

Human-Machine Interface (HMI) Design and Implementation

Course Description

The Human-Machine Interface (HMI) Design and Implementation course equips participants with the skills to create user-friendly and efficient HMIs for industrial and automation systems. Participants will learn HMI design principles, best practices, and real-world applications through theoretical lectures and practical exercises. The course covers topics such as SCADA overview, project creation, basic configuration, alarm configuration, data logging, trend configuration, recipe management, and user administration.

Audience

This course is ideal for individuals involved in industrial automation and control systems, including engineers, technicians, and system integrators. Participants should have a basic understanding of industrial automation systems and processes, and familiarity with programming concepts will be beneficial but not mandatory.

Pre-requisite Knowledge/Skills

Participants enrolling in the Human-Machine Interface (HMI) Design and Implementation course should have the following prerequisites:

- Basic understanding of industrial automation systems and processes: Familiarity with the concepts and functioning of industrial automation systems will help participants grasp the HMI design principles more effectively.
- Familiarity with programming concepts (not mandatory but beneficial): While not mandatory, having some knowledge of programming concepts will be advantageous in understanding certain aspects of HMI design and implementation

Course Objectives

Participants will learn to:

- Understand HMI fundamentals and WinCC software.
- Design user-friendly HMIs for industrial systems.
- Configure alarms, data logging, and recipe management.
- Manage user administration and access protection.
- Apply HMI best practices and troubleshoot issues.
- Consider ethical aspects of HMI design and development.

Course Outline

The course comprises 32-hours of theory and labs. It's divided into 7 different modules.

Module 1: Introduction to HMI

- SCADA Overview: Definition of HMI, Data Acquisition, Data Communication, Data Presentation
- Difference between PLC & PC
- Examples of HMI
- System Overview: WinCC Overview, Components of WinCC, WinCC Engineering Editions, WinCC Runtime Versions, Licensing, System Limits, System Requirements

Module 2: Project Creation & Basic Exercises

- Creating a Project
- Adding Devices and Establishing a Connection
- Setting Ethernet Address
- Connection to PLC
- Not Integrated Runtime Settings
- Configuring Buttons and Output
- Basic Examples Related to PBs & Dos

Module 3: Basic Configuration

- Configuring Graphic View
- Text Display Configuration
- Input and Output for Alphanumeric Display
- Text and Graphic List Configuration
- Movement and Fill Property Configuration

Module 4: Alarm Configuration

- Steps in Configuring an Alarm
- Message Blocks
- Alarm Class
- Trigger Setting for Discrete Alarm
- Displaying Alarms
- Alarm View Properties

- Practices on Different Examples

Module 5: Data Logging & Trend Configuration

- Types of Process Value Logging
- Procedure for Configuring Data Logs
- Properties of Process Value Logs
- Editing Logging Tags
- Displaying the Data Log Trend View
- Properties of the Trend View

Module 6: Recipe Management

- Structure of the Recipe
- Communication with the Controller Using Tags
- Table Representation of WinCC Recipe View
- Sorting in the Recipe View
- Exporting in the Recipe View
- Selection Dialog
- Procedure for Configuring a Recipe
- Recipe Editor
- Entering Recipe Data
- Setting Recipe Properties
- Configuring the Recipe View
- Entry of Recipe Data
- Practices on Different Examples

Module 7: User Administration

- Structure of User Administration
- How Access Protection Works
- Steps in Configuring User Administration
- Structuring Authorizations
- Creating Authorizations
- Assigning Authorizations to Objects
- Configuring User Groups
- Configuring Users
- Configuring the User View
- Configuring Login Dialog
- Displaying the Name of Logged-On Users

- Practices on Different Examples