e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:04/April-2024 Impact Factor- 7.868 www.irjmets.com

PLCS IN INDUSTRIAL AUTOMATION

Shivaji Ambadas Salunkhe*1

*1Department Of Electronics And Communication Engineering, Lingayas Vidyapeeth, Faridabad, Haryana, India.

DOI : https://www.doi.org/10.56726/IRJMETS52934

ABSTRACT

This paper focuses to provide an overview of PLC based technology in industrial automation. PLCs (Programmable Logic Controllers) are most widely used to control the machines and automate processes in Industries. Various terminology related to PLCs and their programming as per IEC 61131-3 standard is explained in this paper.

Keywords: Industrial Automation, PLC.

I. INTRODUCTION

Programmable Logic Controllers (PLCs) are essential components in modern industrial automation. These specialized digital computers streamline control processes by executing user-programmed logic. With their ability to interface with sensors, switches, and actuators, PLCs offer flexibility and efficiency in managing diverse manufacturing tasks. Despite their compact design, PLCs have evolved to incorporate advanced features such as networking capabilities and support for multiple programming languages. As industries embrace digital transformation, PLCs play a crucial role in optimizing production processes and driving innovation.

II. COMMON TERMS AND FACTORS FOR SELECTION OF PLC

Factors to check during selection of PLC:

Application Requirements: Analyze the specific control requirements, including the complexity of control algorithms, scan time constraints, I/O types and counts, and communication protocols needed for interfacing with other devices or systems. This assessment guides the selection of a PLC model with the appropriate processing power, memory capacity, and I/O capabilities.

Performance Specifications: Consider the processing speed, cycle time, and scan time of the PLC to ensure that it can handle the required computational tasks within the specified time constraints. Evaluate the memory capacity for program storage and data handling, ensuring sufficient resources for storing complex logic and handling large datasets.

Environmental Conditions: Assess the environmental factors such as temperature, humidity, shock, vibration, and electromagnetic interference (EMI) present in the installation environment. Select a PLC model with suitable environmental ratings (e.g., IP ratings) and protective features (e.g., conformal coating, shock-resistant construction) to withstand harsh operating conditions and ensure reliable performance.

Integration and Compatibility: Verify compatibility with existing hardware and software components in the automation system, including sensors, actuators, HMI devices, and networking infrastructure. Evaluate support for industry-standard communication protocols (e.g., Ethernet/IP, Modbus, Profibus) and consider the availability of compatible software tools for programming, configuration, and monitoring.

Programming and Software Tools: Assess the PLC's programming capabilities and software development environment. Consider support for programming languages (e.g., ladder logic, structured text, function block diagrams), debugging tools (e.g., online monitoring, simulation), and advanced features (e.g., multitasking, user-defined function blocks) to facilitate efficient development and maintenance of control logic.

Reliability and Maintenance: Evaluate the reliability and serviceability features of the PLC hardware and software. Look for redundant power supplies, CPU modules, and I/O modules to minimize the risk of system downtime due to component failures. Consider built-in diagnostic capabilities, remote monitoring features, and comprehensive documentation to facilitate troubleshooting and preventive maintenance activities.

Cost and ROI: Conduct a cost-benefit analysis to evaluate the total cost of ownership (TCO) of the PLC solution. Factor in the initial purchase cost, installation expenses, training costs, ongoing maintenance expenses, and potential downtime costs associated with system failures. Select a PLC model that offers the best balance of performance, reliability, and affordability, considering the long-term benefits and return on investment (ROI) for the application.

COMMON TERMINOLOGY

Task cycle times:

Each scan cycle includes writing the outputs, reading the inputs, executing the user program instructions, and performing system maintenance or background processing.

Operating modes of the CPU:

The CPU has three modes of operation: STOP mode, STARTUP mode, and RUN mode. Status LEDs on the front of the CPU indicate the current mode of operation. In STOP mode, the CPU is not executing the program, and you can download a project. In STARTUP mode, the CPU executes any startup logic (if present). The CPU does not process interrupt events during the startup mode. In RUN mode, the scan cycle executes repeatedly. Interrupt events can occur and the CPU can process them at any point within the program cycle phase. You can download some parts of a project in RUN mode.

III. IEC 61131-3 FOR PLCS

PLC programming, as defined by the IEC 61131-3 standard, adheres to a set of rules and guidelines aimed at standardizing the development process and ensuring interoperability between different PLC platforms. Here's an explanation of PLC programming within the framework of the IEC 61131-3 standard, using technical terms:

PROGRAMMING LANGUAGES:

The IEC 61131-3 standard defines several programming languages for PLC development, including:

Ladder Diagram (LD): Represents logic control using graphical symbols resembling relay logic circuits, suitable for representing sequential control and discrete logic.

Structured Text (ST): Offers a high-level programming language resembling traditional programming languages like Pascal or C, enabling complex algorithmic control and mathematical operations.

Function Block Diagram (FBD): Utilizes graphical blocks to represent reusable functions or algorithms, facilitating modular programming and code reusability.

Sequential Function Chart (SFC): Represents control logic using graphical state diagrams, useful for representing sequential control and complex state-based behavior.

Instruction List (IL): Represents control logic using a textual list of instructions resembling assembly language, suitable for low-level programming and direct control of PLC resources.

PROGRAM ORGANIZATION UNITS (POUS):

PLC programs are structured into different types of Program Organization Units (POUs), including:

Main Program (POU): The primary control logic executed cyclically by the PLC, typically containing the main control algorithm.

Function (POU): A reusable block of code encapsulating a specific functionality or algorithm, facilitating modular programming and code reuse.

Function Block (POU): A reusable block of code encapsulating a specific function or algorithm, represented as a graphical block in FBD programming.

Function Block Instance: An instance of a function block within a PLC program, representing the instantiation of a reusable function block.

DATA TYPES:

The standard defines various data types for representing different types of data in PLC programs, including:

Boolean: Represents binary states (true/false).

Integer: Represents signed whole numbers.

Real: Represents floating-point numbers.

String: Represents text data.

Arrays and Structures: Composite data types composed of multiple elements of primitive data types.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:04/April-2024 Impact Factor- 7.868 www.irjmets.com

VARIABLE TYPES:

Global, Direct (local), I/O Mapping – Input, Output, I/O, External, Temporary

PROGRAM EXECUTION MODEL:

PLC programs follow a cyclic execution model, where the control logic is executed repeatedly in a continuous loop. The execution cycle typically consists of:

Input Scan: Reading input values from physical I/O devices.

Program Execution: Executing the control logic according to the program's structure and algorithm.

Output Update: Updating output values based on the results of program execution.

Communication Tasks: Handling communication tasks such as data exchange with other devices or systems.

Diagnostic Tasks: Performing diagnostic checks and error handling routines.

DEVELOPMENT ENVIRONMENT:

PLC programming environments conforming to the IEC 61131-3 standard typically provide tools and features for:

Online Monitoring: Real-time monitoring of PLC program execution and variable values.

Debugging: Tools for debugging and troubleshooting PLC programs, including breakpoints, watch variables, and trace functionality.

Simulation: Simulating PLC programs offline to verify behavior and test logic before deployment.

Documentation: Generating documentation for PLC programs, including function block libraries, program descriptions, and variable lists.

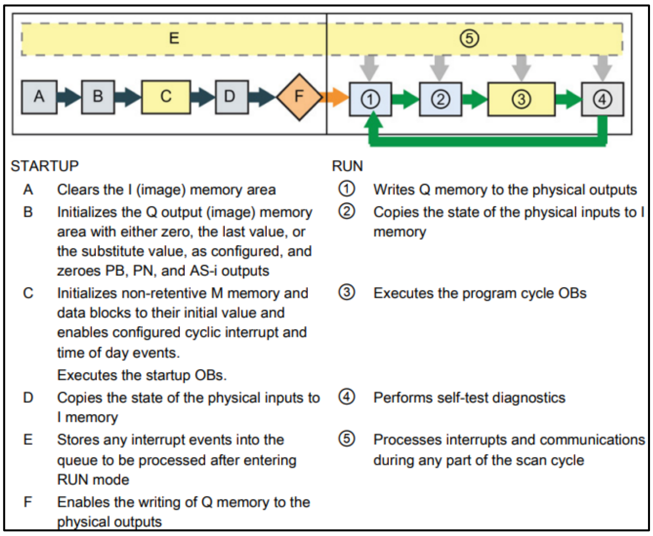

Figure 1: Scan cycle of a Siemens S7 1200 PLC

IV. COMMUNICATION

Most widely used industrial communication protocols supported by the PLCs are mentioned below:

Profibus (Process Field Bus):

Profibus is a widely used open fieldbus standard for industrial communication.

It supports both master-slave and peer-to-peer communication topologies.

Profibus is known for its high-speed data transmission, making it suitable for demanding industrial applications.

It is commonly used in manufacturing, process control, and automation systems.

Modbus:

Modbus is a serial communication protocol commonly used for connecting PLCs and other industrial devices.

It is a simple and robust protocol, making it easy to implement and widely adopted in various industries.

Modbus supports both serial (RS-232/RS-485) and Ethernet communication, offering flexibility in deployment.

It is often used for monitoring and controlling remote devices, such as sensors, actuators, and meters.

Ethernet/IP:

Ethernet/IP is an industrial Ethernet protocol based on the TCP/IP stack.

It allows PLCs and other automation devices to communicate over standard Ethernet networks.

Ethernet/IP supports both cyclic and acyclic communication, enabling real-time control and data exchange.

It is commonly used in modern automation systems due to its scalability, high bandwidth, and interoperability with IT networks.

Profinet (Process Field Net):

Profinet is an industrial Ethernet protocol developed by Siemens.

It offers real-time communication, high-speed data transmission, and flexibility in network topologies.

Profinet supports both cyclic and acyclic communication, enabling precise control and efficient data exchange.

It is widely used in manufacturing, automotive, and process industries for machine control and automation.

CANopen (Controller Area Network Open):

CANopen is a higher-layer protocol based on the CAN (Controller Area Network) bus.

It is commonly used for communication between PLCs, sensors, actuators, and other devices in distributed control systems.

CANopen provides a standardized communication profile, device configuration, and network management features.

It is widely adopted in applications requiring robustness, real-time performance, and interoperability.

V. CONCLUSION

PLCs continue to thrive in industrial automation due to their reliability, real-time performance, flexibility, scalability, interoperability, ease of maintenance, and adherence to industry standards. While newer technologies such as programmable automation controllers (PACs) and edge computing are emerging, PLCs remain a cornerstone of industrial control systems for the foreseeable future.

VI. REFERENCES

[1] Stevic, Tom (5 May 2017). "A very short history of PLC programming platforms". Control Design. Retrieved 11 September 2021.

[2] "IEC 61131-3:2013". International Electrotechnical Commission (IEC). Retrieved 11 September 2021.

[3] International Electrotechnical Commission, International Standard IEC 61131: Programmable Controllers, 4th ed., Geneva, Switzerland: IEC, 2017.

[4] J. W. Webb and R. A. Reis, Programmable Logic Controllers: Principles and Applications, 5th ed., Englewood Cliffs, NJ, USA: Prentice-Hall, 2002

[5] W. Bolton, Programmable Logic Controllers, 6th ed., Oxford, U.K.: Newnes, 2015.

[6] R. W. Lewis, Programming Industrial Control Systems Using IEC 1131-3. London, U.K.: IET, 1998.

[7] Siemens S7 1200 Easybook (Manual).