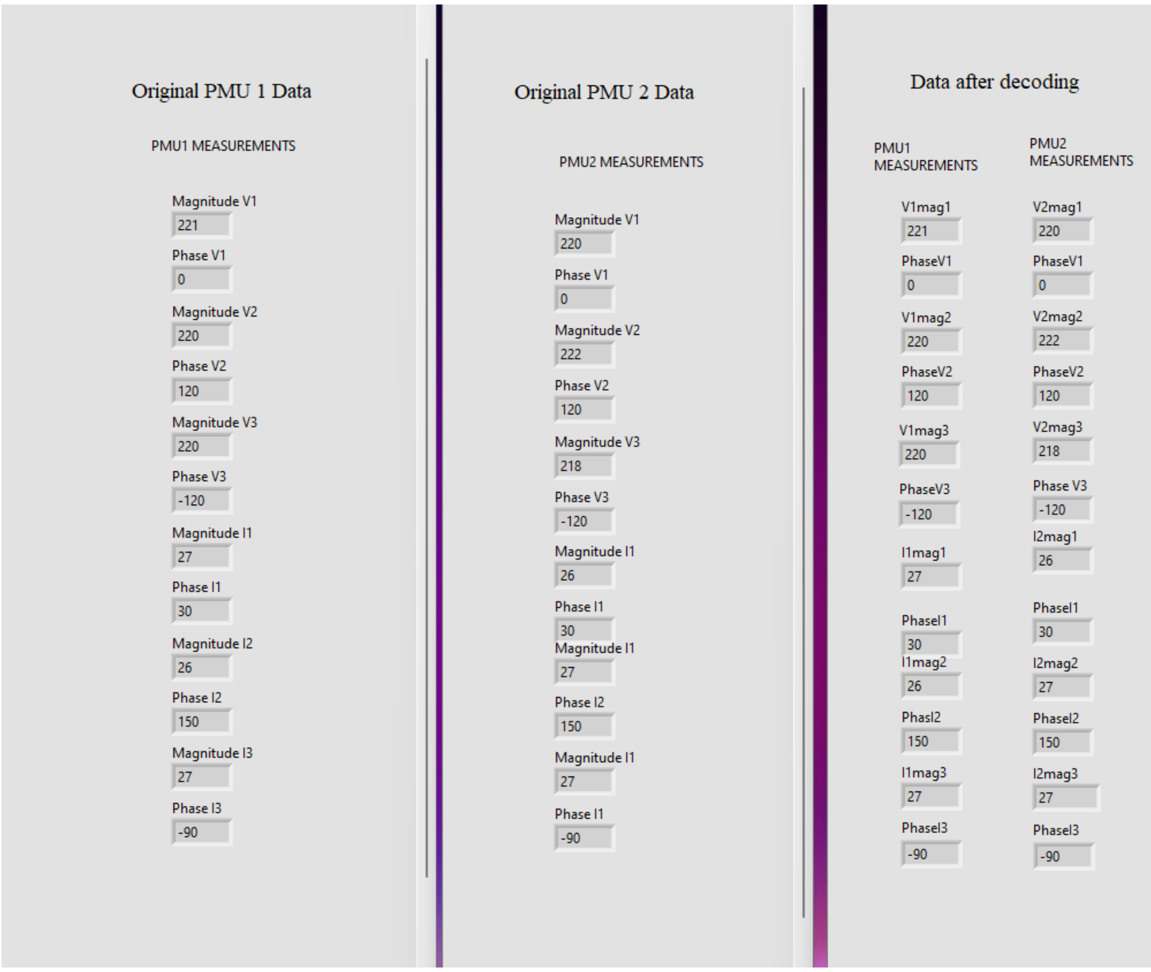


Figure IV.5: Data Comparison

IV.2.2 Data aggregation

False case : When this case is executed, the input consists primarily of data from PMU1 and common type data for all frames, as explained before . The output is the new data frame in binary representation .

True Case: When this case is executed, the input consists of data from both PMU1 and PMU2, where PMU2 provides the magnitude and new phase for the current and voltage of all phases. The output is the new data frame, which is the aggregated data presented in binary format as shown in Figure IV.6.

(a) False case (b) True case

Figure IV.6: Format Case loop

The difference between these two cases is illustrated in the Figures IV.6(a), IV.6(b).It shows that the length of the data frame changes when there is a phase difference between PMU1 and PMU2 for the current and voltage phases. This indicates that the STAT of PMU2 is 80, and the data from PMU2 is encoded into the new data frame. If no difference occurs, this indicates that the STAT of PMU2 is 90, and the data from PMU2 is the same as the data from PMU1, so only the data from PMU1 is encoded in the new data frame. This explains the change in the length of the binary representation of the entire data frame.

Without time alignment: the new phases of PMU2 are the same as the original ones because the timestamp difference is 0. This represents the ordinary case. If the timestamp difference is different from 0, the new phases will also be the same as the original ones. This is because the frequency is 50 Hz, and the modulo of $2\Pi f \triangle t$ is always 0.

With time alignment: a shift in the phases of PMU2 can be observed if the case is true. This occurs when the UNIX timestamp is fractional, which is impossible. Therefore, the only way to observe a phase shift is when the frequency is slightly more than 50 Hz. In our test, with a frequency of 50.1 Hz and a timestamp difference of 21 seconds between the SOC of PMU1 and PMU2, this led to an additional phase shift of 15 degrees. in false case the displayed phases of PMU2 will remain the same, the FigureIV.7 below illustrates this .

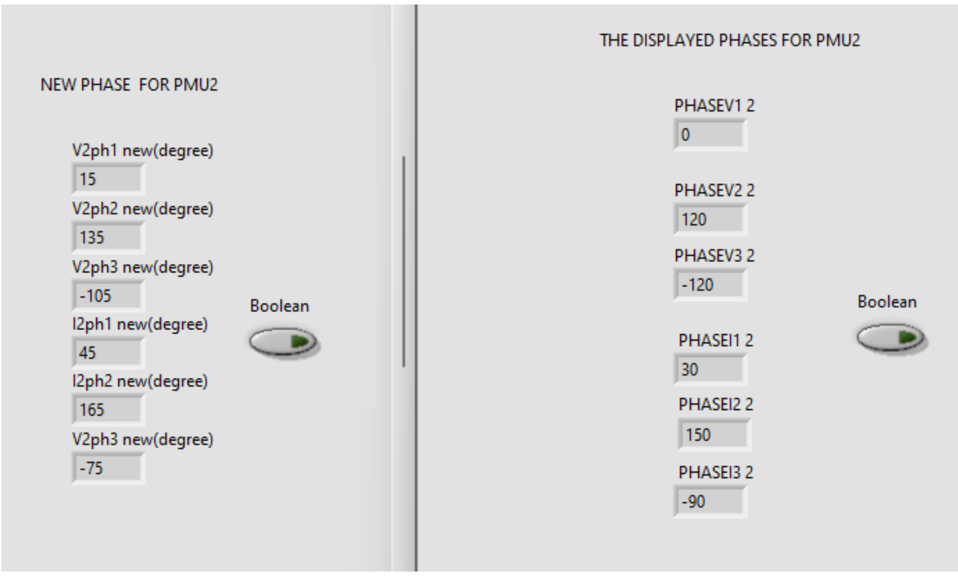


Figure IV.7: Displayed phase shift

IV.2.3 Vector plot

As the binary representation of the data stream is sent to the server, a decoding loop is needed. Here, we are going to test with a Boolean condition. This Boolean corresponds to the result of the comparison between the status of PMU1 and PMU2.

True Case: If the statuses of both PMUs are the same, the data will be different, corresponding to the true case. The result displayed on phasor plots shows:

- **Without time alignment:** the magnitude of voltage and current are different, but the phases of voltage and current for each PMU are the same (See Figures IV.8 and IV.9).

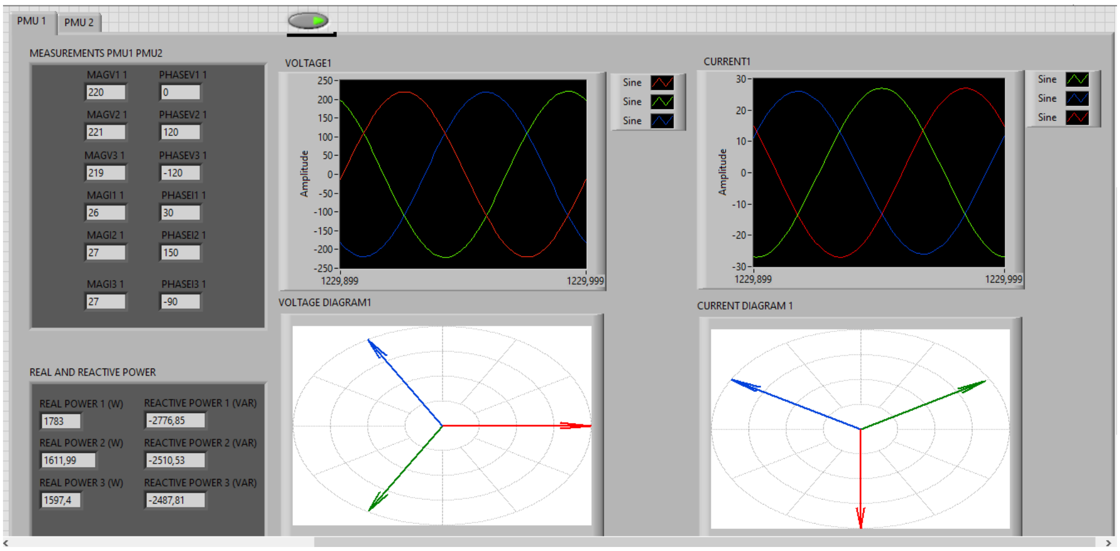


Figure IV.8: True case without time-alignment PMU1

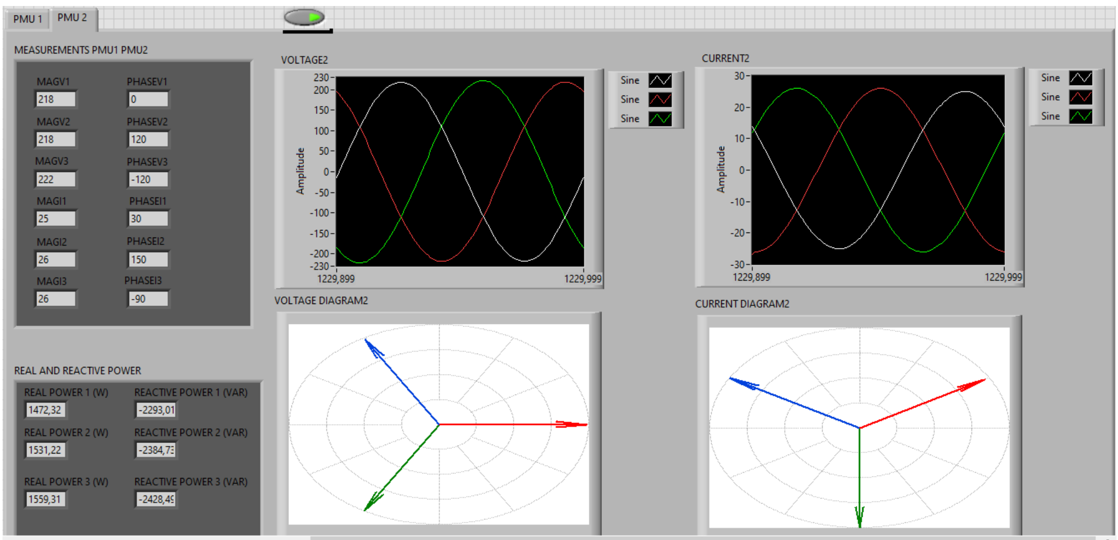


Figure IV.9: True case without time-alignment PMU2

- **With time alignment :** the phases of PMU2 are shifted by a specific amount depending on frequency and timestamp, as shown earlier. The magnitude of voltage and current is different (See Figures IV.10 and IV.11 .

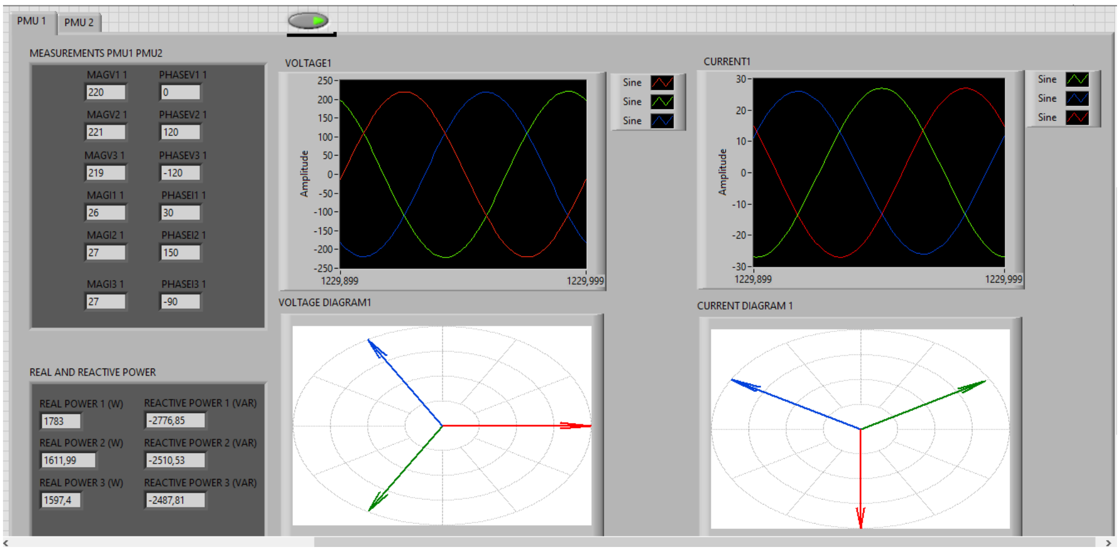


Figure IV.10: True case with time-alignment PMU1

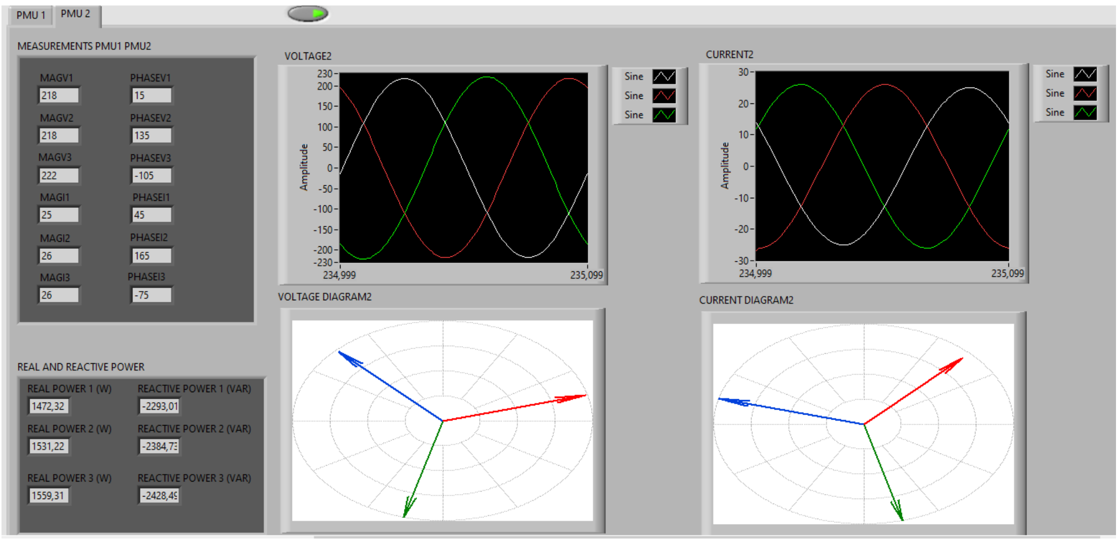


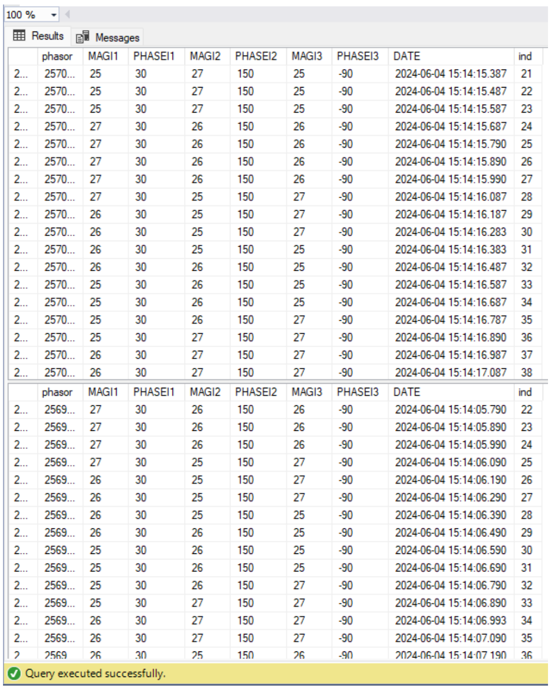
Figure IV.11: True case with time-alignment PMU2

False Case: If the status of PMU1 and PMU2 are different, this indicates that the data of PMU1 is the same as the data of PMU2. The display will be the same with or without time alignment because only the data of PMU1 is in the data stream, and the data of PMU2 is just the same as PMU1.

IV.2.4 Database

IV.2.4.1 Storing data

As noted in Chapter III, the data is stored in an MS SQL database, with separate tables for each PMU. The resulting database data is displayed in Figure IV.12.


100 %

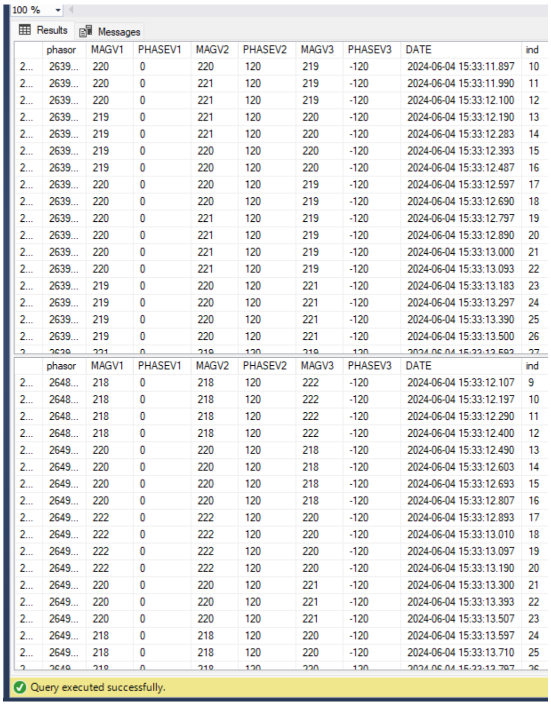
Results | Messages

	phasor	MAGI1	PHASEI1	MAGI2	PHASEI2	MAGI3	PHASEI3	DATE	ind
2...	2570...	25	30	27	150	25	-90	2024-06-04 15:14:15.387	21
2...	2570...	25	30	27	150	25	-90	2024-06-04 15:14:15.487	22
2...	2570...	25	30	27	150	25	-90	2024-06-04 15:14:15.587	23
2...	2570...	27	30	26	150	26	-90	2024-06-04 15:14:15.687	24
2...	2570...	27	30	26	150	26	-90	2024-06-04 15:14:15.790	25
2...	2570...	27	30	26	150	26	-90	2024-06-04 15:14:15.890	26
2...	2570...	27	30	26	150	26	-90	2024-06-04 15:14:15.990	27
2...	2570...	27	30	25	150	27	-90	2024-06-04 15:14:16.087	28
2...	2570...	26	30	25	150	27	-90	2024-06-04 15:14:16.187	29
2...	2570...	26	30	25	150	27	-90	2024-06-04 15:14:16.283	30
2...	2570...	26	30	25	150	25	-90	2024-06-04 15:14:16.383	31
2...	2570...	26	30	26	150	25	-90	2024-06-04 15:14:16.487	32
2...	2570...	25	30	26	150	25	-90	2024-06-04 15:14:16.587	33
2...	2570...	25	30	26	150	25	-90	2024-06-04 15:14:16.687	34
2...	2570...	25	30	26	150	27	-90	2024-06-04 15:14:16.787	35
2...	2570...	25	30	27	150	27	-90	2024-06-04 15:14:16.890	36
2...	2570...	26	30	27	150	27	-90	2024-06-04 15:14:16.987	37
2...	2570...	26	30	27	150	27	-90	2024-06-04 15:14:17.087	38

	phasor	MAGI1	PHASEI1	MAGI2	PHASEI2	MAGI3	PHASEI3	DATE	ind
2...	2569...	27	30	26	150	26	-90	2024-06-04 15:14:05.790	22
2...	2569...	27	30	26	150	26	-90	2024-06-04 15:14:05.890	23
2...	2569...	27	30	26	150	26	-90	2024-06-04 15:14:05.990	24
2...	2569...	27	30	25	150	27	-90	2024-06-04 15:14:06.090	25
2...	2569...	26	30	25	150	27	-90	2024-06-04 15:14:06.190	26
2...	2569...	26	30	25	150	27	-90	2024-06-04 15:14:06.290	27
2...	2569...	26	30	25	150	25	-90	2024-06-04 15:14:06.390	28
2...	2569...	26	30	26	150	25	-90	2024-06-04 15:14:06.490	29
2...	2569...	25	30	26	150	25	-90	2024-06-04 15:14:06.590	30
2...	2569...	25	30	26	150	25	-90	2024-06-04 15:14:06.690	31
2...	2569...	25	30	26	150	27	-90	2024-06-04 15:14:06.790	32
2...	2569...	25	30	27	150	27	-90	2024-06-04 15:14:06.890	33
2...	2569...	26	30	27	150	27	-90	2024-06-04 15:14:06.993	34
2...	2569...	26	30	27	150	27	-90	2024-06-04 15:14:07.090	35
2	2569	26	30	25	150	26	-90	2024-06-04 15:14:07.190	36

Query executed successfully.

(a) Stored current data both PMUs


100 %

Results | Messages

	phasor	MAGV1	PHASEV1	MAGV2	PHASEV2	MAGV3	PHASEV3	DATE	ind
2...	2639...	220	0	220	120	219	-120	2024-06-04 15:33:11.897	10
2...	2639...	220	0	221	120	219	-120	2024-06-04 15:33:11.990	11
2...	2639...	220	0	221	120	219	-120	2024-06-04 15:33:12.100	12
2...	2639...	219	0	221	120	220	-120	2024-06-04 15:33:12.190	13
2...	2639...	219	0	221	120	220	-120	2024-06-04 15:33:12.283	14
2...	2639...	219	0	220	120	220	-120	2024-06-04 15:33:12.393	15
2...	2639...	219	0	220	120	220	-120	2024-06-04 15:33:12.487	16
2...	2639...	220	0	220	120	219	-120	2024-06-04 15:33:12.597	17
2...	2639...	220	0	220	120	219	-120	2024-06-04 15:33:12.690	18
2...	2639...	220	0	221	120	219	-120	2024-06-04 15:33:12.797	19
2...	2639...	220	0	221	120	219	-120	2024-06-04 15:33:12.890	20
2...	2639...	220	0	221	120	219	-120	2024-06-04 15:33:13.000	21
2...	2639...	220	0	221	120	219	-120	2024-06-04 15:33:13.093	22
2...	2639...	219	0	220	120	221	-120	2024-06-04 15:33:13.183	23
2...	2639...	219	0	220	120	221	-120	2024-06-04 15:33:13.297	24
2...	2639...	219	0	220	120	221	-120	2024-06-04 15:33:13.390	25
2...	2639...	219	0	220	120	221	-120	2024-06-04 15:33:13.500	26
2	2639	221	0	219	120	219	-120	2024-06-04 15:33:13.593	27

	phasor	MAGV1	PHASEV1	MAGV2	PHASEV2	MAGV3	PHASEV3	DATE	ind
2...	2648...	218	0	218	120	222	-120	2024-06-04 15:33:12.107	9
2...	2648...	218	0	218	120	222	-120	2024-06-04 15:33:12.197	10
2...	2648...	218	0	218	120	222	-120	2024-06-04 15:33:12.290	11
2...	2648...	218	0	218	120	222	-120	2024-06-04 15:33:12.400	12
2...	2649...	220	0	220	120	218	-120	2024-06-04 15:33:12.490	13
2...	2649...	220	0	220	120	218	-120	2024-06-04 15:33:12.603	14
2...	2649...	220	0	220	120	218	-120	2024-06-04 15:33:12.693	15
2...	2649...	220	0	220	120	218	-120	2024-06-04 15:33:12.807	16
2...	2649...	222	0	222	120	220	-120	2024-06-04 15:33:12.893	17
2...	2649...	222	0	222	120	220	-120	2024-06-04 15:33:13.010	18
2...	2649...	222	0	222	120	220	-120	2024-06-04 15:33:13.097	19
2...	2649...	222	0	222	120	220	-120	2024-06-04 15:33:13.190	20
2...	2649...	220	0	220	120	221	-120	2024-06-04 15:33:13.300	21
2...	2649...	220	0	220	120	221	-120	2024-06-04 15:33:13.393	22
2...	2649...	220	0	220	120	221	-120	2024-06-04 15:33:13.507	23
2...	2649...	218	0	218	120	220	-120	2024-06-04 15:33:13.597	24
2...	2649...	218	0	218	120	220	-120	2024-06-04 15:33:13.710	25
2	2649	218	0	218	120	220	-120	2024-06-04 15:33:13.797	26

Query executed successfully.

(b) Stored voltage data both PMUs

Figure IV.12: Stored data in Database

IV.2.4.2 Retrieving data

The SELECT block in the SQL toolkit produces an output array based on the specified query. The example below illustrates how data can be retrieved within a given time interval as shown in Figures IV.13 and IV.14.

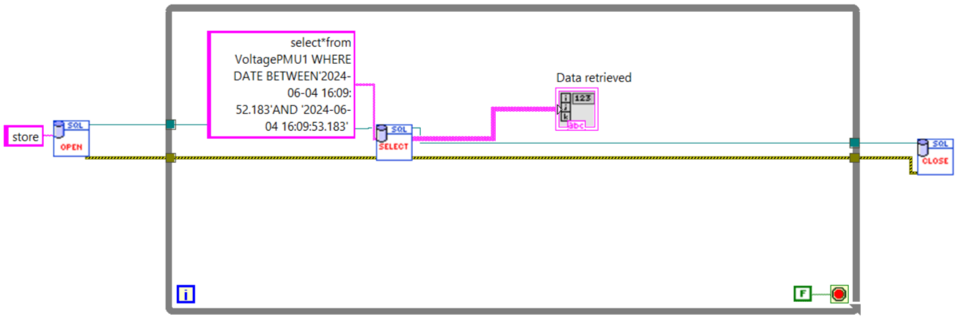
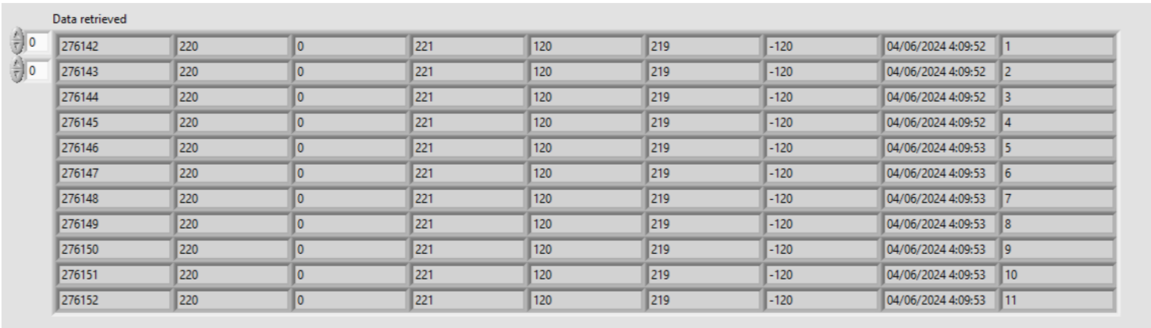


Figure IV.13: Example for retrieving data



Data retrieved

276142	220	0	221	120	219	-120	04/06/2024 4:09:52	1
276143	220	0	221	120	219	-120	04/06/2024 4:09:52	2
276144	220	0	221	120	219	-120	04/06/2024 4:09:52	3
276145	220	0	221	120	219	-120	04/06/2024 4:09:52	4
276146	220	0	221	120	219	-120	04/06/2024 4:09:53	5
276147	220	0	221	120	219	-120	04/06/2024 4:09:53	6
276148	220	0	221	120	219	-120	04/06/2024 4:09:53	7
276149	220	0	221	120	219	-120	04/06/2024 4:09:53	8
276150	220	0	221	120	219	-120	04/06/2024 4:09:53	9
276151	220	0	221	120	219	-120	04/06/2024 4:09:53	10
276152	220	0	221	120	219	-120	04/06/2024 4:09:53	11

Figure IV.14: Retrieved data

In our project, we retrieve data for different time intervals: 1 minute ago, 1 hour ago, and 1 day ago. The results, which include charts and index arrays, are presented in the following figures. These figures visualize the data points and provide detailed breakdowns of the specific data entries and their timestamps, helping to monitor real-time changes, short-term trends, and daily activities effectively.

- **Without Time alignement:**

False case:

Figure IV.15: Retrieved 1min data with no time alignment chart(False case)

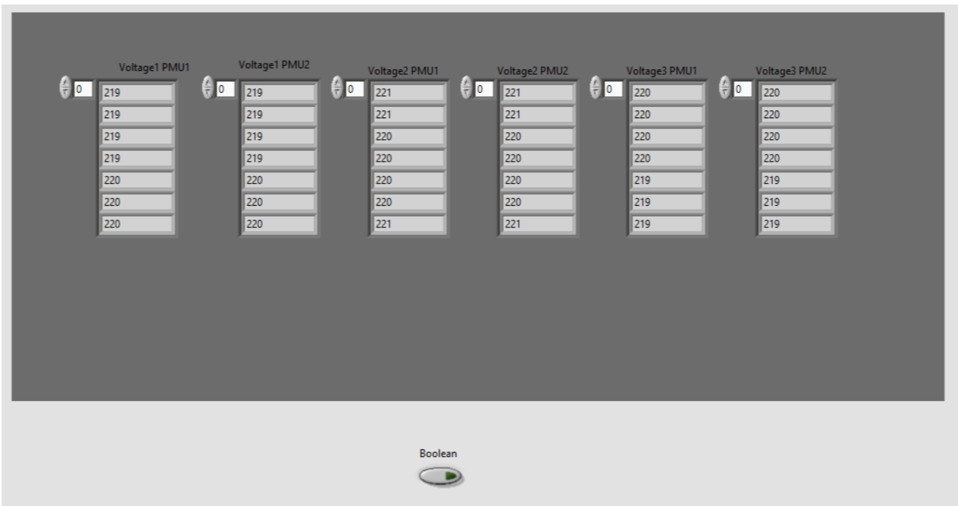


Figure IV.16: Retrieved 1 min data with no time alignment arrays(False case)

True case :

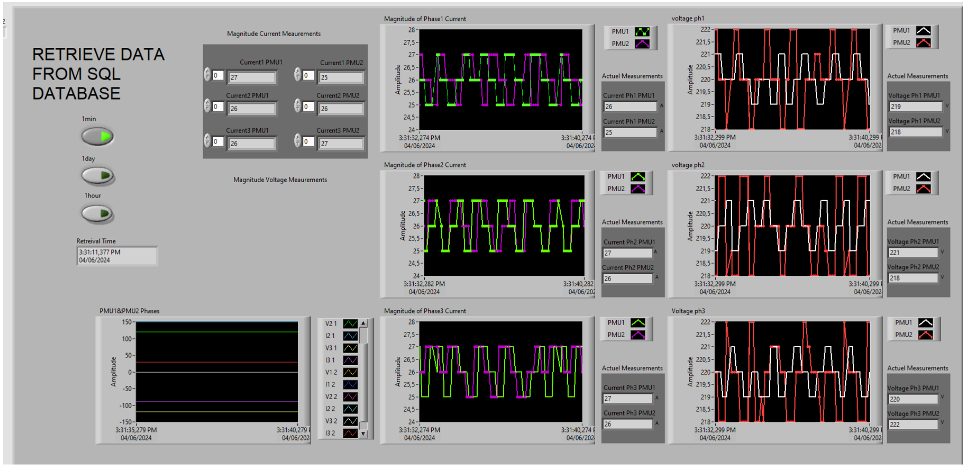


Figure IV.17: Retrieved 1min data with no time alignment chart(True case)

Figure IV.18: Retrieved 1min data with no time alignment array(False case)

- **With time alignement:** In the false case, the data from PMU2 remains unchanged from its previous state(see Figures IV.15,IV.16), indicating no variation or shift in the measurements.However, in the true case, a noticeable shift in phase is observed in the PMU2 data as depicted in Figures IV.19 and IV.20.

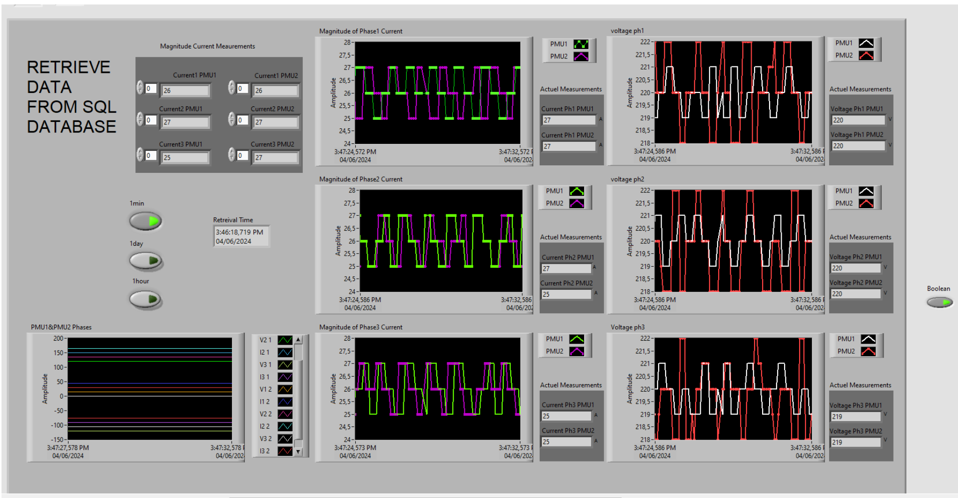


Figure IV.19: Retrieved 1min data chart with time alignment

Figure IV.20: Retrieved 1min data array with time alignment

IV.3 Results

- PMUs successfully constructed data frames from the Excel file without data loss or corruption
- Data frames from PMU1 and PMU2 were transmitted to the PDC with no transmission errors
- The PDC effectively aligned the time-stamped data from both PMUs.
- The PDC accurately aggregated data from PMU1 and PMU2
- Data frames were successfully sent to Server 1 and Server 2 without data loss.
- Data frames were correctly decoded into decimal format, matching the original Excel data
- Waveform charts displayed the decoded data accurately, matching the Excel data.
- Server 2 stored the decoded data accurately and retrieval was consistent with database timestamps.

IV.4 Interpretation of the results

Transmission of Data Frames to PDC: The error-free transmission of data frames from PMU1 and PMU2 to the PDC indicates that the system's communication protocols are robust. This reliability is crucial for maintaining data integrity during transmission, especially in real-time monitoring scenarios where data loss or corruption could lead to incorrect analyses and conclusions.

Time Alignment by PDC: The effective time alignment of data from both PMUs by the PDC is critical for synchronizing data streams from multiple sources. This ensures that the data can be accurately combined and compared, allowing for precise temporal analysis and correlation. Accurate time alignment is particularly important in applications where timing discrepancies can significantly impact the results, such as in power grid monitoring.

Data Aggregation by PDC: The accurate aggregation of data from PMU1 and PMU2 by the PDC demonstrates that the system can effectively combine multiple data streams into a cohesive data frame. This ability is essential for comprehensive data analysis, as it allows for the integration of information from different sources to provide a complete picture of the monitored environment.

Transmission of Aggregated Data to Servers: The successful transmission of aggregated data frames to Server 1 and Server 2 without any data loss underscores the reliability of the system's data transfer mechanisms. This ensures that data can be securely and accurately stored in different locations, providing redundancy and enhancing the robustness of the data management process.

Decoding of Data Frames: The correct decoding of data frames from binary to decimal format, matching the original Excel data, confirms that the system's encoding and decoding processes preserve data accuracy. This is crucial for ensuring that the data remains usable and meaningful after transmission and storage, allowing for accurate analysis and decision-making.

Visualization and Comparison of Data: The accurate display of decoded data in waveform charts, matching the original Excel data, provides visual confirmation of the data's integrity throughout the processing pipeline. This consistency is important for verifying that the data has not been altered or corrupted during any stage of the process, reinforcing the reliability of the system.

Storage and Retrieval of Data on Server 2: The accurate storage and retrieval of decoded data on Server 2, consistent with database timestamps, indicates that the system's data storage mechanisms are reliable. This ensures that data can be securely stored and accurately retrieved for future analysis, providing confidence in the long-term viability of the data management system.

IV.5 Conclusion

In this chapter, we discussed the outcomes of implementing the PDC with LabVIEW, examining how key elements like the PMU encode loop, data aggregation, vector plot visualization, and database operations performed. These results shed light on both the advantages and drawbacks of the PDC design, offering a thorough grasp of its performance. Moreover, this chapter paves the way for future enhancements geared towards improving the system's effectiveness and dependability.

Conclusion

In conclusion, this research successfully demonstrated the development and implementation of a Phasor Data Concentrator (PDC) using LabVIEW. The PDC system was designed to aggregate, align, and process time-stamped data from multiple Phasor Measurement Units (PMUs), ensuring accurate and reliable data transmission to centralized servers. The system's performance was evaluated through a series of comprehensive tests that confirmed its ability to maintain data integrity and reliability throughout the entire process. Key findings include error-free data transmission, effective time alignment, accurate data aggregation, reliable data transmission to servers, correct data decoding, accurate data visualization, and consistent data storage and retrieval.These results underscore the robustness of the communication protocols and data handling mechanisms employed in the PDC system. The successful implementation and testing of the PDC demonstrate its capability to support real-time monitoring and analysis in a Smart Grid environment. The accurate visualization of phasor data in waveform charts provided a clear and consistent representation of the grid's status, reinforcing the system's utility in practical applications.The findings of this research have significant implications for the future of power grid management. The developed PDC system enhances the ability to monitor and control the grid in real time, improving fault detection and grid stability. This contributes to the overall efficiency and resilience of the power infrastructure.Future work could focus on enhancing the system's scalability to integrate advanced analytics and other functions to provide deeper insights into grid performance. Additionally, implementing other PDC functionalities, extend the PDC to recieve data from more than two PMUs, use historian systems to store data and machine learning to improve the system's predictive capabilities, enabling proactive management of potential issues. The integration of such improvements will pave the way for a more resilient and efficient power infrastructure, capable of meeting the evolving demands of modern energy systems. This research lays a strong foundation for continued innovation in the field of power system monitoring and control.

References

[1] "MS Windows NT kernel description." http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm. Accessed: 2024-05-12.

[2] O. M. Butt, M. Zulqarnain, and T. M. Butt, "Recent advancement in smart grid technology: Future prospects in the electrical power network," *Ain Shams Engineering Journal*, vol. 12, no. 1, pp. 687–695, 2021.

[3] A. Ter-Gazarian, "Energy storage for power systems," *IEE Energy Series 6*.

[4] A. H. Al-Badi, R. Ahshan, N. Hosseinzadeh, R. Ghorbani, and E. Hossain, "Survey of smart grid concepts and technological demonstrations worldwide emphasizing on the oman perspective," *Applied System Innovation*, vol. 3, no. 1, 2020.

[5] "Text of the "energy independence and security act of 2007 (eisa)"," 2007.

[6] "Smart grid: Architecture, developments and use cases." https://stellarix.com/article/smart-grid/#wpcf7-f955-o6. Accessed: 2024-05-04.

[7] C. Greer and Wollman, "Nist framework and roadmap for smart grid interoperability standards, release 3.0," 2014.

[8] R. Gore and M. Kande, "Analysis of wide area monitoring system architectures," in *2015 IEEE International Conference on Industrial Technology (ICIT)*, pp. 1269–1274, 2015.

[9] "Ieee standard for synchrophasor measurements for power systems," *IEEE Std C37.118.1-2011 (Revision of IEEE Std C37.118-2005)*, pp. 1–61, 2011.

[10] "Ieee standard for synchrophasor data transfer for power systems," *IEEE Std C37.118.2-2011 (Revision of IEEE Std C37.118-2005)*, pp. 1–53, 2011.

[11] R. Gore and M. Kande, "Platform analysis in embedded systems," in *2nd International Conference on Emerging Technology Trends in Electronics, Communication and Networking (ET2ECN)*, 2014.

[12] M. G. Adamiak, M. Kanabar, M. D. Zadeh, and J. Rodrigues, "Design and implementation of a synchrophasor data concentrator," in *IEEE PES Conference on Innovative Smart Grid Technologies - Middle East (ISGT Middle East)*, IEEE, 2011.

[13] M. Kanabar, M. G. Adamiak, and J. Rodrigues, "Optimizing wide area measurement system architectures with advancements in phasor data concentrators (pdcs)," in *IEEE Power and Energy Society General Meeting (PES)*, IEEE, 2013.

[14] R. Gore and M. Kande, "Analysis of wide area monitoring system architectures," in *2015 IEEE International Conference on Industrial Technology (ICIT)*, pp. 1269–1274, 2015.

[15] S. User, "Wide area monitoring system (wams)." Online. Retrieved from `https://www.slideshare.net/slideshow/wide-area-monitoring-systemswams/72711296`.

[16] Typeset, "What are the advantages of using wide area measurement systems (wams)?." Online. Retrieved from `https://typeset.io/questions/what-are-the-advantages-of-using-wide-area-measurement-2em32bda5p`.

[17] A. Monti, C. Muscas, and F. Ponci, eds., *Phasor Measurement Units and Wide Area Monitoring Systems: From the Sensors to the System.*

[18] S. TM, "Phasor measurement unit (pmu) protection solutions." https://www.scopetnm.com/protection-solutions/phasor-measurement-unit/11-protection, 2024. Accessed: 2024-06-11.

[19] Z. Dong, P. Zhang, and et al., *Emerging Techniques in Power System Analysis*.

[20] A. N. Bellabaci and A. Chaoui, "Final year project report presented in partial fulfilment of the requirements for the degree of master in electronics." Low-Cost M-Class Phasor Measurement Unit with Real-Time Monitoring System.

[21] S. Buttayak, A. Wornpuen, N. Promparn, N. Charbkaew, and T. Bunyagul, "Design of phasor data concentrator for phasor monitoring system," in *2012 IEEE Conference on Sustainable Utilization and Development in Engineering and Technology (STUDENT)*, pp. 102–107, 2012.

[22] "Ieee standard for phasor data concentrators for power systems," *IEEE Std C37.247-2019*, pp. 1–44, 2019.

[23] S. Singh, S. Chakrabarti, D. Shukla, and V. Terzija, "Development and implementation of a matlab-based phasor data concentrator for synchrophasor applications," *International Journal of Electrical Power Energy Systems*, vol. 155, p. 109637, 01 2024.

[24] IEEE, "Ieee standard guide of pdc requirements for power systems," 2013.

[25] "Ieee standard for synchrophasor data transfer for power systems," *IEEE Std C37.118.2-2011 (Revision of IEEE Std C37.118-2005)*, pp. 1–53, 2011.

[26] S. Singh and et al., "International journal of electrical power and energy systems," *International Journal of Electrical Power and Energy Systems*, vol. 155, p. 109637, 2024.

[27] H. Gharavi and B. Hu, "Scalable synchrophasors communication network design and implementation for real-time distributed generation grid,"

[28] J. N. O. Ruela, "Introduction to communication networks," 2012. Retrieved from https://web.fe.up.pt/~jruela/ComNet/1_Introduction_CN_1112.pdf.

[29] Sparkbox, "What are the 7 layers of the osi model?."

[30] C. Panek, *Networking Fundamentals*. 111 River Street: John Wiley & Sons, Inc.

[31] D-Link, "D-link dgs-1210-28 28-port gigabit smart managed switch."

[32] K. Academy, "User datagram protocol (udp)." https://www.khanacademy.org/computing/computers-and-internet/xcae6f4a7ff015e7d:the-internet/xcae6f4a7ff015e7d:transporting-packets/a/user-datagram-protocol-udp, 2024. Accessed: 2024-06-11.

[33] Y. Shen, "Introduction to databases," 2022. Presentation slides.

[34] Hasura, "What is mssql server?." Online. Retrieved from https://hasura.io/learn/database/microsoft-sql-server/what-is-mssql/.

[35] Microsoft, "Microsoft open database connectivity (odbc)." Online. Retrieved from https://learn.microsoft.com/en-us/sql/odbc/microsoft-open-database-connectivity-odbc?view=sql-server-ver16.

[36] J. Travis and J. Kring, *LabVIEW for Everyone: Graphical Programming Made Easy and Fun.* Prentice Hall, third ed., 2006.

Appendix A

Table IV.1: Word definitions unique to data frame

Field	Size (bytes)	Comments
SYNC	2	First byte: AA hex Second byte: 01 hex (frame is version 1, IEEE Std C37.118-2005)
STAT	2	Bit mapped flags: Bit 15-14: Data error: 00 = good measurement data, no errors 01 = PMU error. No information available (do not use values) 10 = PMU in test mode (do not use values) or absent data may have been inserted (do not use values) 11 = PMU error (do not use values) Bit 13: PMU sync, 0 when in sync with a UTC traceable time source Bit 12: Data sorting, 0 by time stamp, 1 by arrival Bit 11: Triggered event, 0 when not triggered Bit 10: Configuration change, set to 1 for 1 min to advise configuration will change, and clear to 0 when change effected. Bit 09: Data modified, 1 if data modified by post processing, 0 otherwise Bit 08: PMU Time Quality. Bits 05-04: Unlocked time: 00 = sync locked or unlocked < 10 s (best quality) 01 = 10 s < unlocked < 100 s 10 = 100 s < unlocked <1000 s 11 = unlocked time > 1000 s Bits 03-00: Trigger reason: 1111-0000: Available for user definition 0111: Digital 0110:Reserved 0101: df/dt High 0100:Frequency high or low 0011:Phase angle diff 0010:Magnitude high 0001: Magnitude low 0000:Manual
PHASORS	4 / 8	16-bit integers, indicated by the FORMAT field in configuration 1, 2, and 3 frames **Rectangular format:** Real and imaginary, real value first 16-bit signed integers, range -32 767 to +32 767 **Polar format:** Magnitude and angle, magnitude first 16-bit unsigned integer range 0 to 65535 Angle: 16-bit signed integer, in radians * 10^4, range -31 416 to +31 416 32-bit values: Indicated by FORMAT field in configuration **Floating-point format:** Real and imaginary, in engineering units, real value first 32-bit values: indicated by FORMAT field in configuration 1, 2, and 3 frames
FREQ	2 / 4	Frequency deviation from nominal, in mHz 16-bit integer, range -32 767 to +32 767 Hz 32-bit floating point Indicated by FORMAT field in configuration 1, 2, and 3 frames
DFREQ	2 / 4	ROCOF, in Hz per second 16-bit integer, range -32 767 to +32 767 Hz per second 32-bit floating point Indicated by FORMAT field in configuration 1, 2, and 3 frames
ANALOG	2 / 4	Data, in integer or floating point 16-bit integer, 16-bit signed integer range 32-bit values indicated by FORMAT field Can be integer or floating point
DIGITAL	2	Digital status word.It could be bit mapped status or flag. Values and ranges defined by user .