Mastering Siemens PLCs: From Fundamentals to Advanced Programming

---

Course Description
This course is designed to provide participants with a comprehensive understanding of Programmable Logic Controllers (PLCs) and their use in industrial automation. It covers the fundamentals of automation, various types of control systems, PLC architecture, programming languages, and I/O modules. Participants will also gain practical insights into analog processing, fault diagnostics, and parameter passing using Siemens technologies. Whether using simulation tools or real hardware, learners will develop skills to confidently program and troubleshoot Siemens PLC systems.

Audience Profile
This course is suitable for a wide range of learners interested in industrial automation, including:

- Engineering students aiming to develop skills in PLC programming and automation
- Technicians, electricians, and maintenance personnel seeking to upgrade their capabilities
- Engineers and managers involved in automation projects
- Anyone interested in acquiring a practical understanding of industrial automation systems

---

Prerequisite Knowledge/Skills
There are no specific prerequisites for this course. However, a basic understanding of electrical and electronic circuits is beneficial.

Course Objectives
Upon completion of this course, participants will be able to:

- Understand the basics of automation and different types of control systems
- Describe the different types of PLCs and their applications
- Explain the architecture of a PLC and various I/O modules
- Program a PLC using ladder logic, function block diagrams, and structured text
- Understand analog signal processing and perform fault diagnostics
- Apply parameter passing techniques and utilize shared data blocks effectively

---

KOENIG
step forward

Table of Contents

Module 4: TIA Portal and Programming Languages

- Importance of DI modules
- Cyclic program processing sequence
- PLC selection criteria
- Introduction to TIA Portal
- Hardware configuration and project setup
- PLC and PC communication
- Clock memory application
- Memory reset procedure
- Programming language conversion
- Ex 4.1 Converting Ladder Logic Program to Functional Block Diagram Logic

Module 5: PLC Operations and Memory Types

- Instruction writing
- Logic operations (AND, OR, NOR, etc.)
- Online diagnostics
- Data types (BIT, BYTE, WORD, DWORD)
- RLO and status concepts
- Load, work, and system memory
- Switch operation examples
- SET-RESET instructions
- Ex lab 5.1 - 2 Lamps and 3 NO/NC Contact coil Program
- Ex lab 5.2 - 3 Lamps and 3 NO/NC Contact coil Program using dual Network
- Ex lab 5.3 - 4 Lamps and 4 NO/NC Contact coil Program using Multiple Branching Network

Module 6: PLC Blocks and Timers

- Comparator block applications
- Program design techniques
- Timer types (pulse, delay, etc.)
- Counter operations

- Comparison functions and examples
- S-7 block types
- Organization blocks and error handling
- Ex lab 6.1 Sequential 2 Motor Control with Delayed Start and Stop Functionality
- Ex lab 6.2 Motor Control with Delayed Start and Timed Stop Hold
- Ex lab 6.3 - Car Parking Control System Using Sensors and Visual Indicator Logic
- Ex lab 6.4 - Sequenced Conveyor Control with Delayed Start/Stop and Cycle Limitation
- Ex lab 6.5 - Controlling Multiple Conveyor Belt with ON – OFF Bit Reset emergency stop functionality
- Additional Timer biased and Counter combined Logic Problems

Module 7: Block Architecture and Program Structure

- Warm, cold, and hot restarts
- FB vs FC blocks
- Block architecture and editing
- Program structure and routine calling
- Time-Of-Day and cyclic interrupts
- Practical programming examples

Module 8: Processing of Analog Values

- Analog value processing
- Analog sensor block diagram
- Signal flow from sensor to PLC
- 2-wire and 4-wire sensors
- Analog module wiring
- Engineering value conversion
- Norm X and Scale X Function Block
- Ex lab 8.1 - Analog Speed Regulation of Conveyor System through Potentiometer Signal Processing in TIA Portal

Module 9: Faults and Cross-Reference Function

- Functional vs system faults

- Forcing I/O variables
- Cross-reference usage
- Parameter passing
- Shared data blocks
- Practical fault diagnosis examples

Module 10: Practical Examples and Conclusion

- Real-world practical examples
- Course review
- Conclusion