CALIBRATION OF EMULATION CARD FOR PLC

THUNWALAI THONGKONG

MASTER OF ENGINEERING
IN
COMPUTER ENGINEERING

SCHOOL OF APPLIED DIGITAL TECHNOLOGY
MAE FAH LUANG UNIVERSITY
2024

CALIBRATION OF EMULATION CARD FOR PLC

THUNWALAI THONGKONG

THIS THESIS IS A PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF ENGINEERING IN COMPUTER ENGINEERING

SCHOOL OF APPLIED DIGITAL TECHNOLOGY
MAE FAH LUANG UNIVERSITY

2024

THESIS APPROVAL

MAE FAH LUANG UNIVERSITY

FOR

MASTER OF ENGINEERING IN COMPUTER ENGINEERING

Thesis Title: Calibration of Emulation Card for PLC

Author: Thunwalai Thongkong

Examination Committee:

Associate Professor Punnarumol Temdee, Ph. D.	Chairperson
Associate Professor Roungsan Chaisricharoen, Ph. D.	Member
Assistant Professor Group Captain Thongchai Yooyativong, Ph. D.	Member
Professor Hamed Yahoui, Ph. D.	Member
Associate Professor Rawid Banchuin, Ph. D.	Member

Advisors:

....Advisor
(Associate Professor Roungsan Chaisricharoen, Ph. D.)

....Co-Advisor
(Assistant Professor Group Captain Thongchai Yooyativong, Ph. D.)

....Co-Advisor
(Professor Hamed Yahoui, Ph. D.)

Dean:

....
(Assistant Professor Nacha Chondamrongkul, Ph. D.)

ACKNOWLEDGEMENTS

I would like to express my profound gratitude to Associate Professor Dr. Roungsan Chaisricharoen and my co-advisor, Assistant Professor Group Captain Dr. Thongchai Yooyativong, for their invaluable patience, insightful feedback, and unwavering support throughout the development of this thesis. Their guidance and suggestions have been instrumental in shaping the direction and quality of this research.

I sincerely appreciate Professor Dr. Hamed Yahoui, who has a generous share of knowledge and expertise, which has significantly contributed to the successful completion of this study. His support has been crucial in navigating the complexities of this research.

I sincerely thank my classmates and committee members for their encouragement and constructive input. Additionally, I am grateful for the financial support from the university's 15,000 THB thesis grant, which facilitated the research process. Thanks also go to the university's librarians, research assistants, and study participants, whose assistance and cooperation have been essential for the successful execution of this project.

Thank you all for your invaluable contributions and support.

Thunwalai Thongkong

Thesis Title Calibration of Emulation Card for PLC

Author Thunwalai Thongkong

Degree Master of Engineering (Computer Engineering)

Advisor Associate Professor Roungsan Chaisricharoen, Ph. D.

Co-Advisor Assistant Professor Group Captain Thongchai Yooyativong, Ph. D.

Professor Hamed Yahoui, Ph. D.

ABSTRACT

Industrial automation systems often face accuracy issues due to signal distortions and calibration errors, which can lead to significant operational disruptions. This research introduces an automated calibration framework for programmable logic controllers (PLCs) to address these challenges. The system dynamically adjusts input and output values in real time, reducing errors and improving signal consistency. The methodology focuses on the interaction between Siemens SIMATIC S7-1500 PLCs and General Instrument Control System (GICS) emulation cards within a hardware-in-the-loop system. By integrating real-time feedback and dynamic adjustment techniques, the framework continuously monitors and corrects the PLC's input-output relationships with the GICS card, specifically addressing signal offset issues during analog-to-digital and digital-to-analog conversions. Validation through TIA Portal software demonstrates significant improvements in data accuracy, reliability, and system stability, leading to reduced operational costs. The proposed solution enhances overall system performance, ensuring more stable operations and minimizing risks related to calibration errors in industrial automation.

Keywords: Programmable Logic Controllers (PLCs), General Instrument Control System Emulation Card (GICS), SIMATIC S7-1500

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Research Problem

Programmable Logic Controllers (PLCs) are vital components in modern industrial automation. Developed in 1968 by Dick Morley and his team at Bedford Associates, PLCs replaced traditional, complex relay-based control systems (Open University, 2018). They provide a more flexible and programmable approach to automating industrial processes. PLCs allow users to design and implement control logic using software, making them an essential part of various systems, including manufacturing, energy management, and water treatment (Toronto Metropolitan University, 2021). These systems process input signals from sensors and devices, execute control algorithms, and manage output actuators based on the programmed logic. Nowadays, PLCs go beyond basic tasks like counting, timing, and arithmetic calculations. They can communicate with external devices, such as GICS (General Instrument Control System) emulation cards. These cards simulate analog inputs and outputs by transmitting a 10-bit analog input signal and a 12-bit analog output signal. The PLC receives the 10-bit input, converts it to a 16-bit digital signal for processing, and sends the processed output back to the GICS card, which receives the 12-bit analog signal (Cleveland State University, 2016). However, this data exchange process is prone to issues like internal offsets and noise interference, which can distort the signals and affect accuracy. Accuracy is critical in PLC systems because even small errors can lead to incorrect control decisions, system instability, and increased operational costs. Common sources of inaccuracy include signal offsets, noise interference, and resolution errors during Analog-to-Digital Conversion (ADC) and Digital-to-Analog Conversion (DAC) (Erickson, 1996). Without proper calibration, these issues can negatively impact system performance. This research introduces an automated calibration framework aimed at overcoming these challenges. By dynamically adjusting input and output values in real time, the framework reduces errors, improves signal consistency, and ensures more precise and stable interactions between the PLC and the GICS emulation card, ultimately enhancing system accuracy and reliability.

1.2 Research Objectives

The main objective of this research is to develop an automated system capable of learning and correcting signal offsets in PLC systems, specifically focusing on interactions with General Instrument Control System (GICS) emulation cards. The specific goals of this study are:

1.2.1 Automated Learning of Offsets

To design a system that can automatically detect and learn signal offsets that occur during Analog-to-Digital Conversion (ADC) and Digital-to-Analog Conversion (DAC), enhancing overall accuracy in PLC systems.

1.2.2 Automatic Learning of Auto-Calibration

To create a system that can autonomously learn and perform auto-calibration, adjust the system in real-time to improve the accuracy of PLC inputs and outputs without manual intervention.

1.2.3 Testing and Validation:

To develop a flexible mechanism that can be easily integrated with different PLCs and create a method that can be adapted for use with future PLC applications as technology evolves.

1.3 Scope of Research

This research focuses on addressing signal offset challenges in Programmable Logic Controllers (PLCs), specifically within controlled systems and environments. The scope is designed to ensure the research objectives are achievable within the defined constraints, while providing actionable and meaningful results. The key areas of the research are outlined as follows:

1.3.1 Systems and Tools Focus

1. The study centers on the SIEMENS SIMATIC S7-1500 PLC and its interaction with GICS emulation cards. The PLC processes a 10-bit analog input, converts it into a 16-bit digital signal, and outputs a 12-bit analog signal to maintain compatibility with the GICS system.

2. The SIEMENS SIMATIC S7-1500 PLC was selected for its advanced processing capabilities and seamless integration with TIA Portal software, which facilitates real-time calibration and debugging.

3. The research involves the interaction between the PLC and GICS emulation card, focusing on signal processing during Analog-to-Digital Conversion (ADC) and Digital-to-Analog Conversion (DAC), where signal offsets and noise are mitigated through the calibration process.

1.3.2 Calibration Methodology

1. This study employs a feedback-driven, automatic calibration framework using TIA Portal software. The methodology includes the following:

   1) Offset Detection: Identifying discrepancies in the signal during transmission and reception using Ladder Diagrams and feedback loops.

   2) Real-Time Adjustments: Applying iterative calibration to adjust signal outputs and align them with theoretical reference values.

   3) Voltage Calibration: Ensuring that voltage levels remain within the acceptable range of -10 to +10 volts, validated through input signals ranging from 512 to 1023.

2. Calibration Evaluation Metrics

   1) Error Reduction: Comparing pre-calibration and post-calibration results to measure the reduction in error margins.

   2) Signal Consistency: Analyzing the stability of the data across multiple iterations of calibration.

   3) Voltage Alignment: Verifying that precise voltage communication is maintained between the GICS emulation card and the PLC.

1.4 Limitations of the Research

While this research makes significant contributions, several limitations must be considered:

1.4.1 System-Specific Focus

The study is centered on the SIEMENS SIMATIC S7-1500 PLC and GICS emulation cards, with compatibility primarily tied to the GICS Tester software. This software sends input values that are analyzed in TIA Portal to validate the proposed calibration framework.

1.4.2 Simulation-Based Testing

The research was conducted in a controlled simulation environment, which may not fully replicate the complexities of real-world industrial settings. Factors such as hardware degradation, environmental variations, and external interferences could influence system performance in actual operations, affecting the external validity of the results.

1.4.3 Signal Range and Resolution

This study is limited to a signal input voltage range of -10V to +10V, corresponding to a 10-bit resolution system with an input range of 512 to 1023. The performance outside this range, or at higher resolution levels (e.g., 12-bit or 16-bit), was not explored, which may restrict the applicability of the results to systems with different specifications.

1.4.4 Short-Term Evaluation

The experimental phase was conducted over six months, which may not accurately reflect the long-term performance of the system under continuous operation. The study did not assess factors like prolonged use, environmental wear, or potential signal drift over time, limiting the understanding of the system's long-term reliability.

1.5 Definitions of Specific Terms

1.5.1 PLC (Programmable Logic Controller)

A digital computer designed to automate industrial processes. PLCs receive input signals from sensors, process the control logic programmed into them, and send and receive 16-bit signals to and from actuators in real-time. This allows for precise and efficient control in various industrial applications such as manufacturing, energy management, and water treatment.

1.5.2 GICS (General Instrument Control System) Emulation Card

A specialized hardware card used to simulate analog signals in control systems. It generates and sends 10-bit input signals to PLCs and receives 12-bit output signals, enabling the testing and validation of system accuracy and calibration in a simulated environment.

1.5.3 Calibration

The process of adjusting a system or instrument to ensure its accuracy and reliability. Calibration involves both correcting known offset values and applying filter values to adjust the system. The goal is to identify and apply these adjustments to achieve the correct output value, ensuring consistent and accurate operation across various industrial applications.

CHAPTER 2

LITERATURE REVIEW

2.1 Literature Purpose

Signal offsets in Programmable Logic Controllers (PLCs) present a persistent challenge in industrial automation, particularly when interfacing with General Instrument Control System (GICS) emulation cards. These offsets, which often occur during Analog-to-Digital Conversion (ADC) and Digital-to-Analog Conversion (DAC), compromise data accuracy and lead to inefficiencies in automated processes (Erickson, 1996). Modern industrial applications, such as manufacturing, energy management, and chemical processing, require precise control and reliable data transmission to maintain operational efficiency and safety (Jadhav & Patil, 2014). Uncalibrated systems can result in increased downtime, defective products, and higher operational costs, making it critical to address these issues (Lu, 2022). Various calibration methodologies have been explored to mitigate signal offsets. Traditional manual calibration, while commonly used, is time-consuming and susceptible to human error (Automation Direct, 2004). In contrast, automated calibration systems, particularly those incorporating predictive algorithms, have shown potential in minimizing error margins and improving signal stability (Chan et al., 2017). However, the integration of real-time calibration within complex systems, such as PLCs paired with GICS emulation cards, remains largely underexplored (Open University, 2018). This research seeks to bridge this gap by proposing a novel real-time calibration approach that combines dynamic feedback and offset correction within a unified framework (Toronto Metropolitan University, 2021). The primary objective is to dynamically calculate and correct offsets, thereby enhancing data stability and accuracy in real-time. By processing a 10-bit analog signal generated by a GICS emulation card, converting it into a 16-bit digital signal via the PLC, and transmitting it back as a 12-bit analog signal, this approach ensures seamless bidirectional communication between the two systems (Szymczyk et al., 2023). This research leverages the SIEMENS SIMATIC S7-1500 PLC and TIA Portal software, integrating real-time auto-calibration techniques to optimize signal alignment.

This research carries far-reaching implications, aimed at:

1. Minimize reliance on manual calibration, reducing both labor costs and operational downtime.

2. Enhance system reliability by ensuring consistent and accurate data transmission.

2.2 Related Work

2.2.1 SCADA Systems

SCADA (Supervisory Control and Data Acquisition) systems are integral to industrial control, providing a comprehensive solution for monitoring and managing industrial processes (Jayasinghe, 2018). These systems consist of various components, including sensors, PLCs, communication networks, and software interfaces, which facilitate real-time monitoring and remote control of equipment across hierarchical levels from centralized locations.

SCADA System Levels:

Level 0: Field Devices such as sensors and actuators that interact directly with the physical processes.

Level 1: Industrial Control Systems (PLCs and DCS) that control the process and respond to signals input.

Level 2: Monitoring Systems that collect and analyze data from field devices and control systems.

Level 3: Production Management systems that oversee and optimize the manufacturing process.

Level 4: Enterprise Resource Planning (ERP) systems that integrate business management and resource allocation.

The relationship between Programmable Logic Controllers (PLCs) and Supervisory Control and Data Acquisition (SCADA) systems is foundational to modern industrial automation.

Each system plays a distinct yet complementary role:

1. PLCs are primary control devices that handle real-time automation tasks. They interact directly with machines, sensors, and actuators to execute control logic, capture input from the field, and adjust parameters to maintain process control.

2. SCADA systems, on the other hand, operate at a supervisory level, providing a platform for monitoring and managing large-scale industrial processes. SCADA systems collect real-time data from multiple PLCs across machines or locations and allow operators to visualize, analyze, and control the entire system remotely (Khaldi, 2019).

3. Communication between PLCs and SCADA systems occurs via standard protocols such as Modbus, Ethernet/IP, or Profibus, enabling seamless data exchange. SCADA systems offer a comprehensive view of the industrial process by providing:

1) Centralized monitoring of all connected equipment.

2) Historical data logging for performance tracking and analysis.

3) Alarm management to notify operators of issues.

4) Long-term process optimization based on data trends and analysis.

While PLCs focus on local control and immediate response to system changes, SCADA systems provide a higher-level view and enable decision-making based on both current data and historical trends (McKoy, 2018).

2.2.2 Integration of Auto-Calibration with PLC and SCADA Systems

Integrating auto-calibration with Programmable Logic Controllers (PLCs) and Supervisory Control and Data Acquisition (SCADA) systems plays a crucial role in enhancing industrial automation. This integration brings significant improvements in precision, efficiency, and reduced manual intervention across various industries. Auto-calibration enables PLCs to automatically adjust the performance of machines, sensors, and other critical equipment based on real-time data feedback. As a result, the system can maintain optimal functionality without requiring constant human oversight. This capability is especially valuable in industries where accuracy is paramount, such as in chemical processing, pharmaceuticals, and manufacturing. These sectors often involve complex processes that require tight control of variables such as temperature, pressure, flow rate, and concentration. Even small deviations from the desired conditions can lead to product quality issues, safety hazards, or expensive downtime. Auto-calibration ensures that these systems continuously operate within the defined optimal parameters, even as environmental factors or equipment conditions change over time. With the incorporation of automatic calibration, PLCs are able to continuously monitor equipment performance, detect any deviations from the set point, and recalibrate systems in real time (Khaldi, 2019). This dynamic adjustment helps maintain system stability and minimizes errors. By addressing performance degradation or sensor drift immediately, the system prevents downtime and minimizes the likelihood of major failures or costly repairs. Furthermore, it significantly improves the accuracy of data collected from sensors and other input devices, leading to more reliable control and decision-making within the SCADA system. Real-time recalibration also reduces the need for frequent manual interventions, which can be labor-intensive and prone to human error. Additionally, the automation of these processes ensures more consistent performance over time, leading to increased reliability and predictability in industrial operations. The ability to integrate auto-calibration with both PLCs and SCADA systems also opens the door for more advanced applications, such as predictive maintenance, process optimization, and adaptive control systems. By utilizing historical performance data and incorporating machine learning algorithms, these systems can anticipate potential issues before they arise, further enhancing system longevity and productivity (McKoy, 2018).

2.2.3 Working and Importance of PLC in Industrial System

Programmable Logic Controllers (PLCs) are sophisticated digital systems designed to control and optimize the performance of industrial machinery. Their role in industrial automation extends far beyond simple monitoring, as they are responsible for executing complex control tasks, continuously tracking sensor data, and dynamically managing outputs to control actuators and other related devices (Cleveland State University, 2016). PLCs enable precise and efficient coordination in industrial operations through advanced programming and real-time processing capabilities. In addition to enhancing operational efficiency, PLCs play a crucial role in safety by implementing fail-safe mechanisms and emergency shutdown systems when necessary. This dual approach maximizing both efficiency and safety ensures smooth operation and helps mitigate risks across various industries. PLCs are integral to sectors such as manufacturing, automotive, chemical processing, power generation, and food and beverage. Their flexibility and versatility make them essential for optimizing production processes, improving efficiency, and ensuring product quality. As the backbone of industrial automation, PLCs monitor and control processes such as assembly lines, robotic systems, chemical reactors, and packaging systems (Bhaskar et al., 2016). By streamlining operations and reducing manual intervention, PLCs drive innovation and help businesses remain competitive in a rapidly evolving market. Furthermore, PLCs are increasingly integrated with cutting-edge technologies, including the Internet of Things (IoT) and Artificial Intelligence (AI), significantly expanding their capabilities and accelerating industrial advancements (Shabbir, 2015). These integrations allow for real-time data collection, predictive maintenance, and smarter automation, boosting overall operational performance. Compared to traditional relay-based control systems, PLCs offer significant advantages by providing greater flexibility, efficiency, and scalability. Their modular design allows for easy integration into a wide variety of applications, adapting to the specific needs of each system. Built with dedicated hardware components, including a central processing unit (CPU), input/output modules, memory, and communication ports, PLCs are designed to withstand harsh industrial conditions and ensure long-term, reliable performance. PLCs also come with specialized programming software that enables engineers to create customized control logic tailored to specific industrial applications. Mastery of PLC programming languages, such as Ladder Diagrams, Function Block Diagrams, Structured Text, and Sequence Function Diagrams, is essential to fully harness the potential of these systems. These languages optimize the unique features of the PLC hardware, allowing for fine-tuned control over applications like motion control, process automation, and modular production (Khaldi, 2019). The combination of robust hardware, flexible software, and integration with modern technologies makes PLCs invaluable tools for enhancing industrial efficiency, productivity, and safety. Their role in industrial systems continues to evolve, driving greater precision, reducing downtime, and ensuring industries can meet the challenges of today's fast-paced technological landscape.

2.2.4 Real-time Automatic Calibration Systems

1. Real-time automatic calibration systems leverage advanced processing capabilities to eliminate the need for manual intervention. These systems can dynamically adjust offsets as environmental conditions or hardware performance fluctuate, ensuring that the system remains accurate and reliable throughout its operation.

2. For instance, using the SIEMENS SIMATIC S7-1500 PLC, the calibration routine can be seamlessly integrated into the system's regular operations, eliminating the need for downtime. This integration enables continuous, automatic adjustment of offsets, enhancing system accuracy and minimizing the potential for errors.

2.2.5 Analog Calibration in Industrial Systems

Analog calibration is essential for ensuring the accuracy of measurement systems by correcting errors caused by factors such as sensor drift, temperature fluctuations, or electrical noise. During calibration, the system's measured values are compared to known standards, and adjustments are made to align the system's output with the actual value. This ensures that the system consistently provides accurate and reliable data over time. In industrial settings, analog calibration is crucial for accurately measuring parameters such as pressure, temperature, and voltage. For instance, when calibrating a pressure sensor, engineers apply known pressure values and adjust the sensor's output to match these values. This process is vital for maintaining the reliability of measurement systems, ensuring that data used for process control, monitoring, and analysis remains consistent and accurate. In the context of Programmable Logic Controllers (PLCs) and Generic Interface Card System (GICS) emulation cards, analog calibration focuses on adjusting analog signal values to ensure precise communication and accurate representation of real-world data. The PLC controls and manages analog inputs, such as temperature, pressure, or voltage, and transmits these signals to the GICS emulation card. The emulation card simulates or tests the system, and during the calibration process, the signal ranges such as current (4-20 mA) or voltage (0-10 V) are fine-tuned to ensure that the analog data sent by the PLC matches the expected values at the GICS card. These adjustments are critical for ensuring that the GICS emulation card accurately replicates the behavior of sensors or actuators in a controlled test environment. Proper analog calibration helps prevent signal drift, noise, and misinterpretation of data, ensuring that the test environment accurately reflects real-world conditions. This is particularly important for validating the control system before full-scale implementation. Implementing a PLC calibration routine ensures accurate instrument readings by applying the calibration method to any analog measurement, such as pressure, amperage, weight, length, or force (Automation Direct, 2004). This involves converting the measured physical parameter to numerical readings via the analog input and adjusting the system accordingly. Additionally, the GICS platform serves as an industrial system laboratory that includes control systems such as PLCs and cybersecurity subsystems, like protocol analyzers. This platform plays a key role in the development and testing of calibration methods within a controlled, simulated environment (Fakih et al., 2024). By using this platform, engineers can test and refine calibration routines, ensuring that systems perform as intended in real-world industrial settings.

2.2.6 Digital Calibration in Industrial Automation

Digital calibration is a critical process in industrial automation, ensuring the accuracy and reliability of measurements from digital sensors, transmitters, and other devices connected to systems such as Programmable Logic Controllers (PLCs) and Supervisory Control and Data Acquisition (SCADA) systems. By adjusting digital signals, digital calibration improves measurement precision, ensuring that sensor readings accurately reflect the physical quantities they measure (Morel et al., 2023). This process is especially important in industries such as pharmaceuticals, food processing, and chemical production, where precision is essential for product quality and safety. As advanced technologies like Internet of Things (IoT) devices and cloud-based analytics become more widespread, the importance of digital calibration continues to grow. It enables more sophisticated data analysis and enhances operational efficiency by ensuring that measurements are accurate, reliable, and aligned with real-world conditions. By providing reliable data, digital calibration supports decision-making processes that optimize production and minimize errors. Integrating digital calibration with PLCs and Generic Interface Control System (GICS) emulation cards is crucial for maintaining system integrity. GICS emulation cards act as virtual test benches, allowing PLCs to simulate and calibrate digital signals without the need for physical hardware. This streamlines the testing and validation process, reducing both time and cost associated with physical setups (Azam et al., 2020). The ability to simulate real-world conditions using GICS emulation cards ensures that PLCs can properly calibrate the digital inputs they receive and accurately reproduce measurements such as temperature, pressure, and flow rate. Leveraging digital calibration with GICS emulation cards helps companies maintain consistent performance across their control systems, reduce measurement errors, and comply with stringent industry quality standards. Automated calibration processes enabled by GICS significantly reduce the need for manual intervention, minimizing the likelihood of human error and enhancing system uptime. Moreover, this integration improves diagnostic capabilities, allowing operators to quickly identify and resolve signal processing inconsistencies, which can prevent costly production delays. As industries continue to evolve with the adoption of intelligent technologies and data-driven decision-making, the synergy between PLCs, digital calibration, and GICS emulation cards is becoming increasingly essential. This integration drives innovation, improves operational efficiency, and supports the development of more agile and responsive manufacturing environments (Altaher et al., 2015).

2.2.7 Voltage Calibration in Industrial Systems

Voltage calibration ensures that a system's voltage measurements or outputs are accurate and aligned with a known reference or standard. It is commonly used in electrical systems, instruments, and control devices to maintain precision and reliability. Voltage calibration compares the system's voltage levels with a calibrated reference voltage, which could be generated by a power supply, sensor, or control unit. If discrepancies are found, technicians adjust the voltage output or display to match the reference standard. This process is essential in industries such as manufacturing, power distribution, and laboratory environments, where precise voltage control is crucial for optimal performance and safety. Voltage calibration helps prevent errors, minimize deviations, and ensures that systems operate within their intended specifications (Calado et al., 2023). In the context of a Programmable Logic Controller (PLC) and a Generic Interface Control System (GICS) emulation card, voltage calibration ensures accurate signal interpretation and seamless communication between the two systems. The calibration process involves the following steps:

1. Generating specific voltage outputs from the GICS card.
2. Simulating sensor signals and verifying that the PLC correctly reads these voltage signals.

Calibration typically begins with selecting an appropriate voltage range, such as 0–5V or 0–10V, based on the system's requirements. The GICS card is then programmed to output reference voltages, such as 0V, 5V, and 10V, which are compared with the voltage values received by the PLC input module. If discrepancies arise between the expected and actual measured values, the PLC's input settings are adjusted through the configuration software. This adjustment fine-tunes the voltage scaling and offset values to ensure that the PLC accurately interprets the voltage signals and reflects the conditions simulated by the GICS card. After making the necessary adjustments, the calibration process is repeated to ensure that the measured values match the reference values across all points in the voltage range. Proper voltage calibration between a PLC and a GICS emulation card is critical for maintaining reliable process control, especially in systems where accurate voltage measurements directly impact automation performance.

2.2.8 Offset Errors in PLC and GICS Systems

In Programmable Logic Controllers (PLCs) and General Instrument Control System (GICS) emulation cards, an offset refers to the deviation or error between the actual signal value and the expected signal value during Analog-to-Digital Conversion (ADC) and Digital-to-Analog Conversion (DAC) (Owen, 1975). This discrepancy results in inaccurate data being processed by the system, causing errors in signal interpretation and control decisions (Toronto Metropolitan University, 2021).

Examples of Offset Errors:

1. A 10V analog signal input into the GICS card might be interpreted as 9.8V or 10.2V by the PLC.
2. Similarly, when the PLC sends a signal back to the GICS card, the output may not match the expected value due to inaccuracies in the DAC.

Although these errors may appear minor, they can have significant consequences for industrial automation systems, particularly in high-precision applications.

Addressing offset errors is critical in several key areas:

1. Data Accuracy and Signal Integrity

Signal accuracy is crucial for effective industrial control systems. Uncompensated offsets can distort critical measurements, leading to poor control decisions. For instance, in temperature-controlled chemical processes, even a small offset can result in out-of-specification products, leading to resource wastage and increased costs.

2. System Reliability and Stability

Persistent offset errors can degrade system stability over time, causing operational inconsistencies. This is particularly critical in sectors such as energy management, where signal inaccuracies could disrupt grid operations, potentially leading to blackouts or equipment damage.

3. Operational Efficiency and Cost Implications

Unaddressed offsets often require frequent manual recalibrations, leading to production downtime and increased labor costs. Automated calibration systems that eliminate these offsets can significantly reduce operational burdens, resulting in cost savings and enhanced productivity.

4. Safety and Risk Mitigation

In industries like oil and gas, power generation, and pharmaceuticals, signal accuracy is directly linked to safety. Inaccurate signals caused by offsets can create unsafe conditions, jeopardizing both personnel and equipment.

Addressing offset in PLC systems

Addressing offsets in PLC systems requires precise calibration processes to detect and compensate for signal deviations. These processes can be carried out using both manual and automated methods, with automated systems becoming more common in modern industrial environments.

1. Offset Detection

1) Offset detection involves comparing the measured signal against a known reference value. This can be done in real-time using feedback loops within the PLC system or through pre-configured test routines.

2) For example, a 10-bit analog signal is input into a GICS emulation card, processed by the PLC into a 16-bit digital format, and then converted back into a 12-bit analog signal. Discrepancies between the input and output values indicate the presence of an offset.

2. Offset Compensation

1) Offset compensation involves applying a calculated adjustment to align the measured signal with the expected value. This is typically implemented through algorithms programmed into the PLC, often using Ladder Logic or function blocks within software like TIA Portal.

2) The system continuously monitors and adjusts the signal in real-time to maintain alignment, ensuring that the data processed by the system remains accurate and reliable.

2.2.9 Filtering Techniques in PLC and GICS Calibration

In industrial automation, precise signal calibration is essential for the reliable operation of systems such as Programmable Logic Controllers (PLCs) and Generic Interface Control System (GICS) emulation cards. Filtering techniques are commonly used in calibration processes to enhance signal quality by eliminating high-frequency noise and compensating for offset errors. These methods ensure that sensor data is accurate and reliable, which is vital for real-time control and process optimization.

1. Low-Pass Filtering in Signal Calibration and Offset Compensation

Low-pass filtering is fundamental in reducing unwanted noise and improving signal clarity in industrial systems. This technique attenuates high-frequency components of a signal while allowing low-frequency components to pass, making it essential in systems like PLCs and GICS emulation cards. The ability to smooth sensor signals and compensate for offset errors enhances the accuracy of real-time data used for control and automation.

1) Low-Pass Filtering

Low-pass filters allow signals with a frequency lower than a specific cutoff to pass through, while higher frequencies are attenuated. This technique is essential in noise reduction and offset compensation in PLCs and GICS systems.

Formula and Working: A low-pass filter can be described by the equation:

$$y(t) = \alpha \cdot x(t) + (1 - \alpha) \cdot y(t - 1)$$

where α is the smoothing factor, calculated as:

$$\alpha = \frac{T}{T + \tau}$$

Where:

a. T is the sampling period,

b. τ is the time constant

The filter's output smooths high-frequency noise by averaging the recent signal and previous output.

2) Low-Pass Filtering in PLCs and GICS Emulation Cards

The Low-Pass Filter (LPF) is one of the simplest, yet most widely used techniques for noise reduction and signal smoothing in industrial automation. It is particularly effective for systems requiring basic noise suppression and signal stability.

The output of a low-pass filter is given by the following formula:

$$V\_Filtered = \alpha \times OutputV + (1 - \alpha) \times V\_previousFiltered$$

Where:

a. V_Filtered is the filtered output.

b. α is the filter coefficient, which determines the smoothing factor.

c. OutputV is the raw output value (raw, noisy signal).

d. V_previousFiltered is the previously filtered output value.

The value of α determines the filter's responsiveness. A larger α value allows the filter to react quickly to changes in the input, while a smaller α value provides smoother output by attenuating rapid changes in the signal.

Application in Calibration and Offset Compensation:

In PLCs and GICS emulation cards, Low Pass Filtering is commonly used to reduce noise and filter out high-frequency disturbances from sensor signals. The filtering process helps achieve more accurate reading by reducing signal fluctuation, thus ensuring better calibration. Adjusting the value of α\alphaα allows engineers to control the trade-off between responsiveness and smoothness, making it a flexible tool in real-time industrial applications.

2. Discrete-Time Implementation

In PLC systems, using cascaded low-pass filters is an effective way to reduce high-frequency noise. By applying multiple stages of filtering, each stage progressively smooths the signal, which helps eliminate unwanted high-frequency components without significantly affecting the desired signal. This approach can be particularly beneficial for noisy signals, ensuring the output remains stable and usable for further processing or control actions.

3. Application in Calibration and Offset Compensation

Low-Pass Filter Applications:

1) Calibration Process:

Low pass filters are often used when moderate noise reduction is sufficient. In systems where only slight noise suppression is needed, this simpler approach can be used for calibrating sensor signals. The low pass filter that helps to smooth the signal adequately without introducing complex processing overhead.

2) Offset Compensation:

Low pass filters are useful for addressing small offset errors that may be present in the system, offering a more straightforward approach to ensure signal stability without complex configuration.

3) Noise Suppression:

In less demanding environments where high-frequency noise is minimal, Low pass filters are an efficient and lightweight solution for noise suppression, ensuring signal integrity without excessive computational cost.

2.3 Related Technology

2.3.1 Software Technology

1. TIA Portal Program

TIA Portal is an integrated software suite from Siemens designed for configuring, programming, and managing automation systems such as Programmable Logic Controllers (PLCs), Human-Machine Interfaces (HMIs), and other industrial devices ("1241-2000 - IEEE standard for terminology and test methods for analog-to-digital converters", 2001). It provides a unified environment that simplifies automation tasks, enabling engineers to design, program, and monitor PLCs, HMIs, and motion control systems within a single platform.

TIA Portal supports multiple programming languages, including:

1) Ladder Logic (LAD)
2) Function Block Diagrams (FBD)
3) Structured Text (ST)
4) Sequential Function Charts (SFC)

The seamless integration of hardware configuration, communication setup, and diagnostic tools enables real-time system monitoring and troubleshooting, making TIA Portal highly versatile for various industrial applications. Additionally, PLCSIM, a simulation tool within TIA Portal, allows engineers to simulate PLC programs without requiring physical hardware (Erlandsson & Rahaman, 2013). This facilitates efficient testing and validation before deployment, ensuring program reliability and accuracy.

1) Voltage Calibration with PLC and GICS Simulation Card Using TIA Portal

When performing voltage calibration in industrial systems, utilizing a Programmable Logic Controller (PLC) and a Generic Interface Card System (GICS) emulation card is essential for ensuring the accuracy of voltage signals used in automated processes. TIA Portal (Totally Integrated Automation Portal) serves as the software interface for configuring and fine-tuning the interaction between the PLC and the GICS emulation card during this calibration process.

2) Role of TIA Portal in Voltage Calibration

TIA Portal plays a critical role in facilitating the configuration of both analog inputs and outputs in the PLC system, ensuring that the voltage signals generated or received by the PLC are correctly calibrated. The following steps highlight the process:

a. Configuring Analog Inputs and Outputs:

Using TIA Portal, engineers can configure the analog input and output channels that are responsible for voltage measurements. The TIA Portal interface allows users to set the range of voltage signals and signal scaling for both inputs and outputs. This ensures that the PLC can correctly interpret the voltage levels and communicate them to the GICS emulation card.

b. Monitoring and Adjusting Voltage Levels:

TIA Portal enables real-time monitoring of voltage levels within the system. Engineers can track the actual voltage signals generated by the PLC and compare them with the expected values for each parameter (such as voltage ranges, thresholds, and output levels). If discrepancies are found, adjustments can be made directly through the software, ensuring that the output from the PLC matches the expected values.

c. Verification of Signal Transmission:

Once the voltage signals are adjusted, the GICS emulation card simulates the load or environment that the system will operate in. This simulation is essential for testing whether the voltage levels generated by the PLC are accurately transmitted and reflected in the GICS system. By using TIA Portal to fine-tune the calibration, engineers can verify the integrity of the signal transmission, ensuring that both the PLC and the emulation card are aligned.

d. Real-Time Adjustments:

The real-time capabilities of TIA Portal streamline the calibration process by allowing immediate feedback and adjustment of voltage levels. This reduces the need for manual recalibration and enhances the overall precision of the system. With continuous monitoring, the software helps ensure that the voltage levels between the PLC and the GICS emulation card stay within the required specifications, improving accuracy and reliability.

3) Benefits of Using TIA Portal for Voltage Calibration:

a. Efficient Calibration:

By integrating the PLC with the GICS emulation card through TIA Portal, engineers can streamline the voltage calibration process. The software ensures that the voltage signals are accurately adjusted and transmitted without the need for complex manual intervention.

b. Real-Time Monitoring:

TIA Portal provides real-time access to the voltage levels, making it easier to troubleshoot and adjust the system as needed during the calibration process. This significantly improves the efficiency of calibration tasks and reduces errors.

2. GICS Tester

The GICS Tester is a software interface developed for interaction with GICS emulation cards. It allows users to simulate physical processes by generating and analyzing analog signals. The GICS Tester establishes a connection between the PLC and the GICS emulation card, enabling users to check data communication and perform calibration tasks. When operating in auto-calibration mode with a PLC and GICS simulation card, the GICS Tester automates the process of balancing and verifying signal accuracy between the two systems. It is designed to facilitate the testing and calibration of industrial control systems, interfacing with both the PLC and the GICS emulation card to measure and adjust signal levels (e.g., voltage or current) to ensure they meet predefined standards (Siemens, 2018).

1) Auto-Calibration Process with PLC and GICS Simulation Card Using GICS tester:

The auto-calibration process is a critical aspect of ensuring the accuracy and reliability of systems that involve Programmable Logic Controllers (PLCs) and Generic Interface Control System (GICS) emulation cards. The process automates the calibration procedure, significantly reducing manual intervention and ensuring that the system operates with high precision throughout.

2) How Auto-Calibration Works

a. Signal Generation and Transmission:

During auto-calibration, the GICS Tester generates test signals and sends them to both the PLC and the GICS simulation card. These signals typically represent standard analog input ranges such as 4-20 mA or 0-10 V, which are commonly used in industrial systems to measure parameters like temperature, pressure, or flow rate.

b. Signal Comparison:

Once the test signals are sent, the system compares the responses from both the PLC and the GICS simulation card. The GICS Tester continuously monitors the output signals and compares them with the expected values, identifying any discrepancies between the two devices.

c. Automatic Adjustment:

The auto-calibration system dynamically adjusts any deviations in the signals, ensuring that both the PLC and GICS emulation card are synchronized and that their outputs match the desired values. This process removes the need for manual intervention, saving time and reducing the likelihood of human error during the calibration process.

d. Real-Time Monitoring:

The GICS Tester plays a crucial role in real-time monitoring. It ensures that the calibration settings are continuously updated, and any discrepancies are promptly corrected. This real-time feedback is vital for achieving accurate calibration in complex industrial systems, where even minor errors in signal transmission can lead to significant deviations in performance.

e. User Interaction and Adjustment:

Although the calibration process is automated, users can interact with the system by entering various signal parameters, observing the PLC's response, and making adjustments to the calibration settings as needed. This feature provides flexibility in fine-tuning the system, allowing users to adapt the calibration process to specific requirements or conditions.

3) Benefits of Using GICS Tester for Voltage Calibration

The GICS Tester plays a crucial role in ensuring precise voltage calibration in industrial automation systems that integrate Programmable Logic Controllers (PLCs) and Generic Interface Control System (GICS) emulation cards. One of the primary benefits of using the GICS Tester is its ability to automate the calibration process, significantly reducing the need for manual adjustments and minimizing human errors. Manual calibration can be time-consuming and prone to inconsistencies, whereas the GICS Tester ensures high-precision voltage calibration by continuously monitoring and adjusting voltage levels in real time. By dynamically fine-tuning the calibration settings, the system can maintain stable and accurate voltage outputs, ensuring seamless synchronization between the PLC and GICS emulation card.

2.3.2 Hardware Technology

1. Central Processing Unit (CPU)

The Central Processing Unit (CPU) is the heart of a computer system, responsible for executing instructions and processing data. In the context of Programmable Logic Controllers (PLCs), the CPU executes user-defined control programs and manages input and output signals to control industrial processes. In automatic calibration with a PLC and GICS (Generic Interface Card System) simulation card, the CPU plays a central role in coordinating and controlling the calibration process to ensure accurate signal exchange between the PLC and the GICS system. The CPU manages the entire calibration routine, processing input signals from the GICS card, comparing them to expected values, and adjusting system outputs accordingly. The CPU continuously monitors both analog and digital signals, such as voltage or current, detecting discrepancies between actual and expected values. Based on the calibration algorithm, the CPU automatically corrects these signals to match the required specifications. This eliminates the need for manual calibration, providing precise and efficient system tuning. The CPU synchronizes the PLC with the GICS card and ensures the correct functioning of both the control logic and hardware interface before deployment [24].

2. PLC (Programmable Logic Controller)

A Programmable Logic Controller (PLC) is an industrial-grade computer that automates control processes. The SIEMENS SIMATIC S7-1500 is an advanced PLC that supports high-speed processing, extensive I/O functions, and integrated communication protocols. It uses programming languages such as Ladder Logic and Structured Text to execute control algorithms and manage field devices. In auto-calibration with a PLC and GICS simulation card, the PLC plays a critical role in managing and executing the calibration process to ensure accurate signal synchronization. Programmed through TIA Portal software, the PLC sends and receives both analog and digital signals to and from the GICS simulation card, emulating real-world devices such as sensors and actuators. During the calibration process, the PLC monitors incoming signals from the GICS card, compares them with predefined reference values or expected ranges (e.g., voltage or current), and adjusts its output signals accordingly (Shedge & Tade, 2019). The PLC's internal logic processes data in real-time, detects deviations, and corrects them automatically, reducing the need for manual adjustments, minimizing human error, and improving efficiency. The PLC ensures precise synchronization of voltage, current, or data signals between the control logic and the emulated hardware (Siemens, 2021).

3. GICs emulation Card (Generic Interface Card System)

The GICS emulation card simulates analog signals for testing and calibration purposes. It interfaces with the PLC, generating and receiving both analog and digital signals to mimic actual process conditions. This allows engineers to test PLC programs and calibration routines without the need for physical sensors and actuators. In auto-calibration, the GICS emulation card works in tandem with the PLC to simulate real devices and facilitate automatic signal calibration. The emulation card simulates hardware components such as sensors, actuators, or other field devices that the PLC typically controls in a real-world process. During auto-calibration, the GICS card exchanges signals with the PLC, enabling the PLC to fine-tune its inputs and outputs based on the simulated conditions. The emulation card processes the test signals from the PLC, providing feedback to the PLC, helping it automatically adjust any signal deviations, such as voltage or current ranges. This ensures that the signals sent and received align with predefined specifications or real-life scenarios encountered by the PLC. The feedback loop between the PLC and the GICS emulation card ensures that the system is correctly configured before deployment, guaranteeing the accuracy and stability of the calibration process (Yao et al., 2024).

Figure 2.1 PLC

Figure 2.2 GICS Tester

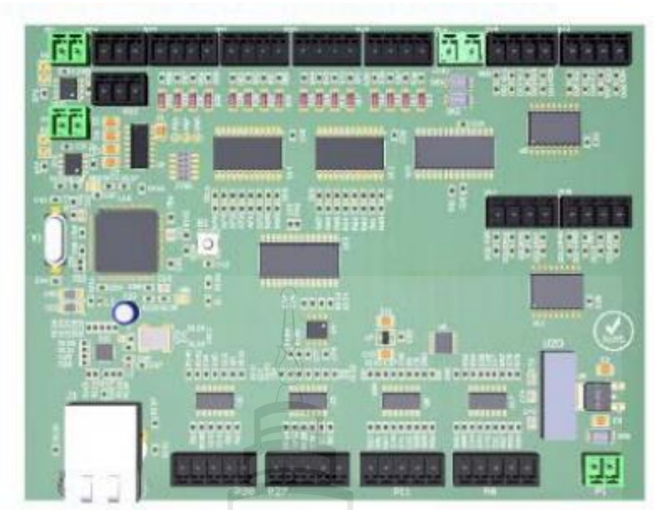


Figure 2.3 Emulation Card (GICs Emulation Card)

CHAPTER 3

METHODOLOGY

3.1 Proposed Method

The proposed method focuses on developing an automated calibration framework for Programmable Logic Controllers (PLCs) to address challenges related to signal offsets and inaccuracies, particularly in systems involving General Instrument Control System (GICS) emulation cards.

3.1.1 Signal Offset Detection and Analysis

The first step involves detecting and quantifying signal offsets that arise during the Analog-to-Digital Conversion (ADC) and Digital-to-Analog Conversion (DAC) processes. Discrepancies between the expected reference signals and the actual PLC outputs will be measured using real-time feedback loops and Ladder Diagrams. This process will identify errors introduced during signal transmission between the PLC and GICS.

3.1.2 Dynamic Adjustment for Real-Time Calibration

The core of the proposed method is to develop a dynamic system that automatically adjusts input and output values in real time. Since GICS accepts 10-bit input signals and outputs 12-bit signals, the PLC will receive a 10-bit input from GICS, convert it to a 16-bit digital signal for processing, and then send the processed output back to the GICS card. The system will continually adapt the signal based on detected offsets, minimizing errors from noise interference and ensuring more accurate signal exchanges.

3.1.3 Iterative Calibration and Voltage Alignment

The calibration process will involve iterative adjustments to align the PLC's outputs with the expected reference values. The system will ensure that voltage levels remain within the acceptable range of -10V to +10V, as per the GICS specifications. This step will also involve continuous voltage alignment to guarantee consistent and accurate signal communication between the PLC and GICS.

3.1.4 General Applicability and Future Adaptability

The system will be designed to be adaptable to other PLC models beyond the SIEMENS SIMATIC S7-1500, making it applicable across various industrial environments. Moreover, the framework will be scalable and capable of evolving with future PLC applications, ensuring it meets the needs of future industrial automation systems.

3.2 Our Designed Framework

3.2.1 Finding the Offset Value Using TIA Portal Program

The first objective of this research is to determine the offset value needed to ensure accurate output values from the GICS emulation card before transmitting them to the PLCs. This process involves the following steps:

1. Real-Time Offset Check:

   Use the TIA Portal program to perform a real-time analysis of the output generated by the GICS emulation card, monitoring how the system behaves during signal transmission and processing.

2. Data Capture:

   Collect output data from the GICS emulation card while systematically varying the input signals. This step helps to understand the relationship between input signal changes and the resulting output, capturing any irregularities.

3. Offset Calculation:

   Analyze the collected data to identify discrepancies between the actual output from the GICS emulation card and the expected reference values. Visualize these discrepancies by plotting the actual output values against the expected ones, which will help to clearly identify any offset values.

4. Adjusting Output Values:

   Calculate the necessary offset correction to align the output values with the desired reference range. By applying the calculated offset, the output from the GICS emulation card can be adjusted to ensure it falls within the expected range before being sent to the PLC.

3.2.2 Auto Calibration of Input-Output Data Between GICS Emulation Card and PLC

The second objective focuses on automating the calibration process to achieve stable input-output values between the GICS emulation card and the PLC. This involves:

1. Implementation of an Automated Calibration Routine: Develop a calibration routine within the TIA Portal that continuously monitors and adjusts the input-output relationships between the GICS emulation card and the PLC.

2. Data Analysis: Analyze the real-time data from both the GICS emulation card and the PLC to identify calibration adjustments needed for stability. This may include using algorithms to minimize error margins between the expected and actual values.

3. Real-Time Adjustments: Apply necessary adjustments automatically to the calibration settings in both the GICS emulation card and the PLC, ensuring that the input and output values remain stable and consistent during operation.

3.2.3 Calculation of Communication Voltage Between PLC and GICS Emulation Card

The final objective is to accurately determine the communication voltage values between the GICS emulation card and the PLC. This includes:

1. Voltage Calculation: For each input-output transaction, calculate the communication voltage based on the data collected. This involves examining the relationship between the values sent from the GICS emulation card and those received by the PLC.

2. Validation of Voltage Values: Ensure that the calculated voltage values are within the specified range (-10 to 10 volts) necessary for effective communication.

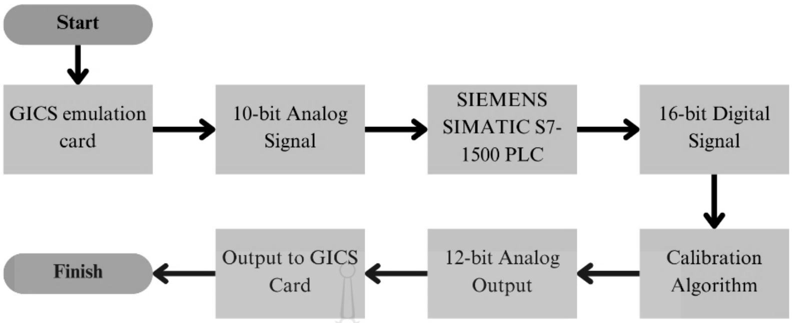


Figure 3.1 Block Diagram Structure

From figure 3.1: Block Diagram Structure: Signal Flow Between GICS Emulation Card, PLC, and Calibration Algorithm.

Components:

1. GICS Emulation Card
   1) Generates 10-bit analog input signals.
   2) Send the signals to the SIEMENS SIMATIC S7-1500 PLC.
2. 10-bit Analog Signal

   Represents the raw signal transmitted from the GICS emulation card before processing.
3. SIEMENS SIMATIC S7-1500 PLC
   1) Converts 10-bit analog signals to 16-bit digital signals.
   2) Passes data through an offset correction algorithm.
   3) Communicates with the Calibration Algorithm for real-time adjustments.
4. 16-bit Digital Signal

   Represents the digitalized version of the input before correction.
5. Calibration Algorithm
   1) Applies real-time corrections to detected offsets.
   2) Ensure signals are accurately adjusted before output.
6. 12-bit Analog Output

   The corrected signal is prepared for transmission back to the GICS emulation card

7. Output to GICS Card

The calibrated 12-bit analog signal is returned to the GICS emulation card, ensuring proper signal integrity.

3.2.4 Ladder Diagram: Real-Time Calibration Logic in TIA Portal.

This ladder logic program implements real-time signal calibration for a PLC-based automation system. The logic ensures that input signals from the GICS Emulation Card are processed, corrected, and stored before being sent to the final output.

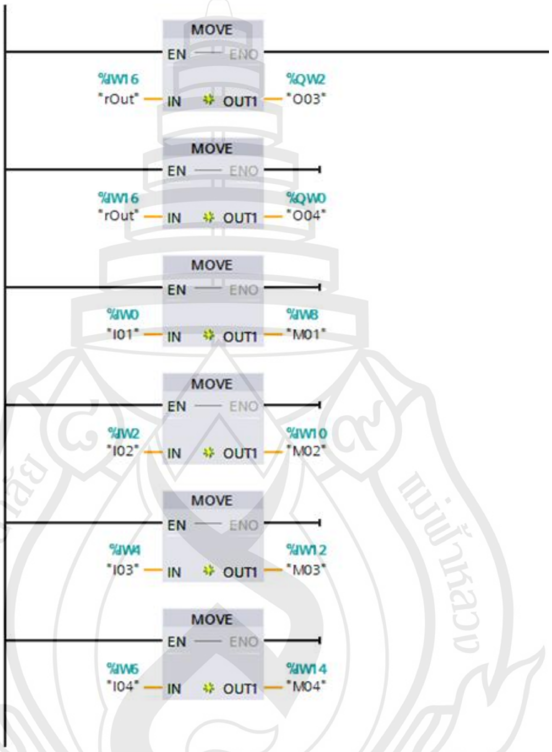


Figure 3.2 Ladder Logic Structure

From figure 3.2: The provided Ladder Diagram represents a PLC logic program that utilizes MOVE instructions to transfer data from input registers to output registers.

This implementation is essential for signal processing, calibration, and control logic in TIA Portal.

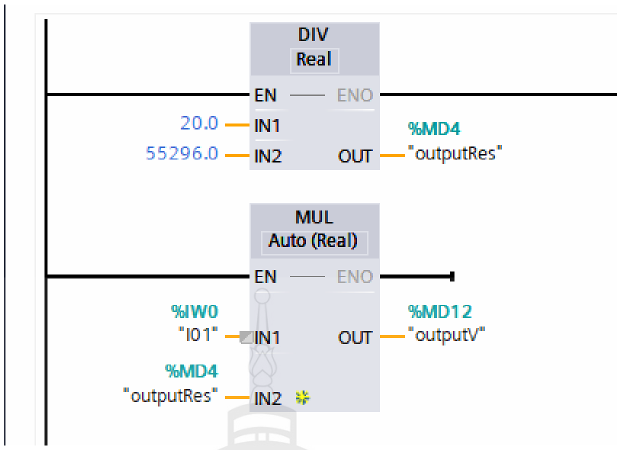


Figure 3.3 Output Conversion in Ladder Logic

From figure 3.3: The Ladder Logic Program shown in the image is part of a real-time signal processing and calibration system in TIA Portal. It consists of two mathematical operations: multiplication (MUL) and division (DIV). These instructions are commonly used for scaling input signals and converting them into appropriate output values.

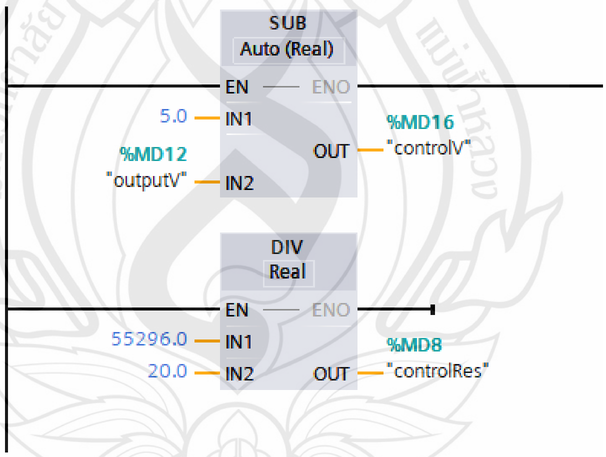


Figure 3.4 Feedback Loop in Ladder Logic

From figure 3.4: This Ladder Logic Program is implementing a feedback control mechanism to ensure that the output voltage is adjusted dynamically. This is achieved using a subtraction (SUB) instruction, which calculates the difference between a reference value and the actual output voltage.

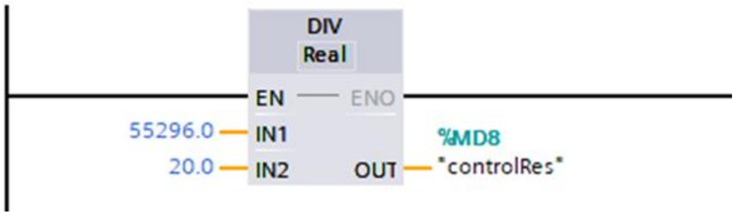


Figure 3.5 Control Conversation in Ladder Logic

From figure 3.5: This Ladder Logic Program is responsible for converting and scaling control values before sending the final processed control signal to the output register. This process ensures that the feedback control system correctly adjusts the output signal.

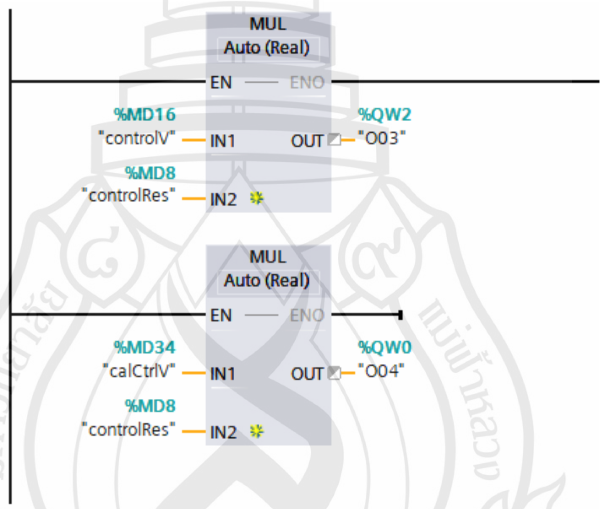


Figure 3.6 Control Processing and Adjustment in Ladder Logic

From figure 3.6: This Ladder Logic is responsible for scaling, adjusting, and applying control corrections to the output signal. The logic consists of three key mathematical operations: division (DIV), multiplication (MUL), and addition (ADD), ensuring that the system dynamically adjusts the control output.

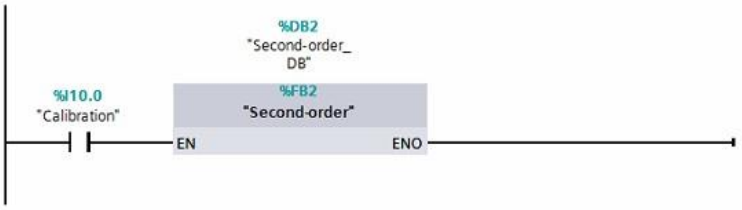


Figure 3.7 Signal Filtering and Calibration

From figure 3.7: This Ladder Logic Program integrates a filtering function, offset calculation, and calibration adjustments to process a signal input.

3.3 Input Calibration

Input calibration is crucial to correct errors in the incoming signals received by the PLC. The system will continuously monitor the 10-bit input signal from the GICS card. By comparing these input signals to known reference values, the system will dynamically adjust for any signal distortion caused by offsets, noise, or interference before they are processed by the PLC. To automatically calibrate the input data from the GICS Emulation Card to the PLC, ensuring accurate signal transmission as follows:

3.3.1 Real-Time Data Collection

1) Continuously collect real-time input data from the GICS Emulation Card, which sends a 10-bit analog input signal to the PLC.

2) Compare the incoming 10-bit input data with the expected reference values to identify any discrepancies.

3.3.2 Signal Analysis

1) Analyze the real-time input signal to detect errors such as noise, offsets, or distortions that could affect the accuracy of data reception at the PLC.

2) Use algorithms to calculate the magnitude of the errors in the signal, based on the difference between the received input and the expected input values.

3.3.3 Automatic Calibration

1) Once discrepancies are detected, automatically apply the necessary corrections to the input signal.

2) Adjust the received input data in real time, correcting for any detected offsets or inaccuracies, and align it with the expected values.

3.3.4 Continuous Monitoring:

The system continuously monitors the adjusted input data to ensure that the calibration is maintained over time. If new discrepancies arise, the system will automatically re-adjust the calibration settings.

3.4 Output Calibration

Output calibration is necessary to ensure the accuracy of the signals sent from the PLC to the GICS emulation card. After processing the input signal, the PLC will adjust the output signals to match expected reference values. The system will be dynamically correct for any discrepancies in the output (e.g., voltage or signal range) by aligning the 12-bit analog output with the desired target values. To automatically calibrate the output data from the PLC to the GICS Emulation Card, ensuring that the PLC's output signal is accurate before being sent to the GICS card as follows:

3.4.1 Real-Time Output Data Monitoring

1. Continuously monitor the 12-bit output signal generated by the PLC, which is sent to the GICS Emulation Card.

2. Compare the output data with the expected reference values to identify any discrepancies or variations.

3.4.2 Output Signal Analysis

1. Analyze the output data to detect any issues such as voltage fluctuations, offsets, or inaccuracies in the signal being sent to the GICS card.

2. Calculate the error margins between the actual and expected output values to understand the degree of necessary calibration.

3.4.3 Automatic Output Adjustment

1. Based on the error analysis, automatically adjust the PLC's output signal to correct for any identified discrepancies.

2. Apply the necessary changes to align the 12-bit output signal with the desired reference range, ensuring that the signal meets the expected specifications before being transmitted to the GICS card.

3.4.4 Feedback Loop for Stabilization:

1. The system applies a continuous feedback loop to monitor the calibrated output signal in real time.

2. If any instability or deviation is detected, the system will automatically re-adjust the output calibration settings to maintain stable performance.

3.5 Method

3.5.1 Method Condition

1. Equipment Configuration

1) GICS Emulation Card

Ensure that the GICS Emulation Card is correctly configured to simulate the required range of inputs and outputs. For example, configure the card to produce 10-bit output values ranging from 512-1023.

2) PLC Settings

Program the PLC to accept and process 16-bit input values within the range of -27,648 to 27,648. Additionally, configure the PLC's output range to match the required specifications for communication with the GICS Emulation Card.

2. Voltage Calibration

Reference Voltage

Use a stable reference voltage source, such as a calibrated power supply, to compare and verify the communication voltage between the GICS Emulation Card and the PLC. The voltage range should be within -10V to 10V to ensure proper functionality and accurate signal processing.

3. Data Acquisition

1) Sampling Rate

Establish a consistent sampling rate for capturing data during the calibration process. This ensures that data is collected reliably at regular intervals for analysis.

2) Data Logging

Use appropriate data logging software to continuously record the input and output values throughout the calibration experiment. This data will be used for later analysis to assess the effectiveness of the calibration procedure.

4. Calibration Procedure

1) Offset Determination:

Start the calibration by sending known input values from the GICS Emulation Card to the PLC and recording the corresponding output values. This serves as a baseline for comparison and determining the required offset corrections.

2) Adjustment Cycle

Systematically apply different offset values and observe their impact on the PLC's output. Adjust these values iteratively to determine the most accurate output, aiming to minimize discrepancies between the expected and actual results.

3) Stability Testing

Perform multiple iterations of sending and receiving signals to assess the stability of the outputs after applying the determined offsets. This step ensures that the calibration is stable and accurate over time, preventing drift or inconsistencies during normal operation.

5. Evaluation Metrics:

1) Accuracy Assessment

Compare the PLC output against the expected reference values to evaluate the accuracy of the calibration. Any deviations should be minimized as much as possible.

2) Consistency Measurement

Analyze the output for any fluctuations or drift over time. Ensuring consistent performance across multiple calibration cycles is key to verifying that the system is operating correctly and that the calibration adjustments are reliable.

3.5.2 Overview of the Ladder Diagram Structure

The ladder diagram is designed to facilitate real-time auto-calibration of the PLC inputs from the GICS Emulation Card. It is structured in several sequential sections, each performing a distinct function that contributes to the overall calibration process. The structure progresses as follows:

1. Data Transfer

The first section of the ladder diagram handles the basic data transfer between the GICS Emulation Card and the PLC, ensuring that the input values are correctly received.

2. Scaling and Mathematical Processing

This section performs necessary scaling and mathematical operations to convert the raw input data into a format suitable for further processing and analysis.

3. Feedback Loop for Dynamic Error Correction

A feedback loop is incorporated to dynamically adjust any discrepancies or errors in the data, ensuring real-time calibration adjustments.

4. Data Logging and SCL Routines:

In industrial automation, data logging and Structured Control Language (SCL) routines play crucial roles in monitoring, analyzing, and optimizing process control systems. SCL is a high-level programming language used in PLCs, like Pascal or C. It enables complex calculations, looping structures, and conditional logic, which are difficult to achieve with standard Ladder Logic.

Each "tag" in these diagrams represents a memory location or signal identifier in the PLC program. These tags are classified into three main types, which are detailed below:

1. Input Tags (I)
   1) Purpose: Represent real-time input signals from field devices.
   2) Example Tags: I01, I02, I03, I04: These tags store incoming voltage or digital status values from various channels. They represent the analog or digital inputs being read by the PLC from the GICS card.
2. Output Tags (Q)
   1) Purpose: Represent Store values used to control actuators or output devices.
   2) Example Tags: O03 and O04: These tags carry the scaled output values after calibration adjustments are made. They represent the corrected and adjusted signals that are used to control system behavior or communicate with the GICS Emulation Card.
3. Memory/Variable Tags (M, DB, or Internal)
   1) Purpose: Store intermediate or processed data within the PLC.
   2) Example Tags:
      a. ControlV: Represents control voltages that are used for system regulation.

b. CalCtrlV: Represents calibration control voltages that ensure accurate signal processing during calibration.

These memory tags are crucial for holding intermediate values and enabling smooth communication between different sections of the ladder diagram.

1. PLC's Tag

PLC tags

	Name	Tag table	Data type
1	I01	Default tag table	Int
2	I02	Default tag table	Int
3	I03	Default tag table	Int
4	I04	Default tag table	Int
5	O04	Default tag table	Int
6	O03	Default tag table	Int
7	M01	Default tag table	Int
8	M02	Default tag table	Int
9	M03	Default tag table	Int
10	M04	Default tag table	Int
11	rOut	Default tag table	Int
12	OutputRes	Default tag table	Real
13	ControlRes	Default tag table	Real
14	OutputV	Default tag table	Real
15	ControlV	Default tag table	Real
16	Vinput	Tag table_Calibration	Real
17	ViF	Tag table_Calibration	Real
18	Offset	Tag table_Calibration	Real
19	ViC	Tag table_Calibration	Real
20	CalCtrlV	Default tag table	Real
21	Calibration	Tag table_Calibration	Bool
22	Vadjust	Tag table_Calibration	Real

Figure 3.8 Tag Assignment (Ladder Diagram – PLC's Tag)

From figure 3.8 (Tag Assignment in Ladder Diagram), the initial stage of the ladder diagram involves assigning raw input data from the GICS Emulation Card to specific PLC tags. In this stage, the MOVE (MOV) instructions are used to transfer physical signal data into dedicated input registers, such as I01, I02, I03, and I04. These registers capture analog or digital readings directly from the hardware.

This assignment is crucial because it ensures that each incoming signal is mapped to a unique identifier, laying the foundation for all subsequent processing steps. By storing the raw data in these tags, the system maintains the integrity of the information before any transformation or calibration takes place. These input tags are referenced throughout the calibration process, playing an essential role in real-time control and troubleshooting.

The list of PLC tags below shows how the tags are categorized based on their type and associated data types:

1. Input Tags (I)

1) Purpose: These tags represent integer values coming from the GICS Emulation Card or external sensors. They store the raw data before processing.

2) Tags: I01, I02, I03, I04: These are the input tags that represent the incoming data from the system. They are used to receive data such as signals or readings that will be processed in the system.

2. Memory Tags (M)

1) Purpose: These tags store integer values related to the system's internal calculations or operational status.

2) Tags: M01, M02, M03, M04: These memory tags are used to store temporary variables or states for internal system operations.

3. Control Tags

1) Purpose: These tags are used for managing the system's control logic and calibration settings.

2) Tags:

a. Calibration (Boolean): This Boolean tag likely controls whether the calibration process is active (True) or inactive (False). It is used to trigger the calibration routine.

b. OutputRes, OutputV (Real type): These are output tags that store real values (floating-point data) corresponding to the results or calculations. These values are typically sent to output devices or used for further control.

c. ControlRes, ControlV (Real type): These tags represent control values that can be used to send control signals to different components of the system based on processed data.

d. CalcCtrlV (Real type): This tag stores a calculated control value, which helps adjust or control outputs based on the processing results.

e. Vinput (Real type): Another real type of tag used to store input voltage or other continuous variables for further processing.

4. Filtering and Calibration Tags (Real type)

1) Purpose: These tags are associated with the filtering and calibration process, helping to refine the system's output.

2) Tags:

a. ViF, ViC: These real tags store values for low filtering calculations or coefficients used in the filtering algorithm.

b. Offset: This tag stores the offset value for the low pass filter, representing the adjustment applied to the filtering process.

c. Vadjust: These real tags store the adjustment values for the low pass filter, helping fine-tune its filtering behavior during calibration.

2. Signal Processing and Control System

This section describes the methodology used to process and adjust input signals in the PLC-based control system. The system utilizes ladder diagram instructions, which are translated into mathematical formulas for signal filtering, calibration, error correction, and feedback control. The process begins with the acquisition of raw data, followed by various mathematical operations that ensure the accuracy and stability of the output signal.

1) Data Acquisition and Tagging

The first step in the system is the acquisition of raw data from external sources (such as sensors or other devices). These input signals are assigned to specific PLC tags using the MOVE (MOV) instruction in the ladder diagram.

Mathematical Formula:

$$\text{Tag} = \text{Register Value}$$

Where:

a. The input registers (e.g., IW8, IW10, IW12) represent raw data from the system or external sensors.

b. The memory registers (MW8, MW10, MW12) temporarily store this raw data for subsequent processing.

For Example:

$$I01 = M01$$

This means that the I01 input value is stored in the PLC memory, channel M01.

Explanation:

The MOV operation is a straightforward data transfer mechanism that ensures raw input values are captured and stored for processing. This operation does not alter the data but prepares it for further manipulation by placing it into specific tags (memory locations). The raw data is now available for calibration, filtering, and error correction.

2) Scaling Data with Multiplication (MUL)

Once the data is acquired and stored, it is often necessary to scale the input values to match the system's operational parameters. The Multiplication (MUL) operation is used to scale or adjust the data.

Mathematical Formula:

$$\text{Output}=\text{IN1}\times\text{IN2}$$

Where IN1 and IN2 are the input values, and the result is stored in the output register.

For example:

$$\text{OutputV} = \text{I01} \times \text{OutputRes}$$

This operation adjusts the input signal (I01) by multiplying it by the scaled value (outputRes) to produce the final output (outputV).

Where:

a. I01 is an input signal, stored in a memory register like MW8, representing a value read from a sensor or external device.

b. OutputRes is a previously computed result (e.g., from a division) that acts as a scaling factor.

The value of OutputRes is equal to

$$\text{OutputRes} = \frac{20.0}{55296.0}$$

Explanation:

The multiplication operation adjusts the raw input signal (I01) based on a scaling factor (OutputRes). This operation is essential for converting the data to a suitable range for further processing, ensuring that the input matches the expected value format for the system.

3) Normalizing Data with Division (DIV)

To ensure that the signal is within a standardized range, Division (DIV) is used to normalize the input value. This ensures that all data processed within the system is consistent with the desired output range.

Mathematical Formula:

$$\text{ControlRes} = \frac{\text{IN1}}{\text{IN2}}$$

Where IN1 is the numerator and IN2 is the denominator.

For example:

$$\text{ControlRes} = \frac{Constant}{Scaling\ Factor}$$

$$\text{ControlRes} = \frac{55296.0}{20.0}$$

This operation normalizes the value by dividing 55296.0 by 20.0, producing the outputRes.

Explanation:

The division operation is used to scale the data down to match the hardware's capabilities. In this case, the values 55296.0 and 20.0 are chosen to produce a scale output that is within the required operating range. This ensures that the input values are compatible with the output interface, such as actuators or display units.

4) Error Calculation with Subtraction (SUB)

Once the data is processed, the system calculates any discrepancies between the desired output and the actual measured output. This is accomplished through the Subtraction (SUB) operation, which is crucial for the feedback loop.

Mathematical Formula:

$$\text{Error} = \text{DesiredOutput} - \text{ActualOutput}$$

This computes the error between the expected (desired) output and the measured (actual) output.

For example:

$$\text{ControlV} = 10.0 - \text{OutputV}$$

Here, 10.0 represents the desired output, and outputV is the actual output. The result, controlV, will be used to adjust the system.

Explanation:

The subtraction operation computes the error by comparing the expected value (DesiredOutput) with the actual value (OutputV). This error is then used to adjust the system's behavior in real time, ensuring that the output stays within the desired range. If there is any discrepancy, the system will adjust accordingly, maintaining control over the output signal.

5) Control Adjustment with Addition (ADD)

After calculating the error, the system uses Addition (ADD) to fine-tune the control signal. This adjustment compensates for minor errors or fluctuations that might occur due to drift, noise, or system disturbances.

Mathematical Formula:

$$\text{Output} = \text{IN1}+\text{IN2}$$

Where IN1 and IN2 are the input values, and the result is stored in the output register.

For example:

$$\text{CalCtrlV} = \text{ControlV} + 1.65$$

Here, the ControlV is adjusted by adding a constant value (1.65), and the result is stored in CalCtrlV.

Explanation:

a. The addition operation introduces a constant value (e.g., 1.65) to the control signal (ControlV). This compensation ensures that the system remains stable, preventing small deviations from becoming larger issues.

b. The result, CalcCtrlV, represents the adjusted control value that is then used for further processing or output.

6) Mathematical Formula for Second Multiplication

After calculating use of Division (DIV) and Multiplication (MUL) operations within a ladder diagram to process control values in a system. These operations are applied to scale and adjust input data, ensuring the output signals are appropriately adjusted to meet system requirements. The figure shows a sequence of operations where a division is followed by two multiplication steps to generate control outputs.

Mathematical Formula:

$$OO3 = ControlV \times ControlRes$$

This setup ensures that the control value (ControlV) is adjusted according to a scaled factor, allowing the system to output the desired signal correctly.

Explanation:

a. The value ControlV is calculated in the feedback loop from a previous subtraction operation, representing the difference between the desired output and the actual output.

b. This error value is then multiplied by ControlRes to adjust the error signal (ControlV) by the previously calculated scaling factor. The result is stored in OO4, which is a modified version of the control value that is more appropriate for use in the system's outputs.

c. By multiplying the error value with ControlRes, we ensure that the adjustment made to the error signal is proportionate to the required scaling factor.

7) Error Handling and Preventing Negative Values

To maintain data integrity, any negative values are reset to zero. This prevents erroneous or invalid values from propagating through the system.

Mathematical Formula:

$$V\_adjusted = max (0, V\_adjusted)$$

Explanation:

a. Negative values are invalid in this context because they could indicate an error in the system or sensor malfunctions.

b. The formula ensures that if V_adjusted is negative, it is reset to zero. This step is crucial for ensuring the reliability and stability of the system.

8) Database Logging for Long-Term Data Storage

Finally, all processed data is logged into a database for long-term storage, diagnostics, and performance analysis. This helps track trends over time and supports troubleshooting efforts when required.

Mathematical Formula:

$$Tag (database) = Processed\ Data$$

Explanation:

The processed values (e.g., V_calibrated, V_adjusted) are stored in a database or data logger. This enables the system to maintain a historical record of its behavior, which can be used for analysis, troubleshooting, or further system optimization.

3. Database's Tag

db_Filtered

	Name	Data type
1	Static	
2	Alpha	Real
3	V_input	Real
4	V_previousFiltered	Real
5	V_filtered	Real
6	Offset	Real
7	V_calibrated	Real
8	V_adjusted	Real
9	CalibrationFlag	Bool
10	FilterInitialized	Bool
11	ErrorFlag	Bool

Figure 3.9 Database Tagging and Data Logging

From Figure 3.9 In this section, the focus shifts from real-time signal processing to long-term storage and analysis of system data. The diagram illustrates how the PLC maps processed input, output, and calibration data to specific database tags. By doing so, every transaction and adjustment made in the system is logged for future reference.

Mathematical Operations:

The raw input values and calculated values (such as ControlV and CalCtrlV) are tagged and stored in the database for historical tracking.

Explanation:

1) Historical Tracking: Keeping a record of raw inputs, computed control voltages (ControlV), and calibration control voltages (CalCtrlV) supports ongoing diagnostics and performance evaluations.

2) Diagnostics and Trend Analysis: Archiving this data helps in tracking the system’s behavior over time, identifying trends, and enabling continuous improvement through data-driven adjustments.

Tag Descriptions:

1) Static: Represents constant values used in filtering processes. These constants may serve as fixed parameters or references.

2) Alpha: A constant (often a smoothing factor) used in filtering calculations to determine the weighting between new input data and previous filtered values.

3) V_input: The raw signal data as captured from the sensor or input source before any processing is applied.

4) V_previousFiltered: Stores the previously filtered data. This value is used in recursive filtering algorithms where past outputs influence the current filtered result.

5) V_filtered: The result of the filtering process. This tag holds the signal data after applying the filtering algorithm.

6) Offset: The offset value calculated during filtering. It is used to adjust the signal, often compensating for systematic biases.

7) V_Calibrated: The signal after calibration adjustments. This value has been corrected to account for systematic errors or biases, providing a more accurate measurement.

8) V_adjusted: The signal after additional adjustments beyond calibration. These may include scaling, normalization, or other modifications tailored to specific application needs.

9) Calibration Flag: A Boolean flag indicating whether calibration is active (true) or inactive (false), controlling the application of calibration corrections.

10) FilterInitialized: A Boolean tag that indicates if the filter has been properly initialized, ensuring that all subsequent filtering operations have a valid starting state.

11) ErrorFlag: A Boolean flag used to signal that an error has been detected during the filtering process, which can trigger error handling or alerts.

By using these tags, the system ensures that each data point, such as the input signal, filtered output, and calibration values, are accurately logged and available for further analysis.

4. SCL Code

```
"db_Filtered".V_input := "OutputV";
"db_Filtered".Alpha := 0.165;
"db_Filtered".V_previousFiltered := "db_Filtered".V_filtered;
"db_Filtered".V_filtered := "db_Filtered".Alpha * "db_Filtered".V_input +
(1.0 - "db_Filtered".Alpha) * "db_Filtered".V_previousFiltered;
"db_Filtered".Offset := 0.0 - "db_Filtered".V_filtered;
"db_Filtered".V_calibrated := "db_Filtered".V_filtered + "db_Filtered".Offset;
"db_Filtered".V_adjusted := "db_Filtered".V_filtered;
```

Figure 3.10 SCL Code for Low Pass Filtering Functionality

From figure 3.10 This section explains the SCL code responsible for applying low pass filtering techniques. The code processes input data and applies calibration adjustments, ensuring accurate measurements.

Explanation of Key Code Operations:

1) Assigning the Input Value:

This step takes the value from the variable OutputV and assigns it to the V_input field of the db_Filtered data structure. This is setting up the latest data or signal input that will be filtered in the subsequent steps.

2) Setting the Filter Constant (Alpha):

The filter constant Alpha is set to 0.165. In the context of a low-pass or exponential moving average filter, this value determines the weighting of the new input relative to the previous filtered value. A smaller alpha (closer to 0) means the filter will be less responsive to rapid changes.

3) Storing the Previous Filtered Value:

Before updating the filtered value, this line saves the current filtered value (V_filtered) into the V_previousFiltered variable. This is necessary because, in the next calculation, we need the previous filtered value to compute the new filtered result.

4) Calculating the New Filtered Value:

This line implements the filtering algorithm. It computes a new filtered value by taking a weighted average of the new input (V_input) and the previous filtered value (V_previousFiltered).

a. The weight for the new input is Alpha (0.165).

b. The weight for the previous filtered value is (1 - Alpha) (0.833).

c. This equation is typical for an exponential smoothing or low-pass filter.

5) Calculating the Offset for Calibration:

a. This line calculates the Offset, which represents the difference between zero and the filtered value. The offset might be used to calibrate or correct the filtered signal. If the signal needs to be shifted to a specific range (like non-negative values), the offset can help adjust that.

b. If the filtered value (V_filtered) is negative (indicating an invalid result), this logic could reset those negative values to zero. This part of the operation could include an ErrorFlag to indicate when the result is invalid, although that specific part is not shown in your code snippet. The offset helps ensure that the final filtered value stays within a valid range.

6) Setting the Adjusted Value:

a. Finally, the V_adjusted field is set equal to V_filtered. In this context, it seems that no further adjustments or corrections are applied beyond the filtering process. V_adjusted might be used later in the program as the final value or might serve as a placeholder for additional processing steps.

Overall, the code implements a simple low-pass filter (exponential moving average) to smooth the input signal, then applies an offset to calibrate (or zero out) the measurement, and finally prepares the adjusted value for further use.This section of the thesis covers the long-term storage and analysis of system data via database tagging, focusing on how data such as raw inputs, filtered outputs, and calibration values are tagged and stored. It also details the SCL code responsible for Low pass filtering, calibration, and error handling processes. These steps ensure the system's accurate and efficient operation, while also preventing errors such as negative values from affecting performance.

CHAPTER 4

EXPERIMENTS AND RESULT

4.1 Experiment

The data obtained prior to the test showed that the output from the GICS emulation card was occasionally inconsistent, a condition likely caused by factors such as voltage fluctuations. The focus of the experiment was to determine the appropriate offset value, which is crucial for stabilizing the displayed readings. Stabilizing the output is essential for accurate calibration, as calibration involves comparing measured values to a reference standard and adjusting the outputs to ensure they align with the expected reference values. For this experiment, the GICS emulation card output is set within a 10-bit range, spanning from -1023 to 1023. In contrast, the input values transmitted from the GICS emulation card to the PLC are in a 16-bit format, ranging from -27,648 to -27,648. Similarly, the output from the PLC is also in a 16-bit format, covering the same range of -27,648 to 27,648. Additionally, the values received by the GICS emulation card from the PLC are in a 12-bit format, ranging from -4095 to 4095. The communication voltage between the GICS emulation card and the PLC is calibrated to range between -10V and 10V. By accurately identifying and applying the necessary offset value, the experiment aims to achieve more stable readings, thereby ensuring the calibration process produces precise and reliable outputs. This approach enhances both the accuracy of the data and the consistency of performance in alignment with predefined reference values. Effective calibration is critical to ensure the system operates reliably in practical applications, improving the overall functionality of the PLC and GICS emulation card.

4.2 Result

4.2.1 Database

The database for the GICS Emulation Card must be structured to efficiently manage and store all critical data related to its operation, communication with the PLC, and calibration processes. It should include the following data types:

1. Analog Values

Raw data acquired from GICS testers, representing the analog signals received by the system.

2. Converted Digital Values

Data processed through the 10-bit ADC (Analog-to-Digital Converter), converting analog signals into digital signals for processing.

3. Processed Output Data

The digital signals sent to and received from the PLC represented in 16-bit and GICs represented in 12-bit formats.

Additionally, the database should store real-time offset values to improve the accuracy of data transmission and maintain traceability of the calibration routines. This data will also include any system adjustments made during operation to ensure optimal performance standards are maintained.

Representation of Analog Values in Voltage Measuring Ranges:

The analog values should be mapped to specific voltage ranges, ensuring consistency and measurement accuracy. The system must support:

1. Standard Voltage Ranges

0-5V, 0-10V, ±5V, and ±10V are the primary voltage ranges.

2. Resolution Considerations

Using the 10-bit ADC, each voltage range is converted into discrete digital steps to maintain precision.

3. Scaling Factors

Define scaling factors to accurately convert between analog input and digital representation.

Representation of Analog Value in Voltage Measuring Ranges.

This table shows the decimal and hexadecimal values corresponding to the possible voltage ranges of ±10V and ±5V. The values are categorized into different operating ranges such as overflow, overrange, nominal range, under range, and underflow.

Table 4.1 The Table Representation of Analog Value in Voltage Measurement Ranges

Values		Voltage measuring range		Ranges
Dec.	**Hex.**	**±10 V**	**±5 V**	
32767	7FFF	>11.759 V	>5.879	Overflow
32511	7EFF	11.759 V	5.879 V	Overrange
27649	6C01			
27648	6C00	10 V	5 V	Nominal range
20736	5100	7.5 V	3.75 V	
1	1	361.7 µV	180.8 µV	
0	0	0 V	0 V	
-1	FFFF			
-20736	AF00	-7.5 V	-3.75 V	
-27648	9400	-10 V	-5 V	
-27648	93FF			Under range
-32512	8100	-11.759 V	-5.879 V	
-32768	8000	-11.759 V	-5.879 V	Underflow

From table 4.1 it is essential for calibrating and scaling the data during the Analog-to-Digital Conversion (ADC) process. It also helps engineers diagnose issues and ensure precise readings are maintained, preventing erroneous data from affecting the automation and control processes in industrial environments.

Explanation of the Table:

1. Nominal Range

The range where the digital values correspond to standard operating voltages, such as 27648 mapping to 10V in the ±10V range and 5V in the ±5V range.

2. Overrange

Values that exceed the nominal range but are still within a tolerable system limit, e.g., 32511, which maps to 11.759V in the ±10V range and 5.879V in the ±5V range.

3. Overflow

Values exceeding the maximum allowed, such as 32767, resulting in overflow conditions where the system can no longer process the voltage correctly.

4. Under range & Underflow

When values fall below the operational limits, such as -32768, leading to potential loss of signal integrity and erroneous readings.

4.2.2 Data Collection

To conduct a thorough examination using the GICS tester, values ranging from 0 to 1023 will be input and transmitted to the GICS Emulation Card. This process involves selecting a set of five specific data values that represent a range of operational scenarios. By choosing these values, the goal is to evaluate the system’s performance across different conditions. The selected values include a mix of lower, middle, and higher values within the range, ensuring comprehensive testing of the system's ability to handle various input levels.

Table 4.2 The Table Representation of Test Data Values

The table representation of test data values
512
639
767
895
1023

Note. The values selected for testing are randomly chosen within the range of 512 to 1023. These values, spread across the lower, middle, and upper parts of the defined range, are as follows:

From table 4.2 By selecting these values, we ensure that the system is evaluated across various input levels, from the lower end (512) to the middle (639, 767) and higher end (1023) of the range. This comprehensive approach helps assess the system's performance and stability across a broad spectrum of input conditions. The random selection of values makes the test more representative of real-world conditions, where inputs are often unpredictable.

Explanation:

1. 512 (Minimum Value)

Represents the minimum input value, serving as the baseline for calibration. This value ensures that the system starts from a known reference point, helping to verify the accuracy of the lower input range.

2. 639 (Upper Mid-Range Value)

Serves as an upper mid-range value, providing insight into the system's performance as the input increases. This helps evaluate how well the system handles moderate input levels.

3. 767 (High Mid-Range Value)

A high mid-range value that ensures accurate and consistent calibration near the upper limit of the input range. It helps assess the system's stability in the middle-to-high input range.

4. 895 (Very High Value)

This very high value tests the system's capability to handle inputs near the maximum range. It evaluates the robustness of the system as the input approaches the upper boundary.

5. 1023 (Maximum Value)

The maximum input value, verifying the system's peak performance and ensuring it operates as expected at the upper limit of the input range.

Test Parameters and Range:

1. The GICS Emulation Card accepts 10-bit values, ranging from -1023 to 1023, corresponding to a voltage range of -10V to 10V.

2. The card cannot accept the value 1024, as this exceeds the input range and would result in incorrect readings. Therefore, 1023 is used as the maximum input value for this test to avoid errors and ensure proper functionality.

4.2.3 Data Representation of Analog-to-Digital Conversion

The following tables represent the conversion of various analog input values into their corresponding voltage measurement ranges. These conversions are crucial for accurate signal processing, calibration, and control within the system. Each table corresponds to a different input type, displaying how raw input values translate into voltage outputs across specific measuring ranges.

This table represents the conversion of Analog Input 01 values to voltage measurements.

Table 4.3 The table representation of analog input 01 to voltage measurement conversion

Input 01	**Voltage measuring range**				
	512	**639**	**767**	**895**	**1023**
I01	1100	8016	14970	21936	28896
OutputV	0.3979	2.8971	5.4145	7.9337	10.4518
ControlV	9.6021	7.1029	4.5855	2.0663	–0.4518
O03	26548	19639	12678	5713	–1250
O04	26548	19639	12678	5713	–1250
CalCtrlV	11.2521	8.7529	6.2355	3.7163	1.1982

From table 4.3 helps illustrate how raw analog input values are mapped to corresponding voltages and processed for use in the system.

1. Column Headers

The first row represents different measurement ranges (512, 639, 767, 895, and 1023), which indicate specific points in the analog-to-digital conversion process.

2. I01

Represents the raw analog input data before conversion.

3. OutputV

Shows the output voltage after processing the raw input data.

4. ControlV

Represents the voltage used for system control.

5. O03 & O04

These columns contain processed values related to specific control mechanisms in the system.

6. CalCtrlV:

Displays the calibration control voltage, ensuring that the system maintains accurate readings.

This table illustrates the conversion of Analog Input 02 values to voltage measurements.

Table 4.4 The Table Representation of Analog Input 02 to Voltage Measurement Conversion

Input 02	**Voltage measuring range**				
	512	**639**	**767**	**895**	**1023**
I02	1169	8106	15102	22096	29084
OutputV	0.4228	2.9319	5.4622	7.9919	10.5197
ControlV	9.5772	7.0681	4.5378	2.0081	–0.5197
O03	26479	19542	12546	5552	–1436
O04	26479	19542	12546	5552	–1436
CalCtrlV	11.2272	8.7181	6.1878	3.6581	1.1303

From table 4.4 follows a similar structure, showing the conversion of I02 values to their corresponding OutputV and ControlV voltages. CalCtrlV ensures that the system operates with accuracy, while O03 and O04 provide values that represent specific control mechanisms.

This table represents the conversion of Analog Input 03 values to voltage measurements.

Table 4.5 The table representation of analog input 03 to voltage measurement conversion

Input 03	Voltage measuring range				
	512	**639**	**767**	**895**	**1023**
I03	890	7812	14777	21743	28700
OutputV	0.3255	2.8255	5.3447	7.8642	10.3805
ControlV	9.6745	7.1745	4.6553	2.1358	–0.3805
O03	26748	19836	6.3053	5905	–1052
O04	26748	19836	12871	5905	–1052
CalCtrlV	11.3245	8.8245	12871	3.7858	1.2695

From table 4.5 demonstrates the conversion of I03 values into voltage outputs and shows the processed control values for each input level.

This table represents the conversion of Analog Input 04 values to voltage measurements.

Table 4.6 The Table Representation of Analog Input 04 to Voltage Measurement Conversion

Input 04	Voltage measuring range				
	512	**639**	**767**	**895**	**1023**
I04	1064	7972	14938	21908	28872
OutputV	0.3848	2.8834	5.4029	7.9239	10.4427
ControlV	9.6152	7.1166	4.5971	2.0761	–0.4427
O03	26584	19676	12710	5740	–1223
O04	26584	19676	12710	5740	–1223
CalCtrlV	11.2652	8.7666	6.2471	3.7261	1.2073

From table 4.6 demonstrates the conversion of I03 values into voltage outputs and shows the processed control values for each input level.

Explanation of Tables:

Each table maps raw analog input values (e.g., I01, I02, I03, I04) to their corresponding output voltages (OutputV) and control voltages (ControlV). The O03 and O04 columns represent processed values linked to specific control mechanisms, while CalCtrlV ensures accurate calibration.

These tables are crucial for:

1. System Calibration: Ensuring precise signal processing for accurate control.
2. Performance Testing: Evaluating how the system behaves across different input values.
3. Voltage Measurement Mapping: Understanding the relationship between raw analog inputs and the processed voltage measurements.

Each of these tables aids in calibrating the system, ensuring reliable operation by maintaining stability and accuracy in the control process.

4.2.4 Voltage Filtered (First Order) Processing and Calibration

In embedded systems and industrial automation, accurate voltage measurement is essential for reliable system performance. Raw output voltage values, however, often contain noise, offsets, and other inaccuracies that need to be corrected before they can be used for control and monitoring. The following tables illustrate the various stages of voltage processing for Input 01- Input 04, including filtering, offset correction, calibration, and final adjustment. These steps ensure that the measured voltage values are accurate and aligned with the system's reference standards.

This table shows the voltage processing stages for Input 01 across different measurement ranges:

Table 4.7 The Table Representation Voltage Filtered Processing and Calibration for Input 01

Input 01	Voltage measuring range				
	512	**639**	**767**	**895**	**1023**
Vi1 (InputV)	0.3979	2.8971	5.4145	7.9337	10.4518
V_filtered	0.3978	2.8969	5.4144	7.9335	10.4517
V_offset	- 0.3978	- 2.8969	- 5.4144	- 7.9335	- 10.4517
V_calibrated	0.0	0.0	0.0	0.0	0.0
V_adjusted	0.0	2.5	5.0	7.5	10.0

From table 4.7 it details the process of transforming raw analog input values (Vi1) for Input 01 into calibrated and adjusted voltage values through a series of processing stages.

This table shows the voltage processing stages for Input 02 across different measurement ranges:

Table 4.8 The table representation voltage filtered processing and calibration for Input 02

Input 02	Voltage measuring range				
	512	**639**	**767**	**895**	**1023**
Vi1 (OutputV)	0.4228	2.9319	5.4622	7.9919	10.5197
V_filtered	0.4226	2.9317	5.4621	7.9917	10.5196
V_offset	- 0.4226	- 2.9317	- 5.4621	- 7.9917	- 10.5196
V_calibrated	0.0	0.0	0.0	0.0	0.0
V_adjusted	0.0	2.5	5.0	7.5	10.0

From table 4.8, it details the process of transforming raw analog input values (Vi1) for Input 03 into calibrated and adjusted voltage values through a series of processing stages.

This table shows the voltage processing stages for Input 03 across different measurement ranges:

Table 4.9 The Table Representation Voltage Filtered Processing and Calibration for Input 03

Input 03	Voltage measuring range				
	512	**639**	**767**	**895**	**1023**
Vi1 (OutputV)	0.3255	2.8255	5.3447	7.8642	10.3805
V_filtered	0.3254	2.8254	5.3446	7.8641	10.3804
V_offset	- 0.3254	- 2.8254	- 5.3446	- 7.8641	- 10.3804
V_calibrated	0.0	0.0	0.0	0.0	0.0
V_adjusted	0.0	2.5	5.0	7.5	10.0

From table 4.9, it details the process of transforming raw analog input values (Vi1) for Input 03 into calibrated and adjusted voltage values through a series of processing stages.

This table shows the voltage processing stages for Input 04 across different measurement ranges:

Table 4.10 The Table Representation Voltage Filtered Processing and Calibration for Input 04

Input 04	Voltage measuring range				
	512	**639**	**767**	**895**	**1023**
Vi1 (OutputV)	0.3848	2.8834	2.8834	7.9239	10.4427
V_filtered	0.3846	2.8832	2.8832	7.9238	10.4425
V_offset	- 0.3846	- 2.8832	- 2.8832	- 7.9238	- 10.4425
V_calibrated	0.0	0.0	0.0	0.0	0.0
V_adjusted	0.0	2.5	5.0	7.5	10.0

From table 4.9, it details the process of transforming raw analog input values (Vi1) for Input 03 into calibrated and adjusted voltage values through a series of processing stages.

Provides a clear demonstration of how raw analog input values from Input 04 are processed and adjusted to produce accurate voltage outputs. Each step, from filtering to offset correction and calibration, ensures that the final adjusted voltage values are reliable and ready for system control. By following these stages, the system achieves accurate voltage measurements, which are essential for maintaining precise control and stable operation in industrial applications.

Summary of Processing Stages:

1. Vi1 (OutputV)

Represents the raw output voltage values from the system before any processing.

2. V_filtered

The filtered voltage, after applying low pass filtering to reduce noise and improve signal quality.

3. V_offset

The voltage offset applied to correct systematic measurement errors, such as environmental or sensor-induced biases.

4. V_calibrated

The calibrated voltage, which has been adjusted to align with standard reference values. This column is currently set to 0.0 in all cases, representing a baseline calibration.

5. V_adjusted

The final adjusted voltage after all processing steps (filtering, offset correction, and calibration) have been applied, ensuring accurate measurements for further system control and monitoring.

This process is crucial for transforming raw analog input values into accurate, reliable voltage outputs that are essential for system calibration, control, and stable operation in industrial automation systems. The filtered, offset-corrected, and calibrated voltages ensure that the system performs with high accuracy and precision, essential for effective control and monitoring in industrial settings.

CHAPTER 5

CONCLUSION

The integration of Programmable Logic Controllers (PLCs) and General Instrument Control System (GICS) emulation cards is essential for achieving high precision and stability in industrial control systems. Through experimental analysis, significant insights were gained regarding the role of calibration, particularly the application of an appropriate offset value, in ensuring accurate signal processing and system reliability. The research focused on developing an automatic calibration method to address offset discrepancies that could destabilize output values, ultimately affecting the communication between PLCs and GICS. During the experiment, the importance of incorporating an offset value became apparent. Without this calibration step, the processed output from the PLC exhibited instability, with fluctuations in values that often exceeded the expected range. For instance, when input values ranged from -27,648 at -10 volts to 27,648 at 10 volts, the output from the PLC consistently exceeded the anticipated 16-bit output of 28,000. This misalignment underscored the necessity of applying the correct offset to bring the output in line with the expected signal range. The desired target output from the GICS emulation card was set at a maximum of 1023, but failure to adjust for the offset resulted in values that were not only unstable but misrepresentative of the actual signal. Building on this observation, the research introduced an automatic calibration system using the SIEMENS PLC SIMATIC S7-1500 and TIA Portal software. This system was designed to compensate for signal offsets during the Analog-to-Digital Conversion (ADC) and Digital-to-Analog Conversion (DAC) processes, ensuring stable communication between the PLC and GICS. In a controlled experimental setup, predefined input signal ranges from the GICS emulation card were converted into digital form by the PLC, and any discrepancies between the expected and actual outputs were identified. The system then calculated the necessary offset and made real-time adjustments to correct these errors, thus ensuring signal accuracy. A main aspect of this calibration system was the use of the TIA Portal's Ladder Diagram programming feature, which facilitated the implementation of a feedback loop. This loop automatically monitored output values and adjusted the signals in real time to maintain alignment with predefined reference values. The use of low-pass filters further enhanced signal accuracy by reducing high-frequency noise and ensuring smooth communication between the systems. The results of the experiment demonstrated the effectiveness of the automatic calibration system in stabilizing output values. The system successfully reduced offset errors, with output values consistently aligning with expected values, minimizing discrepancies. The filtering techniques played a vital role in improving signal clarity, and the system's ability to adjust in real time reduced the need for manual intervention, streamlining the calibration process. In conclusion, the findings emphasize the importance of proper calibration practices, including the identification and application of offset values, in ensuring the accuracy and stability of output signals in industrial systems. The developed automatic calibration system, with its ability to compensate for offset discrepancies in real time, represents a significant advancement over traditional manual calibration methods. By incorporating automated feedback loops and filtering techniques, this approach enhances the precision and reliability of PLC-GICS communication, offering substantial improvements in industrial automation systems. The system's efficiency and effectiveness suggest it could be widely applied across various industrial applications, offering a promising solution for improving signal accuracy and system stability.

REFERENCES

1241-2000 - IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters. (2001). IEEE. https://ieeexplore.ieee.org/document/929859

Altaher, A., Mocanu, S., & Thiriet, J.-M. (2015). *Evaluation of time-critical communications for IEC 61850 -substation network architecture*. https://arxiv.org/pdf/1512.07004

AutomationDirect. (2004). *Implementing a PLC calibration routine to ensure accurate instrument readings*. https://library.automationdirect.com/implementing-a-plc-calibration-routine-to-ensure-accurate-instrument-readings-issue-2-2004/

Azam, H. D., Baturone, L., & Clemente, F. R. (2020). Calibration methods for high precision robot-assisted industrial automation process. In *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)* (pp. 1-8). IEEE.

Belter, D. S. (2013). *A multi-objective parameter calibration approach* (Master's thesis). University of Twente.

Bhaskar, B., Nagashyam, P., Priyanka, S., & Navya, P. (2016). Flow calibration using PLC and SCADA. *International Journal of Scientific & Engineering Research, 7*(3), 29-34.

Calado, J, S., Ferreira, J., Mendonça, J. P., & Jardim-Gonçalves, R. (2023). AI applications in the configuration and calibration of industrial machines. *IEEE Transactions on Industrial Informatics, 19*(3), 2156-2168. https://doi.org/10.1109/ICE/ITMC61926.2024.10794282

Chan, C.-H., Zhu, Y., Li, C., Zhang, W.=H., Ho, L.-M., & Wei, L. (2017). 60-dB SNDR 100-MS/s SAR ADCs with threshold reconfigurable reference error calibration. *IEEE Journal of Solid-State Circuits, 52*(10), 2576-2588. http://doi.org/10.1109/JSSC.2017.2728784

Cleveland State University. (2016). *PLC implementation of beverage routing control*. https://engineering.csuohio.edu/sites/default/files/media/ece/documents/Anusha_Yerroju.pdf

Erickson, K. T. (1996). Programmable logic controllers. *IEEE Potentials, 15*(1), 14-17. http://doi.org/10.1109/45.481370

Erlandsson, T., & Rahaman, M. M. (2013). *Testing and verifying PLC code with a virtual model* (Master's thesis). Chalmers University of Technology.

Fakih, M., Dharmaji, R., Moghaddas, Y., Araya, G. Q., Ogundare, O., & Al Faruque, M. A. (2024). LLM4PLC: Harnessing large language models for verifiable programming of PLCs in industrial control systems. In *the 46th International Conference on Software Engineering: Software Engineering in Practice; for demo website* (pp. 192-203). https://doi.org/10.1145/3639477.36397

Jadhav, A. B., & Patil, M. (2014). PLC based industrial automation system. In *International Conference On Recent Trends in Engineering and Management Science - RTEM* (pp. 19-23). Conference Locate.

Jayasinghe, S. L. (2018). *SCADA system for remote control and monitoring of grid connected inverters* (Master's thesis). Memorial University of Newfoundland.

Khaldi, M. K. (2019). *Analysis and design of a SCADA system* (Master's thesis). National School of Applied Sciences, École Supérieure en Sciences Appliquées de Tlemcen.

Lu, J. (2022). Application of PLC control technology in intelligent automatic control. In *2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*. IEEE.

McKoy, J. D. (2018). *SCADA and PLC systems configuration for the NCREPT test facility* (Master's thesis). University of Arkansas.

Morel, P., Richard, C., & Lefebvre, L. F. (2023). GICS: A generic interface card system for industrial system laboratory applications. In *CIRP 2023 - 56th CIRP Conference on Manufacturing Systems.* Oct 2023, Le Cap, South Africa (pp.1375-1380). https://doi.org/1016/j.procir.2023.09.179.hal-04098029

Open University. (2018). *The programmable logic controller: Its prehistory, emergence and application*. https://oro.open.ac.uk/54687/

Owen, E. (1975). The elimination of offset errors in dual-slope analog-to-digital converters. *IEEE Journal of Solid-State Circuits, 10*(6), 372-377. https://doi.org/10.1109/TCS.1980.1084784

Shabbir, Q. (2015). *Study and implementation of programmable logic controller bachelor of technology in electronics & instrumentation engineering* (Master's thesis). National Institute of Technology Rourkela.

Shedge, S., & Tade, S. L. (2019). Design of instruction list processor for industrial applications. In *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. IEEE.

Siemens. (2018). *Learn-/training document | TIA portal module 031-500*. https://www.automation.siemens.com/sce-static/learning-training-documents/tia-portal/basics-programming-s7-1200/sce-031-500-analog-values-s7-1200-r1709-en.pdf

Siemens. (2021). *SIMATIC S7-1500/ET 200MP automation system*. https://support.industry.siemens.com/cs/attachments/59191792/s71500_et200mp_system_manual_en-US_en-US.pdf

Szymczyk, C., Nieß, N., Breitenbach, J. N., & Bumiller, G. (2023). Calibration method for an on-line PLC blocking filter characterization system. In *2023 IEEE International Symposium on Power Line Communications and its Applications (ISPLC)*. IEEE

Toronto Metropolitan University. (2021). *A study of programmable logic controllers (PLC) in control systems for effective learning*. https://rshare.library.torontomu.ca/articles/thesis/A_Study_of_Programmable_Logic_Controllers_PLC_in_Control_Systems_for_Effective_Learning/14660973

Yao, K.-C., Lin, C.-L., & Pan, C.-H. (2024). Industrial sustainable development: The development trend of programmable logic controller technology. Sustainability, 16(14), 6230. https://doi.org/10.3390/su16146230

APPENDIX

LADDER DIAGRAM AND FUNCTION BLOCK DEVELOPMENT

Ladder diagram with TIA Portal program for real-time offset and auto calibration. The process of creating a ladder diagram for real-time offset and auto calibration using the TIA Portal program involves several key steps.

1. Creating a project file in the TIA PORTAL program which requires knowing the IP Address of the equipment including the PLC, GICS emulation card that sent from GICS emulation card.

From the experiment, the IP of each device will be used as follows:

IP uses of PLC is 10.1.29.191

IP uses of GICS emulation card 10.1.29.194

2. Open the TIA Portal program to create a new project then set up the equipment such as the type of machine.

The machine used in the experiment is PLC SIMATIC S7-1500.

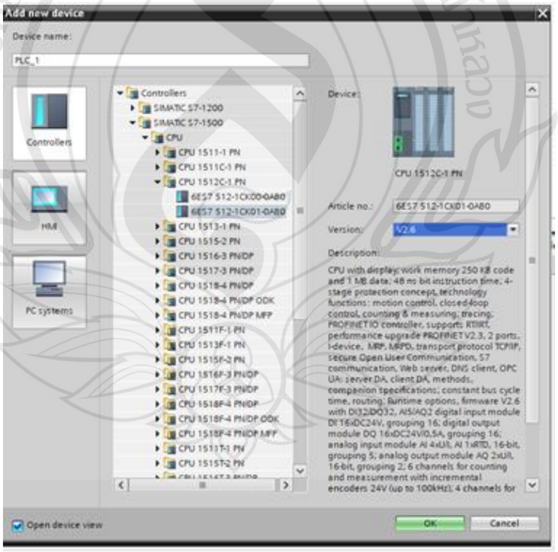
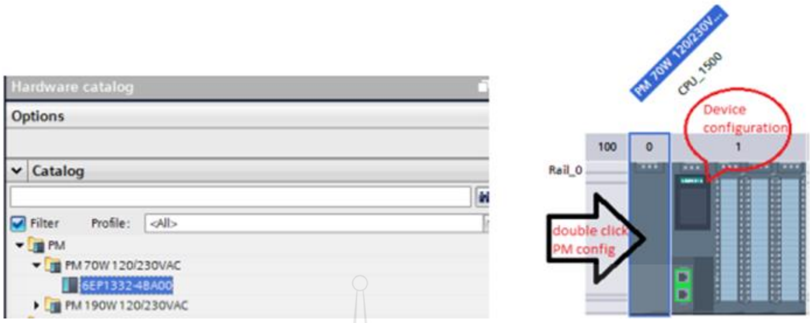


Figure A1 Insert a New Device and Create the CPU with a PLC SIMATIC S7-1500

3. The power module is not automatically detected and must be manually added from the physical PLC.



(a) Insert power Module (PM 70W 120:230V) (b)

Figure A2 Hardware Catalog

4. Configure the network at PROFINET interface and click Ethernet addresses then configure IP protocol.

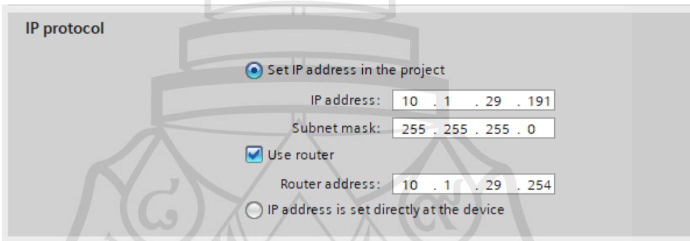


Figure A3 IP user of PLC

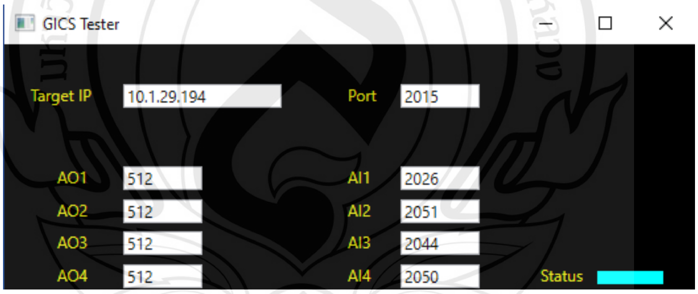


Figure A4 IP User of GICS Tester

5. After inserting a new device and creating the CPU, we must manually add the power module, as it is not automatically detected. The required power module, PM 70W 120:230V, can be added from the hardware catalog.

6. Next, the network configuration is set up through the PROFINET interface, where we configure the IP protocol. It's important to verify that the GICS Tester's IP address is correct, indicated by a blue status display; an incorrect IP address will show a red status display.

7. Then download the hardware configuration to the device and click go online that will show load preview of project.

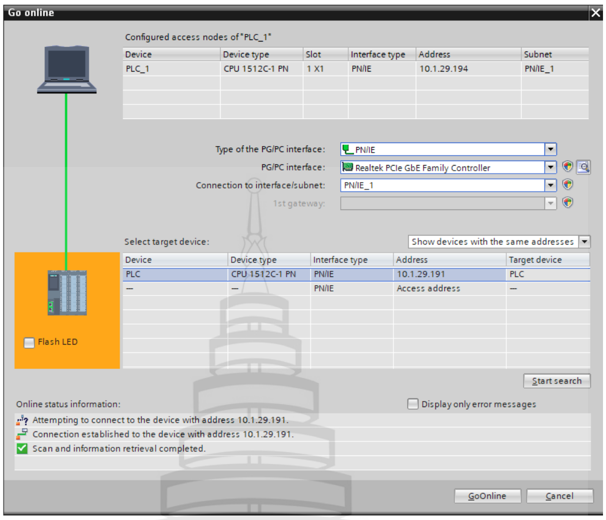


Figure A5 Hardware Detection

8. Once the hardware configuration is downloaded to the device, that can go online to view a load preview of the project. Following this, the input/output configuration is created within the Organization Block (programming block), which facilitates the connection between the PLC and GICS emulation card.

Figure A6 Organization Block

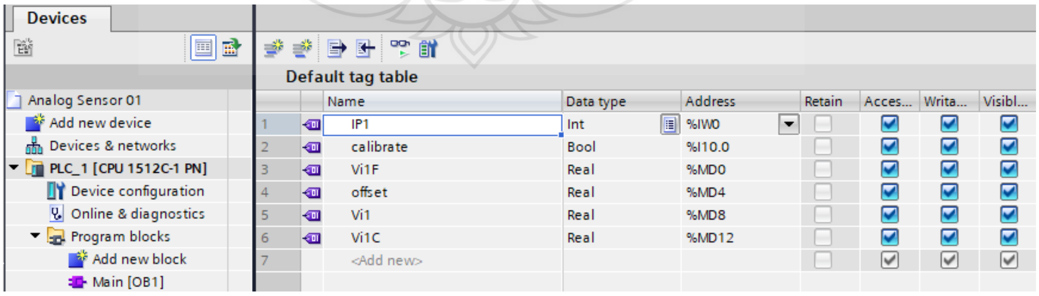


Figure A7 Input/Output Configuration

9. Create ladder diagram that reads database and filtering of inputs for finding offset values and automatically calibrate values on each input sent from GICS emulation card to PLC.

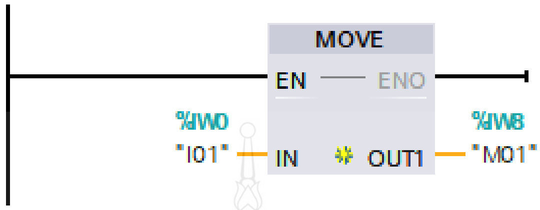


Figure A8 MOVE Instruction for Data Transfer

From figure A8 This Ladder Logic Diagram shows the use of MOVE (MOV) instructions to transfer data from input registers to output registers in a PLC program. The logic is structured to handle multiple input signals and map them to corresponding output locations.

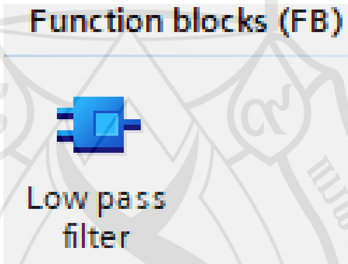


Figure A9 Function Block

From Figure A9 the function block is used in conjunction with two important Data Blocks (DB): db_Filtered. These data blocks play a vital role in the PLC program, as they are used to store the processed values related to filtered output after applying the low pass filtering technique.

10. To implement this, the Data Block (DB) is created for storing the filtered outputs from both filtering methods. The db_Filtered data block holds values such as the filtered voltage, offset, calibrated voltage, and adjusted voltage after the low pass filtering process. These data blocks allow the system to store, organize, and compare the results of both filtering techniques.

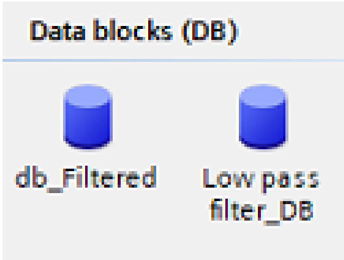


Figure A10 Data Block

From Figure A10 likely presents the detailed configuration of the Data Block, showcasing how the tags are assigned to db_Filtered, which are then used by the PLC to manage the filtered and calibrated values for the system's operation.

11. Create an SCL (Structured Control Language) code to handle the core calibration logic of the system. The process begins by initializing variables for low-pass filtering, including the Alpha coefficient used for filtering. The system then stores previous values from Low pass filter in the db_Filtered data block, ensuring that the necessary historical data is available for processing the next input signal. The filtering process starts by applying a low-pass filter to the raw input signal, which combines the current input with the previously filtered value to reduce high-frequency noise, resulting in smoother and more stable data.

```
"db_Filtered".V_input := "OutputV";

"db_Filtered".Alpha := 0.165;

"db_Filtered".V_previousFiltered := "db_Filtered".V_filtered;

"db_Filtered".V_filtered := "db_Filtered".Alpha * "db_Filtered".V_input +
(1.0 - "db_Filtered".Alpha) * "db_Filtered".V_previousFiltered;

"db_Filtered".Offset := 0.0 - "db_Filtered".V_filtered;

"db_Filtered".V_calibrated := "db_Filtered".V_filtered + "db_Filtered".Offset;

"db_Filtered".V_adjusted := "db_Filtered".V_filtered;

IF "db_Filtered".V_filtered < 0.0 THEN
    "db_Filtered".V_filtered := 0.0;
    "db_Filtered".ErrorFlag := TRUE;
END_IF;

IF "db_Filtered".V_calibrated < 0.0 THEN
    "db_Filtered".V_calibrated := 0.0;
    "db_Filtered".ErrorFlag := TRUE;
END_IF;

IF "Calibration" AND NOT "db_Filtered".CalibrationFlag THEN
    "db_Filtered".Offset := 0.0 - "db_Filtered".V_input;
    "db_Filtered".V_calibrated := "db_Filtered".V_filtered + "db_Filtered".Offset;
    "db_Filtered".V_adjusted := "db_Filtered".V_filtered;
    "db_Filtered".CalibrationFlag := TRUE;
END_IF;

IF NOT "Calibration" THEN
    "db_Filtered".CalibrationFlag := FALSE;
END_IF;
```

Figure A11 SLC Code

```
IF "db_Filtered".V_adjusted > 10.0 THEN
    "db_Filtered".V_adjusted := 10;
ELSIF "db_Filtered".V_adjusted > 9.0 THEN
    "db_Filtered".V_adjusted := 9;
ELSIF "db_Filtered".V_adjusted > 8.0 THEN
    "db_Filtered".V_adjusted := 8;
ELSIF "db_Filtered".V_adjusted > 7.0 THEN
    "db_Filtered".V_adjusted := 7;
ELSIF "db_Filtered".V_adjusted > 6.0 THEN
    "db_Filtered".V_adjusted := 6;
ELSIF "db_Filtered".V_adjusted > 5.0 THEN
    "db_Filtered".V_adjusted := 5;
ELSIF "db_Filtered".V_adjusted > 4.0 THEN
    "db_Filtered".V_adjusted := 4;
ELSIF "db_Filtered".V_adjusted > 3.0 THEN
    "db_Filtered".V_adjusted := 3;
ELSIF "db_Filtered".V_adjusted > 2.0 THEN
    "db_Filtered".V_adjusted := 2;
ELSIF "db_Filtered".V_adjusted > 1.0 THEN
    "db_Filtered".V_adjusted := 1;
ELSE
    "db_Filtered".V_adjusted := 0;
END_IF;

IF NOT "db_Filtered".FilterInitialized THEN
    "db_Filtered".V_filtered := 0.0;
    "db_Filtered".V_previousFiltered := 0.0;
    "db_Filtered".FilterInitialized := TRUE;
END_IF;

IF "OutputV" < 0.0 THEN
    "db_Filtered".V_input := 0.0;
ELSE
    "db_Filtered".V_input := "OutputV";
END_IF;

"Vinput" := "db_Filtered".V_input;
"ViF" := "db_Filtered".V_filtered;
"ViC" := "db_Filtered".V_calibrated;
"Offset" := "db_Filtered".Offset;
"Vadjust" := "db_Filtered".V_adjusted;
```

Figure A12 SLC Code

From Figure A11 illustrates the Function Block created to implement this calibration logic. This block processes the input signals, applies the necessary filters, and adjusts the values as required. The function block is designed to handle low pass filtering, with each stage being stored in separate data blocks (db_Filtered). These data blocks allow the system to organize and compare the results of both filtering techniques, ensuring that the output remains stable, accurate, and dynamically adjusted as new data is received.

12. Following this, the system applies real-time calibration by calculating and applying an offset to the filtered signal to compensate for any measurement errors. This offset correction ensures that the filtered values are accurate and aligned with the

system's reference standards. The processed values, including the raw input, filtered values, offset corrections, and adjusted outputs, are then stored for continuous monitoring. These values are saved in the db_Filtered data block, allowing the system to access and use them in real-time for control and adjustment purposes.

CURRICULUM VITAE

NAME	Thunwalai Thongkong
EDUCATIONAL BACKGROUND	
2020	Bachelor of Science Information Technology Mae Fah Luang University
WORK EXPERIENCE	
2019	Internship National Electronics and Computer Technology Center (NECTEC)