University of Applied Sciences

TYPE OF REPORT – BACHELOR'S DEGREE PROGRAMME
TECHNOLOGY, COMMUNICATION AND TRANSPORT

TOWARDS SUSTAINABLE IOT: CURRENT AND VOLTAGE DATA ANALYSIS IN NB-IOT SYSTEMS

AUTHOR/S Dmitrii Levkin

SAVONIA UNIVERSITY OF APPLIED SCIENCES

THESIS
Abstract

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology, Internet of Things			
Author(s) Levkin Dmitrii			
Title of Thesis Towards Sustainable IoT: Current and Voltage Data Analysis in NB-IoT Systems			
Date	31.05.2024	Pages/Number of appendices	62
Client Organisation /Partners Savonia-ammattikorkeakoulu			

This study analyzed voltage and current data from an electrical system operating at 50 Hz to assess its performance and stability. The key findings included consistent average power across segments, stable Root Means Square (RMS)values for voltage and current, and a unity power factor indicating a purely resistive load. The unity power factor, where voltage and current are perfectly in phase, results in real power equaling apparent power, highlighting the system's efficiency. These results suggest a well-regulated, efficient, and stable electrical system with high power quality, providing valuable insights for optimizing system design and management.

Keywords
Data analysis, IoT, NB-IoT, Current, Voltage, Frequency, MATLAB, ThingSpeak, Spectral Analysis, Arduino, DFRobot7000, AC-Network

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Analysis of electrical power systems is critical to ensure efficient and reliable operation. Power systems must maintain stable and high-quality power to meet the needs of various industrial, commercial, and residential applications. One of the most important aspects of power system analysis is understanding the relationship between voltage, current, and power factors. (Dugan, McGranaghan & Beaty 1996, 328-331.)

This study examines voltage and current data from an electrical system operating at a standard frequency of 50 Hz. The main goal is to evaluate the system's performance, stability, and efficiency. By analysing key metrics such as average power, RMS voltage and current, and power factor, we strive to provide a comprehensive assessment of system performance.

This analysis highlights the power factor, indicating the voltage and current phase relationship. A power factor of unity, when voltage and current are perfectly in phase, is usually associated with purely resistive loads. This condition is ideal because it means all the supplied power is effectively used as active power without any reactive power loss. Understanding these dynamics can help optimize the design and management of power systems to ensure efficient and reliable operation.

This report will detail the methods used to analyse the data, present key findings, and discuss their implications for electrical system performance and control. The findings from this study are expected to contribute to the broader energy field, especially in improving power quality and system stability.

While writing this thesis, I utilized ChatGPT, a language model developed by OpenAI (OpenAI 2024), to assist with language refinement and editing. ChatGPT supported ensuring the thesis content is clear, coherent, and professionally presented. I employed this tool to review my writing for grammatical accuracy, improve sentence structure, and enhance overall readability. This integration of AI tools demonstrates the practical application of advanced technologies not only in technical research but also in academic writing. It highlights the potential benefits of AI in enhancing the quality and clarity of academic documents.

2 BACKGROUND

This section offers foundational information on the primary topics that served as the basis for the study, specifically focusing on IoT (Internet of Things) and the intricacies of data analysis within this domain.

2.1 Internet Of Things

The Internet of Things (IoT) is a network comprising computing devices, mechanical and digital machines, objects, or even people, each equipped with unique identifiers and the capability to exchange data over a network autonomously without requiring direct human interaction. This enables them to gather, analyze, manage, and transmit data to other objects using software, applications, or technical devices. The evolution of IoT stems from the merging of wireless technologies, micro-electromechanical systems, microservices, and the internet (MEMS). Expected to have a significant impact on various economic sectors, the importance of the IoT is undeniable. This interconnected web of devices has unleashed many opportunities, revolutionizing industries, enhancing efficiency, and elevating the overall quality of life. (Goyal 2021.)

2.2 Evolution of IoT

The concept of IoT has evolved significantly since its inception. Kevin Ashton coined the term in 1999 while working at Procter & Gamble (Lueth 2014). The early development of IoT was driven by advancements in RFID (Radio-Frequency Identification) technology, which enabled the tracking and identification of objects wirelessly. (Mazlan 2023.)

Over the years, the convergence of several key technologies has accelerated the growth and adoption of IoT:

- Wireless Communication: Advances in wireless technologies, such as Wi-Fi, Bluetooth, and cellular networks, have enabled seamless connectivity between devices. (Misra 2017.)
- Micro-Electromechanical Systems (MEMS): These small mechanical and electrical components have enabled the development of compact, low-power sensors and actuators. (Gardner, Varadan & Awadelkarim 2001,9-10.)
- Internet Protocol (IP): The adoption of IPv6 has expanded the number of available IP addresses, facilitating the connection of many devices.
- Cloud Computing: The cloud provides scalable storage and processing power, enabling the analysis of large datasets generated by IoT devices.
- Big Data Analytics: Advanced analytics tools allow for extracting valuable insights from the massive amounts of data IoT systems produce.

2.3 Impact on Various Economics Sectors

IoT is transforming numerous industries by enhancing operational efficiency, reducing costs, and creating new business opportunities. Some of the key sectors impacted by IoT include (Huang 2024):

Manufacturing: IoT enables predictive maintenance, real-time monitoring, and automation of manufacturing processes, leading to increased productivity and reduced downtime.

Healthcare: IoT devices such as wearables and remote monitoring systems improve patient care by providing doctors and caregivers with real-time health data.

Agriculture: IoT applications include precision farming, automated irrigation systems, and livestock monitoring, which optimize resource use and increase crop yields.

Smart Cities: IoT technologies contribute to the development of smart cities by improving urban infrastructure, traffic management, and public safety through connected sensors and data analytics.

Energy: Smart grids and energy management systems use IoT to monitor and optimize energy consumption, reducing costs and environmental impact.

2.4 Opportunities and Challenges

The adoption of IoT offers numerous opportunities for innovation and efficiency (Chen 2014):

- Data-Driven Decision Making: IoT provides real-time data that can be analyzed to make informed decisions, improving operational efficiency and customer satisfaction.
- Automation and Control: Automated systems and processes reduce the need for manual intervention, leading to cost savings and increased accuracy.
- Enhanced Customer Experiences: IoT enables personalized services and products by gathering and analyzing customer data.

However, the widespread adoption of IoT also presents several challenges (Roman 2013):

- Security and Privacy: Protecting the vast amounts of data generated by IoT devices from cyber threats and ensuring user privacy are significant concerns.
- Interoperability: The lack of standardization across IoT devices and platforms can hinder seamless integration and data exchange.
- Scalability: Managing and processing the large volumes of data generated by IoT systems requires scalable infrastructure and robust data management strategies.

2.5 Future Trends

The future of IoT is expected to be shaped by several emerging trends:

Edge Computing: Processing data closer to where it is generated (at the edge) reduces latency and bandwidth usage, enhancing real-time decision-making capabilities. (Valero 2023.)

Artificial Intelligence (AI): Integrating AI with IoT enables more sophisticated data analysis and automation, leading to more intelligent and autonomous systems. (Gülen 2023.)

5G/6G Technology: The rollout of 5G/6G networks will provide faster and more reliable connectivity, supporting the proliferation of IoT devices and applications. (Li 2020.)

2.6 Narrowband Internet of Things (NB-IoT)

Narrowband Internet of Things (NB-IoT) is a standards-based LPWA technology specifically designed to improve the deployment of IoT devices and services. It addresses the critical requirements of IoT applications by providing:

- Enhanced Coverage: NB-IoT offers superior indoor and rural area coverage. It achieves this by using a narrow bandwidth of 180 kHz, which allows signals to penetrate deep into buildings and reach remote areas more effectively than traditional cellular technologies.
- The main frequency ranges for NB-IoT operation include bands allocated by 3GPP, typically from 700 MHz to 2100 MHz, depending on regional availability and regulatory considerations. These frequency allocations ensure NB-IoT can coexist with existing cellular networks, providing robust and scalable connectivity for IoT devices globally. (Mohamed 2023.)
- Power Efficiency: One of the standout features of NB-IoT is its ability to support devices with very low power consumption. This is crucial for IoT applications where devices must operate for years on a single battery charge. NB-IoT can support battery life exceeding ten years for many applications.
- System Capacity: NB-IoT significantly increases the number of devices connected to a single cell. This is achieved through its efficient use of the spectrum and support for a massive number of connections per cell, making it ideal for applications that require large-scale deployment of IoT devices. (GSMA.)

2.7 Technical Specifications and Features

Physical Layer Enhancements: NB-IoT introduces new physical layer signals and channels optimized for extended coverage. These include narrowband signals that are more robust against interference and noise and can operate effectively in challenging environments. (Mohamed 2023.)

Low Complexity Devices: NB-IoT devices are designed to be simple and cost-effective. Although their initial cost is comparable to GSM/GPRS modules, the technology's simplicity is expected to drive rapid cost reductions as adoption grows. (Mohamed 2023.)

Coexistence with Existing Networks: NB-IoT can be deployed in-band within an LTE carrier, in the guard band of an LTE carrier, or standalone in a dedicated spectrum. This flexibility ensures that NB-IoT can coexist with existing 2G, 3G, and 4G networks, making it easier for network operators to deploy the technology.

Security Features: NB-IoT inherits the robust security features of LTE, including:

User Identity Confidentiality: Protects the identity of users by encrypting the data transmitted between the device and the network. (Digi International 2024.)

Entity Authentication: Ensures that devices are authenticated before accessing network services, preventing unauthorized access. (Digi International 2024.)

Data Integrity: Ensures that the data transmitted is not tampered with during transmission. (Digi International 2024.)

Mobile Equipment Identification: Tracks and manages the devices connected to the network. (GSMA.)

2.8 Applications and Use Cases

NB-IoT is ideal for a wide range of applications that require low power consumption, extended battery life, and reliable connectivity in challenging environments. Some prominent use cases include:

Smart Metering: Water, gas, and electricity meters can use NB-IoT to send usage data to utility companies in real time, facilitating efficient resource management and billing. (Airtel Business 2022.)

Smart Agriculture: NB-IoT can connect various sensors in agricultural fields to monitor soil moisture, weather conditions, and crop health, enabling precision farming and optimizing resource use. (Digi International 2024.)

Intelligent Cities: NB-IoT supports innovative city applications such as bright street lighting, waste management, and parking solutions, contributing to improved urban infrastructure and services. (Airtel Business 2022.)

Asset Tracking: Businesses can use NB-IoT to track the location and condition of assets during transit, ensuring better supply chain management and reducing losses. (Airtel Business 2022.)

Health Monitoring: Wearable devices and remote health monitoring systems can use NB-IoT to provide continuous health data, improving patient care and management. (Sylvest 2023.)

2.9 Challenges and Future Prospects

Despite its advantages, NB-IoT faces specific challenges:

Interoperability: Ensuring interoperability between different NB-IoT devices and networks is critical for seamless connectivity. NB-IoT has evolved through different versions, notably Cat-NB1 and Cat-NB2. Cat-NB1, introduced in 3GPP Release 13, offers basic NB-IoT functionality with downlink peak data rates up to 26 kbps and uplink peak data rates up to 62 kbps. Cat-NB2, introduced in 3GPP Release 14, enhances these capabilities with faster data rates, improved location accuracy, and better mobility support. (Vos 2022.)

Scalability: Managing many devices and data traffic requires robust and scalable network infrastructure. Managing many devices and data traffic requires robust and scalable network infrastructure. While NB-IoT indeed has good system capacity, which allows for many devices to be connected within a single cell, the challenge lies in the infrastructure's ability to manage and maintain performance as the number of connected devices grows. The network infrastructure must be robust enough to handle the volume of devices and the data traffic they generate. This includes maintaining data integrity, minimizing latency, and ensuring reliable connectivity across all devices. (AVSystem 2020.)

Regulation and Standards: Adherence to regulatory standards and the continuous development of global standards is essential for widespread adoption. (Velez 2019.)

2.10 Internet of Things Analytics

IoT analytics involves systematically analyzing data produced and collected by IoT devices, employing specialized data analytics methods and tools. IoT data analytics aims to convert vast quantities of unorganized data from diverse devices and sensors within the Internet of Things framework into actionable insights. This process involves several steps: data collection, data analysis, and monitoring. (Hakme, Gupta, Kalaspurkar 2022.)

1. Data Collection: IoT devices generate a wide range of data types, such as sensor readings, device logs, and environmental data. This data is collected in real time and transmitted to central data storage systems, often utilizing cloud computing infrastructure for scalability and accessibility.
2. Data Analytics: In this phase, the raw data collected from the sensors undergoes processing and analysis. This involves several sub-steps:

- Data Preprocessing: Cleaning the data to remove any noise or errors, integrating data from various sources, transforming the data into a suitable format, and reducing the data to make it more manageable while retaining critical information.
- Data Analysis: Applying various analytical techniques such as statistical analysis, machine learning algorithms, and other advanced methods to extract meaningful insights from the pre-processed data. This can help identify patterns, trends, and anomalies. (Matellio 2023.)

3. Data Monitoring Applications:
   The last step is to visualize the analyzed data to make it actionable for decision-makers. This involves creating dashboards, charts, and graphs to present the insights clearly and intuitively. These visualizations help stakeholders understand the data and use it to make informed decisions. Data monitoring applications can provide real-time updates and alerts, ensuring timely responses to any issues or opportunities identified through the data analysis. (Hakme, Gupta & Kalaspurkar 2022.)

2.11 Types of Data Analytics

Descriptive analytics involves interpreting real-time data from IoT-connected devices to monitor performance and ensure optimal functioning. It detects anomalies, assesses usage patterns, locates assets, and provides insights into machine outputs. For example, an intelligent factory might use descriptive analytics to monitor the machinery's status and detect deviations from normal operating conditions. (DeTore 2023.)

Diagnostic analytics delves into the reasons behind observed phenomena. Analyzing IoT data identifies inefficiencies, anomalies, and performance issues, helping diagnose underlying problems when devices are not operating efficiently. For instance, if an intelligent HVAC system consumes more energy than usual, diagnostic analytics can help identify whether it is due to a malfunctioning component or an external factor like weather conditions. (DeTore 2023.)

Predictive analytics employs machine learning to forecast the likelihood of future events. Analyzing historical data identifies trends and predicts outcomes, enabling organizations to anticipate future scenarios and take proactive measures to influence desired outcomes based on real-time IoT data.

For example, predictive analytics can forecast equipment failures in a manufacturing plant, allowing maintenance to be scheduled before a breakdown occurs. (Insightsoftware 2023.)

Prescriptive analytics offers actionable insights derived from descriptive, diagnostic, a nd predictive analytics. Based on the analysis of IoT data, it recommends specific actions to prevent failures, enhance efficiency, optimize outputs, and improve overall performance. For instance, an intelligent energy management system might prescribe optimal settings for heating and cooling systems to minimize energy consumption while maintaining comfort levels. (Insightsoftware 2023.)

2.12 Why Use IoT Analytics?

IoT devices have ushered in an era of unprecedented data generation. This surge in data volume, velocity, and complexity drives the need for robust IoT Analytics to manage the deluge of data that IoT devices produce. The primary reasons for using IoT analytics include (Gopalakrishnan 2023.):

Real-Time Decision Making:

- The real-time nature of IoT data necessitates rapid decision-making. IoT Analytics empowers organizations to process and analyze data as it is generated, enabling timely responses and actions. For example, in an intelligent grid, real-time analytics can adjust power distribution based on current demand and supply conditions. (Gopalakrishnan 2023.)
- IoT data often arrives in various formats from diverse sources, challenging data integration and cohesiveness. IoT Analytics platforms are designed to handle this complexity effectively, ensuring that data from different devices and systems can be combined and analyzed. (IoT.Business.News 2023.)
- IoT Analytics goes beyond data organization and processing. It leverages advanced analytics, including machine learning, to identify patterns and anomalies within the data. This capability enables organizations to make predictive decisions, offering a proactive rather than a reactive approach. For instance, machine learning models can analyze data from smart meters to predict energy consumption patterns and suggest ways to reduce usage during peak hours. (Gülen 2022.)
- Furthermore, IoT Analytics helps reduce operational costs and enhance overall efficiency by optimizing processes and resource allocation. By analyzing data from IoT devices, organizations can identify inefficiencies, streamline operations, and allocate resources more effectively. For example, in logistics, IoT analytics can optimize routing and scheduling to reduce fuel consumption and delivery times. (Gopalakrishnan 2023.)

2.13 IoT Analytics Platforms

IoT analytics platforms empower businesses to analyze and visualize sensor data from internet-connected devices, providing critical insights for various applications. Examples of these platforms include AWS IoT Analytics, which offers comprehensive tools for collecting, processing, and analyzing large volumes of data in real-time; Microsoft Azure IoT, known for its scalable and secure edge-to-

cloud solutions; and IBM Watson IoT Platform, recognized for its flexibility and robust analytics capabilities. Other notable platforms include ThingSpeak, which is user-friendly for beginners and supports advanced analytics through MATLAB; Oracle IoT Cloud Service, which excels in predictive analytics and forecasting; and Datadog, which provides extensive monitoring and performance analysis for IoT devices. These platforms enable businesses to harness the full potential of IoT data, enhancing decision-making, operational efficiency, and predictive maintenance capabilities. (G2 2024.)

2.12.1 Benefits of IoT Analytics Platforms

The Critical Factors and Benefits of IoT Analytics (David 2021):

Flexibility: Analogous to individuals adept at adapting to diverse environments, IoT analytics platforms exhibit a capacity to accommodate and analyze heterogeneous data streams. This inherent flexibility facilitates nuanced understanding and interpretation of multifaceted datasets, enabling businesses to derive actionable insights across various operational contexts.

Future-proof Development and Scalability: With enduring educational frameworks, IoT analytics platforms furnish enterprises with robust infrastructures conducive to perpetual evolution and expansion. This scalability engenders an agile analytical ecosystem capable of seamlessly accommodating escalating data volumes and evolving technological paradigms, safeguarding long-term viability and competitiveness. (David 2021.)

Drag and Drop Interface: Drawing parallels with intuitive interfaces ubiquitous in contemporary digital interfaces, IoT analytics platforms furnish users with an accessible modality reminiscent of assembling coherent narratives from disparate elements. This user-centric design ethos democratizes data exploration and analysis, mitigating entry barriers and engendering inclusivity within the analytical landscape.

Easy Integration: Emblematic of synergistic collaborations inherent in interconnected systems, IoT analytics platforms are conduits for harmonizing disparate data reservoirs and operational workflows. This seamless integration augments organizational interoperability, fostering synergies between divergent datasets and engendering a holistic understanding of complex phenomena. (David 2021.)

Device Health Maintenance: Analogous to preventive healthcare regimes predicated on proactive wellness management, IoT analytics platforms proffer real-time monitoring capabilities akin to diagnostic vigilance in industrial settings. This anticipatory approach to device management mitigates operational disruptions, enhances asset longevity, and fortifies organizational resilience against unforeseen contingencies. (David 2021.)

Cloud Agnostic: Reflective of vendor-agnostic principles underscored by interoperable ecosystems, cloud-agnostic IoT analytics platforms afford enterprises the latitude to select from a pantheon of cloud service providers without entailing vendor lock-in or compromise on data sovereignty. This vendor-agnostic architecture augurs a competitive marketplace conducive to innovation while assuring regulatory compliance and data integrity. (David 2021.)

2.12.2 ThingSpeak platform

ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud. You can send data to ThingSpeak from your devices, create instant visualizations of live data, and send alerts. (ThingSpeak 2024.)

2.12.3 ThingSpeak Features

ThingSpeak provides a comprehensive suite of features for IoT data collection and analysis:

Customizable Channels: Users can create private channels for secure data collection and analysis. Public channels can also be accessed, and data can be transferred to private channels. (Think-biganalytics.)

Data Writing Formats: It supports various data writing formats, including REST API, MATLAB, and MQTT, facilitating efficient data distribution across different channels. (Mathworks.)

Integration with Third-party Applications: Seamless integration with external applications enhances the platform's versatility. (Mathworks.)

2.12.4 The Architecture of ThingSpeak

Figure 4 (Mathworks.) depicts the architecture of an IoT (Internet of Things) ecosystem using ThingSpeak and MATLAB, highlighting the flow of data from smart connected devices to data aggregation, analysis, and algorithm development.

Smart Connected Device: At the core of the architecture are smart connected devices equipped with sensors. These devices capture diverse data types, from environmental conditions to user interactions. Serving as the primary data sources, they continuously generate real-time data streams, forming the foundation of the IoT ecosystem. (Mathworks.)

Data Aggregation and Analytics: The subsequent layer involves mechanisms for aggregating, processing, and analyzing the vast influx of data originating from connected devices. ThingSpeak offers comprehensive functionalities for data collection, storage, and visualization. This enables users to consolidate disparate datasets into coherent streams and extract actionable insights through advanced analytics modules. (Mathworks.)

Algorithm Development: The provision for algorithm development complements the data analytics layer. Users can design, customize, and implement algorithms tailored to specific use cases and analytical requirements. Leveraging MATLAB, a renowned platform for numerical computation and data analysis, ThingSpeak empowers users to develop sophisticated algorithms for processing IoT data effectively. (Mathworks.)

Sensor Analytics: Sensor analytics is a particular emphasis in algorithm development. It entails methodologies for extracting meaningful patterns and insights from sensor data. ThingSpeak facilitates sensor analytics through MATLAB-based algorithms, enabling users to perform advanced analyses such as anomaly detection, predictive modeling, and pattern recognition. (Mathworks.)

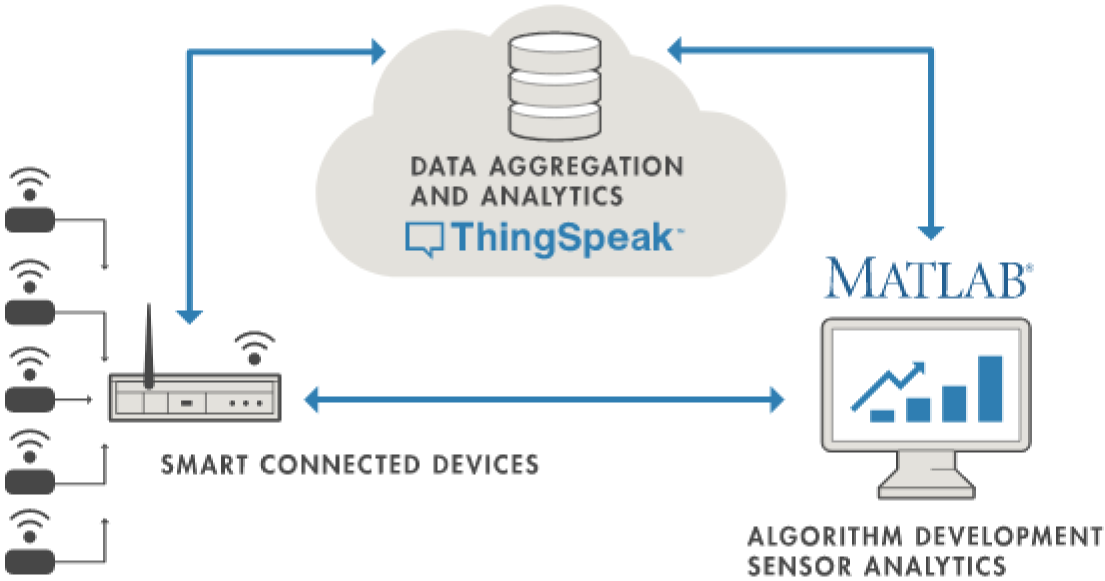


Figure 1. Architecture of ThingSpeak

2.14 Volts, Current and the Basic Concepts of Electricity

Electricity is a fundamental form of energy characterized by the flow of electrons between two points due to a disparity in electrical potential, generating an electric current. Understanding the basic concepts of electricity is crucial for working with IoT devices and systems, which often rely on electrical components for operation and data collection. (Rome 2023.)

Voltage, also known as electric potential difference, represents the driving force behind electron flow in a circuit. It is the difference in electric potential energy between two points in a circuit and is measured in volts (V). Voltage can be thought of as the pressure that pushes electrons through a conductor. Without a sufficient voltage difference, current cannot flow through the circuit. (Rome 2023.)

Current is the rate at which electrons move through a conductor, quantified in amperes (A) or amps. It signifies the flow of electric charge and is one of the critical parameters in any electrical circuit. There are two types of current (Akram 2021.):

Direct Current (DC): Where the flow of electrons is in a single direction.

Alternating Current (AC): Where the flow of electrons periodically reverses direction.

Resistance is a property of a material or component that impedes or regulates the flow of electrical current through it. It is measured in ohms (Ω) and is symbolized by the letter "R." Resistance determines how much current will flow for a given voltage applied across the component. The quintessential component designed to regulate this resistance precisely is the "resistor." (Akram 2021.)

2.15 Ohm’s Law

Ohm’s Law (Antonov 2016.) is a fundamental principle in electrical engineering that describes the relationship between voltage (V), current (I), and resistance (R) in an electrical circuit. It is expressed by the formula:

$$V = IxR$$

V is the voltage across the circuit, I is the current flowing through the circuit, and R is the resistance within the circuit. (Javatpoint.)

This law is crucial for calculating one of these values if the other two are known, allowing for the design and analysis of electrical circuits.

Understanding these basic concepts is essential for designing and troubleshooting IoT devices and systems, which often incorporate electrical components such as sensors, actuators, and microcontrollers. Here are a few applications:

Sensor Interfaces: Sensors convert physical phenomena (e.g., temperature, light, pressure) into electrical signals (voltages or currents) that can be measured and analyzed.

Power Management: IoT devices often operate on battery power, requiring careful management of voltage, current, and resistance to maximize battery life and ensure reliable operation.

Circuit Design: Effective circuit design involves selecting appropriate components and configuring them to achieve desired electrical characteristics, ensuring the device functions correctly under various conditions.

2.16 Spectral Analysis

Spectral analysis is one of the most robust tools for data processing. It analyzes data and indicates characteristic frequencies to suppress noise. The most straightforward signal is a sine wave, defined by its wavelength, amplitude, and phase. Wavelength defines the length of an oscillation. If this wave propagates at a particular speed (for example, a value v in m/s), then you can define its frequency f as (Mathworks 2024):

$$f = \frac{v}{\gamma}$$

which is proportional to the inverse of the wavelength λ.

Wavelength (λ): The distance over which the wave's shape repeats, measured in meters (m). (Blateyron 2023.)

Amplitude (a): The height of the wave, representing the maximum value of the wave's displacement from its mean position. (Blateyron 2023.)

Phase (Φ): The position of a point in time on a waveform cycle, indicating the shift of the wave relative to a reference point. (Blateyron 2023.)

Frequency (f): The number of oscillations or cycles per second, measured in Hertz (Hz). It is inversely proportional to the wavelength. (Blateyron 2023.)

In surface texture analysis, signals represent profiles or surfaces. Because signals are in the spatial domain, wavelengths are often convenient instead of frequencies. A data set's spectrum is called a coordinate function, and it is obtained using specific algorithms to analyze the data. (Podulka 2022.)

Once a data set's spectrum is collected, it is further analyzed using various algorithms to extract meaningful properties. This process helps identify patterns, detect anomalies, and understand the signal's underlying structure. (Podulka 2022.)

2.15.1 Steps in Spectral Analysis

1. Data Collection is the first step in spectral analysis to collect the data that needs to be analyzed. This data can come from various sources, such as sensors, microphones, or other measuring devices. The collected data should be accurately sampled to ensure that it represents the signal of interest without significant loss of information. (Airbyte 2024.)

2. Data preprocessing involves preparing the raw data for analysis. This step includes several critical processes:

- DC Removal: Removing the DC component (the average or mean value) from the signal is essential to focus on the oscillatory components. The DC component can be removed by subtracting the mean value of the signal from each data point. (Airbyte 2024.)
- Windowing Function: Applying a windowing function helps to mitigate the effects of discontinuities at the boundaries of the sampled signal. Standard windowing functions include Hamming, Hanning, and Blackman windows. (Airbyte 2024.)

3. Data Transformation. Once the data is preprocessed, it can be transformed to reveal its frequency content. The Fourier Transform is the most common transformation used in spectral analysis, which converts the time-domain signal into its frequency-domain representation. (Donald B. Percival)

- Fourier Transform: The Fourier Transform decomposes a signal into its constituent sine and cosine components, providing a spectrum of the signal's frequency content. The Discrete Fourier Transform (DFT) is used for discrete data, often implemented efficiently using the Fast Fourier Transform (FFT) algorithm.
- Power Spectrum: The power spectrum represents power distribution into the signal's frequency components. It is beneficial in identifying dominant frequencies and understanding the signal's energy distribution. (Proakis & Manolakis 1996,232-240.)

4. Data Analysis and Monitoring

- After transforming the data, the next step is to analyze and interpret the results to extract meaningful insights. This involves (Airbyte 2024):
- Frequency Analysis: Identifying the significant frequencies present in the signal and their amplitudes.
- Harmonic Analysis: Investigate the signal's harmonic content to understand its periodic components.
- Noise Identification: Distinguishing between signal and noise components to improve the signal quality.

- Anomaly Detection: Detecting unexpected changes or anomalies in the signal that may indicate faults or unusual events.
- Regular monitoring of the spectral content of signals is essential in many applications, such as predictive maintenance in industrial settings. Identifying changes in the frequency spectrum can indicate equipment degradation or impending failure.

2.15.2 The Fourier Transform

The Fourier Transform is a mathematical technique used in spectral analysis to decompose a waveform (a function of time or space) into an equivalent representation characterized by sine and cosine functions of varying frequencies. This method is fundamental in signal processing, allowing any signal shape to be rewritten as a sum of sinusoids. (Thefouriertransform 2022.)

Critical Aspects of the Fourier Transform (Thefouriertransform 2022.):

- Decomposition: The Fourier Transform breaks down a complex signal into its constituent sine and cosine components. This is particularly useful for analyzing periodic signals.
- Frequency Domain Representation: The Fourier transform transforms a signal from the time (or spatial) domain to the frequency domain, providing insight into its frequency content.
- Applications: The Fourier Transform is widely used in various applications, including audio signal processing, image processing, and vibration analysis.

The mathematical expression for the Fourier Transform F(f)F(f) of a continuous time-domain signal g(t). (Thefouriertransform 2022.)

$$F\{g(t)\} = G(f) = \int_{-\infty}^{\infty} g(t)e^{-i2\pi ft}dt$$

Where:

- g(t) is the time-domain signal,
- G(f) is the frequency-domain representation of the signal,
- i is the imaginary unit,
- t is time,
- f is frequency.

This transform allows for analyzing the signal's frequency components, making it easier to study and manipulate the signal for various applications.

The Fourier Transform is integral to many areas of technology and science:

- Signal Processing: Filters noise from signals, compresses data, and analyzes signal frequency content.
- Engineering: Essential for vibration analysis, structural health monitoring, and acoustics.

- Medical Imaging: Applied in MRI and CT scans to reconstruct images from raw data.
- Communications: Facilitates modulation and demodulation of signals in telecommunications.

2.17 Power Analysis in Electrical Circuit

Understanding the different types of power in electrical circuits is crucial for analyzing and optimizing electrical system performance. The key types of power include real power, reactive power, apparent power, average power, instantaneous power, and power factor. (Dugan, McGranaghan & Beaty 1996, 171-177.)

Electrical devices consume real power, measured in watts (W), to perform valuable work. It is the product of the voltage (V), current (I), and the cosine of the phase angle (θ) between them. (Electrical Technology 2013.)

$$P = VI\cos(\theta)$$

- V: Voltage, the potential difference, measured in volts (V).
- I: Current, the flow of electric charge, measured in amperes (A).
- θ: Phase angle between the voltage and the current, measured in degrees or radians.
- Cos (θ): The cosine of the phase angle, representing the power factor.

Reactive power(Q), measured in volt-amperes reactive (VAR), represents the power oscillating between the source and the load due to inductive and capacitive elements in the circuit. It performs no useful work but is necessary to maintain the system's electric and magnetic fields. (Electrical Technology 2013.)

$$Q = VIsin(\theta)$$

- Q: Reactive power is measured in volt-amperes reactive (VAR).
- V: Voltage, measured in volts (V).
- I: Current, measured in amperes (A).
- θ: Phase angle between the voltage and the current.
- Sin (θ): The sine of the phase angle.

Reactive power is a "side effect" of using inductive or capacitive loads, or a combination of these two, in AC systems. These reactive components cause the current and voltage to be out of phase, leading to reactive power. While reactive power does not perform useful work, it is necessary to maintain the electric and magnetic fields in inductive and capacitive components. Managing reactive power is crucial to ensure efficient power system operation and to minimize losses. By analyzing the phase difference between voltage and current, the power factor can be determined, which helps distinguish between real and reactive power, optimizing the design and management of power systems. (Munir 2022.)

Apparent power, measured in volt-amperes (VA), is the product of the voltage and current in the circuit without considering the phase angle. It represents the total power flowing in the circuit. (Electrical Technology 2013.)

$$S = VI$$

- S: Apparent power, measured in volt-amperes (VA).
- V: Voltage, measured in volts (V).
- I: Current, measured in amperes (A).

Average power is the total energy consumed over a period, divided by the duration of that period. It is instrumental in AC circuits where the instantaneous power varies with time. (LibreTexts.)

$$Pavg = \frac{1}{T \int_0^T p(t)\, dt}$$

- Pavg: Average power, the total energy consumed over a period, divided by the duration of that period.
- T: The period over which the power is averaged.
- p(t): Instantaneous power as a function of time.

Instantaneous power is the power at any given moment, calculated as the product of the instantaneous voltage and current. (ElectronicsTutorials.)

$$p(t) = v(t) * i(t)$$

- p(t): Instantaneous power is measured in watts (W) at a given moment.
- v(t): Instantaneous voltage, measured in volts (V), at a given moment.
- i(t): Instantaneous current, measured in amperes (A), at a given moment.

The power factor (PF) is the ratio of real power to apparent power. It measures how effectively electrical power is being used. Power factor 1 indicates that all the power is effectively converted into practical work. (Dugan, McGranaghan & Beaty 1996, 36-37.)

$$PF = \frac{P}{S}$$

- PF: Power factor, the ratio of real power to apparent power, dimensionless.

- P: Real power, measured in watts (W).
- S: Apparent power, measured in volt-amperes (VA).

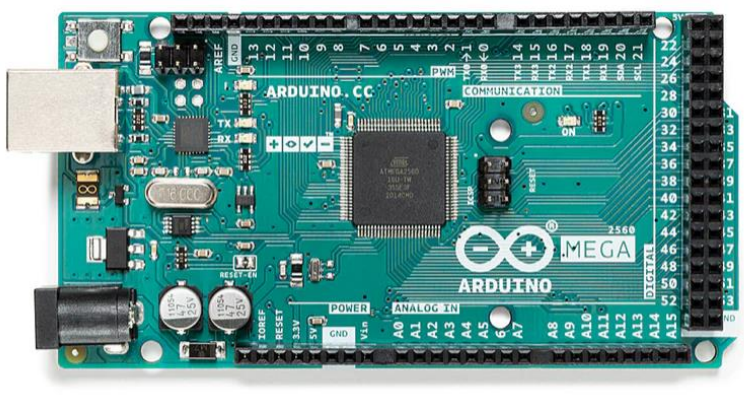
3 HARDWARE

3.1 Arduino Mega 2560

The Arduino Mega 2560 (Figure 2) is a microcontroller board based on the ATmega2560 microcontroller. (Arduino.) It is designed for complex projects that require a lot of input and output pins. The board features:

Table 1. Arduino Mega 2560 Specifications

Digital Input/Output Pins	54 pins, of which 15 can be used as PWM outputs
Analog Inputs	16 analog input pins
UARTs (Serial Ports)	4 hardware serial ports for communication
Clock Speed	16 MHz crystal oscillator
USB Connection	For programming and communication
Power Jack	For external power supply
ICSP Header	For programming the microcontroller
Reset Button	To reset the board



Click to expand

Figure 2. Arduino Mega 2569 Rev3

To start with the Arduino Mega 2560, connect it to a computer using a USB cable or power it with an AC-to-DC adapter or battery. This versatility makes it suitable for various IoT, robotics, and automation applications. (Arduino.)

3.2 DFRobot_SIM7000

The DFRobot_SIM7000 (Figure 3) is an LTE CAT-M1/NB-IoT module that supports multiple frequency bands of LTE-TDD, LTE-FDD, GSM, GPRS, and EDGE. The module offers (DFRobot 2018):

Data Flow: Upstream and downstream data flow peaks at 375kbps.Coverage: 20dB+ coverage gain, allowing wireless communication even in challenging environments like basements. Power Consumption: A stable, low-power communications module suitable for IoT applications requiring reliable and efficient connectivity. (DFRobot 2018.):

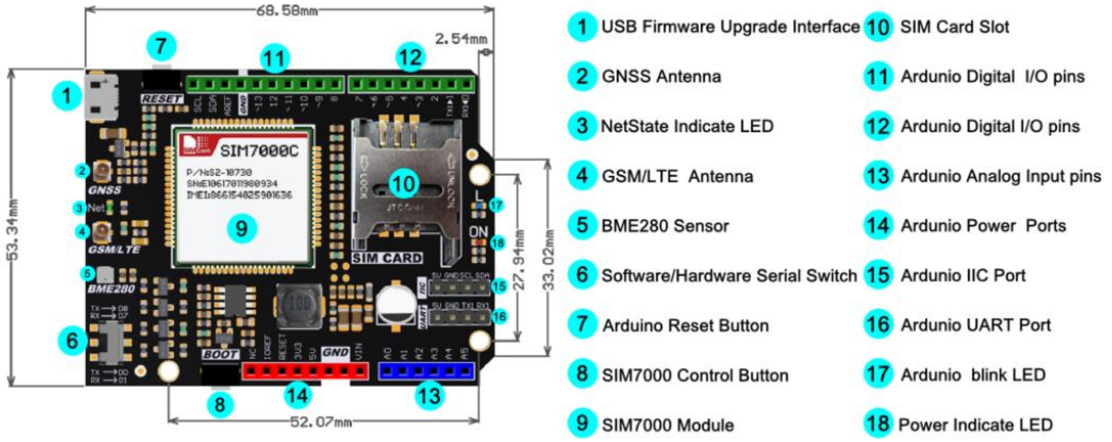


Figure 3. DFRobot_SIM7000

3.3 Ammeter And Voltmeter

In this project, capturing real-time voltage and current values from an AC network using two crucial sensors, a voltmeter, and an ammeter, connected to the Arduino Mega's analog pins A0 and A1, respectively, is paramount.

Voltmeter (SI-9002).

The SI-9002 voltmeter (Figure 4) measures the electrical potential difference across a circuit. This device provides insight into the voltage levels at specific points within the system (Rsdelivers). The SI-9002 voltmeter is connected in parallel to the AC mains, specifically between the phase and neutral wires, to accurately measure the voltage. The voltmeter output is then connected to a level shifter to ensure compatibility with the Arduino's input range before being fed into analog pin A0 of the Arduino Mega. (Tiepie.)

- Connection: Parallel to AC mains (phase and neutral wires)
- Output Range: 0-5V (after level shifter)

Figure 4. SI-9002 Voltmeter

The Fluke i30s ammeter (Figure 5) is used to measure the current flowing through the AC network. This device can provide precise measurements of current (Tme 2024). The Fluke i30s ammeter is connected in series with the load to measure the current accurately. Similar to the voltmeter, the output of the ammeter is connected to a level shifter before being fed into analog pin A1 of the Arduino Mega.

- Connection: Series with the load
- Output Range: 0-5V (after level shifter)

Figure 5. FLUKE I310S

3.4 Real-time clock (RTC)

The DS3231 RTC module (Figure 6) keeps track of the current time and date. It is highly accurate and features an integrated temperature-compensated crystal oscillator and crystal. The RTC module

connects to the Arduino Mega via the I2C bus, with the SDA and SCL lines connected to the corresponding pins on the Arduino (SDA to A4 and SCL to A5). (Aneindia.)

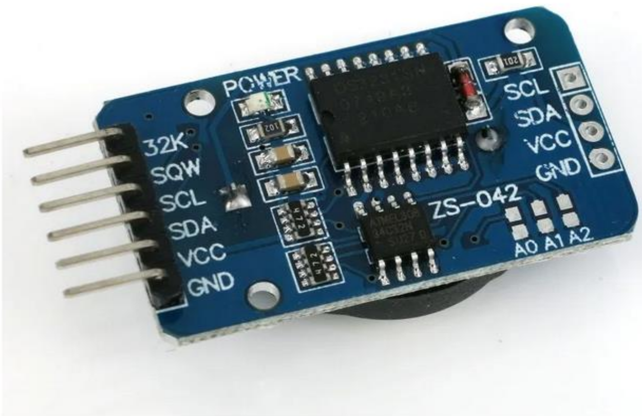


Figure 6. RTC 3231

3.5 PicoScope3000 Series

PicoScope 3000 Series (Elfadistrelec) is a compact, portable oscilloscope connected to a USB cable. (Figure 7) It has four analog channels. Despite its compact size, one main characteristic is its high bandwidth and sampling rate, which is 200 MHz analog bandwidth. (PicoTechnology.)

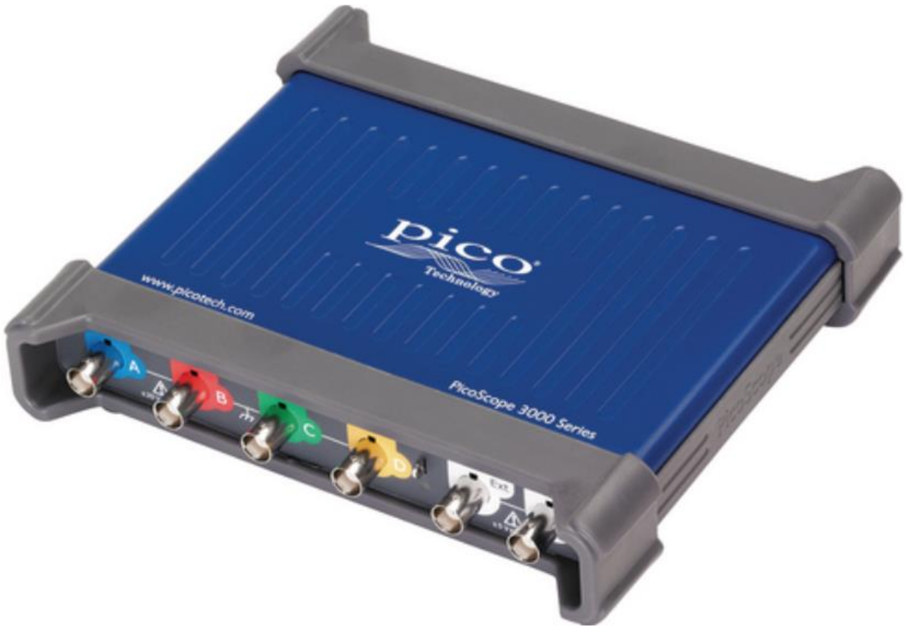


Figure 7. PicoScope

The logic analyzer can display both digital and analog signals simultaneously. It allows setting separate logic thresholds for each 8-bit input port, ranging from –5 V to +5 V. It supports decoding multiple data formats, including CAN, I²C, Ethernet, and more. (PicoTechnology.)

The analyzer shows decoded data in hexadecimal, binary, decimal, or ASCII formats. It provides error frames that can be enlarged to check for noise or signal integrity issues. Advanced logic triggers can be set on analog, digital, or both input channels for complex signal analysis. (PicoTechnology.)

This makes it a powerful tool for detailed and versatile signal troubleshooting.

3.6 Chroma Programmable AC Source

The Chroma programmable AC Power Source(Figure 8) uses Digital Signal Processor (DSP) technology to program voltage signals (Chromausa). It can help test the power input range by changing the voltage or frequency in just one step or setting the slew rate to achieve gradual increases or decreases in voltage and frequency.

The Chroma programmable AC power supply is equipped with an integrated 16-bit precision measurement circuit and built-in utilities for measuring RMS voltage, RMS current, true power, power factor, peak repetitive current, inrush current, current gain, VA (apparent power) and VAR (reactive power). (Chromausa.)

The Chroma programmable AC power supply can provide a high peak output current factor (max. 6) and sufficient transients, the power required to test most switching-type power circuits. Additive programmable control of the on and off phase angle makes the 61600 series the ideal AC source for checking the inrush current. (Chromausa ATE INC.)

Figure 8. Chroma Programmable AC Source 61602

4 PROGRAMMING ENVIRONMENT

Applications for this project are built using Arduino Integrated Development Environment (IDE.) and MATLAB. (MATLABIDE.)

4.1 Arduino Integrated Development Environment (IDE)

Code Compilation: The IDE compiles code written in C/C++ into a format that the microcontroller on the Arduino board can understand. This allows developers to write code using familiar programming languages and download it directly to their Arduino hardware. (Docks. Arduino.)

Serial Monitor: The built-in serial monitor allows developers to interact with their Arduino boards in real time. This feature is handy for debugging code and monitoring sensor data during development. (Docks. Arduino.)

Libraries and Examples: The Arduino IDE supports many libraries and examples that can be easily accessed and integrated into projects. These libraries provide pre-written code for everyday tasks such as sensor interaction, motor control, and communication over protocols such as I2C, SPI, and UART.

User-Friendly Interface: The Arduino IDE's simplicity and versatility make it ideal for amateurs and professionals. Its user-friendly interface and extensive support community make it accessible to users with varying experience levels.

Overall, the Arduino IDE plays a crucial role in the ecosystem by providing a user-friendly platform for programming and developing projects using Arduino boards. Its simplicity and versatility make it an ideal choice for amateurs and professionals. (Docks. Arduino.)

4.2 MATLAB

MATLAB is a desktop and online environment for analysis and design processes, equipped with a programming language that directly expresses matrix and array mathematics. Critical features of MATLAB include:

Live Editor: The Live Editor allows users to create scripts that combine code, output, and formatted text in a single document. This facilitates the development of complex analyses and the documentation of results. (MathWorks.)

Data Visualization and Exploration: MATLAB provides powerful tools for visualizing and exploring data. Users can create a wide range of plots and charts to gain insights into their data. (MathWorks.)

Integration with Other Languages: MATLAB can be used alongside Python, C/C++, JAVA, and other programming languages, making it a versatile tool for various applications. (MathWorks.)

Cloud Capabilities: MATLAB can run in the cloud via MathWorks' cloud services, including AWS and Azure. This allows for scalable and flexible deployment of MATLAB applications, enabling users to leverage cloud computing resources for their projects.

MATLAB is particularly useful for its robust data analysis capabilities, essential for processing and interpreting large datasets generated by IoT devices. Its integration with other languages and cloud capabilities further enhances its utility in complex, data-driven projects. (MathWorks.)

5 PROJECT DESIGN

This section describes the design choices made for this project. It explains the sequence of connecting devices, testing, and visual understanding.

Based on Radomir Pestov's thesis, integrating an AC source with an Arduino-based system using a level shifter is necessary because Arduino's analog-to-digital converters (ADCs) cannot read negative voltages inherent in AC signals.

The Arduino, equipped with a DFRobot NB-IoT shield, facilitates the interface with the conditioned AC signal and processes the obtained data. To address Arduino's limitation of reading negative voltages, a voltage level shifter is used to shift the entire AC signal within the 0-5V range readable by Arduino's ADCs. The voltmeter and ammeter measure the AC voltage and current, with outputs routed through the level shifter to Arduino's analog pins. The Arduino samples these signals at regular intervals, converting them to digital values. The DFRobot NB-IoT shield then transmits this data to the cloud, such as Thingspeak, for real-time monitoring and analysis. This setup ensures accurate measurements and effective remote data access, crucial for IoT applications in monitoring electrical parameters.

Overall, this setup enables the measurement of AC signals by an Arduino-based system, overcoming the limitation of Arduino's inability to read negative voltages directly. The level shifter is crucial in ensuring compatibility and enabling data acquisition and transmission for further analysis and visualization at MATLAB.

5.1 Devices installation

The project's initial phase involved installing and connecting all devices. The first step was setting up the AC source. One wire was connected to the socket, and the other was connected to the device(LED lamp or Heater). These wires have three contacts (Figure 9):

- Green Contact: The ground contact provides a safe path for electric currents in case of a fault. It is essential for safety and is connected to the grounding wire in the electrical system.
- Blue Contact: This is the neutral contact, where the return current wire completes the circuit and returns it to the power source.
- Brown Contact: This is the live contact, where the current-carrying conductor connects. It is typically one of the smaller slots in the socket and is connected to the live wire from the power source.

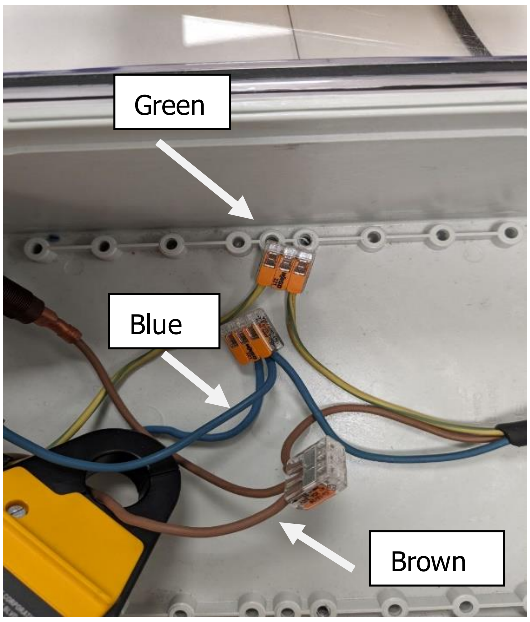


Figure 9. Contacts

A voltmeter is an instrument designed to measure the electrical potential difference between two points in a circuit. In the given setup, the voltmeter is connected using two probes: the red probe is attached to the live (brown) wire, and the black probe is connected to the neutral (blue) wire. This configuration allows the voltmeter to measure the voltage across the load accurately. The measured voltage is crucial for monitoring the electrical system's performance and ensuring its proper functioning. The output from the voltmeter can be connected to an oscilloscope or a level shifter for further analysis and visualization, providing a clear picture of the voltage variations over time.

An ammeter measures the current flowing through a circuit and is essential for understanding the electrical load. In this setup, the ammeter utilizes a clamp around the live (brown) wire to measure the current without breaking the circuit. Additionally, it has two probes: the red probe connects to the live wire, and the black probe connects to the neutral wire. This dual connection allows the ammeter to measure the current flow directly through the live wire and monitor the voltage drop. The ammeter's output is connected to an oscilloscope or level shifter, which provides real-time data for detailed analysis. This setup ensures precise and safe current measurement, critical for analyzing and maintaining the electrical system's efficiency and reliability.

A level converter was created to enable the Arduino to calculate values without reading negative voltages. The RTC 3231 module was used to track time accurately. (Figure 10)

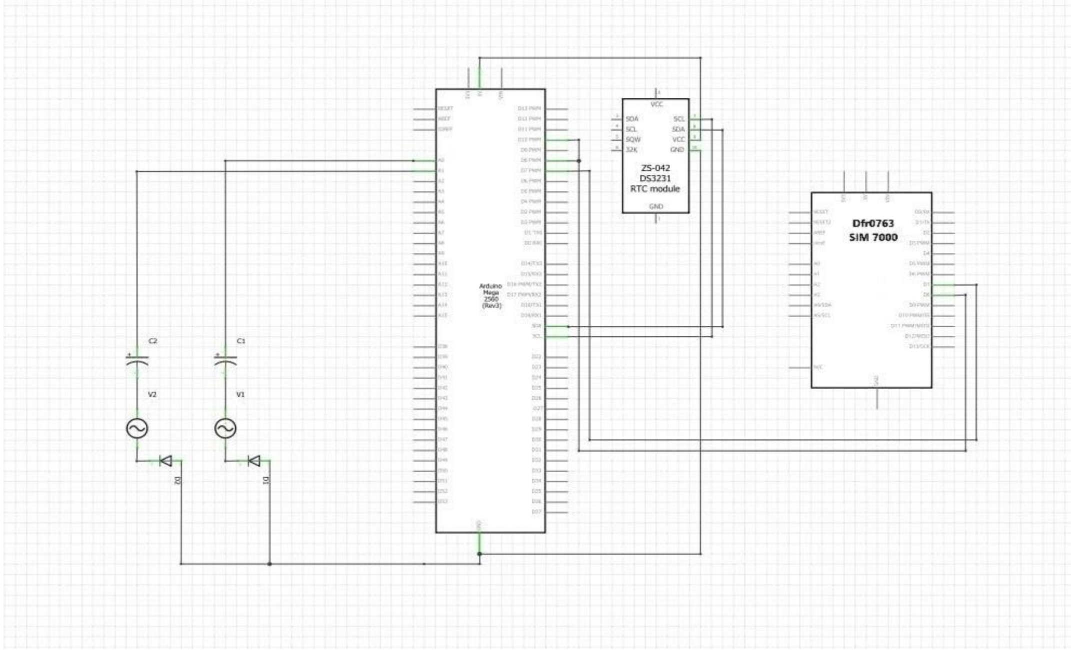


Figure 10. Arduino Mega 2560, RTC 3231, and DFRobot_SIM7000

5.2 Device Testing

5.2.1 The Chroma Programmable AC Source Testing

The Chroma Programmable AC Source was the first device used to test the device. It has a wire connected to a socket and a 3-way socket adapter for connecting devices. After powering up the AC source, the device was connected. A manual setup was carried out using a special control knob. It has the capability of changing the voltage and frequency. Several frequencies were chosen to demonstrate the device's capability of analyzing AC frequency. Finland operates on a 230V supply voltage and a 50 Hz frequency, so the device was tested with frequencies of 50, 48, and 52 Hz to ensure its proper functionality. This setup validated the device's ability to accurately measure and respond to variations in AC frequency data with a 50 Hz frequency was collected and processed in MATLAB.

Key Details:

- Data Duration: The duration of the data collection was 0.5 seconds, sufficient to capture multiple cycles of the 50 Hz signal, ensuring a reliable frequency analysis.
- Sampling Frequency: The sampling frequency was set at 500 Hz, meeting the Nyquist critrion and ensuring accurate capture of the 50 Hz signal.
- FFT Analysis: The FFT analysis converts the time-domain signal into the frequency domain, making it easier to identify and analyze the dominant frequency components.
- Data Points: Each vector contains 256 values, which provides a detailed representation of the signal over the sampled period.

Figures 12 and 14 show the amplitude spectrum of the current signal and voltage at a 50 Hz frequency, acquired using FFT, and Figures 11 and 13 show the angle of voltage and current.

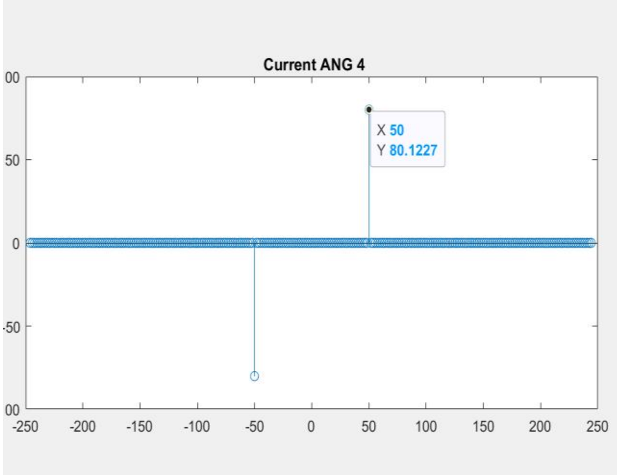


Figure 11. Current 50 Hz Frequency

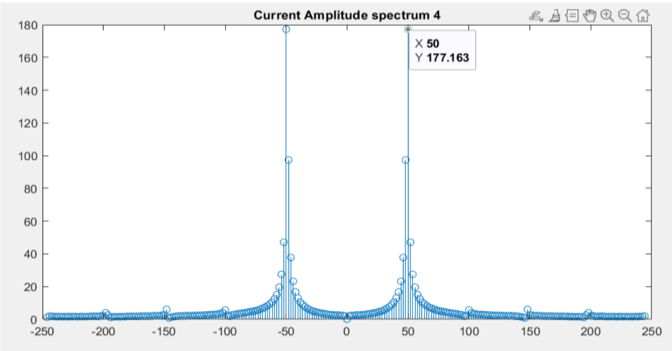


Figure 12. Current Amplitude Spectrum 50 Hz

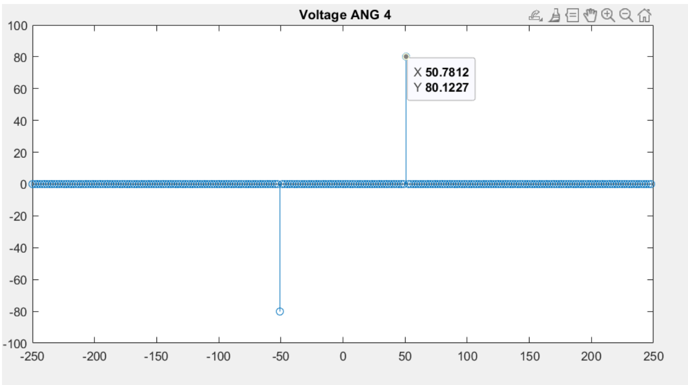


Figure 13. Voltage 50 Hz Frequency

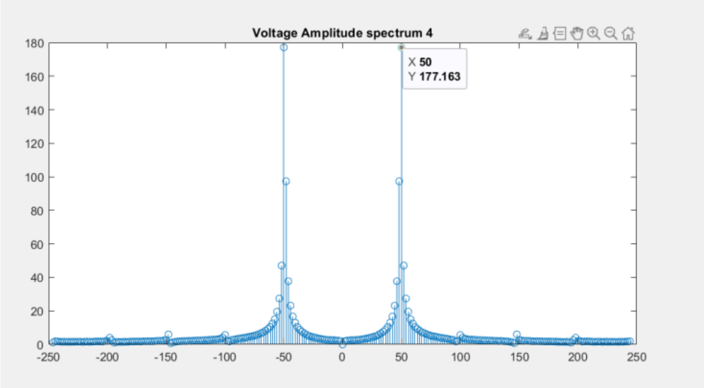


Figure 14. Voltage Amplitude Spectrum 50 Hz

After it was set up, the frequency was 48 Hz and plotted in MATLAB.

- The amplitude spectrum (Figures 16 and 18) for the current and voltage signal peaks at 48 Hz.
- The angel for the current and voltage signal (15 and 17) shows 136.79.
- This peak indicates that the current signal is predominantly at 48 Hz, which validates the accuracy of the measurement setup for this frequency.

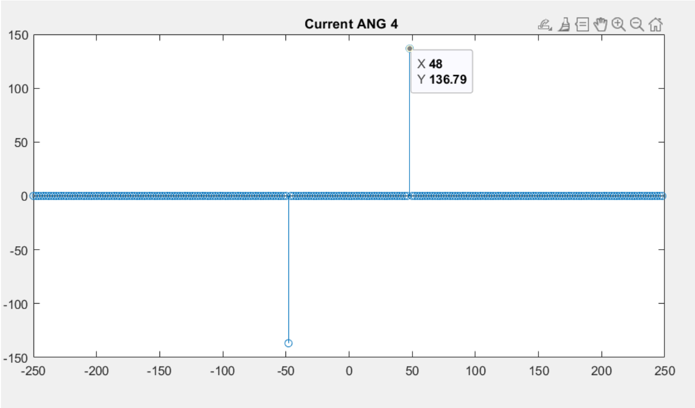


Figure 15. Current 48 Hz Frequency

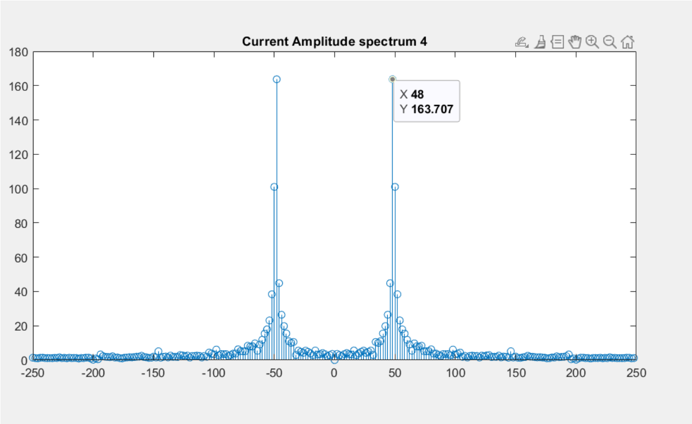


Figure 16. Current Amplitude Spectrum 48 Hz

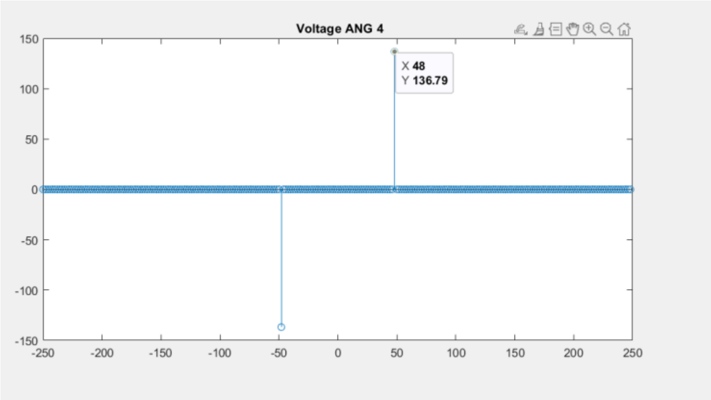


Figure 17. Voltage 48 Hz Frequencies

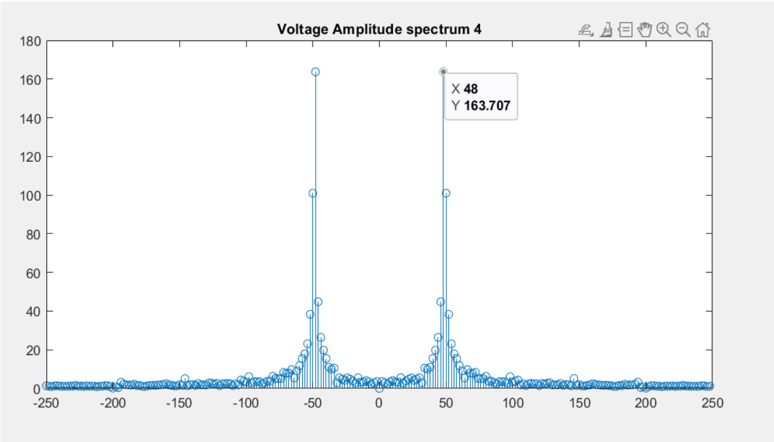


Figure 18. Voltage Amplitude Spectrum 48 Hz

Finally, tests were conducted using a 52 Hz frequency.

- The current and voltage signal's amplitude spectrum(Figures 20 and 22) displays a clear peak at 52 Hz.
- The angel for the current and voltage signal (19 and 21) shows 167.995.
- This peak indicates that the current signal is accurately captured at 52 Hz, confirming the system's capability to measure frequencies slightly above the standard 52 Hz.

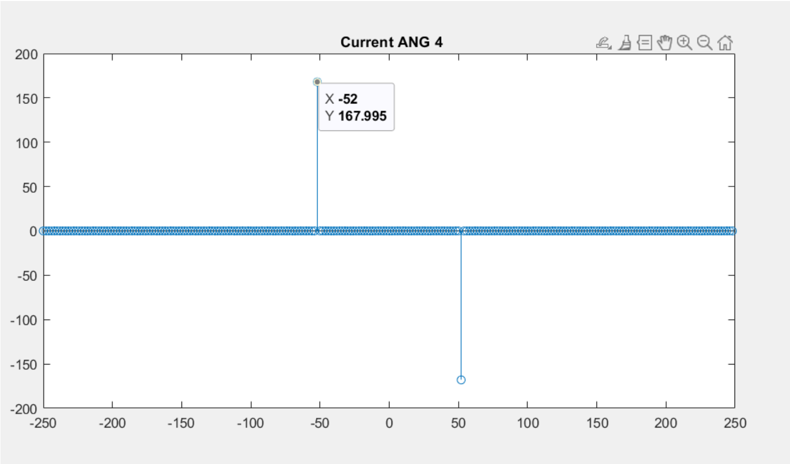


Figure 19. Current 52 Hz Frequencies

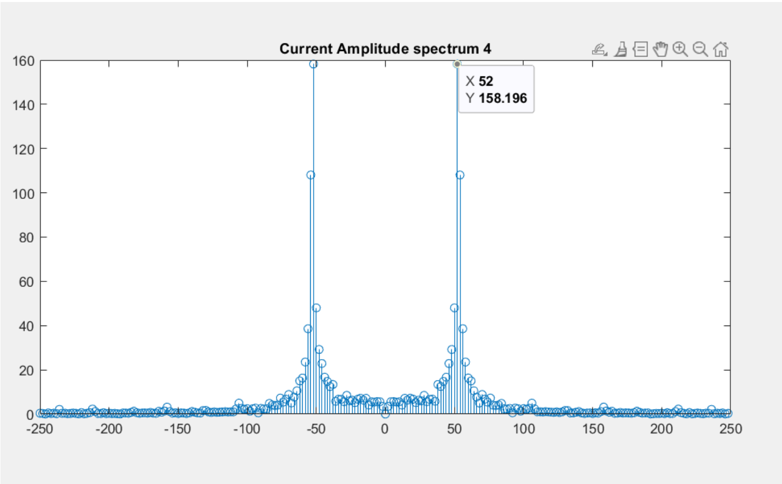


Figure 20. Current Amplitude Spectrum 52 Hz

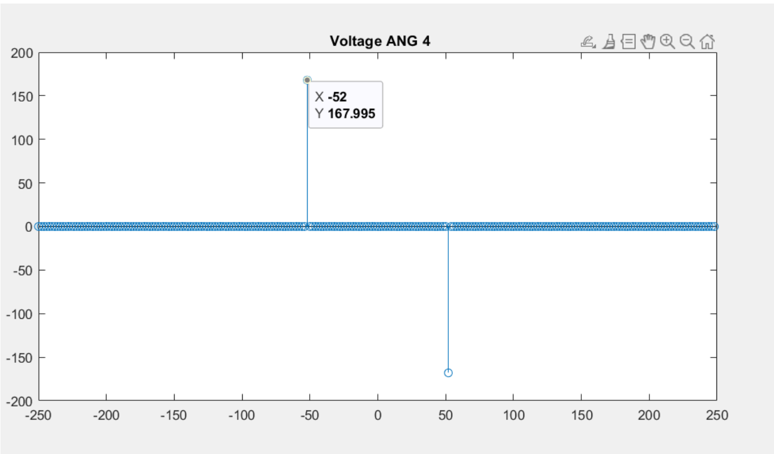


Figure 21. Voltage 52 Hz Frequencies

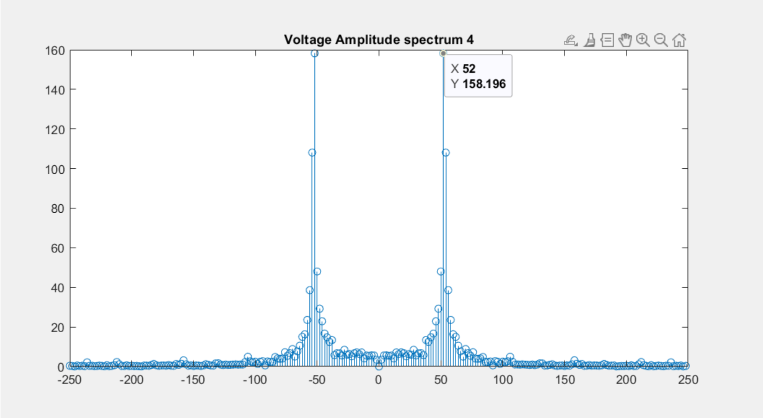


Figure 22. Voltage Amplitude Spectrum 52 Hz

All values in the amplitude spectrum have symmetry due to the properties of the Fourier Transform applied to real-valued signals. This symmetry ensures that for every frequency component in the positive range, there is a corresponding component in the negative range, creating a mirror image. Understanding this symmetry is crucial for correctly interpreting and analyzing frequency components in any real-valued signal.

These tests confirmed that the setup could accurately measure and process AC signals at different frequencies, providing reliable data for further analysis and visualization in MATLAB.

5.2.2 PicoScope 3000 series Testing

The PicoScope 3000 Series is a critical component for testing and visualizing AC signals in this project. This section outlines the testing procedure using the PicoScope 3000 Series to measure and analyze AC signals from the setup. The primary objective is to verify whether the AC signal is 50 Hz, as expected in Finland's 230V, 50 Hz power supply, and to ensure the correct operation of the level shifter.

Device Setup:

Connection to Ammeter and Amperemeter: The probes from the ammeter and amperemeter are connected to PicoScope. The first probe is connected to channel A, and the second to channel B of PicoScope. (Figure 23)

Figure 23. PicoScope’s channels

1) Connection to Laptop:

- PicoScope is connected to a laptop via a USB cable for data acquisition and visualization.

2) Software Setup:

- PicoScope Software Installation: Download the PicoScope 7 T&M desktop program (Technology.) from the official website.

3) Software Configuration:

- Open the PicoScope application and configure channels A and B to display the signal.

4) Testing Procedure:

- Initial Test (Before Level Shifting) (Figure 24)

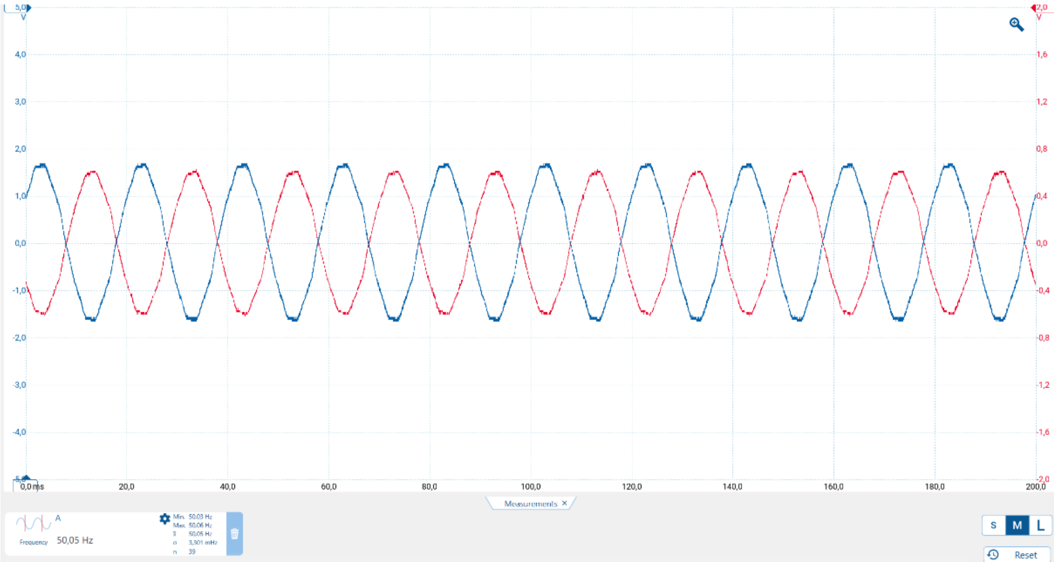


Figure 24. Before Level Shifting

In Figure 24, the Voltage signal (Blue Waveform) oscillates around 0V, with both positive and negative peaks. This is typical for an AC waveform. The peak-to-peak voltage appears to be about ±2V,

giving a total amplitude of around 4V. The signal has a frequency of approximately 50 Hz, as indicated by the periodic nature of the waveform.

Current Signal (Red Waveform) also oscillates around 0V, with positive and negative peaks. The peak-to-peak current is about ±1V, resulting in a total amplitude of around 2V. The frequency is around 50 Hz, similar to the voltage signal, indicating synchronization between voltage and current.

Level Shifter Test:

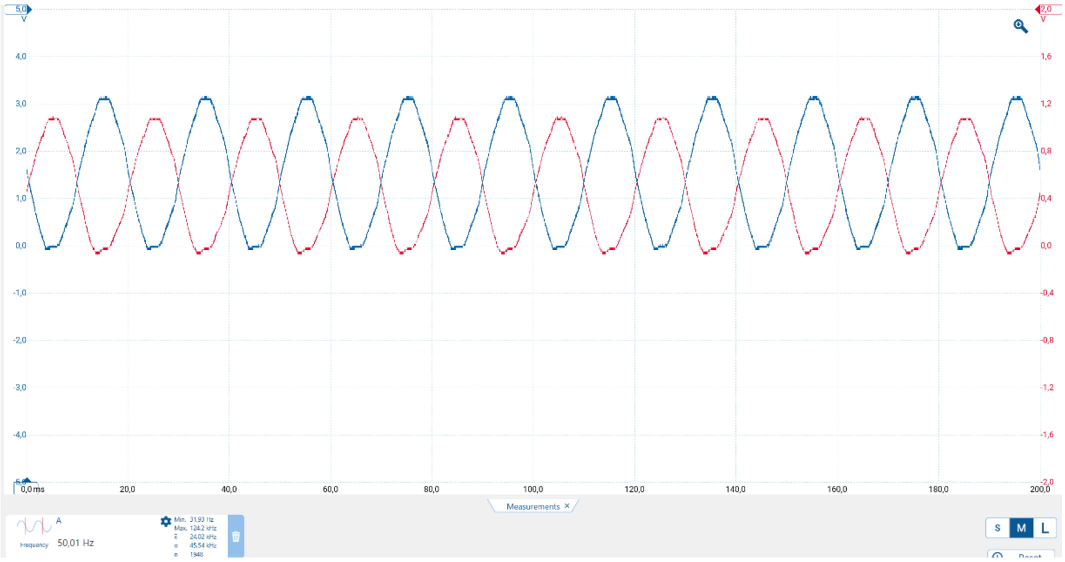


Figure 25. After Level Shifting

In Figure 25, the Voltage signal (Blue Waveform) after level shifting oscillates around a new DC offset, likely close to 2V. This means the waveform has been shifted upwards, ensuring all values are positive.The peak-to-peak voltage remains around 4V, but it is now within a range suitable for the Arduino Mega’s 0-5V input. The frequency remains unchanged at 50 Hz, indicating that the level shifting process does not affect the signal’s frequency content.

The Current Signal (Red Waveform) has also been shifted upwards to oscillate around a new DC offset, likely around 1V. The peak-to-peak current remains around 2V, now fitting within the 0-5V input range of the Arduino. The frequency remains at 50 Hz, similar to the voltage signal, ensuring accurate synchronization and measurement.

The testing procedure using the PicoScope 3000 Series validated the functionality and accuracy of the AC signal measurement both before and after level shifting. The tests confirmed that the level shifter correctly converts the AC signal into a format readable by the Arduino, with the expected 50 Hz frequency in the waveforms. This ensures reliable data acquisition and processing for further analysis and application in IoT environments.

5.3 Implementation of FFT To Arduino Code

This project implemented the Fast Fourier Transform (FFT) to analyze AC signals and transmitted the data to ThingSpeak for further analysis and visualization. This process involved several steps, including setting up hardware and software components, performing FFT computations, and transmitting data.

Implementation Steps:

1) Setup (Figure 26).

Serial Communication: Setting up two serial communication channels at different baud rates allows for robust data exchange.

SIM7000 Initialization: Initializing the SIM7000 module ensures it is ready for data transmission.

Timer1 Configuration: The Timer1 interrupt is set to trigger every 2ms, enabling precise timing for data acquisition.

Interrupt Service Routine (ISR): Attaching the timerISR function to the interrupt allows for handling tasks precisely at each interval.

This configuration enables the Arduino to collect and transmit data efficiently.

```
void setup() {
  Serial.begin(9600);
  mySerial.begin(19200);

  initSIM7000();
  Timer1.initialize(2000);  // Initialize Timer1 to trigger every 2ms
  Timer1.attachInterrupt(timerISR);  // Attach the timerISR function to the interrupt
  Timer1.start();

 delay(1000);
}
```

Figure 26. Setup

2) Analog Signals

   The Arduino reads analog signals from the voltage and current sensors using its analog input pins (A0 and A1) (Figure 27).

```
const uint16_t samples = 256;
const uint16_t samplingFrequency = 500;

#define VOLTAGE_PIN A0
#define CURRENT_PIN A1


Reading takeReading() {

  float voltage = analogRead(VOLTAGE_PIN) * (5.0 / 1023.0);
  float current = analogRead(CURRENT_PIN) * (5.0 / 1023.0);


 return{ voltage, current, millis()};
}
```

Figure 27. Reading Analog Signals

- The analog signals are read using the "analogRead" function and scaled to the appropriate voltage levels. On most Arduino boards, the default reference voltage is set to 5V. This implies that an analog input of 0V corresponds to a digital value of 0, and an analog input of 5V corresponds to a digital value of 1023. By using this reference voltage, the ADC maps the input voltage linearly to the range of digital values. It is also possible to set the reference voltage to other values for more specific applications, allowing for finer resolution over smaller voltage ranges. (Kumar 2016.) The readings are taken at a fixed sampling frequency using a timer interruption to ensure accurate timing. In this case, data is collected every two milliseconds (ms), resulting in a sampling frequency of 500 Hz, and 256 samples are collected for FFT processing.

3) Performing FFT

- FFT is computed separately for the voltage and current signals using the "arduinoFFT.h" library. (Kosme 2023.) This library is essential for converting time-domain signals into their frequency-domain representation.

DC removal:

- The "dcRemoval" (Figure 28) function removes the average value of the signal, which is crucial to eliminate the DC offset and focus on the AC components.

Windowing:

- The "windowing" Figure 28) function applies a Hamming window to the signal.(National Instruments Corp.) Windowing functions reduce spectral leakage, which can distort the signal's frequency representation.

Compute:

- The "compute" (Figure 28) function performs the FFT, transforming the signal from the time domain to the frequency domain.

Complex to Magnitude:

- The "complexToMagnitude" (Figure 28) function converts the complex numbers obtained from the FFT to magnitudes, representing each frequency component's amplitude.

```
FFT.dcRemoval();
FFT.windowing(vReal, samples, FFTWindow::Hamming, FFTDirection::Forward);
FFT.compute(FFTDirection::Forward);
FFT.complexToMagnitude();
```

Figure 28. Performing FFT

4) Data Transmission to ThingSpeak (Figure 29).

Data is transmitted to ThingSpeak using the DFRobot_SIM7000 module. Each reading includes voltage, current, and a timestamp. The data is formatted into an HTTP GET request and sent using the SIM7000 module.

```
void sendDataToThingspeak() {
  int currentIndex = 0;

  while (currentIndex < 256) {
    if (sim7000.openNetwork(DFRobot_SIM7000::eTCP, "api.thingspeak.com", 80) != 0) {
      Serial.println("Failed to open network connection. Retrying...");
      delay(5000);
      continue;
    }

    String url = "GET /update?api_key=" + String(THINGSPEAK_API_KEY) + "&field1=" + String(readings[currentIndex].voltage, 2) + "&field2=" + String(readings[currentIndex].current, 2)
    + "&field3=" + String(readings[currentIndex].timestamp) + " HTTP/1.1\r\nHost: api.thingspeak.com\r\nConnection: close\r\n\r\n";

    Serial.print("Sending data to ThingSpeak: ");
    Serial.println(url);

    if (sim7000.send((char*)url.c_str())) {
      sim7000.closeNetwork();
      currentIndex++;
    } else {
      Serial.println("Failed to send data. Moving to the next data point...");
      delay(5000);
      sim7000.closeNetwork();
      currentIndex++;
    }
  }
```

Figure 29. Data Transmission to ThingSpeak

5.3.1 Detailed Explanation of Key Components

Libraries Used:

- ArduinoFFT: This library performs FFT on the collected analog signal data. It provides functions for DC removal, windowing, FFT computation, and converting complex numbers to magnitudes. (Kosme 2023.)
- DFRobot_SIM7000: This library manages communication with the SIM7000 module, allowing Arduino to send data to the ThingSpeak server over the internet. (DFRobot_SIM7000 2021.)
- RTClib: This library interfaces with the RTC DS3231 module to keep track of time. This is essential for timestamping the readings accurately. (RTClib 2019.)
- TimerOne: This library sets up and manages a timer interrupt, ensuring consistent sampling intervals for reading the analog signals. (TimerOneArduino.)
- Millis function: The "millis" function in Arduino measures the time elapsed since the program started, ensuring precise timing for each sample. (Tutorialspoint.)

FFT Computation Steps:

- DC Removal: This step removes the signal's DC component, which is the average value or offset. Removing the DC component is crucial to focusing on the AC signal's oscillatory nature.
- Windowing: Applying a window function like Hamming reduces spectral leakage because the FFT assumes the signal is periodic. Windowing smooths the discontinuities at the edges of the sampled signal.
- Compute: The FFT algorithm transforms the time-domain signal into its frequency-domain representation, revealing the signal's frequency components.
- Complex to Magnitude: This step converts the FFT's complex output to magnitudes, representing the amplitude of each frequency component.

Output Example:

An example of the output generated by the system after performing FFT and computing key parameters is shown in Figure 30:

```
Voltage Max Frequency = 50.05 Hz
Current Max Frequency = 50.05 Hz
vAngle = 28.536064°
iAngle = -41.514724°
```

Figure 30. Outputs

"vAngle" is a voltage angle, and "iAngle" is a current angle.

Figure 31 shows the voltage and current signals' dominant frequency components and their respective phase angles.

```
Serial.print("Voltage Max Frequency = ");
Serial.print(vPeak, 2);
Serial.println(" Hz");

Serial.print("Current Max Frequency = ");
Serial.print(iPeak, 2);
Serial.println(" Hz");

Serial.print("vAngle = ");
Serial.print(vAngle, 6);
Serial.println("°");

Serial.print("iAngle = ");
Serial.print(iAngle, 6);
Serial.println("°");
```

Figure 31. Serial Output data

In summary, the arrays vReal and vImag store the real and imaginary parts of the signal, respectively (Figure 32).

```
float vReal[samples];
float vImag[samples];
```

Figure 32. vReal and vImag real and imaginary parts

The Arduino FFT instance "FFT" is initialized using the arrays vReal and vImag, the number of samples, and the sampling frequency. Specifically, vReal stores the real parts of the signal, while vImag stores the imaginary parts, both critical for performing the Fourier Transform. The number of samples, defined as samples, indicates how many data points are used in the FFT analysis, and the sampling frequency sets the rate at which these samples are taken, measured in Hertz (Hz). This configuration allows the FFT instance to convert the time-domain signal into its frequency-domain representation, enabling detailed spectral analysis which is vital for real-time monitoring and analysis of AC signals in various IoT applications. (Figure 33).

```
ArduinoFFT<float> FFT = ArduinoFFT<float>(vReal, vImag, samples, samplingFrequency);
```

Figure 33. Initialized with vReal and vImag

This setup allows the FFT instance to perform the FFT on the input signal, transforming it from the time domain to the frequency domain for further analysis. The data is then transmitted to ThingSpeak (Figure 34) for remote monitoring and analysis. This setup is essential for real-time monitoring and analysis of AC signals in IoT applications.

5.3.2 Analysis of Data ThingSpeak Charts

The data obtained from ThingSpeak, as shown in the charts (Figure 41), provides valuable insights into the voltage, current, and time recorded by the Arduino setup. This detailed analysis of the visual data from the charts demonstrates that the system effectively captures and transmits real-time measurements to the ThingSpeak platform. However, sending data to ThingSpeak introduces a significant delay, impacting the overall time efficiency of the monitoring system. Additionally, a desktop version of MATLAB was used to analyze further and visualize the data, enhancing the overall understanding of the system's performance.

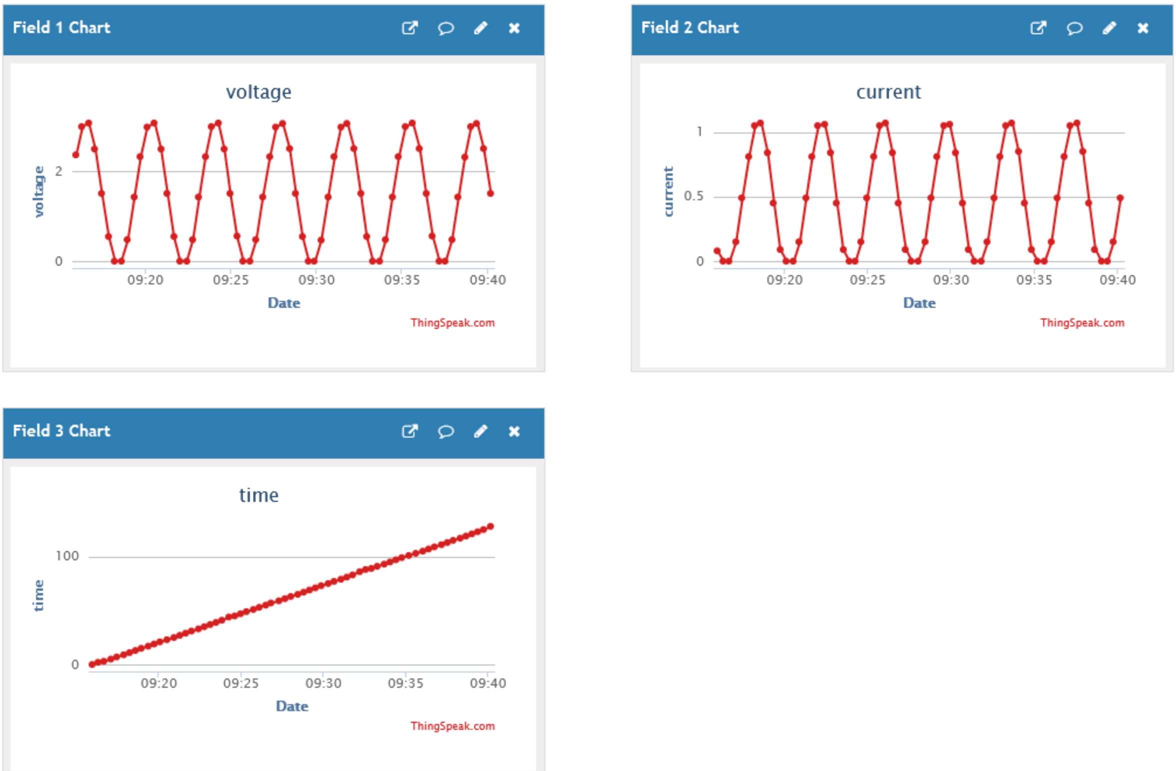


Figure 34. Recorded data

Voltage Chart (Field 1):

- The voltage signal shows a periodic waveform, indicating an alternating current (AC) signal.
- The amplitude of the voltage signal varies between approximately 0 to 3 volts.
- The waveform appears consistent and smooth, suggesting a stable AC source.

Analysis:

- The periodic nature of the voltage indicates that the AC source is functioning correctly.
- The maximum voltage peaks suggest that the system's voltage levels are within expected ranges for the setup.
- The time intervals between peaks can be used to calculate the frequency of the AC signal. Given the regular spacing of peaks, the frequency can be determined by measuring the time difference between successive peaks and taking the inverse of this value.

Current Chart (Field 2):

- The current signal also shows a periodic waveform, which is typical for an AC current
- The amplitude of the current signal varies between approximately 0 to 1 ampere.
- Similar to the voltage chart, the current waveform is consistent and periodic.

Analysis:

- The periodic current waveform confirms the proper functioning of the current measurement setup.
- The amplitude of the current peaks indicates the system's current draw, which can be correlated with the load connected to the AC source.
- One can analyze the phase relationship between the current waveform with the voltage waveform, which is important for calculating power factor and other electrical parameters.

Time Chart (Field 3)

- The time chart shows a linear increase in time values.
- This suggests that the data points are being recorded at regular intervals.

Analysis:

- The consistent increment in time confirms that the data logging functions correctly.
- The slope of the time chart can be used to verify the sampling rate of the data acquisition system. Since the data points are evenly spaced, this regular interval ensures accurate time correlation with the voltage and current readings.

5.4 Data Analysis in MATLAB

In this project, MATLAB was utilized to process and analyze the data collected from the ThingSpeak platform. MATLAB is a powerful computational tool widely used for data analysis, visualization, and algorithm development. By leveraging MATLAB's extensive library of functions and its ease of integration with external data sources, our Arduino-based system achieved a detailed analysis of the voltage and current data. The Arduino code shows (Figure 35) different angles, but MATLAB is

consistent with its values. This consistency in MATLAB is due to its robust numerical computation capabilities and advanced signal processing functions, which ensure accurate and reliable phase angle calculations. Consequently, MATLAB is relied upon for its precision and reliability in analyzing electrical signals, providing a more accurate representation of the system's behavior than the Arduino code.

The process began with setting up "readChannelID" and "readApiKey"(Figure 35) from ThingSpeak and, in another case, importing the CSV files(Figure 36) containing the voltage, current, and time data from ThingSpeak.

```
readChannelID = [redacted];
readAPIKey = [redacted]';
```

Figure 35. Read Channel ID and API Key

```
% Detect import options from the CSV file
url='dataHeaterV2CSV.csv'
opts = detectImportOptions(url);
opts.VariableTypes = {'string','string','double'};
```

Figure 36. The URL of the CSV file

The code extracts voltage, current, and datetime data from ThingSpeak fields (Figure 37) or the table(Figure 38) and defines start and end times for each data segment.

```
fields = [1, 2, 3];
```

Figure 37. Fields

```
% Extract data from the table
voltage_data = T{:, 1};  % Voltage data
current_data = T{:, 2};  % Current data
datetime_data = T{:, 3};  % Datetime data
```

Figure 38. Data from the table

The datetime, voltage, and current data are divided into segments for detailed analysis. (Figure 39)

```
t1 = datetime_data(1:256) / 1000;
t2 = datetime_data(257:512) / 1000;
t3 = datetime_data(513:768) / 1000;
t4 = datetime_data(769:1024) / 1000;
```

Figure 39. Time Vectors

The code performs FFT on the voltage and current data to analyze their frequency components, where the sampling frequency is 500, and DC components are removed. (Figure 40 and 41)

```
fftV_1 = fft(v1 - mean(v1));
fftC_1 = fft(v1 - mean(v1));
```

Figure 40. Example of performing FFT on Voltage and Current

```
fftshiftV_1 = fftshift(fftV_1);
fftshiftC_1 = fftshift(fftC_1);
```

Figure 41. Example Of Shifting FFT Data

The FFT transforms the time-domain signals into the frequency domain, and "fftshift" centers the zero-frequency component.

The code calculates the frequencies and amplitudes for plotting FFT results.

LyV_1 and LyC_1(Figure 42) calculate the length of the array fftV_1a and fft_C and store it in the variable lyV_1 and LyC_1. The length function in MATLAB returns the number of elements in the largest dimension of the array.

```
lyV_1 = length(fftV_1);
lyC_1 = length(fftC_1);
```

Figure 42. Length Of the Arrays

Fv1 and fc1(Figure 43) lines construct an array fV1 representing the frequency bins for the FFT data.

```
fV1 = (-lyV_1 / 2:lyV_1 / 2 - 1) / lyV_1 * fs;
fC1 = (-lyC_1 / 2:lyC_1 / 2 - 1) / lyC_1 * fs;
```

Figure 43. Frequencies bins

The code identifies the maximum FFT amplitudes for voltage and current. This line (Figure 44) of code finds the maximum value in a specific portion of the FFT result (fftshiftV_1) and its corresponding index.

```
[maxValue_V1, index_V1] = max(abs(fftshiftV_1(128:end)));
[maxValue_C1, index_C1] = max(abs(fftshiftC_1(128:end)));
```

Figure 44. Max Values and Indexes

Index _V1 and index_C1 lines (Figure 45) adjust the index of the maximum value found in the subset of the FFT-shifted data (fftshiftV_1).

```
index_V1 = index_V1 + 127;
index_C1 = index_C1 + 127;
```

Figure 45. Adjusted indexes

Strings fMaxValue_V1 and fMaxVakue_C1 calculate frequencies and angles for peaks. (Figure 46)

```
fMaxValue_V1 = fV1(index_V1);
fMaxValue_C1 = fC1(index_C1);
```

Figure 46. Peaks of values

Finally, phase differences between voltage and current values were calculated. (Figure 47)

```
% Calculate phase differences
phaseDiff1 = angleV1 - angleC1
phaseDiff2 = angleV2 - angleC2
phaseDiff3 = angleV3 - angleC3
phaseDiff4 = angleV4 - angleC4
```

Figure 47. Phase differences

Figures 48 and 49 show the amplitude spectra of the voltage and current data segments and Phase Angles. Each subplot indicates the magnitude of different frequency components in the voltage and current signal. The dominant frequency component is at 50 Hz for all four segments. Symmetrical peaks around 50 Hz are observed, typical for signals sampled from power systems operating at 50 Hz.

The phase angles are calculated after applying a threshold to the FFT-shifted data to remove noise and small magnitude components. The significant phase components appear to be centered around zero degrees.

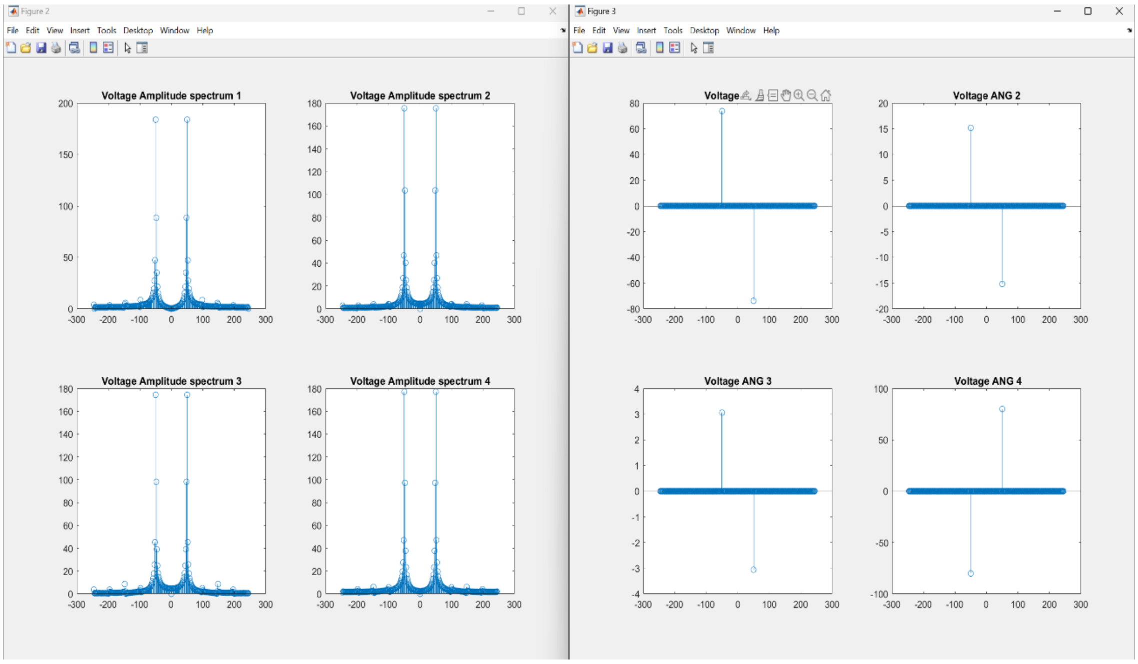


Figure 48. Voltage Amplitude Spectrum and Angle

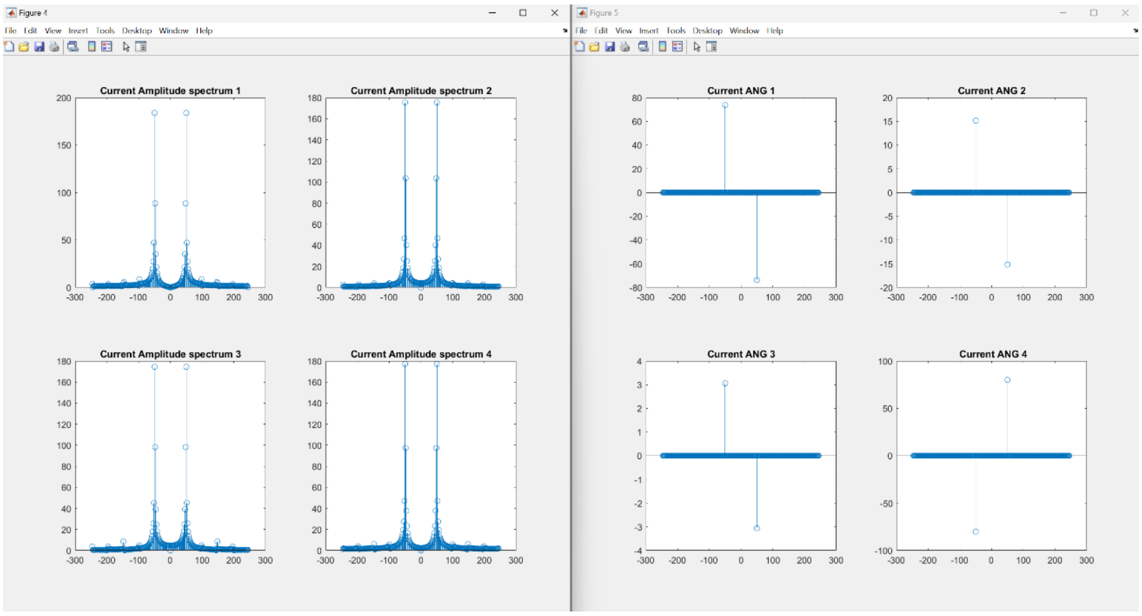


Figure 49. Current Amplitude Spectrum and Angle

The phase differences (Figure 50) between the voltage and current signals (phaseDiff1, phaseDiff2, phaseDiff3, and phaseDiff4) are all zero. This indicates that there is no phase shift between the voltage and current signals for all four segments. A zero-phase difference implies that the voltage and current signals are in phase.

```
{'Phase difference 1'}    {[0]}

{'Phase difference 2'}    {[0]}

{'Phase difference 3'}    {[0]}

{'Phase difference 4'}    {[0]}
```

Figure 50. Phase Differences

The dominant frequency in all spectra is 50 Hz. This aligns with the typical power system frequency, suggesting that the measured signals are from a power system operating at 50 Hz.

Dominant Frequency: The voltage and current signals have a dominant frequency component at 50 Hz, which is typical for power systems in many parts of the world.

Phase Relationship: The voltage and current signals are in phase, as indicated by the zero-phase difference for all four segments. This is often seen in purely resistive loads where the current and voltage waveforms align perfectly.

FFT Analysis: The FFT amplitude spectra show clear peaks at the fundamental frequency (50 Hz) and its harmonics. The phase spectra are centered around zero degrees after filtering out the low-amplitude noise components.

As a result, the analysis suggests a stable and consistent power system signal without significant phase shifts between voltage and current.

Power Calculations:

A current sensor was connected to the Arduino to measure the current, and the output was scaled using a defined "currentScalingFactor" (Figure 51). The current scaling factor was determined experimentally by comparing the sensor's output with the actual current measured using a multimeter. Specifically, was measured the actual current using a multimeter and compared it to the value reported by the current sensor.

- Measured Actual Current**:** 0.82A using a multimeter.
- Sensor Reported Current: 0.68A from the current sensor.
- Calculate Scaling Factor:

$$currentScalingFactor = \frac{Actaul\ Current}{Sensor\ Reported\ Current}$$

Defining the voltage scaling factor (Figure 51) was essential to ensure accurate voltage measurements from the sensor. An experimental procedure involving a known reference voltage determined the voltage scaling factor.

The output from the voltage sensor was read using the Arduino. The sensor provided a scaled-down voltage output of 2.02V corresponding to the 230V applied.

$$voltageScalingFactor = \frac{Actaul\ Current}{Sensor\ Reported\ Output}$$

```
voltageScalingFactor = 113.8;
currentScalingFactor = 1.21;
```

Figure 51. Scaling factors

By applying this voltage scaling factor in MATLAB, the RMS voltage, and current values (Figure 52) were accurately adjusted, ensuring precise power calculations and analysis of the system's performance.

```
% Adjust RMS values for voltage and current
Irms1 = rms(c1) * currentScalingFactor;
Vrms1 = rms(v1) * voltageScalingFactor;
```

Figure 52. RMS values

The real power (Figures 54 and 57) for each segment was calculated using the formula:

$$P = Vrms * Irms * \cos(\theta)$$

where $\cos(\theta)$represents the power factor. Assuming the load is purely resistive, the power factor $\cos(\theta)$is 1.

In Figure 53, the values of angles are converted to radians.

```
% Correct phase angles (in radians)
phi1 = deg2rad(phaseDiff1);
```

Figure 53. Convert to radians

```
% Calculate the real power for each segment
P1 = Vrms1 * Irms1 * cos(phi1);
```

Figure 54. Real Power

The apparent power, representing the total power in the circuit, was calculated in Figures 55 and 58.

```
% Calculate the apparent power for each segment
S1 = Vrms1 * Irms1;
```

Figure 55. Apparent Power

Reactive power (Figures 56 and 59), which is the power that oscillates between source and load, was calculated using:

```
% Calculate the reactive power for each segment
Q1 = sqrt(S1^2 - P1^2);
```

Figure 56. Reactive Power

```
Real Power for Segment 1:
  187.8117

Real Power for Segment 2:
  189.5660

Real Power for Segment 3:
  183.3614

Real Power for Segment 4:
  186.4679
```

Figure 57. Values of Real Power

```
Apparent Power for Segment 1:
  187.8117

Apparent Power for Segment 2:
  189.5660

Apparent Power for Segment 3:
  183.3614

Apparent Power for Segment 4:
  186.4679
```

Figure 58. Values of Apparent Power

```
Reactive Power for Segment 1:
     0

Reactive Power for Segment 2:
     0

Reactive Power for Segment 3:
     0

Reactive Power for Segment 4:
     0
```

Figure 59. Values of Reactive Power

The comprehensive analysis of the voltage and current data collected from the Arduino-based system, processed using MATLAB, has yielded insightful conclusions about the load's power characteristics. By calculating and interpreting real power, apparent power, and reactive power, the study provides a detailed understanding of the system's behavior, particularly in the context of a purely resistive load like a heater.

Real power, also known as active power, is consumed by the load to perform useful work, such as generating heat in a heater. The measured real power values for the four segments are 187.8117 W, 189.5660 W, 183.3614 W, and 186.4679 W, respectively. These values show a stable and consistent power consumption across different segments, with only slight variations. Such stability confirms the heater's reliable operation, consistently converting electrical energy into thermal energy. The minor fluctuations in power readings are likely due to small changes in load conditions or measurement inaccuracies, which are within an acceptable range.

Apparent power represents the total power supplied to the circuit, encompassing both real and reactive components. The measured apparent power values, identical to the real power values for each segment (187.8117 VA, 189.5660 VA, 183.3614 VA, and 186.4679 VA), indicate that the power factor is 1. This identity between real and apparent power signifies that the load is purely resistive. In purely resistive loads, the voltage and current waveforms are in phase, resulting in no phase difference and a power factor of 1. This means all the electrical power supplied to the heater is effectively converted into useful work, with no wastage in the form of reactive power.

Reactive power, which does not contribute to useful work, is associated with inductive and capacitive loads with a phase difference between voltage and current. The measured reactive power values for all segments are 0 VAR, confirming the absence of reactive components in the load. These zero reactive powers indicate that the heater operates purely as a resistive load, further validating the system's effectiveness in converting electrical energy entirely into heat. The lack of reactive power ensures the system is highly efficient, as no power is lost in maintaining electric and magnetic fields.

The consistent values for real and apparent power, combined with the zero reactive power, suggest that the measurements are reliable and accurate. The results confirm the purely resistive nature of the load, which is expected for a heater. The high-power factor of 1 indicates that the system operates at maximum efficiency, with all supplied power being used for useful work. This efficiency is crucial for energy conservation and optimal performance applications.

6 DISCUSSION

The spectral analysis conducted in this study provided crucial insights into the behavior of electrical signals. Using Fast Fourier Transform (FFT) was successfully decomposed time-domain signals into their frequency-domain representations. This allowed us to identify the voltage and current signals' harmonic content and phase information, essential for detecting non-idealities or distortions.

The FFT analysis effectively broke down the electrical signals into their constituent frequencies, providing detailed amplitude and phase information.

Calculating the phase difference between voltage and current using FFT-derived phase information proved highly accurate. This is critical for determining the power factor and distinguishing between real and reactive power, especially confirming the purely resistive nature of the load.

Comparing Arduino and MATLAB highlighted their complementary strengths. Arduino excels in real-time data acquisition and provides a flexible, low-cost platform. On the other hand, MATLAB offers advanced data processing capabilities, enabling precise FFT analysis, phase difference calculations, and robust visualizations.

In the future, comparing with other systems operating under different conditions can identify best practices and areas for improvement. Using advanced power quality analysis techniques such as harmonic analysis and transient analysis can reveal additional information about system performance and potential problems.

In conclusion, the segment analysis of voltage and current data of the electrical system demonstrated a stable, efficient, and high-quality power system. These results are important for optimizing system design and control, ensuring reliable and cost-effective operation.

7 REFERENCES

Artificial intelligence has been used in the work as follows:
ChatGPT 2023. OpenAI. GPT-3.5. Accessed for language check, May 2024.

https://chat.openai.com

The above information is formed from the following details:
Service year. Company. Version. How utilized, when. Web address.

Ahmad, S. 2022. Active Power, Reactive Power, Apparent Power, and the Role of Power Factor. Online publication. https://control.com/technical-articles/active-power-reactive-power-apparent-power-and-the-role-of-power-factor/. Accessed 10.04.2024.

Airbyte. 2024. Data Preprocessing: What it is, Steps, & Methods Involved. Online publication. https://airbyte.com/data-engineering-resources/data-preprocessing. Accessed 24.04.2024.

Airtel Business. 2022. Understanding NarrowBand IoT (NB-IoT) and its Use Cases. Online publication. https://www.airtel.in/b2b/insights/blogs/narrowband-iot-and-its-use-cases. Accessed 24.05.2024.

Akram, M. W. 2021. IOT — Basic Electronics. Online publication. https://medium.com/@MohamedWasim001/iot-basic-electronics-c73484186081. Accessed 10.04.2024.

Aneindia. DS3231 AT24C32 IIC Precision RTC Real Time Clock Memory Module. Online publication. https://www.aneindia.com/product/module/ds3231-at24c32-iic-precision-rtc-real-time-clock-memory-module/. Accessed 07.03.2024.

Antonov, A. 2016. Ohm’s Law Refutes Current Version of the Special Theory of Relativity. Online publication. https://www.scirp.org/journal/paperinformation?paperid=72776. Accessed 01.05.2024.

Arduino. Arduino Mega 2560 Rev3. Online publication. https://store.arduino.cc/products/arduino-mega-2560-rev3. Accessed 07.03.2024.

AVSystem. 2020. What is Narrowband IoT? Online publication. https://www.avsystem.com/blog/iot/narrowband-iot/. Accessed 06.03.2024.

Blateyron, F. 2023. What is spectral analysis? Online publication. https://www.digitalsurf.com/blog/what-is-spectral-analysis/. Accessed 14.05.2024.

Chen, M., Mao, S., & Liu, Y. 2014. Big data: A survey. Mobile networks and applications. Online publication https://link.springer.com/article/10.1007/s11036-013-0489-0. 171-209. Accessed 23.03.2024.

Chromausa ATE INC. PROGRAMMABLE AC POWER SOURCE. Online publication. https://www.chromausa.com/pdf/61600%20Low%20Power%20Programmable%20AC%20Source%20052023.pdf. Accessed 04.04.2024.

Chromausa. Low Power Programmable AC Source. Online publication.https://www.chromausa.com/product/low-power-programmable-ac-source-61600/. Accessed 04.04.2024.

David, S. IoT Data Management platform. Online publication. https://successinsightsindia.com/iot-data-management-platform/. Accessed 23.03.2024.

David, S. 2021. IoT Data Management Platform: Benefits and Its Best Practice, The Future Of IoT in Enterprise. Online publication. https://industryoutreachmagazine.com/iot-data-management-platform/. Accessed 23.03.2024.

DeTore, C. 2023. IoT Analytics Applications and Use Cases. Online publication. https://www.ptc.com/en/blogs/iiot/iot-analytics-applications-and-use-cases. Accessed 03.03.2024.

DFRobot. 2018. SIM7000E Arduino NB-IoT / LTE / GNSS / GPRS / GPS Expansion Shield (Europe). Online publication. https://www.dfrobot.com/product-1732.html. Accessed 07.01.2024.

DFRobot_SIM7000. 2021. DFRobot_SIM7000. Online publication.https://github.com/DFRobot/DFRobot_SIM7000. Accessed 07.01.2024.

Digi International. 2024. What Is NB-IoT? Narrowband IoT Applications. Online publication. https://www.digi.com/blog/post/nb-iot. Accessed 03.03.2024.

Docs.Arduino. Overview of the Arduino IDE 1. Online publication. https://docs.arduino.cc/software/ide-v1/tutorials/Environment/. Accessed 23.04.2024.

Dugan, R. C., McGranaghan, M. F., & Beaty, H. W. 1996. Electrical power systems quality. 36-37, 171-177, 328-331.

Electrical Technology. 2013. Active, Reactive, Apparent and Complex Power. Online publication.https://www.electricaltechnology.org/2013/07/active-reactive-apparent-and-complex.html. Accessed 12.05.2024.

ElectronicsTutorials. Power in AC Circuits. Online publication. https://www.electronics-tutorials.ws/accircuits/power-in-ac-circuits.html. Accessed 12.05.2024.

Elfadistrelec. PICOSCOPE 3404D. Online publication.https://www.elfadistrelec.fi/fi/pc-oskilloskooppi-4x-70mhz-250msps-pico-technology-picoscope-3404d/p/30018071?cq_src=google_ads&cq_cmp=19910015263&cq_con=&cq_term=&cq_med=pla&cq_plac=&cq_net=x&cq_pos=&cq_plt=gp&gad_source=1&gclid=CjwKCAjwl4yyBhAgEiwADSEje. Accessed 03.04.2024.

Fiberbit Technology CO LTD. 2013. Differences between analog and digital signaling. Online publication.https://fiberbit.com.tw/differences-between-analog-and-digital-signaling/. Accessed 23.04.2024.

G2. 2024.Best IoT Analytics Platforms. Online publication. https://www.g2.com/categories/iot-analytics-platforms. Accessed 03.03.2024.

Gardner, J. W., Varadan, V. K., & Awadelkarim, O. O. 2001. Micro Sensors, MEMS, and Smart Devices. John Wiley & Sons, 9-10.

Gopalakrishnan, J. 2023. IoT Analytics and Reporting – A Comprehensive Guide. Online publication. https://www.knowi.com/blog/iot-analytics-and-reporting-a-comprehensive-guide/. Accessed 23.04.2024.

Goyal, P., Sahoo, A. K., & Sharma, T. K. 2021. Internet of things: Architecture and enabling technologies. Online publication. https://www.sciencedirect.com/science/article/abs/pii/S2214785320333253. Accessed 07.05.2024.

GSMA. Narrowband – Internet of Things (NB-IoT). Online publication.https://www.gsma.com/solutions-and-impact/technologies/internet-of-things/narrow-band-internet-of-things-nb-iot/. Accessed 03.03.2024.

Gülen, K. 2023. Exploring the dynamic fusion of AI and the IoT. Online publication.https://dataconomy.com/2023/05/25/artificial-intelligence-in-internet-of-things/. Accessed 02.03.2024.

Gülen, K. 2022. Unlocking the full potential of connected devices with IoT analytics. Online publication. https://dataconomy.com/2022/12/23/iot-analytics-architecture-use-cases-jobs/. Accessed 02.03.2024.

Hakme, A., Gupta, R., & Kalaspurkar, U. 2022. 7 patterns for IoT data ingestion and visualization- How to decide what works best for your use case. Online publication. https://aws.amazon.com/blogs/iot/7-patterns-for-iot-data-ingestion-and-visualization-how-to-decide-what-works-best-for-your-use-case/. Accessed 03.02.2024.

Huang, C., Chen, Q., Lin, Z., Su, W., Zhang, Z., & Liu, S. 2024. The Economic Impact and Application Challenges of IoT Technology in Smart Home and Infrastructure. Online publication. https://www.researchgate.net/publication/378360966_The_Economic_Impact_and_Application_Challenges_of_IoT_Technology_in_Smart_Home_and_Infrastructure. Accessed 03.03.2024.

IDE. Downloads. Arduino. Online publication. https://www.arduino.cc/. Accessed 07.02.2024.

Insightsoftware. 2023. Comparing Descriptive, Predictive, Prescriptive, and Diagnostic Analytics. Online publication. https://insightsoftware.com/comparing-descriptive-predictive-prescriptive-diagnostic-analytics/. Accessed 26.03.2024.

IoT.Business.News. 2023. 5 Challenges of Data Analytics within IoT Systems and Tips to Solve Them. Online publication. https://iotbusinessnews.com/2023/10/12/08780-5-challenges-of-data-analytics-within-iot-systems-and-tips-to-solve-them/. Accessed 12.05.2024.

Javatpoint. Ohm's Law Definition. Online publication. https://www.javatpoint.com/ohms-law-definition. Accessed 09.05.2024.

Kosme. 2023. ArduinoFFT. Online publication. https://github.com/kosme/arduinoFFT. Accessed 07.02.2024.

Kumar, C. 2016. Why is 5v designated to 1023 in Arduino program? Online publication. https://www.quora.com/Why-is-5v-designated-to-1023-in-arduino-program. Accessed 04.03.2024.

Li, S. 2020. The 5G/6G enabled future internet of things. Online publication.https://www.researchgate.net/publication/345094061_The_5G6G_enabled_Future_Internet_of_Things. Accessed 12.05.2024.

LibreTexts. Power in an AC Circuit. Online publication. https://phys.libretexts.org/Bookshelves/University_Physics/University_Physics_(OpenStax)/Book%3A_University_Physics_II_-_Thermodynamics_Electricity_and_Magnetism_(OpenStax)/15%3A_Alternating-Current_Circuits/15.05%3A_Power_in_an_AC_Circuit. Accessed 11.05.2024.

Lueth, K. L. 2014. Why the Internet of Things is called Internet of Things: Definition, history, disambiguation. Online publication. https://iot-analytics.com/internet-of-things-definition/. Accessed 17.03.2024.

Matellio. 2023. IoT Data Analytics: Benefits, Use Cases, and Implementation. Online publication. https://www.matellio.com/blog/iot-data-analytics-benefits-use-cases-implementation/. Accessed 07.02.2024.

MathWorks. 2024. Basic Spectral Analysis. Online publication. https://se.mathworks.com/help/matlab/math/basic-spectral-analysis.html. Accessed 12.05.2024.

MathWorks.2024. Calling MATLAB from Another Language. Online publication. https://se.mathworks.com/products/matlab/matlab-and-other-programming-languages.html. Accessed 03.03.2024.

MathWorks.2024. ThingSpeak for IoT Projects. Online publication. https://se.mathworks.com/products/thingspeak.html. Accessed 03.02.2024.

MATLAB. Math. Graphics. Programming. Online publication. https://se.mathworks.com/products/matlab.html. Accessed 03.03.2024.

MATLABIDE. Try MATLAB, Simulink, and More. Online publication. https://se.mathworks.com/campaigns/products/trials.html?gclid=CjwKCAjwx-CyBhAqEiwAeOcTdUYSoPaj6fTCb0RFQrAPT9lJAL65iEZRzwuYMQFragphPYxXIC433hoC2_0QAvD_BwE&ef_id=CjwKCAjwx-CyBhAqEiwAeOcTdUYSoPaj6fTCb0RFQrAPT9lJAL65iEZRzwuYMQFragphPYxXIC433hoC2_0QAvD_BwE:G:s. Accessed 03.03.2024.

Mazlan, A. 2023. Kevin Ashton: Pioneering the Way for RFID Technology and the IoT. Online publication. https://iotworld.co/2023/03/kevin-ashton-pioneering-the-way-for-rfid-technology-and-the-iot/. Accessed 16.05.2024.

Misra, S., Maheswaran, M., & Hashmi, S. 2017. Security Challenges and Approaches in Internet of Things. Online publication. https://download.e-bookshelf.de/download/0007/9138/73/L-G-0007913873-0015529894.pdf. Accessed 12.05.2024.

Mohamed, I. 2023. 3GPP Release 17: Understanding NB-IoT over NTN. Online publication. https://5ghub.us/3gpp-release-17-understanding-nb-iot-over-ntn/. Accessed 07.02.2024.

National Instruments Corp. Understanding FFTs and Windowing. Online publication. https://download.ni.com/evaluation/pxi/Understanding%20FFTs%20and%20Windowing.pdf. Accessed 14.04.2024.

Percival, D. B., & Walden, A. T. Introduction to Spectral Analysis. Online publication. https://assets.cambridge.org/97811070/28142/excerpt/9781107028142_excerpt.pdf#:~:text=URL%3A%20https%3A%2F%2Fassets.cambridge.org%2F97811070%2F28142%2Fexcerpt%2F9781107028142_excerpt.pdf%0AVisible%3A%200%25%20. Accessed 11.04.2024.

PicoTechnology. PicoScope® 3000 Series. Online publication. https://www.picotech.com/oscilloscope/3000/usb3-oscilloscope-logic-analyzer. Accessed 03.04.2024.

Podulka, P. 2022. Selection of Methods of Surface Texture Characterisation for Reduction of the Frequency-Based Errors in the Measurement and Data Analysis Processes. Online publication. https://www.mdpi.com/1424-8220/22/3/791. Accessed 21.04.2024.

Proakis, J. G., & Manolakis, D. G. 1996. Digital signal processing: principles, algorithms, and applications. New Jersey: Prentice Hall, 232-248.

Roman, R., Zhou, J., & Lopez, J. 2013. On the features and challenges of security and privacy in distributed internet of things. Online publication. https://www.sciencedirect.com/science/article/pii/S1389128613000054?casa_token=CFRGIZv_gW4AAAAA:uv_kR9grjqJcY6sRKF24h9x1hTjS3VsSigL3Uh7L4K2R0ywnkGA1giXxRpHNve41qEVoUM5duQ. 2266-2279. Accessed 08.02.2024.

Rome, P. 2023. Volts, Currents, and the Basic Concepts of Electricity. Online publication. https://dewesoft.com/blog/volts-and-currents-explained. Accessed 11.05.2024.

Rsdelivers. SI Series. Online publication. https://fi.rsdelivers.com/product/testec/tt-si-9002/testec-si-series-tt-si-9002-oscilloscope-probe-120/2241225?cm_mmc=FI-PLA-DS3A-_-google-_-CSS_FI_EN_PMAX-Catch-All-_--_-&matchtype=&&s_kwcid=AL!14853!3!!!!x!!&gad_source=1&gclid=CjwKCAjwx-CyBhAqEiwAeOcTdR. Accessed 22.05.2024.

RTClib. 2019. RTClib. Online publication. https://github.com/adafruit/RTClib. Accessed 03.02.2024.

Sylvest, J. 2023. Understanding narrowband IoT (NB-IoT): benefits, applications, and limitations. Online publication. https://akenza.io/blog/understanding-nb-iot. Accessed 11.01.2024.

Technology, Pico. Download software and manuals for oscilloscopes and data loggers. Online publication. https://www.picotech.com/downloads. Accessed 03.04.2024.

Thefouriertransform. 2022. Fourier Transforms. Online publication. https://www.thefouriertransform.com/. Accessed 12.03.2024.

ThingSpeak. 2024. ThingSpeak for IoT Projects. Online publication. https://thingspeak.com/. Accessed 03.03.2024.

Thinkbiganalytics. ThingSpeak. Online publication. https://thinkbiganalytics.com/thingspeak. Accessed 03.03.2024.

Tiepie. Oscilloscope probe. Online publication. https://www.tiepie.com/en/usb-oscilloscope/accessories/differential-probe-25mhz-x20-x200. Accessed 22.05.2024.

TimerOneArduino. TimerOne. Online publication. https://www.arduino.cc/reference/en/libraries/timerone/. Accessed 03.03.2024.

Tme. 2024. FLUKE I310SFLUKE. Online publication. https://www.tme.eu/fi/details/flk-i310s/pihti-virtamittarit-ja-ac-dc-ilmaisimet/fluke/fluke-i310s/?brutto=1&currency=EUR&gad_source=1&gclid=CjwKCAjwx-CyBhAqEiwAeOcTda1OmPw09YkBe4lB-k582Y0XRg5WYJ4hHXHOmdgsODsRs3gbv2q_kxoCRXYQAvD_BwE. Accessed 22.05.2024.

Ttid. Chroma 61500 Series. Online publication. https://www.ttid.co.uk/ac-power-supplies/chroma/61500-series. Accessed 04.04.2024.

Tutorialspoint. Arduino - millis () function. Online publication. https://www.tutorialspoint.com/arduino/arduino_millis_function.htm. Accessed 05.04.2024.

Valero, A. 2023. Edge Computing: Empowering Real-Time Data Processing and Analysis. Online publication. https://www.suse.com/c/edge-computing-empowering-real-time-data-processing-and-analysis/. Accessed 23.05.2024.

Velez, F. J., Monteiro, E., & Almeida, J. C. 2019. Low-power wide-area networks for sustainable IoT. Online publication. https://ieeexplore.ieee.org/abstract/document/8626078/authors#authors. Accessed 23.04.2024.

Vos, G. 2022. What is Narrowband IoT (NB-IoT)? Online publication. https://blog.sierrawireless.com/what-is-nb-iot. Accessed 06.02.2024.

Worldstandards. Finland – Power plug, socket & mains voltage in Finland. Online publication. https://www.worldstandards.eu/electricity/plug-voltage-by-country/finland/. Accessed 15.04.2024.