



**Competency-Based Learning Materials (CBLMs)
on
Industrial Robot Operation and Troubleshooting
Light Engineering Sector**

**Skills for Industry Competitiveness and Innovation Program (SICIP)
Finance Division, Ministry of Finance**

Copyright Notice

The Competency-Based Learning Material (CBLM) for **Industrial Robot Operation and Troubleshooting** serves as a foundational reference for developing lesson plans, session plans, teaching materials, and assessment instruments tailored to the Light Engineering sector. This document ensures that training delivery aligns with industry requirements, enabling individuals who successfully meet the prescribed competency standards to qualify for relevant occupational roles within the sector.

This CBLM has been developed under the **Skills for Industry Competitiveness and Innovation Program (SICIP)**, and addresses the technical skills requirements essential for Industrial Robot Operation and Troubleshooting. The copyright of this document is held by the **Finance Division**, Ministry of Finance, Government of the People's Republic of Bangladesh.

Public and private training institutions may use the information contained in this document for training activities benefitting Bangladesh. Any reproduction, translation, storage, or transmission of this material - in whole or in part - for other purposes requires prior written permission from the copyright owner.

This document is available at:

Skills for Industry Competitiveness and Innovation Program (SICIP)

Finance Division, Ministry of Finance

Probashi Kallyan Bhaban (Level-15 & 16)

71-72 Old Elephant Road, Eskaton Garden

Dhaka-1000, Bangladesh

Phone: +8802 55138753-55, Fax: +8802 55138752

Website: <https://sicip.gov.bd/>

CBLM Development Committee

To develop this Competency-Based Learning Material (CBLM), a dedicated working group was formed under the supervision of Project Implementation Unit, SICIP Project. The development team conducted a detailed task analysis based on the performance criteria specified in the Competency Standards (CSs) for the trade **Industrial Robot Operation and Troubleshooting**, while also considering the competency gaps and practical needs of the target trainees.

After completing the task analysis, the components of the CBLM were developed through several consultation sessions, content development workshops, and iterative reviews. The draft CBLM was then presented to subject matter experts for validation. Following the final validation session, the CBLM was formally approved for training delivery.

Experts Involved

Name	Organization	Designation
Dr. Md. Jalal Uddin, PEng.	Bangladesh Industrial Technical Assistance Center (BITAC)	Director, Head PIU - SICIP, BITAC
Arif Ahmed	Bangladesh Industrial Technical Assistance Center (BITAC)	Assistant Engineer, Lead Trainer - Industrial Robot Operation and Troubleshooting
Raihan Rushdi	Bangladesh Industrial Technical Assistance Center (BITAC)	Executive Engineer, Trainer - Industrial Robot Operation and Troubleshooting
Stanley Hiron Sikder	Sincos Automation Technologies Ltd.	Sr. Automation Engineer

Introduction

Purpose of the CBLM

The **Competency-Based Learning Material (CBLM)** for *Industrial Robot Operation and Troubleshooting* has been developed to provide a structured, practical, and industry-aligned training resource for the Light Engineering and Manufacturing sectors.

This material follows the principles of **Competency-Based Training (CBT)**, ensuring that learners not only acquire theoretical knowledge but also demonstrate hands-on mastery through practical activities, workplace simulations, and real industrial tasks.

This CBLM aims to equip learners with the essential knowledge, skills, and attitudes required to work confidently with industrial automation systems. By completing this training, learners will be able to:

- Execute Electrical Sequence Circuit and Control using Relay Logic
- Perform Basic PLC Programming
- Analyze Industrial Sensors Data and Visualize on HMI
- Perform Electro Pneumatic Sequence Control
- Simulate Robotic Arm in Software
- Perform Robotic Program, Operate and Troubleshoot Industrial Robots
- Operate Collaborative Robot (COBOT)

Target Audience

This CBLM is designed for a broad range of learners, including:

- TVET students in Electrical, Electronics, Mechatronics, Mechanical, or Industrial Engineering
- Industry workers and technicians seeking to upgrade their automation skills
- Supervisors, foremen, and production personnel in manufacturing industries
- Entrepreneurs and SME owners planning to adopt automation technologies
- Fresh graduates or job seekers aiming to enhance employability

Prior knowledge of basic electricity, electronics, and safety practices is helpful, but not mandatory.

How to use this CBLM

To maximize learning outcomes, the following guidelines are recommended:

- Follow units and competencies in sequence—generic → sectoral → occupational
- Engage actively in all practical exercises; skill mastery develops through repeated hands-on practice
- Use this CBLM both during trainer-led sessions and for self-study
- Each unit includes assessment activities: practical tasks, written tests, and oral questioning
- Trainers are responsible for demonstrating tasks, guiding practice, and ensuring safety compliance
- Learners must practice regularly, maintain safe working habits, and collaborate with peers

By completing this CBLM, learners will be able to operate, configure, and troubleshoot industrial robotic systems confidently—contributing to technological capability development and sustainable industrial growth in Bangladesh’s Light Engineering sector.

Contents

Copyright	i
CBLM Development Committee	ii
Introduction	iii
List of Figures	14
List of Tables	16
1 Module-1: Relay Control and Motor Operation	17
1.1 Learning Outcome: Interpret Relay Logic Operation	17
1.1.1 Information Sheet: Relay Fundamentals.....	17
1.1.2 Information Sheet: Relay Connection pinout.....	18
1.1.3 Information Sheet: Relay, Circuit Breaker and Contactor	19
1.1.4 Information Sheet: Circuit Breaker Selection and Rating Calculation	20
1.1.5 Information Sheet: Circuit Breaker Trip Curve.....	22
1.1.6 Information Sheet: Comparison of Protection Devices	22
1.1.7 Information Sheet: Earth Leakage Current and Protection.....	23
1.1.8 Information Sheet: Boolean Logic Operation Using Relay.....	26
1.1.9 Job Sheet - Construct Boolean Logic Gates Using Relays	27
1.1.10 Job Sheet - Perform 3-Phase Motor Start and Stop Operations Using NO/NC Switch and Contactor Latching Circuit.....	29
1.1.11 Job Sheet - Construct and Test a Direct-On-Line (DOL) Starter Circuit.....	31
1.1.12 Self-Check Questions.....	34
1.1.13 Answer Key	35
1.2 Learning Outcome: Perform mechanical switching operation	36
1.2.1 Information Sheet: Mechanical Switches.....	36
1.2.2 Information Sheet: Major Types of Mechanical Switches	36
1.2.3 Information Sheet: Contact Configurations	38
1.2.4 Information Sheet: Symbol and Circuit Representation.....	38
1.2.5 Information Sheet: Reference Standards.....	39
1.2.6 Self-Check Questions	40
1.2.7 Answer Key	41
1.3 Learning Outcome: Execute timer and counter application.....	42
1.3.1 Information Sheet: Timer and Counter Fundamentals	42
1.3.2 Job Sheet - Construct and Test Sequential Operation Circuit Using Multiple Contactors, Timer, and Counter in Motor Control.....	43

1.3.3	Self-Check Questions	47
1.3.4	Answer Key	48
1.4	Learning Outcome: Perform forward reverse and star delta operation . .	49
1.4.1	Information Sheet: Star-Delta Starter	49
1.4.2	Job Sheet - Construct and Test Forward-Reverse Motor Starter Circuit	50
1.4.3	Job Sheet - Construct and Test Automatic 3-Phase Star-Delta Starter Circuit	52
1.4.4	Self-Check Questions	55
1.4.5	Answer Key	56
2	Module-2: Perform Basic PLC Programming	57
2.1	Learning Outcome: Install PLC and Associated Wiring	57
2.1.1	Information Sheet: PLC and Its Components	57
2.1.2	Information Sheet: Input/Output devices for PLCs	59
2.1.3	Information Sheet: Programming Languages (IEC 61131-3)	60
2.1.4	Information Sheet: PLC Selection Criteria	60
2.1.5	Information Sheet: Short Specifications of PLC (S7-1512C-1 PN Example)	61
2.1.6	Information Sheet: Step-by-Step Process to Install PLC, Related Hardware, and Associated Wiring (SIMATIC S7-1500 Example) .	61
2.1.7	Self-Check Questions	66
2.1.8	Answer Key	67
2.2	Learning Outcome: Setup PLC Hardware Configurations with Software Tool	68
2.2.1	Information Sheet: PLC Programming Software	68
2.2.2	Information Sheet: Basic Networking for Working with PLC . . .	69
2.2.3	Information Sheet: Addressing Configuration via TIA Portal . . .	70
2.2.4	Information Sheet: Hardware Configuration and Download	71
2.2.5	Job Sheet - Setup PLC Hardware Configuration Using Siemens TIA Portal	74
2.2.6	Self-Check Questions	76
2.2.7	Answer Key	77
2.3	Learning Outcome: Create PLC Program	78
2.3.1	Information Sheet: Creating a New Program in TIA Portal	78
2.3.2	Information Sheet: Addressing and Data Types	78
2.3.3	Information Sheet: Create and Assign Tags / Create Tag Table .	79
2.3.4	Information Sheet: Create Data Block (DB)	82
2.3.5	Information Sheet: What is Ladder Diagram (LAD)?	83
2.3.6	Job Sheet - Verify NO/NC Behaviour in Ladder Diagram	84
2.3.7	Information Sheet: Counter and Timer Instructions	87
2.3.8	Information Sheet: Math Functions and Data Handling Instruc- tions in TIA Portal	88
2.3.9	Job Sheet - Observe and Compare the Function of TP, TON, and TOF Timers in PLC	88
2.3.10	Job Sheet - Create a Blinking LED Using Timer in Ladder Logic	90
2.3.11	Job Sheet - Create a Traffic Light Control System Using Timers in Ladder Logic	92

2.3.12	Job Sheet - Count Car Garage Free Space Using Counter in Ladder Logic.....	94
2.3.13	Information Sheet: Use of Different Blocks in TIA Portal.....	96
2.3.14	Information Sheet: How to Use Monitor Mode (Monitoring Program Blocks).....	96
2.3.15	Information Sheet: Use of Force Table	97
2.3.16	Information Sheet: Using Online and Diagnostics Feature.....	98
2.3.17	Job Sheet - Use Math Functions to Solve Equation in Ladder Logic	101
2.3.18	Job Sheet - Read 0-10 V Potentiometer Value Using NORM_X and SCALE_X.....	103
2.3.19	Self-Check Questions.....	105
2.3.20	Answer Key	106
2.4	Learning Outcome: Design HMI for PLC	107
2.4.1	Information Sheet: HMI Installation and Communication Setup .	107
2.4.2	Information Sheet: Configuration of HMI Device and Connection	110
2.4.3	Information Sheet: Designing the HMI Screen (Visualization)	112
2.4.4	Information Sheet: Configuring Advanced HMI Functions	113
2.4.5	Information Sheet: Testing and Finalizing the HMI	114
2.4.6	Self-Check Questions	114
2.4.7	Answer Key	115
3	Module-3: Analyze Industrial Sensors	116
3.1	Learning Outcome: Identify Types of Industrial Sensors	116
3.1.1	Different Types of Industrial Sensors	116
3.1.2	Information Sheet-2: Sensor Schematic Circuit Symbols	117
3.1.3	Working Principles of Common Sensors.....	118
3.1.4	Information Sheet-3: Sensor Related Terminologies	126
3.1.5	Information Sheet: Sensor Mounting Process (Flush, Non-flush) .	127
3.1.6	Self-Check Questions	128
3.1.7	Answer Key	129
3.2	Learning Outcome: Perform Sensor Wiring using Correct Interface.....	130
3.2.1	What is meant by Sensor Interface	130
3.2.2	RS-485 Sensor Wiring with PLC.....	133
3.2.3	RTD (Resistance Temperature Detector) Sensor Wiring with PLC	134
3.2.4	Job Sheet - Detecting the Speed of a Gear.....	135
3.2.5	Self-Check Questions	136
3.2.6	Answer Key	137
3.3	Learning Outcome: 3.3 - Generate PLC Program for Data Acquisition	138
3.3.1	Information Sheet: NORM_X and SCALE_X Function in PLC .	138
3.3.2	Job Sheet: Wiring and Programming Proximity Sensors (Push-Pull Interface).....	139
3.3.3	Job Sheet: Calibrating Proximity Sensors	140
3.3.4	Job Sheet: Perform 4–20 mA Loop Calibration.....	140
3.3.5	Job Sheet: Wiring and Programming 4–20 mA Sensor	141
3.3.6	Job Sheet: Wiring and Programming Modbus Sensor.....	142
3.3.7	Job Sheet: Wiring and Programming Potentiometer.....	142
3.3.8	Job Sheet: Wiring and Programming RTD Sensor	143
3.3.9	Job Sheet: Wiring and Programming High-Speed Counter (Encoder)	144

3.3.10	Job Sheet: Wiring and Programming VFD with PLC	144
3.3.11	Self-Check Questions.....	145
3.3.12	Answer Key	146
3.4	Learning Outcome: Perform Data Analysis on Sensor Outputs.....	147
3.4.1	Information Sheet: Performing Data Analysis on Sensor Outputs via HMI	147
3.4.2	Job Sheet: Visualize Digital Sensor Output in HMI (Optical Diffuse Sensor)	147
3.4.3	Job Sheet: Visualize Analog Sensor Output in HMI (0-10 V Analog Proximity Sensor)	148
3.4.4	Job Sheet: Data Logging in HMI TP700 Comfort	149
3.4.5	Job Sheet: Web-Based Monitoring in HMI.....	150
3.4.6	Job Sheet: OPC-UA with HMI.....	151
3.4.7	Self-Check Questions	154
3.4.8	Answer Key	155
4	Module-4: Perform Electro-pneumatic Sequence Control	156
4.1	Learning Outcome: Understand Electronic, Pneumatic and Elctro-pneumatic Circuit Diagrams	156
4.1.1	Information Sheet: Fundamentals of Pneumatics and Electro-Pneumatics.....	156
4.1.2	Information Sheet - Interpreting Pneumatic Valve Symbols.....	158
4.1.3	Information Sheet: Interpreting Circuit Symbols used in pneumatic and electro-pneumatic systems	164
4.1.4	Information Sheet: Working Principle of Common Pneumatic Sys- tem Components.....	166
4.1.5	Information Sheet - Interpreting Function of Pneumatic Circuits .	172
4.1.6	Information Sheet - Interpreting Function of Electro-pneumatic Circuits	176
4.1.7	Self-Check Questions	181
4.1.8	Answer Key	182
4.2	Learning Outcome: Design and Simulate Electropneumatic Circuit.....	183
4.2.1	Information Sheet: Circuit Design Software	183
4.2.2	Information Sheet - Designing and Simulating a Basic Electro- Pneumatic Circuit in FluidSIM.....	184
4.2.3	Information Sheet: Design Electro-pneumatic Sequence Control Table and Circuit	185
4.2.4	Design and Simulate Pneumatic and Electro-Pneumatic Circuit (A+ A-).....	190
4.2.5	Job Sheet - Design and Simulate Multiple Cylinder Sequence Cir- cuit in FluidSIM	192
4.2.6	Job Sheet - Sliding Door Logic Circuit Simulation in FluidSIM.....	194
4.2.7	Job Sheet - Simulated Cleaning Cycle (Oscillation for 20 s).....	195
4.2.8	Job Sheet - Simulate Pressure-Dependent Stamping Logic.....	196
4.2.9	Job Sheet - Simulate Pressure-Conditioned Clamp and Press Se- quence	197
4.2.10	Self-Check Questions.....	198
4.2.11	Answer Key	199
4.3	Learning Outcome: Carry out Electropneumatic Circuit Connection	200

4.3.1	Information Sheet - ISO / IEC Standards for Emergency Stop (E-Stop)	200
4.3.2	Information Sheet - Overlapping Sequential Control with Double Solenoid Valves (Including Reset Step)	204
4.3.3	Information Sheet - Single Cycle and Continuous Cycle Operating Modes of the Controller	204
4.3.4	Electro-pneumatic implementation of Emergency Stop Function .	207
4.3.5	Job Sheet - Multi-Cylinder Sequence (A+B+A-B-(C+B+)C-B-) .	208
4.3.6	Job Sheet - Add E-Stop Button in Electro-pneumatic Circuit . . .	212
4.3.7	Self-Check Questions	214
4.3.8	Answer Key	215
4.4	Learning Outcome: Generate PLC Program for Sequence Control	216
4.4.1	Information Sheet: Introducing MPS station components, module, elements	216
4.4.2	Information Sheet - Distributing Station (via SysLink I/O Table) Pin Assignment	220
4.4.3	Information Sheet - Sorting Station (via SysLink I/O Table) Pin Assignment	220
4.4.4	Information Sheet: MPS electrical components circuit labels and pin mapping with PLC I/O	221
4.4.5	Information Sheet: DC Motor Controller Pinout	226
4.4.6	Information Sheet: Mini I/O and Centronics Terminal Circuit Interpretation	231
4.4.7	Information Sheet: Sensor Circuit Interpretation	235
4.4.8	Information Sheet - Motor and Actuator Output Wiring Circuit .	236
4.4.9	Job Sheet - PLC Programming and HMI Design for MPS Distribution Station	239
4.4.10	Job Sheet - Programming and Simulation of MPS Sorting Station	241
4.4.11	Self-Check Questions	250
4.4.12	Answer Key	251
4.5	Learning Outcome: Operate Modular Production System (MPS) Stations	252
4.5.1	Modular Production System (MPS) Stations Commissioning Sequence	252
4.5.2	Information Sheet - Commissioning Sequence for the MPS Distributing/Conveyor Station	254
4.5.3	Information Sheet - Commissioning Sequence for the MPS Sorting Station	254
4.5.4	Information Sheet - Calibrating Diffuse Sensor Sensitivity for Workpiece Detection	256
4.5.5	Information Sheet - Diagnosing Network Connection Issues (S7-1516, TP700, PC)	258
4.5.6	Information Sheet - Backup and Restore Procedures for S7-1500 PLC and HMI Devices	261
4.5.7	Job Sheet - Operate Sorting Station	262
4.5.8	Job Sheet - Operate Distributing Station	264
4.5.9	Job Sheet - Calibrate Diffuse Sensor for Black Workpiece Detection	266
4.5.10	Self-Check Questions	268
4.5.11	Answer Key	270

5	Module-5: Simulate Robotic Arm in Software	271
5.1	Learning Outcome: Install and Configure Simulation Software.....	271
5.1.1	Information Sheet 1: Introduction to CIROS Studio	271
5.1.2	Information Sheet 2: Understanding User Interface of Software	272
5.1.3	Information Sheet 3: Communication and Integration.....	272
5.1.4	Self-Check Questions	273
5.1.5	Answer Key	274
5.2	Learning Outcome: Create and Configure Simulation Environment.....	275
5.2.1	Information Sheet 1: Initial Setup and Modeling.....	275
5.2.2	Information Sheet 2: Selection of Robot Model and End Effector .	276
5.2.3	Information Sheet 3: Create Custom 3D Object	278
5.2.4	Information Sheet 4: Add and Configure Source/Sink Mechanisms for Pick and Place Simulation.....	279
5.2.5	Information Sheet 5: Add Sensors and Conveyor.....	280
5.2.6	Job Sheet - Initial Setup and Robot Modeling in CIROS	281
5.2.7	Job Sheet - Creating Custom 3D Workpiece and Environment	282
5.2.8	Self-Check Questions	285
5.2.9	Answer Key	286
5.3	Learning Outcome: Generate Robotic Program	287
5.3.1	Information Sheet 1: I/O Configuration and Programming	287
5.3.2	Information Sheet 2: Create Position List (Teaching)	287
5.3.3	Information Sheet 3: Generate Robot Program (MELFA-BASIC V Example).....	288
5.3.4	Information Sheet 4: Test and Simulation.....	289
5.3.5	Information Sheet 5: Program Export	289
5.3.6	Self-Check Questions	290
5.3.7	Answer Key	291
5.4	Learning Outcome: Transfer Program File to Robotic Arm System.....	292
5.4.1	Information Sheet 1: Transfer Program File to Robotic Arm System	292
5.4.2	Information Sheet 2: Transfer Program into the Robot Controller	292
5.4.3	Job Sheet - Programming and Simulation of Pick-and-Place Robot in CIROS	293
5.4.4	Self-Check Questions	295
5.4.5	Answer Key	296
6	Module-6: Perform Robotic Program	297
6.1	Learning Outcome: Identifying Robot Programming Syntax	297
6.1.1	Information Sheet: Core Terminology in Robotics	297
6.1.2	Information Sheet - Mitsubishi Robot Programming Language and Versions	300
6.1.3	Information Sheet - Variable Types and Characteristics	301
6.1.4	Information Sheet - Command Structures for Programming the Robotic Arm.....	301
6.1.5	Information Sheet - Operators and Logical Conditions.....	302
6.1.6	Information Sheet - Continuous and Conditional Statements.....	302
6.1.7	Self-Check Questions	303
6.1.8	Answer Key	304
6.2	Learning Outcome: Develop Program for Robot Arm Movement	304

6.2.1	Information Sheet - Understanding Robot Coordinate Systems . .	304
6.2.2	Information Sheet -Hand Jog via Teach Pendant Buttons.....	307
6.2.3	Information Sheet - Defining the Robot's Starting (Home) Position	308
6.2.4	Information Sheet - Robot Jogging Using Teaching Box.....	310
6.2.5	Information Sheet - Defining Tool Center Point (TCP)	310
6.2.6	Job Sheet - Perform Robot Jogging.....	311
6.2.7	Information Sheet - Setting Joint Angles or Coordinates (Position Teaching)	313
6.2.8	Information Sheet - Teaching Pre-Programmed Positions	314
6.2.9	Information Sheet - Programming Motion Paths	314
6.2.10	Information Sheet - Program to Control Speed and Acceleration .	315
6.2.11	Information Sheet - I/O Programming for End-Effector (Parallel Gripper)	315
6.2.12	Information Sheet - Example Pick-and-Place Program.....	315
6.2.13	Information Sheet - Executing a Pre-Written Robot Program.....	316
6.2.14	Information Sheet - Manual Execution for Debugging	316
6.2.15	Information Sheet - Error Handling and Alarms	316
6.2.16	Job Sheet - Program Robot to Pick and Place Workpiece.....	317
6.2.17	Job Sheet - Debugging and Verification of MELFA Basic Program	320
6.2.18	Self-Check Questions.....	323
6.2.19	Answer Key	324
6.3	Learning Outcome: Develop Program for Controlling Robot Input and Output.....	324
6.3.1	Information Sheet - Understanding Input and Output Devices in MPS Robot Station	325
6.3.2	Information Sheet - Controlling Input and Output Devices	325
6.3.3	Information Sheet - MPS Multifunction Gripper I/O Programming	327
6.3.4	Information Sheet - Programming Example: Detect Product Color and Place Piston	328
6.3.5	Self-Check Questions	329
6.3.6	Answer Key	330
6.4	Learning Outcome: Operate Robotic Arm	330
6.4.1	Information Sheet - Safety Guidelines	330
6.4.2	Information Sheet - Subroutine Programming	331
6.4.3	Information Sheet - User Base Program (UBP) Usage.....	331
6.4.4	Information Sheet - Palletizing Programming	331
6.4.5	Interrupt Programming.....	332
6.4.6	Information Sheet: Programming Robot for Assembling Process with Screwing.....	333
6.4.7	Job Sheet - Operate Robotic Arm using Teaching Box (T/B).....	336
6.4.8	Self-Check Questions	338
6.4.9	Answer Key	339
7	Module-7 : Operate Collaborative Robot (COBOT)	340
7.1	Learning Outcome: Understand Collaborative Robot Principles, Features and Safety Protocols.....	340
7.1.1	Information Sheet - Introduction to Collaborative Robots (Cobots) and Their Features	340

CONTENTS

7.1.2	Information Sheet - Principles and Applications of Collaborative Robots (Cobots)	341
7.1.3	Information Sheet - Safety Standards and Considerations for Collaborative Robots (Cobots)	342
7.1.4	Self-Check Questions	346
7.1.5	Answer Key	347
7.2	Learning Outcome: Set up and Operate Collaborative Robots	348
7.2.1	Information Sheet - Installation and Mounting of Collaborative Robots (Cobots)	348
7.2.2	Information Sheet - Introduction to the Universal Robots Cobot Programming Interface	350
7.2.3	Information Sheet - Initializing and Configuring the Universal Robots Cobot System	356
7.2.4	Information Sheet - Installation and Configuration of End-Effectors	362
7.2.5	Self-Check Questions	366
7.2.6	Answer Key	367
7.3	Learning Outcome: Integrate COBOT with CNC Machine	368
7.3.1	Information Sheet - Identifying CNC I/O Terminals	368
7.3.2	Information Sheet - Performing Signal Wiring and I/O Configuration	368
7.3.3	Information Sheet - Simulate Robot Programs Using URSim in a Virtual Environment	369
7.3.4	Self-Check Questions	374
7.3.5	Answer Key	375
7.4	Learning Outcome: Program Tasks for Collaborative Robots	376
7.4.1	Information Sheet - Programming Node Functions in the PolyScope Graphical User Interface	376
7.4.2	Information Sheet - Detailed Guide for Programming Palletizing in PolyScope	378
7.4.3	Information Sheet - Integration and Operation of the AG-95 Gripper with Universal Robots	381
7.4.4	Self-Check Questions	384
7.4.5	Answer Key	385
7.5	Learning Outcome: Troubleshoot and Optimize Robot	386
7.5.1	Information Sheet- Safety Instructions for Universal Robots Systems	386
7.5.2	Information Sheet-Troubleshooting and Maintenance Steps for Universal Robots Systems	388
7.5.3	Information Sheet- Communication Protocols of Universal Robots (UR) Systems	391
7.5.4	Self-Check Questions	394
7.5.5	Answer Key	395

List of Figures

1.1	Relay internal construction	17
1.2	Relay pin configuration	17
1.3	8-Pin Relay.....	18
1.4	Relay Pin Designation and Functions.....	18
1.5	Example of Circuit Breaker Trip Curve	22
1.6	Comparison of B, C, and D Trip Curves	22
1.7	Typical arrangement of MCB, MCCB, ELCB, and RCCB in an electrical distribution panel.....	23
1.8	Residual current flow and RCCB protection mechanism	23
1.9	Testing relay contacts using a multimeter	25
1.10	Boolean logic truth table and symbols.....	26
1.11	Basic relay logic circuit	28
1.12	Contacting latching circuit	31
1.13	DOL starter circuit	33
1.14	Push Button Switch.....	37
1.15	Toggle Switch	37
1.16	Selector Switch	37
1.17	Limit Switch	37
1.18	Foot (Pedal) Switch.....	38
1.19	Float Switch.....	38
1.20	Manual Timer.....	42
1.21	Illustration of Manual Counter.....	43
1.22	Power Circuit Diagram of Sequential Control.....	46
1.23	Control Circuit Diagram of Sequential Control.....	46
1.24	Motor forward-reverse operation circuit.....	52
1.25	Start-Delta starter control circuit.....	55
1.26	Star-Delta starter power circuit.....	55
2.1	PLC system components (example: SIMATIC S7-1500)	57
2.2	Common PLC input and output devices	59
2.3	Common PLC input and output devices	60
2.4	Mounting rail and module arrangement for SIMATIC S7-1500	62
2.5	Installation of S7-1500 CPU (left) and I/O modules (right) on mounting rail	63
2.6	Typical wiring diagram for PLC.....	64
2.7	Power connections wiring	65
2.8	Simatic Memory Card.....	65
2.9	CPU display and control key for SIMATIC S7-1500 configuration	66
2.10	Interface Views in SIMATIC TIA Portal.....	68

LIST OF FIGURES

2.11	Basic PLC networking and communication setup	69
2.12	Assigning IP address to unknown PLC via TIA Portal	70
2.13	Setting Up S7-1500 PLC IP Address via CPU Display	71
2.14	TIA Portal hardware download process	73
2.15	Add new program block in TIA Portal	78
2.16	Addressing methods in PLC	79
2.17	Tag table overview	80
2.18	Opening tag table in TIA Portal	81
2.19	Entering tag details	81
2.20	Creating a new Data Block	82
2.21	Selecting Data Block type	82
2.22	Configuring variables in Data Block	83
2.23	Example of Ladder Logic network	83
2.24	AND and OR logic circuits	84
2.25	NOT, NAND, NOR, XOR, and XNOR Ladder logic examples	84
2.26	Timer operations in TIA Portal	87
2.27	TIA Portal program structure	96
2.28	Go online and monitor program blocks	97
2.29	Force table overview	98
2.30	Force value entry example	98
2.31	Establishing online connection	99
2.32	CPU online control tools	99
2.33	Diagnostics buffer view	100
2.34	Hardware comparison and synchronization	100
2.35	Typical HMI hardware setup	107
2.36	Assigning IP address to HMI in TIA Portal	108
2.37	IP assignment from HMI interface	109
2.38	Adding HMI device in TIA Portal	110
2.39	Creating HMI connection in Network view	111
2.40	WinCC HMI screen interface	112
2.41	Screen hierarchy planning	112
2.42	Screen layout editor in WinCC	113
2.43	HMI testing and simulation interface	114
3.1	Different types of industrial sensors used in automation systems	117
3.2	Common industrial sensor symbols (IEC 60617 reference)	118
3.3	Working principle of a through-beam optical sensor	119
3.4	Working principle of a retro-reflective optical sensor	119
3.5	Working principle of a diffuse optical sensor	119
3.6	Fibre-optic cable used for light guidance in sensors	120
3.7	Common types of optical proximity sensors	120
3.8	Working principle of an electrical limit switch	121
3.9	Working principle of a pressure switch	121
3.10	Working principle of a reed switch	121
3.11	Internal structures of proximity sensors	122
3.12	Basic working principles of inductive sensors	122
3.13	Comparison between Magneto-resistive and Hall Effect sensors working principles	123

3.14	Working principle of a thermocouple.....	123
3.15	RTD sensor and its signal conditioning circuit.....	123
3.16	Infrared (IR) pyrometer for non-contact temperature sensing.....	124
3.17	Float-type level switch principle.....	124
3.18	Ultrasonic level sensor operation using time-of-flight principle.....	124
3.19	Capacitive level sensor working concept.....	124
3.20	Differential pressure flow measurement using orifice plate.....	125
3.21	Coriolis mass flow meter showing vibrating tube principle.....	125
3.22	Flush vs. Non-flush Mounting of Inductive Proximity Sensors.....	128
3.23	Typical PNP (Sourcing) sensor wiring diagram.....	131
3.24	Typical NPN (Sinking) sensor wiring diagram.....	131
3.25	Comparison of PNP (Sourcing) vs. NPN (Sinking) Sensor Wiring.....	132
3.26	Push-pull (antivalent) sensor wiring with NO and NC outputs.....	132
3.27	4-20 mA current loop wiring and PLC analog input connection.....	133
3.28	Typical wiring of a 0-10 V analog sensor with PLC analog input.....	133
3.29	Modbus RTU wiring and CM PTP configuration overview.....	133
3.30	3-wire RTD wiring connection to PLC analog module.....	134
3.31	Use of NORM_X and SCALE_X functions for analog data processing in PLC.....	138
4.1	Typical components of an electro-pneumatic system.....	158
4.2	Identification of valve ports and ways.....	159
4.3	Identification of valve positions and flow directions.....	159
4.4	Common actuation and return symbols (ISO 1219-1).....	160
4.5	Example: 4/3 way, Center Closed, Pilot Operated, Spring Return Solenoid Valve with Manual Override.....	161
4.6	Symbols of common pneumatic components.....	162
4.7	Symbols of common electro-pneumatic components.....	163
4.8	symbols of common pneumatic circuit elements.....	163
4.9	Single-acting cylinder - circuit symbol and cutaway view.....	166
4.10	Double-acting cylinder operation.....	166
4.11	3/2-way valve, manually operated, with spring return, normally closed - circuit symbol and cutaway view.....	167
4.12	3/2-way normally open valve operation.....	167
4.13	5/2-way double pilot (bistable) valve.....	167
4.14	One-way flow control valve principle.....	168
4.15	Shuttle valve (OR gate) symbol and function.....	168
4.16	Dual-pressure valve (AND gate) symbol.....	168
4.17	Quick exhaust valve air path.....	169
4.18	Pressure sequence valve structure.....	169
4.19	Pressure regulator valve cross-section.....	169
4.20	Time delay valve (pneumatic timer) schematic.....	170
4.21	Solenoid valve with spring return.....	170
4.22	Double solenoid (impulse) valve diagram.....	170
4.23	Pneumatic limit switch with roller actuator.....	171
4.24	Pneumatic proximity sensor configuration.....	171
4.25	Pneumatic OR gate operation using a shuttle valve.....	172
4.26	Pneumatic AND gate operation using a dual-pressure valve.....	173

LIST OF FIGURES

4.27	Pneumatic circuit diagram for combined boolean logic operations	174
4.28	Pneumatic latching circuit using shuttle and 3/2-way valves.....	175
4.29	Operation of single-acting cylinder controlled by 3/2-way solenoid valve .	176
4.30	Operation of double-acting cylinder using 5/2-way solenoid valve with relay control circuit.....	177
4.31	Pneumatic Circuit of Automatic oscillation of double-acting cylinder using relays and double solenoid valve.....	178
4.32	Pneumatic Circuit of Automatic oscillation of double-acting cylinder using relays and double solenoid valve.....	179
4.33	Sequential operation of two cylinders using relay logic and solenoid valves	180
4.34	Sequential operation of two cylinders using relay logic and solenoid valves	180
4.35	Typical components of an electro-pneumatic system	183
4.36	Circuit for sequence control of A+ and A- using relays and limit switches	189
4.37	Circuit for sequence control of A+ B+ A- B- using relays and limit switches	189
4.38	Circuit for sequence control of A+ B+ A- B- using relays and limit switches	190
4.39	Pneumatic circuit for A+ A- sequence.....	191
4.40	Electro-pneumatic circuit for A+ A- sequence.....	192
4.41	Pneumatic logic circuit for sliding door actuation (simulation).....	194
4.42	Simulated Cleaning Cycle (Oscillation for 20 s) Circuit	195
4.43	Circuit for sequence control stamping operation	196
4.44	Circuit for sequence control of press after clamp.....	197
4.45	Comparison of Incorrect vs Correct PLC E-Stop Implementation	203
4.46	Sequential control of two cylinders using double solenoid valves with reset step (Pneumatic Circuit).....	205
4.47	Sequential control of two cylinders using double solenoid valves with reset step (Electrical Circuit).....	206
4.48	Sequential control of two cylinders with integrated emergency stop function	210
4.49	Sequential control of two cylinders with integrated emergency stop function	211
4.50	Festo Modular Production System (MPS) structure	216
4.51	MPS Robot Station.....	216
4.52	Conveyor module.....	217
4.53	Stack Magazine Module.....	217
4.54	Formation of Distributing/Conveyor Station	217
4.55	Detection Module.....	218
4.56	Chute for storing workpieces.....	218
4.57	Sorting Station.....	218
4.58	Components of Robot Station	219
4.59	Robot Handling Module	219
4.60	Robot Assembly Module	219
4.61	Mini I/O Terminal.....	222
4.62	C-Interface.....	223
4.63	Robot Interface Box (RIA Box).....	224
4.64	RIA box - Front View	224
4.65	RIA box - Rear View.....	224
4.66	MPS Station and Robot I/O connections via RIA Box	225
4.67	DC Motor Controller	227
4.68	DC motor creep speed	231
4.69	Festo Mini I/O Terminal circuit layout and connection designations	232

4.70	Centronics board circuit layout and connection designations	234
4.71	Connection layout of three sensors to Festo Mini I/O Terminal block.....	235
4.72	Simplified pin diagram for solenoid and motor actuator connection.....	237
4.73	Solenoid (-G1MB1) direct control connection via X2 Pin 9 and GND A return	238
4.74	Motor forward/reverse operation via controller.....	239
4.75	Distributing/Conveyor Station Tag Table	244
4.76	Distributing/Conveyor Station Ladder Network 1.....	244
4.77	Distributing/Conveyor Station Ladder Network 2.....	244
4.78	Distributing/Conveyor Station Ladder Network 3.....	245
4.79	Distributing/Conveyor Station Ladder Network 4.....	245
4.80	Distributing/Conveyor Station Ladder Network 5.....	245
4.81	Distributing/Conveyor Station Ladder Network 6.....	245
4.82	Distributing/Conveyor Station Ladder Network 7	246
4.83	Distributing/Conveyor Station Ladder Network 8.....	246
4.84	Distributing/Conveyor Station Ladder Network 9.....	246
4.85	Sorting Station Tag Table.....	247
4.86	Sorting Station Ladder Network 1	247
4.87	Sorting Station Ladder Network 2	247
4.88	Sorting Station Ladder Network 3	248
4.89	Sorting Station Ladder Network 4	248
4.90	Sorting Station Ladder Network 5	248
4.91	Sorting Station Ladder Network 6	248
4.92	Sorting Station Ladder Network 7	248
4.93	Sorting Station Ladder Network 8	249
4.94	Sorting Station Ladder Network 9	249
4.95	Sorting Station Ladder Network 10	249
4.96	Easyport Simulation Box.....	253
4.97	Sensitivity adjustment procedure for diffuse sensors	256
4.98	Lens contamination of reflective sensors.....	257
4.99	Variation of scanning width with object color.....	257
4.100	Variation of scanning width with object size.....	258
6.1	Degrees of freedom in a kinematic system	298
6.2	Work envelope of a robotic arm.....	298
6.3	Different types of end-effectors	299
6.4	Relationship between Payload and Center of Gravity (CoG) for Robots .	299
6.5	Forward Kinematics vs Inverse Kinematics	300
6.6	Hand jog and manual open/close via teach pendant.....	307
6.7	Navigating to UBP program on Teaching Box.....	308
6.8	UBP program	309
6.9	UBP file pre-taught positions	309
6.10	Position jump to any selected position.....	310
6.11	Teaching Box interface for returning robot to Home position.....	311
7.1	Cobot and human working together in a shared workspace	341
7.2	Stop Categories for Robot Safety System Responses	343
7.3	Cobot joint exploded view.....	344
7.4	Universal Robots e-Series Controller I/O Terminals.....	349

LIST OF FIGURES

7.5	PolyScope GUI layout (Header, Footer, Screen).	350
7.6	PolyScope Header Icons and Functions.....	351
7.7	Polyscope Program Tab Interface.....	352
7.8	Polyscope I/O Tab Interface	353
7.9	PolyScope Footer Buttons and Functions.....	353
7.10	PolyScope Play, Pause, Resume Buttons	354
7.11	Universal Robots e-Series Teach Pendant.....	355
7.12	PolyScope Initialize screen.....	356
7.13	Powering ON the universal cobot	357
7.14	Polyscope move tab interface.	358
7.15	Tool center point co-ordinates and center of gravity	359
7.16	Freedrive interface in polyscope.....	360
7.17	Freedrive icons explanation.....	361
7.18	Set waypoint in UR programming	361
7.19	Starting polyscope virtual box image	370
7.20	URSim Linux OS launch screen.	372
7.21	Finding IP addresses for URSim and Windows environment.....	373
7.22	Manual jogging, aligning and homing of UR cobot	376

List of Tables

1.1	Comparison of Features between Relay, Contactor, and Circuit Breaker .	19
1.2	Final MCCB Selection Parameters.....	22
1.3	Comparison of Protection Devices.....	24
1.4	Typical Sensitivity Ratings of RCCBs	25
1.8	Self-Check Questions	34
1.9	Answer Key for Self-Check Questions	35
1.10	Common Contact Configurations of Mechanical Switches.....	38
1.11	Symbols and Circuit Representations of Common Mechanical Switches .	39
1.12	Reference Standards Related to Mechanical Switches and Control Devices	39
2.1	Key Components and Functions of SIMATIC S7-1500 Controller System	58
2.2	Classification and Function of PLC Input and Output Devices.....	59
2.3	Specifications and Explanation of SIMATIC S7-1512C-1 PN PLC	62
2.9	Common Data Types in Siemens PLCs	80
2.10	Comparison between Data Block and Tag Table	83
2.11	Basic Ladder Logic Elements.....	84
2.13	Common Timer Instructions in TIA Portal	87
2.14	Counter Instructions in Siemens LAD.....	87
2.15	Common Data Handling and Math Instructions in TIA Portal.....	88
2.20	Block Types and Their Characteristics	96
4.17	SysLink Pin Assignment and Signal Functions for Conveyor Module	220
4.18	SysLink Pin Assignment and Signal Functions.....	221
4.19	Interpretation of Mini I/O terminal reference code	222
4.22	Example 15-Pin D-Sub HD Pinout (C-Interface - V1XG1 / Distribution Station C-Interface - C1XG1).....	224
4.23	Connection (Robot Interface Box).....	225
4.24	Input Pin Mapping for Modules X0-X4	226
4.25	Output Pin Mapping for Modules X0-X4	227
4.26	Robot Controller Inputs Used	228
4.30	DC Motor Controller Terminal Allocation and Function	228
4.27	Robot Controller Outputs Used.....	230
4.28	Contact Assignment Table (Hand Sensor Connector CON1H)	230
4.29	Contact Assignment Table (Solenoid Valve Connection).....	231
4.31	Pin Mapping between X2 Field Terminals and X1 D-Sub Connector	233
4.32	Mapping of Sensor Outputs from Mini I/O Terminal to PLC Inputs	236
6.8	Robot Coordinate Systems and Their Functions.....	305

LIST OF TABLES

7.1	International Safety Standards for Robotics	342
7.4	PolyScope Header Icons and Functions.....	352
7.5	PolyScope Footer Buttons and Functions.....	354
7.6	Teach Pendant Physical Features.....	355
7.7	Control Box Features and Connections.....	356

Module-1: Relay Control and Motor Operation

1.1 Learning Outcome: Interpret Relay Logic Operation

1.1.1 Information Sheet: Relay Fundamentals

A relay is fundamentally an **electrically operated switch** that allows low-power signals to control high-power circuits. It is an *electromagnetic switch* that uses a low-power control signal to control a high-power circuit.

Key Components

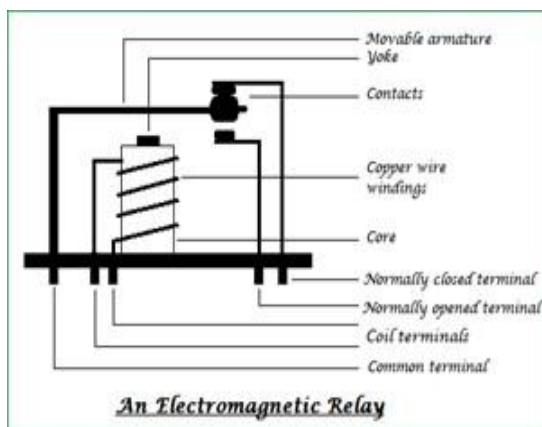


Figure 1.1: Relay internal construction

The essential parts of a relay include:

- **Coil:** The electromagnet that activates the relay when control voltage is applied.
- **Common (COM):** The moving part of the relay that connects to either the NO or NC terminal.
- **Normally Open (NO) Contact:** A terminal connected to COM only when the relay coil is energized.
- **Normally Closed (NC) Contact:** A terminal connected to COM when the relay is not energized.
- **Control Pins:** The terminals connected to the coil for activating the relay.

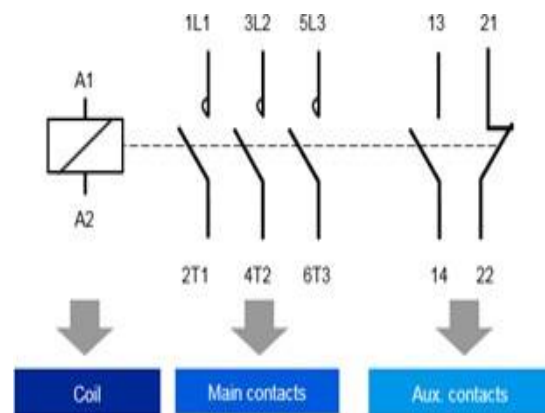


Figure 1.2: Relay pin configuration

Types of Relays

Relays come in various types based on their operation and application. The main types include:

- **Electromechanical Relays (EMR):** Uses an electromagnet to switch contacts.
- **Solid-State Relays (SSR):** Uses semiconductor devices instead of moving parts.
- **Time-Delay Relays:** Activate or deactivate after a preset delay.
- **Overload Relays:** Protect motors from excessive current.
- **Thermal Overload Relay:** Trips when electrical components exceed safe temperature limits, often due to high current or poor ventilation.

1.1.2 Information Sheet: Relay Connection pinout



Figure 1.3: 8-Pin Relay

Figure 1.4: Relay Pin Designation and Functions

Pin No.	Designation	Function
1	Coil (A1)	One side of the relay coil
8	Coil (A2)	Other side of the relay coil
2	Common (COM1)	Common contact for pole 1
3	Normally Closed (NC1)	Connected to COM1 when coil is de-energized
4	Normally Open (NO1)	Connected to COM1 when coil is energized
5	Common (COM2)	Common contact for pole 2
6	Normally Closed (NC2)	Connected to COM2 when coil is de-energized
7	Normally Open (NO2)	Connected to COM2 when coil is energized

1.1.3 Information Sheet: Relay, Circuit Breaker and Contactor**Comparison of Relay, Contactor, and Circuit Breaker**

A **relay**, **contactor**, and **circuit breaker** are all electrical switching devices, but they differ significantly in function, load capacity, and application.

- A **relay** is a low-power, electrically operated switch used for control logic.
- A **contactor** is a heavy-duty relay designed for frequent switching of high-power equipment such as motors.
- A **circuit breaker** is an automatic protection device that safeguards circuits from overcurrents.

Table 1.1: Comparison of Features between Relay, Contactor, and Circuit Breaker

Feature	Relay	Contactor	Circuit Breaker
Primary Function	Control logic and signal switching. Often used to allow a low-power circuit to control a higher-power one.	Frequent switching of high-power loads under normal operating conditions.	Automatically interrupts current flow during a fault (e.g., overload or short circuit).
Load Capacity	Handles low currents, typically under 10 A.	Handles high currents, often more than 10 A and up to thousands of amps.	Rated for a wide range of currents, from household use to industrial applications.
Operation	Uses an electromagnetic coil to switch contacts. Latching relays can maintain their position without continuous power.	Uses an electromagnetic coil that must remain energized to keep contacts closed. De-energizing the coil opens the contacts.	Uses a thermal-magnetic or electronic trip unit that automatically opens contacts during a fault. Also has a manual lever for switching.
Contact Status	Can have normally open (NO) and/or normally closed (NC) contacts.	Most configurations use normally open contacts.	Designed to be normally closed during operation but open instantly during a fault.
Arc Suppression	Not typically equipped with arc suppression because it handles low power.	Contains arc chutes to safely extinguish arcs created when switching high currents.	Uses air, oil, gas, or vacuum arc-quenching mechanisms to safely interrupt large fault currents.

Continued on next page

Table 1.1 – continued from previous page

Feature	Relay	Contactors	Circuit Breaker
Reset	Returns to its original state when control signal is removed.	Returns to its original state when control signal is removed.	Must be manually or automatically reset after a fault.
Protection	Provides no protection against overcurrent—used only for control and sensing.	Requires separate overload relay for protection.	Includes built-in thermal and magnetic trip mechanisms for overcurrent and short-circuit protection.

1.1.4 Information Sheet: Circuit Breaker Selection and Rating Calculation

To properly size a circuit breaker, the following parameters must be considered:

- Full Load Current (FLC) of the load
- Starting (Inrush) Current
- Short Circuit Current (I_{sc})
- Safety Factor
- Breaker Type (MCB, MCCB, RCCB, etc.)

Formulas for Circuit Breaker Rating Calculation

Full Load Current (FLC) Calculation

$$I_{FL} = \sqrt{\frac{P}{3 \times V \times \eta \times PF}}$$

Where:

- I_{FL} = Full Load Current (A)
- P = Motor Power (W or kW)
- V = Supply Voltage (V)
- η = Efficiency (decimal)
- PF = Power Factor (decimal)

Circuit Breaker Current Rating

$$I_{CB} = I_{FL} \times K_s$$

Where:

- I_{CB} = Rated Current of the Circuit Breaker (A)
- K_s = Safety Factor (typically 1.1 to 1.25 depending on application and inrush current characteristics)

Short Circuit Current (I_{sc}) Calculation

$$I_{sc} = \frac{V}{Z_{sc}}$$

Where:

- I_{sc} = Short Circuit Current (A)
- V = Line-to-Line Voltage (V)
- Z_{sc} = Impedance of the short-circuit path (Ω)

Breaking Capacity of Circuit Breaker The breaking (interrupting) capacity must satisfy:

$$I_{breaking} \geq I_{sc}$$

Example: Induction Motor Circuit Breaker Calculation

Given Data:

$$\begin{aligned} P &= 15 \text{ kW (Three-Phase)} \\ V &= 415 \text{ V} \\ \eta &= 0.9 \\ PF &= 0.85 \\ Z_{sc} &= 0.05 \Omega \end{aligned}$$

Step 1: Calculate Full Load Current

$$I_{FL} = \sqrt{\frac{15,000}{3 \times 415 \times 0.9 \times 0.85}} = 27.3 \text{ A}$$

Step 2: Apply Safety Factor

$$I_{CB} = 27.3 \times 1.25 = 34.1 \text{ A}$$

Select the nearest standard breaker rating:

$$I_{rated} = 32 \text{ A or } 40 \text{ A (MCCB)}$$

Step 3: Calculate Short Circuit Current

$$I_{sc} = \frac{415}{0.05} = 8,300 \text{ A} = 8.3 \text{ kA}$$

Therefore, the breaker's breaking capacity should be:

$$I_{breaking} \geq 10 \text{ kA}$$

Table 1.2: Final MCCB Selection Parameters

Parameter	Specification
Breaker Type	MCCB
Rated Current	32 A
Breaking Capacity	≥ 10 kA
Trip Curve	Type D (for high inrush currents)

Final Selection of Circuit Breaker

1.1.5 Information Sheet: Circuit Breaker Trip Curve

Circuit breaker trip curves classify how quickly a breaker will trip in response to a short-circuit current.

- **B Curve:** Trips between 3 and 5 times the rated current, ideal for resistive loads such as lighting and general residential electronics.
- **C Curve:** Trips between 5 and 10 times the rated current, used for mixed loads, commercial applications, and devices with small inrush currents such as small motors and pumps.
- **D Curve:** Trips between 10 and 20 times the rated current, designed for high inrush current applications such as large motors, transformers, and heavy machinery.

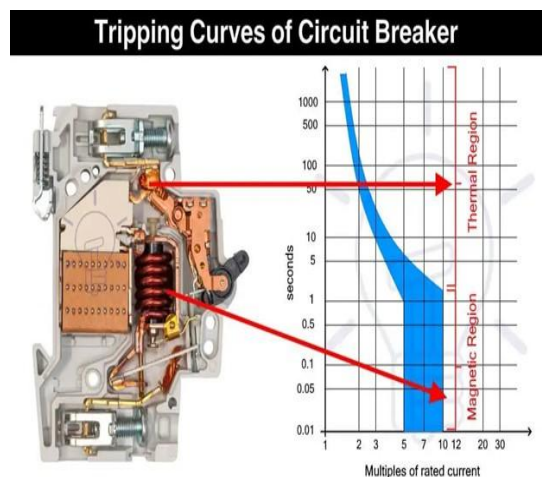


Figure 1.5: Example of Circuit Breaker Trip Curve

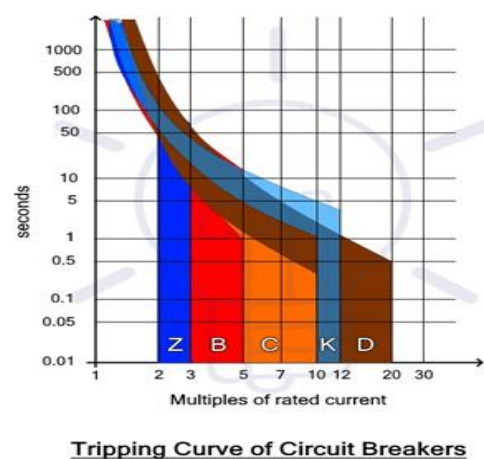


Figure 1.6: Comparison of B, C, and D Trip Curves

1.1.6 Information Sheet: Comparison of Protection Devices

Circuit breakers serve a protective function similar to high-capacity thermal/magnetic relays and are crucial components used alongside control relays.

1.1 Learning Outcome: Interpret Relay Logic Operation



Figure 1.7: Typical arrangement of MCB, MCCB, ELCB, and RCCB in an electrical distribution panel

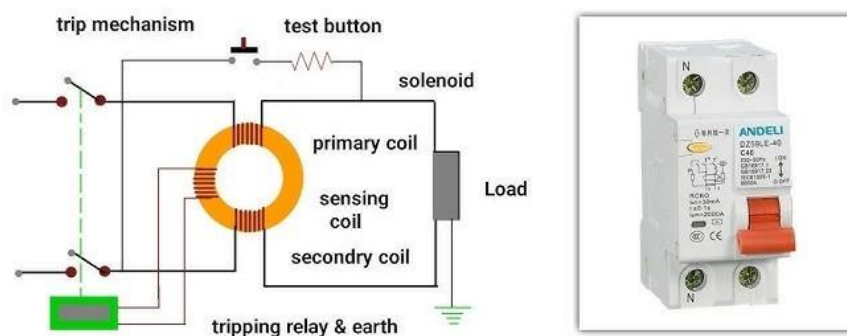


Figure 1.8: Residual current flow and RCCB protection mechanism

1.1.7 Information Sheet: Earth Leakage Current and Protection

Learning Objectives

After studying this information sheet, learners will be able to:

- Explain the concept and causes of earth leakage current.
- Identify protective devices used against earth faults.
- Describe the working principle of RCCB/RCD.
- Understand the sensitivity, coordination, and limitations of earth leakage protection devices.

1. Definition of Earth Leakage Current

Earth leakage current is the unwanted flow of electric current from a live conductor to the earth (ground) through unintended paths. It usually occurs due to insulation damage, moisture, faulty wiring, or accidental contact with live parts.

Earth leakage faults are typically of lower amplitude compared to short circuits but can cause serious electric shocks or fire hazards if not detected.

Table 1.3: Comparison of Protection Devices

Parameter	MCB (Miniature Circuit Breaker)	MCCB (Moulded Case Circuit Breaker)	ELCB (Earth Leakage Circuit Breaker)	RCCB (Residual Current Circuit Breaker)
Primary Function	Protects against short circuits and overloads.	Protects against short circuits, overloads, and high-capacity faults.	Detects earth leakage current.	Protects against leakage current (differential current) and electric shocks.
Trip Mechanism	Thermal and magnetic.	Thermal and magnetic, often with electronic trip units.	Voltage sensing for earth leakage.	Senses differential current between live and neutral conductors.
Overload / SC Protection	Yes.	Yes.	No.	No (requires MCB/MCCB in series).
Current Rating	Typically 6–63 A, up to 125 A.	Typically 64–800 A, up to 2500 A.	5–100 A (load capacity).	16–100 A (load capacity).
Trip Adjustability	Trip settings are fixed.	Trip settings are adjustable.	N/A.	Adjustable residual current rating (e.g., 30 mA, 100 mA).
Typical Usage	Residential and small commercial circuits (lighting, sockets).	Industrial panels, heavy-duty loads, main circuit protection.	Older installations, largely replaced by RCCB in modern systems.	Where human safety is critical, such as homes and offices.

2. Protection Against Earth Leakage

Protection against insulation or earth faults is achieved by **Residual Current Devices (RCDs)** or **Residual Current Circuit Breakers (RCCBs)**, which detect current imbalance and quickly disconnect the circuit.

Note:

- RCCBs do not provide overcurrent protection. They must always be installed in combination with an MCB or MCCB to protect against overload and short-circuit faults.
- RCBOs are preferred where combined protection (leakage + overload) is required or where space saving is needed.
- Ensure proper earthing for accurate operation.

1.1 Learning Outcome: Interpret Relay Logic Operation

- Avoid nuisance tripping by selecting appropriate sensitivity and rated current.

3. Working Principle of RCCB

The RCCB operates on the principle of residual current detection using a toroidal (core-balanced) current transformer.

a. Detection Mechanism Normal Condition: The current through the live (phase) and neutral conductors is equal and opposite. The net magnetic flux in the RCCB's core is zero, and the relay remains ON.

Fault Condition: If current leaks through a person or insulation fault to earth, the return current through the neutral conductor decreases. This imbalance produces a residual current (I_{Δ}) which induces a voltage in the sensing coil.

b. Tripping Action When the residual current exceeds the device's rated limit ($I_{\Delta n}$), the sensing coil energizes the trip mechanism.

The contacts open within **20-40 milliseconds**, disconnecting the supply instantly to prevent electric shock or fire.

4. Sensitivity and Ratings of RCCBs

Table 1.4: Typical Sensitivity Ratings of RCCBs

Application	Typical Sensitivity ($I_{\Delta n}$)	Purpose
Personal Protection	30 mA	Prevents electric shock by interrupting small leakage currents through the human body.
Equipment Protection	100 mA	Prevents equipment damage caused by moderate leakage currents.
Fire Protection	300 mA	Prevents fires caused by insulation failure and large leakage currents.

Information Sheet: Checking Contact Status Using a Multimeter



Figure 1.9: Testing relay contacts using a multimeter

Procedure

1. Set the multimeter to **continuity** or **resistance** mode.
2. Place the probes on the **NO** (Normally Open) and **COM** (Common) terminals — the reading should indicate an **open circuit** when the relay is de-energized.
3. Energize the relay coil and check that the **NO contact closes** (continuity or low resistance is shown).
4. Repeat the test for the **NC** (Normally Closed) and **COM** terminals — these should be **closed when de-energized** and **open when energized**.

Safety Precautions

- Ensure the **power supply is disconnected** before checking relay contacts.
- Never measure resistance while voltage is applied — it can damage the multimeter and cause electric shock.

1.1.8 Information Sheet: Boolean Logic Operation Using Relay

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1



"OR" logic function

A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0



"NAND" logic function

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1



"AND" logic function

A	Output
0	1
1	0



"NOT" logic function

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0



"NOR" logic function

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0



"Exclusive OR" function

Figure 1.10: Boolean logic truth table and symbols

Relays can effectively implement **Boolean logic operations** by configuring the arrangement of their contacts and coils. In a relay control circuit, logical relationships between input signals (switches, sensors, etc.) determine the activation of an output (load or lamp).

1.1.9 Job Sheet - Construct Boolean Logic Gates Using Relays

Section	Content
Job Title	Construct Boolean Logic Gates Using Relays
Personal Protective Equipment (PPE)	Safety gloves, apron, safety goggles, safety shoes, and helmet must be worn while performing wiring and testing operations.
Materials Required	SPDT/DPDT relays (6 units), NO and NC push button switches, indicator lamps or LEDs (6 units), 24 V DC power supply, connecting wires, lugs, terminal blocks, and breadboard or control panel board.
Tools and Equipment	Screwdriver set, wire stripper and cutter, multimeter, crimping tool, measuring tape, and test probes or jumpers.
Procedure	1. Preparation and Safety - Verify that the 24 V DC power supply is properly connected and stable. - Check all relay coils and contacts for correct pin configuration (COM, NO, NC). - Ensure all switches are in OFF position and wiring area is clean. - Confirm no short circuits between relay terminals. 2. Circuit Construction Steps a) <i>AND Gate</i>: Connect two NO contacts in series with a lamp. Energize relays A and B using push buttons. Lamp turns ON only when both relays are energized. b) <i>OR Gate</i>: Connect two NO contacts in parallel. Lamp turns ON when either Input A or Input B relay energizes. c) <i>NOT Gate</i>: Use a NC contact in series with the lamp. Lamp remains ON until Input A relay energizes and opens the contact. d) <i>NAND Gate</i>: Construct AND logic and replace final NO contact with a NC contact for output inversion. e) <i>NOR Gate</i>: Construct OR logic and use NC contacts instead of NO to invert the output. f) <i>XOR Gate</i>: Use two relays combining NO and NC contacts to achieve output HIGH only when A$\neq$ B.
<i>Continued on next page</i>	

Module-1: Relay Control and Motor Operation

Section	Content
3. Testing and Verification	
Procedure	- Apply 24 V DC and test each logic configuration one by one. - Observe the lamp output and compare with the expected truth table. - Record test results in the observation table.
Notes	- Always disconnect power before modifying connections. - Mark each input and output clearly for identification. - Verify correct relay coil polarity and contact numbering. - Avoid cross-connections between coil and contact circuits. - Use multimeter to check contact continuity after each wiring step.
Key Technical Points	- Relays simulate digital logic by switching contacts based on input signals. - Boolean logic functions can be physically realized using relay contact arrangements. - Contact types (NO/NC) determine the logical behavior (normal/inverted). - Circuit verification should match the Boolean truth table for each logic gate.

Specification Sheet

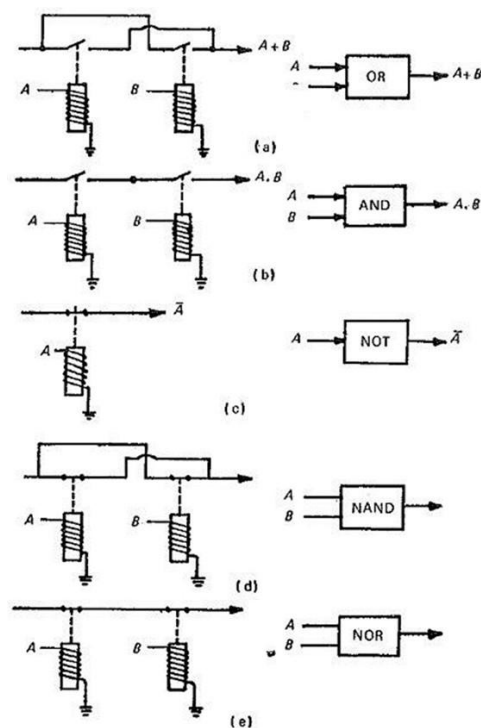


Figure 1.11: Basic relay logic circuit

1.1.10 Job Sheet - Perform 3-Phase Motor Start and Stop Operations Using NO/NC Switch and Contactor Latching Circuit

Section	Content
Job Title	Perform 3-Phase Motor Start and Stop Operations Using NO/NC Switch and Magnetic Contactor
Personal Protective Equipment (PPE)	Safety gloves, apron, safety goggles, safety shoes, and helmet must be worn during wiring and testing operations.
Materials Required	3-Phase induction motor (0.75 HP), magnetic contactor (AC-3 rated), thermal overload relay, NO (Start) push button, NC (Stop) push button, indicator lamp (optional), 3-phase power supply (415 V AC), cables, lugs, terminal blocks, PVC channel, and control panel board.
Tools and Equipment	Screwdriver set, combination pliers, wire stripper and crimping tool, multimeter, measuring tape, drill machine with bits, cable ties, and wire markers.
Procedure	1. Preparation and Safety - Study the control and power wiring diagrams carefully before starting. - Ensure the main power supply is disconnected. - Verify all components (motor, contactor, overload relay, switches) are rated for 415 V AC operation. - Confirm PPE is worn and work area is safe for electrical testing. 2. Control Circuit Wiring - Connect the NC (Stop) push button in series with the contactor coil circuit. - Connect the NO (Start) push button in parallel with the contactor's auxiliary NO contact to form the latching (self-holding) circuit. - Wire one side of the push button circuit to the control phase (L1) and the other side to contactor coil terminal A1. - Connect the other coil terminal (A2) to the neutral line. - Insert the thermal overload relay's NC contact in series with the coil to provide motor protection.
<i>Continued on next page</i>	

Section	Content
textbfProcedure	3. Power Circuit Wiring - Connect the three-phase supply lines (L1, L2, L3) to the contactor input terminals. - Connect the motor terminals to the contactor output terminals through the overload relay. - Verify all power and ground connections are properly tightened.
	4. Testing and Operation - Switch ON the main supply. - Press the START (NO) push button: contactor coil energizes, auxiliary NO contact closes (latching action), and the motor starts. - Press the STOP (NC) push button: coil de-energizes, contactor opens, and the motor stops immediately. - Repeat the test several times to verify reliable operation.
	5. Measurement and Verification - Measure and record control voltage, motor voltage, current, and overload relay setting. - Check coil resistance and ensure correct voltage drop across terminals.
Notes	- Always confirm correct phase sequence before energizing the motor. - Coil voltage must match control supply voltage (e.g., 230 V AC). - Adjust overload relay according to motor's full-load current rating. - Label all terminals and ensure neat wire dressing for safety and maintenance. - Disconnect power supply before making any wiring modifications.
Key Technical Points	- The latching circuit allows the motor to remain ON even after the START button is released. - STOP button provides immediate interruption of control circuit. - Thermal overload relay ensures protection against overcurrent. - Control and power circuits must be isolated for safe operation. - Magnetic contactor provides remote switching and electrical interlocking in motor control systems.

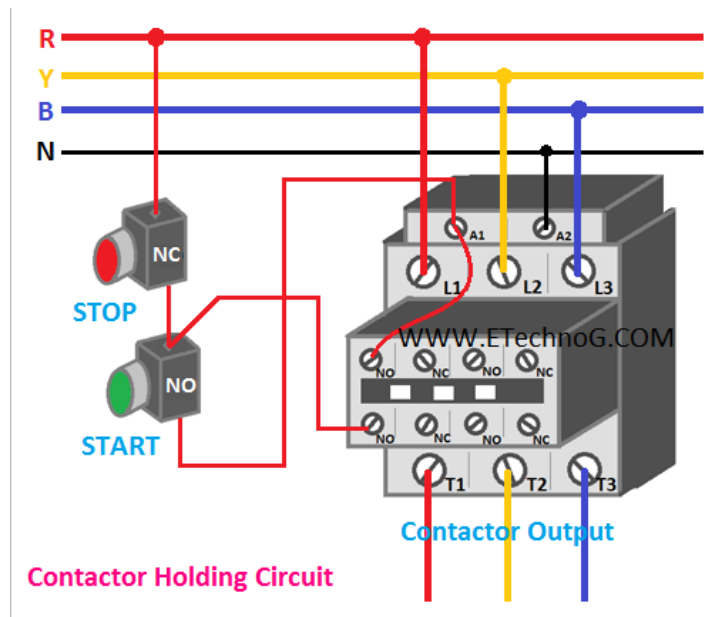
Specification Sheet

Figure 1.12: Contactor latching circuit

1.1.11 Job Sheet - Construct and Test a Direct-On-Line (DOL) Starter Circuit

Section	Content
Job Title	Construct and Test a Direct-On-Line (DOL) Starter Circuit
Personal Protective Equipment (PPE)	Safety gloves, apron, safety goggles, safety shoes, and helmet must be worn while performing wiring and testing operations.
Materials Required	3-phase induction motor (2–5 HP), magnetic contactor (AC-3 rated), thermal overload relay (rated to motor FLC), NO (Start) push button, NC (Stop) push button, indicator lamp (optional), control transformer (if required for 24 V or 110 V control), cables, lugs, terminal blocks, and PVC channel.
Tools and Equipment	Screwdriver set, wire stripper and crimping tool, combination pliers, multimeter, measuring tape, drill machine with bits, electrical knife, cable ties, and wire markers.

Continued on next page

Section	Content
Procedure	1. Preparation and Safety - Study the DOL starter schematic and wiring diagram. - Verify the motor nameplate data (voltage, current, HP, power factor). - Ensure that power is disconnected before wiring. - Wear proper PPE and confirm that all tools are insulated and functional.
	2. Control Circuit Wiring - Connect the NC (Stop) push button in series with the NO (Start) push button. - Connect the Start button to the control supply line (L) and link its output to the contactor coil terminal A1. - Wire the contactor's auxiliary NO contact in parallel with the Start button to create a self-holding circuit. - Connect coil terminal A2 to the neutral (N). - Insert the overload relay's NC contact in series with the coil to provide overload protection.
	3. Power Circuit Wiring - Connect three-phase supply lines (L1, L2, L3) to the input terminals of the contactor. - Connect the overload relay output terminals to the motor terminals (U, V, W). - Ensure proper earthing of the motor frame and panel.
	4. Testing and Operation - Check all wiring connections for tightness and accuracy. - Energize the control supply. - Press the START push button: the contactor coil energizes, auxiliary contact latches, and the motor starts. - Press the STOP push button: the contactor de-energizes and the motor stops immediately. - Test overload relay trip using manual test (if provided).
	5. Measurement and Verification - Measure and record supply voltage, motor current, coil voltage, and overload setting. - Verify that the contactor coil resistance matches its rated value. - Confirm correct operation of Start/Stop and overload protection circuits.

Continued on next page

1.1 Learning Outcome: Interpret Relay Logic Operation

Section	Content
Notes	- The overload relay range must match the motor's full-load current. - Confirm correct phase sequence before starting the motor. - Match control voltage (24 V/110 V/230 V) to contactor coil rating. - Ensure all mechanical and electrical interlocks are functioning. - Maintain neat wire dressing and label all terminals.
Key Technical Points	- The DOL starter is the simplest and most commonly used motor starting method. - Self-holding contact maintains motor operation after releasing the START button. - Thermal overload relay protects the motor from overcurrent and phase loss. - Proper earthing ensures personnel and equipment safety. - The DOL circuit serves as the foundation for advanced starters such as Star-Delta and Forward-Reverse systems.

Specification Sheet

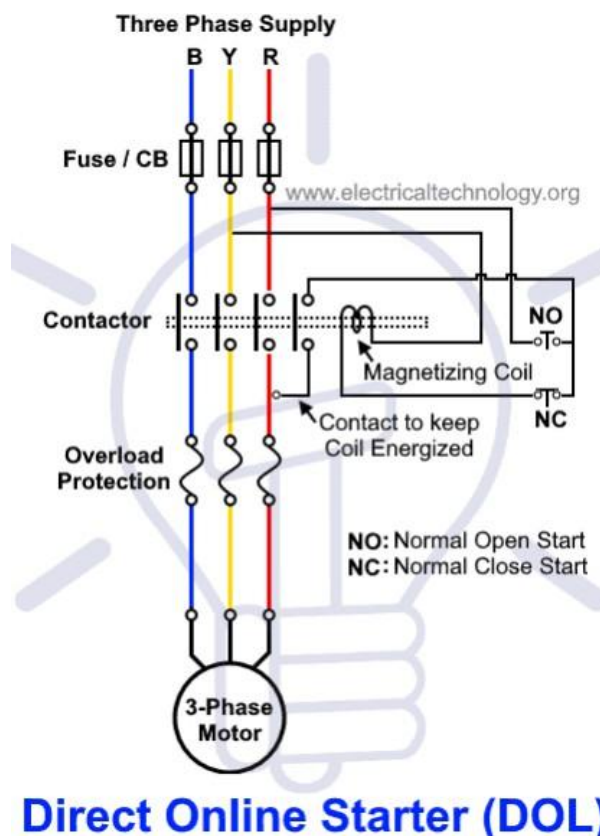


Figure 1.13: DOL starter circuit

1.1.12 Self-Check Questions

Table 1.8: Self-Check Questions

No.	Self-Check Question
1	What is a relay and how does it operate?
2	What is the difference between NO and NC contacts?
3	What are the main components of a relay?
4	What happens when relay coil voltage is applied?
5	Draw or describe how a relay can be used to control a lamp through a push button.
6	What is the purpose of interlocking in relay circuits?
7	What kind of logical operations can be implemented using relays?
8	Why is a relay preferred in control circuits over manual switches for automation?

1.1.13 Answer Key

Table 1.9: Answer Key for Self-Check Questions

No.	Self-Check Question	Expected Answer / Guidance
1	What is a relay and how does it operate?	A relay is an electromechanical switch that uses an electromagnetic coil to open or close contacts in a circuit.
2	What is the difference between NO and NC contacts?	NO (Normally Open) contacts close when the relay is energized; NC (Normally Closed) contacts open when energized.
3	What are the main components of a relay?	Coil, armature, spring, movable contact, and fixed contacts.
4	What happens when relay coil voltage is applied?	The coil generates a magnetic field that pulls the armature, changing the contact state.
5	Draw or describe how a relay can be used to control a lamp through a push button.	When the push button energizes the relay coil, its NO contact closes, completing the lamp circuit.
6	What is the purpose of interlocking in relay circuits?	Prevents two conflicting operations from occurring simultaneously (e.g., forward/reverse operation).
7	What kind of logical operations can be implemented using relays?	AND, OR, NOT, NOR, NAND logic gates can be implemented using series and parallel relay contacts.
8	Why is a relay preferred in control circuits over manual switches for automation?	Relays allow low-voltage control of high-power circuits and enable automatic control logic.

1.2 Learning Outcome: Perform mechanical switching operation

1.2.1 Information Sheet: Mechanical Switches

A **mechanical switch** is an electromechanical device used to make, break, or change connections in an electrical circuit through physical movement (mechanical actuation). When actuated, it either allows or interrupts the current flow to control an electrical load.

Mechanical switches are widely used in industrial control panels, motor circuits, machine tools, and automation systems for signal or power control.

Working Principle

A mechanical switch operates by bringing conductive contacts together or separating them using a mechanical force (manual or automatic).

- When the contacts touch → the circuit **closes** (current flows).
- When the contacts separate → the circuit **opens** (current stops).

Mechanical switches can be manually operated (like push buttons and toggle switches) or automatically operated (like limit switches or pressure switches). They form the foundation of relay logic and PLC input control in industrial automation.

1.2.2 Information Sheet: Major Types of Mechanical Switches

This section describes the major categories of mechanical switches commonly used in electrical and industrial control systems. Each switch type differs in actuation method, contact configuration, and application.

A. Push Button Switch

Description: Operated by pressing a button; returns to its original state upon release (momentary type). **Types:**

- **NO (Normally Open):** Circuit closes when pressed.
- **NC (Normally Closed):** Circuit opens when pressed.
- **Dual (NO+NC):** Both NO and NC contacts in one assembly.

Applications: Start/Stop buttons in motor starters, machine controls, and emergency stops.



Figure 1.14: Push Button Switch



Figure 1.15: Toggle Switch

B. Toggle Switch

Description: Has a lever or handle that can be flipped between two or more positions.

Operation: Maintains position (latching) until toggled again. **Contact Configurations:** SPST, SPDT, DPDT, etc. **Applications:** Used for controlling power circuits and mode selection.

C. Selector Switch

Description: Rotary-type switch used to select between multiple operating conditions.

Operation: Each position corresponds to a specific circuit path. **Example:** Auto/Manual selector, Forward/Reverse mode selector.



Figure 1.16: Selector Switch



Figure 1.17: Limit Switch

D. Limit Switch

Description: Actuated by the physical movement of a machine part or object. **Structure:** Contains a lever, plunger, or roller actuator linked to an internal contact mechanism.

Applications: Used in machine automation to detect motion limits, presence, or position.

E. Foot (Pedal) Switch

Description: Operated by foot pressure to free the operator's hands. **Types:** Momentary or maintained action. **Applications:** Sewing machines, metalworking equipment, and conveyor controls.



Figure 1.18: Foot (Pedal) Switch



Figure 1.19: Float Switch

F. Float Switch

Description: Detects the level of liquid in tanks using a buoyant float connected to a mechanical switch. **Applications:** Used in pumps, sump systems, and water treatment plants.

1.2.3 Information Sheet: Contact Configurations

Mechanical switches are identified by their contact arrangement, which determines how many circuits can be controlled and how the switching occurs.

Table 1.10: Common Contact Configurations of Mechanical Switches

Type	Full Form			Description
SPST	Single	Pole	Single	One input, one output — acts as a simple ON/OFF switch.
SPDT	Single	Pole	Double	One input, two outputs — allows switching between two outputs (changeover).
DPST	Double	Pole	Single	Two separate circuits switched together simultaneously.
DPDT	Double	Pole	Double	Two changeover circuits — can reverse polarity or direction (e.g., in motor control).

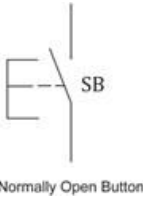
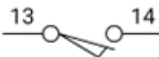
1.2.4 Information Sheet: Symbol and Circuit Representation

Mechanical switches are represented in circuit diagrams using standardized electrical symbols as per IEC and ISO conventions. These symbols visually describe the contact

1.2 Learning Outcome: Perform mechanical switching operation

behavior and function of the switch.

Table 1.11: Symbols and Circuit Representations of Common Mechanical Switches

Switch Type	Symbol	Description
Push Button (NO)		Closes the circuit when pressed; returns to open state when released.
Push Button (NC)		Opens the circuit when pressed; returns to closed state when released.
Limit Switch		Activated by the mechanical movement of a machine part to detect position or limit of travel.

1.2.5 Information Sheet: Reference Standards

This section lists the international standards relevant to the design, operation, and documentation of mechanical switches and control devices.

Table 1.12: Reference Standards Related to Mechanical Switches and Control Devices

Standard	Title	Purpose
IEC 60947-5-1	Low-voltage switchgear and controlgear	Specifies mechanical and electrical performance requirements for control circuit devices such as switches, relays, and sensors.
ISO 13850	Emergency stop device — Functional aspects	Defines design and safety requirements for emergency stop devices, ensuring reliable human-machine safety interfaces.
IEC 60617	Graphical symbols for electrical diagrams	Provides standardized graphical symbols used in circuit diagrams for relays, switches, and control components.

1.2.6 Self-Check Questions

No.	Question
1	What is a mechanical switch?
2	Name four types of mechanical switches used in control systems.
3	What is the difference between a push button and a selector switch?
4	How does a limit switch work?
5	What are NO and NC contacts in a mechanical switch?
6	In an industrial start-stop circuit, which type of switches are used for start and stop?
7	Why is it important to label switch terminals and maintain wiring color codes?
8	How does a foot switch improve operational efficiency?

1.2.7 Answer Key

No.	Answer / Guidance
1	A device that manually opens or closes an electrical circuit through physical movement.
2	Push Button, Selector, Limit, and Foot Switch.
3	A push button is momentary; a selector switch stays in position until changed.
4	It operates when an object physically contacts its actuator, changing the contact state.
5	NO contacts are open until actuated; NC contacts are closed until actuated.
6	Start → NO push button; Stop → NC push button.
7	Ensures safety, simplifies maintenance, and meets IEC 60204-1 standards.
8	Allows hands-free operation, improving efficiency in machines like presses or conveyors.

1.3 Learning Outcome: Execute timer and counter application

1.3.1 Information Sheet: Timer and Counter Fundamentals

Timers and counters are essential components in industrial control systems that enable time-based or count-based logic operations. They are widely used in automation, machine sequencing, and production monitoring.

Timers

Timers are used when the control sequence requires introducing a time delay in the activation or deactivation of devices.

- **On-Delay Timer (TON):** Activated after a preset time delay once the input signal is applied. The contacts change state only after the set time has elapsed.
- **Off-Delay Timer (TOF):** Activated immediately when the input signal is applied, but maintains the output ON for a preset delay after the signal is removed.
- **Retentive Timer (RTO):** Retains or “remembers” the accumulated time even if the input signal is lost or power is interrupted. The timing resumes from the last recorded value upon reactivation.



Figure 1.20: Manual Timer

Example Application: A delay timer is used in an automatic lighting system where lights turn ON a few seconds after motion is detected.

Counters

Counters are employed in applications that require event tracking or counting of objects, signals, or operations. They are integral to automation systems such as packaging lines, production monitoring, and batching operations.

- **Up Counter (CTU):** Increments its count by one each time a signal (such as a sensor pulse) is received. Commonly used to track the number of items produced or detected.

1.3 Learning Outcome: Execute timer and counter application

- **Down Counter (CTD):** Counts down from a preset number to zero, typically used for operations that must terminate after a specific count (e.g., a machine cycle that runs ten times).

Example Application: An up counter counts each product passing on a conveyor, and when a preset number is reached, it triggers a diverter or an alarm.



Figure 1.21: Illustration of Manual Counter

1.3.2 Job Sheet - Construct and Test Sequential Operation Circuit Using Multiple Contactors, Timer, and Counter in Motor Control

Section	Content
Job Title	Construct and Test Sequential Operation Circuit Using Multiple Contactors, Timer, and Counter in Motor Control Application
Personal Protective Equipment (PPE)	Safety gloves, apron, safety goggles, safety shoes, and helmet must be worn during wiring and testing operations.
Materials Required	Three magnetic contactors (C1, C2, C3 - AC-3 rated), two ON-delay timers (T1, T2), one up-counter unit (presettable type), Start (NO) and Stop (NC) push buttons, indicator or pilot lamps (3 units for sequence display), control transformer (230 V/24 V if required), 3-phase power supply (415 V), cables, lugs, terminal blocks, and PVC channel.

Continued on next page

Module-1: Relay Control and Motor Operation

Section	Content
Tools and Equipment	Screwdriver set, wire stripper and crimping tool, combination pliers, multimeter, measuring tape, drill machine, cable ties, electrical knife, and wire marker.
Procedure	1. Preparation and Safety – Study the sequential control circuit diagram thoroughly. – Ensure the power supply is switched OFF before wiring. – Verify timer and counter voltage ratings match the control circuit (typically 230 V AC). – Collect all tools, materials, and wear appropriate PPE.
Procedure (cont.)	2. Sequence Logic Description – Pressing the Start button energizes Contactor C1 and starts Motor 1. – After the preset delay of Timer T1, Contactor C2 energizes and starts Motor 2. – After the delay of Timer T2, Contactor C3 energizes and starts Motor 3. – Each full cycle increments the counter by one count. – When the counter reaches the preset value, all contactors de-energize and the system stops automatically. – The Stop button or counter reset deactivates all outputs and resets the sequence.
Procedure (cont.)	3. Control Circuit Wiring – Connect the Stop (NC) and Start (NO) push buttons in series with the control supply. – Wire C1 coil directly after the Start button for first-stage activation. – Connect Timer T1 in parallel with C1 coil; its delay output energizes C2 coil. – Connect Timer T2 in parallel with C2 coil; its delay output energizes C3 coil. – Connect the counter input to one of the contactor output lines (e.g., C3) for counting each completed sequence. – Configure the counter's output contact to open the control circuit upon reaching the preset count. – Use auxiliary interlocks between contactors to prevent simultaneous energization. – Connect pilot lamps in parallel with each contactor coil to visualize the active stage.

Continued on next page

1.3 Learning Outcome: Execute timer and counter application

Section	Content
Procedure (cont.)	4. Power Circuit Wiring – Connect the 3-phase supply (L1, L2, L3) to the contactor input terminals. – Connect the output terminals of each contactor to its respective motor (M1, M2, M3) through overload relays. – Ensure all motors or pilot lamps are properly grounded and phase connections are correct.
Procedure (cont.)	5. Testing and Operation – Check all wiring continuity and insulation resistance. – Energize the control circuit and press the Start button. – Observe sequential operation: C1 energizes first, followed by C2 and C3 after timer delays. – Confirm that the counter increments correctly after each full sequence. – Verify that the system stops automatically when the preset count is reached. – Press Stop or Reset to restart the system.
Notes	– Timer delays must be adjusted to avoid overlapping of motor starts. – Interlocks ensure that only one contactor is energized at a time. – The counter simulates event- or batch-based process control. – Conduct initial testing with pilot lamps before connecting motors. – Perform all tests under no-load conditions before final operation.
Key Technical Points	– Sequential control enables automated multi-stage operations with time and count-based logic. – Timers provide controlled delays; counters manage repetition or batch control. – Electrical interlocking ensures safe operation between contactors. – Commonly used in conveyor systems, pump control, and automatic production cycles. – The circuit concept resembles PLC-based sequential automation.

Specification Sheet

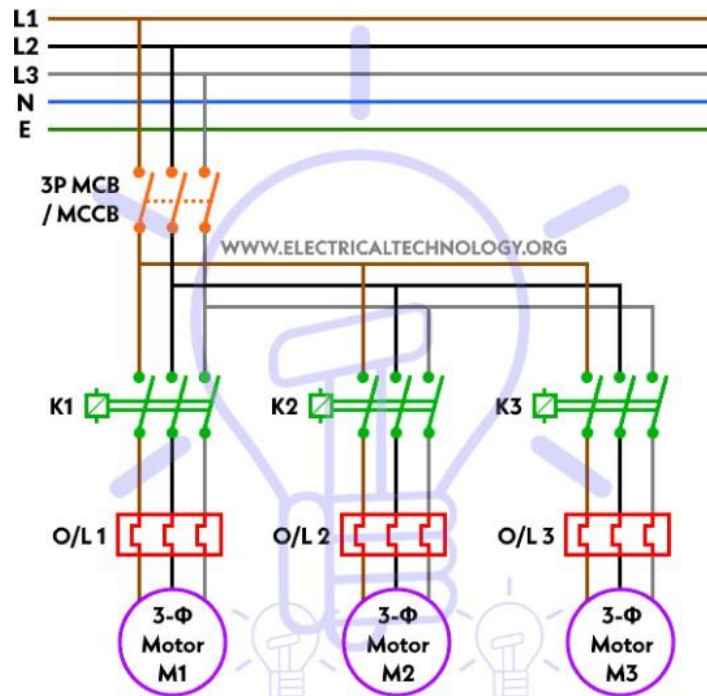


Figure 1.22: Power Circuit Diagram of Sequential Control

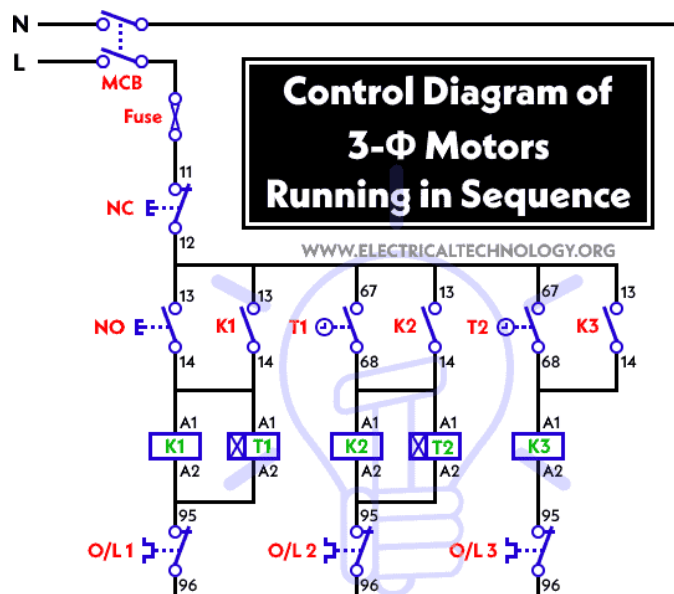


Figure 1.23: Control Circuit Diagram of Sequential Control

1.3.3 Self-Check Questions

No.	Question
1	What is a timer in automation?
2	Name the main types of timers.
3	In a star-delta starter, which timer function is typically used?
4	What is the function of a counter in PLC systems?
5	What are the key parameters of a timer?
6	What are Up and Down counters?
7	How can a timer be used to blink an LED?
8	What is the difference between hardware timer and software timer in PLC?
9	Describe the characteristic operation of an Off-Delay Timer (TOF).
10	What type of application commonly uses an Up Counter (CTU)?

1.3.4 Answer Key

No.	Answer / Guidance
1	A control device that delays turning ON/OFF a circuit after a preset time interval.
2	ON-delay, OFF-delay, Interval, and Cyclic timers.
3	ON-delay timer - used to change connection after the delay period.
4	Counts input pulses or events and performs an action when a preset value is reached.
5	Preset time, elapsed time, and contact type (NO/NC).
6	Up counter increments with each pulse; Down counter decrements from a preset value.
7	Use an ON-delay or pulse timer in cyclic mode to switch LED ON/OFF periodically.
8	Hardware timers are external devices; software timers are internal PLC logic functions.
9	The TOF activates immediately when the input signal is applied, but maintains the output ON for a preset delay after the signal is removed.
10	They are commonly used to track the number of items produced or detected, such as counting each product passing on a conveyor.

1.4 Learning Outcome: Perform forward reverse and star delta operation

1.4.1 Information Sheet: Star-Delta Starter

The **Star-Delta (Y-Δ) Starter** is one of the most common and economical methods of starting a three-phase induction motor. Its main purpose is to reduce the **starting current** and **starting torque** of the motor by initially connecting the stator windings in a **Star (Y)** configuration. After a preset time delay, the connection is automatically changed to a **Delta (Δ)** configuration for normal running operation.

Working Principle:

- In the **Star (Y)** connection, each phase receives only $\frac{1}{\sqrt{3}}$ of the line voltage, thus reducing the current to one-third of the direct-on-line (DOL) starting current.
- After the motor reaches approximately 80% of its rated speed, the control circuit switches the winding connections to **Delta (Δ)**, applying full line voltage for normal operation.
- This transition is typically controlled through a timer relay and interlocked contactors to prevent short circuits.

Advantages:

- Reduces starting current and torque.
- Simple and cost-effective design.
- Reliable operation for medium and large motors.

Limitations:

- Not suitable for motors below 5 HP.
- Momentary torque dip during Star-Delta transition.
- Unsuitable for applications requiring high starting torque.

Information Sheet: Principle of Operation

A three-phase induction motor draws a high inrush current (6–8 times the full-load current) when started directly (DOL). In **Star-Delta** starting, the motor is first connected in the **Star (Y)** configuration, which reduces the voltage per winding to $\frac{1}{\sqrt{3}}$ (approximately 58%) of the line voltage. After the motor reaches approximately 80% of its rated speed, the winding connections are switched to **Delta (Δ)**, allowing the motor to run at full voltage.

Table: Voltage and Current Relationship in Star and Delta Connections

Parameter	Star Connection	Delta Connection
Line Voltage	$V_L = \sqrt{3} \times V_{Ph}$	$V_{Ph} = V_L$
Line Current	$I_L = I_{Ph}$	$I_L = \sqrt{3} \times I_{Ph}$
Torque	$\frac{1}{3}$ or full-load torque	100% rated torque

Voltage and Current Relationship**1.4.2 Job Sheet - Construct and Test Forward-Reverse Motor Starter Circuit**

Section	Content
Job Title	Construct and Test Forward-Reverse Motor Starter Circuit
Personal Protective Equipment (PPE)	Safety gloves, apron, safety goggles, safety shoes, and helmet must be worn during wiring and testing operations.
Materials Required	3-phase induction motor (0.75 HP), two magnetic contactors (AC-3 rated, identical coil voltage), thermal overload relay (matched to motor FLC), Start-Forward (NO) push button, Start-Reverse (NO) push button, Stop (NC) push button, indicator lamps (optional for forward/reverse), control fuse or MCB, control transformer (230/24 V if required), power and control cables, lugs, terminal blocks, and PVC ducts.
Tools and Equipment	Screwdriver set, wire stripper and crimping tool, combination pliers, multimeter, measuring tape, drill machine, cable ties, electrical knife, and wire marker.
Procedure	Part 1 - Preparation and Safety - Study wiring diagrams and gather PPE. - Verify contactor coil voltage and motor direction. - Confirm main power is OFF before wiring. Part 2 - Control Circuit Wiring - Connect Stop (NC) button in series with Start buttons. - Connect Forward (NO) button to energize Contactor C1. - Connect Reverse (NO) button to energize Contactor C2. - Add self-holding and interlocking contacts. - Insert overload relay NC contact in series for protection.

Continued on next page

1.4 Learning Outcome: Perform forward reverse and star delta operation

Section	Content
	Part 3 - Power Circuit Wiring - Connect 3-phase supply lines (L1, L2, L3) to contactors. - Forward contactor output: connect directly to motor. - Reverse contactor output: swap two phases (e.g., V, W). - Route outputs through overload relay to motor.
	Part 4 - Testing and Operation - Check continuity and terminal tightness. - Press Start and Stop buttons, observe motor direction. - Verify interlock and overload functions.
Notes	- The overload relay must be set to match the motor's full-load current (FLC). - Always confirm correct phase sequence before energizing the circuit. - Match control voltage to the contactor coil rating (24 V, 110 V, or 230 V). - Verify both electrical and mechanical interlocks for proper functioning. - Maintain neat wire dressing, proper labeling, and documentation.
Key Technical Points	- The Forward-Reverse starter changes motor direction by swapping two supply phases. - Electrical interlocking ensures that only one contactor operates at a time, preventing short circuits. - Thermal overload relay protects against prolonged overcurrent conditions. - Commonly used in conveyor drives, hoists, and machine tool direction control. - Can be expanded with timers or limit switches for automatic directional control.

Specification Sheet

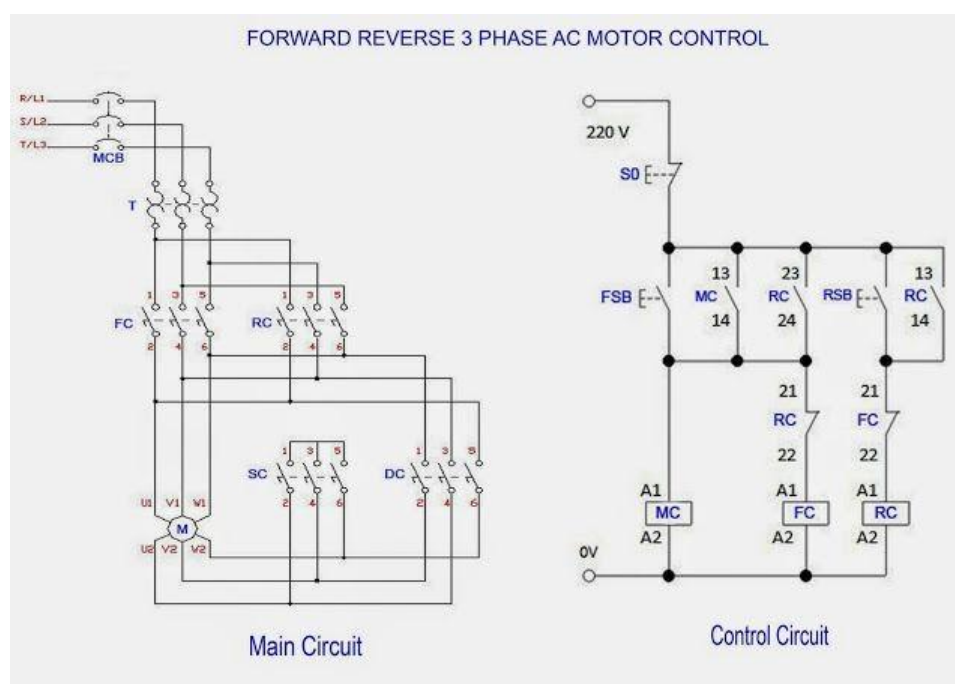


Figure 1.24: Motor forward-reverse operation circuit

1.4.3 Job Sheet - Construct and Test Automatic 3-Phase Star-Delta Starter Circuit

Section	Content
Job Title	Construct and Test Automatic 3-Phase Star-Delta Starter Circuit
Personal Protective Equipment (PPE)	Safety gloves, apron, safety goggles, safety shoes, and helmet must be worn during wiring and testing operations.
Materials Required	3-phase induction motor (0.75 HP), three magnetic contactors (Main, Star, Delta - AC-3 rated), star-delta timer (on-delay type), thermal overload relay (rated to motor FLC), Start (NO) and Stop (NC) push buttons, indicator lamps (Green - RUN, Red - STOP), control fuse/MCB/MCCB, 3-phase power supply (415 V AC), control cables, lugs, and terminal blocks.
Tools and Equipment	Screwdriver set, wire stripper and crimping tool, combination pliers, multimeter, measuring tape, drill machine, cable ties, electrical knife, and wire marker.

Continued on next page

1.4 Learning Outcome: Perform forward reverse and star delta operation

Section	Content
Procedure	1. Preparation and Safety - Study the star-delta starter schematic and sequence diagram carefully. - Ensure the motor rating, contactor coil voltage, and overload relay range are correct. - Collect all materials and tools and confirm PPE is worn. - Disconnect the main supply before wiring.
	2. System Overview - The starter initially connects the motor windings in a star configuration to limit inrush current. - After a preset delay (set on the timer), the circuit automatically switches to delta connection for full-speed operation. - Electrical interlocking ensures that Star and Delta contactors cannot energize simultaneously.
	3. Control Circuit Wiring - Connect the Stop (NC) and Start (NO) push buttons in series with the Main contactor coil. - Use the Main contactor's auxiliary NO contact as a self-holding circuit parallel to the Start button. - Wire the Star contactor coil through the timer's NO contact and the Delta contactor coil through the timer's NC contact. - Provide cross-interlocks between Star and Delta contactors using their NC auxiliary contacts. - Insert the overload relay's NC contact in series with the control circuit for protection. - Connect indicator lamps (RUN and STOP) across appropriate points for visual status.
	4. Power Circuit Wiring - Connect 3-phase supply lines (L1, L2, L3) to the Main contactor input terminals. - Route outputs from the Main contactor through the overload relay to the motor terminals. - Connect motor windings (U1, V1, W1 - U2, V2, W2) according to the nameplate. - The Star contactor connects U2, V2, and W2 together during start. - The Delta contactor interconnects the windings after timer delay for full-load running. - Ensure proper earthing of the motor and panel.

Continued on next page

Section	Content
	5. Testing and Operation - Energize the control circuit and press the Start button. - The motor starts in Star connection (Main and Star contactors energized). - After the timer delay (typically 5-10 seconds), the Star contactor de-energizes and the Delta contactor energizes, running the motor in Delta mode. - Observe indicator lamps and motor speed transition for smooth operation. - Press the Stop button to de-energize all contactors and stop the motor.
Notes	- Confirm that all contactor coil voltages are correct before energizing. - Verify electrical and mechanical interlocks between Star and Delta contactors. - Adjust timer delay (5-10 seconds) according to motor capacity and load. - Ensure phase sequence and insulation are correct before operation. - Maintain neat wiring, proper labeling, and verify all terminals are tight.
Key Technical Points	- Star-Delta starting reduces motor starting current to roughly one-third of DOL starting current. - The timer governs the automatic changeover from Star to Delta connection. - Electrical interlocks prevent short-circuiting between Star and Delta contactors. - Overload relay and MCB/MCCB provide essential protection against faults. - Star-Delta starters are widely used for large induction motors to ensure smooth acceleration and reduced voltage drop.

Specification Sheet

1.4 Learning Outcome: Perform forward reverse and star delta operation

Control Circuit Diagram of Automatic Star / Delta Starter (Y-Δ) using Timer for 3-Φ Motor

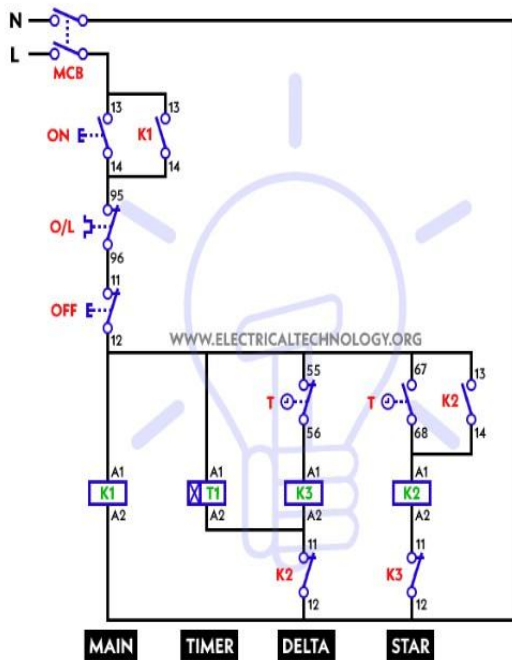


Figure 1.25: Start-Delta starter control circuit

Power Circuit Diagram of Automatic Star / Delta Starter (Y - Δ) for 3-Φ Motor

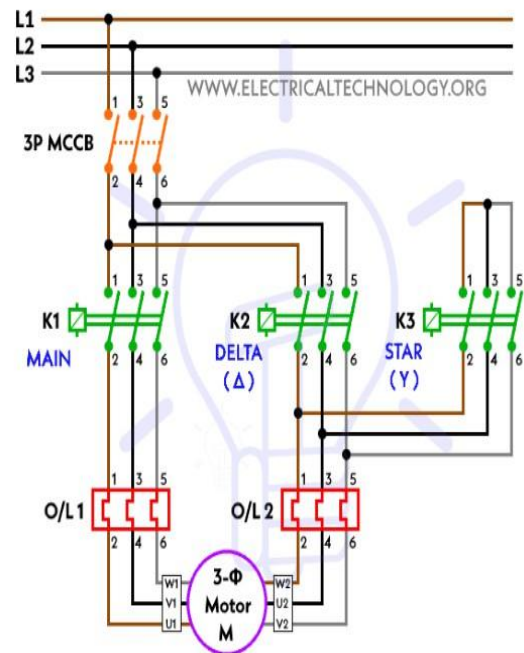


Figure 1.26: Star-Delta starter power circuit

1.4.4 Self-Check Questions

No.	Question
1	Why is interlocking used in a forward-reverse starter?
2	How many contactors are required for a forward-reverse motor circuit?
3	What happens if both contactors in a forward-reverse circuit close together?
4	What is the purpose of the timer in a star-delta starter?
5	How many contactors are used in a star-delta starter?
6	What is the main advantage of a star-delta starter over a DOL starter?
7	What type of switch or push button is used for start and stop control in these circuits?
8	What safety device protects the motor from overload in both circuits?
9	What is the function of auxiliary contacts in a forward-reverse control circuit?
10	Why verify wiring and mechanical interlocks before energizing?

1.4.5 Answer Key

No.	Answer / Guidance
1	To prevent both forward and reverse contactors from energizing at the same time.
2	Two contactors (Forward and Reverse) plus an overload relay.
3	It causes a short circuit between supply phases.
4	To control the time delay between Star and Delta transitions during motor startup.
5	Three (Main, Star, and Delta).
6	It reduces starting current and torque to one-third.
7	Start — NO push button; Stop — NC push button.
8	Thermal Overload Relay (OLR).
9	Provide electrical interlocking and feedback to control logic.
10	Ensures safe operation and prevents equipment damage or electrical hazards.

Module-2: Perform Basic PLC Programming

2.1 Learning Outcome: Install PLC and Associated Wiring

2.1.1 Information Sheet: PLC and Its Components

What is a PLC?

A Programmable Logic Controller (PLC) is a controller that executes a user program to monitor inputs and control outputs of a machine or process. The PLC scans input modules (addresses such as

Components of a PLC

Typical components include CPU, power supply, digital/analog I/O modules, communication interfaces, and backplane/rail.

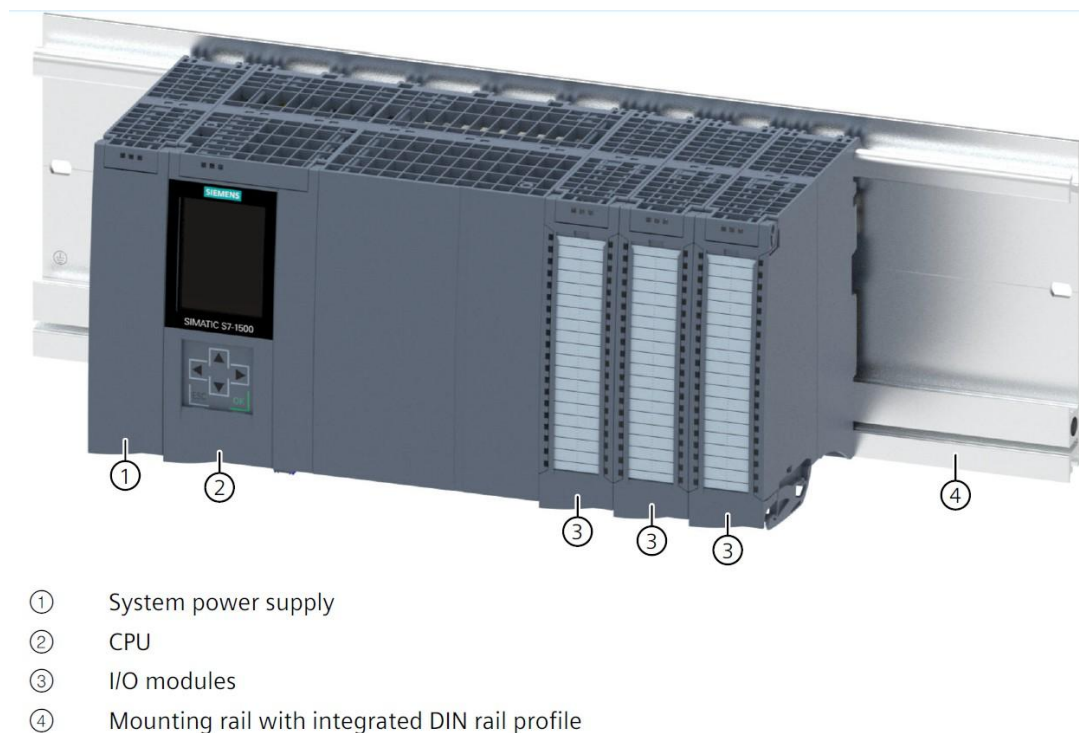


Figure 2.1: PLC system components (example: SIMATIC S7-1500)

SIMATIC S7-1500 Components Overview

The **SIMATIC S7-1500** is a modular controller system designed for the middle to upper performance range. It provides high-speed processing, advanced communication capabilities, and flexible modular expansion. Its key components and functions are summarized below:

Component	Function
Central Processing Unit (CPU)	Executes the user program, processes inputs, generates outputs, and integrates the system power supply (PS). Many CPUs include an integrated display for operation and diagnostics.
Mounting Rail	The physical rack (often with an integrated DIN rail profile) onto which all modules are installed in a single-row configuration.
Load Current Supply (PM)	Supplies 24 V DC power to the CPU's system power supply, the input/output circuits of the I/O modules, sensors, and actuators.
System Power Supply (PS)	Integrated within the CPU or added as a module, it supplies the internal system voltage to the module electronics and LEDs via the backplane bus.
I/O Modules (SM)	Modules for Digital Input (DI), Digital Output (DQ), Analog Input (AI), or Analog Output (AQ) that interface the controller with the physical process.
Communication Modules (CM/CP)	Used to expand communication capabilities, such as adding interfaces for PROFIBUS, PROFINET, or point-to-point connections.
SIMATIC Memory Card	Provides non-volatile load memory for storing the user program, data, and configuration. It is often required for the CPU to operate.

Table 2.1: Key Components and Functions of SIMATIC S7-1500 Controller System

Advantages of PLC over Relay Systems

- **Reliability:** Solid-state design with fewer mechanical parts.
- **Troubleshooting:** Fewer wires; online diagnostics simplify fault-finding.
- **Expandability:** Add functionality by changing the program and modules.
- **Compact size:** Smaller cabinet footprint than equivalent relay logic.

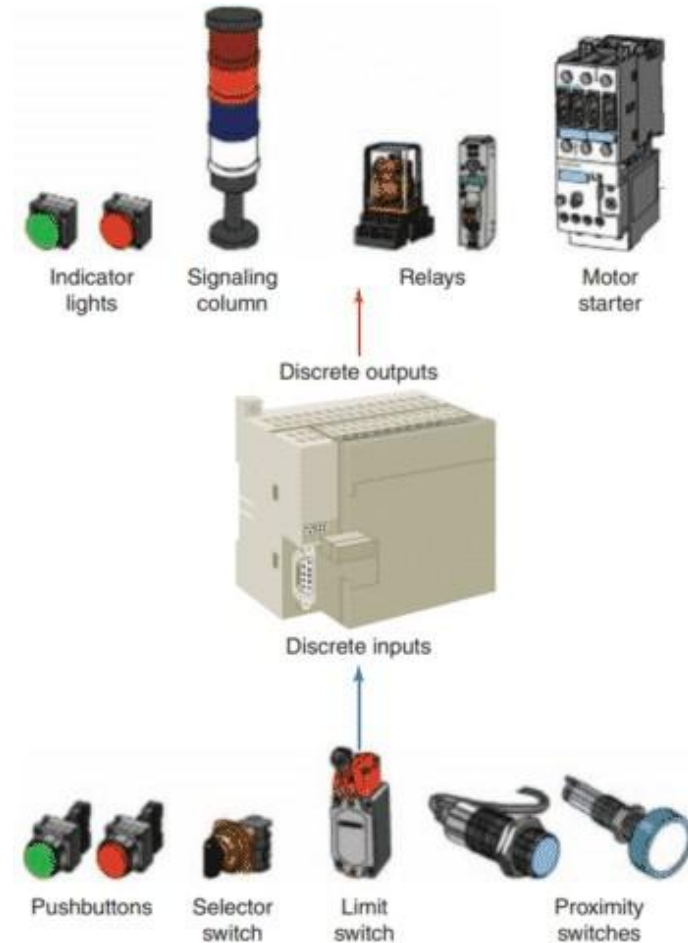
2.1.2 Information Sheet: Input/Output devices for PLCs**PLC Input and Output Devices**

Figure 2.2: Common PLC input and output devices

Programmable Logic Controllers (PLCs) interact with external devices through input and output modules. Input devices send electrical signals to the PLC to represent process conditions, while output devices receive commands from the PLC to perform physical actions.

Device Type	Function	Examples
Input Devices	Provide signals or data to the PLC for processing and decision-making.	Sensors, switches, push buttons, thermocouples, pressure sensors, level sensors (analog input sources).
Output Devices	Receive control signals from the PLC to actuate or perform mechanical/electrical actions.	Actuators, relays, motors, solenoids, high-power devices.

Table 2.2: Classification and Function of PLC Input and Output Devices

2.1.3 Information Sheet: Programming Languages (IEC 61131-3)

- **LAD (Ladder Diagram):** Graphical, relay-logic style.
- **FBD (Function Block Diagram):** Graphical, block-oriented logic.
- **SCL/ST:** Text-based, high-level structured text.

Programming Software and Versions

TIA Portal integrates STEP 7 (PLC) and WinCC (HMI). The project version must be compatible with device firmware. Older projects can be upgraded to newer versions, but newer projects cannot be opened in older software.

Identification Parameters

Always record device part numbers, firmware, and the TIA Portal project version for PLCs and HMIs to ensure compatibility and maintainability.



Figure 2.3: Common PLC input and output devices

2.1.4 Information Sheet: PLC Selection Criteria

Selecting the appropriate **Programmable Logic Controller (PLC)** model is crucial for ensuring system performance, reliability, and cost-effectiveness. Selection should consider technical, operational, and economic factors. The key criteria are described below:

- **Application Requirements:**
 - *Complexity:* Determines whether a simpler Compact PLC (for single-machine control) or a Modular PLC (for complex production lines) is required.
 - *I/O Needs:* Evaluate the number and type of input/output points (digital, analog, or special functions such as high-speed counters or encoder inputs). The PLC must support current needs and allow for expansion.
- **Power Supply:** Choose between *AC-powered* (where mains supply is available) or *DC-powered* PLCs (common in systems with battery backup or mobile setups).
- **Output Type:** Select between:
 - *Relay Outputs:* Suitable for AC/DC switching, slower, used for motors or solenoids.

- *Transistor Outputs*: Fast switching, DC-only, used for high-speed operations.
- **Scan Time**: For applications requiring high-speed processing (e.g., motion control, counting), choose PLCs with fast, deterministic scan cycles.
- **Scalability and Flexibility**: Modular PLCs allow future system upgrades or customization by adding CPU or I/O modules.
- **Environmental Conditions**: Consider the operating environment (temperature, humidity, vibration, dust) and choose PLCs with suitable IP ratings and compact enclosures for tight spaces.
- **Processing Power and Memory**: Ensure sufficient CPU speed and memory capacity (program and data memory) for the application, especially for advanced logic, PID control, or data logging.
- **Programming and Software**: Verify that the PLC supports standard IEC 61131-3 programming languages and ensure the software is user-friendly and cost-effective.
- **Communication Requirements**: The PLC should support industrial communication protocols such as *PROFINET*, *Modbus*, *Ethernet/IP*, or *OPC UA*, and provide options for *remote monitoring* or *SCADA/HMI integration*.
- **Reliability and Redundancy**: For mission-critical applications, choose PLCs with high Mean Time Between Failures (MTBF), redundant CPUs, and hot-swappable modules.
- **Safety and Compliance**: Ensure compliance with relevant standards such as *SIL*, *ISO 13849*, *CE*, *UL*, or *ATEX*, especially in hazardous or safety-critical environments.

2.1.5 Information Sheet: Short Specifications of PLC (S7-1512C-1 PN Example)

The **SIMATIC S7-1512C-1 PN** (Part Number: 6ES7512-1CK01-0AB0) is a compact PLC designed for medium-scale industrial automation applications. It integrates CPU, digital, and analog I/O, making it ideal for standalone or distributed control systems.

2.1.6 Information Sheet: Step-by-Step Process to Install PLC, Related Hardware, and Associated Wiring (SIMATIC S7-1500 Example)

The **SIMATIC S7-1500 system** uses a mounting rail (DIN rail profile) for module installation in a single-row configuration. Modules connect via U-connectors to form a backplane bus, starting from the left side of the rail. Typically, the maximum configuration allows up to 32 modules (Slots 0-31) if a System Power Supply (PS) is used next to the CPU.

Specification	Value	Explanation
CPU Performance	Bit performance: 48 ns	Represents the execution time for basic bit operations, indicating high processing speed and deterministic response.
Working Memory	Program: 250 KB; Data: 1 MB	Memory available for executing logic and storing runtime data.
Storage	Requires SIMATIC Memory Card	Supports memory cards up to 32 GB for load memory, data retention, and firmware.
Interfaces	PN/DP (2-port switch)	PN: PROFINET for HMI, I/O, PLC-to-PLC networking. DP: PROFIBUS/serial (RS232) for field devices.
Integrated Technology Functions	6 HSC (100 kHz), 4 PTO/PWM outputs	Enables precise counting, frequency measurement, and motion/pulse control.
Power Supply	24 V DC	Operates from 19.2 to 28.8 V DC.
Integrated Digital I/O	32 DI (sinking), 32 DO (sourcing)	Supports direct field wiring without separate modules.
Integrated Analog I/O	5 AI, 2 AO	Supports ± 10 V, 0–10 V, 0–20 mA, 4–20 mA (13-bit + sign resolution).

Table 2.3: Specifications and Explanation of SIMATIC S7-1512C-1 PN PLC

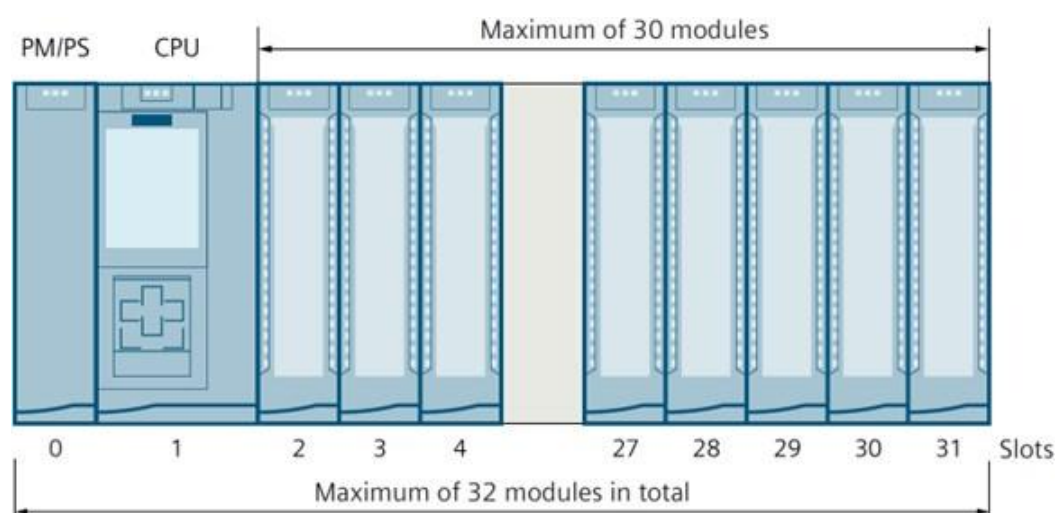


Figure 2.4: Mounting rail and module arrangement for SIMATIC S7-1500

Installing the CPU

The CPU is typically installed immediately to the right of the PS (if present) or acts as the leftmost module (Slot 1).

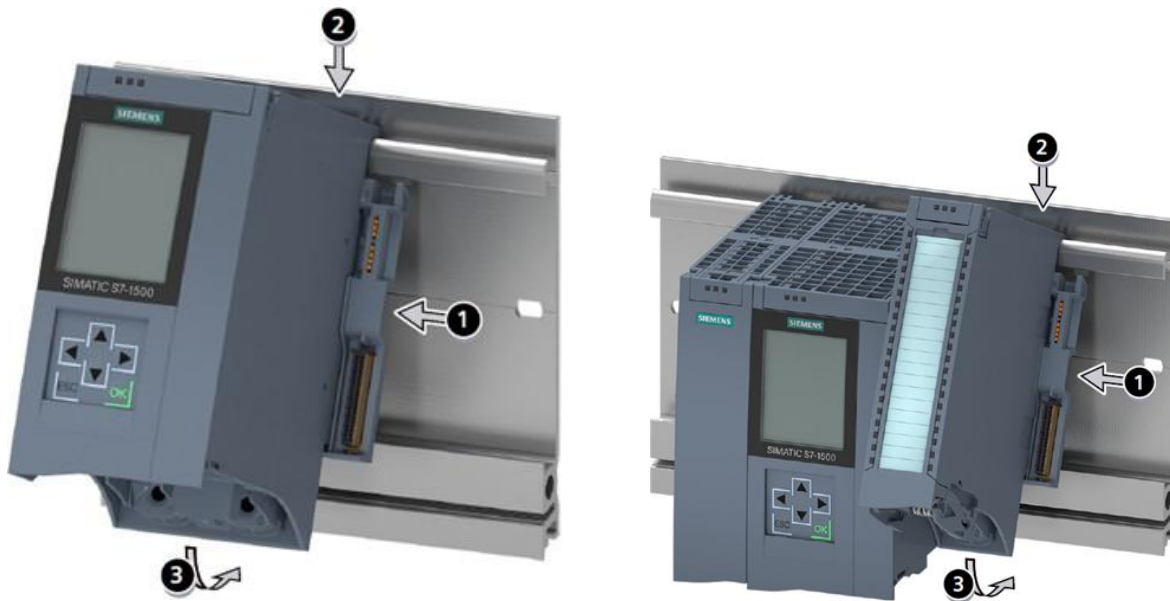


Figure 2.5: Installation of S7-1500 CPU (left) and I/O modules (right) on mounting rail

1. **Insert U-Connector:** Place a U-connector into the back right of the CPU.
2. **Mount and Connect:** Install the CPU onto the mounting rail. Push the CPU tightly against the left module to ensure mechanical and electrical connection.
3. **Swivel to Rear:** Rotate the CPU backward until it sits flat against the rail.
4. **Secure:** Tighten the mounting screws (recommended torque: 1.5 Nm).

Uninstalling the CPU

To uninstall a CPU:

1. Open the front cover and switch CPU to STOP mode.
2. Turn off the feed supply voltage.
3. Disconnect supply voltage connector.
4. Remove PROFIBUS/PROFINET bus connectors.
5. Unscrew the CPU fixing screws.
6. Pivot the CPU out of the mounting rail carefully.

Installing I/O Modules (SM/TM/CM)

I/O modules (Digital, Analog, Technology, or Communication Modules) are installed to the right of the CPU (Slots 2 onward).

1. **Insert U-Connector:** Insert a U-connector in the back right for each module except the last one.

Module-2: Perform Basic PLC Programming

2. **Mount and Connect:** Slide the I/O module onto the rail and push against the previous module.
3. **Swivel to Rear:** Pivot the module into final position.
4. **Secure:** Tighten the screw (torque: 1.5 Nm).

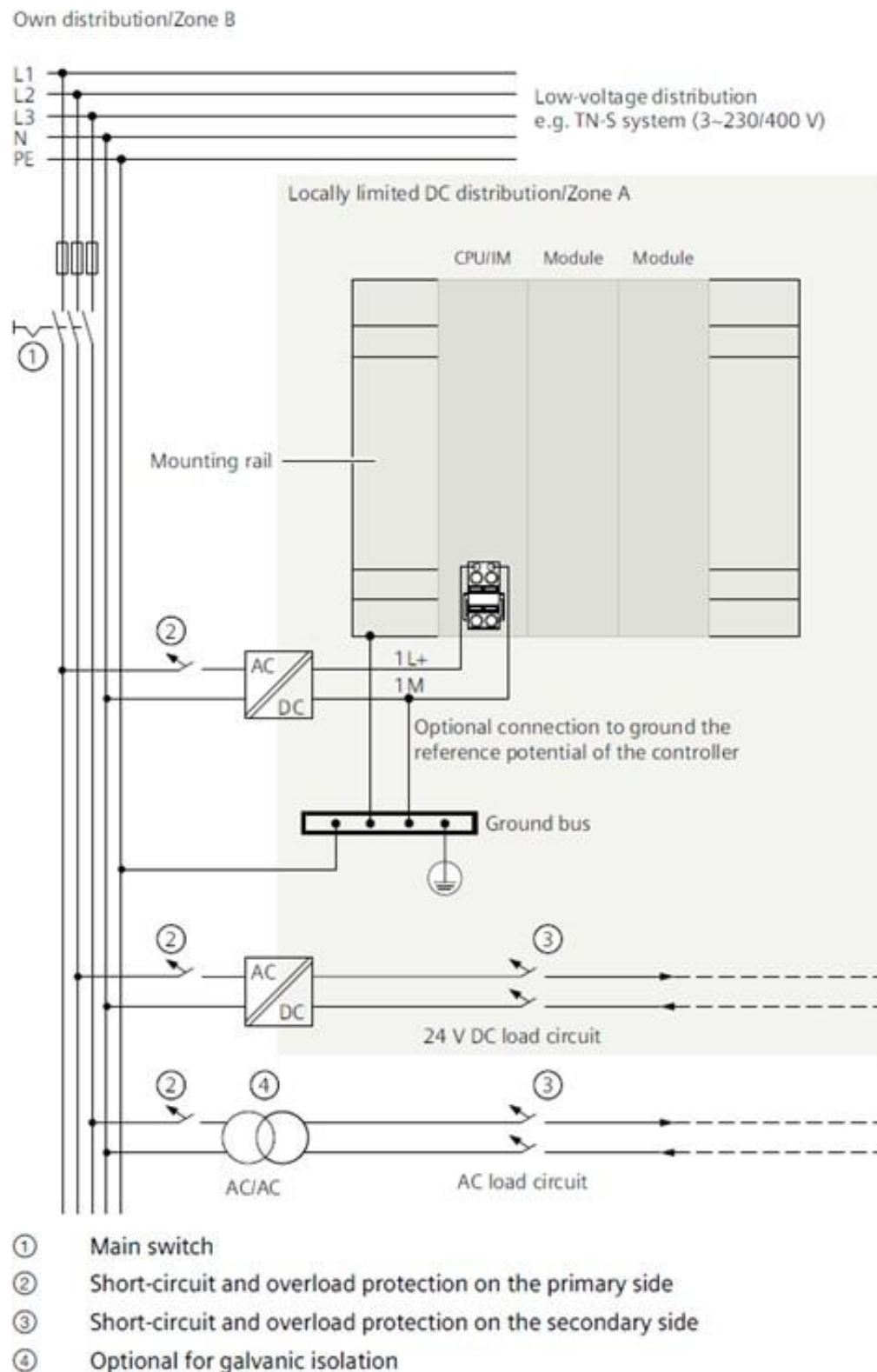


Figure 2.6: Typical wiring diagram for PLC

II. Associated Wiring

Reference Video: https://www.youtube.com/watch?v=UDOU_YmGLnE

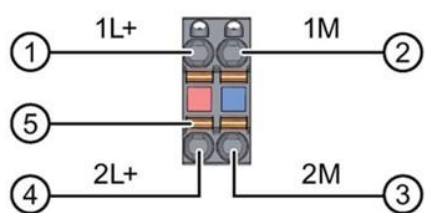
Mains Connection Plug Wiring: Remove the connector cover. Connect the mains cable according to the wiring diagram and voltage rating. Close and tighten the connector screw.

Load Current Supply (PM) to CPU Wiring: Wire the 4-pin connector of the load current supply to the CPU mains plug. Ensure polarity and secure connection.

Digital Input Module Wiring: Insert the front connector in the pre-wiring position (no electrical contact yet). Thread in a cable tie.

Digital Output Module Wiring: Feed supply voltage from digital input terminals (40: M, 39: L+) to output terminals (20: M, 19: L+). Connect potential bridges between 30-40 and 29-39.

Finalizing Front Connectors: Connect wires into front connectors per wiring diagram. Apply cable tie for strain relief, and move connectors to final position to complete electrical connection.



Label Descriptions:

1. +24 V DC of supply voltage (max 10 A)
2. Ground of supply voltage (max 10 A)
3. Ground for loop-through
4. +24 V DC for loop-through
5. Spring opener for each terminal

Figure 2.7: Power connections wiring

III. Power On and Initial Configuration

1. **Connect Power:** Insert the mains connection plug and energize the Power Module.
2. **Insert Memory Card:** Insert SIMATIC memory card into CPU slot.

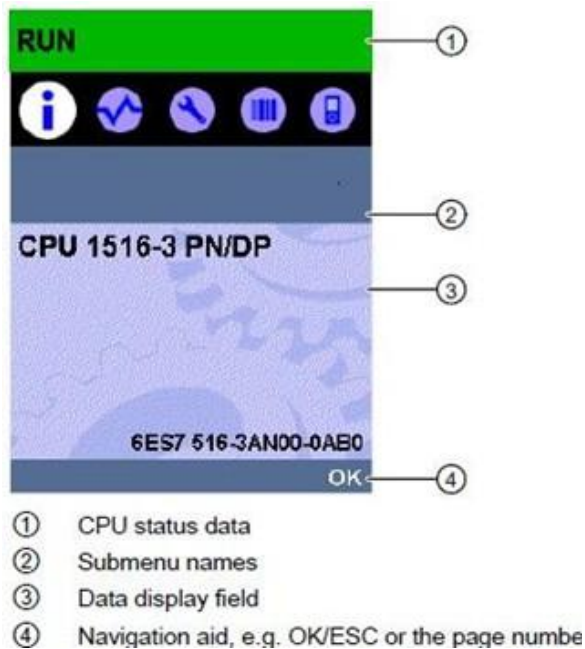


Figure 2.8: Simatic Memory Card

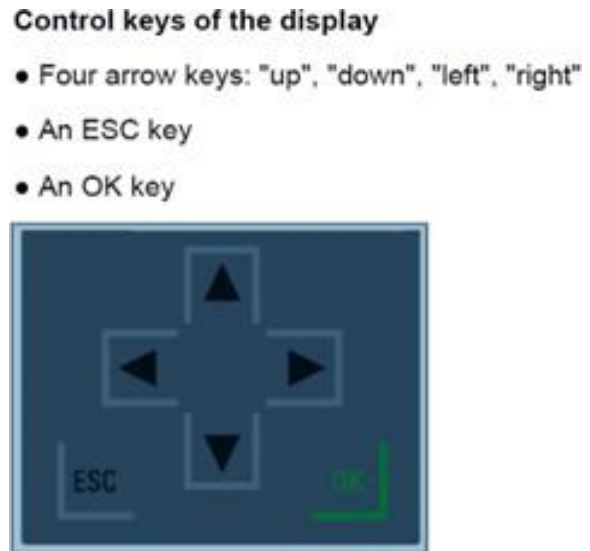
3. **Startup:** Switch the load current supply (PM) to RUN position — CPU enters STOP mode.

Module-2: Perform Basic PLC Programming

4. **Assign IP Address:** Navigate on CPU display: Settings → Addresses → X1 (IE/PN) → IP Addresses. Set desired IP (e.g., 192.168.0.10) and subnet mask (255.255.255.0), then Apply → OK.



(a) S7-1500 display



(b) Display Control Key

Figure 2.9: CPU display and control key for SIMATIC S7-1500 configuration

2.1.7 Self-Check Questions

No.	Question
1	Define a PLC and explain its basic scan cycle.
2	List the core hardware components of a PLC system.
3	State two advantages of PLCs over relay logic.
4	Give two examples each of PLC input and output devices.
5	Name three IEC 61131-3 programming languages.
6	Why must TIA Portal version and device firmware be compatible?
7	What identification parameters should be recorded for PLC and HMI?
8	Mention three criteria for selecting a PLC for a new application.

2.1.8 Answer Key

No.	Answer / Guidance
1	A solid-state controller that cyclically reads inputs, executes the user program, and updates outputs.
2	CPU, power supply, I/O modules (DI/DO/AI/AO), comms interfaces, backplane/rail.
3	Higher reliability; easier troubleshooting; scalable via software.
4	Inputs: push button, sensor, analog transmitter; Outputs: lamp, relay/contact, analog valve.
5	LAD, FBD, SCL/ST.
6	To ensure projects can compile, download, and operate correctly with the device.
7	Part numbers, firmware versions, and TIA Portal project version.
8	I/O count/types, environmental conditions, scan-time needs, scalability.

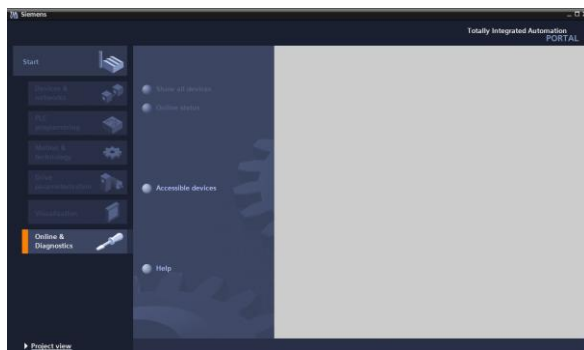
2.2 Learning Outcome: Setup PLC Hardware Configurations with Software Tool

2.2.1 Information Sheet: PLC Programming Software

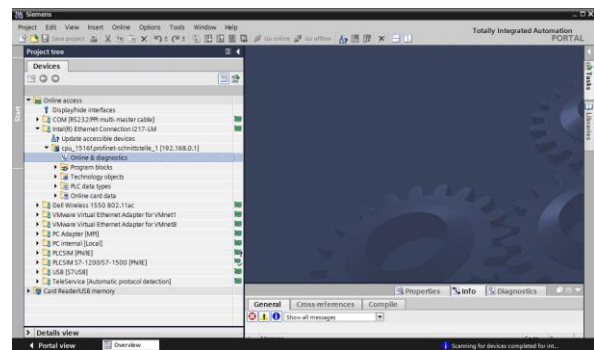
The primary programming tool used for SIMATIC automation systems, including the SIMATIC S7-1500, S7-1200, S7-300, S7-400, and SIMATIC WinAC, is SIMATIC STEP 7 Professional software, which is integrated within the Totally Integrated Automation Portal (TIA Portal). Required software versions include TIA Portal V13 to V19. Note that compatibility is limited; older versions cannot open projects created in newer versions, though older projects can generally be upgraded.

Views: The TIA Portal operates primarily using two views: Portal View: The default starting view, which offers a task-oriented approach for beginning users.

Project View: Used for complex tasks like hardware configuration, programming, and visualization.



(a) TIA Portal - Portal View

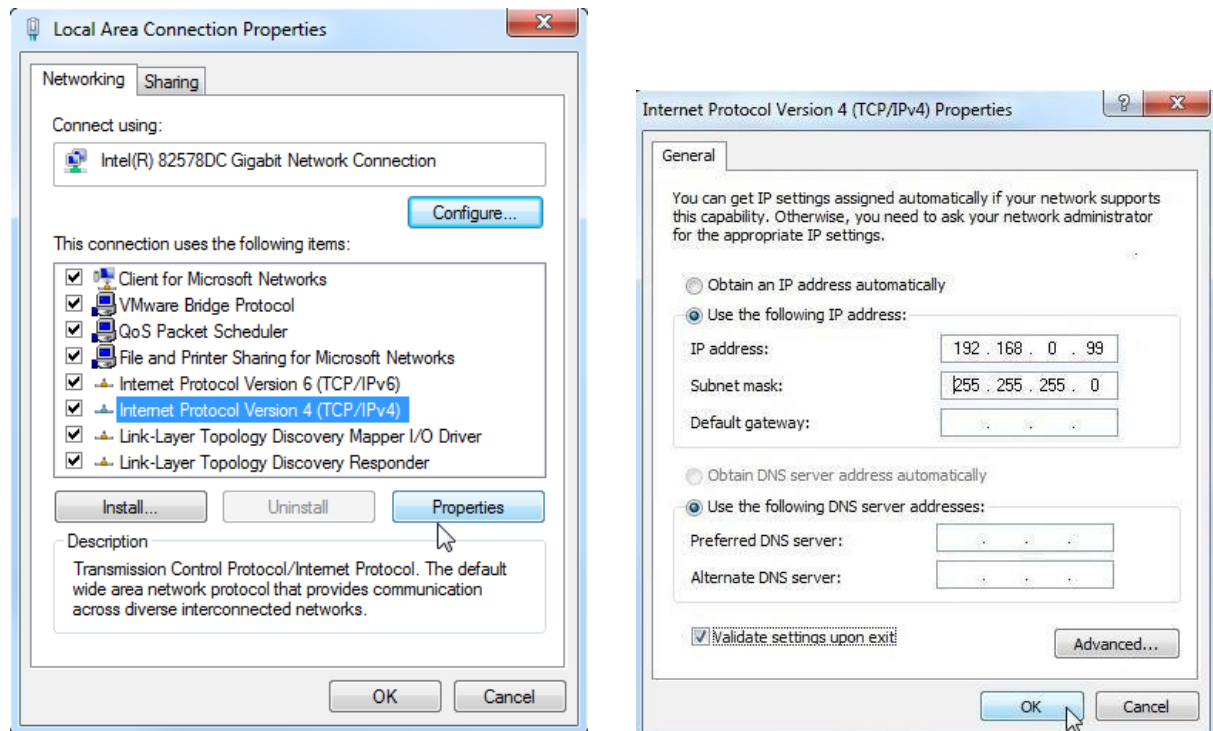


(b) TIA Portal - Project View

Figure 2.10: Interface Views in SIMATIC TIA Portal

2.2 Learning Outcome: Setup PLC Hardware Configurations with Software Tool

2.2.2 Information Sheet: Basic Networking for Working with PLC



(a) Key PLC networking components

(b) Addressing and subnet overview

Figure 2.11: Basic PLC networking and communication setup

Effective communication between the Programming Device (PC/Engineering Station) and the PLC (Controller) relies on fundamental networking concepts and protocols.

Key Networking Requirements and Concepts:

- **Physical Connection:** An Ethernet connection is required between the engineering station and the controller. The connection typically uses TCP/IP.
- **IP Address and Subnet:** For communication via TCP/IP, the IP addresses of both the PC and the PLC must be configured to match in the subnet portion (as defined by the subnet mask) but must have different node addresses.
- **IP Address:** A unique identifier assigned to a device (e.g., 192.168.0.1). PLCs use static IP addresses for stable communication.
- **Subnet Mask:** Defines the network range (e.g., 255.255.255.0) to ensure devices are in the same subnet and can communicate.
- **MAC Address:** A unique hardware identifier assigned to every network device (PLC, HMI, PC).
- **Gateway:** The IP address of a router needed if the PLC must communicate with devices located outside its local network.

Industrial Protocols:

- **PROFINET:** Industrial Ethernet protocol for real-time communication between PLCs, HMIs, and I/O devices.

Module-2: Perform Basic PLC Programming

- **Modbus TCP:** Ethernet-based protocol typically using a client-server model.
- **OPC UA:** Vendor-independent protocol for data exchange with SCADA and supervisory applications.

2.2.3 Information Sheet: Addressing Configuration via TIA Portal

The IP address of an S7-1500 can be configured through:

- The **CPU display panel**, or
- The **TIA Portal "Online & Diagnostics"** menu.

Steps for TIA Portal:

1. Open Online & Diagnostics in TIA Portal.
2. Select Functions-Assign IP Address
3. Select Accessible devices.
4. Select the CPU interface (e.g., PROFINET [X1]). and click Start Search.
5. A list of accessible devices will appear. Select the desired CPU from the list. Then click Apply.
6. Enter the desired IP address of PLC (e.g., 192.168.0.10) and subnet mask (e.g., 255.255.255.0).
7. Click Assign IP address to save settings.

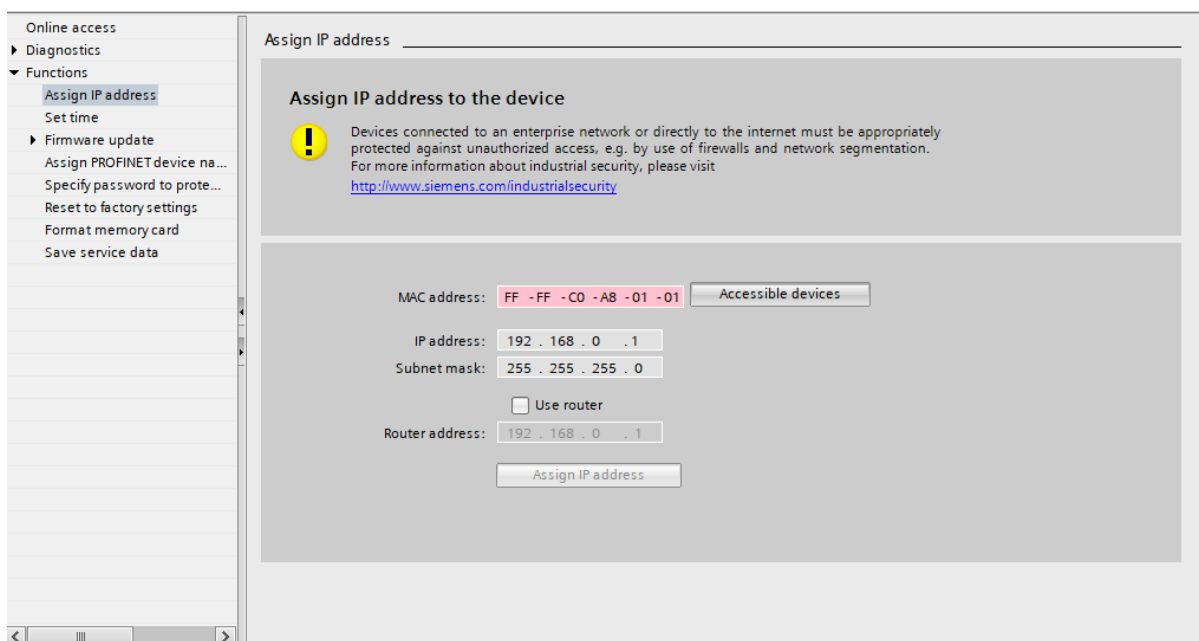


Figure 2.12: Assigning IP address to unknown PLC via TIA Portal

Steps for IP setting from PLC Display:

2.2 Learning Outcome: Setup PLC Hardware Configurations with Software Tool

1. Set the CPU to STOP mode first.
2. Press the right arrow button to navigate to the "Setting" menu, then press the "OK" button to enter the menu.
3. Navigate to the "Address" menu and press "OK" to enter
4. Select the "X1" menu and press "OK" to enter.
5. In the "X1" menu, locate the "IP address" option in the list, then press "OK" to select the submenu.
6. Press "OK" again to enter the edit mode of the "IP Address" menu. Use the directional arrows to set the desired value, for example, "192.168.0.10", as shown in the image.
7. The same method can be used to set the subnet mask. Once all settings are complete, click on "Apply" to apply the changes and make them effective.



(a) CPU Display home screen



(b) Navigate address settings in STOP mode



(c) Setting up IP address

Figure 2.13: Setting Up S7-1500 PLC IP Address via CPU Display

2.2.4 Information Sheet: Hardware Configuration and Download

I. Inserting and Configuring the CPU

- **Configure a Device:** In the "Start" portal, navigate to "First steps" > "Devices & Networks" > "Configure a device".
- **Add New Device:** Switch to the "Add new device" menu.
- **Select CPU:** Choose the controller model (e.g., SIMATIC S7-1500 → CPU 1512C-1 PN).
- **Name and Add:** Assign a device name (e.g., "CPU_1512C"), select "Open device view," and click "Add".

II. Configuring Network Communication

- Access Properties: Double-click the CPU in Device Configuration.
- Ethernet Addresses: Open the “PROFINET-interface [X1]” menu and select “Ethernet addresses”.
- Add Subnet and Set IP: Adjust IP address and subnet mask as needed.

(Note for Safety CPUs:) If configuring a fail-safe CPU like the 1516F-3 PN/DP, disable “F-activation” under the Fail-safe menu.

III. Inserting Hardware Modules

- Slot 0: Power Module (e.g., PM 190W 120/230VAC)
- Slot 1: CPU 1512C-1 PN
- Slot 2: Digital Input (DI 32x24VDC HF)
- Slot 3: Digital Output (DQ 32xDC24V/0.5A HF)

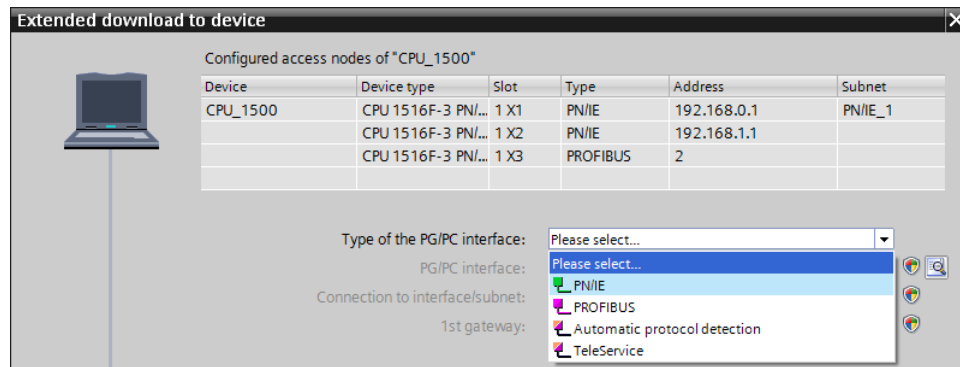
IV. Configuring and Checking I/O Address Areas

- Verify DI Address: Check address (I0.0-I3.7)
- Verify DQ Address: Check address (Q0.0-Q3.7)

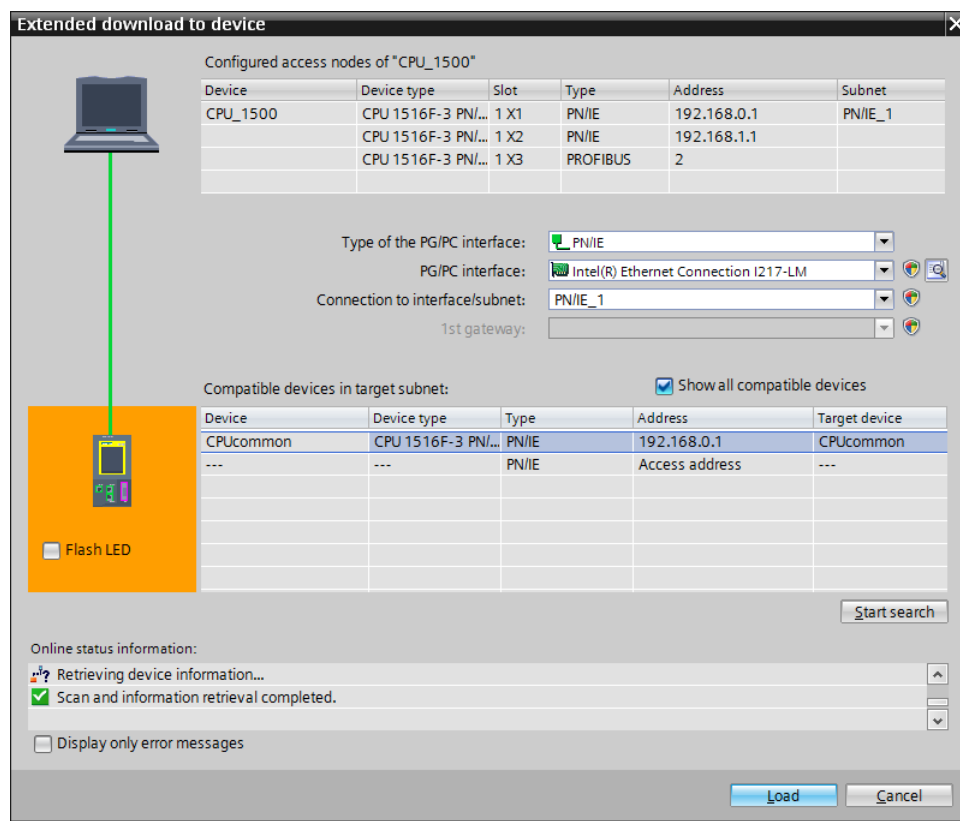
V. Compiling and Downloading the Configuration

1. Save the project frequently.
2. Compile: Hardware and software (only changes).
3. Download to Device → Select correct subnet.
4. Confirm “Show all compatible devices” → Search → Load → “Start all”.

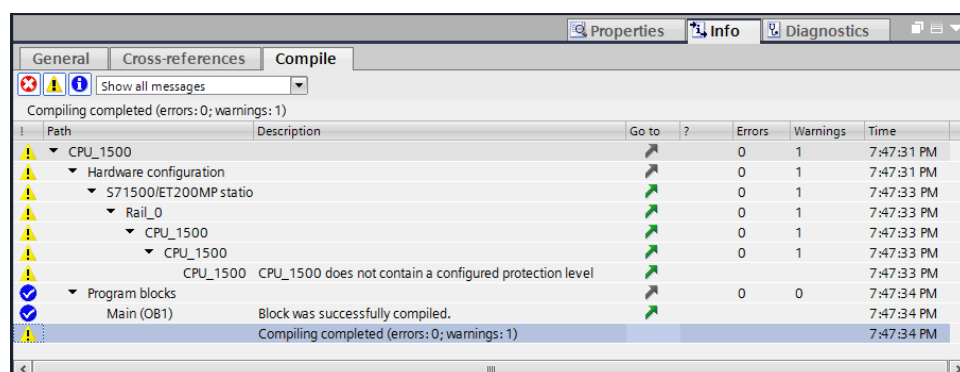
2.2 Learning Outcome: Setup PLC Hardware Configurations with Software Tool



(a) Select PG/PC interface



(b) Search compatible devices



(c) Finalize hardware download

Figure 2.14: TIA Portal hardware download process

2.2.5 Job Sheet - Setup PLC Hardware Configuration Using Siemens TIA Portal

Section	Content
Job Title	Setup PLC Hardware Configuration Using Siemens TIA Portal
Personal Protective Equipment (PPE)	Safety gloves, anti-static wrist strap, ESD-safe workstation, and safety shoes must be used during PLC installation and configuration.
Materials Required	Siemens PLC (S7-1200 or S7-1500), Siemens TIA Portal software (v15 or above), PLC power supply (24 V DC), Ethernet communication cable (CAT6 or industrial type), input/output simulation board or push buttons and indicators, computer or laptop with TIA Portal installed, 24 V DC power source, and USB or Ethernet programming cable.
Tools and Equipment	Laptop or PC with administrator access, Ethernet/USB programming cable, multimeter for voltage verification, screwdriver set, cable markers, and wiring accessories.
Procedure	1. Preparation and Safety - Ensure the PLC is powered OFF before connecting or removing any module. - Connect the PLC to a 24 V DC power supply and verify voltage with a multimeter. - Establish a proper Ethernet or USB connection between the PLC and the PC. - Open Siemens TIA Portal and create a new project. 2. Adding Hardware in TIA Portal - From the “Project View,” select <i>Devices & Networks</i> and add a new device. - Choose the correct PLC model (e.g., CPU 1214C, CPU 1511-1 PN) and firmware version. - Add associated modules such as DI, DO, AI, AO, or communication modules as per the physical setup. - Arrange the modules in the correct rack and slot order in the configuration window.

Continued on next page

2.2 Learning Outcome: Setup PLC Hardware Configurations with Software Tool

Section	Content
	3. Configuring Hardware Parameters - Open CPU properties and set the device name, PROFINET interface, and IP address (e.g., 192.168.0.1). - Assign module addresses automatically or manually for digital/analog channels. - Save and compile the hardware configuration to verify there are no errors.
	4. Establishing Communication - Connect the PC and PLC using an Ethernet or USB cable. - In TIA Portal, select <i>Online > Accessible Devices</i> to detect the PLC. - Download the hardware configuration to the CPU and switch to “Online” mode. - Verify communication by checking CPU and module status indicators.
	5. Verification and Testing - Check that each module appears under the correct address in the device tree. - Test I/O channels using simulation switches or digital inputs/outputs. - Observe diagnostic messages and confirm communication status as “Successful.” - Save and back up the project once testing is complete.
Notes	- Always ensure unique IP addresses on the same network. - Use proper grounding and ESD protection during PLC handling. - Save the project frequently to prevent data loss. - Clearly label all I/O channels and modules for easy identification. - Return the PLC to STOP mode after testing is complete.
Key Technical Points	- The hardware configuration defines the physical and logical structure of the PLC system. - Each module is addressed by its rack, slot, and channel number. - PROFINET communication enables data exchange between PLCs, HMIs, and drives. - Proper configuration ensures reliable operation and accurate diagnostics. - Mastering hardware setup is essential for programming, commissioning, and troubleshooting PLC systems.

2.2.6 Self-Check Questions

No.	Question
1	Which physical connection is typically used between PC and PLC, and which protocol?
2	Explain IP address and subnet mask in the PLC context.
3	Name two industrial protocols commonly used with PLCs.
4	Where do you configure the CPU's Ethernet settings in TIA Portal?
5	Outline the steps to add a CPU and assign an IP address.
6	How do you insert DI/DQ modules into the device configuration?

2.2 Learning Outcome: Setup PLC Hardware Configurations with Software Tool

2.2.7 Answer Key

No.	Answer / Guidance
1	Ethernet cable; TCP/IP (PROFINET over Ethernet for IO).
2	IP uniquely identifies a device; subnet mask defines the network range that can communicate directly.
3	PROFINET, Modbus TCP, OPC UA.
4	In the CPU Properties under the PROFINET interface (Ethernet addresses).
5	Create project → Configure device → Add CPU → Open device view → Set Ethernet address and subnet.
6	Use the Hardware catalog in Device Configuration; drag-and-drop modules or search by order number.

2.3 Learning Outcome: Create PLC Program

2.3.1 Information Sheet: Creating a New Program in TIA Portal

Creating a new program in the TIA Portal involves setting up the project structure, configuring the hardware, and creating the functional code blocks.

Program Block Creation

1. **Access Program Blocks:** In the Project tree, expand the "Program blocks" folder.
2. **Add New Block:** Double-click "Add new block".

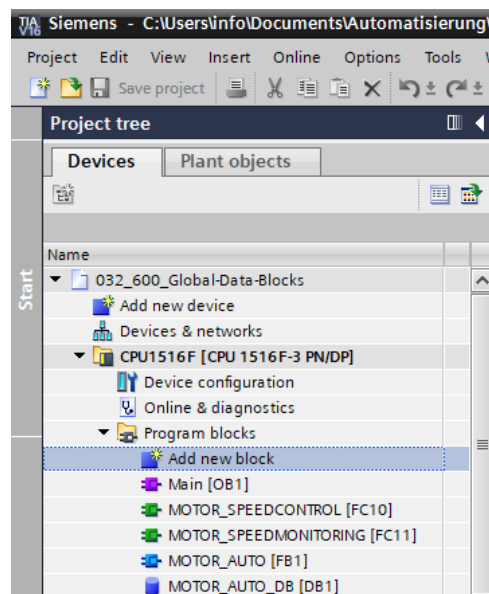


Figure 2.15: Add new program block in TIA Portal

3. **Define Block Type and Language:** In the dialog, select the type of block (e.g., Function (FC), Function Block (FB), or Organization Block (OB)). Then, select the desired programming language (LAD, FBD, or SCL). Note that a Data Block (DB) does not require a language selection as it only stores data.
4. **Open Block:** Clicking "OK" creates the block. If "Add new and open" (default) is selected, STEP 7 opens the newly created block in the editor immediately. The main program logic is typically executed in an Organization Block (OB), such as Main [OB1] (cycle OB).

2.3.2 Information Sheet: Addressing and Data Types

Addressing

In PLC programming, operands (inputs, outputs, and memory locations) can be referenced in two primary ways.

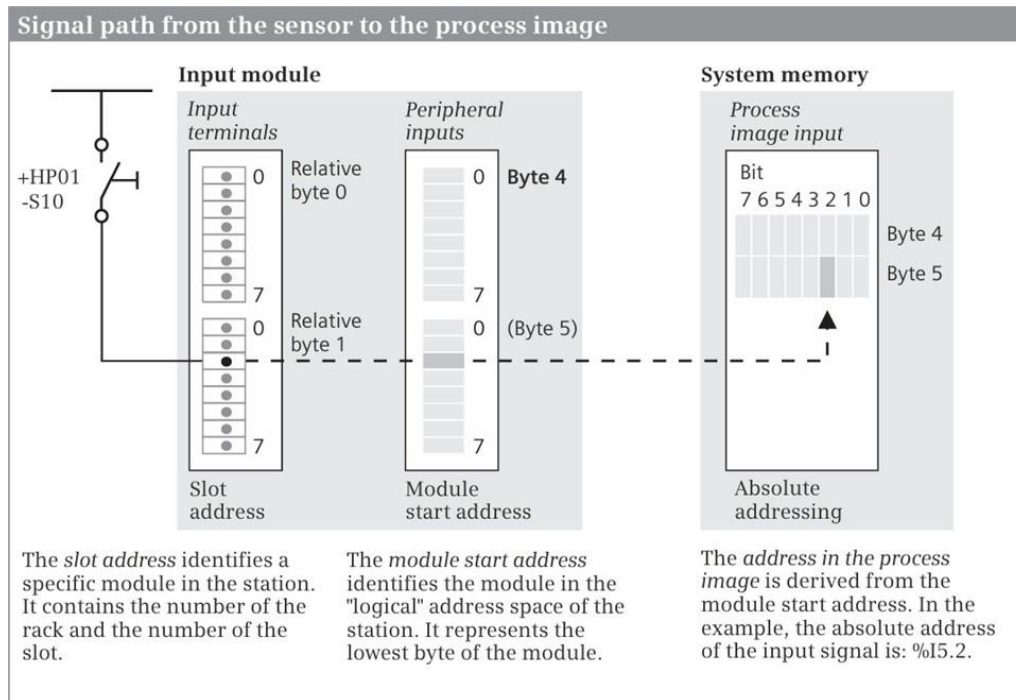


Figure 2.16: Addressing methods in PLC

1. **Absolute Addressing:** This method uses specific memory locations defined in the hardware configuration. The absolute address consists of:
 - **Memory Area Identifier:** I (Input), Q (Output), M (Bit Memory)
 - **Size Identifier:** B (Byte), W (Word), D (Double Word)
 - **Address Location:** Defined by byte and optional bit (e.g., Byte 3, Bit 4)
2. The process image is accessed by default (e.g., %I5.2). To access the physical input/output directly, append ":P" (e.g., I0.3:P or "Stop:P").
3. **Symbolic Addressing (Tags):** This uses descriptive names (tags) instead of absolute addresses. The combination of operand, name, and data type forms a "tag". STEP 7 supports symbolic programming.

Data Types

Data types define the properties of tags and allowable value ranges.

Note: Tagged data types can be accessed by slices, e.g. <tag>.xn (bit), .bn (byte), or .wn (word).

2.3.3 Information Sheet: Create and Assign Tags / Create Tag Table

A PLC tag table defines global tags that are known throughout the user program.

Creating and Assigning Tags in the Tag Table

Data Type Category	Examples	Size (Bits)	Addressing Example
Bit/Boolean	BOOL	1	I1.0, Q0.0
Integer (Short)	SINT, USINT, CHAR, BYTE	8	IB1, QB4
Integer (Standard)	INT, UINT, WORD, S5TIME, DATE	16	IW1, QW10
Integer (Double)	DINT, UDINT, DWORD, REAL, TIME	32	ID0, QD1
Complex/User Defined	ARRAY, STRUCT, PLC Data Type	Variable	Symbolic Access

Table 2.9: Common Data Types in Siemens PLCs

	Name	Data type	Address	Retain	Comment
12	Manual switch-on	Bool	%I0.1	☐	Switch on in manual mode
13	Manual switch-off	Bool	%I0.2	☐	Switch off in manual mode
14	/Stop	Bool	%I0.3	☐	Stop signal from pushbutton
15	Start	Bool	%I0.4	☐	Start pushbutton on the control panel
16	Continue	Bool	%I0.5	☐	Continue button on the control panel
17	Acknowledge	Bool	%I0.6	☐	Acknowledge from control panel
18	Set	Bool	%I0.7	☐	Set signal from control panel
19	Display error	Bool	%Q8.0	☐	Fault indication on control panel
20	Display ready	Bool	%Q8.1	☐	"Finished" display on control panel
21	Belt motor 1	Bool	%Q8.2	☐	Switch on conveyor belt 1
22	Belt motor 3	Bool	%Q8.4	☐	Switch on conveyor belt 3
23	Belt motor 4	Bool	%Q8.5	☐	Switch on conveyor belt 4
24	Ready for load	Bool	%M42.0	☐	Conveyor belt is ready for goods
25	Ready for remove	Bool	%M42.1	☐	Piece goods can be picked up
26	Quantity parts	Int	%MW44	☐	Number of piece goods
27	Monitoring	Timer	%T12	☐	Monitoring time
28	Manual mode	Bool	%M40.0	☐	Manual mode (operating mode)
29	Automatic mode	Bool	%M40.1	☐	Automatic mode
30	Quantity counter	Counter	%C11	☐	Counter for piece goods
31	Automatic switch-on	Bool	%I1.0	☐	Switch on in automatic mode

Figure 2.17: Tag table overview

1. **Open Tag Table:** Navigate to "PLC tags" and open the "Default tag table" or create a new one.

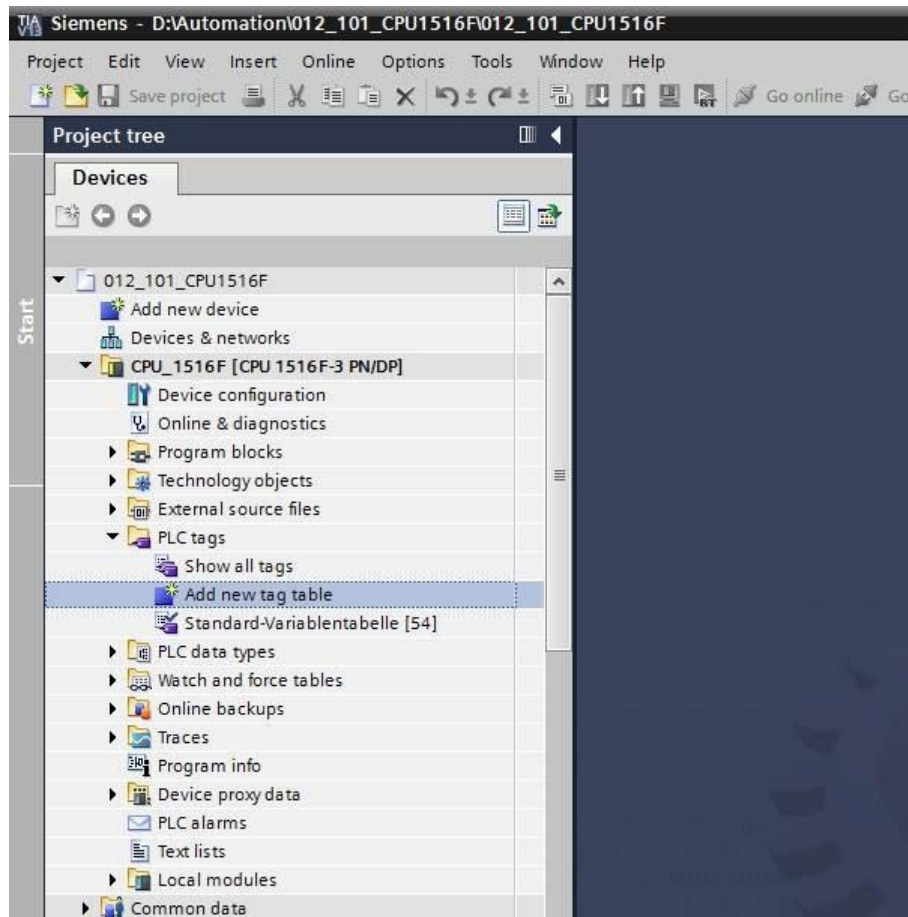


Figure 2.18: Opening tag table in TIA Portal

2. Enter Tag Details (Manual Method):

- Name: Unique symbolic name
- Data Type: Bool, Int, etc.
- Address: e.g., %I0.0 or %Q4.1

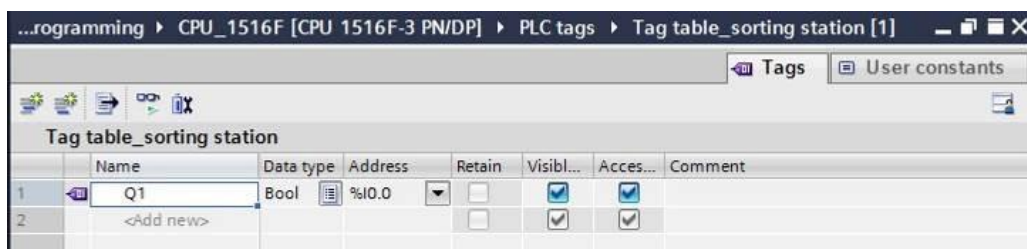


Figure 2.19: Entering tag details

3. **Assign Tags (Drag-and-Drop Method):** Drag I/O points from the CPU graphic to the tag table. Rename default tags (e.g., “I0.0”) to meaningful names (e.g., “Start”).

Once tags are defined, they are available throughout the program.

2.3.4 Information Sheet: Create Data Block (DB)

A Data Block (DB) is a non-volatile memory area used to store and organize data (variables, tags, values).

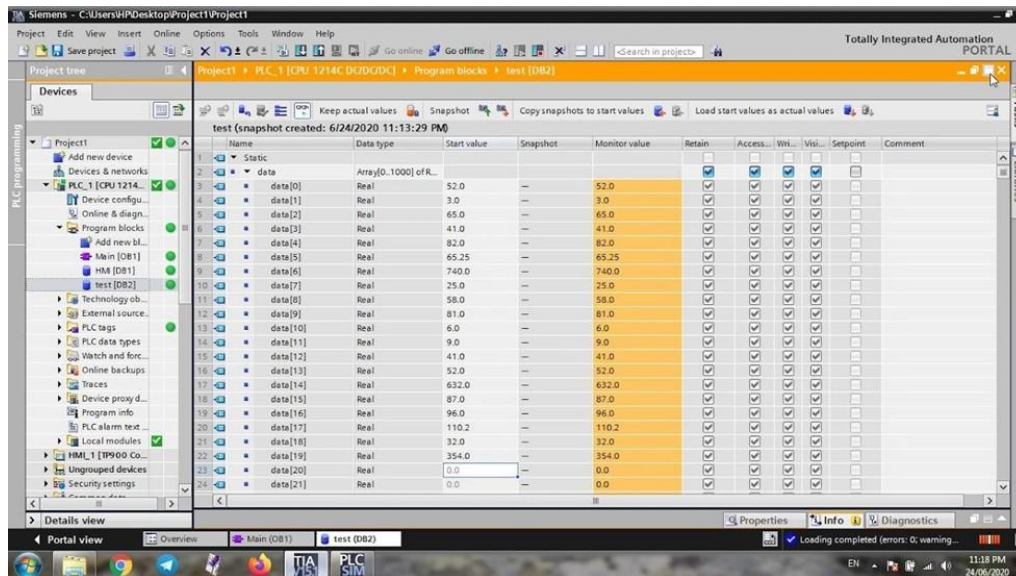


Figure 2.20: Creating a new Data Block

1. **Access Block Creation:** In Project View, open "Program Blocks" and click "Add New Block".
2. **Select Data Block:** Click "Data Block".
3. **Choose DB Type:**
 - Global DB: Shared data for multiple blocks.
 - Instance DB: Data for one Function Block instance.

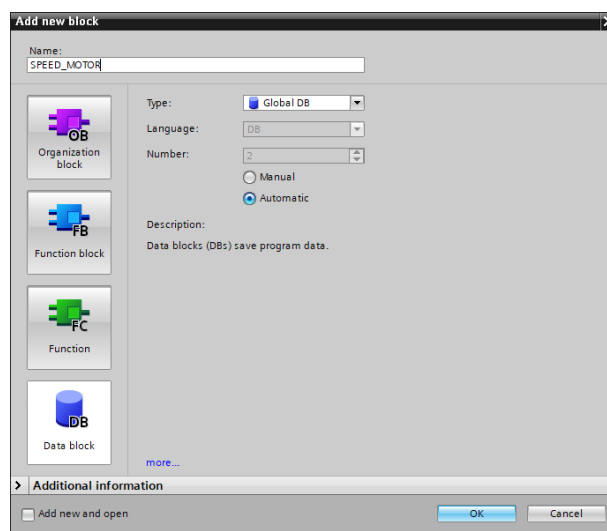


Figure 2.21: Selecting Data Block type

4. **Naming and Properties:** Assign a unique name. Block numbers can be manually adjusted.

5. **Define Variables:** Check "Add new and open" to declare tags (variables).
6. **Set Retentivity:** Define variables as retentive to preserve data during restarts.

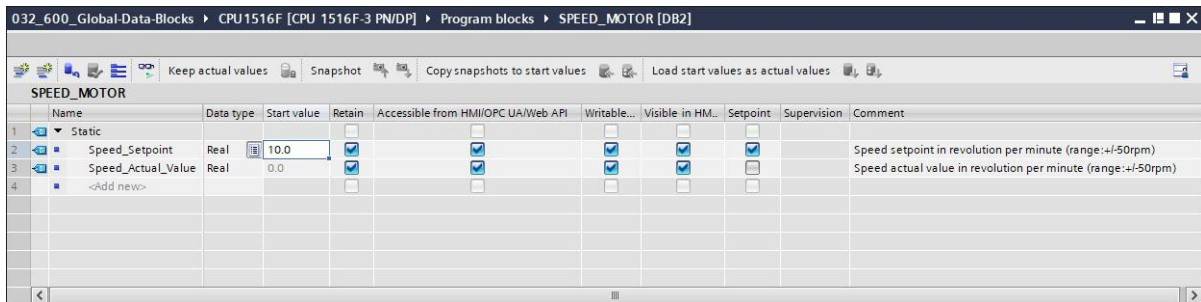


Figure 2.22: Configuring variables in Data Block

When to Use Data Block vs. Tag Table

Feature	Data Block (DB)	PLC Tag Table
Primary Use	Control data, machine states, FB data	I/O mapping and HMI variables
Data Complexity	Complex structures (Array, Struct, UDT)	Simple variables (Bool, Int)
Data Retention	Can be retentive	Non-persistent
Scope of Data	Global or local to FB instance	Global (mapped to hardware)

Table 2.10: Comparison between Data Block and Tag Table

2.3.5 Information Sheet: What is Ladder Diagram (LAD)?

Ladder Logic (LAD) is a graphical programming language used for PLCs.

- **Representation:** Based on circuit diagrams, structured as rungs of a ladder.
- **Elements:** Normally open/closed contacts and coils.
- **Logic Implementation:** Series = AND; Parallel = OR.
- **Structure Rules:** Every network ends with an output coil or box instruction.
- **Execution:** CPU executes sequentially; power flow = signal ON.



Figure 2.23: Example of Ladder Logic network

I. Basic Logic Elements (Contacts and Coils)

Logic Gates Implemented in LAD

- AND Logic - contacts in series.
- OR Logic - contacts in parallel.
- NOT Logic - inverted contact.

Function / Element	Description
Normally Open Contact	Closed (ON) when signal = 1; series = AND logic.
Normally Closed Contact	Closed (ON) when signal = 0.
Output Coil	Sets bit to 1 if power flow exists; otherwise 0.
Inverted Output Coil	Inverts coil logic.

Table 2.11: Basic Ladder Logic Elements

- NAND, NOR, XOR, XNOR - combinations of above.



Figure 2.24: AND and OR logic circuits

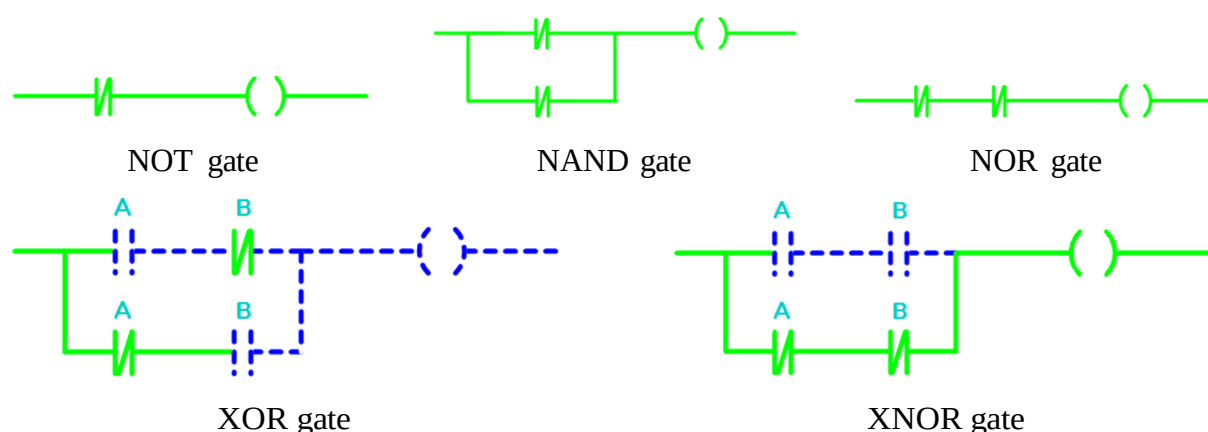


Figure 2.25: NOT, NAND, NOR, XOR, and XNOR Ladder logic examples

2.3.6 Job Sheet - Verify NO/NC Behaviour in Ladder Diagram

Section	Content
Job Title	Verify NO/NC Behaviour in Ladder Diagram
Personal Protective Equipment (PPE)	Safety gloves, anti-static wrist strap, ESD-safe workstation, and safety shoes must be used while handling PLC hardware and wiring.
<i>Continued on next page</i>	

Section	Content
Materials Required	Siemens S7-1200 or S7-1500 PLC, TIA Portal software (v15 or above), 24 V DC power supply, two push button switches (NO type - Inputs I0.0 and I0.1), one indicator lamp (Output Q0.0), Ethernet programming cable (CAT6), and input/output simulation board (optional).
Tools and Equipment	Laptop or PC with TIA Portal software, Ethernet/USB programming cable, multimeter for voltage verification, screwdriver set, cable markers, and wiring accessories.
Procedure	1. Preparation and Safety - Ensure the PLC and associated components are properly powered and grounded. - Connect the PLC to the PC using an Ethernet or USB programming cable. - Open Siemens TIA Portal and create a new project. - Configure the hardware and assign input/output addresses as follows: I0.0 → NO push button; I0.1 → NO push button; Q0.0 → indicator lamp. 2. Programming in TIA Portal - Open the main program block (OB1). - Create a new network and insert two input contacts and one output coil. - Configure the contacts to implement XOR logic using NO/NC instructions. - Label all variables clearly (e.g., Start1, Start2, Lamp). 3. Wiring the Physical Circuit - Connect both push buttons to the PLC input terminals (I0.0 and I0.1). - Connect the indicator lamp to the PLC output terminal (Q0.0). - Use 24 V DC as the control voltage and verify all wiring with a multimeter.

Continued on next page

Section	Content
	4. Downloading and Testing the Program - Compile the program and download it to the PLC. - Switch the PLC to RUN mode. - Test the XOR logic by pressing push buttons in different combinations. - Observe the output lamp (Q0.0): it should turn ON only when one input is TRUE.
	5. Verification and Observation - Confirm that both inputs OFF → output OFF. - Either input ON → output ON. - Both inputs ON → output OFF. - Record results in the observation table.
Notes	- Always verify the input/output addressing before downloading the program. - Keep the PLC in STOP mode while changing wiring. - Use M-memory bits for intermediate virtual logic if needed. - XOR logic is commonly used in interlock, comparison, and safety verification circuits.
Key Technical Points	- NO (Normally Open) and NC (Normally Closed) contacts represent real or virtual signal states in ladder logic. - Physical inputs simulate real-world devices, while virtual contacts handle internal decisions. - XOR logic output remains TRUE only when one input is TRUE. - Understanding NO/NC logic helps link physical wiring and PLC program structure effectively.

2.3.7 Information Sheet: Counter and Timer Instructions

Timers and Counters are typically implemented in LAD as **box instructions** or **output coils**. They utilize a structure stored in a data block (DB) to maintain data.

A. Timer Operations

Instruction (IEC Type)	Description
TP (Pulse Timer)	Generates a pulse with a preset width time.
TON (ON-Delay Timer)	Sets the output (Q) to ON after a preset time delay.
TOF (OFF-Delay Timer)	Sets the output (Q) to ON and then resets it to OFF after a preset delay.
TONR (ON-Delay Retentive Timer)	Sets the output (Q) to ON after a delay; elapsed time accumulates until reset.
Preset Timer (PT) Coil	Loads a new preset time value into the specified timer.
Reset Timer (RT) Coil	Resets the specified timer.

Table 2.13: Common Timer Instructions in TIA Portal

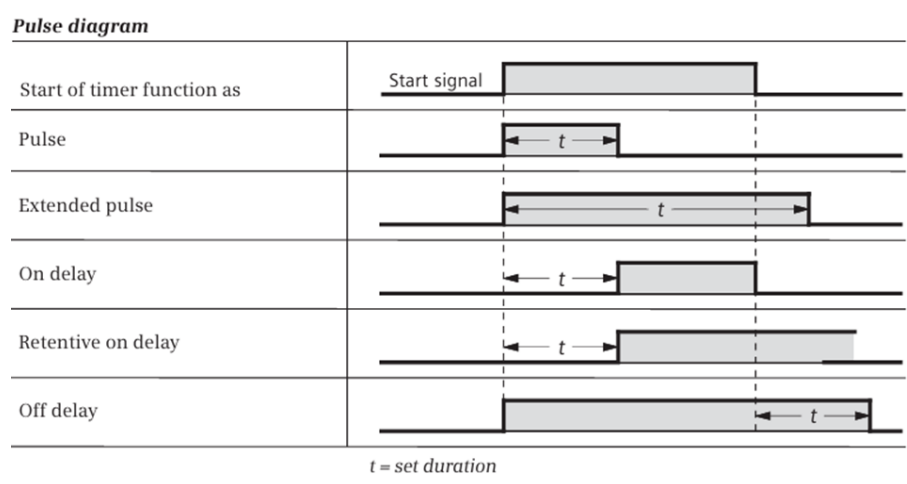


Figure 2.26: Timer operations in TIA Portal

B. Counter Operations

Instruction	Description
CTU (Count Up)	Counts up by 1 on the 0→1 transition of CU input.
CTD (Count Down)	Counts down by 1 on the 0→1 transition of CD input.
CTUD (Count Up and Down)	Counts up or down by 1 depending on input transition.
CTRL_HSC (High-Speed Counter)	Controls hardware-based counters for events faster than scan rate.

Table 2.14: Counter Instructions in Siemens LAD

2.3.8 Information Sheet: Math Functions and Data Handling Instructions in TIA Portal

Function / Instruction	Description
MOVE	Copies a data element from IN to OUT1.
MOVE_BLK / UMOVE_BLK	Block move instructions; MOVE_BLK is interruptible, UMOVE_BLK is not.
Comparator (==, <, >, etc.)	Compares two tags; activates output if TRUE.
Addition (+), Subtraction (-), Multiplication (*), Division (/)	Standard arithmetic operations.
CALCULATE	Creates a complex mathematical equation with multiple inputs.
CONV (Data Type Conversion)	Converts data from one type to another (e.g., DWORD_TO_REAL).
NORM_X (Normalize)	Normalizes a value between 0.0 and 1.0 based on MIN/MAX.
SCALE_X (Scale)	Scales a normalized value to a specific range.

Table 2.15: Common Data Handling and Math Instructions in TIA Portal

2.3.9 Job Sheet - Observe and Compare the Function of TP, TON, and TOF Timers in PLC

Section	Content
Job Title	Observe and Compare the Function of TP, TON, and TOF Timers in PLC
Personal Protective Equipment (PPE)	Safety gloves, ESD wrist strap, safety shoes, and eye protection must be worn while handling PLC and wiring components.
Materials Required	Siemens S7-1200 or S7-1500 PLC, TIA Portal software (v15 or above), 24 V DC power supply, one NO push button (Start input), three indicator lamps or LEDs (for timer outputs), and Ethernet/USB programming cable.
Tools and Equipment	Laptop or PC with TIA Portal installed, multimeter for 24 V DC verification, screwdriver set, wire stripper, and crimping tool.
<i>Continued on next page</i>	

Section	Content
Procedure	1. Preparation and Safety - Ensure the PLC is properly powered and grounded before starting. - Connect the PLC to the PC using Ethernet or USB cable. - Open TIA Portal and create a new project. - Configure CPU (e.g., 1214C DC/DC/DC) and assign addresses as follows: I0.0 = Start Push Button; Q0.0 = TON Lamp; Q0.1 = TOF Lamp; Q0.2 = TP Lamp.
	2. Create Timer Networks - Open the Main Program Block (OB1) and create three separate networks. <i>Network 1 - TON (On-Delay Timer):</i> Output turns ON after input stays ON for the preset time. Insert a TON instruction, assign input I0.0, output Q0.0, and preset time T#5s. <i>Network 2 - TOF (Off-Delay Timer):</i> Output remains ON after input turns OFF for the preset time. Insert a TOF instruction, assign input I0.0, output Q0.1, and preset time T#5s. <i>Network 3 - TP (Pulse Timer):</i> Generates a fixed-duration pulse when input turns ON. Insert a TP instruction, assign input I0.0, output Q0.2, and preset time T#5s.
	3. Download and Testing - Compile the project and download it to the PLC. - Switch the PLC to RUN mode. - Press the push button and observe each lamp's response: <i>TON:</i> Lamp turns ON after delay when button held. <i>TOF:</i> Lamp stays ON briefly after button released. <i>TP:</i> Lamp turns ON for fixed duration regardless of input hold time.
	4. Observation and Verification - Use TIA Portal's Online Monitoring to observe timer progress and elapsed time. - Adjust preset times and retest to visualize behavior differences.
Continued on next page	

Module-2: Perform Basic PLC Programming

Section	Content
Notes	- Assign distinct outputs for each timer to easily observe behavior. - Do not trigger input repeatedly during the TP timer pulse period. - Monitor real-time operation via online diagnostics in TIA Portal. - Properly label I/O addresses and keep circuit wiring neat and safe.
Key Technical Points	- TON (On-Delay): Output energizes after input stays ON for the preset delay. - TOF (Off-Delay): Output remains energized for preset delay after input turns OFF. - TP (Pulse): Output produces a fixed-duration pulse on input's rising edge. - Timers are essential for sequencing, interlocks, and time-based control in PLC automation.

2.3.10 Job Sheet - Create a Blinking LED Using Timer in Ladder Logic

Section	Content
Job Title	Create a Blinking LED Using Timer in Ladder Logic
Personal Protective Equipment (PPE)	Safety gloves, ESD wrist strap, safety shoes, and safety goggles must be worn while working with PLC hardware and 24 V DC wiring.
Materials Required	Siemens S7-1200 or S7-1500 PLC, TIA Portal software (v15 or above), 24 V DC power supply, one NO push button (Start), one indicator lamp or LED (Output Q0.0), Ethernet or USB programming cable, and I/O simulation board (optional).
Tools and Equipment	Laptop or PC with TIA Portal installed, screwdriver set, wire stripper, multimeter, cable markers, and ties for wiring organization.

Continued on next page

Section	Content
Procedure	1. Preparation and Safety - Connect the PLC output Q0.0 to an indicator lamp or LED via 24 V DC circuit. - Open Siemens TIA Portal and create a new project. - Configure PLC hardware (e.g., CPU 1214C DC/DC/DC). - Assign addresses: I0.0 = Start push button, Q0.0 = LED output.
	2. Construct Ladder Logic <i>Network 1 - Start and Timer Trigger</i> - Insert NO contact (I0.0) connected to a TON (On-Delay Timer) block. - Set timer tag (e.g., T1) and preset time (T#1s). - Use NOT (T1.Q) contact in series to reset timer automatically, creating a repeat cycle. <i>Network 2 - LED Output Control</i> - Connect the timer output (T1.Q) to output coil Q0.0 (LED). - The LED remains ON while T1.Q is TRUE and turns OFF when T1 resets.
	Logic Explanation: - The timer continuously switches ON and OFF due to its self-reset condition. - The LED blinks at an interval equal to twice the preset delay (ON + OFF cycle).
Notes	3. Download and Test - Compile and download the project to the PLC. - Switch CPU to RUN mode. - Press Start button (I0.0) to activate the blinking function. - Observe LED flashing ON and OFF continuously with approximately 1 s per state.
	- Modify preset time (T#0.5s, T#2s, etc.) to adjust blink frequency. - TP (Pulse Timer) can be used for single-pulse output instead of cyclic blinking. - Avoid pressing Start repeatedly during timer operation. - Observe timer bits (EN, TT, Q) in TIA Portal online monitoring for learning.

Continued on next page

Module-2: Perform Basic PLC Programming

Section	Content
Key Technical Points	- TON timer energizes output after a preset ON-delay interval. - Using NOT (T1.Q) as a feedback creates continuous ON/OFF oscillation. - Blinking frequency = $(2 \times \text{preset time})$. - Timer-controlled blinking is commonly used in status indicators, machine fault alarms, and visual signaling systems.

2.3.11 Job Sheet - Create a Traffic Light Control System Using Timers in Ladder Logic

Section	Content
Job Title	Create a Traffic Light Control System Using Timers in Ladder Logic
Personal Protective Equipment (PPE)	Safety gloves, ESD wrist strap, safety shoes, and protective goggles must be worn when connecting or testing the PLC hardware.
Materials Required	Siemens S7-1200 or S7-1500 PLC, TIA Portal software (v15 or higher), 24 V DC power supply, three indicator lamps (Red, Yellow, Green), one NO push button (Start), Ethernet/USB programming cable, and resistors (if using external LEDs).
Tools and Equipment	Laptop/PC with TIA Portal installed, multimeter, screwdriver set, wire stripper, cable markers, and cable ties.
Procedure	1. Preparation and Safety - Connect three output lamps (Red, Yellow, Green) to PLC output terminals Q0.0, Q0.1, and Q0.2 respectively. - Connect a push button (Start) to input terminal I0.0. - Open Siemens TIA Portal and create a new project. - Configure CPU (e.g., 1214C DC/DC/DC) and assign I/O addresses as shown: I0.0 = Start, Q0.0 = Red, Q0.1 = Yellow, Q0.2 = Green.

Continued on next page

Section	Content
	2. Design of Timer Sequence - Create sequential logic using TON timers to simulate a traffic light cycle. Example timing: Red = 5 s, Yellow = 2 s, Green = 5 s. - Each lamp is energized for its preset time before the next activates. - Use memory bits and timer completion signals (T.Q) to trigger the next stage.
	3. Ladder Logic Development <i>Network 1 - Start Logic:</i> NO contact (I0.0) → Coil (M0.0 Run bit). <i>Network 2 - Red Timer (T1):</i> When M0.0 AND NOT T3.Q → TON T1 (5 s); Output Q0.0 = T1.EN. <i>Network 3 - Yellow Timer (T2):</i> Triggered when T1.Q = TRUE → TON T2 (2 s); Output Q0.1 = T2.EN. <i>Network 4 - Green Timer (T3):</i> Triggered when T2.Q = TRUE → TON T3 (5 s); Output Q0.2 = T3.EN. <i>Network 5 - Cycle Reset:</i> When T3.Q = TRUE → Reset all timers and restart sequence (M0.0 remains TRUE).
	4. Download and Testing - Compile and download the program to the PLC. - Switch the CPU to RUN mode. - Press Start button and observe sequence: RED → YELLOW → GREEN → RED. - Verify cycle duration and transition timing.
Notes	- Adjust timer preset values to simulate real-world intervals. - Test each lamp output manually before running full automation. - Use online monitoring in TIA Portal to visualize timer progress. - Avoid overlapping ON states between signals by using interlock conditions. - Extend logic for pedestrian signals or two-way intersections using additional timers.
Key Technical Points	- Traffic light control is a sequential timer application using TON blocks. - Each timer's output acts as a trigger for the next event. - A reset loop ensures continuous automatic operation. - Timer synchronization and interlocks are crucial for safety. - This principle is also used in conveyor sequencing and automated machine cycles.

2.3.12 Job Sheet - Count Car Garage Free Space Using Counter in Ladder Logic

Section	Content
Job Title	Count Car Garage Free Space Using Counter in Ladder Logic
Personal Protective Equipment (PPE)	Safety gloves, ESD wrist strap, safety shoes, and protective eyewear must be worn while working with PLC circuits.
Materials Required	Siemens S7-1200 or S7-1500 PLC, TIA Portal software (v15 or higher), two sensors or push buttons (Entry and Exit), indicator lamps or display for “Full” and “Free” status, counter functions (CTU/CTD), 24 V DC power supply, and Ethernet/USB programming cable.
Tools and Equipment	Laptop/PC with TIA Portal installed, multimeter, screwdriver, wire stripper, marker labels, and cable ties.
Procedure	1. Preparation and Safety - Connect Entry and Exit sensors (or push buttons) to PLC input terminals. - Configure I/O addresses: I0.0 = Entry Sensor, I0.1 = Exit Sensor, Q0.0 = Garage Full Indicator, Q0.1 = Free Space Indicator. - Assume total garage capacity = 10 slots. - Verify proper 24 V DC wiring and sensor signal polarity.

Continued on next page

Section	Content
	2. Program Setup in TIA Portal - Create a new TIA Portal project and configure CPU (e.g., 1214C DC/DC/DC). - Open Main [OB1] and insert the following logic sequence: <i>Network 1 - Count Up (CTU):</i> - Input: I0.0 (Entry Sensor). - Counter tag: C1 (CTU). - Preset value: 10 (maximum capacity). - Function: Increments count each time a vehicle enters. <i>Network 2 - Count Down (CTD):</i> - Input: I0.1 (Exit Sensor). - Counter tag: C1 (CTD). - Function: Decrements count when a vehicle exits. <i>Network 3 - Display Logic:</i> - If count < 10 → “Free Space” lamp (Q0.1) ON. - If count = 10 → “Garage Full” lamp (Q0.0) ON. <i>Network 4 - Reset Logic (Optional):</i> - Add push button I0.2 for resetting the counter to zero if needed.
Notes	- Use rising-edge (positive pulse) logic to avoid multiple counts from a single input. - Simulate sensors using push buttons during testing. - Verify preset (PV) matches total parking capacity. - Real-world systems use photoelectric or proximity sensors for detection. - Optionally, display remaining slots on an HMI or 7-segment display.
Key Technical Points	- CTU (Count Up): Increments counter on each entry event. - CTD (Count Down): Decrements counter on each exit event. - Combined CTU/CTD counters track current occupancy dynamically. - Logic interlocks prevent new entries when preset value is reached. - Applications include parking systems, conveyor part counting, and batch process control.

2.3.13 Information Sheet: Use of Different Blocks in TIA Portal

The structure of the user program in TIA Portal is defined by various block types that organize automation logic efficiently. The total number of blocks (OBs, FCs, FBs, and DBs) is limited to 1024.

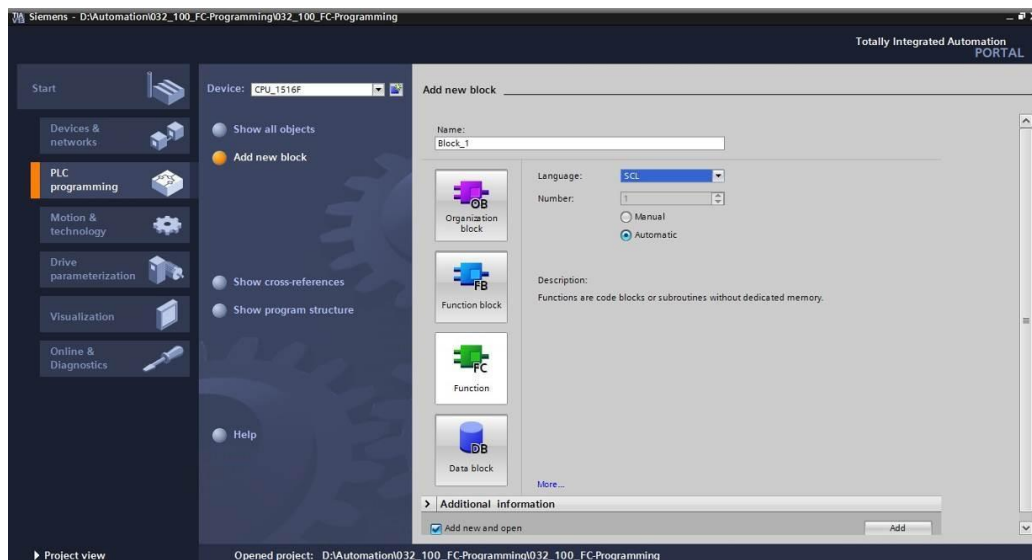


Figure 2.27: TIA Portal program structure

Block Type	Description	Memory and Data Handling
Organization Block (OB)	Interface between operating system and user program; event-driven.	Defines program structure. Examples: Main [OB1], startup, interrupt, error OBs.
Function (FC)	Logic block without memory; used for specific tasks.	No data retention after execution; parameters must be linked at call.
Function Block (FB)	Code block with memory for multi-cycle control.	Stores static tags in Instance DB (IDB) for state retention.
Data Block (DB)	Pure data storage blocks without logic.	Global DBs share data; Instance DBs hold local FB data.

Table 2.20: Block Types and Their Characteristics

2.3.14 Information Sheet: How to Use Monitor Mode (Monitoring Program Blocks)

Monitoring mode allows you to observe the execution and data flow of your program online with the CPU.

1. **Go Online:** Establish a connection to the CPU.

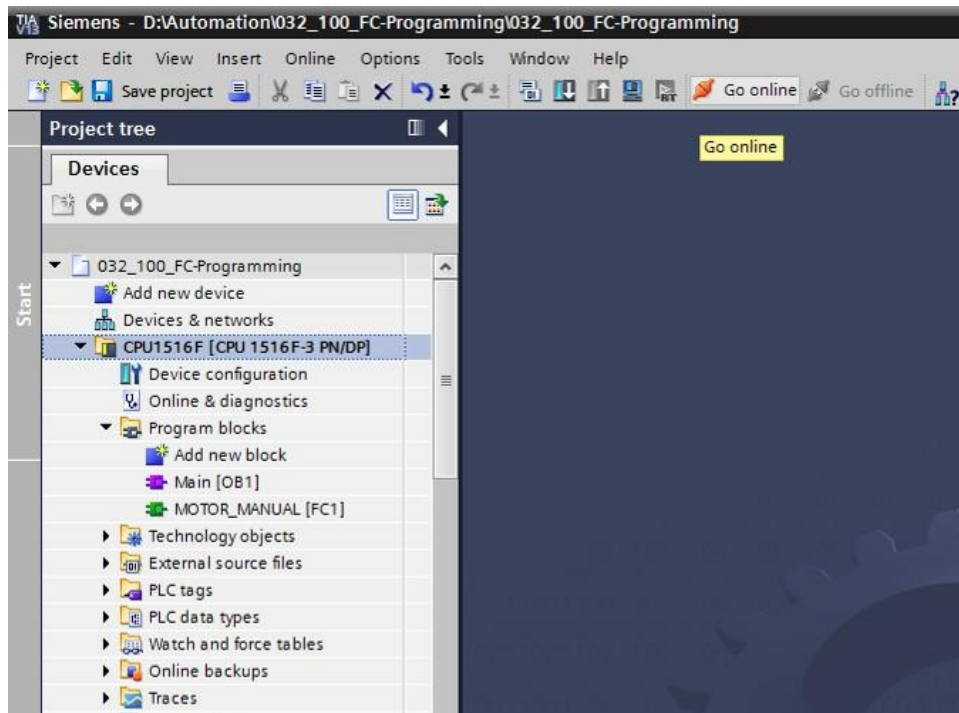


Figure 2.28: Go online and monitor program blocks

2. **Open the Block:** Open OB, FC, or FB for monitoring.
3. **Activate Monitoring:** Click the “Monitoring on/off” icon (glasses symbol).
4. **View Status:**
 - LAD/FBD editors display live logic with green power flow.
 - Monitor up to 50 tags with real-time values.
5. **Monitoring Types:** Can be signal-related or function-related.
6. **Temporary Modification:** Right-click or use Watch Table to change tag values (Modify Now).
7. **Monitoring Data Blocks (DBs):** View live DB data; capture snapshots for resets.
8. **Monitor Program Blocks:** Click the monitoring icon (e.g., Main [OB1]).

2.3.15 Information Sheet: Use of Force Table

The Force Table provides the "**Force**" function—an advanced testing tool that overwrites physical I/O values regardless of logic.

1. **Purpose:** Assigns fixed values to physical I/O for diagnostics and testing, even in STOP mode.
2. **Access:** Found under “Online and Diagnostics” in TIA Portal.

Module-2: Perform Basic PLC Programming

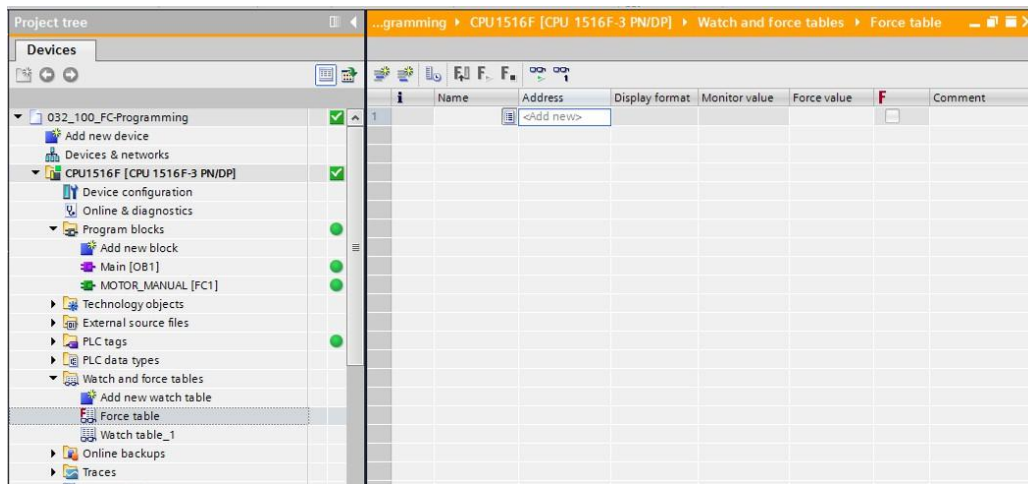


Figure 2.29: Force table overview

3. **Addressing Requirement:** Forcing applies to peripheral addresses (e.g., Q1.7:P).

Name	Address	Display format	Monitor value	Force value	Comment
<Add new>					
-P7		Bool	%Q1.3		display cylinder...
-Q1		Bool	%Q0.0		conveyor motor...
-Q2		Bool	%Q0.1		conveyor motor...
-Q3		Bool	%Q0.2		conveyor motor...
-S0		Bool	%I0.2		mode selector ...
-S1		Bool	%I0.3		pushbutton aut...
-S2		Bool	%I0.4		pushbutton aut...
-S3		Bool	%I1.4		pushbutton ma...

Figure 2.30: Force value entry example

4. **Mechanism and Persistence:**

- Forced values overwrite logic inputs/outputs.
- Stored in CPU memory; MAINT LED indicates active forcing.
- Forced values persist even if disconnected or CPU stopped.

5. **Starting/Stopping Forcing:**

- Enter desired value → Enable Force checkbox → “Start or Replace Forcing”.
- Stop forcing via “Stop Forcing” in Force Table.

6. **Limitations:** Forcing not available for internal memory (I, Q, DB, T, C, M) or HSC/PWM/PTO points.

2.3.16 Information Sheet: Using Online and Diagnostics Feature

The “Online & Diagnostics” feature offers tools for communication, commissioning, and monitoring PLC status.

1. Establishing Online Connection

- Select the CPU and click “Go Online”.

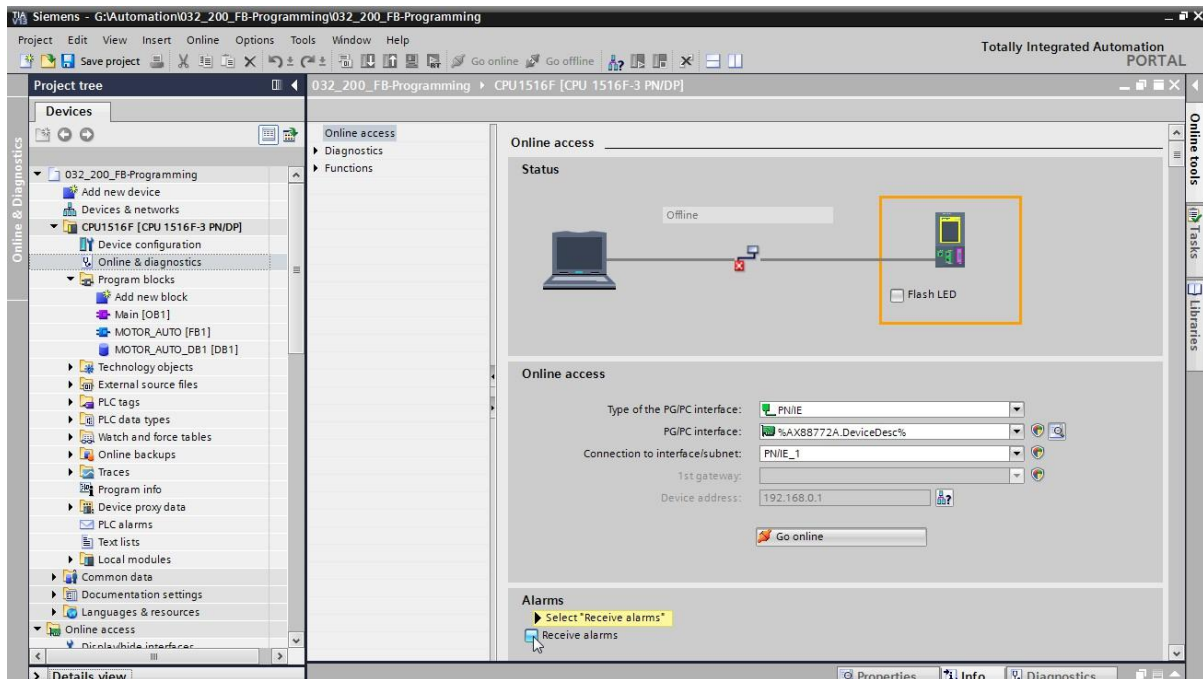
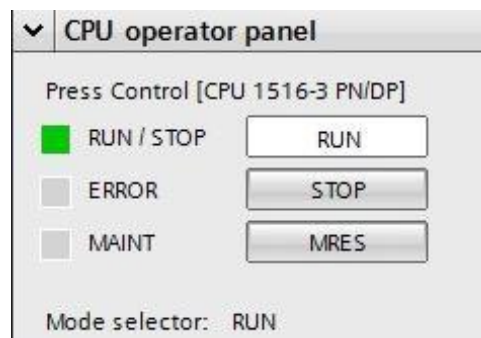


Figure 2.31: Establishing online connection

- Choose the correct PG/PC interface and adapter.
- Header turns orange; green circle = synchronized CPU/project.

2. CPU Control and Information

- **Operating Panel:** Allows RUN/STOP/MRES control.



(a) CPU operator panel

Position	Meaning	Explanation
RUN	RUN mode	The CPU processes the user program.
STOP	STOP mode	The CPU is not executing the user program.
MRES	Memory reset	Position for CPU memory reset.

(b) Status monitoring view

Figure 2.32: CPU online control tools

Module-2: Perform Basic PLC Programming

- **Status Monitoring:** Shows CPU info, cycle time, and memory use.
- **Settings:** Adjust IP, clock, or update firmware.

3. Diagnostic Buffer and Alarms

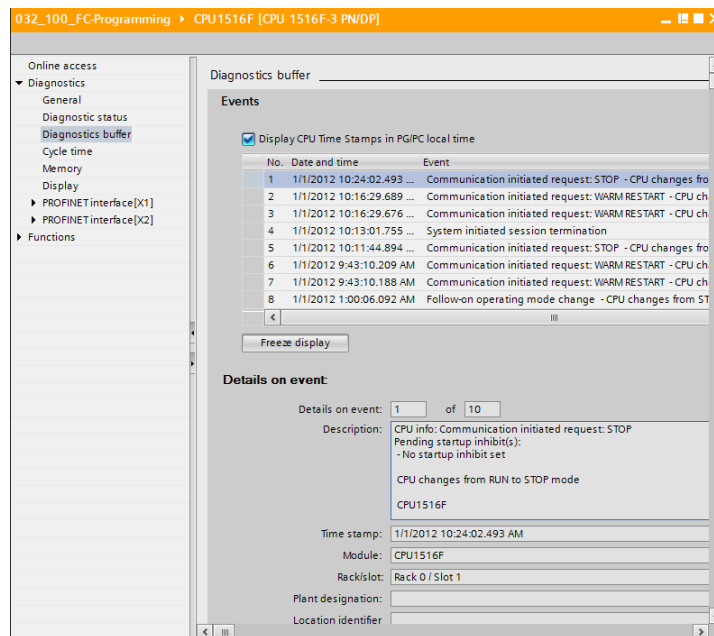


Figure 2.33: Diagnostics buffer view

- Displays chronological fault entries and alarm texts.
- Faults also appear on CPU display and Web server.

4. Hardware and Online/Offline Comparison

- Device configuration shows diagnostic colors (yellow = maintenance, red = fault).
- “Compare Offline/Online” highlights differences and syncs via Download/Upload.

Symbol	Meaning
	Folder contains objects whose online and offline versions are different (only in the project tree)
	Online and offline versions of the object are different
	Object only exists online
	Object only exists offline
	Online and offline versions of the object are the same

Figure 2.34: Hardware comparison and synchronization

Note: Minor edits can be downloaded during RUN. For >20 modified blocks, CPU must be STOP.

2.3.17 Job Sheet - Use Math Functions to Solve Equation in Ladder Logic

Section	Content
Job Title	Use Math Functions to Solve Equation in Ladder Logic
Personal Protective Equipment (PPE)	ESD wrist strap, safety gloves, safety shoes, and safety goggles must be worn while handling PLC hardware and connections.
Materials Required	Siemens S7-1200 or S7-1500 PLC, TIA Portal software (v15 or higher), 24 V DC power supply, PC/Laptop with Ethernet or USB cable, and a predefined Data Block (DB1) for variable storage.
Tools and Equipment	Laptop/PC with TIA Portal installed, Ethernet/USB programming cable, and multimeter (optional).
Procedure	1. Preparation and Setup - Open TIA Portal and create a new project. - Configure CPU (e.g., 1214C DC/DC/DC) and create Data Block (DB1). - Define the following variables in DB1: a: REAL, b: REAL, c: REAL, x: REAL (Result). - Assign initial values to a, b, and c manually or via HMI. 2. Define the Target Equation $x = \frac{(2a)}{(b + 5)} - (7c)$ The goal is to calculate x automatically using PLC math instructions and display the result in memory.

Continued on next page

Section	Content
	3. Ladder Logic Implementation <i>Network 1 - Multiply a by 2:</i> Use MUL block → $\text{Temp1} = a \times 2$. <i>Network 2 - Add b with 5:</i> Use ADD block → $\text{Temp2} = b + 5$. <i>Network 3 - Divide Temp1 by Temp2:</i> Use DIV block → $\text{Temp3} = \text{Temp1} \div \text{Temp2}$. <i>Network 4 - Multiply c by 7:</i> Use MUL block → $\text{Temp4} = c \times 7$. <i>Network 5 - Subtract Temp4 from Temp3:</i> Use SUB block → $x = \text{Temp3} - \text{Temp4}$. <i>Network 6 - Store Result:</i> Use MOVE instruction to transfer $\text{DB1.x} \rightarrow \text{MW10}$.
	4. Download and Test - Download the project to PLC and switch CPU to RUN mode. - Monitor variable values in real time from TIA Portal. - Adjust values of a, b, and c and confirm that x updates automatically.
	5. Optional Task - Recreate the same equation inside a “Calculate” program block for modular comparison.
Notes	- All variables must be declared as REAL to support floating-point arithmetic. - Use temporary memory (M) addresses for intermediate variables. - Avoid division by zero ($b \neq -5$). - Display x on HMI numeric field or monitor directly in TIA Portal.
Key Technical Points	- PLC arithmetic operations (ADD, SUB, MUL, DIV) execute with REAL precision. - Data Blocks (DB) store persistent and structured process variables. - MOVE instruction transfers computed data to memory or output registers. - Mathematical logic is crucial for process control, analog scaling, and computation tasks in PLC systems. - Complex equations can be modularized using Function Blocks (FBs) for reusability.

2.3.18 Job Sheet - Read 0-10 V Potentiometer Value Using NORM_X and SCALE_X

Section	Content
Job Title	Read 0-10 V Potentiometer Value Using NORM_X and SCALE_X
Personal Protective Equipment (PPE)	ESD wrist strap, insulated gloves, safety shoes, and protective goggles must be used when working with analog circuits and PLC hardware.
Materials Required	Siemens S7-1200 or S7-1500 PLC with analog input module (e.g., SM1231 AI 4 × 13-bit), TIA Portal software (v15 or higher), 0-10 V potentiometer, 24 V DC power supply, multimeter for voltage verification, and analog input wiring cables.
Tools and Equipment	Laptop/PC with TIA Portal installed, Ethernet/USB programming cable, screwdriver, wire stripper, marker labels, and cable ties.
Procedure	1. Preparation and Wiring - Connect the potentiometer ends to +24 V DC and 0 V. - Connect the wiper terminal to analog input AI0 of the PLC. - Power ON the PLC and open TIA Portal. 2. Hardware Configuration - Open Device Configuration in TIA Portal. - Select analog input AI0 and set measuring range to 0-10 V. - Note: For 13-bit resolution, 0-10 V corresponds to raw counts 0 - 27 648 (INT). 3. Tag Definition Create the following variables: AI_Voltage (INT) → from IW64 (analog input) Normalized (REAL) → result of NORM_X Scaled (REAL) → result of SCALE_X

Continued on next page

Section	Content
	4. Program Implementation (Main [OB1]) <i>Network 1 - Read Analog Input:</i> Assign AI_Voltage := IW64. <i>Network 2 - Normalize Using NORM_X:</i> Block NORM_X → IN = AI_Voltage, HI_LIM = 27648, LO_LIM = 0, OUT = Normalized (0.0-1.0). <i>Network 3 - Scale Using SCALE_X:</i> Block SCALE_X → IN = Normalized, HI_LIM = 100.0, LO_LIM = 0.0, OUT = Scaled (0-100 %). <i>Network 4 - Display Output:</i> Use MOVE (Scaled, MD10) to transfer scaled value to memory and display in Watch Table or HMI numeric field.
	5. Download and Test - Compile and download the program to the PLC. - Switch CPU to RUN mode. - Rotate the potentiometer knob slowly from 0 to 10 V and observe the corresponding 0-100 % output in the Watch Table. - Cross-verify with a multimeter to ensure correct voltage-to-percentage relation.
Notes	- Confirm analog wiring polarity and grounding (0 V common). - NORM_X converts raw input data to a normalized REAL (0-1). - SCALE_X converts the normalized data to user-defined engineering units. - For 13-bit modules, each bit ≈ 0.00036 V (≈ 27 648 counts / 10 V). - This scaling method is standard for analog sensors such as level, temperature, pressure, or speed.
Key Technical Points	- Analog channels output raw integer values that must be scaled to engineering ranges. - NORM_X normalizes any signal range to 0-1. - SCALE_X converts normalized values to desired measurement units (e.g., 0-100 %). - Proper scaling ensures accuracy in control loops and data visualization. - Essential for analog process control, PID tuning, and real-time monitoring in automation systems.

2.3.19 Self-Check Questions

No.	Question
1	List the main code block types used in TIA Portal (e.g., OB, FB, FC) and their purpose.
2	What are tag tables and how are they used in a PLC program?
3	Name two advantages of using Ladder for discrete control logic.
4	What steps are required before downloading a program to the CPU?

2.3.20 Answer Key

No.	Answer / Guidance
1	OB: organization block for scheduling/execution; FB: function block with memory; FC: reusable function without its own memory.
2	Symbol tables mapping symbolic names to addresses; used to reference I/O and internal memory.
3	Familiar relay-logic representation; easy to visualize discrete interlocks and sequences.
4	Compile without errors, establish online connection, set the correct target device, ensure version compatibility and stop CPU if required by configuration.

2.4 Learning Outcome: Design HMI for PLC

2.4.1 Information Sheet: HMI Installation and Communication Setup

The HMI provides a dashboard for operating and monitoring machines and plants in Runtime, connecting a person to a machine, system, or device.

The HMI configuration is handled by **SIMATIC WinCC Basic** software, which is an integral component of STEP 7 Basic or STEP 7 Professional within the TIA Portal.

1. Hardware Overview

SIMATIC HMI devices such as **Basic Panels** (e.g., KTP400, KTP700) or **Comfort Panels** (e.g., TP1200) are typically used for local operator input and monitoring tasks. These devices communicate with the controller via **Ethernet** or **PROFINET**.

Rear view of the TP700 Comfort

Front view of the TP700 Comfort

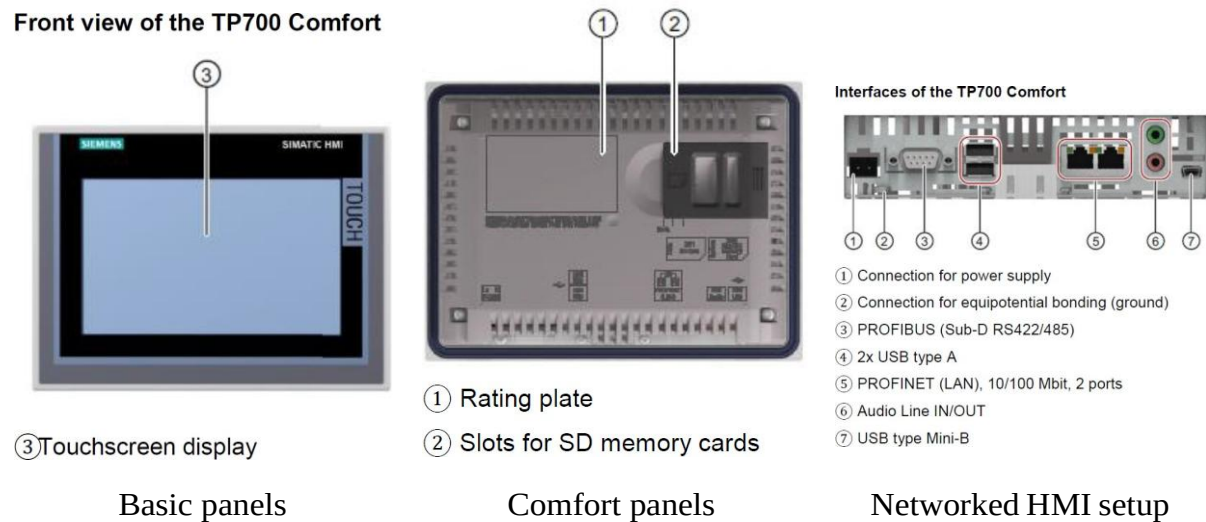


Figure 2.35: Typical HMI hardware setup

2. Physical Setup and IP Addressing from TIA Portal

Before configuration, the HMI device must be on the same network as the CPU. Since the CPU does not have a pre-configured IP, both IP and subnet mask must be assigned manually.

1. Open the TIA Portal.
2. Select "**Online & Diagnostics**" → "**Project view**".
3. In the project tree, select the computer's network adapter under "**Online access**".
4. Click "**Update accessible devices**" to find the HMI.
5. Open "**Online & diagnostics**".
6. Select "**Assign IP address**" and enter:
 - IP: 192.168.0.10

Module-2: Perform Basic PLC Programming

- Subnet: 255.255.255.0

7. Click "**Assign IP address**" and verify result in the "**Info**" window.

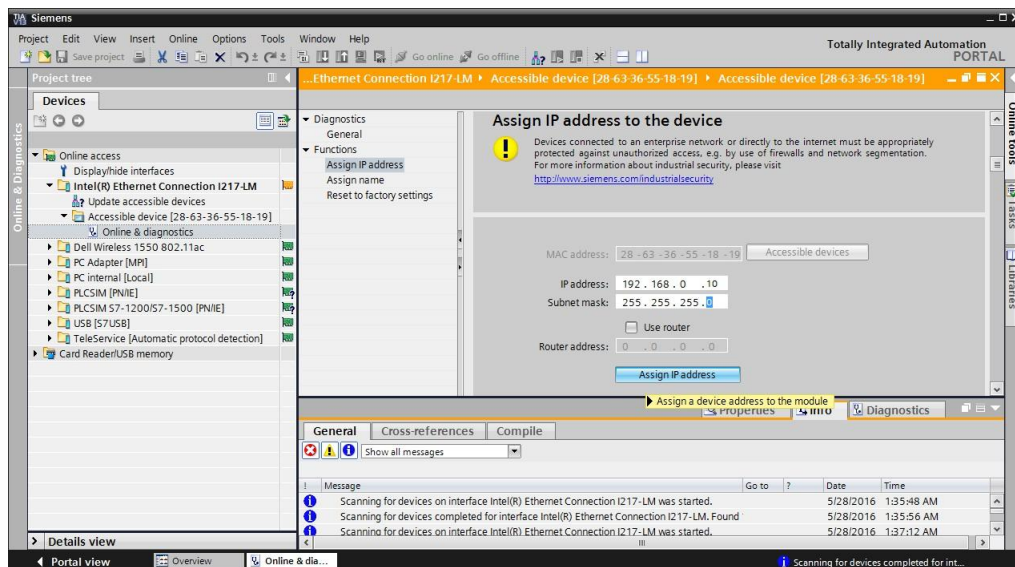
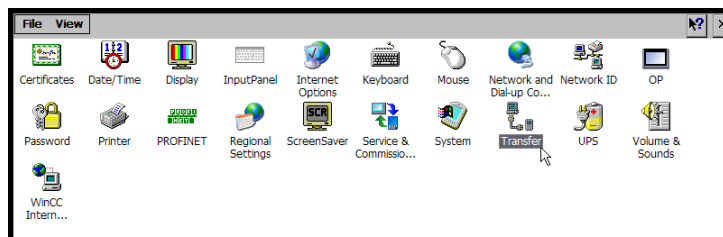


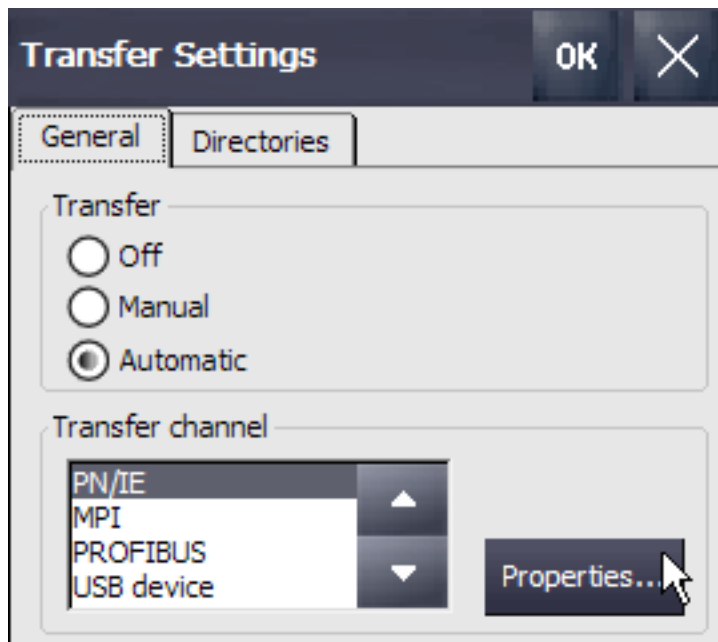
Figure 2.36: Assigning IP address to HMI in TIA Portal

3. Assign HMI IP Address from HMI Panel

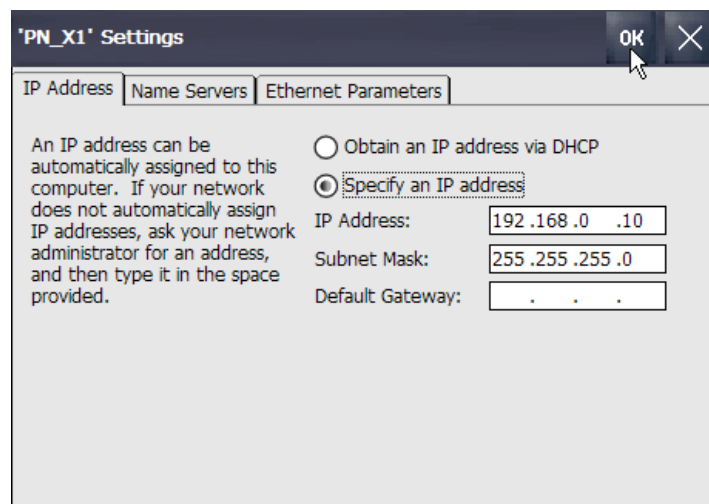
1. Select the network setup icon to open **Transfer Settings**.
2. Choose **Automatic Transfer** from WinCC Advanced.
3. Select **Ethernet PN/IE** as the transfer channel.
4. Click the setup button to assign IP and confirm settings.



Transfer settings dialog



Select Ethernet interface



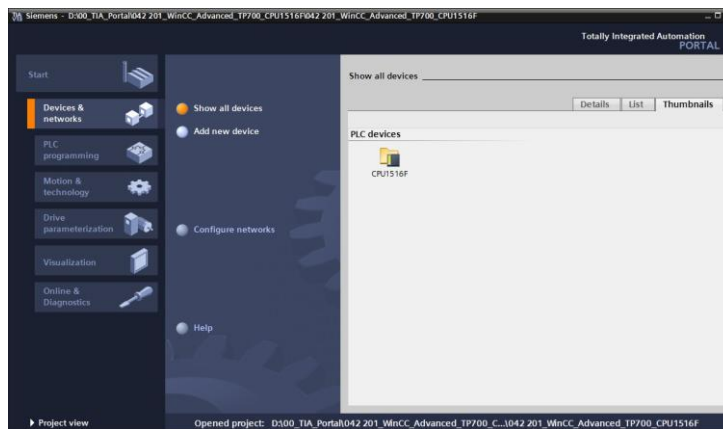
Configure IP parameters

Figure 2.37: IP assignment from HMI interface

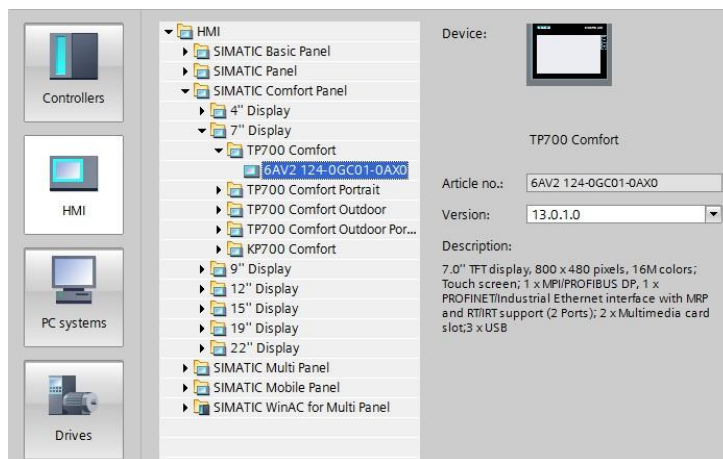
Module-2: Perform Basic PLC Programming

2.4.2 Information Sheet: Configuration of HMI Device and Connection

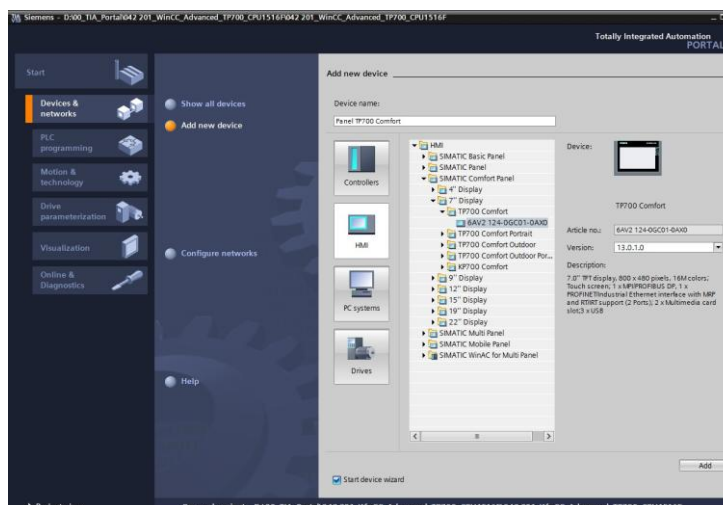
HMI configuration involves integrating the device into the PLC project and defining its data exchange with the CPU.



Launch the HMI wizard



Select SIMATIC HMI family



Confirm device selection

Figure 2.38: Adding HMI device in TIA Portal

1. Adding the HMI Device to the TIA Project

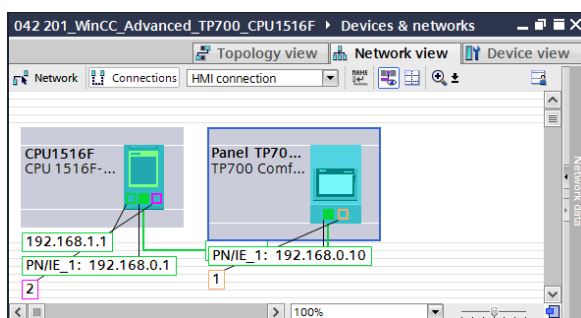
1. In TIA Portal, open "**Add new device**".
2. Click "**SIMATIC HMI**".
3. Choose the specific HMI model (e.g., KTP700 Basic or TP1200 Comfort).
4. Click "**OK**" to add. Alternatively, drag from the Global Library.

2. Creating the Network Connection

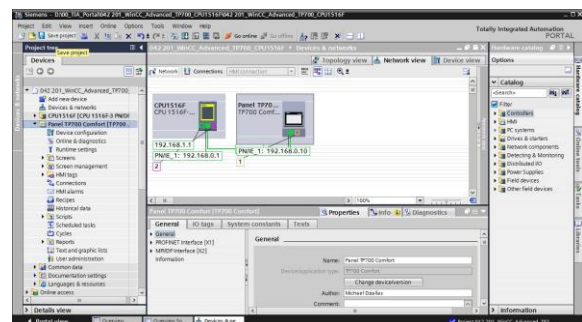
1. Go to "**Devices and Networks**" → Network View.
2. Drag a connection between the CPU's and HMI's Ethernet ports to form a PROFINET link.

3. Configuring the HMI Connection (Tag Sharing)

1. Select the created network connection.
2. Click "**Connections**" → "**HMI connection**".
3. Drag from the CPU PROFINET interface to the HMI interface.
4. HMI tags are now synchronized with PLC tags.



Create PROFINET connection



Link CPU and HMI interfaces

Figure 2.39: Creating HMI connection in Network view

4. Linking PLC and HMI Tags

1. Open **HMI Tag Editor**.
2. Choose the HMI connection under **Connections**.
3. Link CPU tags to HMI display elements.

2.4.3 Information Sheet: Designing the HMI Screen (Visualization)

The screen design process involves planning, layout design, and configuration using **WinCC Basic**.

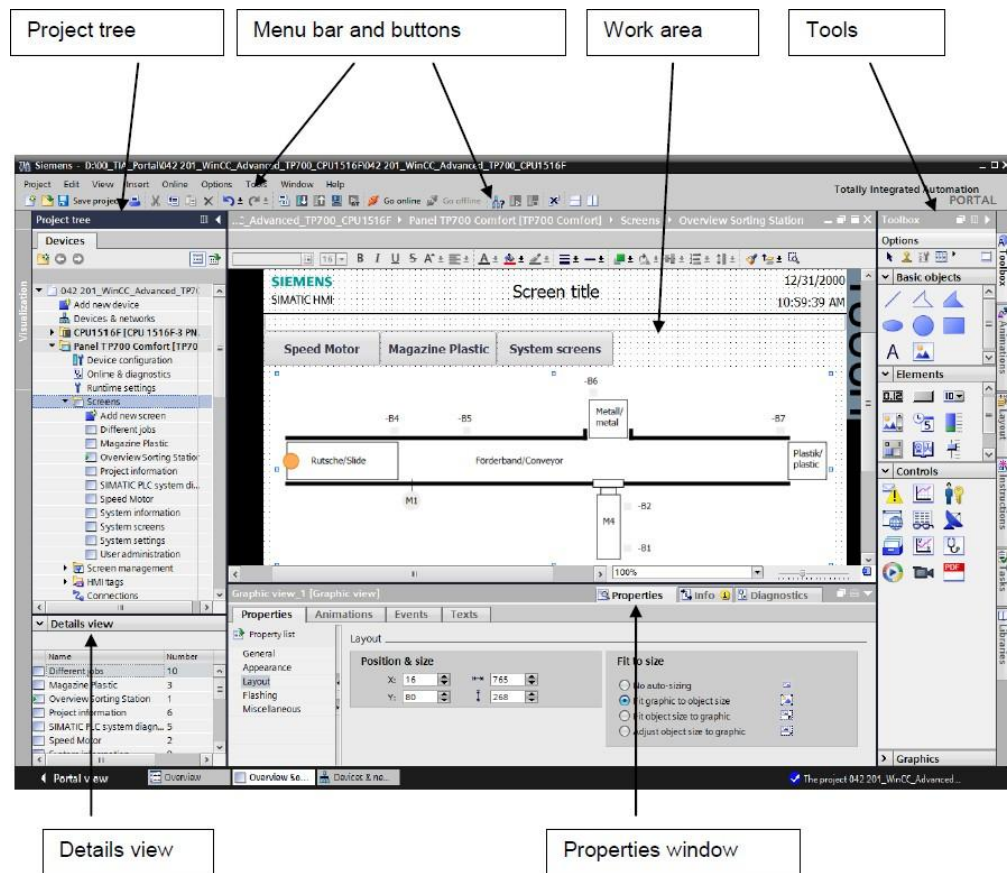


Figure 2.40: WinCC HMI screen interface

1. Planning the Screen Structure and Design

- **Screen Structure:** The entry point is the **root screen**.
- Group information hierarchically for better navigation.

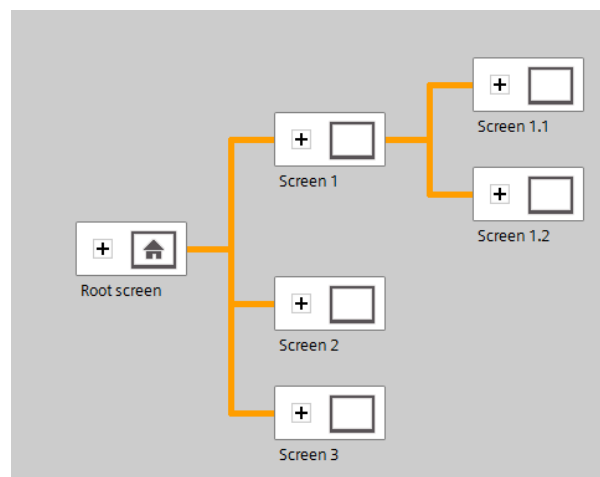


Figure 2.41: Screen hierarchy planning

Design Rules:

- Subdivide screen into work, status, and controller info.
- Keep utilization at 30–40% of total area.
- Group numbers (e.g., 66 234) and use minimal color coding.

2. Creating the Screen Layout

1. In WinCC, open "**Screens**" under Project navigation.
2. Use **Toolbox** to insert graphical objects.
3. Drag PLC tags to the screen for binding.
4. Use **Properties** to customize visuals and logic.

Tip: Use consistent colors and grouping to improve clarity and operator understanding.

2.4.4 Information Sheet: Configuring Advanced HMI Functions

WinCC Basic supports enhanced features for runtime visualization and control.

- **Recipes:** Store grouped parameters (e.g., production ratios) transferable to the PLC.
- **Alarms and Diagnostics:** Display faults, diagnostic buffers, and system events on runtime screens.
- **User Management:** Restrict access based on user levels for safety and data protection.
- **Scripts (VBS):** Define user functions to modify visualization dynamically.
- **Multilingualism:** Maintain project texts in multiple languages for operator selection.
- **Reports:** Generate runtime logs, shift reports, and alarm data summaries.

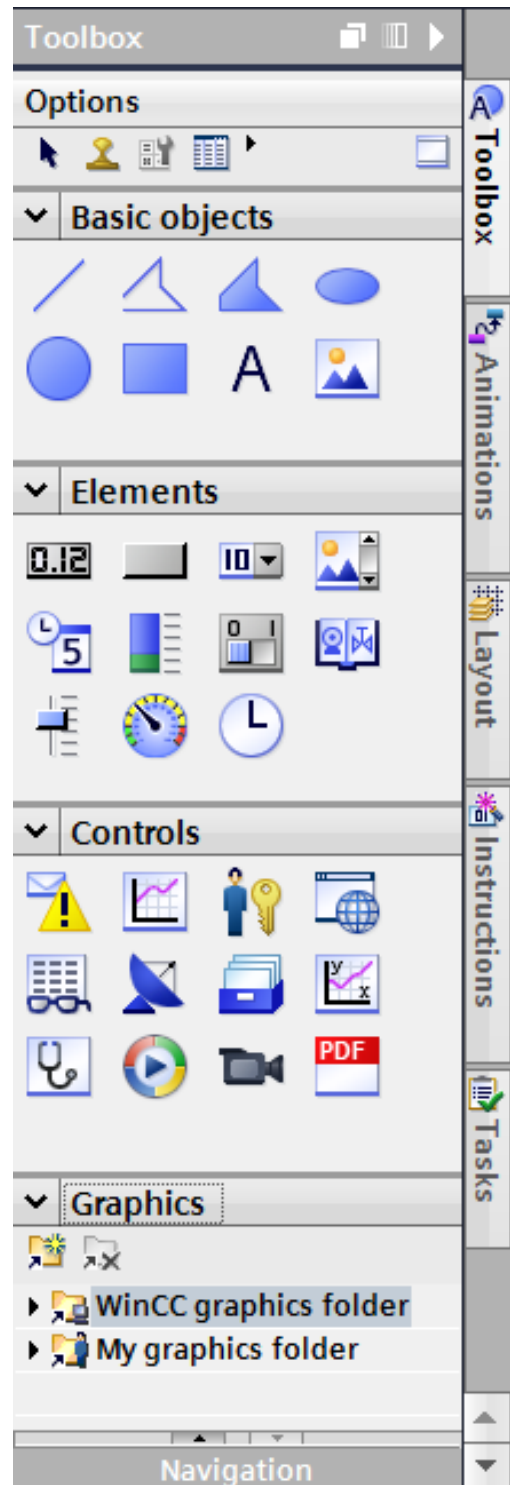


Figure 2.42: Screen layout editor in WinCC

2.4.5 Information Sheet: Testing and Finalizing the HMI

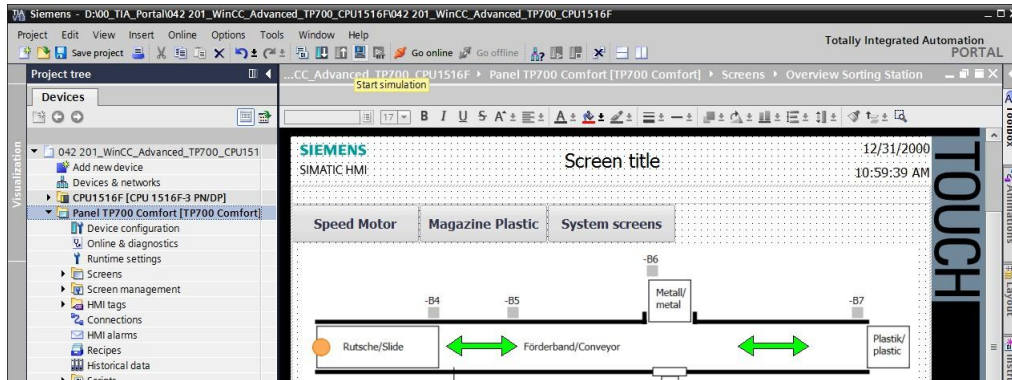


Figure 2.43: HMI testing and simulation interface

1. **Simulation:** Use the HMI Simulator (*Online* → *Simulation*) to test configuration logic before deployment.
2. Validate alarms, display updates, and interlocks.
3. **Download:** After successful testing, download the HMI project (Hardware & Software) to the physical device.

2.4.6 Self-Check Questions

No.	Question
1	What software component is used for HMI configuration within TIA Portal?
2	How are HMI tags linked to PLC tags?
3	List two common HMI screen objects used to visualize process values.

2.4.7 Answer Key

No.	Answer / Guidance
1	WinCC (within TIA Portal).
2	By assigning connection and mapping HMI tags to PLC symbolic addresses in the tag editor.
3	Lamps/indicators, I/O fields, trend views, buttons, and gauges.

Module-3: Analyze Industrial Sensors

3.1 Learning Outcome: Identify Types of Industrial Sensors

Objective The learner will be able to interpret the types of industrial sensors, identify and describe their classifications, explain the function and role of sensors in relation to industrial applications, and interpret the applications of industrial sensors.

3.1.1 Different Types of Industrial Sensors

Industrial sensors convert physical parameters into measurable electrical signals, enabling machines to interact with their environment.

- **Proximity Sensors:** Detect the presence or absence of an object within a certain range without physical contact. They are widely used for object detection, counting, and safety applications.
- **Inductive Sensors:** A type of proximity sensor that detects metallic objects using electromagnetic fields. They are ideal for harsh industrial environments due to their durability.
- **Capacitive Sensors:** Detect changes in capacitance to identify non-metallic objects or materials (e.g., plastics, liquids). They are suitable for level detection and proximity sensing.
- **Optical Sensors (Optoelectronic Sensors):** Use light beams (infrared, laser, or visible) to detect objects or measure distances. This category includes photoelectric sensors, laser distance sensors, and vision sensors.
- **Electrical Limit Switches:** Electromechanical devices that detect physical movement or position and trigger a contact mechanism. Commonly used for end-of-travel detection.
- **Pressure Sensors:** Convert pressure values of gases or liquids into electrical signals. Used in hydraulic and pneumatic systems for monitoring and control.
- **Position Sensors:** Measure the position of an object in a linear or rotary fashion, providing accurate feedback in robotic arms and actuators.
- **Magneto-resistive Proximity Switches:** Specialized non-contact sensors that detect magnetic fields, commonly used for pneumatic and hydraulic cylinder position detection.



Figure 3.1: Different types of industrial sensors used in automation systems

Classification of Sensors

Industrial sensors are broadly classified based on their interaction with the target object:

Classification	Description	Examples
Contact Type	These sensors require physical contact with the object or surface to detect a specific condition or parameter.	Limit switches, Strain gauges, Thermocouples.
Non-Contact Type (Proximity Sensors)	Detect properties of objects without physical contact, relying on magnetic, electric, or optical fields.	Inductive sensors, Capacitive sensors, Optical sensors, Reed switches.

Proximity sensors (non-contact type) are the most commonly used in industrial automation due to their long operational life, high-speed response, and resistance to harsh environments.

3.1.2 Information Sheet-2: Sensor Schematic Circuit Symbols

The following table presents some common industrial sensor symbols, often referencing the IEC 60617 standard for use in circuit diagrams:

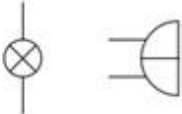
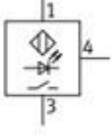
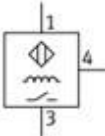
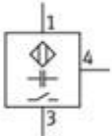
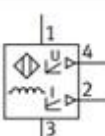
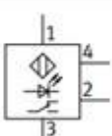
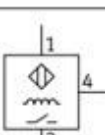
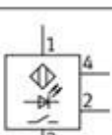
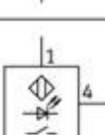
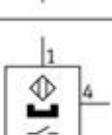
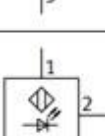
Component	Graphic symbol	Component	Graphic symbol
Indicator unit and distributor, electrical		Retro-reflective sensor	
Proximity switch, inductive, M12		Proximity switch, capacitive, M12	
Proximity sensor, inductive, with analogue output, M12		Fibre-optic unit	
Proximity switch, inductive, M18		Diffuse sensor with background suppression	
Through-beam sensor, receiver		Proximity switch, magneto-resistive	
Through-beam sensor, transmitter			

Figure 3.2: Common industrial sensor symbols (IEC 60617 reference)

3.1.3 Working Principles of Common Sensors

Optical Sensors Optical proximity sensors employ optical and electronic means for object detection, using red or infrared light. The basic mechanism involves a transmitter (emitter) and a receiver.

- **Through-Beam Sensors:** Transmitter and receiver are separate. Object detection occurs when the light beam is interrupted. Offers long detection range and high accuracy.

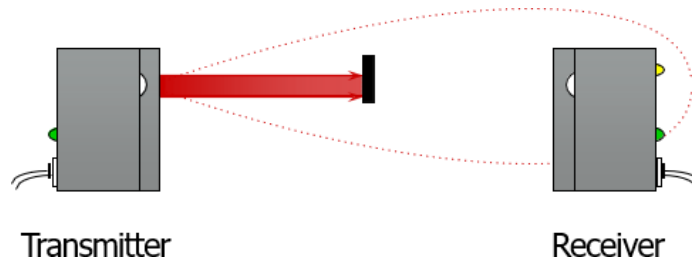
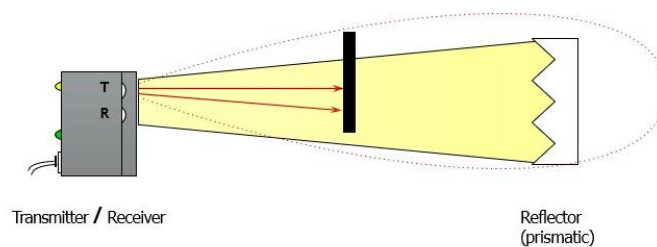


Figure 3.3: Working principle of a through-beam optical sensor

- **Retro-Reflective Sensors:** Transmitter and receiver share a single housing. Detection relies on reflected light from a reflector; interruption indicates object presence.



**Retroreflective scan sensors may not be able to detect shiny objects.

Figure 3.4: Working principle of a retro-reflective optical sensor

- **Diffuse Sensors:** Transmitter and receiver are combined; detection occurs when light reflects diffusely off an object surface.

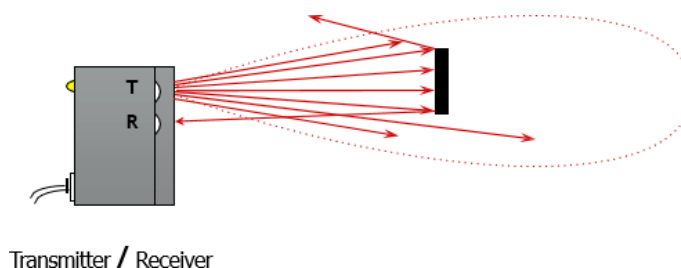
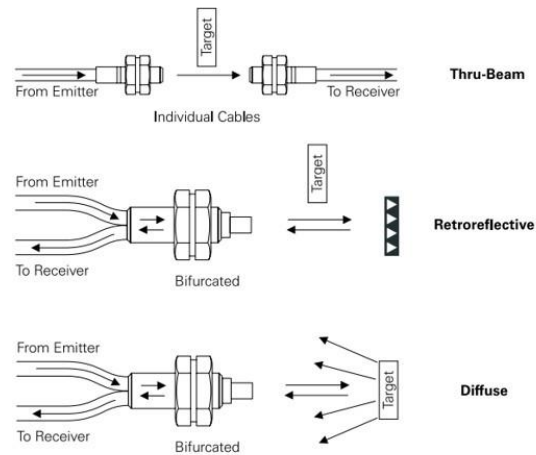


Figure 3.5: Working principle of a diffuse optical sensor

- **Fibre-Optic Cable:** Guides light to confined areas where standard sensors cannot fit, using total internal reflection.



Fiber optics can be used with thru-beam, retro-reflective or diffuse sensors.

Figure 3.6: Fibre-optic cable used for light guidance in sensors

Proximity Sensors Proximity sensors detect the presence or distance of nearby objects without physical contact. They use various technologies, including capacitive, inductive, and ultrasonic methods, to sense changes in the environment.

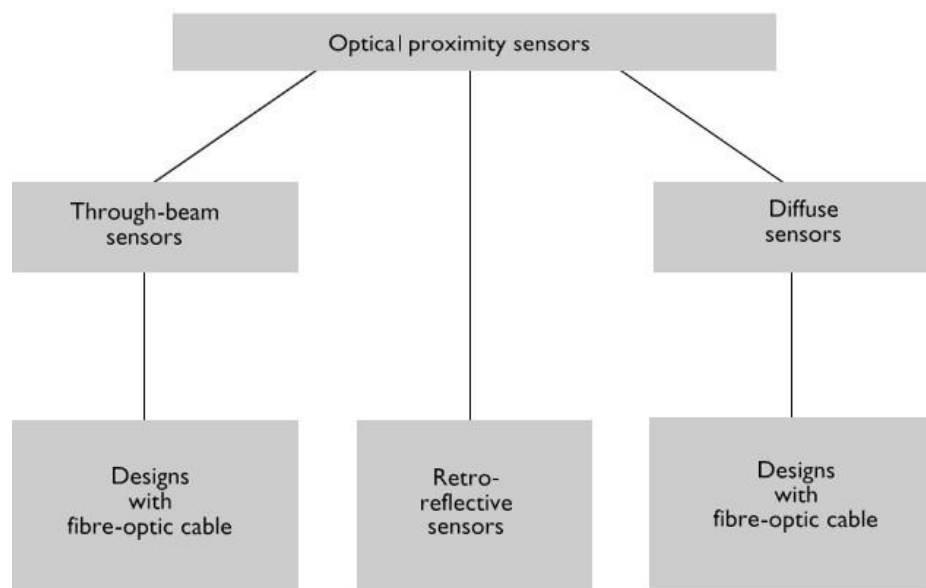


Figure 3.7: Common types of optical proximity sensors

Limit Switch (Electrical Limit Switch) Contact-type sensors that output signals upon actuation. When an object contacts the actuator (lever or plunger), it triggers internal contacts to open or close a circuit. Used for end-of-travel or mechanical positioning detection.

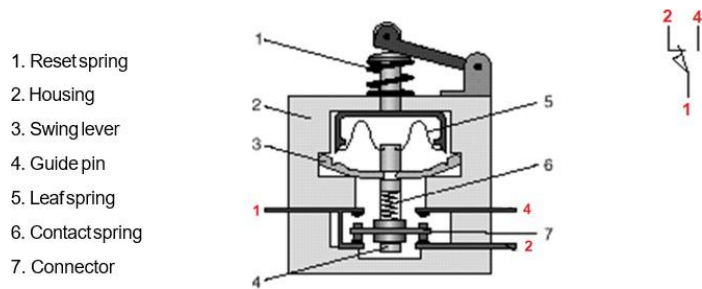


Figure 3.8: Working principle of an electrical limit switch

Pressure Switch (Pneumatic/Electric Converter) Pressure switches convert pneumatic pressure into electrical signals. They activate or deactivate circuits when a preset pressure is reached, enabling electrical signal generation from pneumatic inputs.

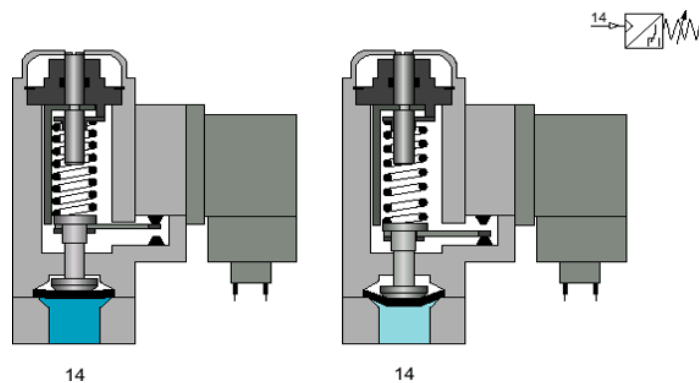


Figure 3.9: Working principle of a pressure switch

Reed Switch (Magnetic Reed Switches) Magnetically actuated proximity sensors consisting of two reeds sealed in a glass tube. When a magnetic field is present, the reeds attract, completing an electrical circuit. Commonly used as cylinder switches in pneumatic systems.

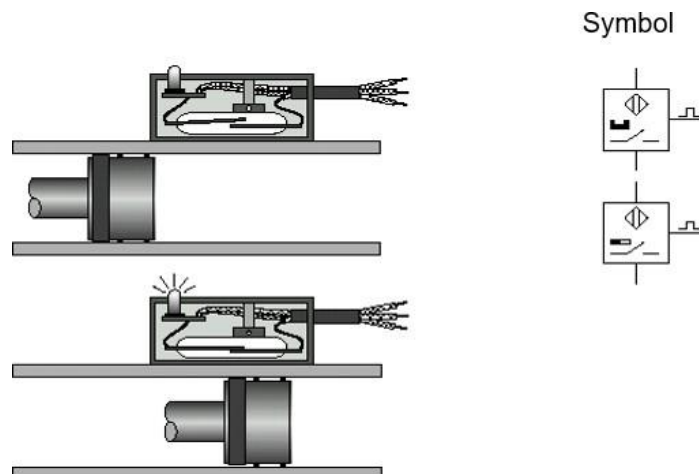


Figure 3.10: Working principle of a reed switch

Capacitive Proximity Sensors Operate on capacitance variation within an RC circuit. As an object approaches, capacitance increases, changing oscillation characteristics. Sensitivity depends on the dielectric constant (ϵ_r). High ϵ_r materials (like water, $\epsilon_r \approx 80$) are easily detected.

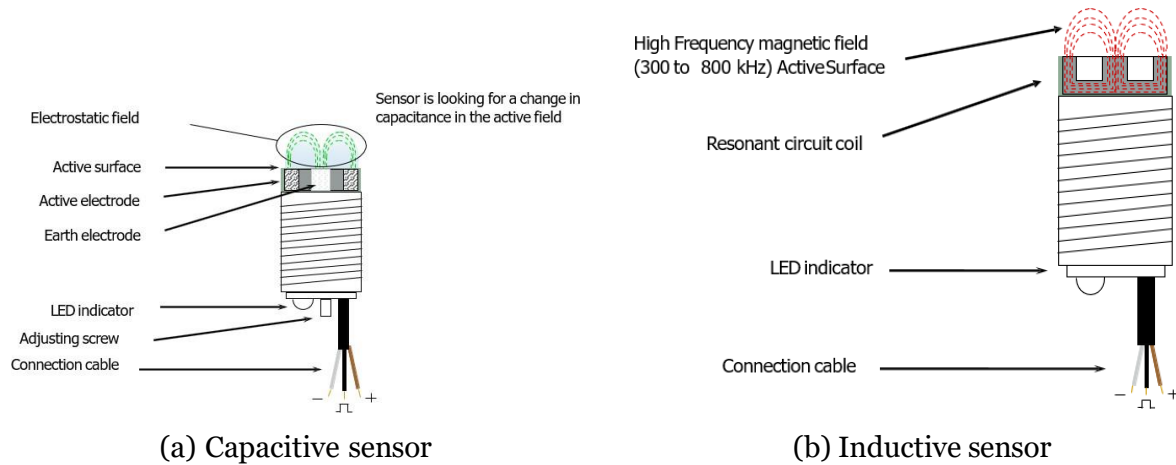


Figure 3.11: Internal structures of proximity sensors

Inductive Proximity Sensors Use an oscillator circuit with a coil to create an electromagnetic field. When a conductive material enters the field, eddy currents attenuate the oscillation, generating an output signal corresponding to target presence.

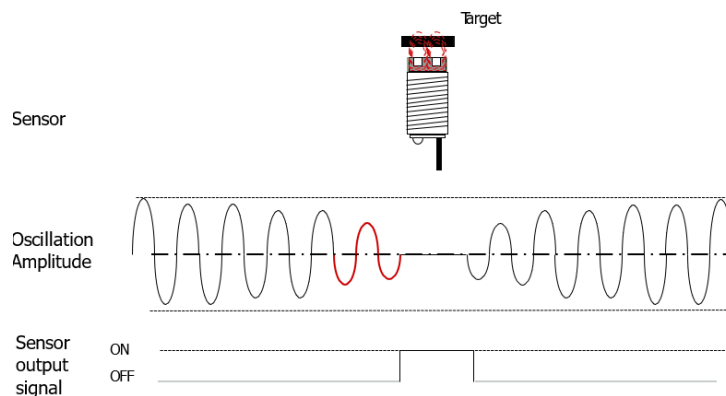


Figure 3.12: Basic working principles of inductive sensors

Magneto-resistive and Hall Effect Sensors Magnetoresistive (MR) sensors use the change in electrical resistance of a material in a magnetic field, while Hall effect sensors use the voltage generated across a conductor when a magnetic field is applied perpendicular to the current. MR sensors are often more sensitive and can detect a parallel magnetic field, while Hall effect sensors are simpler, commonly used for position sensing, and are sensitive to a perpendicular field.

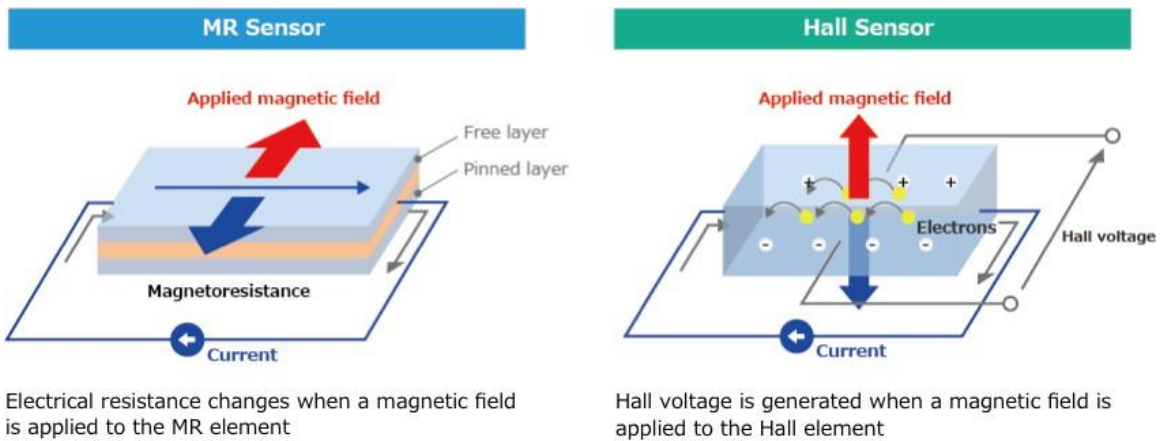


Figure 3.13: Comparison between Magneto-resistive and Hall Effect sensors working principles

Temperature Sensors

- **Thermocouples:** Operate on the Seebeck effect—voltage generated by temperature difference between two dissimilar metals.

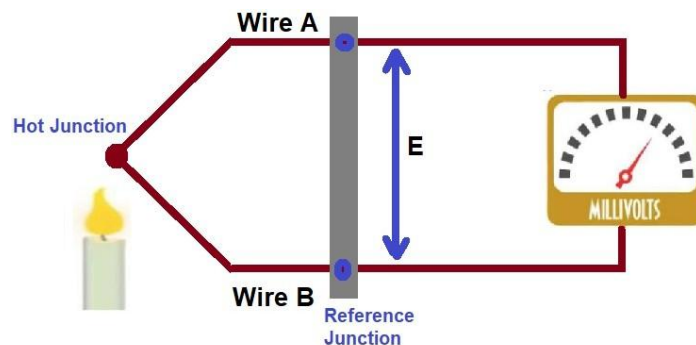


Figure 3.14: Working principle of a thermocouple.

- **RTDs (Resistance Temperature Detectors):** Measure resistance change with temperature; Platinum RTDs are known for accuracy.

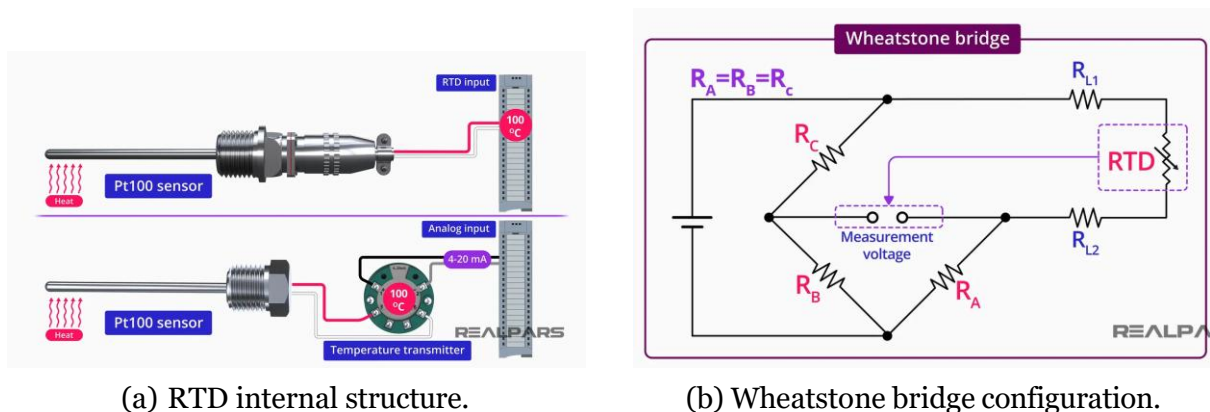


Figure 3.15: RTD sensor and its signal conditioning circuit.

- **Infrared (IR) Pyrometers:** Detect infrared radiation for non-contact temperature measurement.



Figure 3.16: Infrared (IR) pyrometer for non-contact temperature sensing.

Level Sensors

- **Float Switches:** The simplest form of level sensing, a float switch uses a buoyant object (the float) that rises and falls with the liquid level. As the float moves, it actuates a switch, indicating either a high or low level.



Figure 3.17: Float-type level switch principle.

- **Ultrasonic Sensors:** These sensors emit high-frequency sound waves towards the surface of the material. The time it takes for the sound waves to reflect back to the sensor is measured. Since the speed of sound in the medium is known, the distance to the surface, and thus the level, can be calculated. They are non-contact, making them suitable for corrosive liquids.

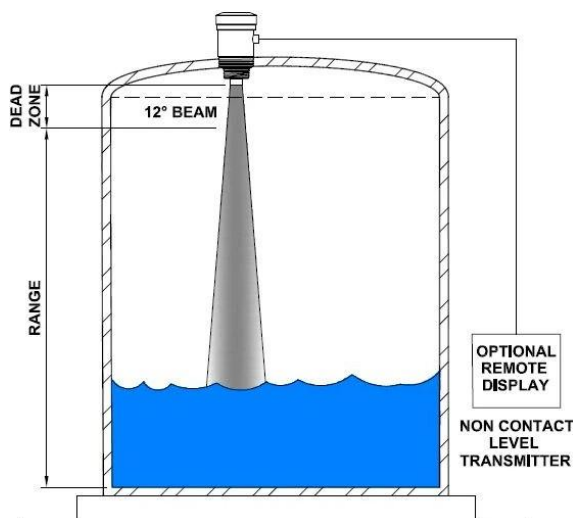


Figure 3.18: Ultrasonic level sensor operation using time-of-flight principle.

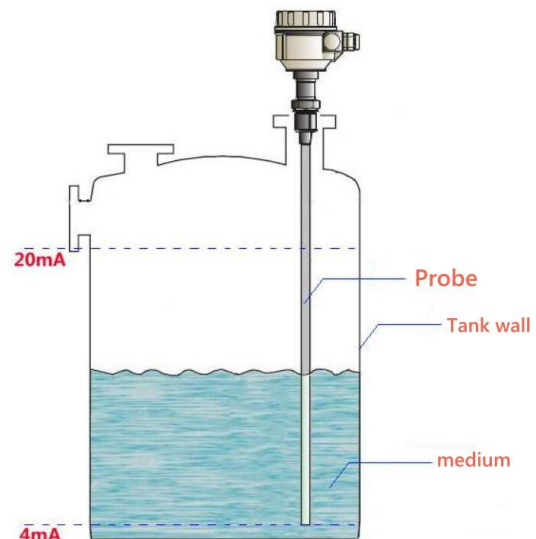


Figure 3.19: Capacitive level sensor working concept.

- **Capacitive Level Sensors:** These sensors detect the level of a material by measuring changes in capacitance. When the material (which acts as a dielectric) enters

the electric field created by the sensor, it changes the capacitance. This change is then converted into a level measurement. They are often used for detecting both liquids and solids.

Flow Rate Sensors Flow rate sensors are essential devices used across countless industries to measure the volume or mass of a fluid (liquid or gas) passing through a pipe or conduit per unit of time.

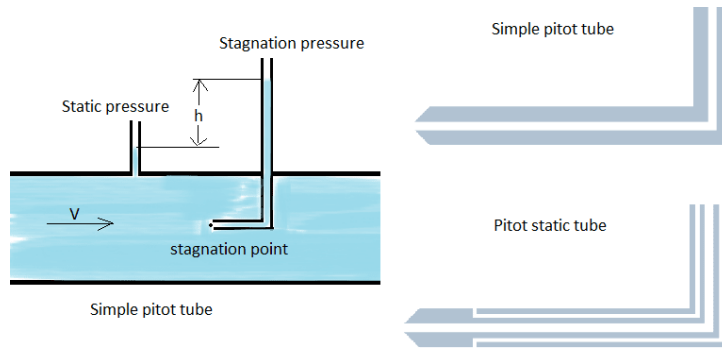


Figure 3.20: Differential pressure flow measurement using orifice plate.

- **Differential Pressure Flow Sensors:** These sensors measure the pressure drop across an obstruction in the flow path.⁴ The flow rate is proportional to the square root of the pressure differential. Orifice Plates: A simple plate with a hole constricts the flow, creating a pressure difference. Venturi Tubes: A specially shaped constriction accelerates the fluid, causing a pressure drop. Pitot Tubes: These measure the difference between static and total pressure to determine local fluid velocity.
- **Coriolis Mass Flow Sensors:** Detect mass flow through phase shift due to the Coriolis effect.

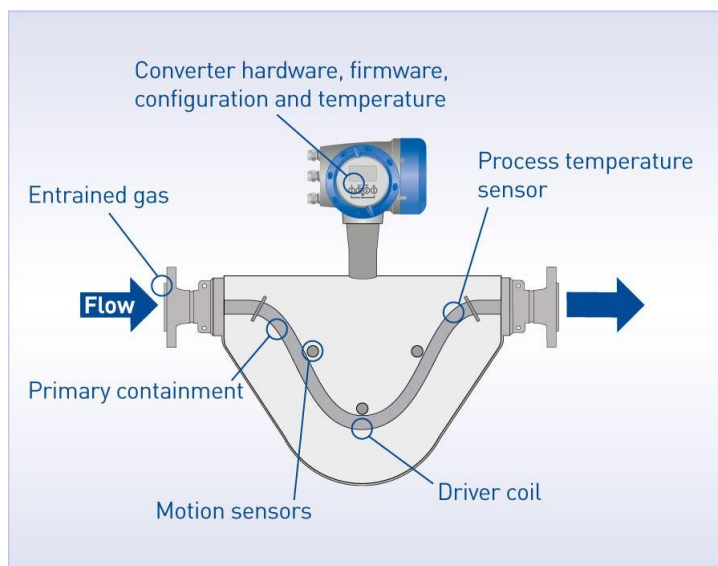
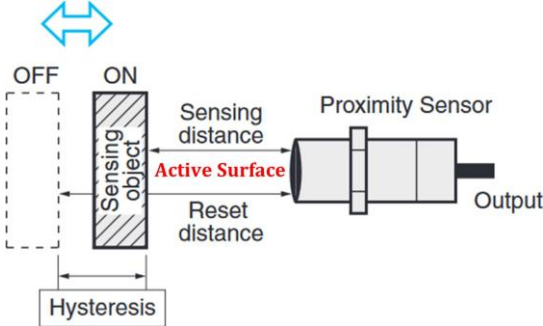


Figure 3.21: Coriolis mass flow meter showing vibrating tube principle.

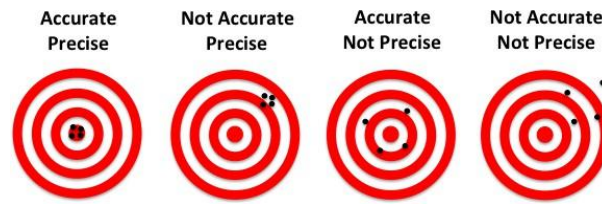
Humidity Sensors

- **Capacitive Humidity Sensors:** Detect changes in capacitance as dielectric absorbs moisture.
- **Resistive Humidity Sensors:** Detect changes in resistance as sensing material absorbs water.

3.1.4 Information Sheet-3: Sensor Related Terminologies

Terminology	Definition / Explanation
Hysteresis	The difference between the switch-on point and the switch-off point when a calibrating plate axially approaches and then moves away from the active surface of a proximity switch. Prevents signal flicker and ensures stable switching. 
Switching Distance (S_n)	The distance at which a metal plate approaching the sensor's surface causes signal switching. Determined within 10% accuracy at nominal voltage and temperature.
Light Switching	Output is activated when light strikes the receiver.
Dark Switching	Output is activated when light is blocked from reaching the receiver.
Repeatability (Reproducibility)	Ability of the sensor to provide consistent readings over time. For M18 inductive sensor, repetition accuracy is 0.2 mm.
Range (Sensing Distance)	Maximum operational distance for reliable detection (e.g., M18 inductive sensor: 0-8 mm).
Accuracy & Precision	Sensor accuracy is how close a measurement is to the true value, while precision is the consistency of repeated measurements. A sensor can be precise but inaccurate if it consistently produces the same wrong result, or it can be accurate but imprecise if its readings sometimes hit the true value but vary widely from each other.

3.1 Learning Outcome: Identify Types of Industrial Sensors



Linearity

Sensor linearity is the degree to which a sensor's output signal is proportional to the physical quantity it is measuring, meaning its output follows a straight line rather than a curve. For analog sensors, linearity is critical (e.g., inductive sensor analog output linear between 1-5 mm).

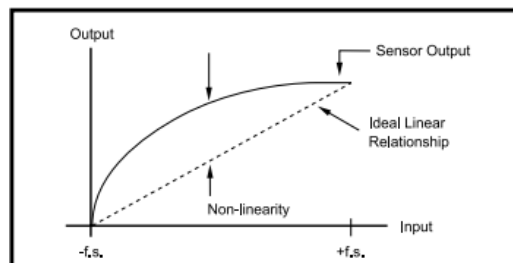


Figure 1: Hypothetical Sensor Non-Linearity

Response Time / Switching Frequency / Maximum switching rate. M18 inductive sensor: max. 300 Hz; reed switches: ≈ 0.2 ms.

Calibration / Verifying and adjusting output against a known reference per datasheet.

Sensitivity / Ratio of change in output to change in input. Often adjustable via potentiometer (e.g., in optical sensors).

Resolution / The smallest change a sensor can detect or the smallest step in its output value. Resolution is the smallest change detectable, while accuracy is how close a measurement is to the true value. Resolution refers to the smallest increment the sensor can display, while precision is the reproducibility of a measurement.

3.1.5 Information Sheet: Sensor Mounting Process (Flush, Non-flush)

Sensor Mounting Process (Flush, Non-flush) Proximity sensors, particularly inductive sensors, are designed for flush or non-flush mounting, which affects sensing range and field distribution.

Feature	Flush Sensors (Shielded)	Non-Flush Sensors (Unshielded)
Mounting Position	Active face is level with metal surface; sensor is surrounded by metal shielding.	Sensor protrudes from the surface; requires clear space around sensing face.

Module-3: Analyze Industrial Sensors

Sensing Range	Shorter range due to shielding.	Longer range than flush-mounted sensors.
Sensing Field	Field concentrated in front of sensing face.	Field extends forward and sideways, allowing lateral detection.
Application	Ideal for confined spaces or flat mounting.	Used when greater detection distance is required.

Note: Example — the Festo M18 inductive proximity sensor is designed for non-flush mounting and is not suitable for flush installation in metal.

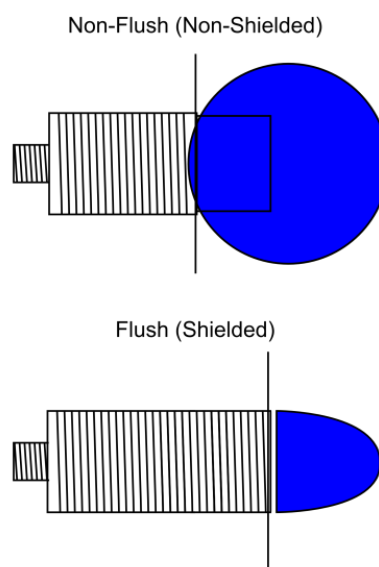


Figure 3.22: Flush vs. Non-flush Mounting of Inductive Proximity Sensors

3.1.6 Self-Check Questions

No.	Question
1	What is the fundamental purpose of an industrial sensor?
2	Name four types of mechanical switches used in industrial control systems that detect position or presence.
3	How do Inductive Sensors primarily function, and what type of object do they detect?
4	Contrast the operational difference between a contact type sensor and a non-contact type sensor, giving an example of each classification.
5	What is the difference between a through-beam optical sensor and a diffuse optical sensor?
6	Define Hysteresis in the context of a proximity switch.

3.1.7 Answer Key

No.	Answer / Guidance
1	Industrial sensors convert physical parameters into measurable electrical signals.
2	Push Button, Selector, Limit, and Foot Switch. (Also accepting Inductive, Capacitive, Optical, and Magneto-resistive Proximity Switches, since these are widely used industrial sensors.)
3	Inductive sensors use an oscillator circuit to create an electromagnetic field. When a conductive material enters the field, eddy currents attenuate the oscillation, generating an output signal corresponding to target presence. They detect metallic objects.
4	Contact type sensors require physical contact with the object or surface (e.g., Limit switches). Non-contact type sensors (Proximity Sensors) detect properties of objects without physical contact, relying on magnetic, electric, or optical fields (e.g., Inductive sensors, Capacitive sensors, Optical sensors).
5	In through-beam sensors, the transmitter and receiver are separate, and detection occurs when the light beam is interrupted. In diffuse sensors, the transmitter and receiver are combined, and detection occurs when light reflects diffusely off an object surface.
6	Hysteresis is the difference between the switch-on point and the switch-off point when a calibrating plate axially approaches and then moves away from the active surface of a proximity switch. Its function is to prevent signal flicker and ensure stable switching.

3.2 Learning Outcome: Perform Sensor Wiring using Correct Interface

Objective The learner must be able to select the appropriate sensor based on application requirements, identify and correctly understand the sensor's communication protocols, wire the sensor using the correct wiring interface according to specifications, and check the wiring for proper polarity, grounding, and shielding.

3.2.1 What is meant by Sensor Interface

The sensor interface refers to the physical and electrical connection point between a sensor (a field device) and the central control system, such as a Programmable Logic Controller (PLC). Its primary function is to convert non-electrical signals produced by the sensor into electrical signals that the PLC can recognize and to provide the appropriate voltage levels for communication.

Different Types of Wiring Interfaces

Wiring Interface	Function	Signal Type
PNP (Sourcing)	Switches the positive voltage to the load.	Digital
NPN (Sinking)	Switches the load to ground (0 V).	Digital
Push-Pull (Antivalent)	Switches both positive and negative potentials.	Digital
4-20 mA Loop	Current output proportional to measurement.	Analog (Current)
0-10 V	Voltage output proportional to measurement.	Analog (Voltage)
RS-485	Serial standard used for data communication.	Digital (Serial)
RTD	Resistance sensor connection method.	Analog (Resistance)

PNP (Sourcing)

In a sourcing configuration, the sensor provides the positive voltage potential (e.g., 24 V DC) to the load when actuated. Common in North America and safety-critical systems.

Terminal	Wire Color	Function
1	Brown (BN)	Positive Supply Voltage (+24 V DC)
3	Blue (BU)	Negative Supply Voltage (0 V DC)
4	Black (BK)	Switching Output (Signal)

3.1 Learning Outcome: Identify Types of Industrial Sensors

Connection to Load/PLC: The load (PLC input or relay) connects between BK (Pin 4) and 0 V. When ON, Pin 4 switches to +24 V.

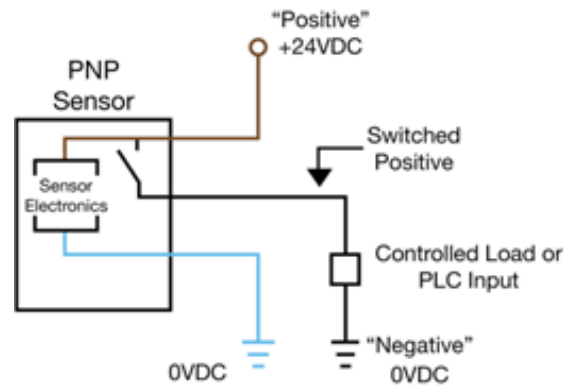


Figure 3.23: Typical PNP (Sourcing) sensor wiring diagram

NPN (Sinking)

The sensor provides a path to ground for current flow from the load. Widely used in Europe and Asia.

Terminal	Wire Color	Function
1	Brown (BN)	+24 V DC
3	Blue (BU)	0 V DC
4	Black (BK)	Signal Output

Connection to Load/PLC: The load connects between +24 V and BK. When ON, Pin 4 switches to 0 V.

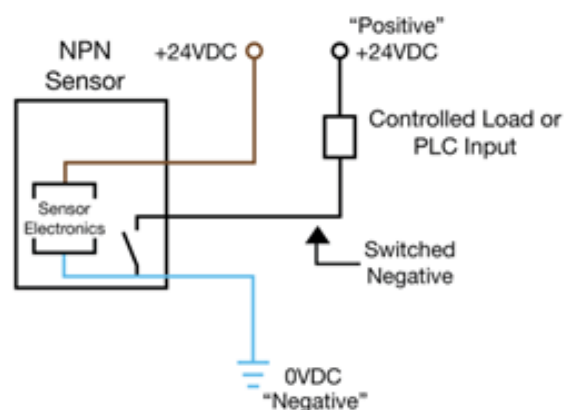


Figure 3.24: Typical NPN (Sinking) sensor wiring diagram

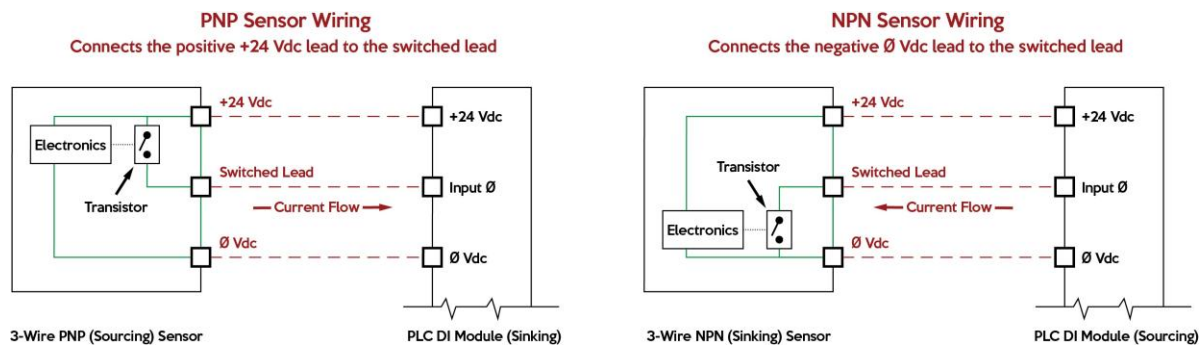


Figure 3.25: Comparison of PNP (Sourcing) vs. NPN (Sinking) Sensor Wiring

Important Note

Understanding **PNP and NPN wiring of sensors** is a very important step. Any mistake in wiring may either damage the sensor or the PLC input device.

Push-Pull (Antivalent)

Combines sourcing (PNP) and sinking (NPN) in one output stage. Example: Fibre-optic sensor units in Festo TP 1311.

Wire Color	Function	Description
Brown (BN)	Positive terminal	Power supply +24 V DC
Blue (BU)	Negative terminal	Power supply 0 V DC
Black (BK)	Q1 (NO) output	Normally Open output
White (WH)	Q2 (NC) output	Normally Closed output

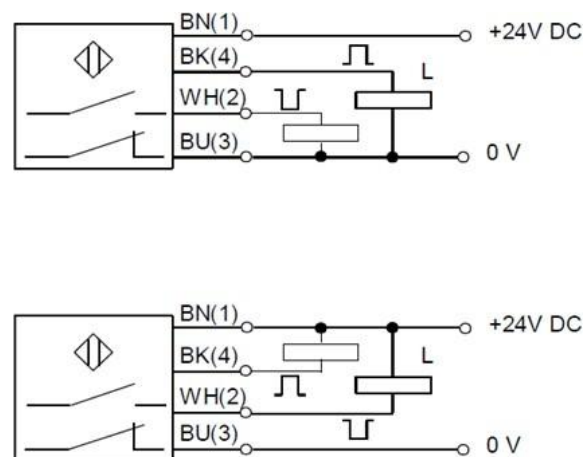


Figure 3.26: Push-pull (antivalent) sensor wiring with NO and NC outputs

4-20 mA Connection with PLC

The 4-20 mA current loop is an analog interface robust against noise and distance losses. Example: Festo inductive proximity sensor (M12 analog output, Order No. 548644).

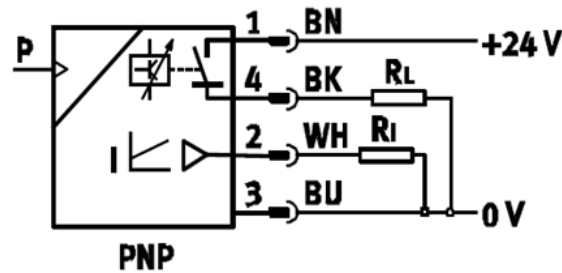


Figure 3.27: 4-20 mA current loop wiring and PLC analog input connection

0-10 V Sensor Wiring with PLC

Voltage-based analog output (0 – 10V), often used for short-distance sensing due to susceptibility to noise.

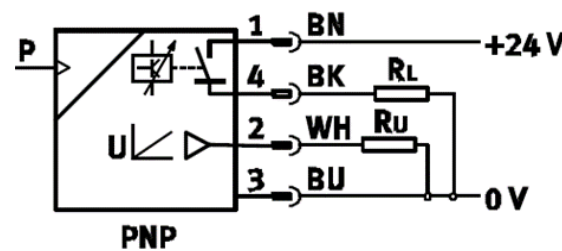
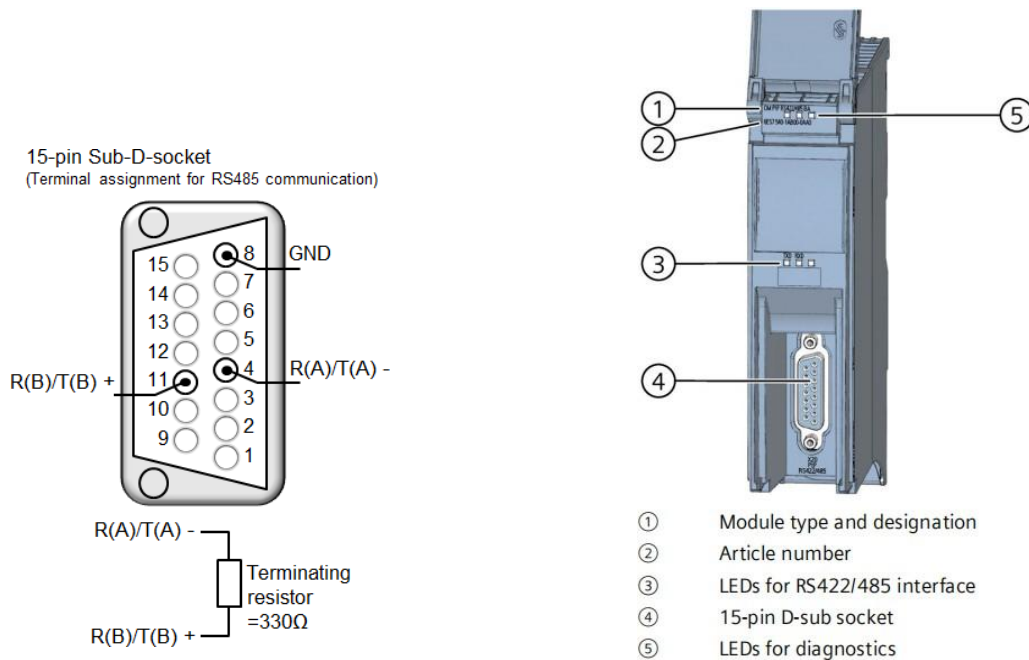


Figure 3.28: Typical wiring of a 0-10 V analog sensor with PLC analog input

3.2.2 RS-485 Sensor Wiring with PLC

RS-485 is a differential serial communication standard suitable for long-distance, noise-resistant transmission. Used for Modbus RTU and PROFIBUS field devices.



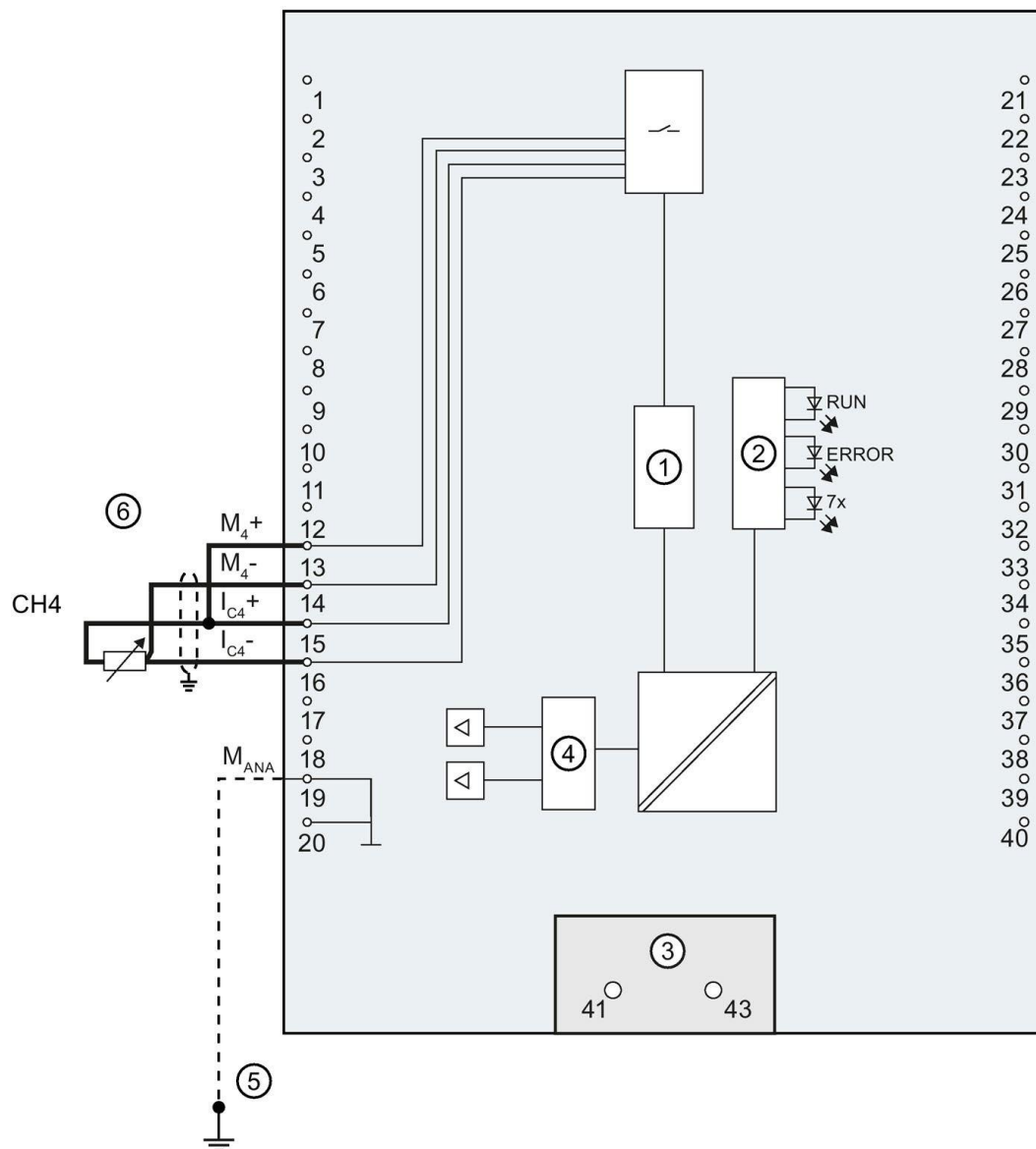
(a) RS-485 two-wire differential connection for Modbus RTU sensors

(b) CM PTP configuration in TIA Portal

Figure 3.29: Modbus RTU wiring and CM PTP configuration overview

3.2.3 RTD (Resistance Temperature Detector) Sensor Wiring with PLC

RTDs measure temperature through resistance change. Typical wiring uses a 3-wire connection for compensation.



1. Analog-to-digital converter (ADC)
2. LED interface
3. Infeed element (for shielding only)
4. Digital-to-analog converter (DAC)
5. Equipotential bonding cable (optional)
6. 3-wire connection

Figure 3.30: 3-wire RTD wiring connection to PLC analog module

3.2.4 Job Sheet - Detecting the Speed of a Gear

Section	Content
Job Title	Detecting the Speed of a Gear using a Magneto-resistive Sensor
Personal Protective Equipment (PPE)	Safety gloves, safety glasses, and insulated shoes must be worn during assembly and testing.
Materials Required	Magneto-resistive proximity sensor; Workpiece holder; Magnet 1; Magnet 2; 24 V DC power supply unit; Connection leads.
Tools and Equipment	Positioning slide; Indicator unit and electrical distributor; Ruler; Digital multimeter (optional); Mounting hardware.
Procedure	1. Preparation and Setup - Mount the magneto-resistive sensor securely on the positioning slide. - Connect the sensor to the indicator unit and the 24 V DC power supply. - Ensure that all wiring connections are correct and secure. - Check the supply voltage and polarity using a multimeter before energizing. 2. Operation and Measurement - Place Magnet 1 on the workpiece holder and move it longitudinally in 2 mm steps. - Observe and record the switch-on and switch-off points on the indicator. - Repeat the test using Magnet 2 and compare the switching behavior. 3. Analysis and Verification - Analyze the difference between the two magnets' switching curves. - Evaluate how the magnetic axis orientation affects switching performance. - Verify hysteresis between switch-on and switch-off points to ensure stability.
Notes	- Turn off power before assembly or sensor repositioning. - Avoid exposing the sensor to strong external magnetic fields. - Keep magnets clean and free of debris during testing. - Hysteresis helps prevent false triggering due to signal noise.

Continued on next page

Section	Content
Key Technical Points	- Magneto-resistive sensors detect magnetic field changes via resistance variation. - Switching distance and curve depend on magnet orientation and field strength. - Proper hysteresis ensures reliable detection and prevents chattering. - Measurement accuracy depends on stable power supply and precise positioning.

3.2.5 Self-Check Questions

No.	Question
1	What is the primary function of a sensor interface?
2	Name four distinct wiring interfaces used for industrial sensors.
3	How does a PNP (Sourcing) sensor configuration operate upon actuation?
4	Identify the standard wire colors and corresponding functions for the three terminals of a typical PNP sensor, relating to power supply and signal output.
5	How does the operational principle of an NPN (Sinking) sensor differ from that of a PNP sensor?
6	Why is the 4-20 mA current loop often favored over the 0-10 V voltage interface for analog sensor communication?
7	What dual functionality is combined within a Push-Pull (Antivalent) sensor output?
8	What critical risk is associated with incorrect wiring of PNP and NPN sensors?
9	Describe RS-485 and its primary advantage when used as a sensor wiring interface.
10	When wiring an RTD (Resistance Temperature Detector) sensor to a PLC analog module, why is a 3-wire connection commonly used?

3.2.6 Answer Key

No.	Answer / Guidance
1	The sensor interface is the physical and electrical connection point between a sensor and the central control system (like a PLC); its primary function is to convert non-electrical signals into electrical signals the PLC can recognize and to provide appropriate voltage levels.
2	PNP (Sourcing), NPN (Sinking), 4-20 mA Loop, 0-10 V. (Also acceptable: Push-Pull, RS-485, RTD).
3	In a PNP (Sourcing) configuration, the sensor provides the positive voltage potential (e.g., 24 V DC) to the load when actuated.
4	Brown (BN) is the Positive Supply Voltage (+24 V DC); Blue (BU) is the Negative Supply Voltage (0 V DC); and Black (BK) is the Switching Output (Signal).
5	An NPN sensor provides a path to ground (sinking) for current flow from the load when actuated, while a PNP sensor provides the positive voltage potential (sourcing) to the load.
6	The 4-20 mA current loop is robust against noise and distance losses, whereas the 0-10 V interface is susceptible to noise and typically used for short distances.
7	A Push-Pull (Antivalent) sensor output combines both sourcing (PNP) and sinking (NPN) functionality in one output stage.
8	Any mistake in wiring of PNP and NPN sensors may either damage the sensor or the PLC input device.
9	RS-485 is a differential serial communication standard that is suitable for long-distance, noise-resistant transmission, often used for Modbus RTU and PROFIBUS field devices.
10	The 3-wire connection method is utilized for resistance compensation when interfacing an RTD sensor with a PLC analog module.

3.3 Learning Outcome: 3.3 - Generate PLC Program for Data Acquisition

3.3.1 Information Sheet: NORM_X and SCALE_X Function in PLC

The NORM_X (Normalize) and SCALE_X (Scale) functions are standard Siemens TIA Portal functions for processing analog data. They convert raw analog values into meaningful engineering units.

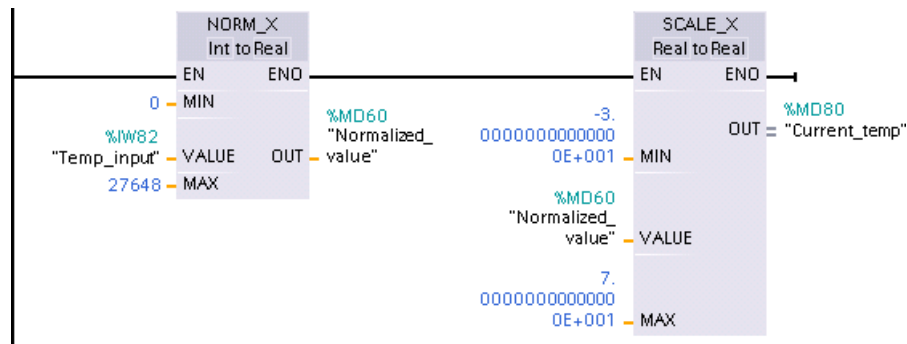


Figure 3.31: Use of NORM_X and SCALE_X functions for analog data processing in PLC

NORM_X (Normalize) Function

- **Purpose:** Normalizes raw integer input (VALUE) into 0.0-1.0 fractional range.
- **Formula:** $OUT = \frac{VALUE - MIN}{MAX - MIN}$
- **Typical Range:** MIN = 0, MAX = 27648 (for 0-10 V input).

SCALE_X (Scale) Function

- **Purpose:** Converts normalized (0.0-1.0) values into engineering units (e.g., 0.0-100.0).
- **Formula:** $OUT = VALUE \times (MAX - MIN) + MIN$
- **Application:** Scaling analog sensor values (temperature, flow, pressure).

Parameter	Data Type	Description
EN	BOOL	Enable input
MIN	(Varies)	Minimum limit of the range (raw or engineering)
VALUE	(Varies)	Input value to normalize/scale
MAX	(Varies)	Maximum limit of the range
ENO	BOOL	Enable output (0 = error)
OUT	REAL	Resulting normalized or scaled value

3.3.2 Job Sheet: Wiring and Programming Proximity Sensors (Push-Pull Interface)

Qualification	Programmable Logic Controller (PLC) Operation
Job Title	Wiring and Programming Optical, Inductive, and Capacitive Sensors (Push-Pull Interface)
Learning Element	Perform Sensor Wiring Using Correct Interface
PPE	Apron, Safety Shoe
Materials Required	PLC, Proximity Sensor (Optical, Inductive, or Capacitive), Relay (if required), SMPS, Magnetic Contactor, Control Valve
Tools and Equipment	Computer (for programming), Philips Screwdriver, Neon Tester

Performance Criteria

1. Appropriate proximity sensor is selected based on application.
2. Sensor interface (PNP, NPN, Push-Pull) is identified.
3. Sensor is wired correctly to the PLC digital input as per specifications.
4. PLC ladder program is written and simulated to read digital state.
5. Functionality verified through PLC input response.

Procedure

1. **Preparation:** Identify the sensor type and output configuration; ensure power is off before wiring.
2. **Task Execution:**
 - PNP Sensor: Brown → +24V, Blue → 0V, Black → PLC input.
 - NPN Sensor: Brown → +V, Blue → 0V, Black → PLC input.
 - Write ladder logic and simulate.
3. **Testing:** Download program, actuate sensor, and verify input bit response.

Notes

- Inductive sensors detect metallic objects; capacitive sensors detect both metallic and non-metallic; optical sensors use light.
- PNP = sourcing; NPN = sinking.

Key Technical Points

- Maintain correct polarity to prevent short circuits.
- Proximity sensors rely on magnetic, electrostatic, or optical fields.

3.3.3 Job Sheet: Calibrating Proximity Sensors

Qualification	Programmable Logic Controller (PLC) Operation
Job Title	Calibrating Optical, Inductive, and Capacitive Sensors
Learning Element	Perform Sensor Calibration According to Datasheet
PPE	Apron, Safety Shoe
Materials Required	Sensors, Calibrating Plate, Positioning Slide
Tools and Equipment	Screwdriver, Multimeter, PLC Input or Indicator Unit

Performance Criteria

1. Sensor is powered and connected correctly.
2. Switching distance determined using standard target.
3. Sensitivity adjusted per application.
4. Hysteresis measured and documented.

Procedure

1. Mount and connect sensor to supply and indicator.
2. Move target to detect switch-on and switch-off points.
3. Adjust sensitivity using potentiometer.
4. Calculate hysteresis = $D_{on} - D_{off}$.

Notes

- Hysteresis prevents flicker.
- For optical sensors, align beam accurately.

Key Technical Points

- Capacitive detection depends on dielectric constant.
- Nominal switching distance (S_n) defined using standard plate.

3.3.4 Job Sheet: Perform 4–20 mA Loop Calibration

Qualification	Programmable Logic Controller (PLC) Operation
Job Title	Perform 4–20 mA Loop Calibration
Learning Element	Performing Sensor Calibration
PPE	Apron, Safety Shoe
Materials Required	Analog Transmitter, PLC Analog Module, Wiring, Calibration Resistor

Tools and Equipment Multimeter, Current Source, Computer

Performance Criteria

1. Loop current range understood.
2. Physical quantities mapped to current range.
3. Scaling configured in PLC.
4. Calibration verified.

Procedure

1. Simulate 4 mA and 20 mA conditions.
2. Adjust transmitter zero/span.
3. Configure PLC scaling (NORM_X, SCALE_X).
4. Verify mid-range output (12 mA).

Notes

- 4–20 mA signals resist noise and voltage drop.
- Regular calibration maintains accuracy.

Key Technical Points

- Scaling links sensor current to engineering values.
- Typical raw PLC range: 0–27648.

3.3.5 Job Sheet: Wiring and Programming 4–20 mA Sensor

Qualification	PLC Operation
Job Title	Wiring and Programming 4–20 mA Sensor
Learning Element	Generate PLC Program for Data Acquisition
PPE	Apron, Safety Shoe
Materials Required	PLC, Analog Module, 4–20 mA Sensor, SMPS, Lamp/Valve
Tools and Equipment	Computer (TIA Portal), Philips Screwdriver, Neon Tester

Procedure

1. Connect sensor to PLC analog input.
2. Create tags: Raw, Normalized, Scaled.
3. Program NORM_X and SCALE_X for scaling.
4. Simulate and verify conversion.

Notes

- Analog inputs convert voltage/current to digital data.
- Use proper channel addressing.

Key Technical Points

- 4–20 mA = continuous analog signal.
- Scaling ensures meaningful engineering values.

3.3.6 Job Sheet: Wiring and Programming Modbus Sensor

Qualification	PLC Operation
Job Title	Wiring and Programming Modbus Sensor
Learning Element	Perform Sensor Wiring; Control AC Drive using PLC

Procedure

1. Identify Modbus standard (RTU/TCP).
2. Wire RS485 (A, B lines) between PLC and sensor.
3. Configure communication parameters.
4. Program Modbus register read/write blocks.
5. Verify data exchange and scaling.

Notes

- Modbus is used for device-to-device communication.
- Scaling factors convert register data to engineering values.

Key Technical Points

- RS485 is half-duplex differential signaling.
- Modbus registers (e.g., 40016) control drive/sensor parameters.

3.3.7 Job Sheet: Wiring and Programming Potentiometer

Qualification	PLC Operation
Job Title	Wiring and Programming Potentiometer
Learning Element	Generate PLC Program for Data Acquisition

Procedure

1. Connect potentiometer terminals: VCC, GND, Output.

2. Configure analog input tag.
3. Use NORM_X and SCALE_X for conversion.
4. Simulate rotation and verify range.

Notes

- Potentiometers act as variable voltage dividers.
- Ensure correct analog input scaling (0–10 V).

Key Technical Points

- Potentiometer used as manual input device.
- Smooth linear response indicates correct scaling.

3.3.8 Job Sheet: Wiring and Programming RTD Sensor

Qualification	PLC Operation
Job Title	Wiring and Programming RTD Sensor
Learning Element	Interface Sensors with PLC; Generate PLC Program for Data Acquisition

Procedure

1. Configure PLC RTD module.
2. Perform 3-wire RTD connection.
3. Create and scale tags for temperature display.
4. Verify measured temperature vs. actual.

Notes

- 3-wire connection reduces resistance error.
- RTDs output resistance proportional to temperature.

Key Technical Points

- PT100 sensors have 100 Ω at 0°C.
- SCALE_X converts raw data into degrees.

3.3.9 Job Sheet: Wiring and Programming High-Speed Counter (Encoder)

Qualification

PLC Operation

Job Title

Wiring and Programming Encoder (High-Speed Counter)

Procedure

1. Identify encoder type and resolution.
2. Wire A/B channels to PLC HSC inputs.
3. Configure HSC parameters (mode, direction).
4. Program to process encoder pulses.
5. Test counts and direction.

Notes

- Encoders convert rotation to pulses.
- Quadrature detection provides direction.

Key Technical Points

- Encoder resolution measured in CPR.
- Use CTRL_HSC_EXT for pulse processing.

3.3.10 Job Sheet: Wiring and Programming VFD with PLC

Qualification

PLC Operation

Job Title

Wiring and Programming VFD with PLC

Procedure

1. Identify VFD parameters (e.g., acceleration, max frequency).
2. Wire PLC outputs to VFD control terminals.
3. Configure analog speed reference (0–10V or 4–20mA).
4. Write ladder logic to control Start/Stop.
5. Test motor control and verify speed.

Notes

- VFDs use PWM to control motor voltage/frequency.
- Communication possible via Modbus RTU/TCP.

Key Technical Points

- PLC can control VFD via digital or analog signals.
- Modbus register 40016 commonly used for frequency setpoint.

3.3.11 Self-Check Questions

No.	Question
1	What is the fundamental purpose of the NORM_X function in PLC programming when dealing with analog inputs?
2	Explain the two functions performed by NORM_X and SCALE_X blocks combined when converting a raw analog sensor value to an engineering unit?
3	If a 13-bit analog input module is used for a 0 – 10 V sensor, what raw integer count corresponds to 10 V?
4	Describe the standard input range for the NORM_X function and the output range for the SCALE_X function when converting a raw 0 – 27648 value to a percentage?
5	What physical parameters are typically measured using analog sensors that require scaling and data acquisition in a PLC?
6	What is the primary difference in the control logic configuration between a Push-Pull interface sensor and a standard PNP sensor when programming a PLC?
7	What communication standard is often used when wiring and programming sensors that require register read/write blocks, such as when controlling a VFD?
8	Why is a 3-wire connection commonly utilized when wiring an RTD sensor to a PLC module?
9	What function block is typically used in a PLC to process encoder pulses for applications involving high-speed counting?
10	When reading an analog input signal from a potentiometer in a PLC program, what two data types are essential for the variables Normalized and Scaled to support arithmetic operations?

3.3.12 Answer Key

No.	Answer / Guidance
1	The primary purpose of the NORM_X (Normalize) function is to convert the raw integer value received from the analog input module into a normalized fractional range, typically between 0.0 and 1.0.
2	NORM_X converts the raw integer input data to a normalized REAL value between 0.0 and 1.0. SCALE_X then converts this normalized REAL value into user-defined engineering units (the desired measurement range).
3	For an analog input module with 13-bit resolution, the maximum raw count corresponding to 10 V is 27648 (INT).
4	The standard input range for NORM_X is the raw sensor range, e.g., 0 to 27648. The output range for SCALE_X is the desired engineering unit range, e.g., 0.0 to 100.0 (% scale).
5	Analog sensors requiring scaling measure continuous physical quantities such as temperature, flow, pressure, or level.
6	A standard PNP sensor provides a single digital output (e.g., Normally Open), while a Push-Pull (Antivalent) sensor combines both sourcing (PNP) and sinking (NPN) outputs, often providing both a Normally Open (Q1) and a Normally Closed (Q2) output signal.
7	Modbus RTU or Modbus TCP protocols are commonly used for sensors and devices, such as a VFD, that require configuration via register read/write blocks.
8	The 3-wire RTD connection method is used to reduce resistance measurement errors by compensating for the lead wire resistance.
9	The CTRL_HSC (High-Speed Counter) instruction is used to process encoder pulses for applications requiring high-speed counting.
10	The variables used in the mathematical functions (NORM_X and SCALE_X) for analog scaling, such as Normalized and Scaled, must be declared as REAL to support floating-point arithmetic and precision.

3.4 Learning Outcome: Perform Data Analysis on Sensor Outputs

3.4.1 Information Sheet: Performing Data Analysis on Sensor Outputs via HMI

Data analysis of sensor outputs involves recording, logging, and processing signals (including filtering, scaling, and conversion) to accurately capture and verify readings. The Human-Machine Interface (HMI) is critical for this element, as it provides a graphical interface for operators to interact with, display real-time process data, and allow control and adjustment of the industrial process. HMI functions typically include displaying processes, communicating with setpoints, outputting alarms, and archiving (logging) process values and alarms.

Sensor Data Visualization Concepts

1. **Digital Signal Visualization:** Digital sensors (e.g., optical proximity sensors) have binary outputs (ON/OFF). Visualization on HMI relies on animated lamps or rectangles that change color/state based on Boolean PLC input tags.
2. **Analog Signal Visualization:** Analog signals (e.g., 0-10 V) are continuous and must be scaled from raw PLC integer values (e.g., 0-27648) into engineering units (REAL/FLOAT). Visualization is typically achieved via numerical IO fields, bar graphs, or trend curves.
3. **Data Logging and Archiving:** HMI systems like SIMATIC WinCC can log alarms and process values to record process history. Logged data are stored in persistent log files (e.g., CSV) within CPU memory or external storage.
4. **OPC Unified Architecture (OPC UA):** A platform-independent, vendor-neutral industrial protocol for data exchange between automation systems and SCADA. Comfort Panels can act as OPC UA clients or servers with encryption and digital signature support.

3.4.2 Job Sheet: Visualize Digital Sensor Output in HMI (Optical Diffuse Sensor)

Qualification	HMI Programming and Interfacing
Job Title	Visualize digital sensor output in HMI (Optical Diffuse Sensor)
Learning Element	Displaying Sensor Data on the HMI in Usable Format
PPE	Apron, Safety Shoes
Materials Required	PLC, HMI (e.g., SIMATIC KTP700 Basic), Optical Diffuse Sensor, Ethernet Cable
Tools and Equipment	Computer (TIA Portal/WinCC Basic), Phillips Screwdriver

Performance Criteria

Module-3: Analyze Industrial Sensors

1. Optical diffuse sensor identified as digital (binary) output.
2. Sensor wired to PLC Digital Input.
3. HMI tag linked to PLC Boolean tag.
4. HMI displays ON/OFF state using an animated element.

Procedure

1. **Preparation:** Connect PLC and HMI via Ethernet; wire optical sensor to PLC input (e.g., %I0.0).
2. **Task Execution:** Create a rectangle element in HMI screen → Add visibility animation → Link tag (e.g., Sensor_1) → Configure state (color = green for logic 1).
3. **Testing/Verification:** Download to HMI → Actuate sensor → Verify HMI animation matches ON/OFF state.

Notes

- Optical diffuse sensors detect reflected light from objects.
- Use clear color coding (green = active, red = fault).

Key Technical Points

- HMI-PLC communication via configured connection.
- Digital visualization depends on tag's Boolean value.

3.4.3 Job Sheet: Visualize Analog Sensor Output in HMI (0-10 V Analog Proximity Sensor)

Qualification	HMI Programming and Interfacing
Job Title	Visualize analog sensor output in HMI (0-10 V analog proximity sensor)
Learning Element	Processing Sensor Signals for Scaling, and Displaying Data on HMI
PPE	Apron, Safety Shoes
Materials Required	PLC (Analog Input Module), 0-10 V Sensor, HMI, Communication Cable
Tools and Equipment	Computer (TIA Portal/WinCC Basic), Phillips Screwdriver

Performance Criteria

1. Analog sensor wired to PLC Analog Input.
2. PLC scaling (NORM_X/SCALE_X) implemented to convert raw value.

3. Scaled value displayed on HMI via IO Field or Bar Graph.

Procedure

1. **Preparation:** Wire the analog sensor and configure PLC scaling from raw (0-27648) to engineering range (e.g., 0.0-10.0 m).
2. **Task Execution:** Create IO Field on HMI, link to tag Scaled_Value, adjust numeric format (e.g., s999.99). Optionally add a bar graph for visual range display.
3. **Testing/Verification:** Simulate varying voltages and verify displayed scaled values (e.g., 5 V → 50% scale).

Notes

- Analog input converts 0-10 V signal to digital word (0-27648).
- Scaled value can be reused for process control logic.

Key Technical Points

- Analog visualization uses numeric and graphical HMI elements.
- Accurate scaling is essential for linear representation.

3.4.4 Job Sheet: Data Logging in HMI TP700 Comfort

Qualification	HMI Programming and Interfacing
Job Title	Data logging in HMI TP700 Comfort
Learning Element	Recording and Logging Sensor Data for Accurate Capture; Archiving Process Values
PPE	Apron, Safety Shoes
Materials Required	SIMATIC HMI TP700, PLC, USB Storage (optional)
Tools and Equipment	Computer (TIA Portal/WinCC)

Performance Criteria

1. HMI data archiving functions are understood.
2. Alarm logs configured to record process values.
3. Logs verified for proper storage and retrieval.

Procedure

1. Configure TP700 in project and identify relevant PLC tags.
2. Create Alarm Log ("Tank_Level") linked to PLC trigger tags.
3. Configure Process Value Archive to store cyclic data (1 s or 100 ms).

4. Download and test by simulating process events.

Notes

- HMI stores historical process/alarm data in CSV format.
- Archived data supports traceability and diagnostics.

Key Technical Points

- Logging enables trend analysis and predictive maintenance.
- Use cyclic archiving for continuous monitoring.

3.4.5 Job Sheet: Web-Based Monitoring in HMI

Qualification	HMI Programming and Interfacing
Job Title	Web-Based Monitoring in HMI
Learning Element	Operation of HMI to Control Digital I/O; Remote Control and Diagnostics
PPE	Apron, Safety Shoes
Materials Required	SIMATIC HMI Panel, PC Client, Ethernet Network
Tools and Equipment	Computer (TIA Portal/WinCC), Web Browser

Procedure Web-Based Monitoring Setup for Siemens TP700 Comfort Panel and S7-1500 PLC Web-based monitoring for SIMATIC devices can be established using the built-in Web Server functionality of the S7-1500 CPU or the remote control services, such as Sm@rtServer, configured on the Comfort Panel (like the TP700) using WinCC Comfort/Advanced (V14 or higher) within the TIA Portal.

Enabling Web-Based Monitoring on the SIMATIC S7-1500 CPU (Web Server)

The S7-1500 CPU features an integrated Web server that provides access to process data using a standard Web browser.

1. **Configure Access Protection in TIA Portal:** Before utilizing the Web server, access rights must be configured in the CPU's device configuration using STEP 7. Users and privilege levels are set up here to protect the system against unauthorized access.
2. **Deactivate Restrictions (Optional):** Ensure that the PUT/GET communication is permitted if you intend to allow client-side access to CPU data that is not otherwise configured. This option is generally deactivated by default.
3. **Transfer Configuration:** Compile and download the updated hardware and software configuration, including IP address settings, to the S7-1500 controller.
4. **Access the Web Server:** Access the Web server by entering the CPU's IP address into the Web browser of your PC or mobile device.

5. **Use Standard Web Pages for Monitoring:** Authorized users can perform the following functions via the standard Web pages:

- View general information about the CPU on the Start page.
- Change the operating mode (STOP and RUN) of the CPU.
- Monitor and modify PLC tags, data block tags, and I/O values using the Variable Status page. This page allows input of a direct address (e.g., I0.0), a PLC tag name, or a tag from a specific program block.
- View the diagnostic buffer of the CPU on the Diagnostic Buffer page.
- View module information and update firmware on the Module Information page.

Enabling Web-Based Monitoring on the TP700 Comfort Panel (Sm@rtServer)

The TP700 Comfort Panel (which is part of the Comfort Panel series) supports remote control features, typically via Sm@rtServer, which can be accessed by a client application.

1. **Access Panel Configuration in TIA Portal:** Open the WinCC Comfort project for the TP700 Comfort Panel (or equivalent Comfort Panel configuration).
2. **Enable Sm@rtServer:** In the project tree, double-click to open the Runtime settings for the panel. Navigate to the Services section, and under Remote control, select the option to Start Sm@rtServer.
3. **Download Configuration:** Compile the panel configuration and download it to the TP700 Comfort Panel.
4. **Configure Panel Settings (Windows CE):** After downloading and starting the panel, quickly access the Start Center and select the Settings button before Runtime automatically starts.
5. **Set Web Server Security:** Navigate to the Web server settings (usually under Transfer, Network & Internet). In the Sm@rtServer menu item, assign passwords for Force Write Access under Security Settings (e.g., using "sce").
6. **Initiate Remote Access:** Use the Sm@rtClient tool (installed with TIA Portal software) on the remote PC. Enter the IP address of the Comfort Panel (e.g., "192.168.0.10" if configured as such).
7. **Connect and Monitor:** Click Connect and enter the previously set password. This provides the ability to monitor and operate the panel remotely and even change Windows CE settings.

3.4.6 Job Sheet: OPC-UA with HMI

Qualification	HMI Programming and Interfacing
Job Title	OPC-UA with HMI

Module-3: Analyze Industrial Sensors

Learning Element	Establishing Secure Communication Protocols; Data Analysis on Sensor Outputs
PPE	Apron, Safety Shoes
Materials Required	Comfort Panel HMI and S7-1500 PLC (networked), Ethernet Cable
Tools and Equip-ment	Computer (TIA Portal/WinCC Advanced or Comfort), USB Drive

Performance Criteria

1. OPC UA server configured with security policy (e.g., Basic128Rsa15).
2. Client device connected with correct discovery URL and matching policy.
3. Certificates exchanged and accepted for encrypted communication.
4. Client browses and accesses server tags.

Procedure: Before configuring OPC UA, ensure the following conditions are met:

1. The Comfort Panel (e.g., TP700) and the communicating partner are integrated into a PROFINET network.
2. The date and time are synchronized on all communicating devices, as certificates specify a validity period which is critical for secure connections.
3. The required software, such as WinCC Advanced V14 or higher, is available.

TP700 Comfort Panel as OPC UA Server

The configuration process for setting up a Comfort Panel (like the TP700) as an OPC UA Server is performed within the TIA Portal project environment.

1. **Enable OPC UA:** In the project tree, select the Comfort Panel and navigate to “Runtime settings” → “Services”. Under the “Read/write tags” section, check the box labeled *Operate as OPC-UA server*.
2. **Configure OPC Settings:** Under “Runtime settings,” open the “OPC settings” menu. Specify the port number; the application example uses the default address 4870.
3. **Set Security Policy:** In the “Security policy of the application” table, uncheck the “None” checkbox to enable encryption. In the second row, specify the encryption type, typically Basic128Rsa15 with “Sign and encrypt” mode.
4. **Create Tags:** Insert all necessary tags, which can include both internal HMI tags and tags connected to a PLC.
5. **Transfer and Start Runtime:** Transfer the configuration to the Comfort Panel. Start the Comfort Panel runtime.

6. **Handle Client Certificates (for Secure Connection):** When a client attempts to connect securely, its certificate is saved to the Comfort Panel's "rejected" file folder, for example: C:\flash\simatic\SystemRoot\SSL
7. **Trust the Client Certificate:** Stop the Comfort Panel runtime. Navigate to the "rejected" folder on the panel, cut the certificate, and paste it into the "certs" folder within the same directory structure to trust it.
8. **Restart Runtime:** Close the file system and start the Comfort Panel runtime again to complete the secure configuration.

TP700 Comfort Panel as OPC UA Client

The configuration for the Comfort Panel (like the TP700) acting as an OPC UA Client requires specifying the connection details for the external OPC UA Server.

1. **Create the OPC UA Connection:** In the project tree, open the "Connections" folder. Create a new connection and select OPC UA as the communication driver.
2. **Specify OPC Server Parameters:** Define the OPC server details, including the UA server discovery URL (e.g., `opc.tcp://[IP address of server]:[Port]`), the Security policy (e.g., Basic128Rsa15), and the Message security mode (e.g., Sign and encrypt). The security policy must match the server configuration.
3. **Add Tags (Online Browsing):**
 - (a) Temporarily check the "None" security policy checkbox in the *server's* OPC settings configuration and transfer it to the server device to allow online browsing without encryption.
 - (b) In the Comfort Panel's tag editor, insert a new tag. Access the dialog box to browse to the server's tags (e.g., by clicking the arrow next to the "server object" in the address column).
 - (c) Navigate the server object hierarchy (e.g., to the Tags folder) and double-click the desired tags to apply them to the HMI project.
 - (d) After adding tags, uncheck the "None" box on the server configuration and transfer the configuration back to re-enable security.
4. **Transfer Configuration:** Transfer the Comfort Panel configuration to the device.
5. **Handle Server Certificates (for Secure Connection):**
 - (a) Start the Comfort Panel runtime. The server transfers its certificate to the Comfort Panel's "rejected" folder (e.g., C:\flash\simatic\SystemRoot\OPC\PKI\CA\default\rejected) via the network connection.
 - (b) Stop the Comfort Panel runtime.
 - (c) Manually move the server certificate from the "rejected" folder to the "certs" folder (e.g., cut from rejected and paste into certs).
6. **Restart Runtime:** Close the file system and start the Comfort Panel runtime. A connection should establish if the certificates were correctly assigned.

Notes

- OPC UA enables secure exchange between HMI and PLC or between HMIs.
- Encryption and authentication ensure integrity and confidentiality.

Key Technical Points

- OPC UA uses encryption, digital signing, and authentication for secure data exchange.
- Synchronize time and ensure consistent security policy on both devices.

3.4.7 Self-Check Questions

No.	Question
1	What is the role of the Human-Machine Interface (HMI) in the data analysis of sensor outputs?
2	How are digital sensor outputs typically visualized on an HMI screen?
3	How does the visualization of analog signals differ from digital signals on an HMI?
4	Name two common HMI screen objects used to display analog sensor values.
5	What must be done to an analog sensor's raw PLC integer value (e.g., 0 – 27648) before it can be meaningfully visualized on an HMI?
6	What function in HMI systems allows for recording and storing process history and alarms?
7	What common file format is often used for storing process history and alarm data logged by an HMI?
8	What protocol is described as a platform-independent, vendor-neutral industrial communication standard used for data exchange with SCADA and supervisory applications, supported by Comfort Panels?
9	When visualizing a digital sensor output on an HMI, what type of visualization element is commonly used to reflect the Boolean state?
10	Why is the accuracy of scaling the analog sensor signal critical for HMI visualization?

3.4.8 Answer Key

No.	Answer / Guidance
1	The HMI provides a graphical interface for operators to interact with, display real-time process data, allow control and adjustment of the industrial process, and log/archive process values and alarms.
2	Digital sensors have binary outputs (ON/OFF), and their visualization relies on animated elements like lamps or rectangles that change color or state based on the Boolean PLC input tags.
3	Analog signals (e.g., 0 – 10 V) are continuous and must be scaled from raw PLC integer values (e.g., 0–27648) into engineering units (REAL/FLOAT) for visualization, typically using numeric fields, bar graphs, or trend curves.
4	Lamps/indicators, I/O fields, trend views, buttons, and gauges. (Accepting numeric I/O fields, bar graphs, or trend curves).
5	The raw integer value must be scaled from the raw sensor range into engineering units (e.g., NORM_X and SCALE_X functions).
6	Data Logging and Archiving functions, typically using systems like SIMATIC WinCC, are used to log alarms and process values to record process history.
7	Logged data are often stored in persistent log files using the CSV format.
8	OPC Unified Architecture (OPC UA) is the platform-independent, vendor-neutral protocol for data exchange with SCADA and supervisory applications, supported by Comfort Panels.
9	An animated lamp or rectangle that changes color/state based on the Boolean PLC tag is commonly used.
10	Accurate scaling is essential for ensuring that the visualization (numeric field, bar graph, etc.) provides a linear and correct representation of the physical quantity being measured.

Module-4: Perform Electro-pneumatic Sequence Control

Module Description

This unit covers the knowledge, skills and attitudes required to perform electro pneumatic sequence control. It specifically includes the tasks of understanding robot electrical and pneumatic diagrams, designing and simulate electropneumatic circuit, carry out electropneumatic circuit connection, generating PLC program for sequence control and operating modular production system (MPS) stations.

Minimum Time: 55 hrs.

Learning Outcome

1. Understand Robot Electrical and Pneumatic Diagrams.
2. Design and Simulate Electropneumatic Circuit.
3. Carry out Electropneumatic Circuit Connection.
4. Generate PLC Program for Sequence Control.
5. Operate Modular Production System (MPS) Stations.

4.1 Learning Outcome: Understand Electronic, Pneumatic and Elctro-pneumatic Circuit Diagrams

Objective: To recognize and interpret the standards, symbols, and fundamental components (both electrical and pneumatic) necessary for understanding robot and electropneumatic circuit diagrams, including the basic components in robots and their control systems.

4.1.1 Information Sheet: Fundamentals of Pneumatics and Electro-Pneumatics

Fundamentals of Pneumatics

Pneumatics represents a branch of physics and engineering dedicated to the properties, motion, and behavior of air, deriving its name from the Greek word *Pneuma*, meaning “breath”. In an industrial context, pneumatic systems utilize compressed air to execute mechanical work.

Understanding Compressed Air Compressed air is regular atmospheric air that has been compressed using a compressor to reach a pressure exceeding atmospheric pressure (greater than 1 bar). This process reduces the air's volume while increasing its pressure. Standard pressure relationships used in documentation are:

$$1 \text{ atm} = 1.013 \text{ bar} = 14.6 \text{ psi} = 101.3 \text{ kPa} = 760 \text{ torr}$$

Advantages of Pneumatic Systems Pneumatic systems are widely adopted in automation due to several inherent advantages:

- **Availability:** Air is an infinitely available energy source.
- **Safety:** Compressed air presents no risk of explosion or fire, making it suitable for hazardous environments.
- **Cleanliness:** Unlubricated compressed air is clean, minimizing contamination risks.
- **Transport and Storage:** Compressed air can be easily transported over long distances and efficiently stored in reservoirs.
- **Speed and Overload Safety:** Actuators operate rapidly, and components are inherently overload-safe.
- **Maintenance:** Components generally require minimal maintenance and provide long service life.

Components Used in Electro-Pneumatics An electro-pneumatic system comprises several functional levels:

- **Energy Supply Components:** Compressors, reservoirs, and service units (FRL - Filter, Regulator, Lubricator) that prepare and distribute compressed air.
- **Signal Input Devices:** Push-buttons, electrical limit switches, and various proximity switches (magnetic reed, inductive, optical) that initiate signals.
- **Signal Processing / Control Elements:** Relays, timers, solenoid valves, and PLCs that interpret and process input signals.
- **Power Components:** Pneumatic cylinders (linear motion) and pneumatic motors (rotary motion) that perform mechanical work.

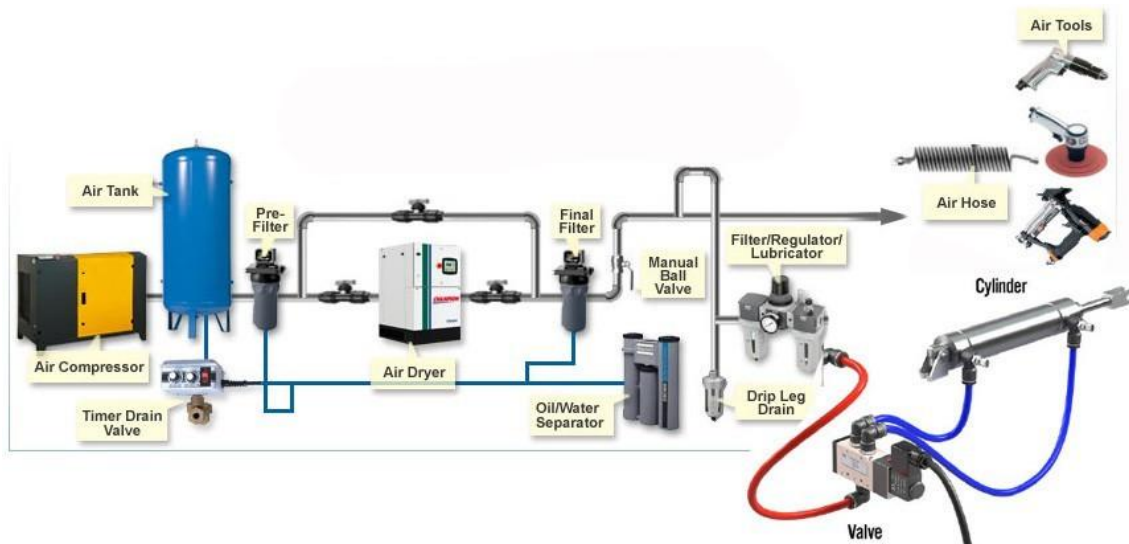


Figure 4.1: Typical components of an electro-pneumatic system

4.1.2 Information Sheet - Interpreting Pneumatic Valve Symbols

This information sheet provides guidance on how to interpret pneumatic valve symbols commonly found in industrial circuit diagrams and training materials (such as Festo or SMC). The correct understanding of valve symbols enables accurate circuit analysis, troubleshooting, and design of pneumatic and electro-pneumatic systems.

Standards Used to Create Circuit Diagrams

The following standards are applied when preparing circuit diagrams for pneumatic and electro-pneumatic systems:

- **ISO 1219-1:** Covers fluid power systems and components, specifying graphic symbols and circuit diagrams.
- **DIN EN 81346-2:** Concerns industrial systems, installations, equipment, and products, covering structuring principles and reference designations. This ensures that reference designations (like -MM1 for cylinders or -QM1 for valves) are compliant.

Valve Ports and Ways

A **port** refers to an opening in the valve body that connects to an air line or component, while a **way** indicates a possible flow path through the valve. The number format “5/2” means:

5 ports, 2 possible switching positions.

Direction control valves ports and positions (ways)

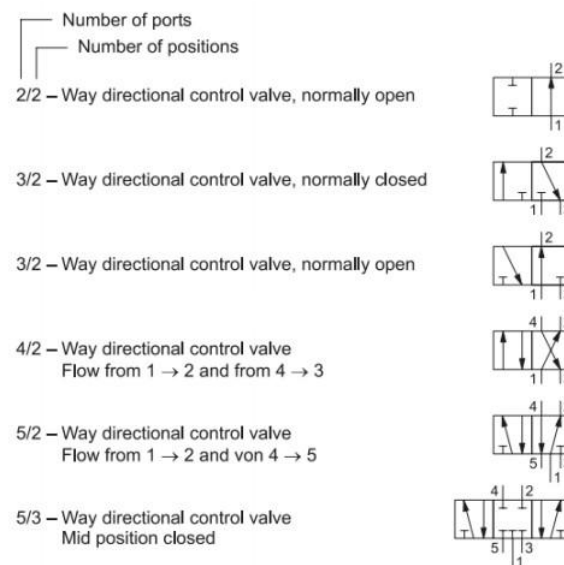


Figure 4.2: Identification of valve ports and ways

Flow Direction and Valve Positions

Each square in a valve symbol represents a distinct **position**. Inside each square, arrows show the **direction of air flow** between ports in that position.

- **Arrow:** Indicates the direction of compressed air flow.
- **T-shape:** Represents a blocked port (no flow).
- **Square orientation:** Each square corresponds to a stable or actuated position.

In a 5/2 valve, two adjacent squares depict two positions:

Left square - normal (unactuated) position; right square - actuated position.

Arrows change direction depending on which position is active.

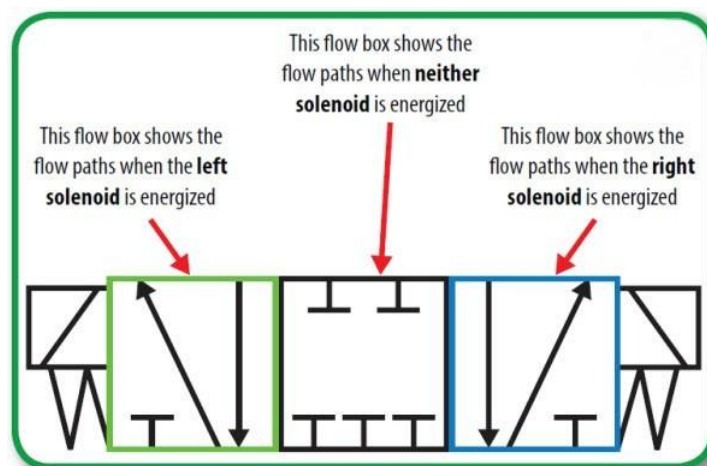


Figure 4.3: Identification of valve positions and flow directions

Valve Actuation and Return Methods

Each switching position in the symbol includes small graphical elements indicating how the valve is operated and reset. Actuation mechanisms may be:

- **Manual** (push button, lever, pedal, manual override)
- **Mechanical** (roller, plunger)
- **Pneumatic pilot signal**
- **Electrical solenoid**
- **Spring return**
- **Bistable** (no return spring; maintains position)

Symbol according ISO	Description	Actuation type
	Manually actuated	Manually or mechanically
	Manually actuated with pull and push button	
	Manually actuated with push button	
	Manually actuated with pull button	
	Lever actuated	
	Pedal actuated	
	Pedal with two operating directions	
	Actuated by plunger	
	Actuated by roller plunger	
	Actuated by roller lever	
	Actuated by spring	
	Actuated by pressure in general	Actuated via pressure
	Actuated by pressure	
	Pressure actuated indirectly with pneumatically actuated preamplifier	
	Parallel actuated here: by push button or pneumatic preamplifier	
	Electrically actuated in general	Electrically actuated
	Serial actuation here: electrically actuated preamplifier	
	Electrically actuated with electro-motor	

Figure 4.4: Common actuation and return symbols (ISO 1219-1)

The actuating symbol is placed on the side of the square corresponding to the direction of operation:

- Left side = rightward operation (actuating the right-hand position)
- Right side = leftward operation (actuating the left-hand position)

Interpreting Example

A common industrial valve is the **5/2-way single solenoid valve with spring return**. Interpretation of the symbol:

- Five ports: 1 (Pressure), 2 (Cylinder A), 4 (Cylinder B), 3 and 5 (Exhausts).
- Two positions: Normal (spring return) and Actuated (solenoid energized).
- Left side spring returns the valve to its normal state when power is removed.
- Right side solenoid coil energizes to switch flow direction (A+ / A-).

In the normal state, pressure connects to Port 2 (A+) while Port 4 (B-) exhausts. When the solenoid energizes, the valve shifts, connecting pressure to Port 4 (B+) and exhausting Port 2 (A-).

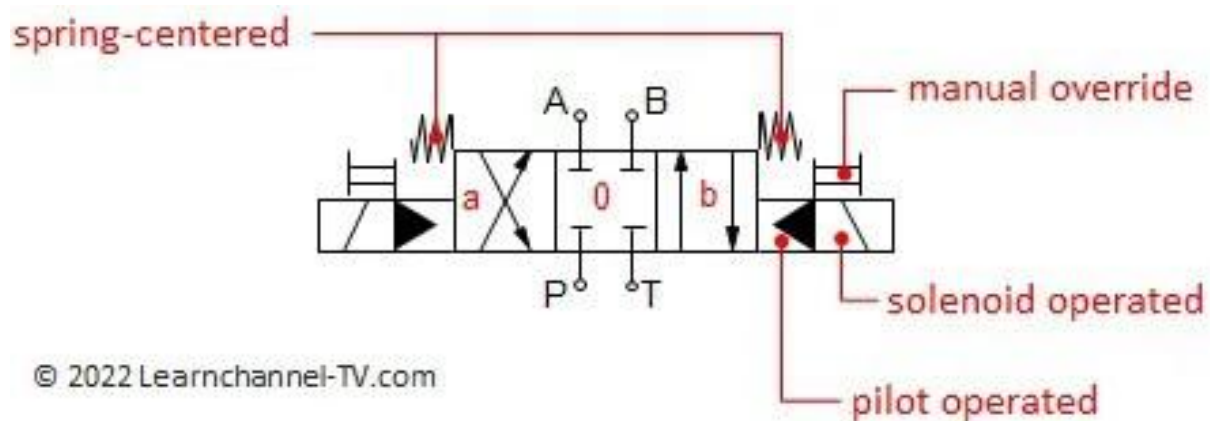


Figure 4.5: Example: 4/3 way, Center Closed, Pilot Operated, Spring Return Solenoid Valve with Manual Override

List of Common Valve Symbols

Component	Symbol	Component	Symbol
3/2-way-panel mounted with pushbutton actuator, normally closed		Single-acting cylinder	
3/2-way panel mounted valve with pushbutton actuator, normally open		Double-acting cylinder	
3/2-way panel mounted valve with selector switch		Pneumatic proximity sensor	
3/2-way roller-actuated valve, normally closed		One-way flow control valve	
3/2-way pneumatic valve, pneumatically actuated, one side		Quick-exhaust valve	
5/2-way panel mounted valve with selector switch		Pressure gauge	
OR Valve		Pressure regulator valve with pressure gauge	
Dual-pressure valve		Pneumatic timer, normally closed	

Figure 4.6: Symbols of common pneumatic components

4.1 Learning Outcome: Understand Electronic, Pneumatic and Electro-pneumatic Circuit Diagrams

5/2-way double solenoid valve	
Proximity sensor, electronic	
Pressure sensor	
Proximity sensor, optical	
Electric limit switch	

Figure 4.7: Symbols of common electro-pneumatic components

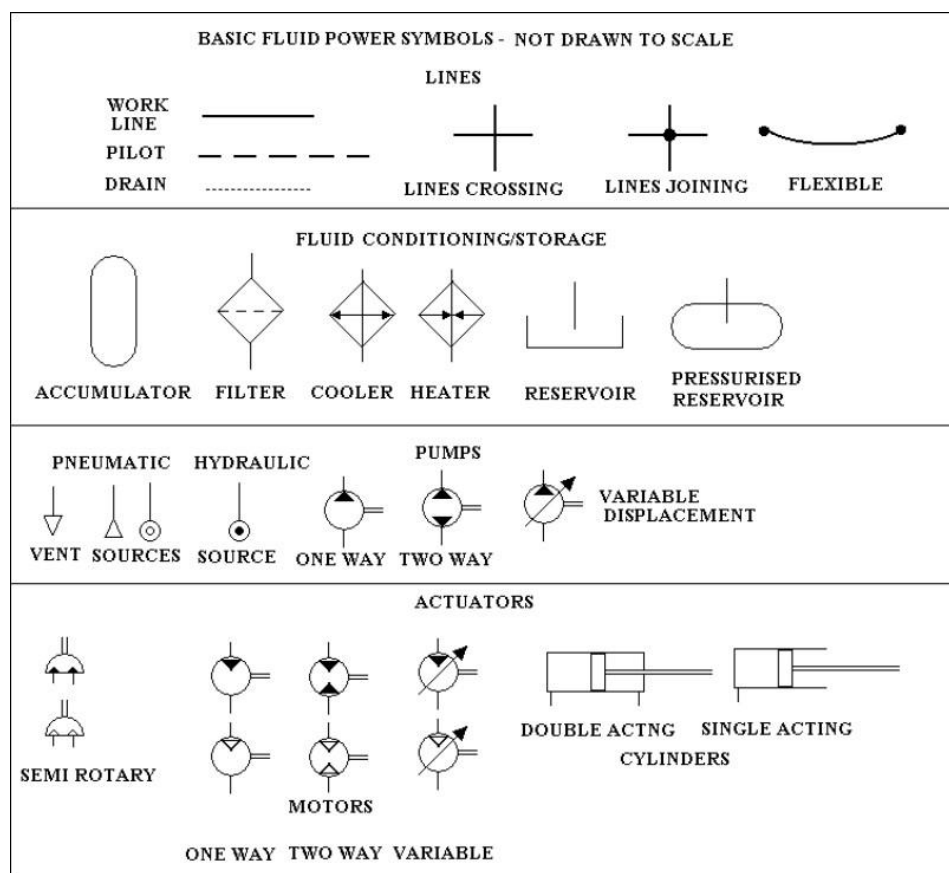


Figure 4.8: symbols of common pneumatic circuit elements

4.1.3 Information Sheet: Interpreting Circuit Symbols used in pneumatic and electro-pneumatic systems**Reference Designations as per DIN EN 81346-2**

Component Type	Example Designation / Code	Typical Function / Use in Pneumatics or MPS Automation
Cylinders / Actuators	-MM1, -MM2	Pneumatic or electric linear actuators; used for movement, clamping, or lifting operations.
Valves (General)	-QM1, -QM2	Directional control valves (2/2, 3/2, 5/2-way); solenoid or manual operation.
Shut-off / Isolation Valves	-KH1, -KH2	Manual shut-off or lockable isolation valves used for maintenance safety.
Pressure Control Valves	-RM1, -RZ1	Maintain or limit air pressure (pressure reducing, relief, or sequence valves).
Flow Control Valves	-RZ2, -RZ3	Regulate actuator speed via adjustable throttling (with check valve).
Logic Valves	-QM3, -QM4	Pneumatic logic elements (AND, OR, NOT gates) for sequencing and control.
Proximity Sensors	-BG1, -BG2	Detect piston position; can be reed, inductive, or capacitive type.
Photoelectric Sensors	-BF1, -BF2	Detect workpieces on conveyors using light beam (through-beam or reflective).
Pressure Sensors / Switches	-BP1, -BP2	Detect pneumatic pressure and provide electrical feedback for monitoring or safety.
Pushbuttons	-SF1, -SF2	Manual input devices for Start/Stop operations or manual override.
Selector Switches	-SJ1, -SJ2	Mode selection inputs (Auto/Manual/Reset).
End Stops / Limit Switches	-AZ1, -AZ2	Define end of travel; provide mechanical position feedback.
Mechanical Accessories	-XM1, -XM2	Brackets, mounts, or couplings used for mechanical support or installation.
Pressure Gauges	-PG1	Display pneumatic system or line pressure visually.

4.1 Learning Outcome: Understand Electronic, Pneumatic and Electro-pneumatic Circuit Diagrams

Motors	-M1, -M2	Drive conveyor belts, rotary tables, or pumps in electro-pneumatic systems.
Contactors / Relays	-K1, -K2	Electrical switching devices for actuator and solenoid control.
Fuses	-F1, -F2	Electrical protection elements preventing overcurrent damage.
Circuit Breakers	-Q1, -Q2	Manually resettable electrical protection for power circuits.
Signalling Devices	-H1, -H2	Visual or audible indicators (lamps, buzzers) showing system status.
Terminal Blocks	-X1, -X2	Electrical connection points for sensor or actuator wiring.
Solenoids / Coils	-Y1, -Y2	Electromagnetic coils that actuate pneumatic valves (ON/OFF control).
Accessories / Utility Components	-AZ1, -XM1, -PG1	Start-up valves, distributors, silencers, and quick connectors.

Pneumatic Port Designations

Designation	Function	Example Numbering
P	Input Pressure (Supply port)	1
A, B	Operation Port (Consumer port)	2, 4
R, S, T	Exhaust Port (Exhaust on Tank)	3, 5
12, 14	Pilot Ports	Used for pneumatic actuation signals

4.1.4 Information Sheet: Working Principle of Common Pneumatic System Components

Pneumatics is the branch of physics that deals with the properties, motions, and behavior of air. Pneumatic systems utilize compressed air, which is regular atmospheric air compressed to a pressure higher than the surrounding atmospheric pressure.

Pneumatic Actuators (Cylinders)

An actuator is an output device that converts supply energy (compressed air) into useful work.

- **Single-Acting Cylinder (SAC):** The piston rod moves to the front end position when compressed air is activated. A built-in return spring pushes the piston back when air is exhausted. SACs can perform work only in one direction.

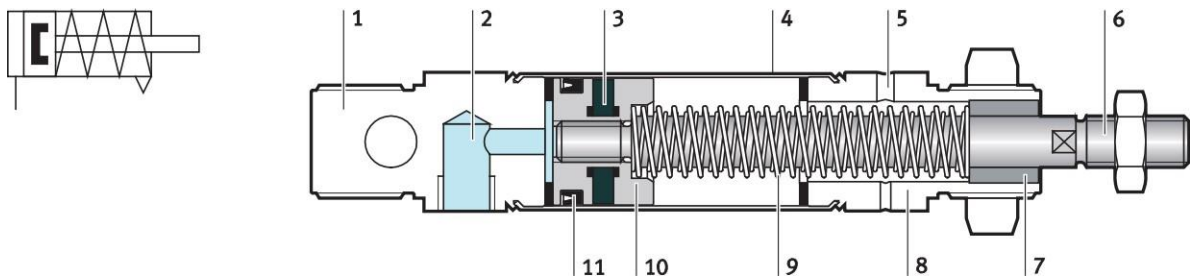


Figure 4.9: Single-acting cylinder - circuit symbol and cutaway view

- **Double-Acting Cylinder (DAC):** The motion is reversed by alternately pressurizing the respective ends with compressed air.
 - *Forward Stroke:* Compressed air flows into the piston chamber, building pressure and advancing the piston rod.
 - *Return Stroke:* Air flows into the piston rod chamber, forcing the piston back.

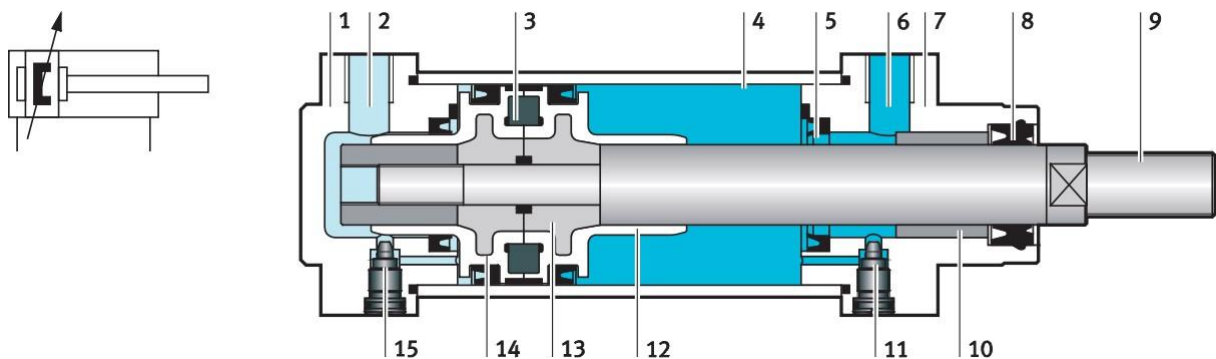


Figure 4.10: Double-acting cylinder operation

Directional Control Valves

Valves control the direction, pressure, and flow of compressed air. Directional Control Valves (DCVs) influence the path taken by an air stream. DCVs are characterized by the number of ports, control positions, and actuation method.

- **3/2-way Valve, Normally Closed (NC):** Allows flow when actuated and exhausts when released.

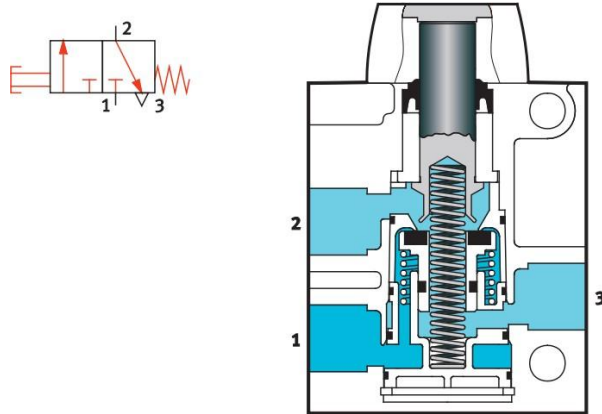


Figure 4.11: 3/2-way valve, manually operated, with spring return, normally closed - circuit symbol and cutaway view

- **3/2-way Valve, Normally Open (NO):** Air flows in normal position, stops when actuated.

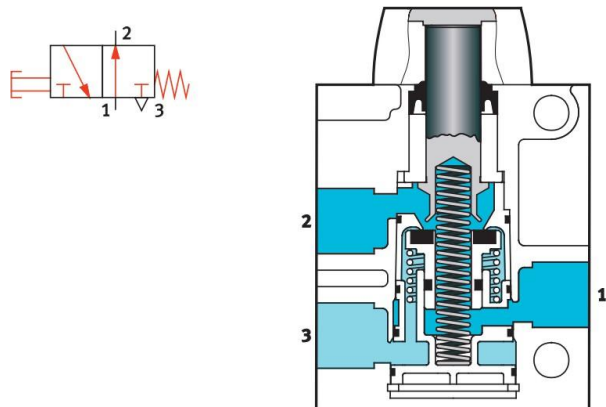


Figure 4.12: 3/2-way normally open valve operation

- **5/2-way Double Pilot Valve (Bistable):** Latching valve that maintains its position until a reverse pilot signal is received.

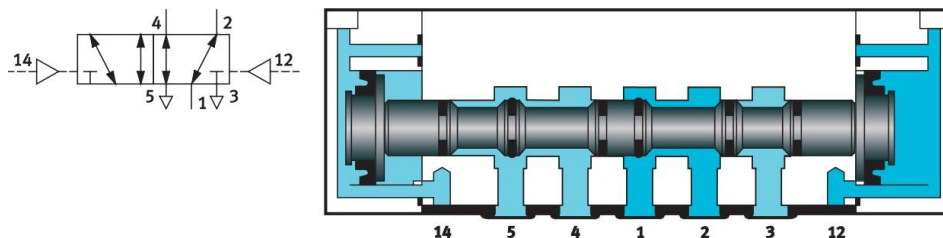


Figure 4.13: 5/2-way double pilot (bistable) valve

Flow Control Valves

- **One-Way Flow Control Valve:** Controls piston speed by allowing free flow in one direction and restricted flow in the other.

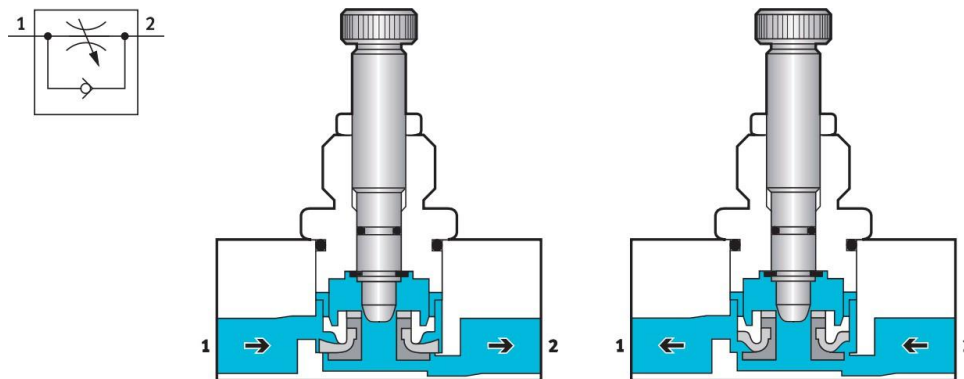


Figure 4.14: One-way flow control valve principle

- **Shuttle Valve (OR Gate):** Provides output when either input signal is active.

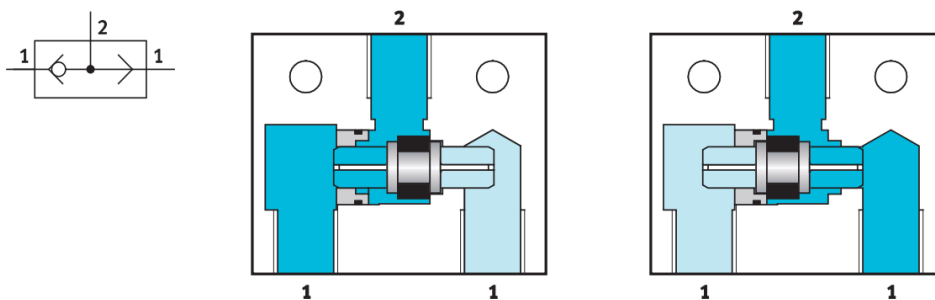


Figure 4.15: Shuttle valve (OR gate) symbol and function

- **Dual-Pressure Valve (AND Gate):** Allows output only when both input signals are active.

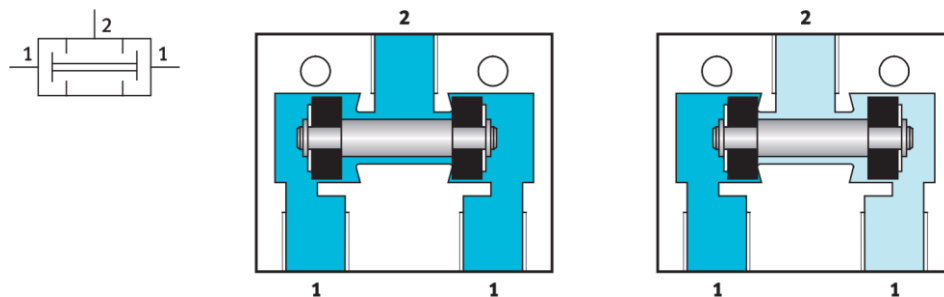


Figure 4.16: Dual-pressure valve (AND gate) symbol

- **Quick Exhaust Valve:** Increases piston speed by releasing exhaust air directly through a large orifice.

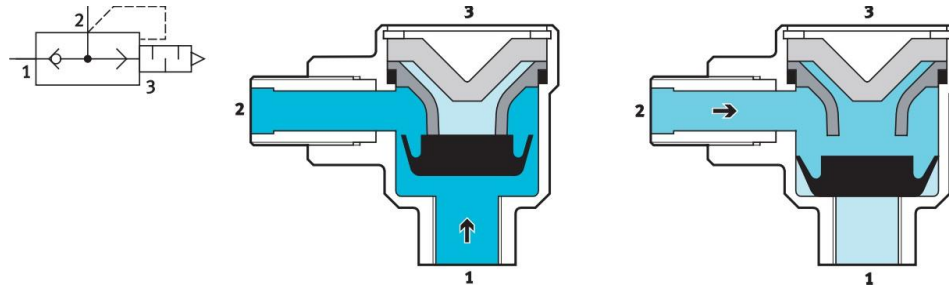


Figure 4.17: Quick exhaust valve air path

Pressure Control Valves

- **Pressure Sequence Valve:** Activates when pilot pressure exceeds a preset limit.

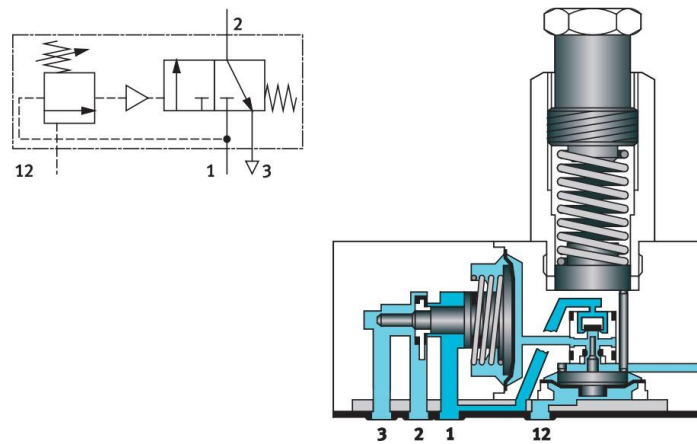


Figure 4.18: Pressure sequence valve structure

- **Pressure Regulator Valve:** Maintains constant downstream pressure despite supply fluctuations.

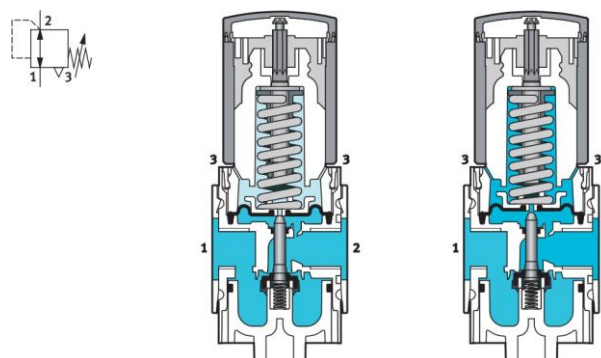


Figure 4.19: Pressure regulator valve cross-section

Pneumatic Timer (Time Delay Valve)

Passes air after a preset time delay.

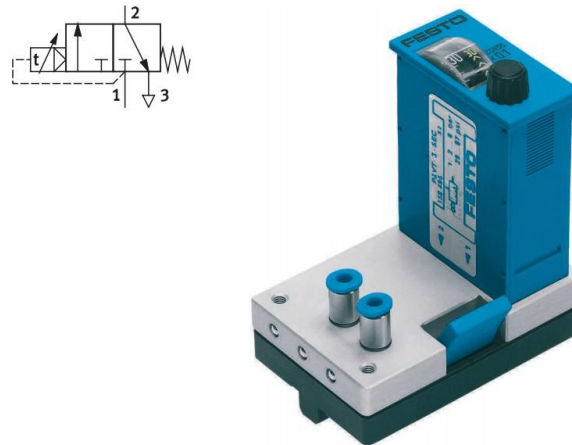


Figure 4.20: Time delay valve (pneumatic timer) schematic

Electro-Pneumatic Components (Solenoid Valves)

- **Solenoid Valve with Spring Return:** Returns to default position when current is off.

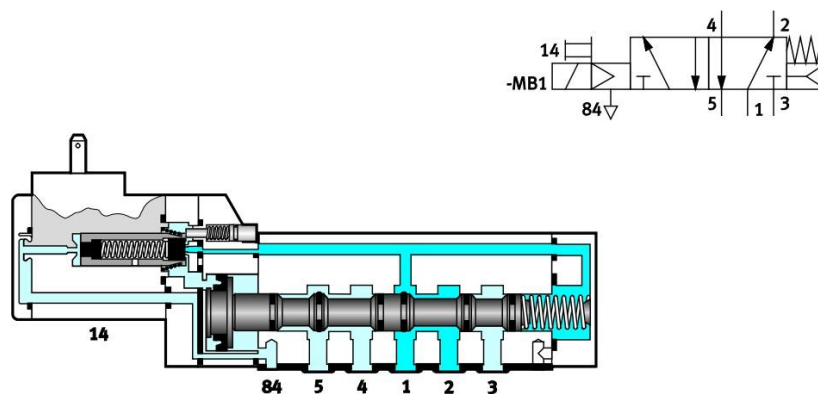


Figure 4.21: Solenoid valve with spring return

- **Double Solenoid Valve (Impulse Valve):** Retains position after brief electrical pulse.

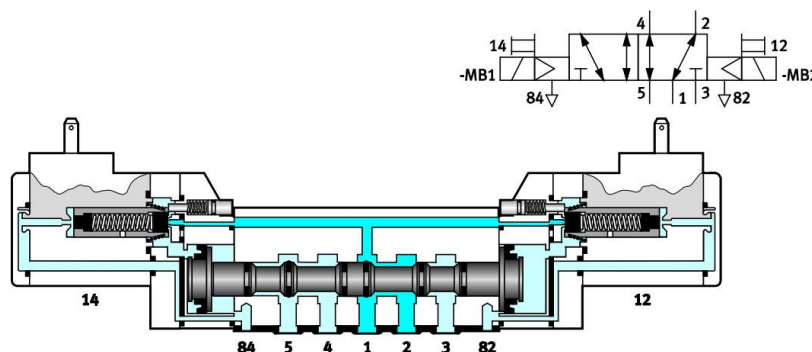


Figure 4.22: Double solenoid (impulse) valve diagram

Sensors and Switches

- **Pneumatic Limit Switch (Roller Lever Valve):** Contact-based detection of motion limits.

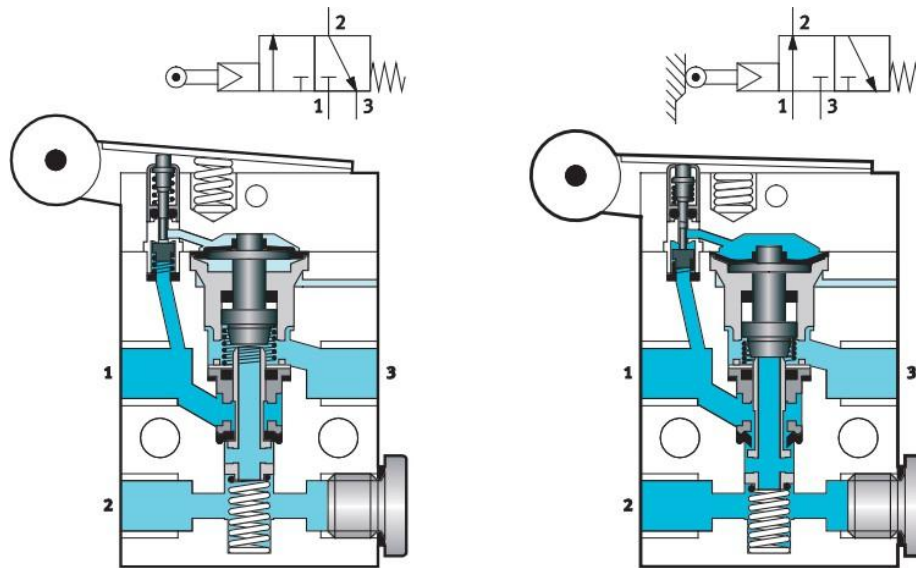


Figure 4.23: Pneumatic limit switch with roller actuator

- **Magnetic Proximity Switch (Reed Switch):** Non-contact magnetic sensor activated by piston magnet.
- **Pneumatic Proximity Sensor:** Detects piston position using a 3/2-way air signal.

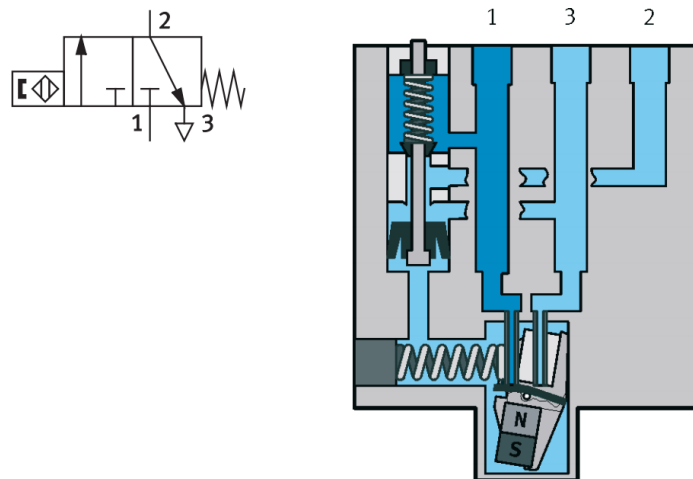


Figure 4.24: Pneumatic proximity sensor configuration

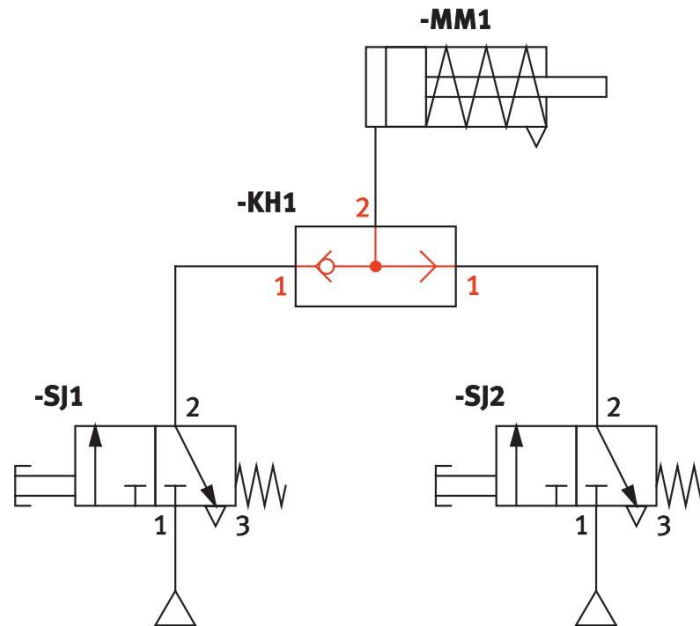
4.1.5 Information Sheet - Interpreting Function of Pneumatic Circuits**Pneumatic OR Gates**

Figure 4.25: Pneumatic OR gate operation using a shuttle valve

The **OR gate** allows a signal from either of two sources to activate a common output. In this circuit, the 3/2-way valves **-SJ1** and **-SJ2** act as input signals connected through a shuttle valve **-KH1**. The output from the shuttle valve drives the cylinder **-MM1**.

Initial Position: Both 3/2-way valves **-SJ1** and **-SJ2** are closed. The piston rod of cylinder **-MM1** remains retracted.

Step 1-2: When either pushbutton **-SJ1** or **-SJ2** is pressed, compressed air flows through the shuttle valve **-KH1** to the piston chamber of cylinder **-MM1**, causing it to advance.

Step 2-3: When both pushbuttons are released, the 3/2-way valves exhaust. The air supply to the shuttle valve is cut off, and the spring return in the cylinder **-MM1** pushes the piston back to its initial position.

This example demonstrates how pneumatic logic and latching circuits enable reliable sequential control without electrical components, widely used in training modules and automation systems.

Pneumatic AND Gate

The pneumatic **AND gate** allows an output signal only when both input signals are present simultaneously. It is implemented using a **dual-pressure valve (-KH1)**, which outputs compressed air only if both control lines are pressurized at the same time. This logic is essential in ensuring two conditions are met before an actuator performs a motion.

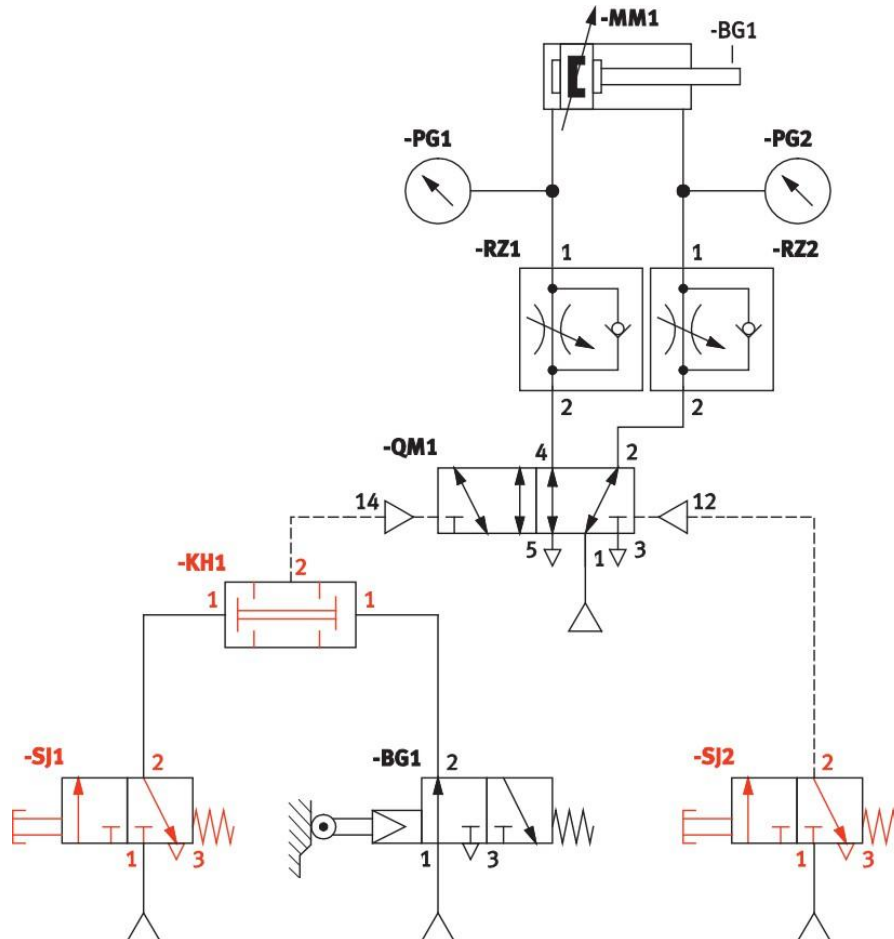


Figure 4.26: Pneumatic AND gate operation using a dual-pressure valve

Initial Position: The system is pressurized. The 3/2-way roller-actuated valve **-BG1** is actuated, exhausting the piston chamber of the double-acting cylinder **-MM1**, while its piston rod chamber remains pressurized. The piston rod stays at the **rear end position**.

Step 1-2: When the pushbutton of the 3/2-way valve **-SJ1** is pressed, compressed air is supplied to both input ports of the dual-pressure valve **-KH1**. Since both signals are present, the valve opens and delivers output pressure. This signal activates pilot port 14 of the 5/2-way double pilot valve **-QM1**, causing it to switch positions.

As a result, the piston rod chamber of the cylinder **-MM1** is exhausted, while the piston chamber becomes pressurized, driving the piston rod forward. When the roller-actuated valve **-BG1** returns to its normal position, the output signal from the dual-pressure valve is removed.

However, the cylinder's piston rod remains extended because of the **self-latching function** of the 5/2-way double pilot valve **-QM1**. This ensures the actuator maintains its position even after the input signals are released, completing the AND logic operation.

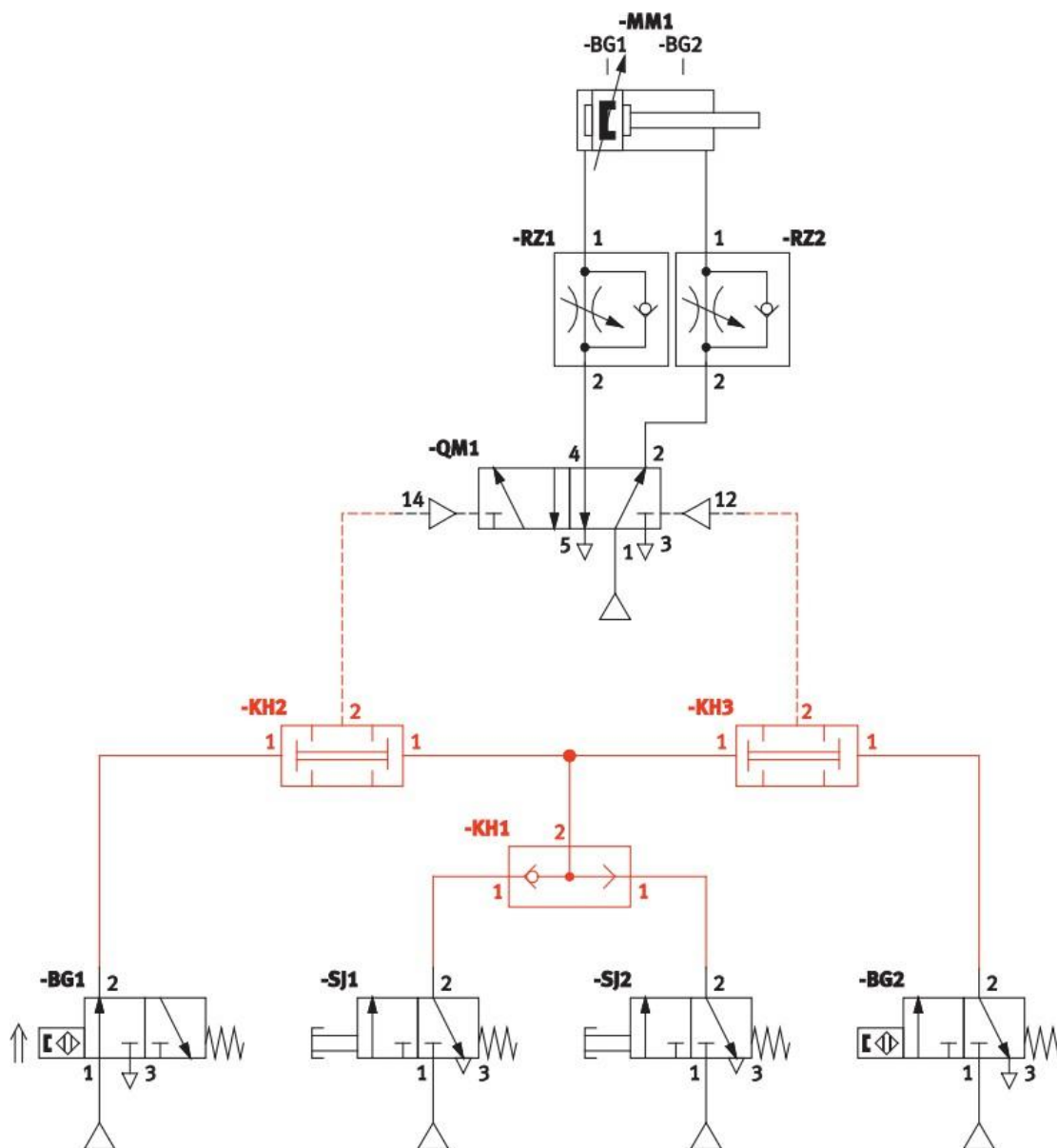


Figure 4.27: Pneumatic circuit diagram for combined boolean logic operations

Pneumatic Latching Circuit

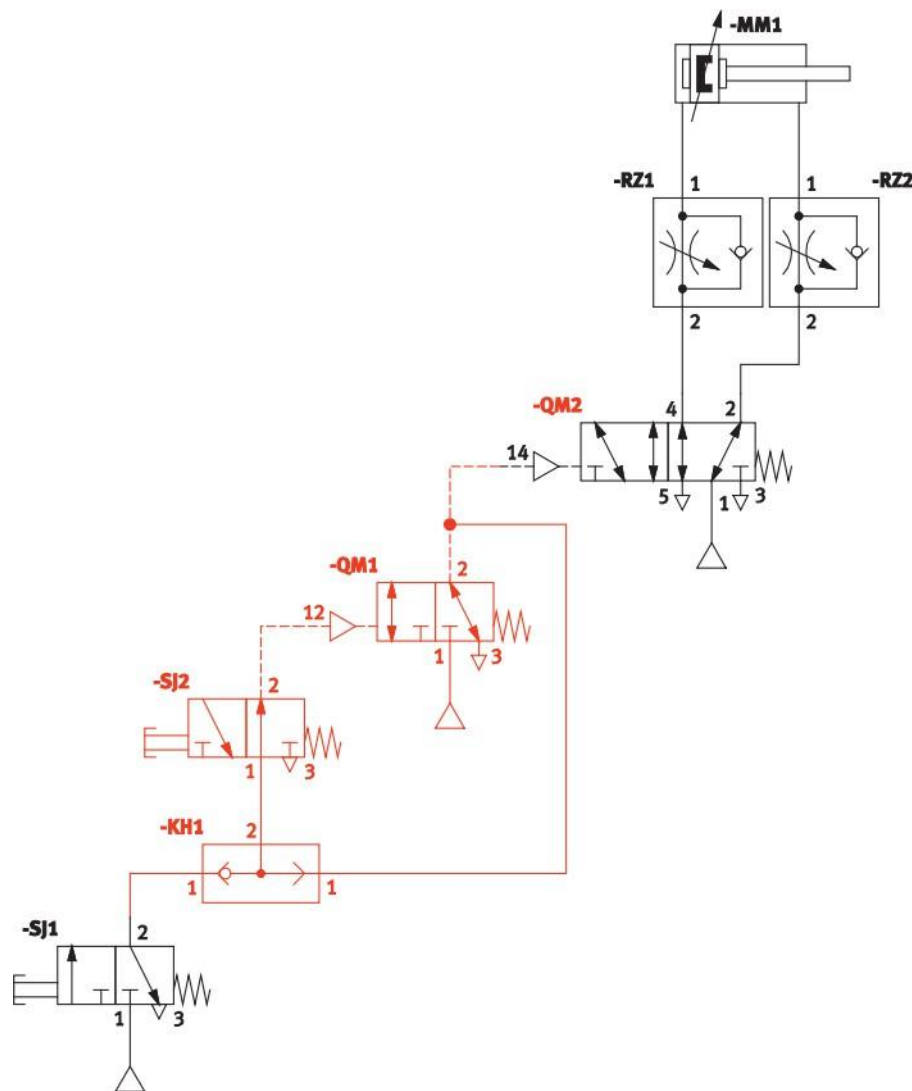


Figure 4.28: Pneumatic latching circuit using shuttle and 3/2-way valves

A latching circuit converts a brief signal into a continuous one. This type of self-holding circuit is used in valve groups such as **-SJ1**, **-SJ2**, **-KH1**, and **-QM1**. When the pushbutton **-SJ1** is pressed briefly, a continuous signal is sent to the output of valve **-QM1**.

When the pushbutton for the 3/2-way valve **-SJ1** is pressed, the shuttle valve **-KH1** opens to allow compressed air to pass. Since the 3/2-way valve **-SJ2** is normally open, the compressed air actuates the control port 12 of valve **-QM1**. This in turn actuates control port 14 on the 5/2-way valve **-QM2**. The directional control valve **-QM2** switches, and the piston rod of cylinder **-MM1** advances. Simultaneously, the return line to the shuttle valve **-KH1** ensures that valve **-QM1** remains actuated.

When the 3/2-way valve **-SJ2** is actuated, the self-latching loop is broken. The signal at control port 12 of valve **-QM1** disappears, returning the valve to its normal position. As a result, control input 14 of the 5/2-way valve **-QM2** and the return line to the shuttle valve exhaust. Valve **-QM2** switches back to its initial position, and the piston rod retracts.

If both pushbuttons **-SJ1** and **-SJ2** are pressed simultaneously, no signal appears at the output because the shuttle valve prevents dual input overlap.

4.1.6 Information Sheet - Interpreting Function of Electro-pneumatic Circuits

Single Acting Cylinder Control using 3/2 Solenoid Valve

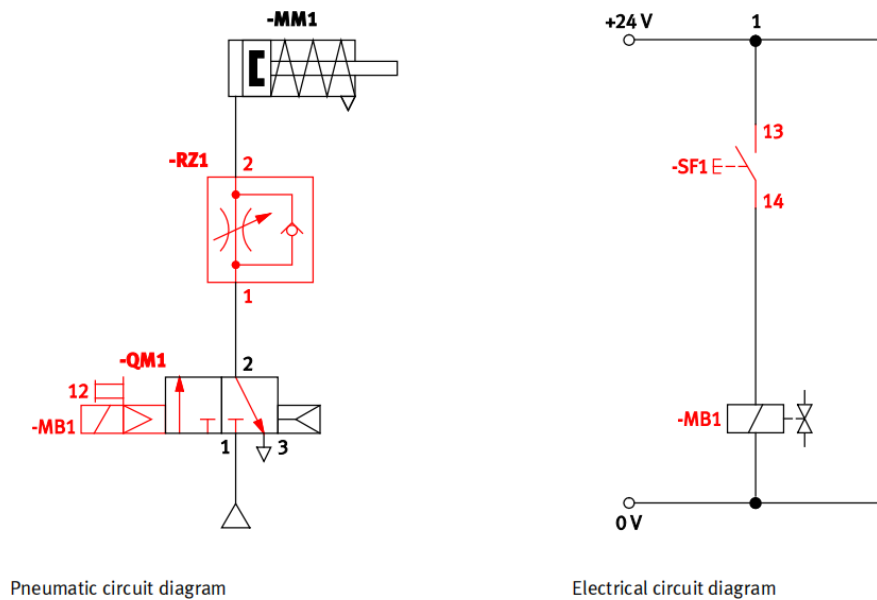


Figure 4.29: Operation of single-acting cylinder controlled by 3/2-way solenoid valve

Initial Position Cylinder **-MM1** is in the rear end position, and the piston rod is fully retracted.

Step 1-2 After activating the pushbutton **-SF1** (Normally Open contact), the electrical circuit for solenoid coil **-MB1** at the 3/2-way solenoid valve **-QM1** is closed. The solenoid valve **-QM1** is energized and switches to its actuated position, allowing compressed air to flow into the piston chamber of cylinder **-MM1**. As a result, the piston rod of cylinder **-MM1** advances.

Step 2-3 When pushbutton **-SF1** (NO contact) is released, the electrical circuit to solenoid coil **-MB1** opens. The 3/2-way solenoid valve **-QM1** returns to its initial position by means of its internal spring. The piston chamber of cylinder **-MM1** is exhausted, and the return spring inside the cylinder moves the piston rod back to its rear end position.

Operation of Double-Acting Cylinder using 5/2 Solenoid Valve with Relay Control

Initial Position Cylinder **-MM1** is in the rear end position and the piston rod is fully retracted.

Step 1-2 After activating the pushbutton **-SF1** (Normally Open contact), relay **-KF1** is energized. The changeover contact of relay **-KF1** (connected as an NO contact) in current path 2 closes, completing the electrical circuit for solenoid coil **-MB1** of the 5/2-way solenoid valve **-QM1**. As a result, the valve **-QM1** switches to its actuated position. Compressed air flows into the piston chamber of cylinder **-MM1**, while the piston rod chamber is exhausted. Consequently, the piston rod of cylinder **-MM1** advances.

Step 2-3 When the pushbutton **-SF1** (NO contact) is released, relay **-KF1** is de-energized. Its changeover contact (connected as an NO contact) in current path 2 opens, interrupting the electrical circuit of solenoid coil **-MB1**. The 5/2-way directional control valve **-QM1** then returns to its initial position via its spring return mechanism. The piston chamber of cylinder **-MM1** is exhausted and compressed air flows into the piston rod chamber, causing the piston rod to retract to the rear end position.

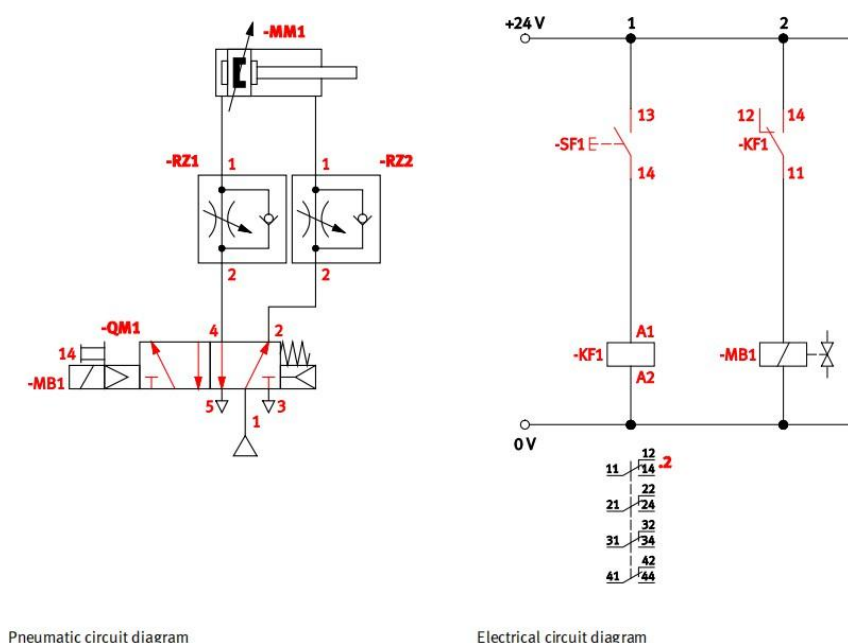


Figure 4.30: Operation of double-acting cylinder using 5/2-way solenoid valve with relay control circuit

Automatic Oscillating Motion of Double-Acting Cylinder

Initial Position Cylinder **-MM1** is in the rear end position. Magnetic proximity sensor **-BG1** is active (NO contact), energizing relay **-KF2**. Its NO contact in current path 7 closes, preparing the circuit for solenoid coil **-MB1**.

Step 1-2 When pushbutton **-SF1** (NO) is pressed, relay **-KF1** energizes and self- latches via its own NO contact. Another NO contact of **-KF1** in current path 7 closes, energizing solenoid coil **-MB1** of the 5/2-way double solenoid valve **-QM1**. The valve shifts, supplying compressed air to the piston chamber of cylinder **-MM1**, causing the piston rod to extend. As the rod leaves the rear position, sensor **-BG1** deactivates, opening the circuit to **-MB1**. The valve remains latched in its right-hand position.

Step 2-n At the front end, proximity sensor **-BG2** (NO) activates and energizes relay **-KF3**. Its NO contact in current path 8 closes, energizing solenoid coil **-MB2**, which reverses valve **-QM1**. Compressed air now flows into the piston rod chamber, retracting the cylinder. When **-BG2** deactivates, relay **-KF3** drops out, and the cycle restarts automatically as **-KF1** remains latched.

Step n-(n+1) Pressing stop pushbutton **-SF2** (NC) breaks the self-latching of relay **-KF1**, halting oscillation. The cylinder retracts to the rear end position and remains stationary.

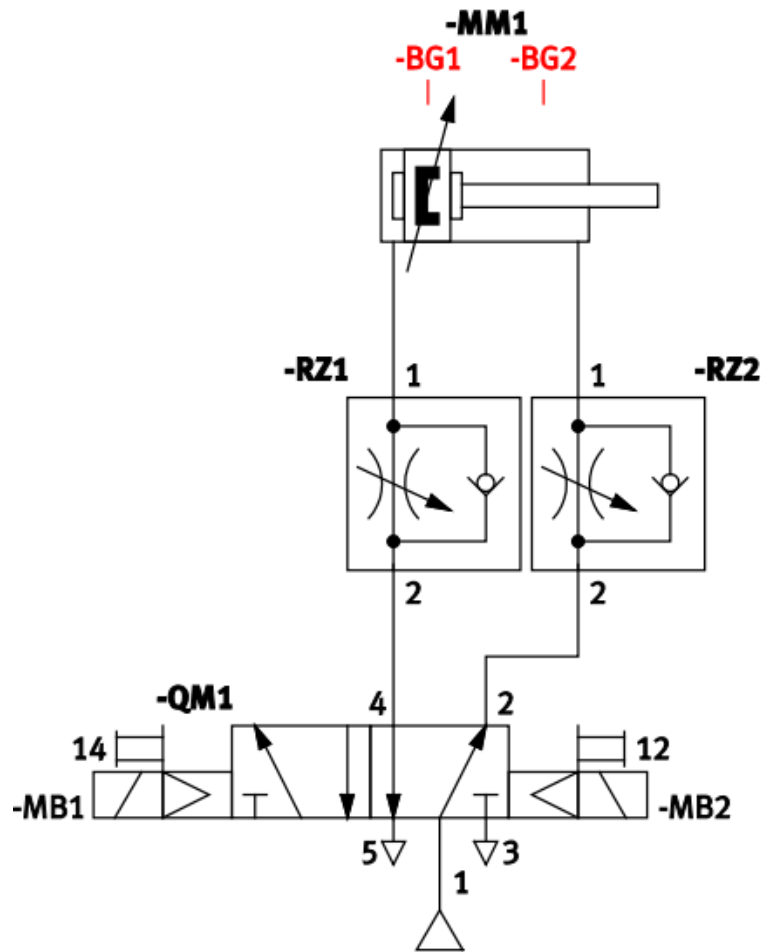


Figure 4.31: Pneumatic Circuit of Automatic oscillation of double-acting cylinder using relays and double solenoid valve

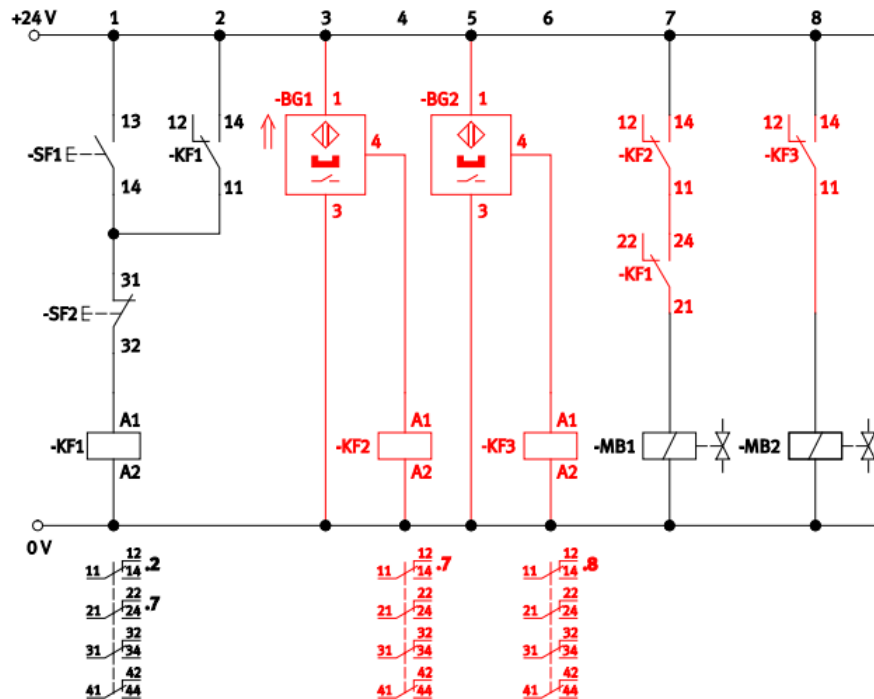


Figure 4.32: Pneumatic Circuit of Automatic oscillation of double-acting cylinder using relays and double solenoid valve

Sequential Control of Two Cylinders using Electro-Pneumatic System

Initial Position Cylinders **-MM1** and **-MM2** are in the rear end position with piston rods retracted.

Step 1-2 Magnetic proximity sensor **-BG1** is active, closing NO contact **-KF1** in current path 7. When pushbutton **-SF1** (NO) is pressed, solenoid coil **-MB1** of the 5/2- way double solenoid valve **-QM1** energizes, reversing the valve. Compressed air enters the piston chamber of cylinder **-MM1**, extending the piston rod. When **-SF1** is released, current path 7 de-energizes, switching off **-MB1**.

Step 2-3 At the front end of cylinder **-MM1**, proximity sensor **-BG2** (NO) activates and energizes relay **-KF2**. Its NO contact in current path 9 closes, energizing solenoid coil **-MB3** of 3/2-way valve **-QM2**. Valve **-QM2** switches, supplying compressed air to cylinder **-MM2**, which advances.

Step 3-4 When cylinder **-MM2** reaches the front end, optical sensor **-BG3** (NO) activates and energizes relay **-KF3**. Its NO contact in current path 8 closes, energizing solenoid coil **-MB2**, which switches valve **-QM1** back. Cylinder **-MM1** retracts, de-energizing relay **-KF2**, which opens current path 9. Solenoid coil **-MB3** de-energizes, and spring return switches valve **-QM2** back, retracting cylinder **-MM2**.

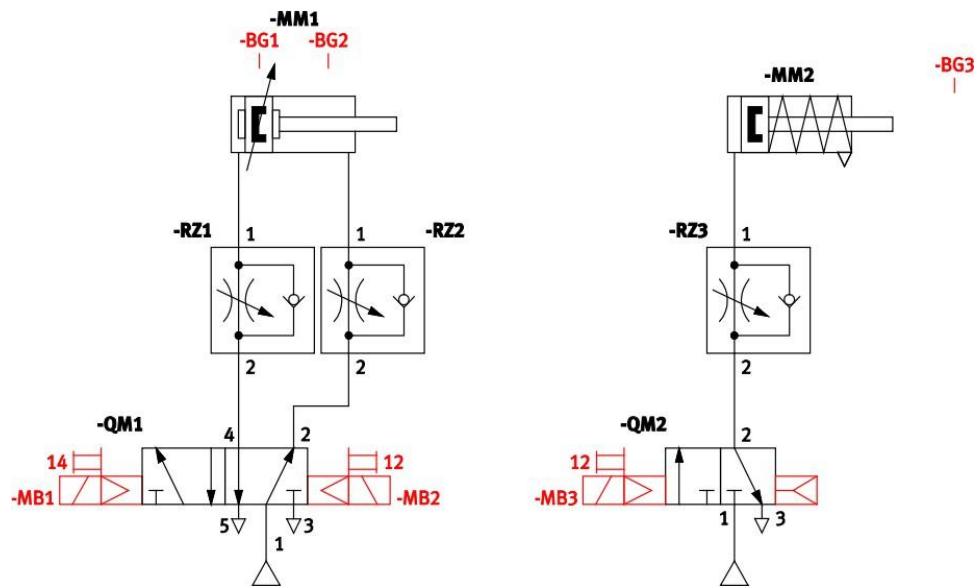


Figure 4.33: Sequential operation of two cylinders using relay logic and solenoid valves

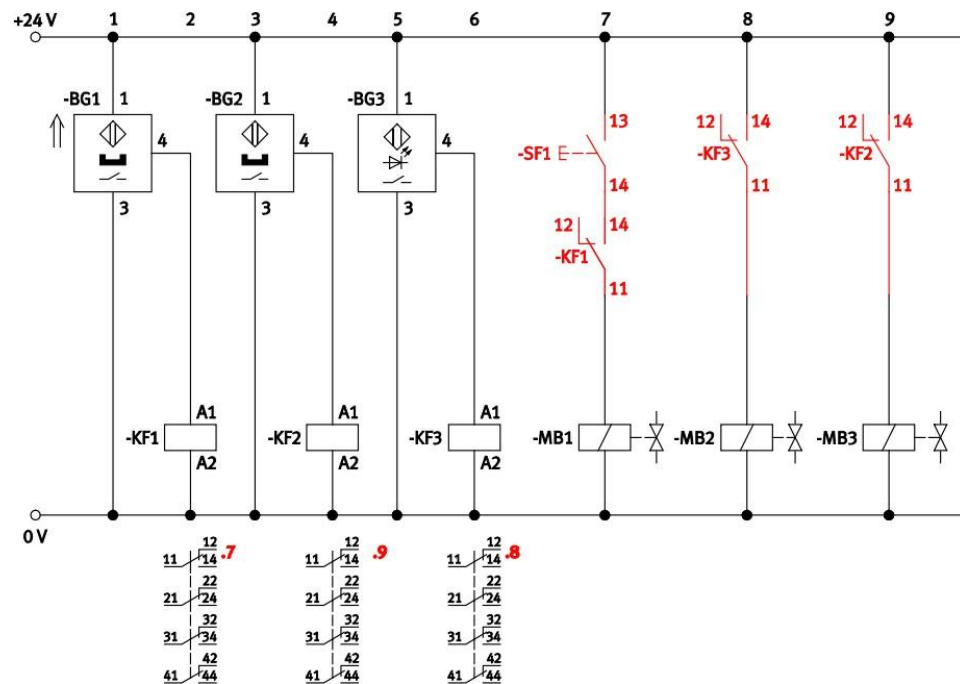


Figure 4.34: Sequential operation of two cylinders using relay logic and solenoid valves

4.1.7 Self-Check Questions

No.	Question
1	According to ISO 1219-1, what does the symbol "5/2" on a directional control valve signify?
2	What is the fundamental purpose of utilizing compressed air in pneumatic systems?
3	Name three advantages of using pneumatic systems in automation compared to other systems.
4	Contrast the operation of a single-acting cylinder (SAC) and a double-acting cylinder (DAC).
5	In an electro-pneumatic circuit, what is the function of a solenoid valve?
6	What distinguishes a bistable directional control valve from a single solenoid valve with spring return?
7	What two main standards are applied when preparing circuit diagrams for pneumatic and electro-pneumatic systems?
8	Describe the function of a shuttle valve (OR Gate) and a dual-pressure valve (AND Gate) in pneumatic logic.
9	In pneumatic valve symbols, what does an arrow inside a square represent, and what does a T-shape represent?
10	According to DIN EN 81346-2, what reference designation would typically be used for an electrical solenoid coil that actuates a pneumatic valve?

4.1.8 Answer Key

No.	Answer / Guidance
1	The symbol "5/2" means the valve has 5 ports and 2 possible switching positions.
2	Pneumatic systems use compressed air to execute mechanical work, converting supply energy into useful work.
3	Three advantages are: air is infinitely available, it presents no risk of explosion or fire (safety), and it allows for easy storage and transportation. (Also accepting speed of actuators, overload safety, and minimal maintenance).
4	A SAC performs work only in one direction using compressed air to advance the piston, while a built-in spring returns it. A DAC performs work in both directions by alternately pressurizing the respective ends with compressed air.
5	The solenoid valve acts as a signal processing or control element that converts an electrical signal from the control circuit (like a relay or PLC) into a pneumatic signal, influencing the path of the compressed air.
6	A single solenoid valve with spring return returns to its default position when the electrical current is removed. A bistable (double solenoid) valve maintains its position until a reverse electrical pulse is received.
7	The standards applied are ISO 1219-1 (graphic symbols and circuit diagrams) and DIN EN 81346-2 (reference designations).
8	A shuttle valve (OR Gate) provides an output when <i>either</i> input signal is active. A dual-pressure valve (AND Gate) allows output only when <i>both</i> input signals are present simultaneously.
9	An arrow indicates the direction of compressed air flow, and a T-shape represents a blocked port (no flow).
10	Electrical solenoids/coils that actuate pneumatic valves are typically designated using the DIN EN 81346-2 code starting with -Y (e.g., -Y1, -Y2 or -MB1).

4.2 Learning Outcome: Design and Simulate Electropneumatic Circuit

Objective: To recognize and interpret the standards, symbols, and fundamental components (both electrical and pneumatic) necessary for understanding robot and electropneumatic circuit diagrams, including the basic components in robots and their control systems.

Learning Activity: Study the specific industry standards (ISO 1219-1 and DIN EN 81346-2) applied in pneumatic and electrical circuit diagram representation; identify and describe the symbols for directional control valves, actuators, and logical pneumatic elements; recognize and define basic electrical control components (e.g., relays, limit switches, solenoids); and analyze how electrical and pneumatic systems are integrated to form control systems.

4.2.1 Information Sheet: Circuit Design Software

FluidSIM® is the standard simulation and CAD tool used for pneumatic and electropneumatic circuit design. It enables:

- Drawing DIN-compliant circuit diagrams.
- Realistic dynamic simulation of physical air behavior.
- Logic-based control simulation and PLC interface via EasyPort USB.

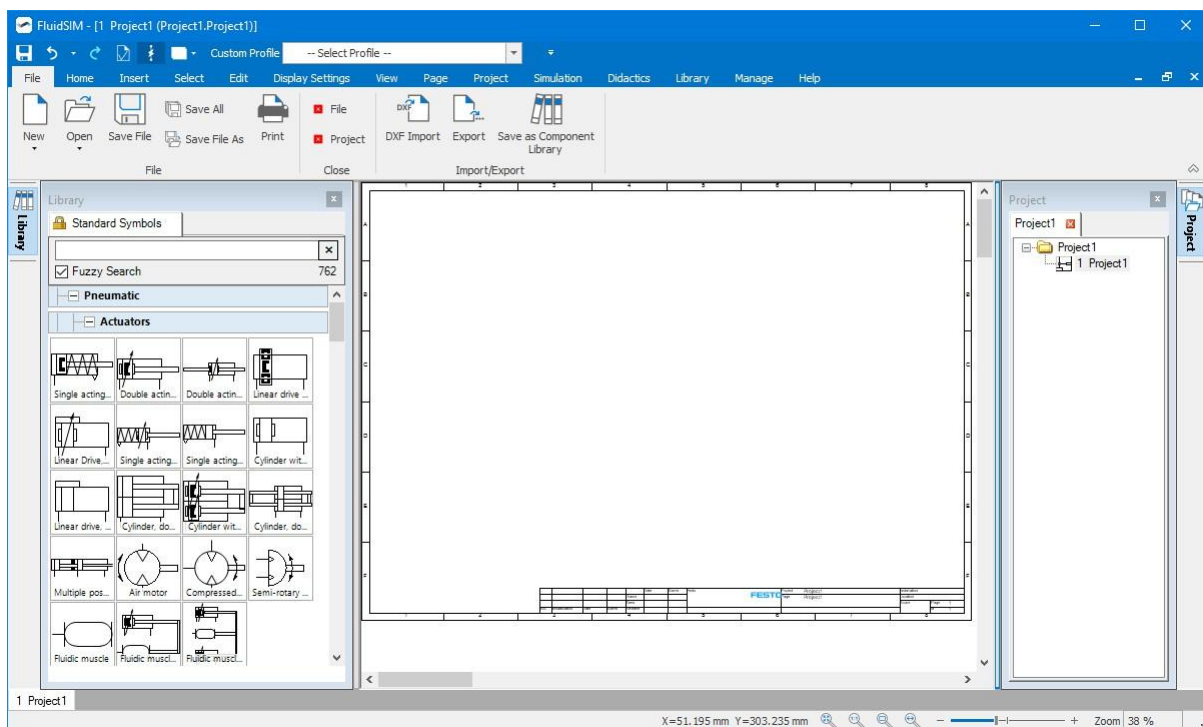


Figure 4.35: Typical components of an electro-pneumatic system

4.2.2 Information Sheet - Designing and Simulating a Basic Electro-Pneumatic Circuit in FluidSIM

Step-by-Step Circuit Design in FluidSIM

Step 1 - Open Project and Select Library

- Launch **FluidSIM** and create a new *Electro-pneumatics project*.
- From the component library, select:
 - Pneumatic elements under “Actuators”, “Valves”, “Air Preparation”.
 - Electrical elements under “Switches”, “Relays”, “Solenoids”.

Step 2 - Build the Pneumatic Section

1. Place an **Air Source** and connect it to a **Service Unit (FRL)**.
2. Insert a **5/2-way DCV (Double Solenoid type)** and connect:
 - Port 1 → Pressure source.
 - Ports 2 and 4 → Cylinder ports (A and B).
 - Ports 3 and 5 → Exhaust (open to atmosphere).
3. Add a **Double-acting Cylinder** and connect its ports to valve outlets.

Step 3 - Build the Electrical Control Section

1. Add a **24V DC Power Supply**.
2. Place two push buttons:
 - Start Button (Normally Open)
 - Stop Button (Normally Closed)
3. Insert two **relays (K1 and K2)** to create Set/Reset or sequential logic.
4. Add solenoid coils:
 - Y1 → Extends cylinder (A+)
 - Y2 → Retracts cylinder (A-)
5. Wire the circuit according to desired sequence (e.g., pressing Start energizes Y1, cylinder extends; limit switch triggers Y2 for retraction).

Step 4 - Integrate Electrical and Pneumatic Sections

- Connect solenoid coils (Y1, Y2) to their corresponding pneumatic valve actuation ports (14 and 12).
- Ensure common ground for the electrical system.

Step 5 - Label and Annotate

- Label each component using ISO designations:

- Cylinders: -MM1, -MM2
 - Valves: -QM1, -QM2
 - Sensors: -BG1, -BG2
 - Coils: -Y1, -Y2
- Add connection lines with flow arrows for clarity.

Simulation and Testing

Step 1 - Simulation Setup

- Click the **Simulation Mode (F9)** button.
- Verify air pressure lines (blue) and electrical flow (red) are visible.
- Ensure all electrical components are powered.

Step 2 - Functional Simulation

1. Press the Start button → observe solenoid Y1 energize and cylinder extend.
2. When cylinder reaches end position, limit switch LS2 activates → relay energizes solenoid Y2 → cylinder retracts.
3. Observe LED indications on the relay and valve solenoids.

Step 3 - Analyze Circuit Behavior

- Verify correct timing and switching.
- Observe pressure distribution (dark blue = pressurized, light blue = exhaust).
- If circuit fails to operate:
 - Check electrical continuity.
 - Ensure solenoid coil is assigned to correct valve port.
 - Confirm limit switches return to normal after actuation.

4.2.3 Information Sheet: Design Electro-pneumatic Sequence Control Table and Circuit

The process for implementing sequential operation generally involves three major stages:

1. Make a Sequence Table
2. Remember Three Crucial Programming Points
3. Draw the Circuit

1. Make a Sequence Table

Creating a sequence table is the first step in planning sequential operations. The table defines the motion order, sensor conditions, relay steps, and output actions (solenoid valves). It helps to visualize the logical relationship between **inputs, control relays, and outputs** before programming.

Row Category	Description and Examples
Flow / Step	Defines the sequence of actuator movements. Example: A+, A-, A+, B+, B-, C+, C-, A-. The final column is usually marked as END .
Sensor / Switch	Specifies the input device (limit switch, proximity sensor, or pushbutton) confirming each motion. Examples: SW, LS2, LS1, LS4, LS3, etc.
Relay	Represents the internal control relay or program step number. Examples: R1, R2, R3, etc.
Output	Lists the solenoid valve (SOL) or actuator output corresponding to that step. Examples: SOL1, SOL2, SOL3, SOL4.

Example Sequence Table (A+ A-)

Parameter	Step 1	Step 2	Step 3
Flow	A+	A-	END
Sensor / Switch	Switch	LS2	LS1
Relay	1	2	3
Output	SOL1	SOL2	X

Example Sequence Table (A+ B+ A- B-)

Parameter	Step 1	Step 2	Step 3	Step 4	Step 5
Flow	A+	B+	A-	B-	END
Sensor / Switch	SW	LS2	LS4	LS1	LS3
Relay	1	2	3	4	5
Output	SOL1	SOL3	SOL2	SOL4	X

Example Sequence Table (A+ A- B+ B-)

Parameter	Step 1	Step 2	Step 3	Step 4	Step 5
Flow	A+	A-	B+	B-	END
Sensor Switch /	SW	LS2	LS1	LS3	LS4
Relay	1	2	3	4	5
Output	SOL1	SOL2	SOL3	SOL4	X

Example Sequence Table (A+ B+ C+ A- B- C-)

Parameter	S1	S2	S3	S4	S5	S6	S7
Flow	A+	B+	C+	A-	B-	C-	END
Sensor Switch /	SW	LS2	LS3	LS1	LS4	LS5	LS6
Relay	1	2	3	4	5	6	7
Output	SOL1	SOL3	SOL5	SOL2	SOL4	SOL6	X

Example Sequence Table (A+ B+ A- B- (B single-acting))

Parameter	S1	S2	S3	S4	S5
Flow	A+	B+	A-	B-	END
Sensor Switch /	SW	LS2	LS3	LS4	LS1
Relay	1	2	3	4	5
Output	SOL1	SOL3	SOL2	X (spring return)	X

Example Sequence Table A+ B+ A- B- (valve 5/2 spring return))

Parameter	S1	S2	S3	S4	S5
Flow	A+	B+	A-	B-	END
Sensor Switch /	SW	LS2	LS3	LS4	LS1
Relay	1	2	3	4	5
Output	SOL1	SOL3	X (spring return)	SOL4	X

Notes on Solenoid Valves and Cylinders

- Cylinders can be **single-acting** or **double-acting**.
- Double-acting cylinders are controlled using two solenoid valves.
- For example, SOL1 may control extension (A+), while SOL2 controls retraction (A-).

- In case of single-acting cylinders, the relay for retraction may not be needed.
- In case of single solenoid valves, the return relay will not be used. Instead, the forward relay will be deenergized to allow spring return.

2. Remember Three Programming Points

The following three logic points form the foundation of sequential control programming, whether using relay logic or PLC memory bits.

1. **Set Circuit (except END):** Latch (set) the relay for every line, except the END.
2. **Pre-step Output Condition:** The output or relay of the previous step must be part of the condition that enables the next step to start. For the first line this will be the Start Switch (SW).
3. **Reset Circuit (by next step output):** Each relay should be reset by the output signal of the next step in the sequence. A **Reset Switch (SW)** or break contact is commonly used to achieve this.

3. Draw the Circuit

After defining the sequence and programming logic, the next step is drawing the actual circuit. This may be a relay-based diagram or a PLC ladder logic diagram (LD format). Typical components in the circuit include:

- Input switches or sensors (SW, LS)
- Relays (R1, R2, etc.)
- Solenoid valves (SOL1, SOL2, etc.)

An example of a basic control flow:

- Input signal from R1 and LS2 triggers the output relay R2.
- R2 latches itself until LS2 condition changes.

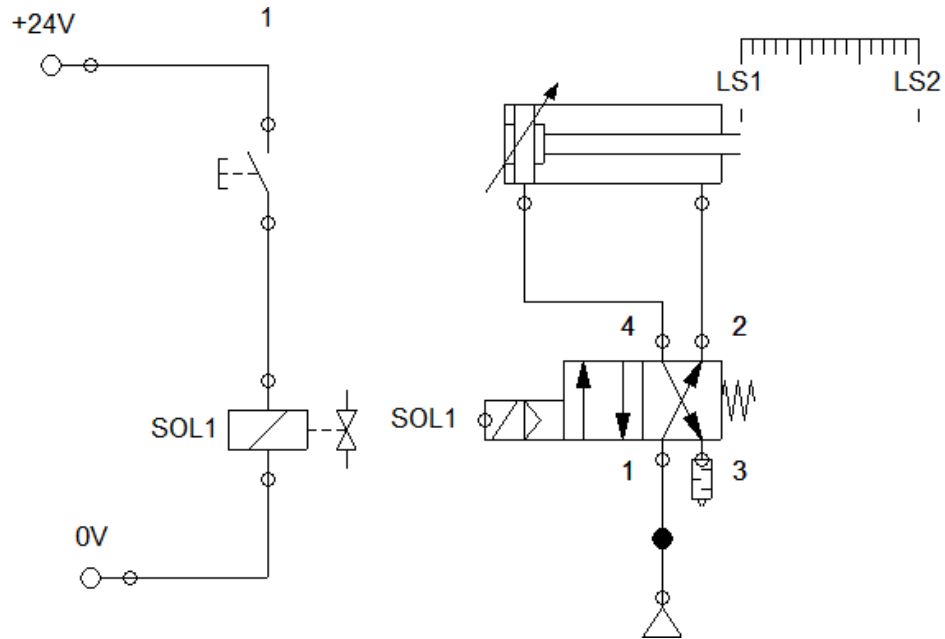


Figure 4.36: Circuit for sequence control of A+ and A- using relays and limit switches

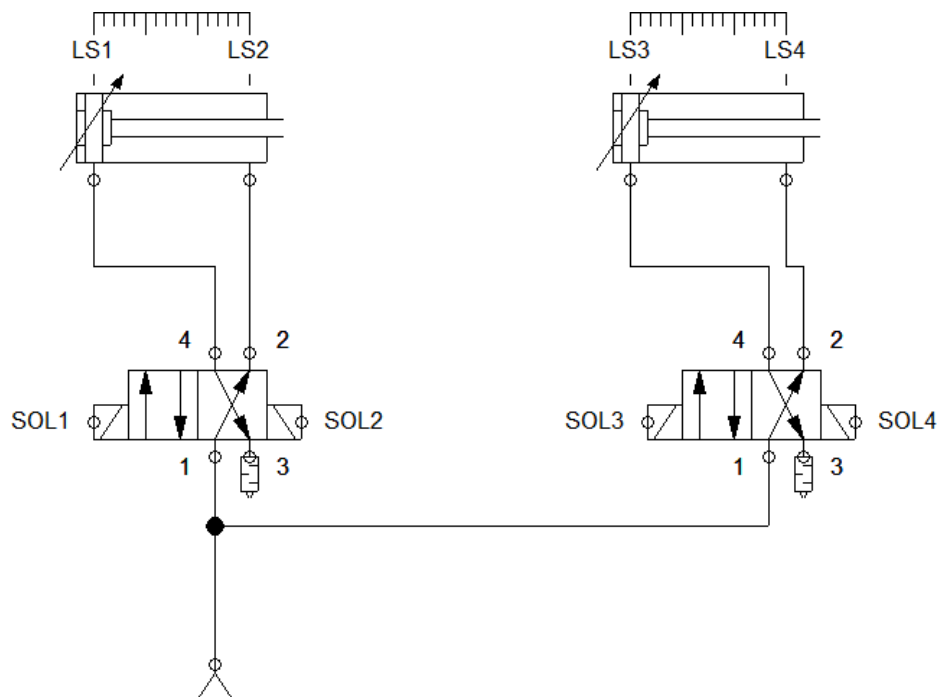


Figure 4.37: Circuit for sequence control of A+ B+ A- B- using relays and limit switches

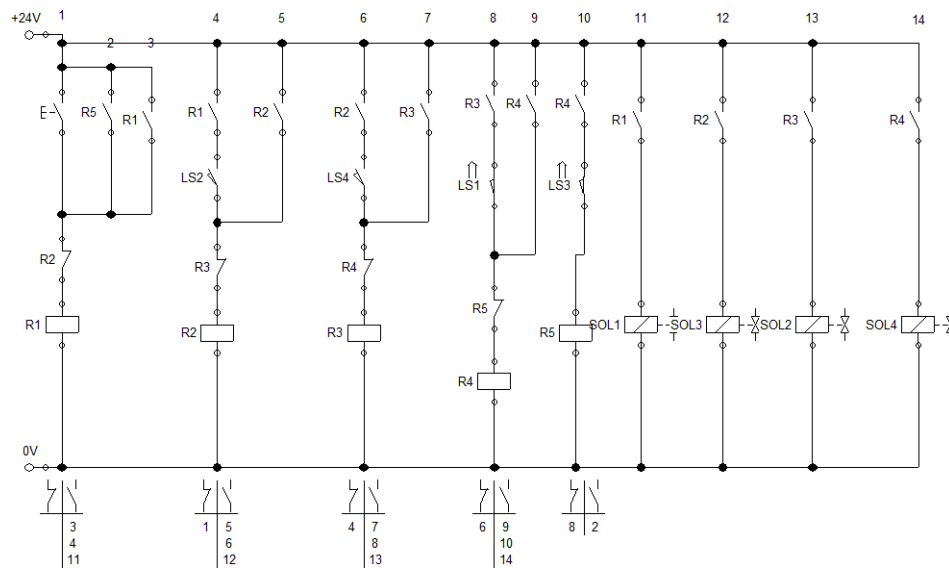


Figure 4.38: Circuit for sequence control of A+ B+ A- B- using relays and limit switches

Special Considerations for Sequence Control

- **Repeat Sequence:** To repeat a sequence, include:
 1. A new Set circuit.
 2. An Initial Position check (ensuring all cylinders are retracted).
- **Initial Position:** The repeat condition should begin when all initial position sensors (e.g., LS1 for A-) are active.
- **Emergency Stop (EMS / EMO):** If an E-Stop is pressed, all motion stops immediately. When released, the system may resume operation except at the first step, which requires a manual “Repeat Start” switch.
- **Output Connection:** Ladder logic outputs should be **serially connected (interlocked)** for reliability — for example, connecting SOL1 (Y21) and SOL2 (Y22) in series when mutual exclusion is needed.

4.2.4 Design and Simulate Pneumatic and Electro-Pneumatic Circuit (A+ A-)

Section	Content
Objective	Design and simulate a pneumatic and electro-pneumatic circuit that performs one complete cylinder cycle (A+ A-), demonstrating pilot-latching and feedback control.
<i>Continued on next page</i>	

4.2 Learning Outcome: Design and Simulate Electropneumatic Circuit

Section	Content
Tools and Equipment	PC with FluidSIM P/E software, EasyPort (optional), component library (cylinders, solenoid valves, limit switches, relays).
Procedure	1. Place components in FluidSIM workspace. 2. Connect main air supply to valve inlet and ports 2, 4 to cylinder. 3. Add pilot lines: left for A+, right for A-. 4. Install flow control valves for meter-out speed regulation. 5. Connect pushbutton and limit switch. 6. Connect solenoid coils, relays, and supply voltage for electro-pneumatic circuit. 7. Simulate and verify the sequence: A+ → limit switch → A-.
Key Points	- Circuit demonstrates one full cylinder cycle (A+ A-). - Ensure proper pilot pressure and leak-free connections. - Use meter-out flow control for smooth motion. - Operate within 400-600 kPa pressure range.

Specification Sheet

Circuit Diagram for Single Cylinder A+ A- Operation

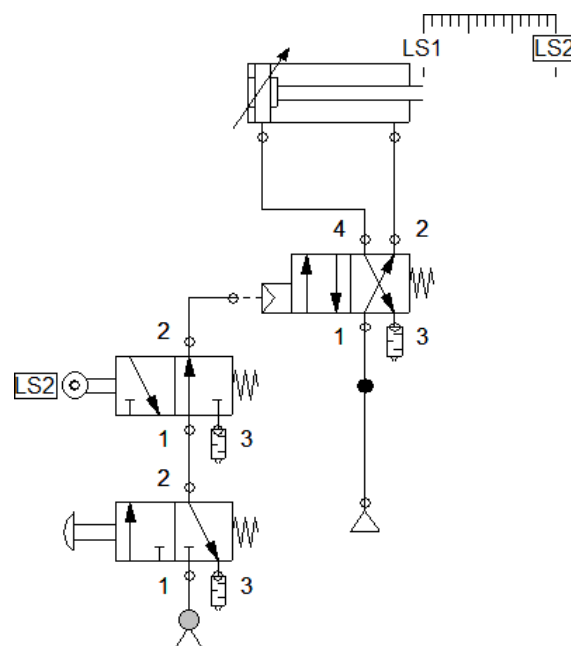


Figure 4.39: Pneumatic circuit for A+ A- sequence

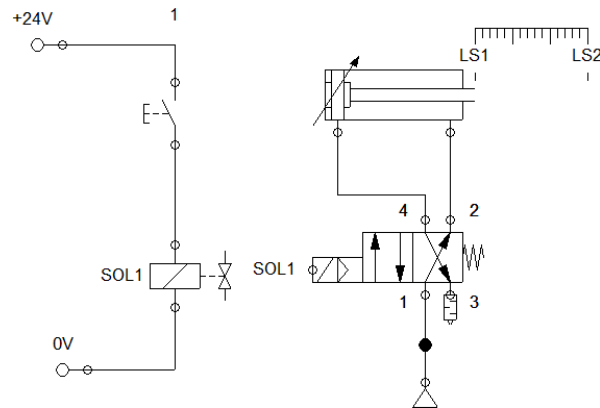


Figure 4.40: Electro-pneumatic circuit for A+ A- sequence

4.2.5 Job Sheet - Design and Simulate Multiple Cylinder Sequence Circuit in FluidSIM

Section	Content
Job Title	Design and Simulate Multiple Cylinder Sequence Circuit in FluidSIM
Objective	To design, simulate, and verify various electro-pneumatic control circuits in FluidSIM software based on specific sequential cylinder operations. The task develops the trainee's ability to interpret sequence tables, apply control logic, and validate proper motion sequences in simulation.
Personal Protective Equipment (PPE)	Safety shoes and anti-static wrist strap (if connected to a hardware trainer).
Tools and Equipment	PC with FluidSIM P/E software, EasyPort (optional), component library (cylinders, solenoid valves, limit switches, relays).

Continued on next page

4.2 Learning Outcome: Design and Simulate Electropneumatic Circuit

Section	Content
Procedure	1. Start a new FluidSIM project and save it using the circuit name. 2. Insert required pneumatic and electrical components from the library. 3. Arrange the circuit logically from air supply to actuator, maintaining clear signal flow. 4. Draw electrical links between sensors, relays, and control coils. 5. Assign input and output signals for control and feedback. 6. Apply logic to achieve the desired motion sequence. 7. Run the simulation, observe the actuator movement, and adjust timing or control parameters as needed. 8. Verify correct sequence operation and ensure all motions return safely to the initial state.

Specification Sheet

Tasks and Required Sequences

Sequence Name	Description / Design Requirement
A+ A-	Single-cylinder extend/retract operation using two solenoid valves.
A+ B+ A- B- (both sides solenoid)	Two double-acting cylinders; latching logic for each step.
A+ B+ A- B- (one side solenoid)	Combination of bistable and monostable valves; include coil self-hold.
A+ B+ A- B- (single / continuous cycle)	One-cycle and repeat mode using Start and Repeat switches.
A+ B+ A- B- (A single acting)	Mixed-type circuit: 3/2 for A, 5/2 for B.
A+ B+ C+ A- B- C-	Three-cylinder sequence with forward and reverse order.
A+ B+ A- B- C+ C-	Interleaved two-cylinder sequence followed by C-cycle; use relays R1-R6.
A+ A- B+ C+ B- C-	Advanced pattern with partial repetition; use relay contacts or internal bits for signal memory.

4.2.6 Job Sheet - Sliding Door Logic Circuit Simulation in FluidSIM

Section	Content
PPE	None required (software-only simulation).
Materials	PC with FluidSIM P/E , component library (valves, cylinder, proximity switches, shuttle valve, dual-pressure valve).
Procedure	1. Create new project and insert cylinder, 5/2 valve, proximity switches, AND valve and OR valve. 2. Arrange circuit so that OR creates start command from two pushbuttons, AND requires end-position + start. 3. Connect logical elements with signal lines and add flow control valves to adjust motion speed. 4. Run simulation and trigger pushbuttons to verify door opens and closes only when end-position is valid. 5. Observe latching behaviour of the 5/2 valve during state holding.
Key Points	- OR allows start from two locations. - AND ensures start only when end-position + pushbutton true. - 5/2 double pilot valve maintains state after pulse (latching).

Specification Sheet - Circuit Diagram

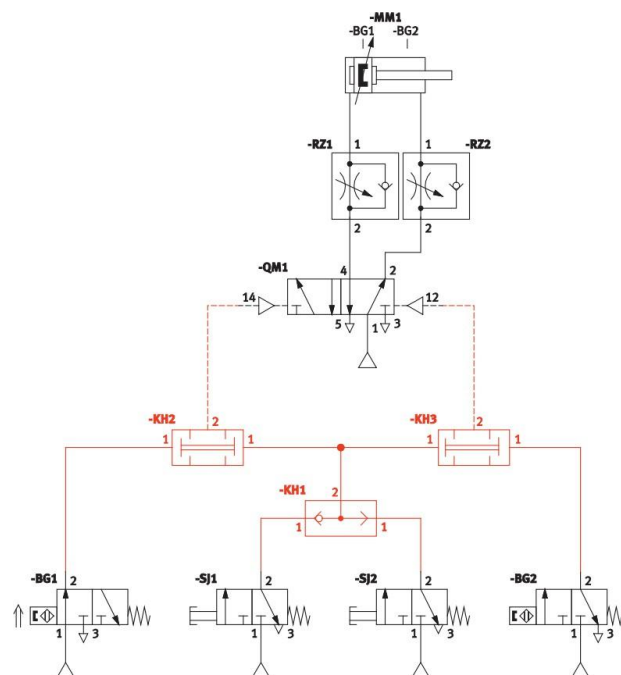


Figure 4.41: Pneumatic logic circuit for sliding door actuation (simulation)

4.2.7 Job Sheet - Simulated Cleaning Cycle (Oscillation for 20 s)

Section	Content
PPE	None required (software-only simulation).
Materials	PC with FluidSIM P/E, component library (cylinder, proximity switches, double pilot valve, shuttle valve, pneumatic timer, flow control valves).
Procedure	1) Create new project and place cylinder, timer, proximity sensors, pilot valve, shuttle valve, pushbutton. 2) Convert one valve to 3/2 function by plugging one output. 3) Assign proximity sensors to define mid and front end positions. 4) Use shuttle valve to combine start signal and timer return signal. 5) Run simulation — cylinder oscillates for 20 seconds, then returns home.

Specification Sheet - Circuit Diagram

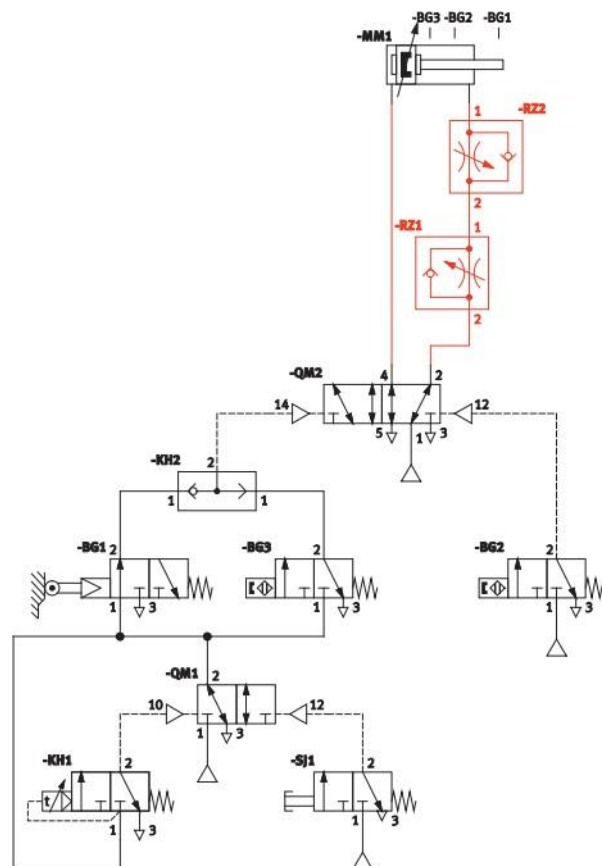


Figure 4.42: Simulated Cleaning Cycle (Oscillation for 20 s) Circuit

4.2.8 Job Sheet - Simulate Pressure-Dependent Stamping Logic

Section	Content
Materials	PC with FluidSIM P/E and component library (cylinder, pressure sequence valve, pressure regulator, dual-pressure valve, double pilot valve, flow controls, sensors, pushbuttons).
Procedure	1) Analyse logic: stamping allowed only when rear position confirmed and minimum pressure reached. 2) Place components in FluidSIM: cylinder, pressure sequence valve, pressure regulator, dual-pressure valve, 5/2 valve, flow controls, pushbuttons, sensor. 3) Connect logic: start = pushbutton AND rear sensor AND pressure OK. 4) Limit stamping force using regulator. 5) Run simulation — verify slow controlled A+ stroke and quick A- return.

Specification Sheet

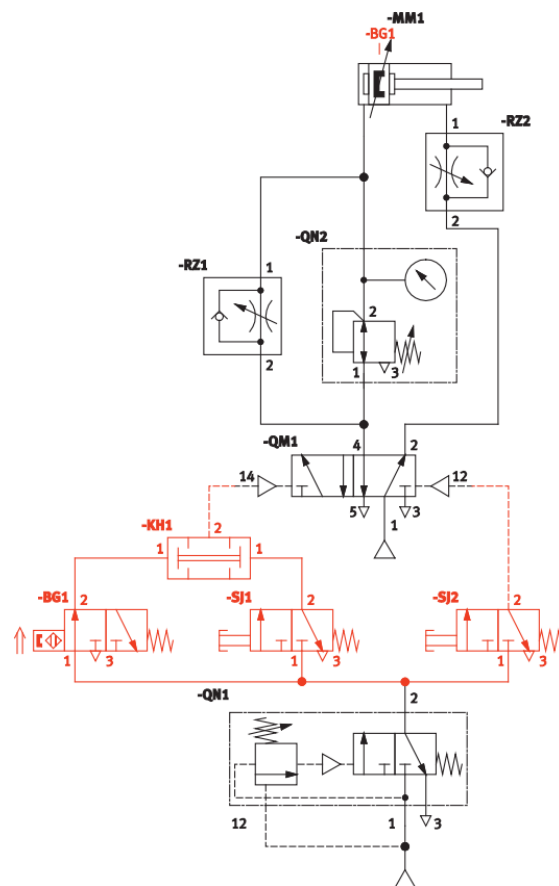


Figure 4.43: Circuit for sequence control stamping operation

4.2.9 Job Sheet - Simulate Pressure-Conditioned Clamp and Press Sequence

Section	Content
Materials	PC with FluidSIM P/E and component library (single-acting cylinder, double-acting cylinder, pressure sequence valve, dual-pressure valves, double pilot valve, flow controls, sensors, pushbutton).
Procedure	1) Analyse logic: clamping must complete <i>and</i> reach pressure threshold before pressing is permitted. 2) Build circuit in FluidSIM: clamping circuit → press circuit. 3) Convert DP valve to 3/2 for clamping logic. 4) Implement AND logic: rear position + pressure $\geq$ threshold = enable press. 5) Set flow control for smooth timings. 6) Run simulation — confirm: clamp → press → both retract.

Specification Sheet

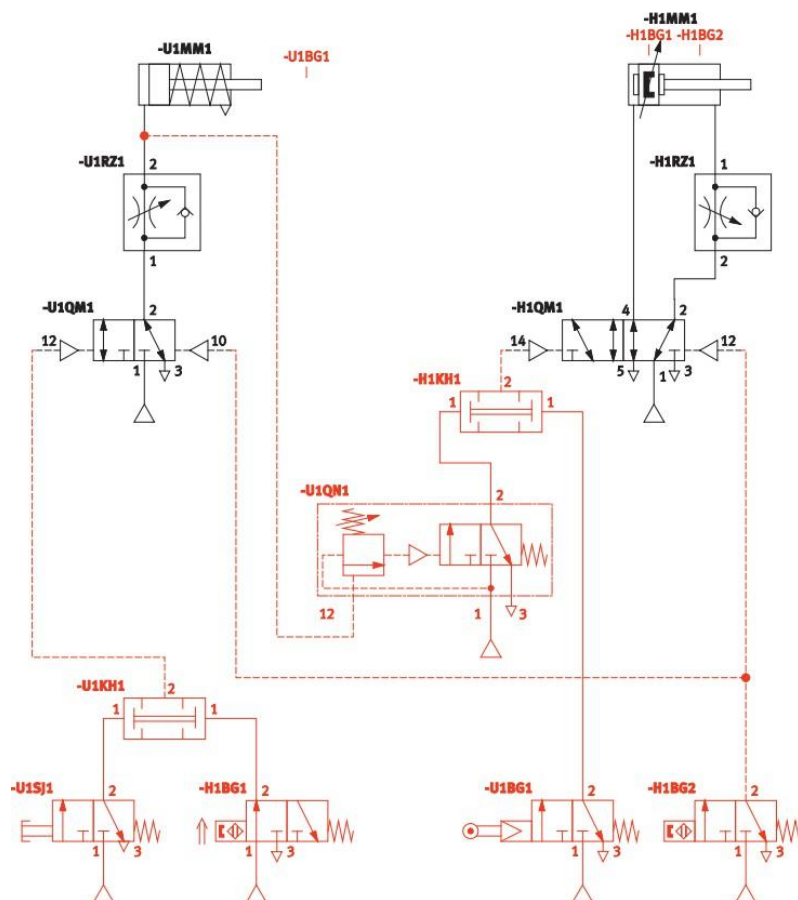


Figure 4.44: Circuit for sequence control of press after clamp

4.2.10 Self-Check Questions

No.	Question
1	What is the name of the standard simulation and CAD tool recommended for designing and simulating pneumatic and electro-pneumatic circuits?
2	According to the sequencing concept, what three crucial programming points must be addressed when designing control logic for sequential operation?
3	In the context of a sequential operation table, what condition typically triggers the start of a step (e.g., Step 2)?
4	What logic element is commonly used in electro-pneumatic circuits to ensure that a relay or step remains active until the subsequent step is initiated?
5	Describe the arrangement of components in FluidSIM used to define the pneumatic section of an electro-pneumatic circuit, starting from the air source.
6	What is the primary purpose of adding flow control valves to the cylinder ports in a pneumatic circuit simulation?
7	What two elements are required in the electrical control section of an electro-pneumatic circuit to initiate and halt the sequence manually?
8	In a sequence table, if SOL1 controls A+ and SOL2 controls A-, what sensor/switch typically triggers the transition from A+ to A-?
9	When implementing sequential control using relays, what component should reset the relay for the current step?
10	How does the FluidSIM software indicate which lines are pressurized in the pneumatic circuit and which components are active in the electrical circuit during simulation?
11	How are ISO designations, such as -MM1 (cylinder) and -Y1 (solenoid coil), used when labeling components in the FluidSIM software?

4.2.11 Answer Key

No.	Answer / Guidance
1	FluidSIM is the standard simulation and CAD tool used for pneumatic and electro-pneumatic circuit design.
2	The three crucial programming points are: 1) Set Circuit (latch the relay for every line, except the END); 2) Pre-step Output Condition (the output or relay of the previous step must be part of the condition that enables the next step); and 3) Reset Circuit (each relay should be reset by the output signal of the next step).
3	The next step (e.g., Step 2) is triggered by the sensor/switch that confirms the completion of the previous movement (e.g., A+), typically indicated by a limit switch like LS2. For the first step, the trigger is the Start Switch (SW).
4	The Set Circuit (latching) is used to maintain the active state of a relay until it is reset by the signal of the next step in the sequence.
5	The pneumatic section starts with the Air Source, connected to a Service Unit (FRL), which then connects to the directional control valve (DCV), and finally to the actuator (Cylinder).
6	Flow control valves are added to adjust the motion speed (traversing speed) of the cylinder piston, specifically for smooth motion regulation (often meter-out control).
7	The electrical control section requires Start (Normally Open) and Stop (Normally Closed) push buttons, typically wired along with relays and solenoid coils.
8	The transition from A+ (extension) to A- (retraction) is triggered by the sensor (limit switch or proximity sensor) that detects that cylinder A has reached its front end position (e.g., LS2 or BG2).
9	Each relay should be reset by the output signal of the next step in the sequence. For the final step, a Reset Switch or break contact can be used.
10	During simulation in FluidSIM, pressurized air lines are highlighted (e.g., dark blue), and active electrical flow is indicated by red lines. Additionally, LED indications can be observed on active relays and valve solenoids.
11	Components must be labeled using ISO designations (e.g., -MM1, -QM1, -Y1) to comply with standards and ensure clear identification of cylinders, valves, sensors, and coils.

4.3 Learning Outcome: Carry out Electropneumatic Circuit Connection

Objective: To recognize and interpret the standards, symbols, and fundamental components (both electrical and pneumatic) necessary for understanding robot and electropneumatic circuit diagrams, including the basic components in robots and their control systems.

Learning Activity: Study the specific industry standards (ISO 1219-1 and DIN EN 81346-2) applied in pneumatic and electrical circuit diagram representation; identify and describe the symbols for directional control valves, actuators, and logical pneumatic elements; recognize and define basic electrical control components (e.g., relays, limit switches, solenoids); and analyze how electrical and pneumatic systems are integrated to form control systems.

4.3.1 Information Sheet - ISO / IEC Standards for Emergency Stop (E-Stop)

The International Organization for Standardization (ISO) and related safety standards (such as **IEC 60204-1** and **EN ISO 13849-1**) specify the critical requirements for **Emergency Stop (E-Stop)** devices to ensure machinery safety. These define both the **physical wiring/hardware setup** and the **programming/control logic** necessary for integration with PLCs or robot controllers.

The core standard defining functional requirements and design principles for the E-Stop function is:

ISO 13850:2015 - Safety of Machinery - Emergency Stop Function

I. General and Functional Requirements (ISO 13850)

The E-Stop function is a complementary protective measure intended to avert actual or impending emergency situations initiated by a single human action.

1. **Highest Priority Override:** The E-Stop function shall be available and operational at all times. It must override all other functions and operating modes of the machine.
2. **Stopping Behavior:** Hazardous machine movements must stop in a controlled and safe manner, *without introducing new hazards* and without requiring additional intervention.
3. **No Automatic Restart:** Once actuated, the E-Stop function must be maintained until manually reset. Resetting (unlatching) the E-Stop device shall not automatically restart the machine. A separate, deliberate action (such as pressing a Start or Reset button) is required to resume operation.
4. **Decision Simplicity:** The E-Stop must be designed such that activation requires no consideration of the resulting effects — the operator must be able to act instinctively in an emergency.

II. Critical Wiring and Hardware Requirements

Safety standards require that the E-Stop safety circuit operates independently of the PLC or robot controller's software, ensuring **fail-safe behavior**.

1. Fail-Safe Contact Logic (NC)

- E-Stop buttons must use **Normally Closed (NC)** contacts.
- NC wiring follows the fail-safe principle: an open circuit (caused by wire break or button press) is interpreted as an active stop command.

2. Hardware Independence and Redundancy (Duplication)

- The E-Stop circuit must be established **independently of the PLC**—typically using safety relays or contactor logic.
- Modern robot controllers integrate a **duplicated emergency stop circuit**.
- The E-Stop switch must be a **double contact (2-line)** type for redundancy.
- Dual-line (redundant) wiring ensures the system remains functional even if one contact fails.

3. Physical Device Specifications

- The E-Stop actuator must be **mushroom-shaped, red, with a yellow background**.
- It must be **mechanically latching** — push-to-stop and twist/pull/key-to-release.

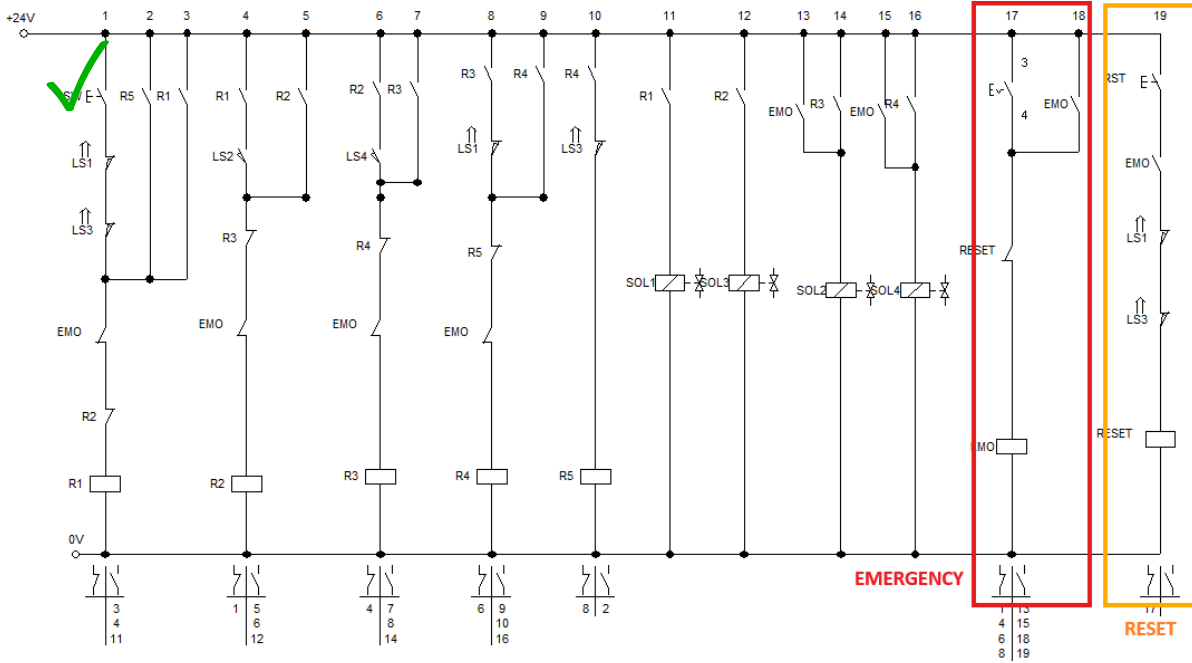
4. Wiring Integrity and Diagnostics

- Use **shielded cables** to reduce noise interference.
- Limit total cable length between switch and terminal block to **15 m or less**.
- Operate within **24 V DC protective extra-low voltage (PELV)**.
- Do not ground the positive side of the 24 V DC power supply used for E-Stop I/O.
- To achieve the highest safety category (**SIL 3, Category 4, PL e**), enable test pulse diagnosis and verify both emergency stop lines function independently.

III. Programming and Control Logic Requirements

Programming defines how PLCs and robot controllers respond to the emergency input signal according to **IEC 60204-1 Stop Categories**.

4.3 Learning Outcome: Carry out Electropneumatic Circuit Connection



(b) Correct PLC Implementation - E-Stop ensures safe stop of all outputs, needs manual reset

Figure 4.45: Comparison of Incorrect vs Correct PLC E-Stop Implementation

3. Post-Stop Handling

- System restart requires a separate manual action (Reset/Start button).
- After an E-Stop, robot systems must be moved to a safe or home position before resuming automatic mode.
- The operator must ensure no interference with peripheral systems before restarting.

4. Safety PLCs (F-CPU)

- Use **Safety PLCs (F-CPUs)** for high-integrity safety functions meeting SIL or PL levels.
- Safety programs (**F-programs**) run independently of the standard PLC cycle.
- These include redundant CPUs, diagnostic routines, and fail-safe outputs.
- Verify that the F-runtime group cycle time is within limits and F-function blocks (F-FCs) initialize outputs correctly.

4.3.2 Information Sheet - Overlapping Sequential Control with Double Solenoid Valves (Including Reset Step)

Overview An additional control step (**Step 5**) is implemented in circuits using double solenoid valves to ensure all steps are cleared after one complete cycle.

Initial Position Cylinders **-MM1** and **-MM2** are in the rear end position. Limit switches **-BG1**, **-BG3** and proximity switches **-BG2**, **-BG4** are connected to relays **-KF1**, **-KF2**, **-KF3**, and **-KF4** respectively in the circuit.

Step 1-2 If cylinder **-MM1** is at the rear end (NO contact **-BG1** actuated, relay **-KF1** energized) and the Start button **-SF1** is pressed (relay **-KF5** energized), relay **-KF6** energizes and self-latches through its NO contact (path 9). Its contact in path 10 initializes Step 2, and contact in path 18 energizes solenoid coil **-MB1** of valve **-QM1**. Valve **-QM1** switches, extending cylinder **-MM1**.

Step 2-3 When cylinder **-MM1** reaches the front end, sensor **-BG2** (NO) activates and energizes relay **-KF2**. If Step 1 is active (**-KF6** latched), relay **-KF7** energizes and self-latches (path 11). Its NO contacts close paths 12 and 21, energizing solenoid coil **-MB2** of valve **-QM2** to advance cylinder **-MM2**. At the same time, contact **-KF7** in path 20 opens, de-energizing **-MB1**.

Step 3-4 When cylinder **-MM2** reaches the front end, sensor **-BG4** (NO) activates and energizes relay **-KF4**. If Step 2 is active (**-KF7** latched), relay **-KF8** energizes and self-latches (path 13). Its NO contacts close paths 14 and 20, energizing solenoid coil **-MB4** to reset valve **-QM2**, retracting cylinder **-MM2**. Simultaneously, contact **-KF8** in path 19 opens, de-energizing **-MB2**.

Step 4-5 When cylinder **-MM2** reaches its rear end, sensor **-BG3** (NO) activates and energizes relay **-KF3**. If Step 3 is active (**-KF8** latched), relay **-KF9** energizes and self-latches (path 15). Contacts **-KF9** in paths 16 and 21 energize solenoid coil **-MB2**, resetting valve **-QM1** to retract cylinder **-MM1**. Simultaneously, contact **-KF9** in path 20 opens, de-energizing **-MB4**.

Step 5-6 (Reset Step) When cylinder **-MM1** reaches its rear end (sensor **-BG1** actuated, relay **-KF1** energized) and Step 4 is active (**-KF9** latched), relay **-KF10** energizes and self-latches (path 17). Contact **-KF10** in path 21 opens, de-energizing **-MB2**. At the same time, contact **-KF10** in path 8 opens, clearing the latching of all previous steps (**-KF6** through **-KF10**). All cylinders return to the rear end position, and the sequence is reset.

4.3.3 Information Sheet - Single Cycle and Continuous Cycle Operating Modes of the Controller

Single Cycle Sequence Cylinder **-MM1** is in the rear end position (NO contact **-BG1** actuated, relay **-KF3** energized, and contact **-KF3** in path 13 closed). When the Start button is pressed (relay **-KF2** energized, contact **-KF2** in path 13 closed), solenoid coil **-MB1** energizes and valve **-QM1** switches. Cylinder **-MM1** advances.

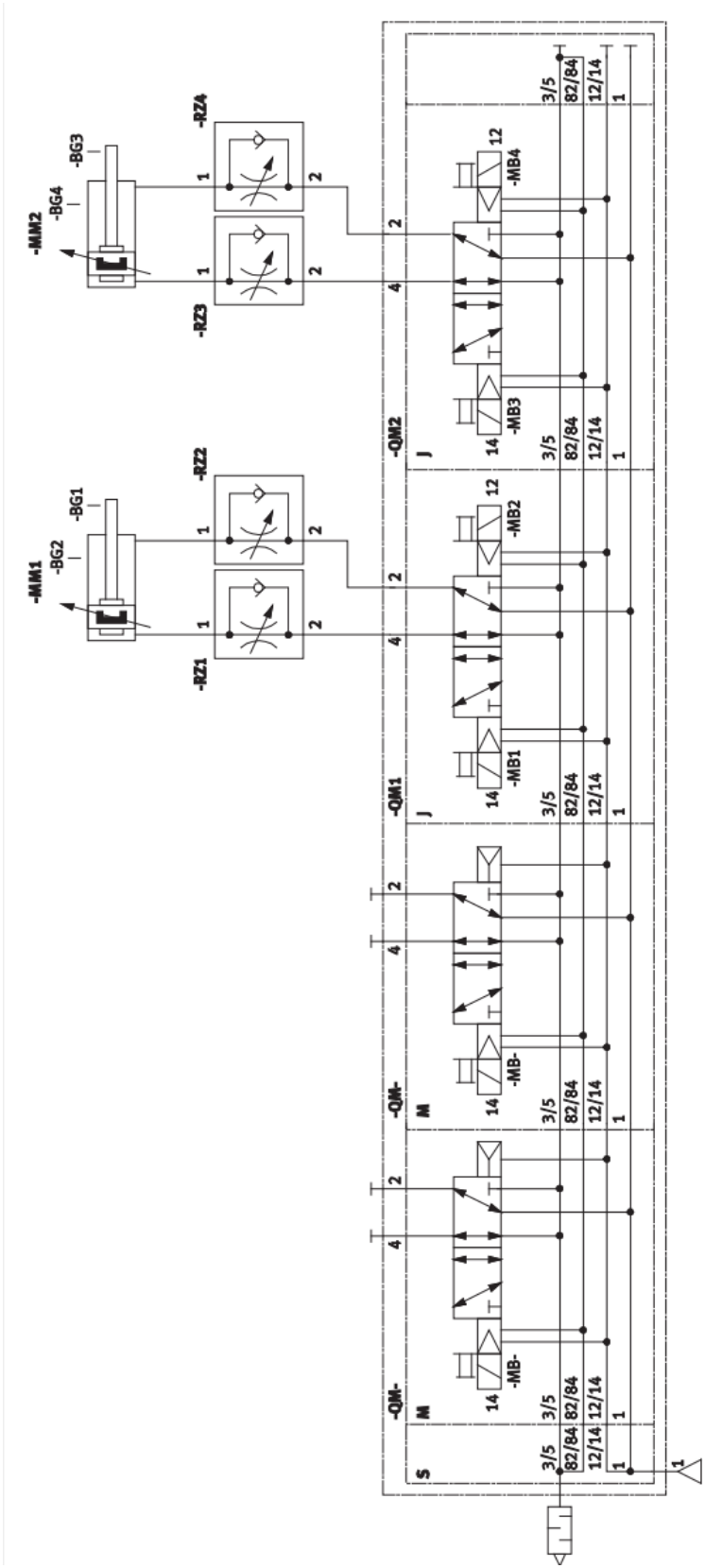


Figure 4.46: Sequential control of two cylinders using double solenoid valves with reset step (Pneumatic Circuit)

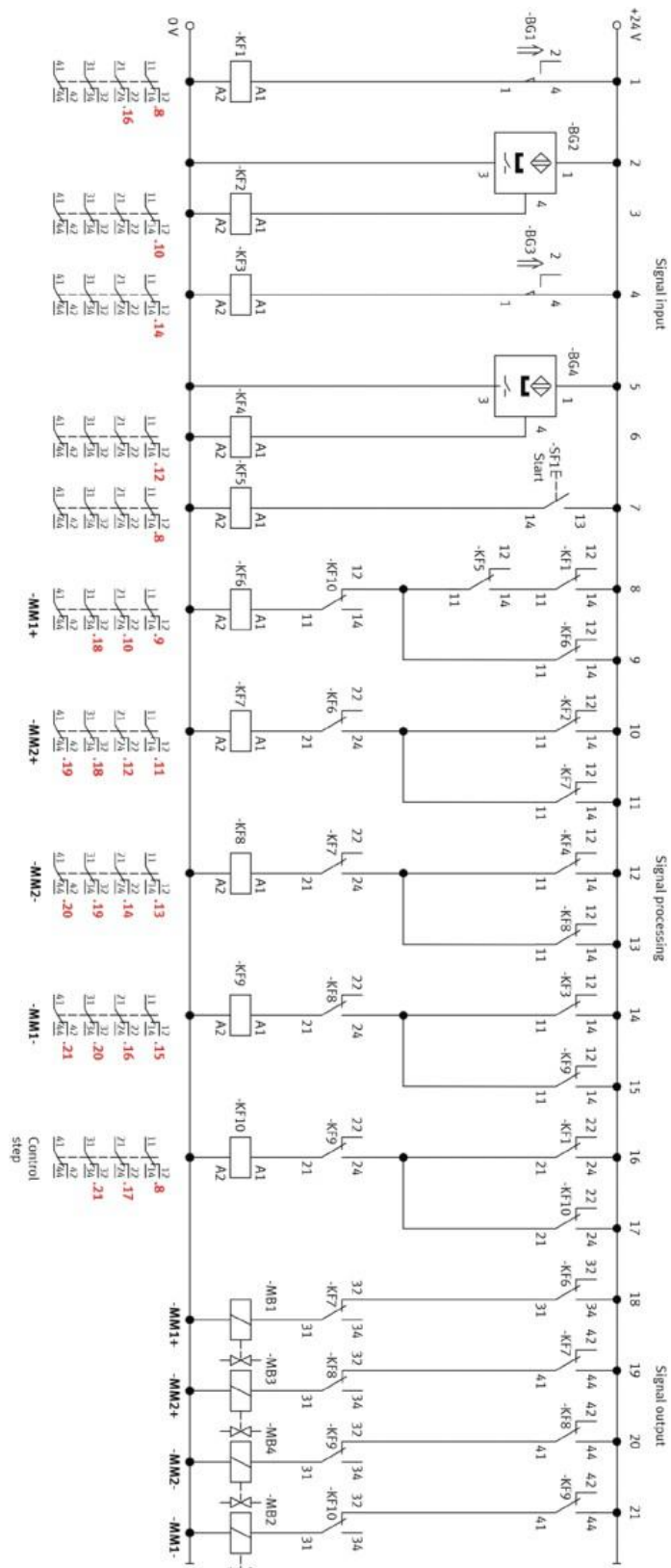


Figure 4.47: Sequential control of two cylinders using double solenoid valves with reset step (Electrical Circuit)

4.3 Learning Outcome: Carry out Electropneumatic Circuit Connection

As the cylinder leaves its rear end position, sensor **-BG1** deactivates, opening contact **-KF3** in path 13 and de-energizing **-MB1**. When cylinder **-MM1** reaches the front end (sensor **-BG2** actuated, relay **-KF4** energized, contact **-KF4** in path 14 closed), coil **-MB2** energizes, resetting valve **-QM1** and retracting the cylinder. At the same time, contact **-KF4** in path 11 closes, decrementing the preset counter **P6** by one.

If the cylinder leaves its front end, sensor **-BG2** deactivates, opening contact **-KF4** in path 14 and interrupting the **-MB1** circuit.

Pressure Sensing A new cycle can only begin when the operating pressure is detected by the pressure sensor **-BP1** (relay **-KF5** energized, contact **-KF5** in path 3 closed). If the pressure falls below the threshold, the current cycle stops and the system halts. This safety ensures operations occur only under sufficient pneumatic pressure.

Preset Counter Function Regardless of operating mode, the preset counter **-KK1** limits the total number of cycles. Each time cylinder **-MM1** reaches its front end (sensor **-BG2** actuated, relay **-KF4** energized, contact **-KF4** in path 11 closed), one count is subtracted. When the set count is reached, the system stops automatically. The counter must be reset manually for the next batch.

Automatic and Single Cycle Modes When relay **-KF1** is **de-energized**, self-latching is disabled at the Start button (path 4 open). Only one complete cycle executes per Start press — this is the **Single Cycle Mode**.

Continuous Cycle Mode When relay **-KF1** is **energized**, self-latching is activated via contact **-KF1** in paths 2 and 4. Multiple cycles are executed continuously until:

- the “Stop at End of Cycle” or “Continuous Off” control is activated, or
- the preset counter **P6** completes its programmed number of cycles.

4.3.4 Electro-pneumatic implementation of Emergency Stop Function

Initial Position Cylinders **-MM1** and **-MM2** are in the rear end position. Limit switches **-BG1**, **-BG3** and proximity switches **-BG2**, **-BG4** are connected to relays **-KF1**, **-KF2**, **-KF3**, and **-KF4** respectively in the electrical circuit.

Step 1-2 If cylinder **-MM1** is at its rear end (NO contact **-BG1** actuated, relay **-KF1** energized) and the Start button **-SF1** is pressed (relay **-KF5** energized), relay **-KF6** (path 10) is energized and latched through its own NO contact (path 11). Its NO contact in path 12 initializes Step 2, and in path 18 energizes solenoid coil **-MB1** of valve **-QM1**. Valve **-QM1** switches, extending cylinder **-MM1**. When **-SF1** is released, current path 7 de-energizes, but latching at **-KF6** maintains Step 1.

Step 2-3 When cylinder **-MM1** reaches the front end, sensor **-BG2** (NO) activates and energizes relay **-KF2**. If Step 1 is set (**-KF6** latched), relay **-KF7** (path 12) energizes and self-latches via path 13. Its NO contacts in paths 14 and 19 energize solenoid coil **-MB2** of valve **-QM2**, causing cylinder **-MM2** to advance and initialize Step 3.

Module-4: Perform Electro-pneumatic Sequence Control

Step 3-4 At the front end of cylinder **-MM2**, optical sensor **-BG4** (NO) activates and energizes relay **-KF4**. If Step 2 is active (**-KF7** latched), relay **-KF8** (path 14) energizes and latches via path 15. Its NO contact in path 16 starts Step 4, while another NO contact in path 19 opens, de-energizing coil **-MB2**. Valve **-QM2** returns to its normal position via spring return, retracting cylinder **-MM2**.

Step 4-5 When cylinder **-MM2** reaches its rear end (sensor **-BG3** actuated, relay **-KF3** energized) and Step 3 is active (**-KF8** latched), relay **-KF9** (path 16) energizes and latches via path 17. Its NO contact in path 18 opens, de-energizing solenoid coil **-MB1** and causing valve **-QM1** to spring-return, retracting cylinder **-MM1**. At the same time, **-KF9** opens current path 10, deactivating latching of **-KF6**, which sequentially resets relays **-KF7**, **-KF8**, and **-KF9**, clearing all steps.

Emergency Stop Function The Start button **-SF1** and sensors **-BG1-BG4** remain powered even when the emergency stop (E-Stop) is activated, allowing machine status monitoring. The E-Stop is designed as a **Normally Closed (NC)** contact, ensuring fail-safe operation in case of a broken wire. Relay **-KF10** (path 9) disconnects power from the entire signal processing and output section. When E-Stop is pressed or power fails, solenoid valves **-QM1** and **-QM2** return to their normal positions by spring return, and cylinders **-MM1** and **-MM2** retract safely.

4.3.5 Job Sheet - Multi-Cylinder Sequence (A+B+A-B-(C+B+)C-B-)

Section	Content
Objective	Execute and verify a 3-cylinder sequential circuit where one step contains a parallel stroke (C+ & B+). Work performed in both FluidSIM simulation and physical pneumatic test station.
Tools Required	FluidSIM P/E software, pneumatic trainer kit, air supply module, valves, sensors, tubing.
Procedure - Simulation	1) Create a new circuit in FluidSIM. 2) Insert 3 cylinders, valves, sensors and relays. 3) Build sequence logic ensuring each movement is dependent on the correct sensor confirmation. 4) Implement one parallel step where C+ and B+ are energized simultaneously. 5) Run simulation and confirm correct step order. 6) Adjust virtual flow restrictors for smooth timing.
Procedure - Physical Assembly	1) Mount cylinders and valves on profile plate. 2) Route tubing neatly and ensure leak-free connections. 3) Align sensors to trigger at correct end positions. 4) Set flow control valves for smooth extension / retraction. 5) Apply air supply and verify step order physically. 6) Compare physical motion against FluidSIM results.

Continued on next page

4.3 Learning Outcome: Carry out Electropneumatic Circuit Connection

Section	Content
Notes	Parallel stroke step only completes after both cylinders reach their end sensors. Differences between simulation and hardware timing are mostly due to mechanical friction and actual pressure losses.

Specification Sheet

Sequence Table

Parameter	1	2	3	4	5	6	7
Flow	A+	B+	A-	B-	C+ & B+	C-	B-
Sensor	Start	A front	B front	A rear	B rear	C front	C rear
Relay	R1	R2	R3	R4	R5	R6	R7
Output	Y1	Y3	Y2	Y4	Y5 + Y3	Y6	Y4

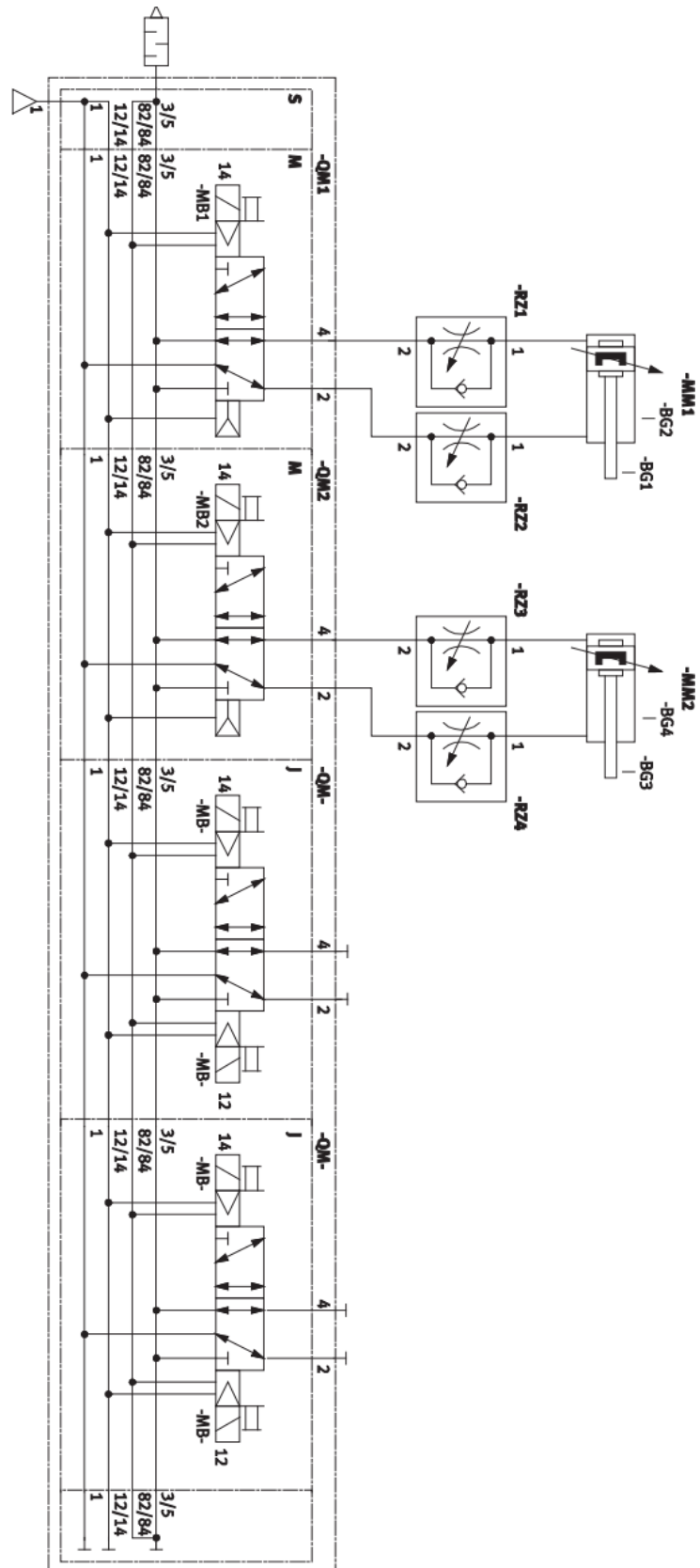


Figure 4.48: Sequential control of two cylinders with integrated emergency stop function

4.3 Learning Outcome: Carry out Electropneumatic Circuit Connection

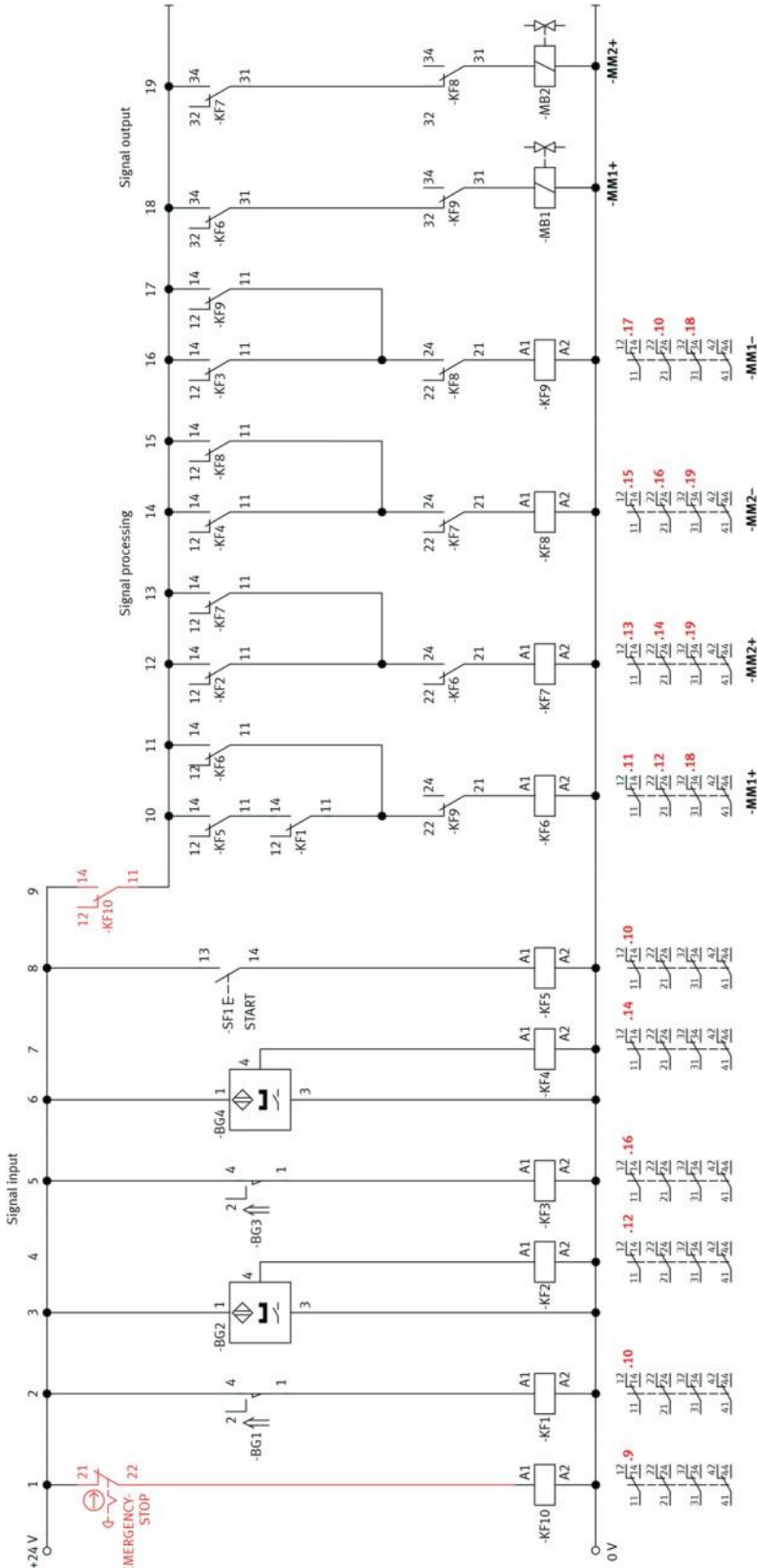


Figure 4.49: Sequential control of two cylinders with integrated emergency stop function

4.3.6 Job Sheet - Add E-Stop Button in Electro-pneumatic Circuit

Section	Content
Job Title	Add E-Stop Button in Electro-pneumatic Circuit
Objective	To implement a fail-safe Emergency Stop (E-Stop) circuit that immediately halts the electro-pneumatic process and requires a manual reset before restart.
Safety Principle	The E-Stop circuit uses dual-line, normally closed (NC) contacts connected to safety inputs. If the circuit opens—whether due to the E-Stop being pressed or a wiring fault—the system registers a stop condition automatically. This ensures fail-safe behavior in compliance with ISO 13850 and EN 60204-1.
Procedure	1. Wiring Connections - Connect the physical E-Stop pushbutton (and any safety fence switch) as an NC contact to a designated PLC input or safety relay input. - Ensure both contact lines are wired in series for redundancy and fault detection.
	2. PLC Logic Integration - Insert the E-Stop signal into the main program network in series with the power enable or Start logic. - The circuit must open (logic “0”) when the E-Stop is pressed, disabling all output energizing coils (solenoids).
	3. Control Action - When the E-Stop button is pressed, the PLC input changes to logic “0,” immediately cutting power to all active actuators and de-energizing the control relay circuit. - The pneumatic system should vent or return actuators to their safe rest positions.
	4. Reset Operation - Provide a separate “Reset” pushbutton wired to the PLC. The system must require a reset signal before restarting after an E-Stop event. This reset action should only re-enable logic when the E-Stop input is verified released (logic “1”).

Continued on next page

4.3 Learning Outcome: Carry out Electropneumatic Circuit Connection

Section	Content
Notes	- The E-Stop logic must override all automatic or manual operations. - The system must not restart automatically after the E-Stop is released. - Regularly test the E-Stop function for mechanical and electrical reliability. - All interlocks must comply with relevant safety standards (ISO 13850, EN 60204-1).
Key Technical Points	- The E-Stop input must be normally closed to ensure detection of wire breaks or contact faults. - Dual-channel wiring enhances fault tolerance in safety circuits. - PLC programs should latch the stop state until manual reset occurs. - Use a safety-rated relay or safety PLC input for critical machine functions.

4.3.7 Self-Check Questions

No.	Question
1	According to ISO 13850, what is the primary functional requirement for an Emergency Stop (E-Stop) function?
2	What type of contact configuration is mandated for E-Stop buttons and why is this configuration considered fail-safe?
3	Describe the two critical actions that must happen after an E-Stop has been actuated and then released, as required by safety standards.
4	In the electrical control circuit for an electro-pneumatic sequence (using relays or PLC logic), where must the E-Stop signal be integrated to ensure all outputs are safely stopped?
5	What distinguishes Stop Category 0 from Stop Category 1 according to IEC 60204-1?
6	Why do safety standards require dual-line (redundant) wiring for E-Stop circuits?
7	What standard defines the principles for reference designations (e.g., -Y1 for a solenoid) in electro-pneumatic circuit diagrams?
8	In an electro-pneumatic sequence involving double solenoid valves, what is the purpose of the Reset Step (e.g., Step 5) described in the A+B+A-B- sequence?
9	How is the Continuous Cycle Mode typically enabled in a relay-based sequence controller?
10	Which type of cylinder returns to its initial position by means of an internal spring when the air supply is exhausted?

4.3.8 Answer Key

No.	Answer / Guidance
1	The E-Stop function must be available and operational at all times, overriding all other functions and operating modes of the machine to avert actual or impending emergency situations.
2	E-Stop buttons must use Normally Closed (NC) contacts. This is fail-safe because an open circuit (caused by a wire break or the button press) is interpreted as an active stop command.
3	First, the function must be maintained until manually reset (unlatching). Second, resetting the device shall not automatically restart the machine; a separate, deliberate action (Start or Reset button) is required to resume operation.
4	The E-Stop (EMO) signal must be placed in series with all Set circuits except the END step to ensure all active outputs are immediately de-energized.
5	Stop Category 0 involves the immediate removal of power from actuators (uncontrolled stop/STO). Stop Category 1 involves a controlled deceleration to stop, followed by power removal (STO) once the stop is achieved (Safe Stop 1).
6	Dual-line (redundant) wiring ensures the system remains fail-safe and functional even if a single contact or wire fails (fault tolerance).
7	DIN EN 81346-2 defines the structuring principles and reference designations for industrial systems.
8	The Reset Step (e.g., Step 5) is implemented to ensure all previous steps are cleared (unlatched) after one complete cycle, preparing the sequence for a reliable restart.
9	Continuous Cycle Mode is activated when a control relay (e.g., -KF1) is energized, enabling a self-latching circuit that ensures the sequence automatically restarts upon completion of the last step.
10	A Single-Acting Cylinder (SAC) performs work in one direction using compressed air and returns the piston rod using a built-in spring when air is exhausted.

4.4 Learning Outcome: Generate PLC Program for Sequence Control

Objective: To recognize and interpret the standards, symbols, and fundamental components (both electrical and pneumatic) necessary for understanding robot and electropneumatic circuit diagrams, including the basic components in robots and their control systems.

Learning Activity: Study the specific industry standards (ISO 1219-1 and DIN EN 81346-2) applied in pneumatic and electrical circuit diagram representation; identify and describe the symbols for directional control valves, actuators, and logical pneumatic elements; recognize and define basic electrical control components (e.g., relays, limit switches, solenoids); and analyze how electrical and pneumatic systems are integrated to form control systems.

4.4.1 Information Sheet: Introducing MPS station components, module, elements

Festo Modular Production System (MPS) Structure

The Festo Modular Production System (MPS) is designed so that each station can run independently for a specific manufacturing process or be integrated with other stations to form a comprehensive system. The system uses a modular design for step-by-step learning approaches.

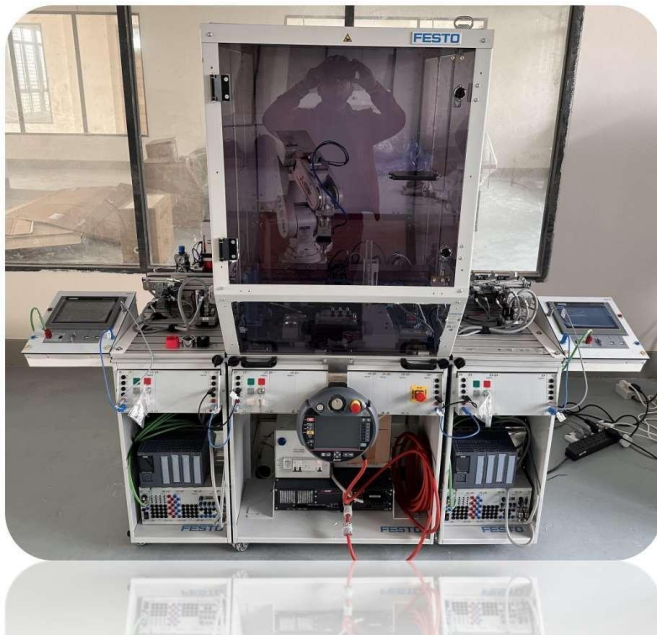


Figure 4.50: Festo Modular Production System (MPS) structure



Figure 4.51: MPS Robot Station

High-Level Structure

MPS Station → Composed of Modules → Composed of Elements

Key MPS Stations and Modules

- **1. Distributing/Conveyor Station**

- **Conveyor Module:** Transports and separates workpieces from workpiece sets (e.g., 40 mm diameter basic body or cylinder assembly sets).

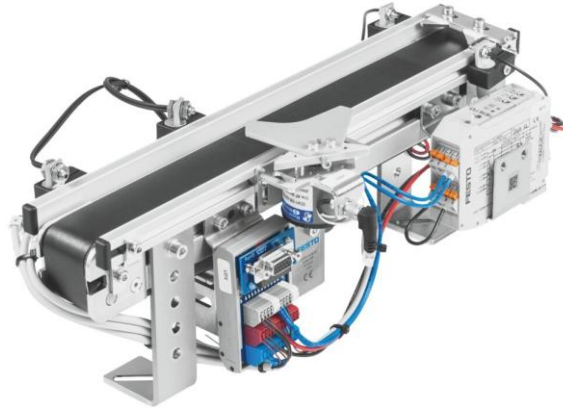


Figure 4.52: Conveyor module

- **Stack Magazine Module:** Separates workpieces stored in a magazine tube. A double-acting cylinder pushes out workpieces one at a time.

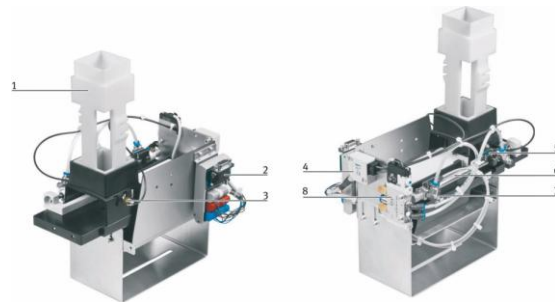


Figure 4.53: Stack Magazine Module



Figure 4.54: Formation of Distributing/Conveyor Station

- **2. Sorting Station**

- **Conveyor Module:** As above.

- **Detection Module:** Capable of verifying the presence of red, black, and metallic workpieces using 3 proximity sensors with digital output to detect material or color.

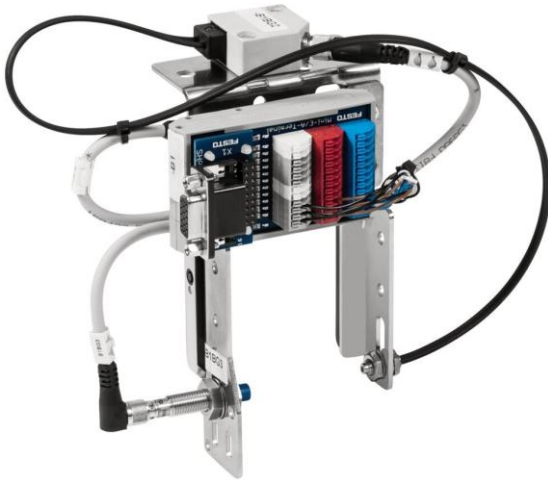


Figure 4.55: Detection Module

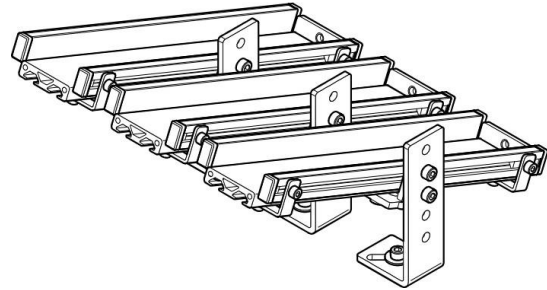


Figure 4.56: Chute for storing workpieces

- **Chute:** Used for transporting or storing workpieces.

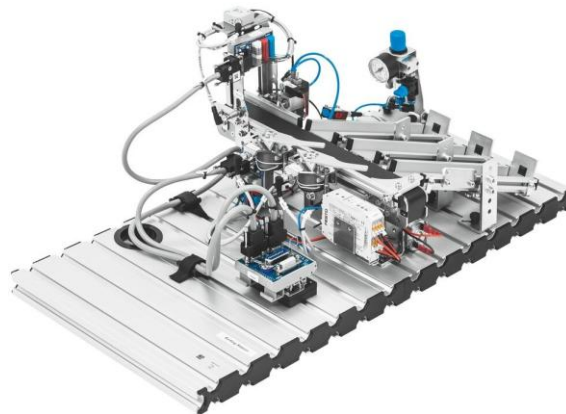
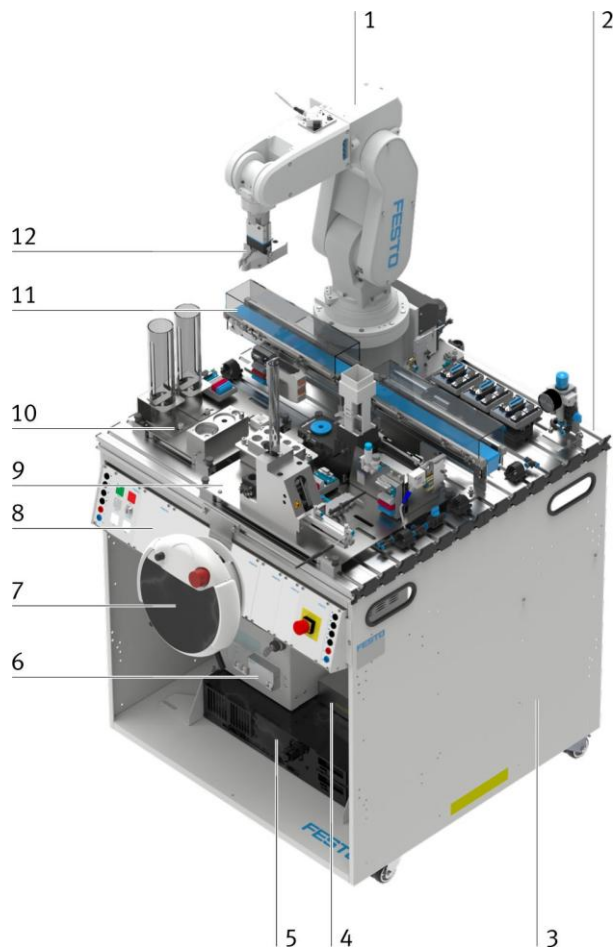


Figure 4.57: Sorting Station

• 3. Robot Station

- **Robot Handling Module:** Includes positions for Cylinder Assembly, Cylinder Color & Orientation Check, and a Magazine for bad products.
- **Robot Assembly Module:** Contains magazines for Caps and Springs, as well as positions for Cap Pickup, Spring Pickup, and a Piston Pallet.



Label Identification:

1. Robot arm RV-2FRB-D-S25
2. Profile plate
3. Carriage
4. Signal distributor (RIA box)
5. Control unit CR800-O2VP-S25 with I/O interface 2D-TZ378
6. Supply module
7. Teaching Box R56TB
8. Operator panel
9. Robot assembly module
10. Robot handling module
11. Conveyor belt
12. Multifunction gripper

Figure 4.58: Components of Robot Station



Figure 4.59: Robot Handling Module

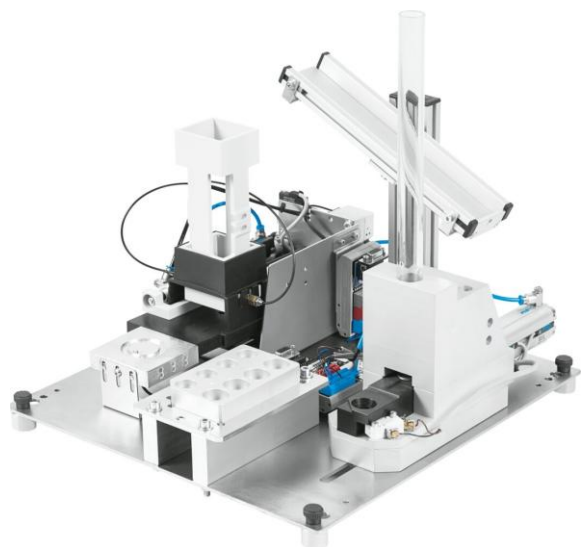


Figure 4.60: Robot Assembly Module

4.4.2 Information Sheet - Distributing Station (via SysLink I/O Table) Pin Assignment

The Festo Modular Production System (MPS) Distributing Station utilizes a SysLink interface for communication between the station's components and the control system. The following table outlines the pin assignments and signal functions for the Distributing Station's Conveyor Module.

Table 4.17: SysLink Pin Assignment and Signal Functions for Conveyor Module

Function	SysLink Pin	Color Code	Designation / Description
I0	13	Grey-pink	Workpiece at beginning of conveyor
I1	14	Red-blue	Workpiece in middle of conveyor
I2	15	White-green	No workpiece at end of conveyor
I3	16	Brown-green	—
I4	17	White-green	Slide retracted
I5	18	Brown-yellow	Slide advanced
I6	19	White-yellow	Magazine empty
I7	20	Grey-brown	—
Q0	1	White	Conveyor forward
Q1	2	Brown	Conveyor reverse
Q2	3	Green	Advance feed separator
Q3	4	Yellow	—
Q4	5	Grey	Advance slide
Q5	6	Pink	—
Q6	7	Blue	—
Q7	8	Red	—
24 V A	9+10	Black	24 V power supply for outputs
24 V B	21+22	White-pink	24 V power supply for inputs
GND A	11	Brown-pink	0 V power supply for outputs
GND A	12	Purple	0 V power supply for outputs
GND B	23+24	White-blue	0 V power supply for inputs

4.4.3 Information Sheet - Sorting Station (via SysLink I/O Table) Pin Assignment

The Festo Modular Production System (MPS) Sorting Station utilizes a SysLink interface for communication between the station's components and the control system. The following table outlines the pin assignments and signal functions for the Sorting Station's Conveyor Module.

Table 4.18: SysLink Pin Assignment and Signal Functions

Function	SysLink Pin	Color Code	Designation / Description
I0	13	Gray-pink	Workpiece at beginning of belt
I1	14	Red-blue	Deflector 1 advanced
I2	15	White-green	Chute full
I3	16	Brown-green	Deflector 2 advanced
I4	17	White-green	Workpiece detected
I5	18	Brown-yellow	Workpiece not black
I6	19	White-yellow	Chrome-plated workpiece
I7	20	Gray-brown	—
Q0	1	White	Conveyor belt forward
Q1	2	Brown	Advance deflector 1
Q2	3	Green	Advance deflector 2
Q3	4	Yellow	Retract stopper
Q4	5	Gray	—
Q5	6	Pink	—
Q6	7	Blue	—
Q7	8	Red	—
24V A	9+10	Black	24 V power supply for outputs
24V B	21+22	White-pink	24 V power supply for inputs
GND A	11	Brown-pink	0 V power supply for outputs
GND A	12	Purple	0 V power supply for outputs
GND B	23+24	White-blue	0 V power supply for inputs

4.4.4 Information Sheet: MPS electrical components circuit labels and pin mapping with PLC I/O

Mini I/O Terminal The Mini I/O Terminal serves as an external electrical interface, often connecting sensors and actuators of a module (like the stacking magazine module) to the larger control system. It uses D-Sub 15 HD connectors (X1) and typically includes terminals labeled X1, X2, X3, and X4. It facilitates the routing of digital I/O signals. Mini I/O terminals in MPS may have reference designations such as:

-G1XG1, -B1XG1, -C2XG1, etc.

The designation structure can be interpreted as follows:

Element	Description
XG1	Mini I/O terminal of module G1.
G	Represents the <i>Conveyor Module (Band)</i> .
1	Indicates the module number (Module 1).
XG1	Connector block number 1 inside that module.

Table 4.19: Interpretation of Mini I/O terminal reference code

Therefore, the reference designation:

-G1XG1

means “**the Mini I/O terminal connector XG1 of Conveyor Module G1.**”

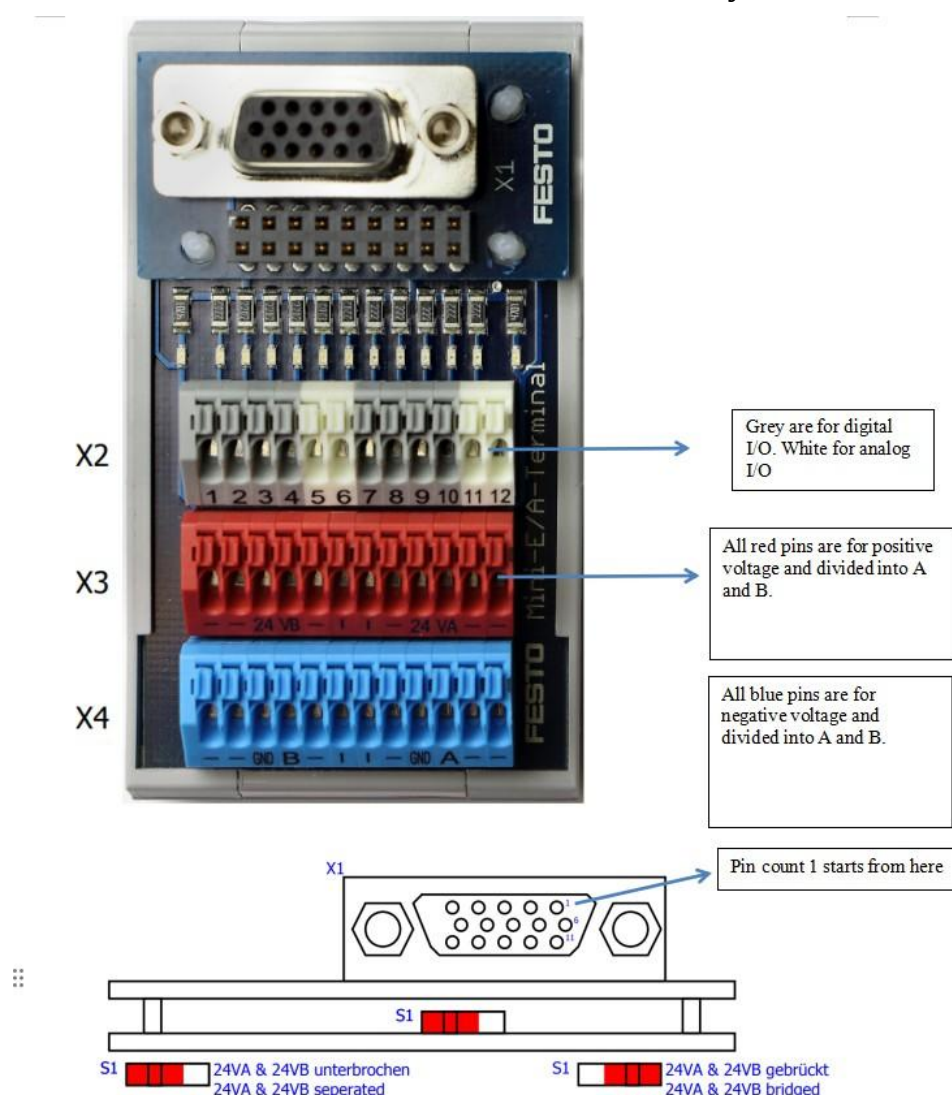


Figure 4.61: Mini I/O Terminal

C-Interface The C interface is laid out so that two modules can be easily connected to a PLC via SysLink. The C-Interface (e.g., -V1XG1) acts as the connection point between MPS modules and the PLC, often using a 15-pin D-Sub-HD socket (X1, X2)

4.4 Learning Outcome: Generate PLC Program for Sequence Control

, one 24-pin centronics socket (X3) and one 15 pin D-Sub socket (X4). This connector handles digital I/O (I0-I3, Q0-Q3), analog I/O (AI0, AI1, AQ0), and segregated power supplies (24VA/GND A and 24VB/GND B).

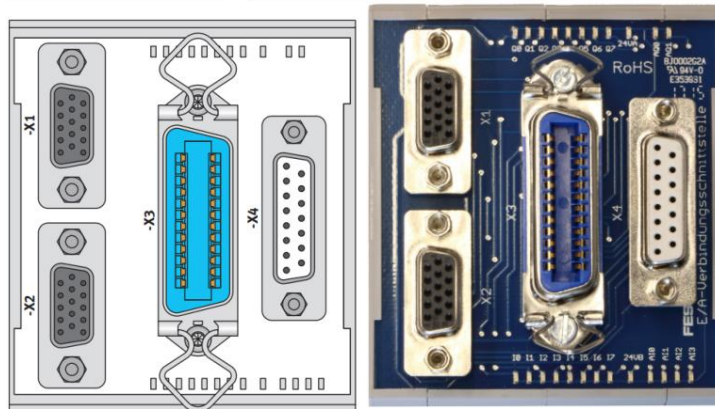


Figure 4.62: C-Interface

Color Codes of Cables Used

Cable color codes ensure proper wiring and identification in 24V DC systems.

Festo Didactic Safety Cable (General)

Color	Meaning
Red	24 V DC (Positive supply voltage)
Blue	0 V DC (Negative/GND)
Black/White	Signal input/output (SELV)
Green-Yellow	Protective ground conductor

SysLink (IEEE 488) I/O Cable Colors (Festo)

Pin No.	Color Designation	Function
1	White	Output 0
2	Brown	Output 1
7	Blue	Output 6
8	Red	Output 7
13	Gray-pink	Input 0
14	Red-blue	Input 1
15	White-green	Input 2
17	White-yellow	Input 4
9+10	Black	24V power for outputs (24V A)
21+22	White-pink	24V power for inputs (24V B)

Note: Pins 5-11 are typically dedicated to other I/O signals, including analog I/O (AI0, AI1, AQ0) and additional digital inputs (I0-I3).

Module-4: Perform Electro-pneumatic Sequence Control

Table 4.22: Example 15-Pin D-Sub HD Pinout (C-Interface - V1XG1 / Distribution Station C-Interface - C1XG1)

Pin No.	Color Code	Function	Purpose
1	White (wh)	Q0	Digital Output 0
2	Brown (bn)	Q1	Digital Output 1
3	Green (gn)	Q2	Digital Output 2
4	Yellow (ye)	Q3	Digital Output 3
12	Red-blue (rd-bl)	24VA	24 V DC for Group A components (e.g., conveyor)
13	White-green (wh-gn)	24VB	24 V DC for Group B components (e.g., magazine)
14	Brown-green (bn-gn)	GND A	Ground for Group A
15	White-yellow (wh-ye)	GND B	Ground for Group B

Robot Interface Box (RIA Box) The Robot Interface Box (RIA box) facilitates connection between the robot control unit and peripherals. It uses Centronics connectors:



Figure 4.63: Robot Interface Box (RIA Box)

- One 50-pin (X0) Centronics socket for connecting the robot control unit.
- Four 24-pin (X1-X4) Centronics sockets for connecting peripherals.

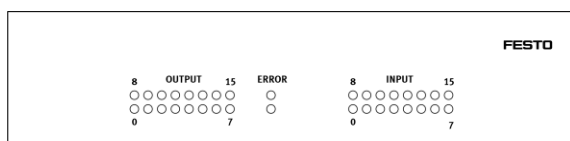


Figure 4.64: RIA box - Front View

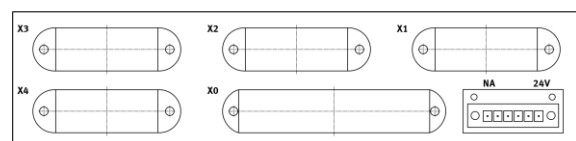


Figure 4.65: RIA box - Rear View

The X0 connector transmits all 16 robot inputs (I0 to I15) and 16 robot outputs (Q0 to Q15). For example, robot input I0 is mapped to pin 10 and robot output Q0 is mapped to pin 4.

Field Device Pin Mapping and PLC/Controller I/O Allocation

Field devices (Input Devices like sensors, Output Devices like solenoid coils/motors) communicate with the controller (PLC or Robot Controller) via these intermediate connectors (Mini I/O, C-Interface/SysLink, RIA Box).

Example: Robot Station I/O Mapping via RIA Box The Robot Interface Box (RIA Box) centralizes inputs (I0-I15) and outputs (Q0-Q15) before connecting to the

Table 4.23: Connection (Robot Interface Box)

Pin Type	Purpose
X0	50-pin Centronics socket - Connects to the robot controller
X1-X4	24-pin Centronics sockets - Connect peripherals (e.g., handling module, operator panel, assembly module, belt)

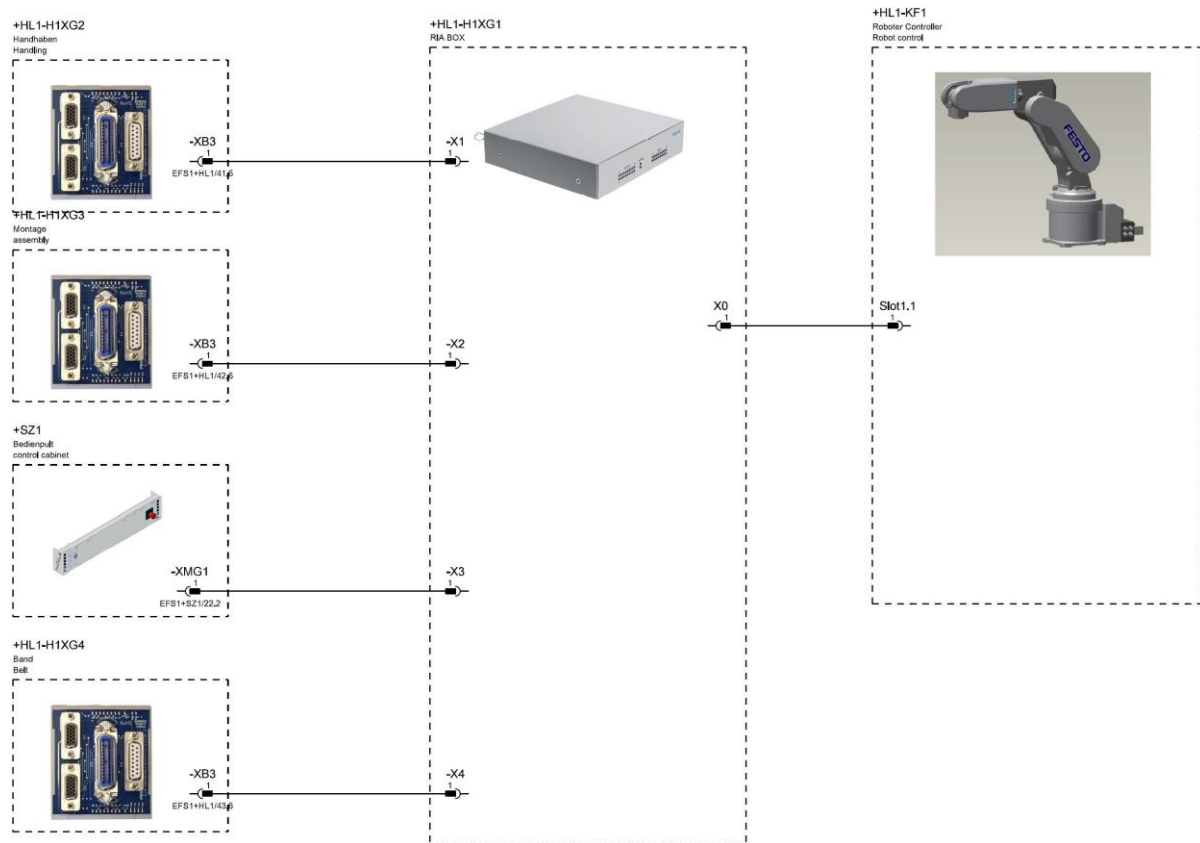


Figure 4.66: MPS Station and Robot I/O connections via RIA Box

Mitsubishi Robot Controller (X0 connector). The RIA Box links various peripherals via X1, X2, X3, and X4 connectors (Assembly, Handling, Operator Panel, Belt):

- **Robot Controller Interface (X0):** This is the main connection for I/O signals entering/leaving the controller (PLC equivalent), supporting up to 16 inputs (I0- I15) and 16 outputs (Q0-Q15).
- **Peripherals Connection (X1 to X4):** These connectors route signals from the physical modules to specific I/O lines identified at X0. For example:
 - Signals connected to X1 (Assembly module) pins 13, 14, 15, 16 might be mapped to Controller Inputs I1, I2, I6, I11 at X0.
 - Output Q0 (Conveyor belt forward/reverse) and Q1 (Conveyor creep speed) for the Conveyor Module (Belt) are provided via the RIA box output connection for the Belt (X4).

The Miniature I/O terminals and C-Interfaces on individual modules (like the Distributing/Conveyor Station or Sorting Station) translate signals from proximity sensors

Module-4: Perform Electro-pneumatic Sequence Control

Table 4.24: Input Pin Mapping for Modules X0-X4

X0		X1		X2		X3		X4	
(Controller)		(Assembly)		(Handling)		(Operator panel)		(Belt)	
Input	Pin	Input	Pin	Input	Pin	Input	Pin	Input	Pin
I0	15	n.c.		n.c.		n.c.		n.c.	
I1	16							I0	13
I2	17					I0	13		
I3	18					I1	14		
I4	19					I3	16		
I5	20							I1	14
I6	21			I7	20				
I7	22								
I8	40	I0	13						
I9	41	I1	14						
I10	42	I2	15						
I11	43	I4	17			I2	15		
I12	44	I5	18						
I13	45	I6	19						
I14	46							I3	16
I15	47								

(e.g., magnetic proximity sensors, through-beam sensors) and solenoid valves into standardized digital signals (logic 1 or logic 0) before they are passed along via SysLink cabling to the main controller or the RIA Box.

4.4.5 Information Sheet: DC Motor Controller Pinout

The DC Motor Controller functions as the interface between the control system (like a PLC or simulation box) and the conveyor's 24 V DC gear motor, converting digital and analog signals into movement commands. Key specifications for this controller include controlling 24 V motors up to a continuous current of 4 A. The recommended wire cross-section for connections is 0.5 to 1.0 mm².

DC Motor Controller Terminal Allocations (Pinout)

The controller utilizes 16 terminals grouped for power, motor connection, digital control inputs, and analog signaling.

4.4 Learning Outcome: Generate PLC Program for Sequence Control

Table 4.25: Output Pin Mapping for Modules X0-X4

X0		X1		X2		X3		X4	
(Controller)		(Assembly)		(Handling)		(Operator panel)		(Belt)	
Output	Pin	Output	Pin	Output	Pin	Output	Pin	Output	Pin
Q0	4					Q0	1		
Q1	5					Q1	2		
Q2	6					Q2	3		
Q3	7					Q3	4		
Q4	29					Q4	5		
Q5	30							Q0	1
Q6	31							Q1	2
Q7	32							Q2	3
Q8	10	Q0	1						
Q9	11	Q1	2						
Q10	12	Q2	3						
Q11	13					Q3	4		
Q12	35	Q4	5						
Q13	36	Q5	6						
Q14	37	Q6	7						
Q15	38	Q7	8						



(a) Front View



(b) Rear view

Figure 4.67: DC Motor Controller

Table 4.26: Robot Controller Inputs Used

Signal	Designation	E-plan Designation	Contact Signal Distributor (RIA Box)	Description
0	–	–	–	Not used
1	m_SensOriCheck	C3BG1	RIA1-X2-I0	Sensor for checking the orientation of the workpieces on the assembly device.
2	m_SensBeltFront	C3BG2	RIA1-X2-I1	Sensor at the start of the belt.
3	m_StartButton	–	RIA1-X3-I0	Start button on the MPS control panel.
4	m_StopButton	–	RIA1-X3-I1	Stop button (opener) on the MPS control panel.
5	m_ResetButton	–	RIA1-X3-I3	Reset button on the MPS control panel.
6	m_SensBeltMid1	–	–	Not used.
7	m_NextStatFree	–	RIA1-X3-I7	Laboratory socket I7 - release of the following station.
8	m_SprRetracted	C1BG1	RIA1-X1-I0	Ejector end position sensor retracted on the spring magazine.
9	m_SprExtended	C1BG2	RIA1-X1-I1	Ejector end position sensor extended on the spring magazine.
A	m_SprAtPickup	C1BG3	RIA1-X1-I2	Spring lever switch present in removal position.
B	m_SensBeltMid2	–	–	Not used.
C	m_CoverRetracted	C2BG1	RIA1-X1-I4	Ejector end position sensor retracted on the cover magazine.
D	m_CoverExtended	C2BG2	RIA1-X1-I5	Ejector end position sensor extended on the cover magazine.
E	m_CoverAtPickup	C2BG3	RIA1-X1-I7	Diffuse sensor - cover available in removal position.
F	m_SensBeltEnd	–	RIA1-X1-I6	Sensor at the belt end.
30	m_IsSimMode	–	–	To distinguish whether the program is running on a real robot or in the Ciro simulation.

Table 4.30: DC Motor Controller Terminal Allocation and Function

Terminal Function		Description / Signal Details
Power Supply		
14	+24 V DC In	Main supply voltage, accepts $\pm 10\%$ variation.
13, 15	GND	0 V connection.
Motor Connections		
<i>(Continued on next page)</i>		

4.4 Learning Outcome: Generate PLC Program for Sequence Control

Terminal Function		Description / Signal Details
10	Motor connection +	Connects to the positive terminal of the DC motor.
9	Motor connection -	Connects to the negative terminal of the DC motor.
Digital Control Inputs (P-Switching)		
2	Clockwise rotation	Digital input to start forward movement (e.g., Q0: Conveyor forward).
1	Counterclockwise rotation	Digital input to start reverse movement (e.g., Q1: Conveyor reverse).
4	Creep speed	Digital input; when active, the set speed is halved.
12	Enable Clockwise/Acknowledge	Used for safety/limit switch detection (P-switching).
11	Enable Counterclockwise/Acknowledge	Used for safety/limit switch detection (P-switching).
Analog & Auxiliary		
6	Analog input	0-12 V input for external speed control. >11 V uses internal potentiometer (TR3).
3	GND for external potentiometer	0 V reference for analog input (T6), max. 0.5 A.
5	Ready for operation	Digital output, High active (24 V / 0.7 A).
8	+24 V Auxiliary Out	Auxiliary voltage output, max. 0.5 A.
7	+10 V Auxiliary Out	Auxiliary output, approx. 50 mA (PTC fuse protected).
16	+24 V Out	Auxiliary +24 V output.

Interpretation of Control Logic and Circuit Operation

P-Switching (Input Activation): The digital inputs (Terminals 1, 2, 4, 11, 12) operate using P potential switching (also known as sourcing logic).

- **Definition:** P-switching means that to achieve a logical ‘1’ (ON state), the input terminal must be connected to the positive supply voltage (+24 V DC).
- **Compatibility:** This structure requires the external switching device (like a PLC output or a control console pushbutton) to supply, or “source,” the positive voltage to the input pin to activate the function.

Motor Control Functions:

- **Direction Control:** Rotation is initiated by providing a continuous Logic 1 signal to either Terminal 2 (Clockwise/Forward) or Terminal 1 (Counterclockwise/Reverse).

- **Speed Selection:** Speed is determined primarily by the analog input (Terminal 6).

4.4 Learning Outcome: Generate PLC Program for Sequence Control

Table 4.27: Robot Controller Outputs Used

Signal	Designation	E-plan Designation	Contact Signal Distributor (RIA Box)	Description
0	m_LedStart	–	RIA1-X3-Q0	LED of the start button on the MPS control panel.
1	m_LedReset	–	RIA1-X3-Q1	LED of the reset button on the MPS control panel.
2	m_LedQ1	–	RIA1-X3-Q2	Q1 light on the MPS control panel.
3	m_LedQ2	–	RIA1-X3-Q3	Q2 light on the MPS control panel.
4	m_StationFreeQ4	–	RIA1-X3-Q4	Q4 laboratory socket - release to previous station.
5	m_BeltFwd	–	RIA1-X2-Q0	Not used.
6	m_BeltSlow	–	RIA1-X2-Q1	Not used.
7	–	–	RIA1-X2-Q2	Not used.
8	m_SpringExtend	C1MB1	RIA1-X1-Q0	Extend the spring magazine ejector valve.
9	–	–	RIA1-X1-Q1	Not used.
A	–	–	RIA1-X1-Q2	Not used.
B	–	–	RIA1-X1-Q3	Not used.
C	m_CoverExtend	C2MB1	RIA1-X1-Q4	Extend valve ejector on the cover magazine.
D	–	–	RIA1-X1-Q5	Not used.
E	–	–	RIA1-X1-Q6	Not used.
F	–	–	RIA1-X1-Q7	Not used.

Table 4.28: Contact Assignment Table (Hand Sensor Connector CON1H)

Type	Function	Pin	Color	Signal Designation
Digital	HC1	A1 (7)	Black	900 - Part is not black
	HC2	A2	–	901 -
	HC3	A3	–	902 -
	HC4	A4	–	903 -
	–	A5	–	Reserved
	–	A6	–	Reserved
–	–	B1	–	Reserved
	–	B2	–	Reserved
	–	B3 (3)	Brown	24 V Supply
	–	B4 (4)	Blue	0 V Supply
	–	B5	–	Reserved
	–	B6	–	Reserved

- An external voltage source (0-10 V DC) between Terminal 6 and Terminal 3 (GND) sets speed proportionally (0-100%).
- If the input voltage exceeds 11-11.5 V, the controller defaults to the internal potentiometer (TR3) setting.

Table 4.29: Contact Assignment Table (Solenoid Valve Connection)

Type	Function	Connection	Color	Signal Designation
Digital	MB1 +	GR1-1	Black	900 - Open gripper
	MB1 –	GR1-2	Black	0 V
	MB2 +	GR2-1	Black	901 - Close gripper
	MB2 –	GR2-2	Black	0 V

- **Quick Stop:** Triggered when both Clockwise (T2) and Counterclockwise (T1) inputs simultaneously receive Logic 1, immediately connecting motor windings (T9, T10) to GND for a rapid stop.
- **Creep Speed:** Activation of Terminal 4 halves the active speed setpoint (analog or internal).

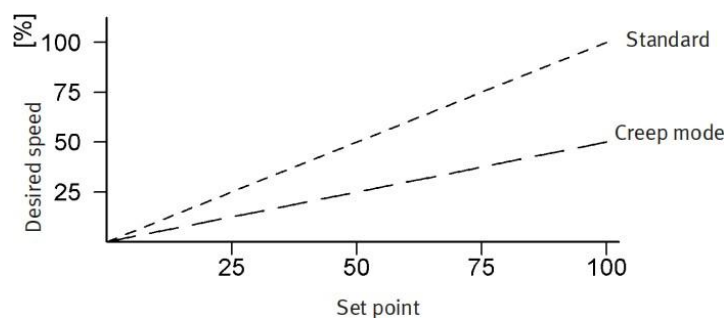


Figure 4.68: DC motor creep speed

This integration of digital inputs for directional control and analog input for speed modulation allows the motor controller to manage high starting currents effectively, preventing PSU overloads.

4.4.6 Information Sheet: Mini I/O and Centronics Terminal Circuit Interpretation

Overview The Festo Mini I/O Terminal represents a **passive I/O terminal block** (breakout board) that simplifies wiring between a controller (such as a PLC) and field devices. The multi-pin connector **X1 (D-Sub)** receives bundled signals from the controller, while terminals **X2, X3, and X4** provide accessible screw terminals for connecting sensors and actuators.

4.4 Learning Outcome: Generate PLC Program for Sequence Control

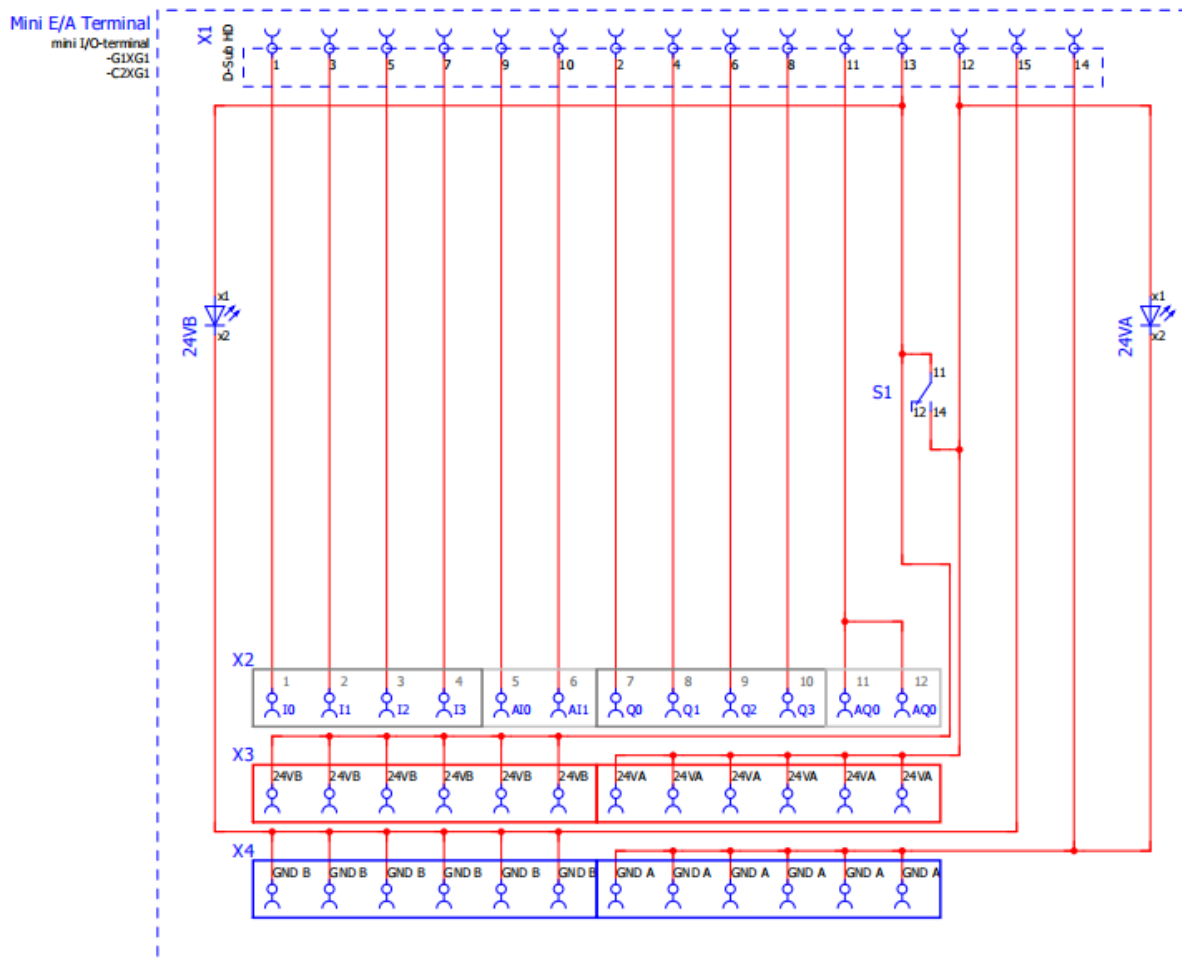


Figure 4.69: Festo Mini I/O Terminal circuit layout and connection designations

Connectors and Terminal Designations The designations comply with **IEC 61346** and **ISO 1219-1** standards. Each label beginning with “X” indicates a connector or terminal block.

- **X1 (D-Sub HD)** - Main 15-pin D-Sub connector linking the terminal block to the controller through a multi-core cable.
- **X2 (I/O Terminal)** - Terminal strip for all process I/O signals; connects sensors, switches, and actuators.
- **X3 (+24V Supply)** - Terminal strip distributing the positive 24 V DC supply.
- **X4 (0V / GND)** - Terminal strip providing the 0 V or common ground reference.

Power Supply Grouping The terminal uses two isolated power groups to reduce electrical noise and ensure safe operation of mixed signal types.

Group B - Inputs Terminals 1-5 on **X3** and **X4** are powered by **24VB / GND B**. These supply the input channels (**I0-I3**) and the analog input (**AI0**). Power enters via **X1 pin 1** (24VB), with the left blue LED indicating voltage presence.

Group A - Outputs Terminals 6-10 on **X3** and **X4** use **24VA / GND A**. These power the output channels (**Q0-Q3**) and analog output (**AQ0**). Power is supplied via **X1 pin 15 (24VA)** and indicated by the right blue LED.

This isolation (galvanic separation) minimizes interference between high-current output devices and low-level input signals.

I/O Signal Mapping (Terminal X2) The signal designations comply with **IEC 61131-2**:

- **I** - Digital Input
- **Q** - Digital Output
- **A** - Analog Input/Output

Table 4.31: Pin Mapping between X2 Field Terminals and X1 D-Sub Connector

X2 Terminal	Signal Function	Connected to X1 Pin
I0	Digital Input 0	2
I1	Digital Input 1	3
I2	Digital Input 2	4
I3	Digital Input 3	5
AI0	Analog Input 0	6
Q0	Digital Output 0	7
Q1	Digital Output 1	8
Q2	Digital Output 2	9
Q3	Digital Output 3	10
AQ0	Analog Output 0	11

Special Components - Switch S1 The switch **S1** connects or disconnects the analog output path. It is used for selecting analog output configurations.

- **X2 Pin 11 (AQ0)** → routed via **S1** → **X1 Pin 12**
- **X2 Pin 12 (AQ0 Return)** → routed via **S1** → **X1 Pin 13**

This allows configuration of the analog channel as either:

- Voltage output (0-10 V),
- Current output (4-20 mA), or
- Shielded differential output.

S1 thus acts as a selector for analog output mode and isolation.

4.4 Learning Outcome: Generate PLC Program for Sequence Control

Example Field Device Connection A field sensor is connected to the Mini I/O Terminal via the screw terminal:

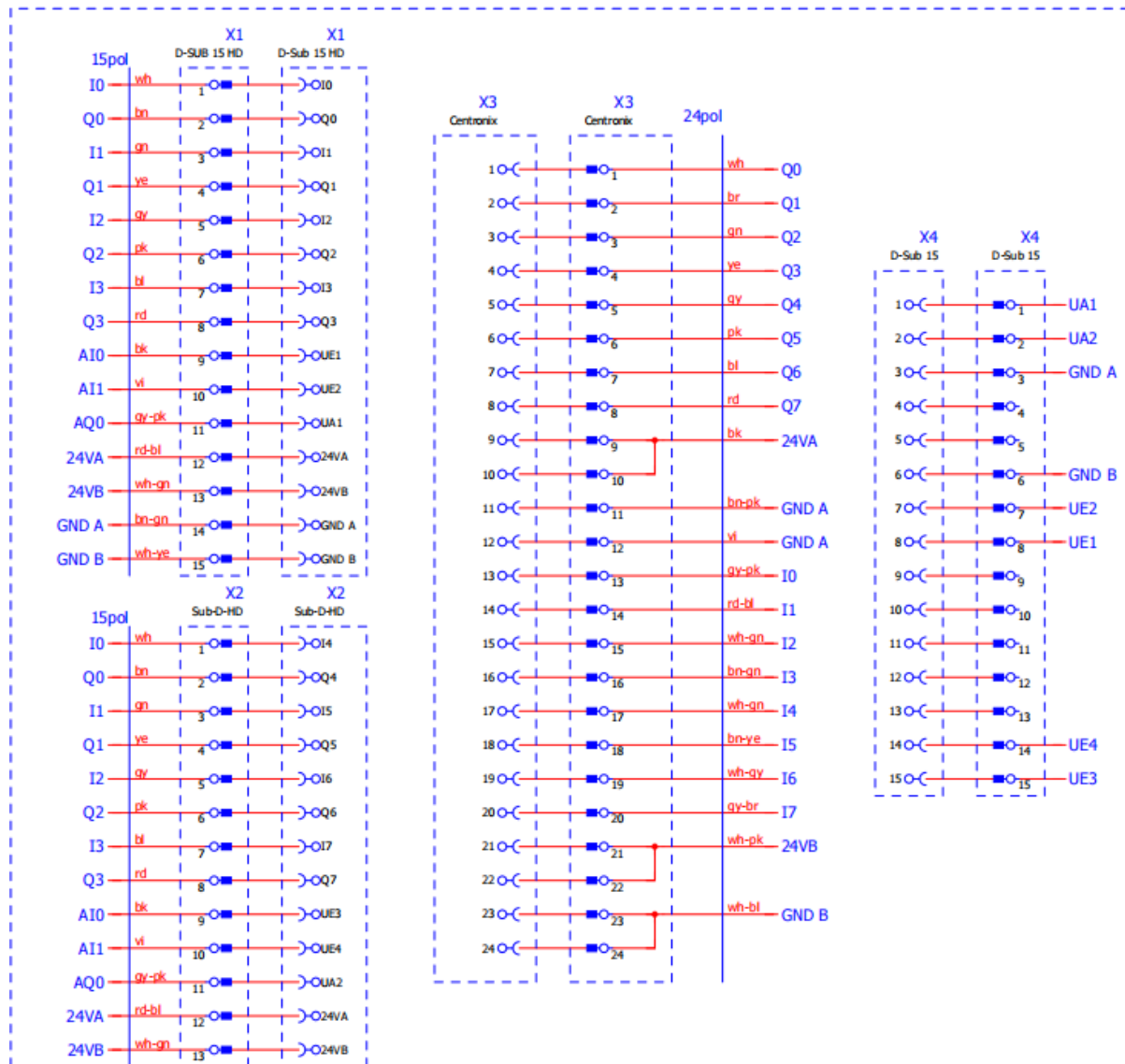


Figure 4.70: Centronics board circuit layout and connection designations

- Terminal: **X2, Pin 2**
- Signal Label: **I1 (Digital Input 1)**

This represents the sensor's output signal being wired into the I/O terminal's input channel.

Internal Wiring within Mini I/O Terminal Internally, the signal from terminal **X2 (Pin 2)** is hardwired to **X1 (Pin 3)** on the D-Sub 15 HD connector.

Mini I/O to Centronics Board via D-Sub Cable Connection The **D-Sub 15 cable** connects the Mini I/O Terminal's **X1** port to the **Centronics Terminal Break-out Board**. The board features two D-Sub ports (**X1** and **X2**).

- Plug the D-Sub cable into the adapter's **X1 port** → signal goes to PLC input **I1**.

Module-4: Perform Electro-pneumatic Sequence Control

- Plug the same cable into the adapter's **X2 port** → signal is rerouted to PLC input **I5**.

5. Final Connection to PLC The **X3 Centronics plug** (24-pin connector) on the adapter cable is plugged directly into the PLC's I/O interface. This connector carries all selected sensor and actuator signals to the PLC based on the chosen configuration path (**I1** or **I5**).

4.4.7 Information Sheet: Sensor Circuit Interpretation

Circuit Overview This circuit illustrates the physical wiring of three **input sensors** to the **Festo Mini I/O Terminal**. The terminal acts as a passive distribution block that supplies 24 V DC power to the sensors and transmits their signal outputs to the PLC. It simplifies field wiring by providing labeled terminal strips and internal D-Sub pin mapping.

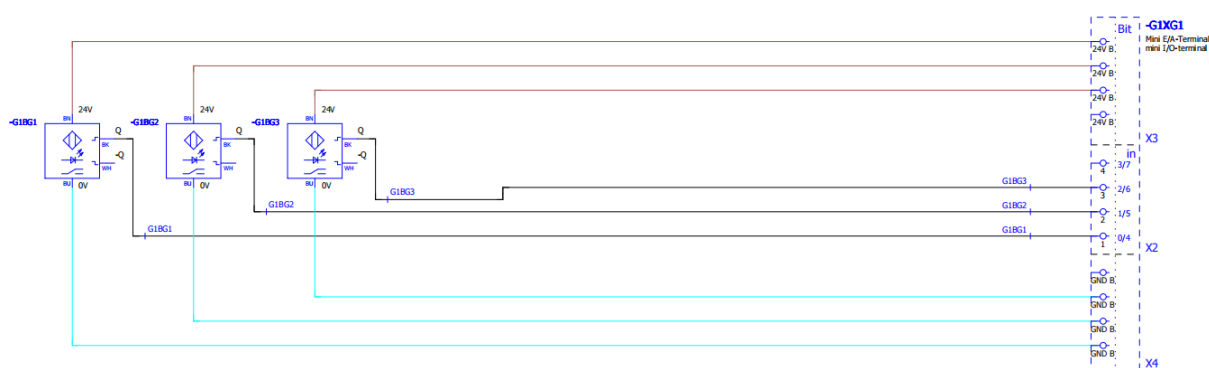


Figure 4.71: Connection layout of three sensors to Festo Mini I/O Terminal block

1. Components

- **Sensors:** **-G1BG1**, **-G1BG2**, and **-G1BG3** are 3-wire DC sensors (e.g., photo-electric or inductive proximity). Each follows the ISO reference format:
-G1BG3 ⇒ Sensor BG3 in main module G1 (e.g., Conveyor Module 1)
- **Terminal Block:** **-G1XG1** denotes the Festo Mini I/O Terminal. It serves as a passive breakout interface with three key terminal strips:
 - **X2** - Signal I/O terminals
 - **X3** - 24 V DC supply distribution
 - **X4** - 0 V (GND) common return

2. Power Distribution The sensors receive operating voltage directly from the Mini I/O Terminal.

- **+24V Supply:** Distributed through terminal strip **X3 (24V B)**. A red wire connects this to each sensor's **BN (Brown)** wire.
- **0V Supply (Ground):** Distributed through terminal strip **X4 (GND B)**. A blue wire connects this to each sensor's **BU (Blue)** wire.

4.4 Learning Outcome: Generate PLC Program for Sequence Control

This ensures all sensors share a common reference voltage while maintaining noise isolation between input and output groups.

3. Signal Path (Example: Sensor -G1BG3) The following steps trace the signal of sensor -G1BG3 from the field device to the PLC input.

1. **Field Device Connection:** The sensor's signal wire (**BK - Black**) is connected to **X2, Pin 3**. The diagram also labels this as terminal "3/6", denoting its physical position on the X2 block.
2. **Internal Terminal Mapping:** Inside the Mini I/O Terminal, **X2 Pin 3** is internally routed to **Pin 5** of the **X1 D-Sub 15** connector.
3. **Adapter Cable and PLC Input:** The D-Sub 15 connector links to a Centronics adapter cable, which determines the final PLC input address:
 - **Path A:** Cable connected to adapter port **X1** → Signal routed to PLC input **I2**.
 - **Path B:** Cable connected to adapter port **X2** → Signal rerouted to PLC input **I6**.

Thus, the same field signal can be dynamically assigned to different PLC input addresses without rewiring—useful during trainer reconfiguration or module swapping.

Table 4.32: Mapping of Sensor Outputs from Mini I/O Terminal to PLC Inputs

Sensor ID	Signal Wire (BK)	Terminal on X2	PLC Input Mapping
-G1BG1	Black (Signal)	X2 Pin 1	I0 / I4
-G1BG2	Black (Signal)	X2 Pin 2	I1 / I5
-G1BG3	Black (Signal)	X2 Pin 3	I2 / I6

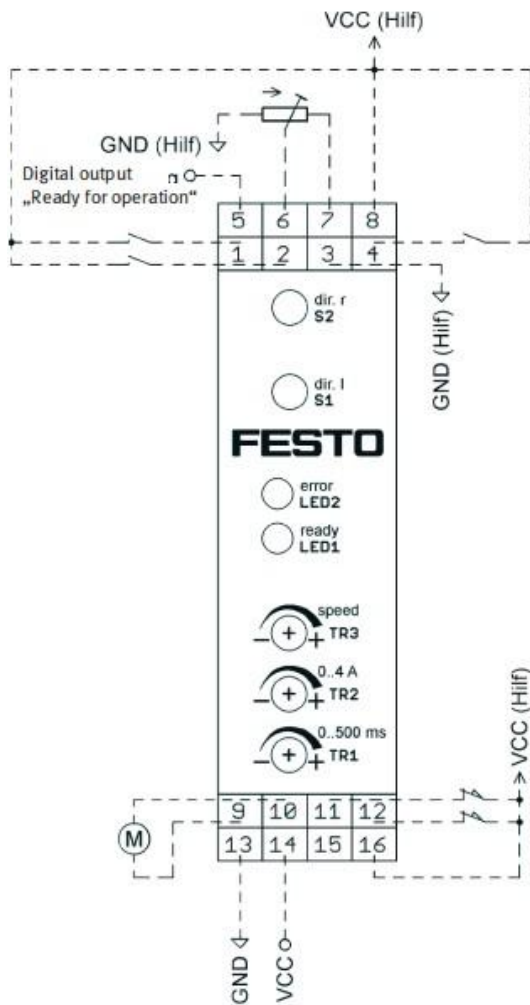
4.4.8 Information Sheet - Motor and Actuator Output Wiring Circuit

Circuit Overview This circuit illustrates the **output wiring** for two actuators controlled by the main PLC system. Both actuators receive power from the "**A**" group supply (24V A / GND A), which is reserved for actuators to isolate them from sensitive input or sensor power lines.

The two connected devices are:

1. Solenoid Coil - **-G1MB1**
2. DC Motor - **-G1MA1**, powered through a controller module **-G1KF1**

Stand-alone mode



PLC mode

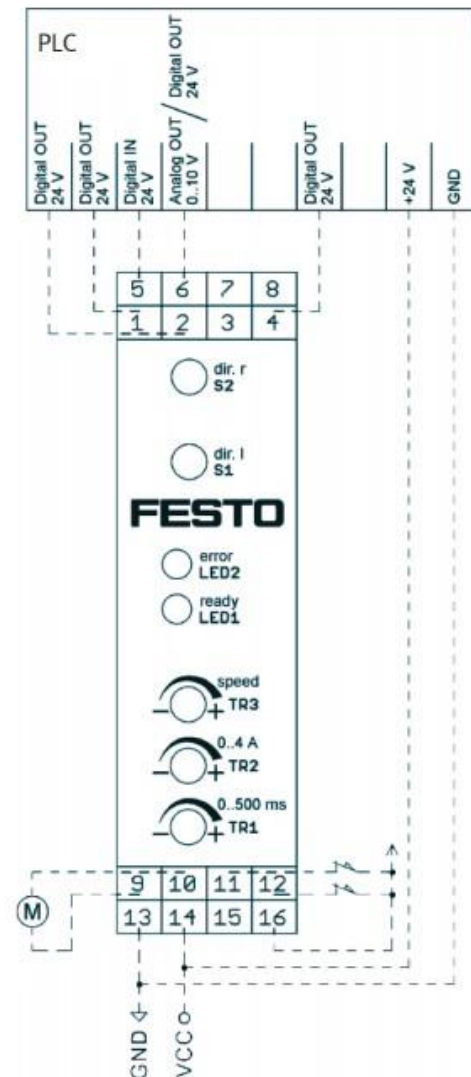


Figure 4.72: Simplified pin diagram for solenoid and motor actuator connection

1. Components

- **Terminal Block:** Terminal strips **X2**, **X3**, and **X4** are part of the Mini I/O Terminal block (**-G1XG1**), used to connect and distribute actuator power and control signals.
- **-G1MB1 (Solenoid Coil):** The identifier “MB” represents a magnetic coil or solenoid. Its terminals **A1** and **A2** correspond to the coil’s positive and return connections.
- **-G1MA1 (DC Motor):** The motor symbol with “M” indicates a rotary DC actuator, used for mechanical movement or conveyor drive.
- **-G1KF1 (Controller / Interface Module):** The controller or relay interface that switches the higher current to the motor, isolating the PLC from direct load.

4.4 Learning Outcome: Generate PLC Program for Sequence Control

2. Actuator 1 - Solenoid Coil (-G1MB1) The solenoid coil is directly controlled by a PLC digital output channel.

Control Signal Path:

1. The PLC output (e.g., **Q2** or **Q6**) provides a 24V DC control signal.
2. The signal travels through the Centronics and D-Sub adapter cable, reaching **X2, Pin 9** (labeled “2/6”) on the Mini I/O Terminal.
3. A black wire connects **X2, Pin 9** to terminal **A1** of solenoid coil **-G1MB1**.

Ground Path:

1. The other side of the coil, terminal **A2**, is connected via a blue wire to the **X4** terminal strip (**GND A**), completing the circuit.

Function: When the PLC output is energized, 24V DC is applied to **-G1MB1**, energizing the coil and actuating the solenoid (for example, operating a pneumatic valve).

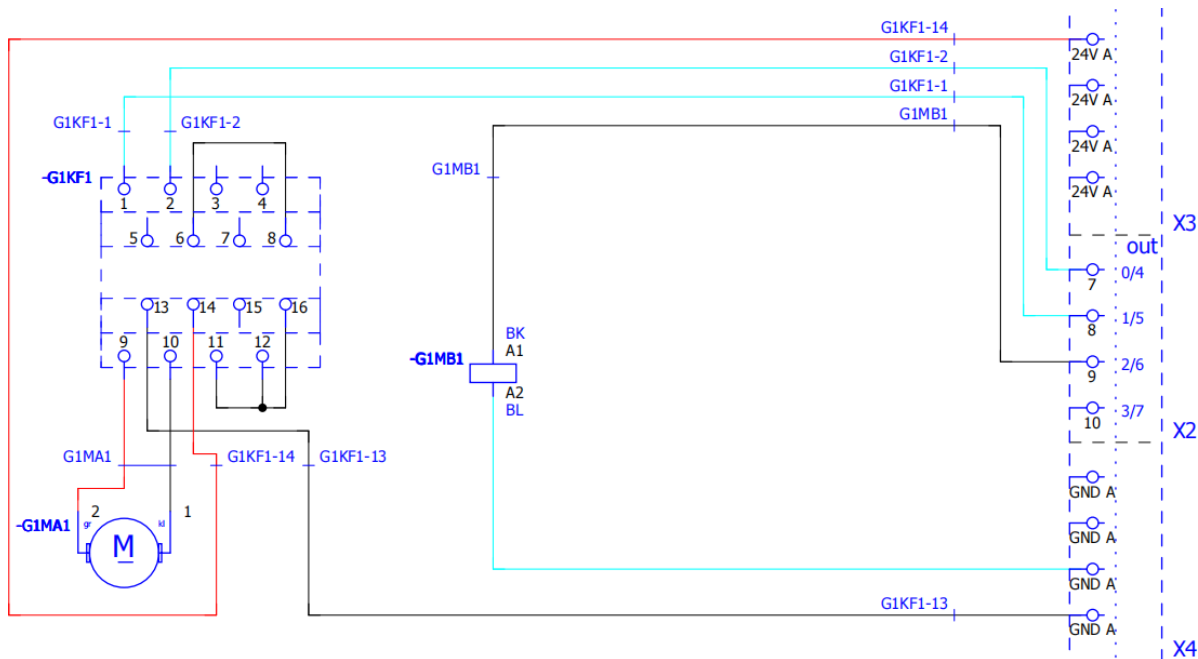


Figure 4.73: Solenoid (-G1MB1) direct control connection via X2 Pin 9 and GND A return

3. Actuator 2 - DC Motor (-G1MA1) with Controller (-G1KF1) The motor circuit uses a controller interface to handle power switching and protection. The PLC controls the motor indirectly through this controller.

Power Supply Path:

1. The controller **-G1KF1** receives 24V DC power from terminal strip **X3 (24V A)** through wire **G1KF1-14**.
2. The ground reference is provided by terminal strip **X4 (GND A)** through wire **G1KF1-13**.

Module-4: Perform Electro-pneumatic Sequence Control

3. Additional internal supply connections (e.g., **G1KF1-1** and **G1KF1-2**) feed logic or relay circuits inside the controller.

Motor Connection:

- The positive terminal of motor **-G1MA1 (Pin 1)** connects to controller terminal **Pin 10**.
- The negative terminal of motor **-G1MA1 (Pin 2)** connects to controller terminal **Pin 11**.

Function: When the controller receives a control signal from the PLC, it switches the motor's supply path. The motor then runs according to the configured control mode (e.g., forward, reverse, or speed control).

Direction "clockwise" (2)	Direction "counterclockwise" (1)	Limit switch "right" (12)	Limit switch "left" (11)	Motor + (10)	Motor - (9)	Function
1	0	1	X	VCC	 GND	Clockwise rotation
0	1	X	1	 GND	VCC	Counterclockwise rotation
1	1	X	X	GND	GND	Quick stop
0	0	X	X	Open	Open	Stop

Figure 4.74: Motor forward/reverse operation via controller

4.4.9 Job Sheet - PLC Programming and HMI Design for MPS Distribution Station

Section	Content
Job Title	PLC Programming and HMI Design for Festo MPS Distribution Station
Objective	To develop, simulate, and test a PLC control program and HMI interface for the Festo MPS Distribution Station, covering signal mapping, sequence design, ladder programming, and process visualization.
Tools and Equipment	PC with TIA Portal and FluidSIM software, Festo MPS Distribution Station , PLC (Siemens S7-1500), HMI Panel, EasyPort or SysLink C-interface cable, 24 V DC power supply, compressed air (600 kPa / 6 bar).

Continued on next page

4.4 Learning Outcome: Generate PLC Program for Sequence Control

Section	Content
Procedure	1. Preparation - Verify system connections: 24 V DC power and 6 bar compressed air. - Ensure actuators (ejector, conveyor) are in their home positions and no workpieces are on the conveyor. - Open a new project in TIA Portal and create a PLC device (S7-1500).
	2. Create Tag Table - Follow the Festo MPS pin assignment table to define all I/O addresses. - Assign descriptive tag names (e.g., Ejector_Retracted, Conveyor_Motor, Magazine_Empty). - Group tags logically under Inputs, Outputs, and Memory Bits.
	3. Develop Sequence Table - Define the motion sequence for the station (Eject → Convey → Stop). - Create a step-based sequence table with the columns: Step, Sensor / Switch, Relay / Memory, and Output. - Verify that every actuator transition is triggered by a unique sensor condition.
	4. Create Ladder Program - Implement the sequence logic in ladder diagram format using Set/Reset or Step Sequence logic. - Include interlocks to prevent overlapping motion. - Test logic in TIA Portal Simulation (PLCSIM) or FluidSIM before hardware deployment.
	5. Design HMI Interface - Create a new HMI screen in TIA Portal. - Add controls: START, STOP, and RESET buttons. - Add process indicators: Magazine Empty, Workpiece Present, System Running. - Link each HMI object to the PLC tags via the established tag table.

Continued on next page

Section	Content
	6. Download and Test on Physical Station - Download the PLC and HMI programs to the hardware. - Connect the PLC to the MPS station via the SysLink cable. - Activate air supply and test each actuator manually via simulation box or HMI. - Run the full sequence and verify: (1) Workpiece ejects, (2) Conveyor runs until sensor detects presence, (3) Conveyor stops.
Notes	- Always depressurize pneumatic circuits before physical adjustments. - Ensure correct SysLink cable orientation (male-female). - Tag consistency between PLC, HMI, and FluidSIM is essential for proper communication. - For offline verification, use EasyPort USB for virtual signal testing with FluidSIM or EasyVeep.
Key Technical Points	- Input and output signals are mapped as per Festo pin allocation (I0-I7, Q0-Q7). - Sequence control ensures synchronized ejector and conveyor operation. - The HMI interface provides operator interaction and fault indication. - Combined simulation and physical testing ensures robust commissioning.

Specification Sheet

4.4.10 Job Sheet - Programming and Simulation of MPS Sorting Station

Section	Content
Job Title	PLC Programming and HMI Design for Festo MPS Sorting Station
Objective	To develop and test a PLC and HMI program for the Festo MPS Sorting Station, implementing logic-based workpiece sorting using multiple sensors and pneumatic deflectors in both simulation and physical setups.

Continued on next page

4.4 Learning Outcome: Generate PLC Program for Sequence Control

Section	Content
Tools and Equipment	PC with TIA Portal and FluidSIM software, Festo MPS Sorting Station (with conveyor, detection, and three chutes), Siemens S7-1500 PLC, HMI Panel, EasyPort USB or SysLink cable, 24 V DC power supply, compressed air (600 kPa).
Procedure	1. Preparation - Ensure 24 V DC power and compressed air supply (6 bar) are available. - Verify initial actuator positions: conveyor stopped, stopper and both deflectors retracted. - Remove all workpieces from the conveyor and chutes before testing. - Create a new PLC project in TIA Portal and configure the device (CPU + HMI). 2. Create Tag Table - Refer to the Festo pin assignment table and assign PLC I/O tags (e.g., I4: Workpiece detected, I5: Color sensor, I6: Metal sensor, Q1: Deflector 1, Q2: Deflector 2, Q3: Stopper). - Define symbolic names and descriptions for all sensors, actuators, and indicators. - Group tags logically as Inputs, Outputs, and Memory bits. 3. Develop Sequence Table - Create a step-wise sequence table showing sorting flow based on detected material and color. - Define each step with its corresponding sensor condition, control relay, and output. - Example: Metallic → Middle chute, Red → Front chute, Black → End chute. 4. Create Ladder Program - Implement logical operations (AND, OR, NOT) in ladder diagram to control deflectors and stopper. - Add conditions so only one deflector is activated per sorting decision. - Test the program logic in TIA Portal PLCSIM or FluidSIM before downloading.
<i>Continued on next page</i>	

Section	Content
	5. Design HMI Screen - Create a user interface with START, STOP, and RESET buttons. - Include indicators for Workpiece Detected, Chute Full, and Sorting Complete. - Link HMI elements to PLC tags via the project tag table.
	6. Download and Test on Physical Station - Download the PLC and HMI programs to their respective devices. - Connect the PLC to the Sorting Station via the SysLink or EasyPort interface. - Run the sequence and place one workpiece at a time on the conveyor. - Verify: Metallic → Middle chute, Red → Front chute, Black → End chute. - Observe operation via HMI and confirm accurate signal feedback.
Notes	- Always depressurize the station before manual adjustments. - The “Chute Full” sensor must stop the sequence automatically. - Verify correct sensor calibration for consistent sorting accuracy. - Use the simulation box or EasyPort to test I/O mapping before full run.
Key Technical Points	- Sorting logic uses multi-sensor detection and conditional PLC programming. - Deflectors are controlled by Boolean logic and synchronized with conveyor timing. - The HMI enables operator control, fault indication, and safe operation. - Simulation and hardware validation ensure reliable commissioning.

4.4 Learning Outcome: Generate PLC Program for Sequence Control

distributionPLC2 ▶ PLC_1 [CPU 1516-3 PN/DP] ▶ PLC tags

Tags Use

PLC tags

	Name	Tag table	Data type	Address	Retain	Access...	Write...	Visibl.
1	WP_Beginning	Tag table_1	Bool	%I0.0		☒	☒	☒
2	Is_Mag_Empty	Tag table_1	Bool	%I0.6		☒	☒	☒
3	Ejector_Retracted	Tag table_1	Bool	%I0.4		☒	☒	☒
4	Ejector_Advanced	Tag table_1	Bool	%I0.5		☒	☒	☒
5	Conveyor_Motor_FWD	Tag table_1	Bool	%Q0.0		☒	☒	☒
6	Conveyor_Motor_Back	Tag table_1	Bool	%Q0.1		☒	☒	☒
7	WP_End	Tag table_1	Bool	%I0.2		☒	☒	☒
8	Start_SW	Tag table_1	Bool	%I1.0		☒	☒	☒
9	Stop_SW	Tag table_1	Bool	%I1.1		☒	☒	☒
10	Reset_SW	Tag table_1	Bool	%I1.3		☒	☒	☒
11	Key_SW	Tag table_1	Bool	%I1.2		☒	☒	☒
12	Mag_Empty_Led	Tag table_1	Bool	%Q1.2		☒	☒	☒
13	Start_LED	Tag table_1	Bool	%Q1.0		☒	☒	☒
14	Reset_LED	Tag table_1	Bool	%Q1.1		☒	☒	☒
15	Advance_Slide	Tag table_1	Bool	%Q0.4		☒	☒	☒
16	R_Cyl_Adv	Tag table_1	Bool	%M0.0		☒	☒	☒
17	R_Cyl_Retr	Tag table_1	Bool	%M0.1		☒	☒	☒
18	R_Conv_FWD	Tag table_1	Bool	%M0.2		☒	☒	☒
19	R_Conv_Stop	Tag table_1	Bool	%M0.3		☒	☒	☒
20	R_END	Tag table_1	Bool	%M0.4		☒	☒	☒
21	EMO_RLY	Tag table_1	Bool	%M0.5		☒	☒	☒
22	RST_RLY	Tag table_1	Bool	%M0.6		☒	☒	☒
23	RLO_bit	Tag table_1	Bool	%M0.7		☒	☒	☒
24	Mag_Empty_Relay	Default tag table	Bool	%M1.0		☒	☒	☒

Figure 4.75: Distributing/Conveyor Station Tag Table

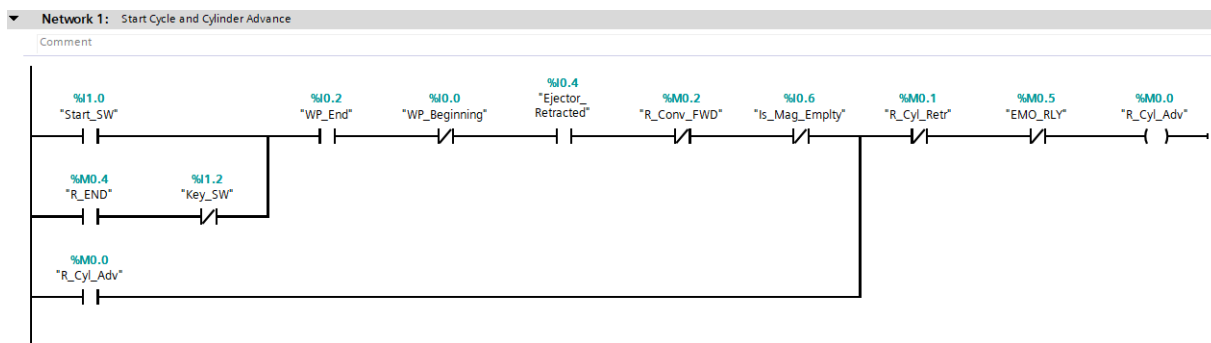


Figure 4.76: Distributing/Conveyor Station Ladder Network 1

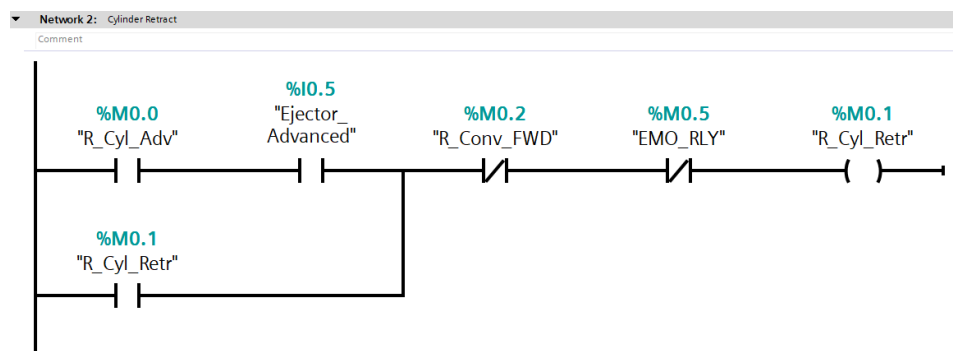


Figure 4.77: Distributing/Conveyor Station Ladder Network 2

Module-4: Perform Electro-pneumatic Sequence Control

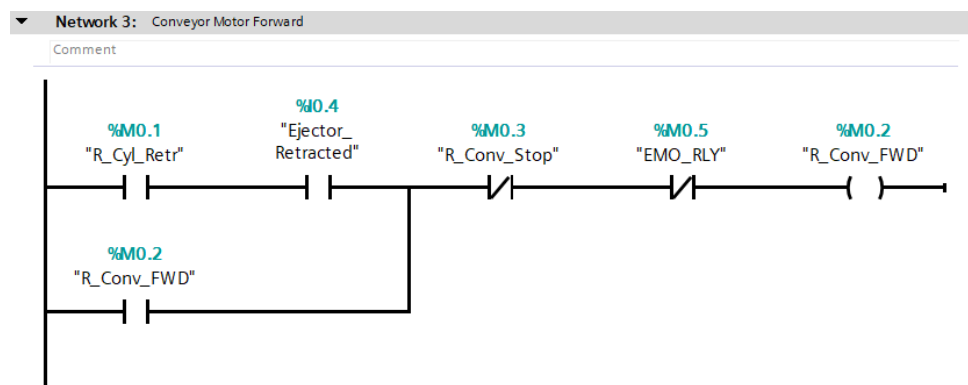


Figure 4.78: Distributing/Conveyor Station Ladder Network 3

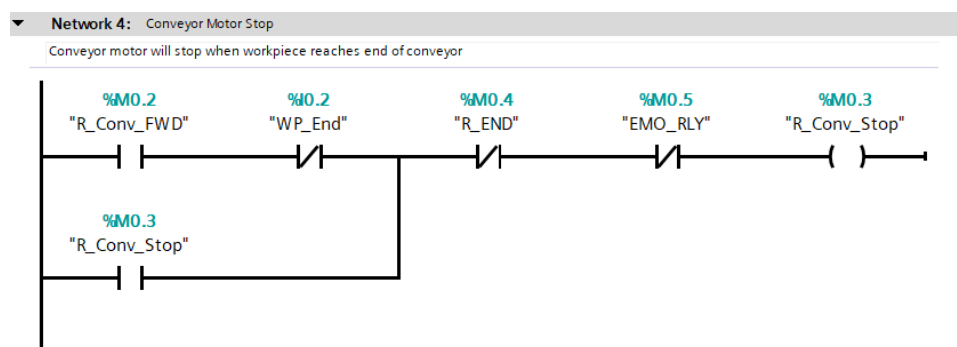


Figure 4.79: Distributing/Conveyor Station Ladder Network 4

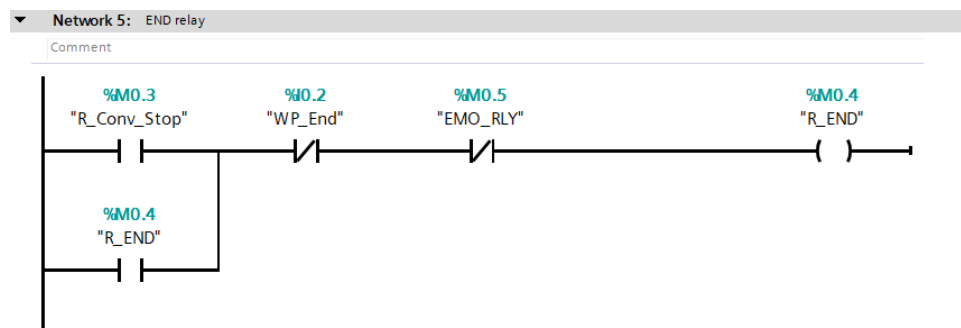


Figure 4.80: Distributing/Conveyor Station Ladder Network 5

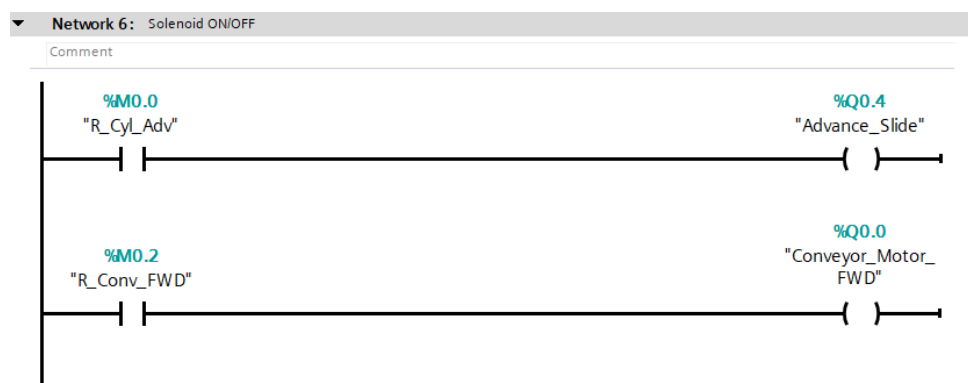


Figure 4.81: Distributing/Conveyor Station Ladder Network 6

4.4 Learning Outcome: Generate PLC Program for Sequence Control

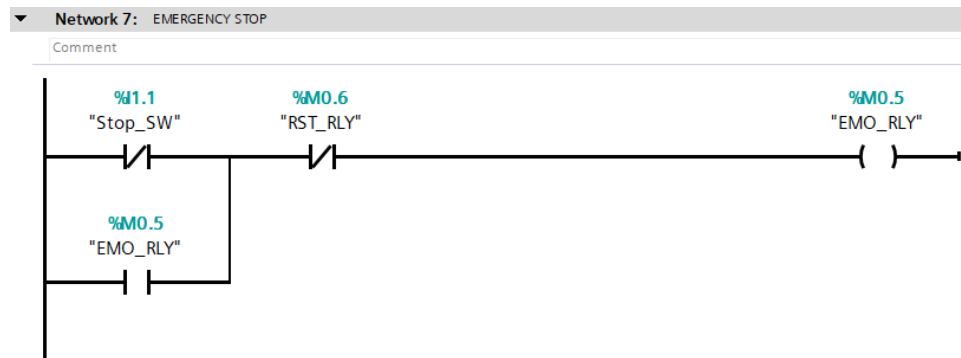


Figure 4.82: Distributing/Conveyor Station Ladder Network 7



Figure 4.83: Distributing/Conveyor Station Ladder Network 8

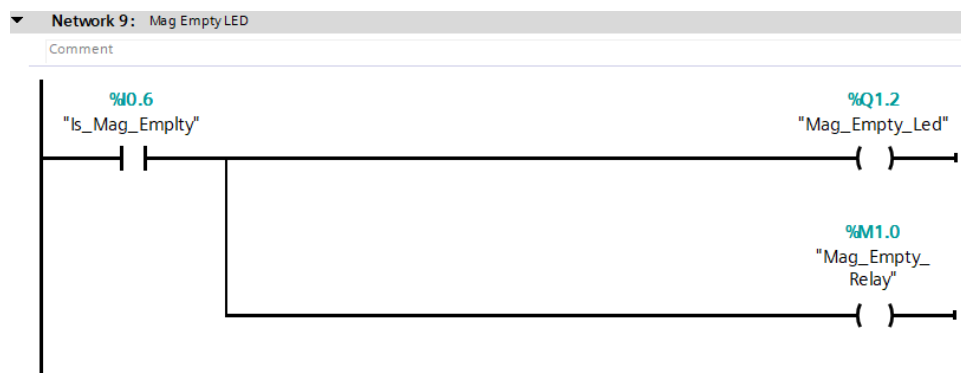


Figure 4.84: Distributing/Conveyor Station Ladder Network 9

Specification Sheet

WP_Beginning	Tag table_1	Bool	%I0.0	☐	☒	☒	☒
Conveyor_Motor_FWD	Tag table_1	Bool	%Q0.0	☐	☒	☒	☒
Chutes_Full	Tag table_1	Bool	%I0.2	☐	☒	☒	☒
Start_SW	Tag table_1	Bool	%I1.0	☐	☒	☒	☒
Stop_SW	Tag table_1	Bool	%I1.1	☐	☒	☒	☒
Reset_SW	Tag table_1	Bool	%I1.3	☐	☒	☒	☒
Key_SW	Tag table_1	Bool	%I1.2	☐	☒	☒	☒
Chutes_Full_Led	Tag table_1	Bool	%Q1.2	☐	☒	☒	☒
Start_LED	Tag table_1	Bool	%Q1.0	☐	☒	☒	☒
Reset_LED	Tag table_1	Bool	%Q1.1	☐	☒	☒	☒
Mag_Empty_Relay	Default tag table	Bool	%M1.0	☐	☒	☒	☒
Def1_EXT	Tag table_1	Bool	%I0.1	☐	☒	☒	☒
Def2_EXT	Tag table_1	Bool	%I0.3	☐	☒	☒	☒
WP_Detect	Tag table_1	Bool	%I0.4	☐	☒	☒	☒
WP_not_Black	Tag table_1	Bool	%I0.5	☐	☒	☒	☒
WP_Metallic	Tag table_1	Bool	%I0.6	☐	☒	☒	☒
EXT_DEFL1	Tag table_1	Bool	%Q0.1	☐	☒	☒	☒
EXT_DEFL2	Tag table_1	Bool	%Q0.2	☐	☒	☒	☒
Retract_Stopper	Tag table_1	Bool	%Q0.3	☐	☒	☒	☒
R_Conv_ON	Tag table_1	Bool	%M0.0	☐	☒	☒	☒
R_Stop_Retr	Tag table_1	Bool	%M0.1	☐	☒	☒	☒
R_Def1_Adv	Tag table_1	Bool	%M0.2	☐	☒	☒	☒
R_Def2_Adv	Tag table_1	Bool	%M0.3	☐	☒	☒	☒
R_End	Tag table_1	Bool	%M0.4	☐	☒	☒	☒
R_Black	Tag table_1	Bool	%M0.5	☐	☒	☒	☒
R_Red	Tag table_1	Bool	%M0.6	☐	☒	☒	☒
R_Silver	Tag table_1	Bool	%M0.7	☐	☒	☒	☒
R_WP_Detected	Tag table_1	Bool	%M1.1	☐	☒	☒	☒

Figure 4.85: Sorting Station Tag Table

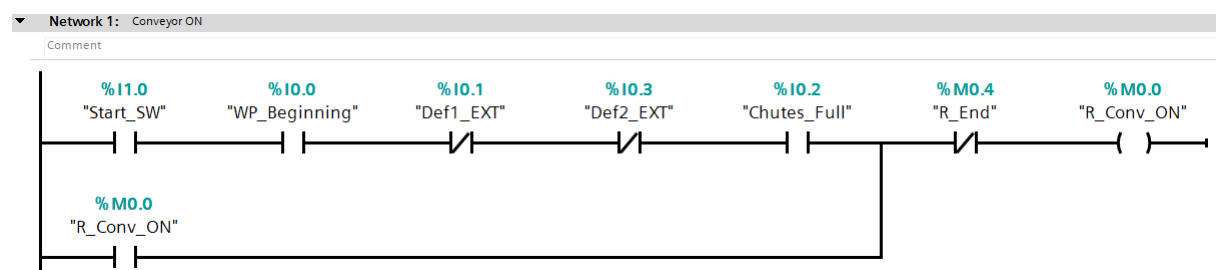


Figure 4.86: Sorting Station Ladder Network 1

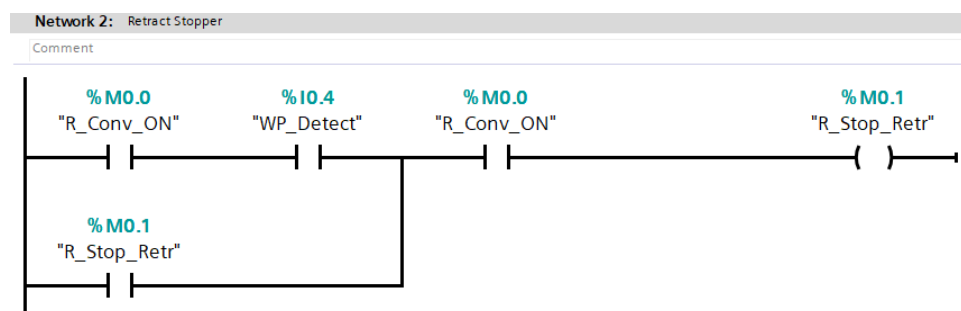


Figure 4.87: Sorting Station Ladder Network 2

4.4 Learning Outcome: Generate PLC Program for Sequence Control

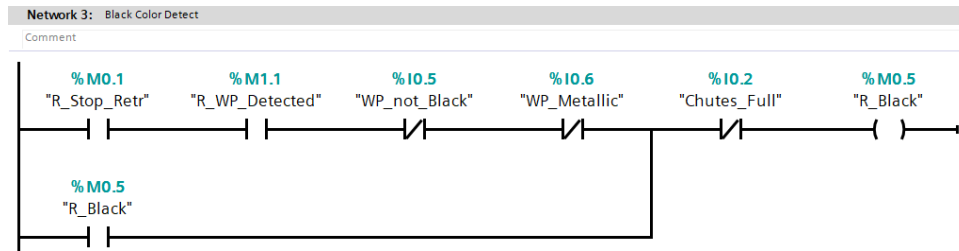


Figure 4.88: Sorting Station Ladder Network 3

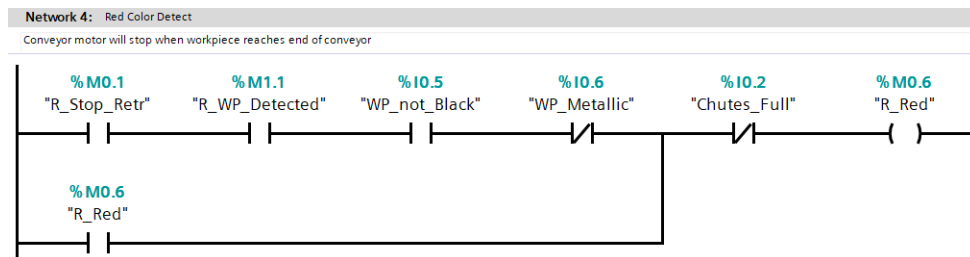


Figure 4.89: Sorting Station Ladder Network 4

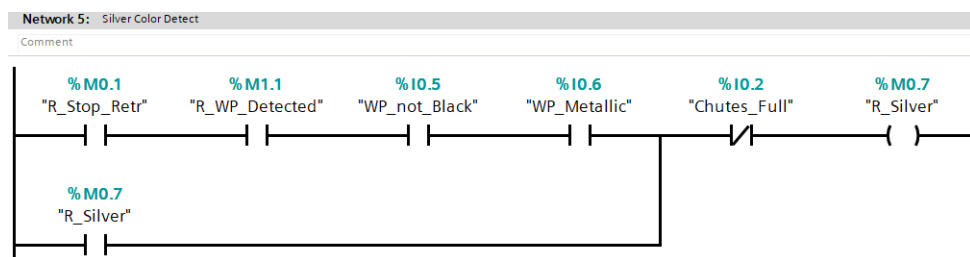


Figure 4.90: Sorting Station Ladder Network 5

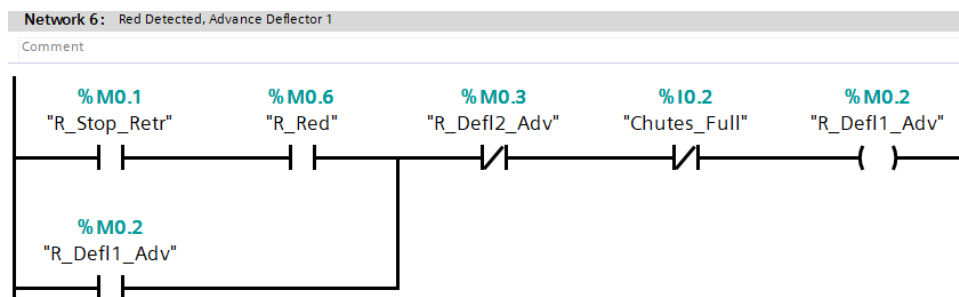


Figure 4.91: Sorting Station Ladder Network 6

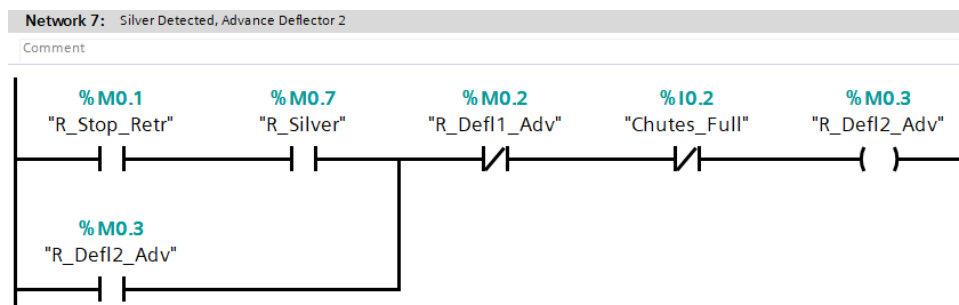


Figure 4.92: Sorting Station Ladder Network 7

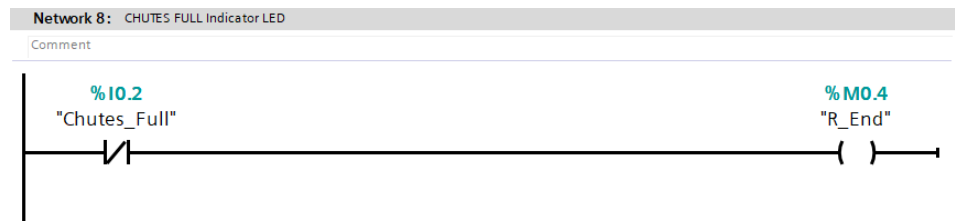


Figure 4.93: Sorting Station Ladder Network 8

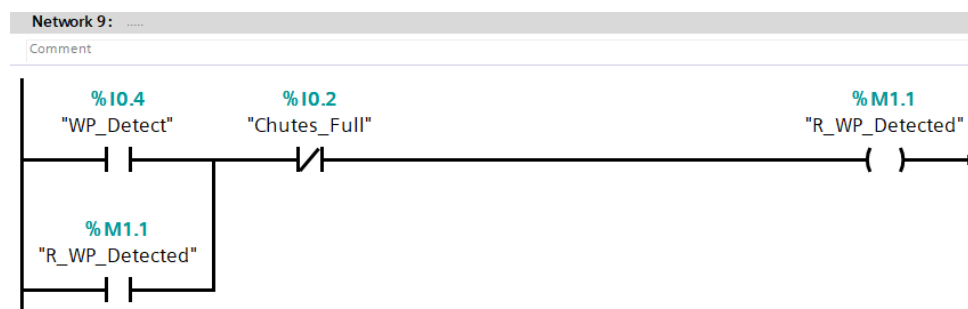


Figure 4.94: Sorting Station Ladder Network 9

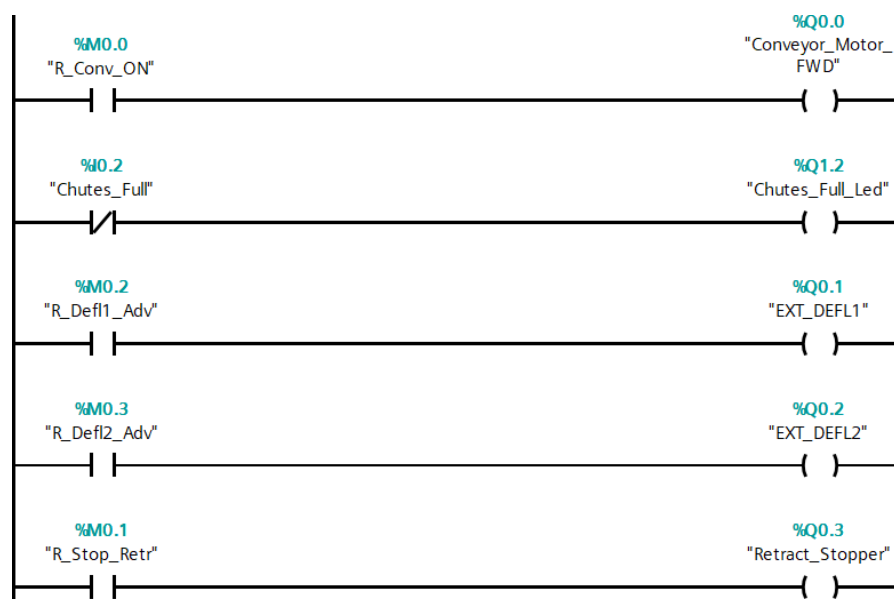


Figure 4.95: Sorting Station Ladder Network 10

4.4.11 Self-Check Questions

No.	Question
1	What is the fundamental structure of the Festo Modular Production System (MPS) based on components, modules, and elements?
2	How are I/O signals organized in the MPS Distributing Station when interfacing with a PLC using the SysLink standard?
3	In the MPS Distributing Station, what physical event triggers the start of the conveyor motor (Q0 active)?
4	If a metallic workpiece is detected by the Sorting Station, which deflector is activated, and where is the workpiece sorted?
5	Describe the function of the DC Motor Controller in the MPS system concerning the conveyor belt, and identify the required control signal type for directional movement.
6	Why is the SysLink I/O cable (e.g. Pin 13 Grey-Pink Input 0) used in MPS stations, and what does it connect to?
7	When programming a sequence, what are the NORM_X and SCALE_X functions primarily used for, and why is this relevant in MPS programming?
8	Name two control components found on the Robot Interface Box (RIA Box) used to interface the robot controller with peripherals.
9	What are the primary functions of the HMI designed for an MPS station, and name three elements typically included on the interface.
10	When programming the sequential logic for an MPS station (e.g. Eject → Convey → Stop), what type of logic implementation is typically utilized in the PLC?

4.4.12 Answer Key

No.	Answer / Guidance
1	MPS is structured hierarchically: MPS Station → composed of modules → composed of elements.
2	SysLink uses dedicated pins (I0-I7, Q0-Q7) and color codes to route I/O signals between station modules and the PLC.
3	The conveyor motor turns ON after the ejector cylinder returns to its fully retracted end-position.
4	Metallic workpiece → Deflector 2 activates, sorted to the chute at the middle of the conveyor belt.
5	The DC Motor Controller converts PLC signals into movement commands. Direction control requires a continuous logic “1” sourcing signal to CW or CCW terminals.
6	The SysLink I/O cable connects station sensors/actuators to the PLC via a Centronics connector and mapped pin assignments.
7	NORM_X normalizes raw analog to 0.0-1.0, SCALE_X converts to engineering units. Used if analog (e.g. speed control) exists.
8	RIA Box includes Centronics sockets (X0-X4) for robot controller + peripherals (e.g., Handling module, Operator panel).
9	HMI allows operator interaction, real-time process display, and fault indication. Typical elements: START, STOP, RESET + status indicators.
10	Sequential logic is implemented using Set/Reset or Step-Sequence logic in ladder diagram.

4.5 Learning Outcome: Operate Modular Production System (MPS) Stations

Objective: To recognize and interpret the standards, symbols, and fundamental components (both electrical and pneumatic) necessary for understanding robot and electropneumatic circuit diagrams, including the basic components in robots and their control systems.

Learning Activity: Study the specific industry standards (ISO 1219-1 and DIN EN 81346-2) applied in pneumatic and electrical circuit diagram representation; identify and describe the symbols for directional control valves, actuators, and logical pneumatic elements; recognize and define basic electrical control components (e.g., relays, limit switches, solenoids); and analyze how electrical and pneumatic systems are integrated to form control systems.

4.5.1 Modular Production System (MPS) Stations Commissioning Sequence

The commissioning procedure for Modular Production System (MPS) stations generally involves cautious, systematic steps, moving from individual component checks to overall system verification. The Distributing/Conveyor Station and the Sorting Station are complex modular systems, each with specific operating sequences and safety prerequisites.

General MPS Station Commissioning Procedure

Commissioning typically occurs after installation is complete to eliminate installation and programming errors. Caution and care must be exercised, as power components might move unforeseeably, risking injury or damage.

Safety and Preparation

- Trainees must work only under the supervision of an instructor.
- Read and understand all safety instructions and warnings, confirming with a signature before participation.
- Wear personal safety equipment, such as safety glasses, hearing protection, and safety shoes.
- Switch off working and control power before working on the circuit, and only reach into the setup when it is completely stopped. Observe potential overtravel times for drives.
- Secure all components firmly on the profile plate.
- Use a tool, such as a screwdriver, to actuate limit switches during troubleshooting.
- Make sure limit valves are not actuated from the front.

4.5 Learning Outcome: Operate Modular Production System (MPS) Stations

Commissioning Stages It is advisable to break down the project into functional structures and proceed from individual devices to function groups, and finally to the overall system:

1. **Prepare Sources of Supply Power:** Check that mains voltage meets required values. All control console functions must be switched off, and the emergency stop buttons must be engaged.
2. **Commission Pneumatic Control Chains:** Check manual overrides on valves and set them to the neutral position. Set endangered actuators to a safe position if bistable valves are used. Set the pressure regulator to a minimum of 1.5 bar, then gradually increase to the specified operating pressure while eliminating leaks. Adjust the ideal traversing speed for cylinders using one-way flow control valves.
3. **Commission Electrical Equipment:** Check connections for unequivocal identification/coding and insert plugs/connectors carefully (with mains switch turned off). Turn on the mains switch and activate control voltage (24 Vdc).
4. **Commission Function Groups:** Test homing if necessary, travel to initial positions, and commission function groups one after the other.
5. **Commission Overall System:** Verify the overall system functions based on the interaction of all sub-functions and meets specified requirements.

Performance Test using a Simulation Box

A performance test can be conducted using a digital simulation box by manually controlling the system outputs (Q0-Q7) and checking inputs (I0-I7) against the pin allocation table.

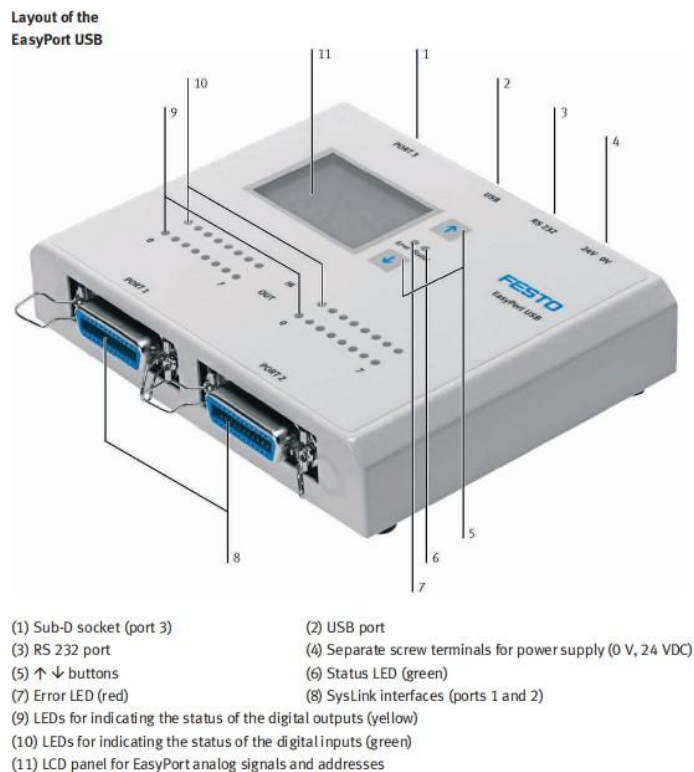


Figure 4.96: Easyport Simulation Box

4.5.2 Information Sheet - Commissioning Sequence for the MPS Distributing/Conveyor Station

The Distributing/Conveyor Station separates workpieces from a stacking magazine using a double-acting cylinder and transports them using a conveyor module.

Start-up Prerequisites and Initial Position The sequence begins with the following conditions:

- No workpiece at the beginning of the conveyor.
- Magazine filled with workpieces.
- Ejector cylinder retracted (Initial Position).
- Conveyor motor off (Initial Position).

Sequence Description The sequence proceeds as follows:

1. When the start key is pressed, the ejector cylinder is advanced (pushed out) and pushes a workpiece out of the magazine.
2. When the ejector cylinder reaches its retracted end-position again, the conveyor motor is switched on.
3. The workpiece is transported to the end of the conveyor.
4. The presence of the workpiece is checked at the end of the conveyor, and the conveyor motor is switched off.

Starting the Sequence (Operational)

- Check power supply (24 V DC, 4.5 A) and compressed air supply (600 kPa / 6 bar).
- Manually remove workpieces from transfer points before executing the alignment/adjustment procedure (prompted by the blinking ALIGN key).
- Fill the magazine with workpieces.
- Start the sequence by pressing the illuminated START key.
- If the magazine is empty (MAG. EMPTY lamp lit), load workpieces and acknowledge by pressing the START key.

4.5.3 Information Sheet - Commissioning Sequence for the MPS Sorting Station

The Sorting Station classifies workpieces by material and color (black, red, metallic) onto three chutes using electrically operated deflectors and a detection module.

Start-up Prerequisites and Initial Position

- Workpiece at the beginning of the conveyor belt (Start-up prerequisite).
- Conveyor motor off (Initial Setting).
- Stopper advanced (Initial Setting).
- Deflector 1 and Deflector 2 retracted (Initial Setting).
- Chutes not full (Initial Setting).

The functionality relies on added components like a deflector, pneumatic stopper module, and sensors for chute full status and deflector extended signals.

Sequence Description (Main Steps and Sorting Logic) The sequence begins when a workpiece is detected:

1. Workpiece detected at the beginning of the conveyor belt.
2. Conveyor motor turns ON.
3. Color/material detection occurs.
4. **Sorting Logic:** The workpiece is handled based on detection:
 - **If Metallic Workpiece Detected:** Extend deflector 2 → Retract stopper → Workpiece sorted out → Retract deflector 2. (Ablage on the chute at the middle of the conveyor belt).
 - **If Red Workpiece Detected:** Extend deflector 1 → Retract stopper → Workpiece sorted out → Retract deflector 1. (Ablage on the chute at the beginning of the conveyor belt).
 - **If Black Workpiece Detected:** Retract stopper → Workpiece sorted out → Idle step. (Ablage on the chute at the end of the conveyor belt).
5. **End of Cycle:** Conveyor motor turns OFF → Stopper advances.

Starting the Sequence (Operational)

- Check power and compressed air supply.
- Manually remove workpieces from transfer points before conducting the alignment procedure.
- Place a workpiece onto the beginning of the conveyor.
- Start the sequence by pressing the illuminated START key.
- If 6 workpieces are on a chute (CHUTE FULL indicator lights up), remove them and acknowledge by pressing the START key.

4.5.4 Information Sheet - Calibrating Diffuse Sensor Sensitivity for Work-piece Detection

The adjustment of the diffuse sensor with fiber-optic cable is crucial for ensuring reliable detection, particularly for dark objects like the black workpiece, since light colors reflect light better than dark colors. Diffuse sensors only react to dark materials (such as matt black plastic or black rubber) over a very short distance, or not at all.

To commission and adjust the diffuse sensor with fiber-optic cable at the beginning of the conveyor belt to reliably detect the black workpiece, you should follow a sequential procedure outlined in the practical exercises:

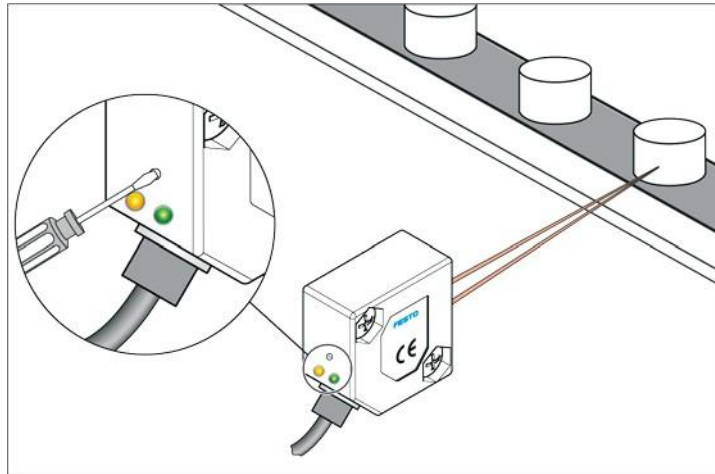


Figure 4.97: Sensitivity adjustment procedure for diffuse sensors

1. **Maximize Sensitivity:** Begin by turning the sensitivity adjusting screw on the fiber-optic unit in the clockwise direction until a soft clicking sound is heard, indicating that the potentiometer is turned all the way up.
2. **Test Initial Detection (Metallic Workpiece):** Place the metallic workpiece at the position where it should be detected. When reliably detected, both the yellow and the green LEDs should light up at the fiber-optic unit.
3. **Adjust for Metallic Workpiece:** Turn the sensitivity adjusting screw in the clockwise direction until the metallic workpiece is reliably detected. This step ensures a baseline detection for a highly reflective material.
4. **Adjust for Red Workpiece:** Place the red workpiece at the position where it should be detected. Observe the reaction of the LEDs and set the sensitivity so that the red workpiece is also reliably detected.
5. **Adjust for Black Workpiece:** Place the black workpiece at the position where it should be detected. Observe the reaction of the LEDs and set the sensitivity so that the black workpiece is also reliably detected.

After this adjustment procedure, all three workpieces (metallic, red, and black) should be reliably detected.

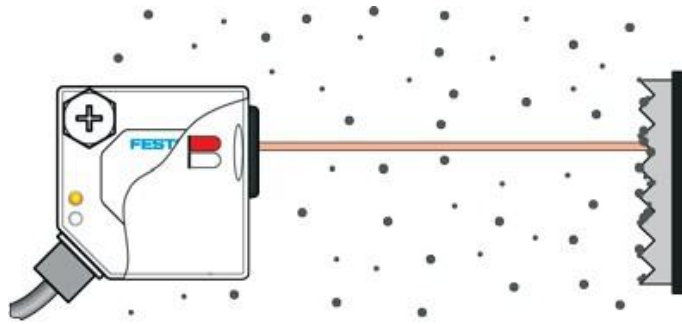


Figure 4.98: Lens contamination of reflective sensors

When adjusting the sensitivity of an optical proximity sensor, the goal is to operate within the "Reliable object detection" range, referred to as range "2" in the context of function reserve. Function reserve defines the additional amount of light beyond what is normally required for switching, compensating for light loss due to attenuation caused by contamination. The intensity of reflected light is evaluated during diffuse reflection, and the sensing distance (scanning width) is adjusted by changing the receiver's sensitivity.

The scanning width of diffuse sensors is related to the object's reflection characteristics; reflection is much greater on flat surfaces than on curved surfaces. Light colors reflect light better than dark colors.

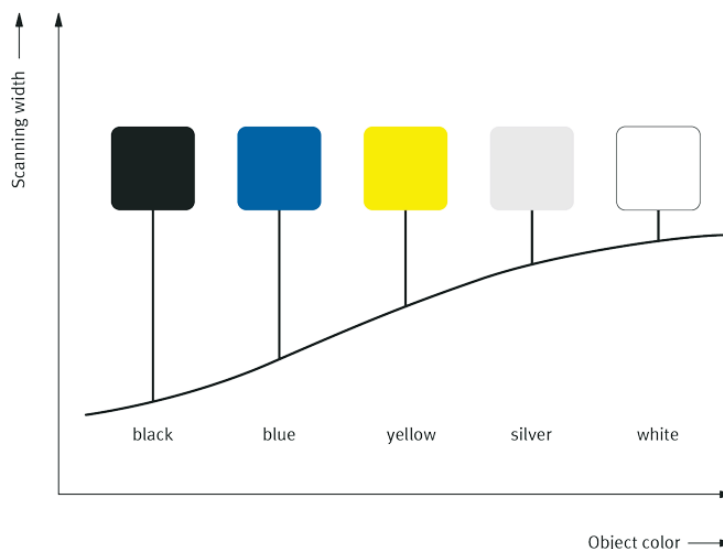


Figure 4.99: Variation of scanning width with object color

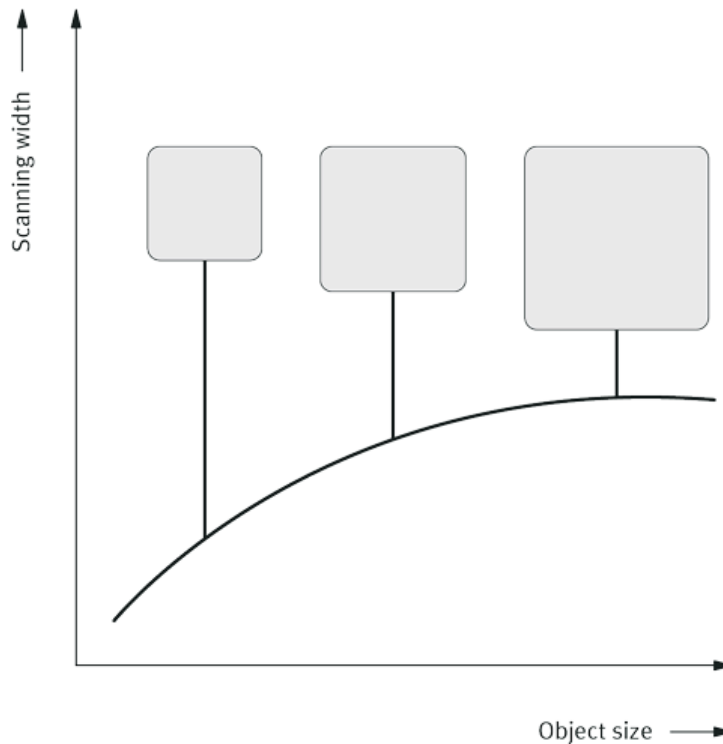


Figure 4.100: Variation of scanning width with object size

4.5.5 Information Sheet - Diagnosing Network Connection Issues (S7-1516, TP700, PC)

Objective:

To identify and resolve communication issues between Siemens **S7-1516 PLC**, **TP700 Comfort HMI**, and **Engineering PC** using both **TIA Portal diagnostic tools** and **Windows CMD commands**.

1. Network Overview

A typical Siemens automation network includes:

- **PLC:** Siemens S7-1516 (with Profinet port X1 P1/P2)
- **HMI:** TP700 Comfort (connected via Profinet)
- **PC:** Engineering Station (with TIA Portal)

All devices communicate over a **Profinet industrial Ethernet network** using static IP addressing.

Device	IP Address	Subnet Mask	Gateway
PC (Engineering Station)	192.168.0.1	255.255.255.0	192.168.0.1
PLC (S7-1516)	192.168.0.2	255.255.255.0	192.168.0.1
HMI (TP700)	192.168.0.3	255.255.255.0	192.168.0.1

2. Troubleshooting in TIA Portal

Step 1 - Verify Device Configuration

- Open **Devices & Networks** → **Network View**.
- Confirm PLC, HMI, and PC are in the same subnet.
- Ensure each has a unique IP address.
- For each device, check: **Properties** → **Ethernet addresses**.

Step 2 - Detect Online Devices

- Go to **Online** → **Accessible Devices**.
- Select the correct **PG/PC interface** (e.g., PN/IE adapter).
- Click **Start Search**.

If a device is not found:

- Check Ethernet cable or switch connection.
- Verify PLC port LEDs (LINK & ACT) are active.

Step 3 - Assign IP and Device Name

- If the device has no name, select it and assign IP and name (e.g., PLC_1, HMI_1).
- Ensure the assigned name matches the project configuration.

Step 4 - Go Online to PLC

- Navigate to **PLC** → **Online & Diagnostics** → **Go Online**.
- If connection fails:
 - Verify PG/PC interface under **Online** → **PG/PC Interface**.
 - Ensure PC IP matches PLC subnet.

Use **Diagnostics** → **General** → **Module Information** to confirm connection status.

Step 5 - Check HMI Communication

- Open **WinCC Configuration** → **Connections**.
- Verify PLC connection name matches the TIA PLC name.
- Confirm correct IP and interface.
- To download, put HMI in **Transfer Mode** (press corner during boot).

Step 6 - PLC Web Server (if enabled)

- Access via browser: `http://192.168.0.2`
- Navigate to **Diagnostics Buffer** and **Profinet IO Device Status**.

3. Troubleshooting using Windows CMD Tools

Check Network Adapter IP

`ipconfig`

Verify Ethernet adapter IP, subnet, and gateway match the PLC/HMI network. If IP address start with 169.254.x.x (APIPA address), there is no valid connection to the network. The DHCP server is not available, or the network cable is unplugged.

Test Connectivity

```
ping 192.168.0.2    ← PLC
ping 192.168.0.3    ← HMI
```

If Request Timed Out, check cable or subnet mismatch.

Check IP Conflicts

`arp -a`

Compare MAC addresses. Duplicate MACs indicate IP conflict. To clear cache:

`arp -d *`

Check Adapter Transmission

`netstat -e`

If “Bytes Sent = 0,” adapter is disabled or not connected.

Trace Network Route

`tracert 192.168.0.2`

If no hops are displayed, the device is unreachable.

Reset Network Interface

```
netsh interface show interface
netsh interface set interface "Ethernet" admin=disabled
netsh interface set interface "Ethernet" admin=enabled
```

4. Common Issues and Solutions

Issue	Possible Cause	Solution
PLC not detected in TIA	Wrong PG/PC interface	Select correct “PN/IE” adapter
IP conflict	Duplicate IPs	Assign unique IPs to all devices
HMI download fails	Wrong target IP or device name	Check HMI IP and connection name
Ping fails	Cable or subnet mismatch	Verify physical connection and subnet
PLC offline after restart	Lost project or name mismatch	Reassign device name and redownload
HMI shows “PLC not reachable”	Wrong connection path	Match HMI connection name to PLC name

4.5.6 Information Sheet - Backup and Restore Procedures for S7-1500 PLC and HMI Devices

The processes for backing up and restoring SIMATIC S7-1500 PLCs and associated HMI panels generally involve using the Totally Integrated Automation (TIA) Portal for program and configuration management, or utilizing specialized functions via the HMI’s USB port for panel-specific operations and recovery.

S7-1500 PLC Program and Configuration Management (via TIA Portal)

The primary methods for managing the S7-1500 program and configuration involve downloading the project from the PC (restoring the program) or uploading the configuration and program from the PLC (creating a backup copy in the TIA Portal project).

- **Uploading Program and Hardware (Backup):** You can upload the hardware configuration and software of a connected CPU as a new station. To upload program blocks, you prepare the offline project, go online with the CPU, select the "Program blocks" folder in the offline project, and click the "Upload" button. This action copies program blocks from the online CPU to the offline project.
- **Downloading Project to CPU (Restore):** To restore a project, you select the CPU in the project tree and use the "Download to device" command (Hardware and software only changes). During the download process, you select the PG/PC interface type, interface, and connection with the subnet, select the compatible CPU, and click "Load".
- **Managing Data Consistency:** The S7-1200 (Firmware V4.0) and S7-1500 controllers support "Downloading without reinitialisation" to expand interfaces of optimized function or data blocks during operation, allowing changes to be loaded without switching the controller to STOP or influencing the actual values of loaded tags. This requires setting a memory reserve beforehand.

- **Resetting the Controller:** Retentive data (which retains its value after a restart) can be deleted using actions such as "memory reset" or "Reset to factory settings" via the operating switch on the controller, the controller display, or online via STEP 7 (TIA Portal). When a memory reset is commanded from STEP 7, it clears all work memory, clears retentive and non-retentive memory areas, copies load memory to work memory, and sets outputs to the configured "Reaction to CPU STOP". A complete reset to factory settings requires the CPU to have an online connection and be in STOP mode.

HMI Panel Backup and Restore Operations

SIMATIC HMI panels, such as the KTP700 Basic, utilize the TIA Portal for configuration downloads and feature USB functionality for localized backup, recovery, and updates.

- **HMI Project Backup/Restore:** The project file for HMI devices can be stored as a backup on USB mass storage. For recovery, specific "Recovery software for resetting to factory settings via USB" can be stored on the USB mass storage.
- **Configuration Download (Restore):** To download the visualization (program) to the panel, you select the panel folder in the TIA Portal and click the "Download to device" button. The Extended Download manager requires correctly selecting the interface type (e.g., PN/IE), the PG/PC interface, and the connection subnet.
- **Backup/Restore via ProSave:** For Basic Panels, the backup/restore of configuration, operating system, and firmware to a PC can be performed using ProSave.
- **USB Reset to Factory:** Basic Panels offer a "USB Rescue Stick" feature to reset the panel if the connection is impossible. The procedure for an HMI Panel Reset to Factory using a USB stick involves several steps:
 1. Create the USB stick (must be FAT32 formatted, and the Image must be TIA V14 or higher).
 2. Download the required software and unzip it onto the USB stick.
 3. Copy the Image file from the TIA folder into a folder on the USB stick.
 4. Start the Basic Panel with the USB stick inserted.
 5. Follow the instructions displayed on the panel.
 6. Restart the device.

4.5.7 Job Sheet - Operate Sorting Station

Section	Content
Job Title	Operate Sorting Station

Continued on next page

4.5 Learning Outcome: Operate Modular Production System (MPS) Stations

Section	Content
Personal Protective Equipment (PPE)	Safety glasses, hearing protection, and safety shoes must be worn during operation.
Materials Required	Assembled Sorting Station (including conveyor, detection, and three chute modules), workpieces (black, red, metallic/chrome-plated), C-Interface, PLC with HMI (TP700 Comfort), SysLink cables, and compressed-air supply (600 kPa / 6 bar).
Tools and Equipment	PC with PLC programming software (TIA Portal or FluidSIM® P), screwdriver, multimeter, EasyPort USB, and simulation box for I/O monitoring.
Procedure	1. Preparation and Safety - Verify that power supply (24 V DC) and air pressure (6 bar) are connected and stable. - Ensure conveyor motor and all actuators (deflectors, stopper) are in the initial position. - Confirm that no workpieces are left on the conveyor or in the chutes before starting. - Check proper sensor alignment for material and color detection.

Continued on next page

Section	Content
	2. Initial Conditions - Workpiece at beginning of conveyor (I0 active). - Conveyor OFF (Q0 inactive), stopper advanced (Q3 inactive). - Deflector 1 and Deflector 2 retracted (Q1, Q2 inactive). - All chutes empty (I2 inactive). 3. Operating Sequence 1. Place a workpiece at the conveyor start (I0 = 1). 2. Press START → Conveyor motor turns ON (Q0 = 1). 3. Sensors detect material/color characteristics. 4. Sorting Logic: - If Metallic: Extend Deflector 2 (Q2) → Retract Stopper (Q3) → Retract Deflector 2. - If Red: Extend Deflector 1 (Q1) → Retract Stopper (Q3) → Retract Deflector 1. - If Black: Retract Stopper (Q3) → Workpiece passes to end chute. 5. End of Cycle: Conveyor stops (Q0 OFF), Stopper advances (Q3 OFF). 6. If chute is full (I2 = 1), remove workpieces and acknowledge via START.
Notes	- Ensure CHUTE FULL indicator (I2) halts the process until cleared. - Always depressurize the pneumatic system before servicing. - Maintain proper spacing between workpieces to avoid detection overlap. - Verify all deflectors and stopper movements are within mechanical limits.
Key Technical Points	- Sorting is a sequence with branching logic based on sensor feedback. - Boolean control uses multiple inputs (AND/OR/NOT) for decision making. - Proper I/O mapping ensures coordination between pneumatic deflectors and conveyor motion. - Safety interlocks (Chute Full, Stopper Advanced) prevent unintended actuation.

4.5.8 Job Sheet - Operate Distributing Station

4.5 Learning Outcome: Operate Modular Production System (MPS) Stations

Section	Content
Job Title	Operate Distributing Station
Personal Protective Equipment (PPE)	Safety glasses, hearing protection, and safety shoes must be worn during operation.
Materials Required	Assembled Distributing/Conveyor Station with stacking magazine module, conveyor module, C-Interface, PLC with HMI (TP700 Comfort), SysLink cables, workpieces, and compressed-air supply (600 kPa / 6 bar).
Tools and Equipment	PC with PLC programming software (TIA Portal or FluidSIM® P), EasyPort USB, screwdriver, tubing cutter, multimeter, and simulation box for testing I/O signals.
Procedure	1. Preparation and Safety - Verify that power supply (24 V DC, 4.5 A) and air pressure (6 bar) are stable. - Ensure all actuators are in the initial position: ejector cylinder retracted, conveyor motor OFF. - Remove any workpieces from transfer points before starting. - Check all electrical connections and pneumatic fittings for tightness and correct placement. - Confirm that the emergency stop circuit is functioning.
	2. Initial Conditions - Magazine filled with workpieces. - No workpiece at the beginning of the conveyor. - Ejector cylinder retracted. - Conveyor motor OFF.
	4. Operating Sequence 1. Press the illuminated START key to begin operation. 2. Ejector cylinder advances (extends) and pushes a workpiece out of the magazine. 3. When the cylinder retracts to its home position, the conveyor motor starts (Q0 = 1). 4. The workpiece is transported to the end of the conveyor. 5. When detected by the end sensor (I2 = 1), the conveyor motor stops (Q0 = 0). 6. If the magazine becomes empty (I6 = 1), the MAG EMPTY lamp illuminates and the system pauses. 7. After refilling the magazine, press START to resume operation.

Continued on next page

Module-4: Perform Electro-pneumatic Sequence Control

Section	Content
Notes	- Never reach into the conveyor or ejector area during operation. - Use a screwdriver to manually actuate limit switches during testing. - If abnormal movement occurs, press E-STOP immediately and check supply lines. - Restart only after verifying all components are in their initial positions.
Key Technical Points	- Sequential control follows a fixed Set → Reset logic sequence. - Each cycle is initiated by a start signal and completed upon end-sensor detection. - Pneumatic motion and electrical control are synchronized through I/O mapping. - Safety interlocks (Magazine Empty, E-Stop) ensure safe and reliable operation.

4.5.9 Job Sheet - Calibrate Diffuse Sensor for Black Workpiece Detection

Section	Content
Job Title	Calibrate Diffuse Sensor for Black Workpiece Detection
Personal Protective Equipment (PPE)	Safety glasses, ESD wrist strap, and safety shoes.
Materials Required	Sorting Station (Detection Module), diffuse optical sensor with fiber-optic cable, workpiece set (metallic, red, and black), power supply (24 V DC), PLC or simulation box for signal observation, and alignment tools.
Tools and Equipment	Small screwdriver (for sensitivity adjustment), multimeter or HMI for input monitoring, compressed air source (6 bar), cleaning cloth, and optional oscilloscope for signal verification.

Continued on next page

4.5 Learning Outcome: Operate Modular Production System (MPS) Stations

Section	Content
Procedure	1. Preparation and Safety - Ensure power is switched OFF during wiring. - Verify 24 V DC and 0 V connections according to sensor color code: Brown (+24 V), Blue (0 V), and Black (Signal Output). - Mount the diffuse sensor correctly with its optical axis perpendicular to the conveyor surface. - Clean the sensor lens and ensure no obstructions are present in the sensing area.
	2. Initial Connection Check - Connect the sensor output to a PLC input (e.g., I5: “Workpiece not black”). - Power ON the system and observe LED indicators on the sensor. - Verify that the Yellow LED (switch output) and Green LED (operational reserve) function correctly when any reflective object is placed in front.
	3. Calibration Steps - Step 1 - Set Baseline: Rotate the sensitivity screw (clockwise) to the maximum position (up to 12 turns). - Step 2 - Metallic Workpiece: Place a metallic workpiece in detection position. Adjust sensitivity until both Yellow and Green LEDs are ON, indicating reliable detection. - Step 3 - Red Workpiece: Replace with a red workpiece and slightly reduce sensitivity until both LEDs remain ON for the red surface as well. - Step 4 - Black Workpiece: Place a black workpiece. Adjust sensitivity so that the black surface does not activate the Yellow LED (no detection). - Step 5 - Verification: Recheck all three materials to ensure distinct responses: Metallic - detected Red - detected Black - not detected - Fine-tune to maintain stable switching points and avoid flicker caused by ambient light.

Continued on next page

Module-4: Perform Electro-pneumatic Sequence Control

Section	Content
Notes	- Over-rotation of the sensitivity screw (> 12 turns) may damage the sensor. - Detection should remain stable despite minor surface contamination. - If the black workpiece is falsely detected, reduce sensitivity until output switches OFF reliably. - For PLC verification, monitor bit status of input I5 in real-time.
Key Technical Points	- Diffuse sensors rely on reflected light; darker surfaces produce weaker signals. - Correct calibration ensures differentiation between metallic, red, and black workpieces. - LEDs provide a quick visual reference of switching and function reserve. - The calibrated output is used in sorting logic to ensure accurate classification of materials.

4.5.10 Self-Check Questions

No.	Question
1	What is the fundamental structure of the Festo Modular Production System (MPS) in terms of its components, modules, and elements?
2	Name three essential components that must be checked for stable supply before operating an MPS station.
3	In the MPS Distributing/Conveyor Station sequence, what action confirms the completion of the ejector cylinder's work, and what output is subsequently activated?
4	Describe the initial mechanical prerequisites for the MPS Sorting Station before beginning operation.
5	Explain the sorting logic in the MPS Sorting Station for a Metallic workpiece, specifying the deflector activated and the sequence of movement.
6	Why must the sensitivity of a diffuse sensor on the Sorting Station be calibrated, especially regarding black workpieces?
7	What is the required condition regarding the E-Stop buttons and control console functions before performing the Commissioning Stages of an MPS station?
8	What must happen if the CHUTE FULL indicator lights up during the operation of the MPS Sorting Station?

4.5 Learning Outcome: Operate Modular Production System (MPS) Stations

9	What are the recommended settings for air pressure (kPa/bar) and electrical supply (V DC) before starting the sequence of an MPS station?
10	How is the ALIGN key/procedure used in the start-up sequence of the MPS Distributing/Conveyor Station?

4.5.11 Answer Key

No.	Answer / Guidance
1	The MPS system is hierarchically structured as: MPS Station → Composed of Modules → Composed of Elements.
2	Compressed air supply (e.g., 600 kPa / 6 bar), DC power supply (e.g., 24 V DC, 4.5 A), and electrical connections/plugs.
3	The conveyor motor is switched ON when the ejector cylinder reaches its retracted end-position again.
4	The workpiece must be at the beginning of the conveyor belt, the stopper must be advanced, and Deflector 1 and Deflector 2 must be retracted.
5	If a Metallic workpiece is detected, Deflector 2 is extended, the stopper is retracted, the workpiece is sorted out (onto the middle chute), and then Deflector 2 is retracted.
6	Darker materials like the black workpiece reflect light less effectively than light colors, meaning diffuse sensors only react to them over a very short distance or not at all. Calibration ensures reliable detection.
7	All control console functions must be switched off, and the emergency stop buttons must be engaged.
8	Workpieces must be removed from the chute, and the operator must acknowledge the removal by pressing the START key before the sequence resumes.
9	The compressed air supply should be checked for 600 kPa / 6 bar (or gradually increased from 1.5 bar), and the electrical supply should be 24 V DC.
10	The ALIGN key/procedure is used to manually remove workpieces from transfer points before executing the alignment/adjustment procedure.

Module-5: Simulate Robotic Arm in Software

Module Description

This unit covers the knowledge, skills and attitudes required to perform robotic program. It specifically includes the tasks of identifying programming syntax, developing program for robot arm movement, developing program for controlling robot input and output and operating robotic arm.

Minimum Time: 25 Hours

Learning Outcomes

1. Programming syntax is identified.
2. Program for robot arm movement is developed.
3. Program for controlling robot input and output is developed.
4. Robotic arm is operated.

5.1 Learning Outcome: Install and Configure Simulation Software

5.1.1 Information Sheet 1: Introduction to CIROS Studio

CIROS Studio (**Computer Integrated Robot Simulation**) is a **universal 3D simulation system** that replaced the previous product **COSIMIR® Professional Release 4.2**. It is distributed by **Festo Didactic**.

System Requirements and Licensing:

1. Festo Didactic provides two types of licenses:
 - **CIROS Studio:** Intended for **teachers and researchers**, allowing users to create new learning scenarios, prepare lessons, manage error behavior, and connect to selected physical robot controllers.
 - **CIROS Education:** Intended for **learners and students**, typically used to work within ready-made or provided models and generate results such as programs or simulation videos.

5.1.2 Information Sheet 2: Understanding User Interface of Software

The user interface of **CIROS Studio** features various menus and windows that allow the user to create, program, and simulate robotic work cells efficiently.

1. **Modeling Menu:** Contains tools like **Model Libraries** and the **Model Explorer** (shortcut CTRL+T) for accessing and managing all elements of a work cell.
2. **Programming Menu:** Includes functions for programming robots, compiling programs, and accessing **Teach-In** mode (shortcut F8) for simulating jog operations and setting positions.
3. **Simulation Menu:** Used to **Start/Stop** the simulation, configure parameters such as the simulation cycle and visualization cycle, and activate checks like **Collision Detection** and **Sensor Simulation**.
4. **Extras Menu:** Provides special functions such as **Camera Cruise** (for creating simulation videos in .AVI format), **Fault Simulation**, and **Online Management** tools.
5. **RCI Explorer (Robot Controller Interface):** Serves as the **information-processing center** of CIROS Studio, enabling drag-and-drop management of robot programs and position lists between the PC and the physical robot controller, as well as online debugging capabilities.

5.1.3 Information Sheet 3: Communication and Integration

CIROS Studio provides interfaces for both direct robot control and integration with external automation systems.

1. **Robot Controller Connection:** CIROS Studio can communicate with physical Mitsubishi robot controllers through either a **serial interface (RS-232)** or a **network interface (TCP/IP)**. The **RCI Explorer** manages these connections, allowing users to check connection properties and upload or download robot programs and position lists between the PC and the physical robot controller.
2. **PLC Integration:** CIROS supports interfacing with external control systems such as PLCs. It uses the **OPC Client** and **OPC Controller Connection** features. Data exchange between CIROS process models and external controllers (such as **S7-PLCSIM** or **CODESYS Soft-SPS**) is typically handled by the **EzOPC server**, often utilizing the **EasyPort USB** interface for communication.

5.1.4 Self-Check Questions

No.	Question
1	Identify the two main licensing types available for CIROS simulation software, based on the user's role (e.g., teaching vs. learning).
2	What primary task is associated with the RCI Explorer component in CIROS Studio, particularly concerning the relationship between the PC and a physical robot controller?.
3	Contrast the purposes of the Modeling Menu and the Simulation Menu in the CIROS Studio user interface.
4	If you needed to create a simulation video in AVI format showing a robotic task, which menu function in CIROS Studio would you utilize?.
5	Describe the function of the OPC Client and EzOPC server when integrating the CIROS simulation environment with an external PLC system.
6	What is the required physical interface used by CIROS Studio to communicate with a Mitsubishi robot controller? Name two options.
7	What mechanism enables CIROS Studio to exchange data in real-time between the virtual process models and an external PLC simulator (e.g., S7-PLCSIM or CODESYS Soft-SPS)?.
8	A user has enabled Collision Detection and Sensor Simulation. Which menu category did they access to perform this configuration?.
9	Which specific CIROS Explorer functionality allows a user to simulate jog operations and set robot positions without running a full program?.
10	The CIROS Studio software is described as replacing a previous product. What was the name of this predecessor software?.

5.1.5 Answer Key

No.	Answer / Guidance
1	The two license types are CIROS Studio (for teachers/researchers who create scenarios and manage error behavior) and CIROS Education (for learners/students who work within ready-made models and generate results).
2	The RCI Explorer serves as the information-processing center, enabling drag-and-drop management of robot programs and position lists between the PC and the physical robot controller.
3	The Modeling Menu contains tools like Model Libraries and Model Explorer for accessing and managing all elements of a virtual work cell. The Simulation Menu is used to Start/Stop the simulation, configure simulation cycles, and activate checks like Collision Detection.
4	The Extras Menu provides special functions such as Camera Cruise, which is used for creating simulation videos in AVI format.
5	The OPC Client allows CIROS to interface with external control systems like PLCs. The EzOPC server acts as a virtual switchboard, utilizing the EasyPort USB interface to connect the signal ranges of the communicating partners and enable synchronized data exchange.
6	CIROS Studio can communicate with physical Mitsubishi robot controllers through either a serial interface (RS-232) or a network interface (TCP/IP).
7	The EzOPC server handles the data transfer between the process models in CIROS and an external PLC simulator (S7-PLCSIM or CODESYS Soft-SPS), often utilizing the EasyPort USB interface.
8	Collision Detection and Sensor Simulation are configured via the Simulation Menu.
9	The Teach-In mode (shortcut F8), accessed through the Programming Menu, is used for simulating jog operations and setting robot positions.
10	CIROS Studio is a universal 3D simulation system that replaced COSIMIR® Professional Release 4.2.

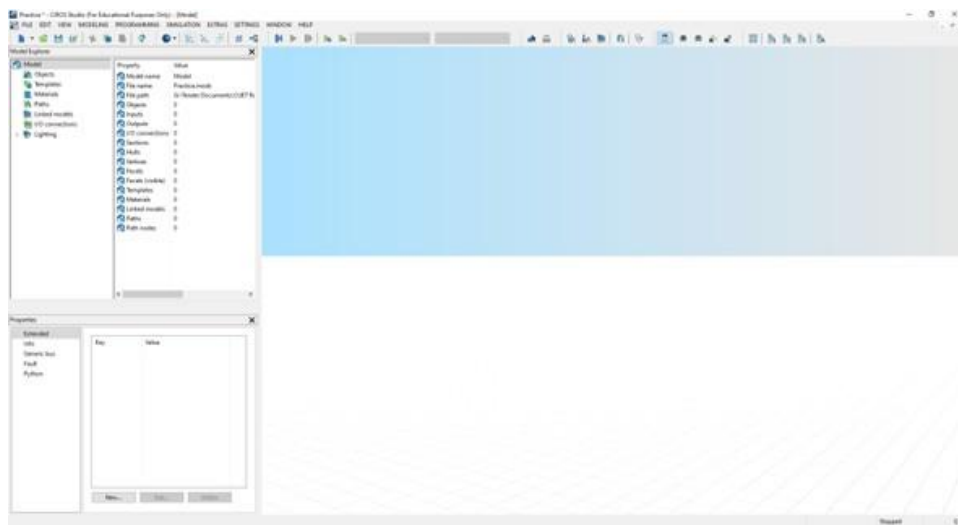
5.2 Learning Outcome: Create and Configure Simulation Environment

5.2.1 Information Sheet 1: Initial Setup and Modeling

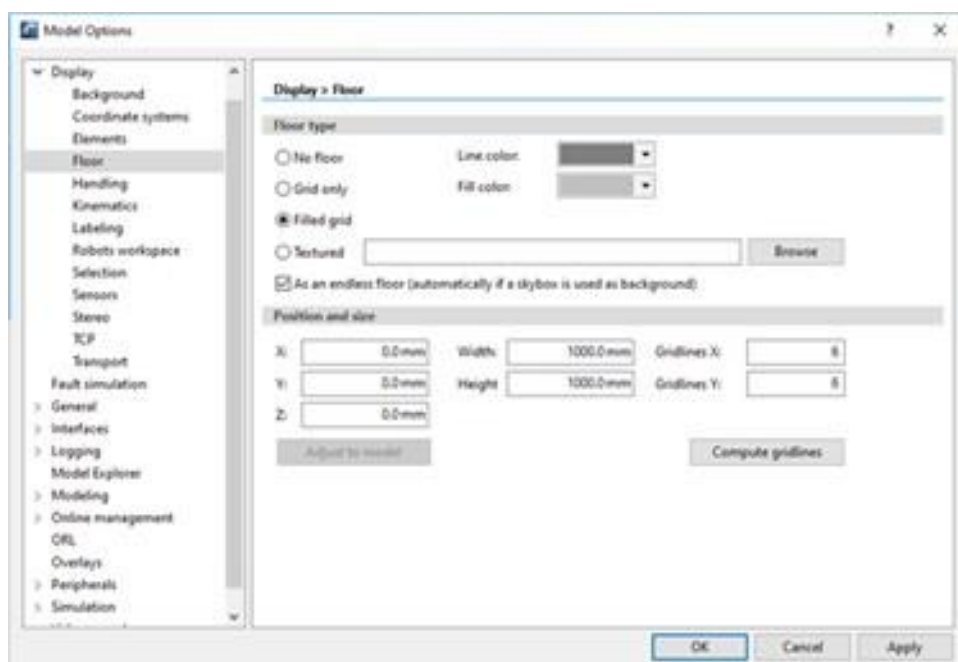
The goal is to create the first CIROS work cell.

Prepare Work Cell and Floor:

1. Start CIROS Studio and use the menu command File\New\Model to create a new work cell.

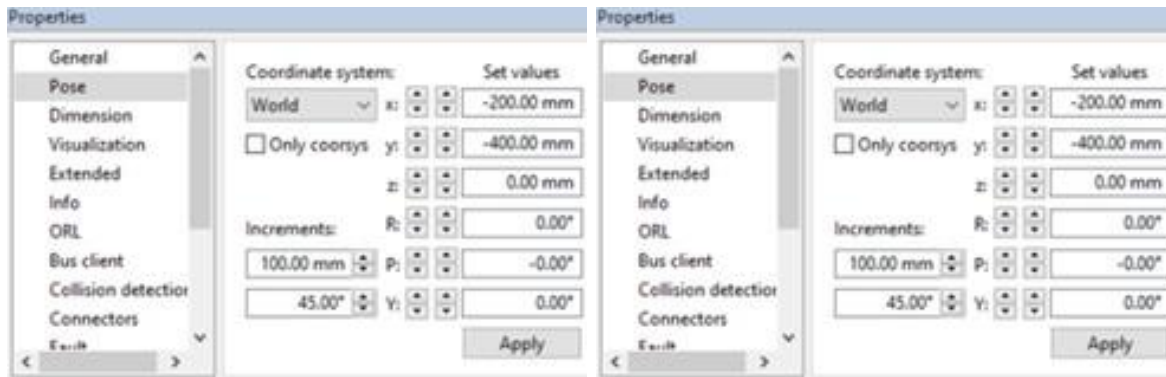


2. Save the work cell file to a directory with write access.
3. Configure the floor by selecting Settings\Model Options\Display\Floor, and check As an endless floor. Optionally, configure the position and size (e.g., set x-value of floor size to **2200 mm**).

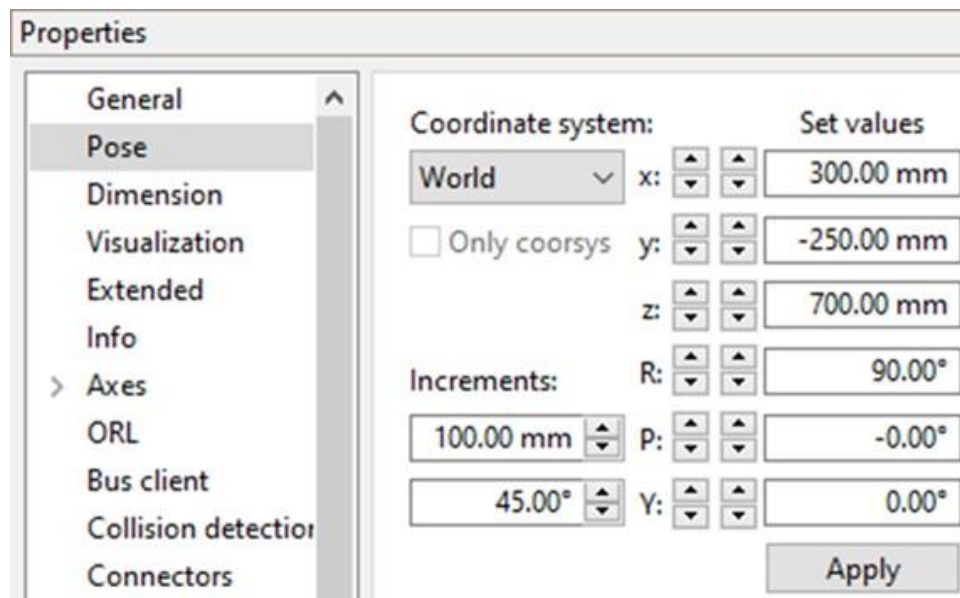


5.2.2 Information Sheet 2: Selection of Robot Model and End Effector

1. **Add Robot Stand:** Use the Modeling\Model Libraries menu to add StationTableWithProfi from the ModLibs\Racks And Tables library and position it.

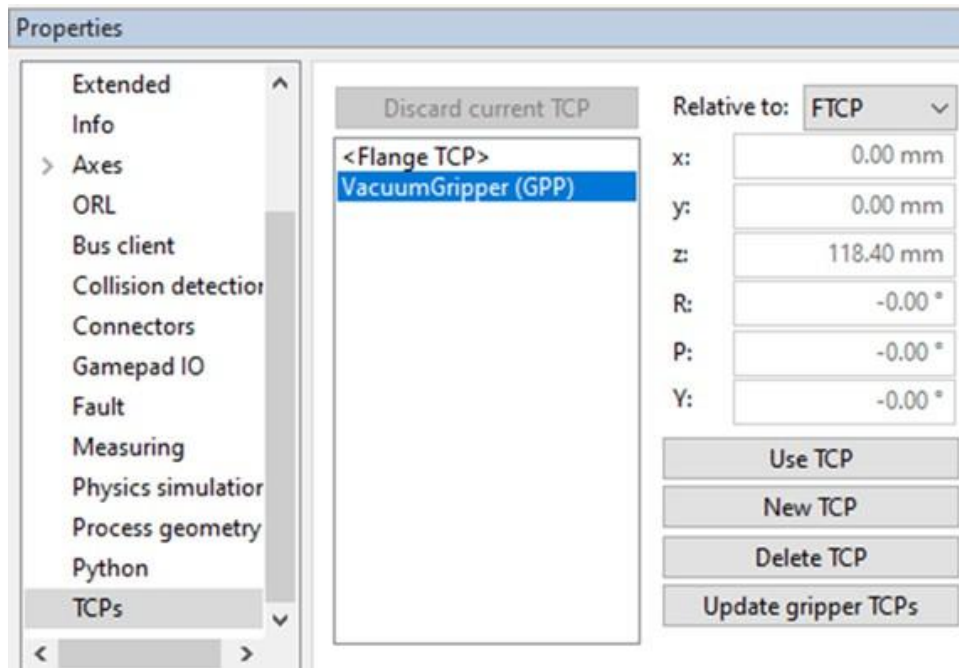


2. **Add Robot:** Open the Model Libraries dialog. Select the Mitsubishi robot **RV-2FL** from the Robots\Mitsubishi\F type library and add it to the work cell. Select the robot object in the Model Explorer and adjust its position using the **Pose** page in the Properties dialog.

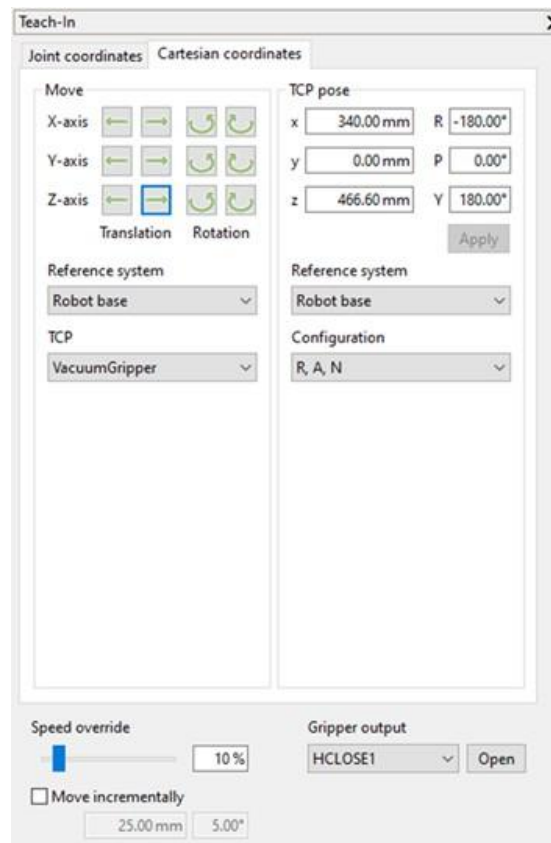


3. **Add Gripper:** Select the **Vacuum Gripper** from Grippers\Vacuum grippers\Vacuum gripper and add it. It is attached automatically to the robot's flange.

5.2 Learning Outcome: Create and Configure Simulation Environment



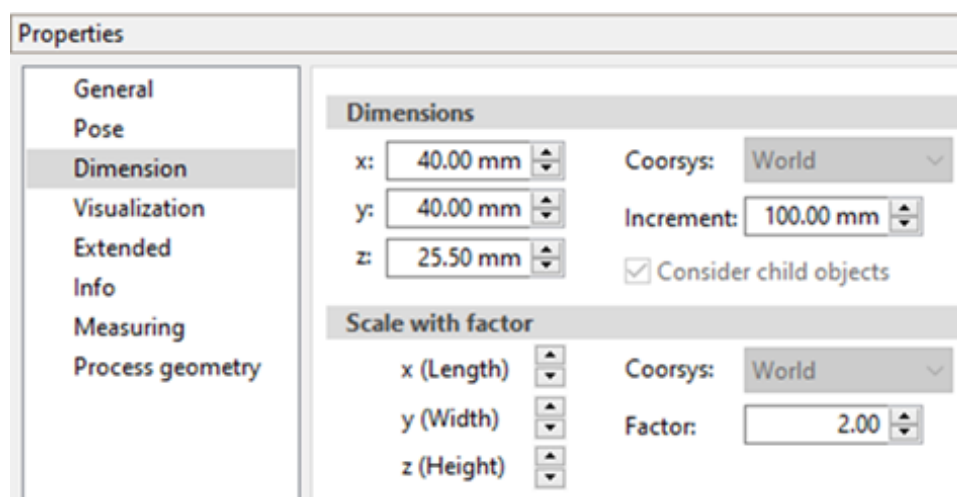
4. **Connect Gripper:** Select the robot (RV-2FL) in the Model Explorer. On the TCPs page of the Properties dialog, select the gripper (e.g., VacuumGripper(GPP)) and check Use TCP.
5. **Connect Gripper I/O:** Open the **Manual Operation** window (Modeling\Manual Operation). Drag a connection line from the robot output (e.g., **HCLOSE1**) to the gripper input (e.g., **Gripper Close**).



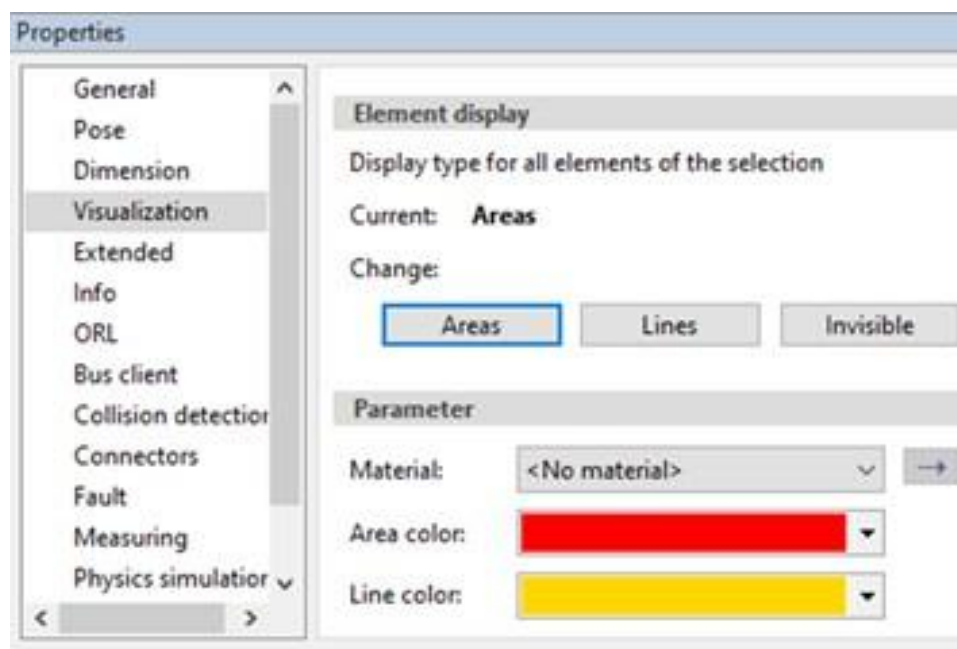
5.2.3 Information Sheet 3: Create Custom 3D Object

Create Workpiece Template (Product):

1. In the Model Explorer, select **Objects** and use the context menu New\Object. Rename it (e.g., **Product**).

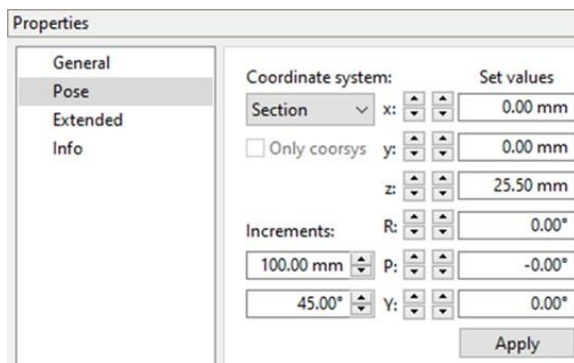


2. Create a section (Product\New\Section) and a geometric primitive (e.g., **Cylinder**). Configure the size and color via the **Dimension** and **Visualization** pages.

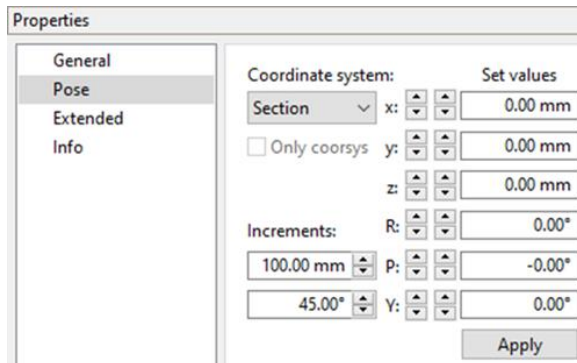


3. Create necessary **Grip points** on the object's base (e.g., **GPTop** and **GPBottom**).

5.2 Learning Outcome: Create and Configure Simulation Environment



GPTop

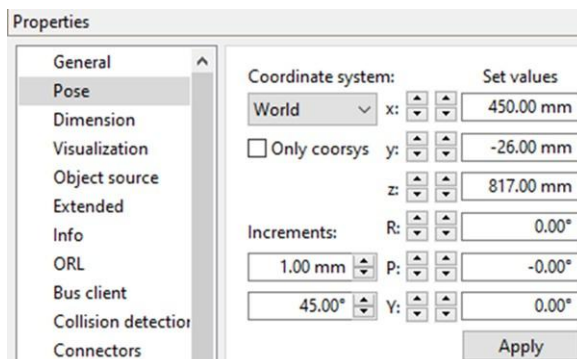
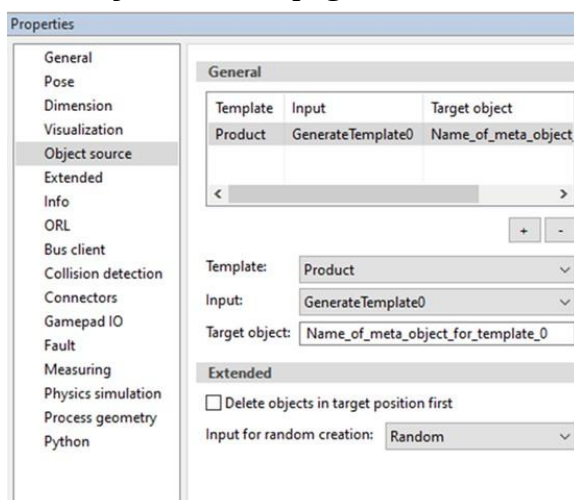


GPBottom

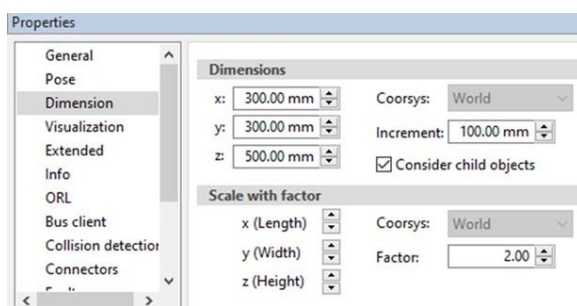
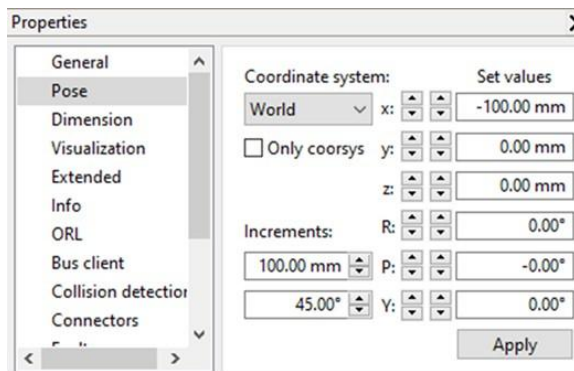
4. Drag the created workpiece object (**Product**) to the **Templates** item in the Model Explorer to register it as a template.

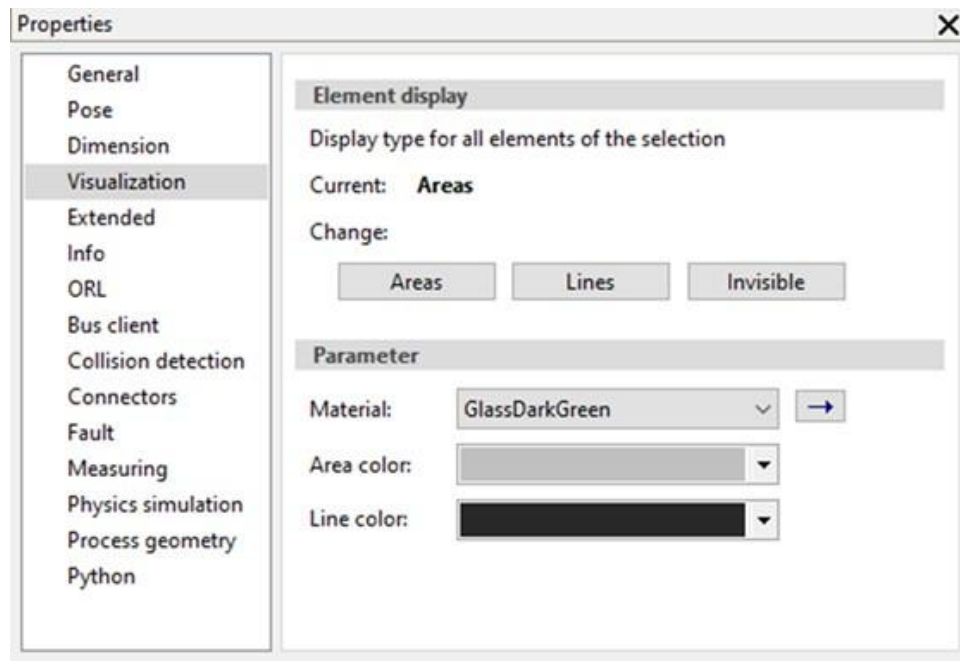
5.2.4 Information Sheet 4: Add and Configure Source/Sink Mechanisms for Pick and Place Simulation

1. **Replicator (Source):** Add the **Object source** mechanism from Modelingessentials\Source and sink. Rename it **Replicator** and position it. Configure its properties on the **Object source** page.

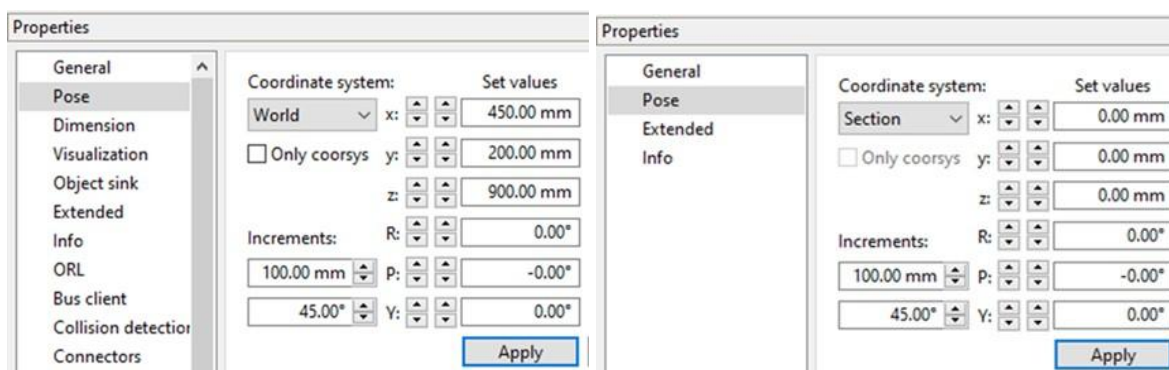


2. **Trash Can (Sink):** Create a trash can receptacle (e.g., **TrashCanPod**) using objects or sections. Add the **Object sink** mechanism from Modelingessentials\Source and sink to the work cell, drag it onto the TrashCanPod object, and rename it **TrashCan**.





3. **Define Target Position:** Add a grip point (e.g., **GPTrash**) to the **TrashCan** object to define the delivery position.

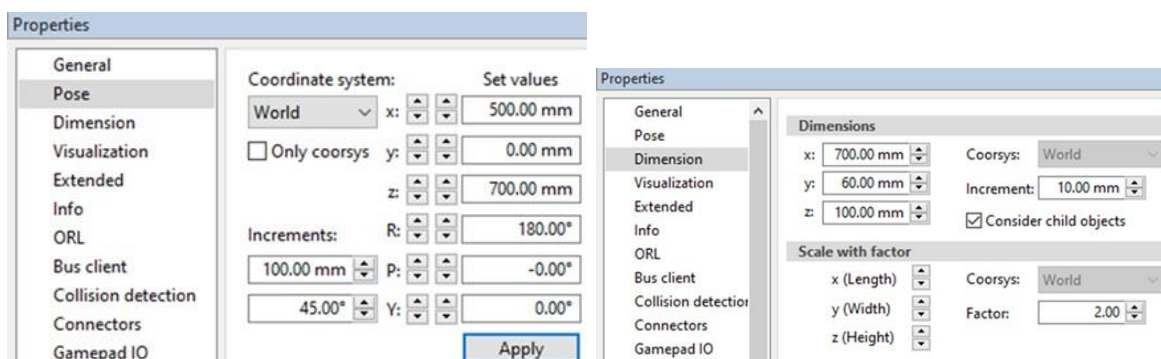


TrashCan Position

GPTrash Position

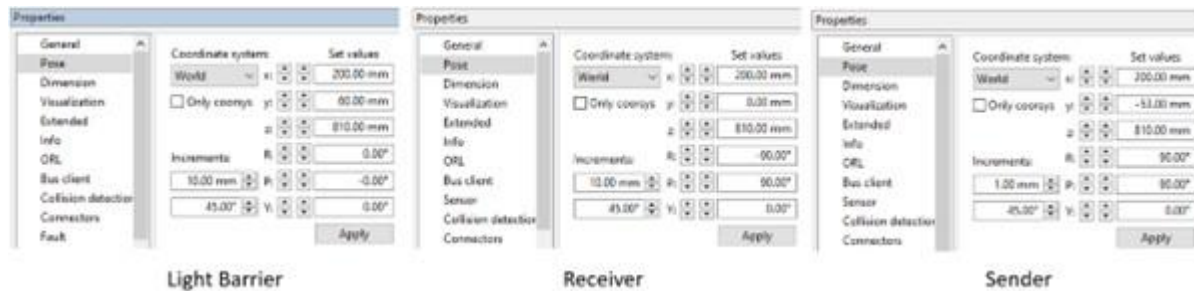
5.2.5 Information Sheet 5: Add Sensors and Conveyor

1. Add a **ConveyorBelt** from ModLibs\Conveyors and position it.



2. Add a **Light Barrier** sensor from Sensors\Light barrier and position its sender and receiver.

5.2 Learning Outcome: Create and Configure Simulation Environment



5.2.6 Job Sheet - Initial Setup and Robot Modeling in CIROS

Section	Content
Job Title	Create Initial CIROS Work Cell and Add Robot with Gripper
Personal Protective Equipment (PPE)	None required (software simulation environment).
Materials Required	CIROS Studio software; Mitsubishi RV-2FL robot model (from library); Vacuum gripper model (from library); Station table or profile-top workbench.
Tools and Equipment	PC with CIROS Studio installed; Mouse and keyboard for 3D navigation and configuration.
Procedure	1. Create a New Work Cell - Launch CIROS Studio and select File → New Work Cell. - Set the workspace size and define a suitable floor layout. - Save the work cell as RobotCell_Setup.cir. 2. Add and Position Models - From the object library, insert the station table into the work area. - Add the Mitsubishi RV-2FL robot from the robot library. - Place the robot accurately on the stand or floor reference. 3. Add the Gripper - Insert the vacuum gripper model from the library. - Attach it to the robot's Tool Center Point (TCP). - Verify correct orientation and alignment of the gripper.

Continued on next page

Module-5: Simulate Robotic Arm in Software

Section	Content
	4. Configure I/O Connections - Open the I/O Configuration panel. - Assign digital output channels to control the vacuum gripper. - Test manual ON/OFF operation to ensure communication.
	5. Save and Test Setup - Save the cell configuration. - Simulate robot motion to confirm stable model attachment and workspace limits.
Notes	- Always save incremental versions when modifying the work cell. - Maintain proper naming conventions for models and signals. - Check the robot's workspace reach to avoid collisions during simulation. - Use orthographic and isometric views for accurate placement.
Key Technical Points	- CIROS Studio enables virtual setup of industrial robot cells with real kinematic parameters. - Robot and gripper must be correctly linked to simulate real pick-and-place motion. - I/O mapping is essential for simulating control logic between robot and end-effector. - A well-configured work cell forms the base for subsequent programming and process simulation.

5.2.7 Job Sheet - Creating Custom 3D Workpiece and Environment

Section	Content
Job Title	Create Custom Workpiece, Source/Sink, Conveyor, and Sensor
Personal Protective Equipment (PPE)	None required (software simulation environment).
Materials Required	CIROS Studio software; 3D primitive library; Modeling Essentials components (Source, Sink, Light Barrier, ConveyorBelt); Registered robot cell from previous job.
Tools and Equipment	PC with CIROS Studio installed; Mouse and keyboard; 3D navigation tools within CIROS interface.

Continued on next page

5.2 Learning Outcome: Create and Configure Simulation Environment

Section	Content
Procedure	1. Create Custom 3D Workpiece - In the Model Explorer, right-click Objects → New → Object and rename it Product. - Add a geometric primitive (e.g., Cylinder) from the 3D library. - Adjust size, color, and position using the Dimension and Visualization tabs. - Add grip points (GPTop, GPBottom) to define handling positions. - Drag and drop the completed object into the Templates folder to register it as a reusable product.
	2. Add Source and Sink Mechanisms - Insert a Source from Modeling Essentials → Object Source and rename it Replicator. - Position it near the conveyor input area. - Create a new object TrashCanPod, add an Object Sink, and rename it TrashCan. - Define a grip point (GPTrash) on the TrashCan to represent the drop-off location.
	3. Integrate Conveyor and Sensors - Add a ConveyorBelt from ModLibs → Conveyors. - Insert a Light Barrier sensor and align sender and receiver across the conveyor path. - Adjust sensor position and height so that the product passes precisely through the detection zone.
	4. Save and Validate - Check alignment between product path, source, and sink. - Verify sensor detection by simulating product movement. - Save the project as Workpiece_Environment.cir.
Notes	- Always maintain consistent naming conventions for all objects and templates. - Use orthogonal and 3D views to ensure precise positioning. - Verify object hierarchy (Parent-Child relationships) in the Model Explorer. - Test detection range of the light barrier before connecting I/O signals.

Continued on next page

Section	Content
Key Technical Points	- Templates enable repetitive generation of identical objects in CIROS simulations. - Source and Sink mechanisms control product creation and removal automatically. - Light Barrier sensors perform non-contact object detection along the conveyor. - Grip points define precise pick-and-place positions essential for robotic accuracy. - Integrating all components creates a realistic production environment for simulation testing.

5.2.8 Self-Check Questions

No.	Question
1	After creating a new model file in CIROS Studio, what initial configuration step is required to define the scope of the virtual environment floor, and what size might be optionally set for the x-value?
2	If a user is setting up a standard robot cell environment, which specific Mitsubishi robot model and corresponding gripper are typically selected from the component libraries?
3	When configuring a newly added Vacuum Gripper to be the active tool, what setting must be selected in the TCPs page of the robot Properties dialog?
4	In the I/O Configuration panel, describe the data flow direction required to successfully link a robot control output (e.g., HCLOSE1) to the corresponding end effector input (e.g., Gripper Close).
5	A new object, such as a Cylinder, has been defined in the Model Explorer. What subsequent step must be performed on this object to allow the Replicator mechanism to generate copies of it during the simulation?
6	Why are Grip points (e.g., GPTop) defined on the base of a custom 3D workpiece, and what role do these points play in pick-and-place automation?
7	Identify the two mechanisms from the Modeling essentials library used to simulate the start and end of material flow in a pick-and-place process, and what descriptive names are conventionally assigned to them?
8	When modeling a Light Barrier sensor along a ConveyorBelt path, what two complementary sensor components must be positioned and aligned across the conveyor?
9	If a user creates a TrashCan object to simulate the disposal of parts, what specific type of object must be dragged onto the TrashCan object to enable it to functionally remove simulated objects?
10	How is Collision Detection typically activated and managed within the CIROS Studio environment during the configuration phase?

5.2.9 Answer Key

No.	Answer / Guidance
1	The floor must be configured by navigating to Settings → Model Options → Display → Floor and checking As an endless floor. Optionally, the x-value of floor size might be set to 2200 mm.
2	The Mitsubishi robot RV-2FL (from the Robots/Mitsubishi/F type library) is typically selected, and the corresponding end effector is the Vacuum Gripper (from Grippers/Vacuum grippers library).
3	The user must select the Vacuum Gripper (e.g., VacuumGripper(GPP)) in the TCPs page and check the Use TCP box.
4	A connection line must be dragged from the robot output (e.g., HCLOSE1) to the gripper input (e.g., Gripper Close) using the I/O Configuration panel or Manual Operation window.
5	A geometric primitive (e.g., Cylinder) must be created as a section (Product/New/Section) under the workpiece object to define its shape, size, and visualization.
6	Grip points (GPTop, GPBottom) are defined to mark handling positions on the object's base. These points define the precise coordinates the robot TCP must reach to accurately interact with (pick up or drop off) the workpiece.
7	The Object source mechanism (conventionally named Replicator) is used for material inflow, and the Object sink mechanism (conventionally named TrashCan) is used for material outflow/disposal.
8	The Light Barrier sensor requires both the sender and receiver components to be positioned and aligned across the conveyor path to detect the object passing between them.
9	The user must insert the Object sink mechanism from Modeling essentials onto the TrashCanPod object to enable object removal during simulation.
10	Collision Detection is activated through the Simulation Menu.

5.3 Learning Outcome: Generate Robotic Program

5.3.1 Information Sheet 1: I/O Configuration and Programming

3.1 Configure I/O Mapping: Inputs and outputs of the robot (RV-2FL) and connected devices must be renamed and linked via drag-and-drop in the **Model Explorer** to synchronize simulation behaviors.

Device Input/Output	Robot I/O (RV-2FL)	Description
Replicator Random 003	RV-2FL Replic 255	Trigger replicator generation.
Conveyor Belt OnOff 000	RV-2FL BeltOn 254	Control conveyor movement.
TrashCan RemoveAtGPP1 000	RV-2FL DelRed 253	Trigger object removal.
VacuumGripper Grasp 000	RV-2FL HCLOSE1 901	Control gripper activation.
RV-2FL WPAvail 006	LightBarrier.Sender Detect 000	Receive workpiece detection signal.

Note: To stop the conveyor when a product is detected, the LightBarrier output must be connected to the ConveyorBelt input, and the connection should be **inverted** using the context menu command Edit\Invert.

5.3.2 Information Sheet 2: Create Position List (Teaching)

1. Open the **Teach-In** window (Programming\Teach-In).



2. Manually move the robot arm to the required positions (e.g., home, pick position, drop position) using jog operations or by right-clicking a grip point (e.g., GPTop_...

or GPTrash) and selecting **Move Robot to**.

3. Save the current robot positions to a position list file (e.g., *.pos or *.psl).

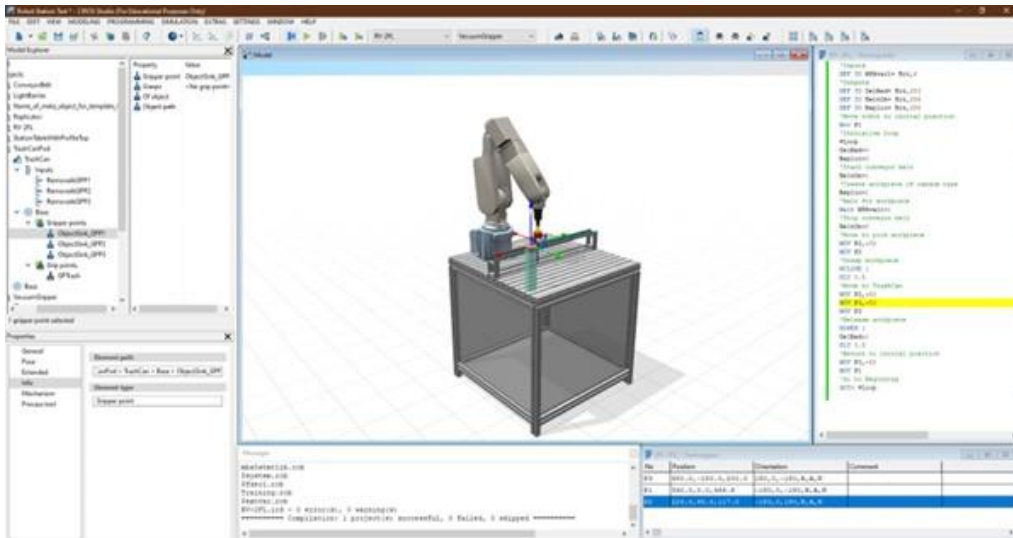
5.3.3 Information Sheet 3: Generate Robot Program (MELFA-BASIC V Example)

1. Create a new robot project (File\New\File) selecting **MELFA-BASIC V (MBA-V)** as the Project Type (e.g., named "Training").
2. Create a new robot program file (e.g., Training.mb5) within the project.
3. Write the robot motion and logic program, ensuring the program references the defined positions and controls the mapped I/O signals (e.g., HCLOSE1 901, DLY 1000).

```
'Inputs
DEF IO WPAvail= Bit,6
'Outputs
DEF IO DelRed= Bit,253
DEF IO BeltOn= Bit,254
DEF IO Replic= Bit,255
'Move robot to initial position
Mov P1
'Initialize Loop
*Loop
DelRed=0
Replic=0
'Start conveyor belt
BeltOn=1
'Create workpiece of random type
Replic=1
'Wait for workpiece
Wait WPAvail=1
'Stop conveyor belt
BeltOn=0

'Move to pick workpiece
MOV P2,-50
MOV P2
'Grasp workpiece
HCLOSE 1
DLY 0.5
'Move to TrashCan
MOV P2,-50
MOV P3,-50
MOV P3
'Release workpiece
HOPEN 1
DelRed=1
DLY 0.5
'Return to initial position
MOV P3,-50
MOV P1
'Go to Beginning
GOTO *Loop
```

5.3.4 Information Sheet 4: Test and Simulation



1. **Compile and Link:** Use the command Programming\Compile or Programming\Compile+Link (shortcut CTRL+F9) to check for syntax errors and translate the program into IR-DATA code.
2. **Start Simulation:** Execute the simulation using Simulation\Start / Stop.
3. **Monitor I/O:** Open the **I/O Monitor** window to track inputs and outputs.
You can force simulation states (e.g., force the Replicator to generate a part by setting Override active and Override value to True for the input GenerateTemplate0).
4. **Reset:** To reset all simulation elements, use Simulation\Reset Workcell or Simulation\Reset.

5.3.5 Information Sheet 5: Program Export

Programs created in **CIROS** for Mitsubishi robots generate the following output files:

- **Robot program file (.mb5)**
- **Position list file (.pos)**

These files contain the motion logic and corresponding taught positions required for execution on the physical Mitsubishi robot controller.

5.3.6 Self-Check Questions

No.	Question
1	What is the fundamental requirement for the robot program concerning its defined positions?
2	What is the name of the file type generated by CIROS Studio that contains the actual robot motion logic for Mitsubishi controllers?
3	How is the vacuum gripper activated and controlled within a MELFA-BASIC V robot program, and what command is used?
4	What command is typically used in MELFA-BASIC V to enforce a pause in the program flow to allow time for an I/O action (like grasping) to complete?
5	Describe the purpose of the Programming Menu command Compile +Link in CIROS Studio and what output file format is ultimately created.
6	What CIROS Studio functionality allows a user to manually check or manipulate the status of virtual inputs and outputs during program testing?
7	What does the instruction Mov PWAIT imply regarding the robot's motion path, contrasting with an instruction like Mvs PGET?
8	Why are the robot program file (.mb5) and the position list file (.pos) both necessary for program execution on a physical Mitsubishi robot controller?
9	When defining a motion stage such as approaching a pick point, what kind of MELFA-BASIC V command is used to ensure the robot moves linearly to a point offset from the target position?
10	How can a programmer verify that the robot execution is synchronized with external inputs/outputs (like sensors or a replicator) during simulation in CIROS Studio?

5.3.7 Answer Key

No.	Answer / Guidance
1	The robot program requires positions to be defined and stored in a position list file (.pos) before they can be referenced and executed in the motion logic.
2	The robot motion logic is written in the MELFA-BASIC V program file, which uses the extension .mb5.
3	The vacuum gripper is controlled using a dedicated Hand I/O signal. For example, HCLOSE1 (Hand Close 1) might be mapped to Output 901 to activate the grip.
4	The Dly command is used to introduce a time delay (measured in seconds) to allow the completion of I/O operations such as grasping or releasing the workpiece.
5	The Compile +Link command checks the MELFA-BASIC V program for syntax errors and translates it into IR-DATA code required for the robot controller.
6	The I/O Monitor window is used to track the status of inputs and outputs in real time, and allows the user to force simulation states (e.g., setting Override active and Override value to True to generate a part from the Replicator).
7	Mov PWAIT performs Joint interpolation (Mov), meaning the robot moves along a curved path defined by the joint angles. Mvs PGET performs Linear interpolation (Mvs), meaning the Tool Center Point (TCP) moves along a straight-line path.
8	The .mb5 file contains the motion logic and I/O commands, while the .pos file contains the actual coordinate data (taught positions) that the .mb5 program references. Both are required for execution.
9	The Mvs command with an offset is used, such as Mvs PGET, +20, which directs the robot to move linearly to a point 20 mm away from the target pick position (PGET).
10	The programmer must monitor the I/O Monitor window to track signals (e.g., WPAvail 006 from the LightBarrier) and use Wait instructions in the MELFA-BASIC V program to synchronize movement with input status.

5.4 Learning Outcome: Transfer Program File to Robotic Arm System

5.4.1 Information Sheet 1: Transfer Program File to Robotic Arm System

Integrate Simulation Software with PLC Programming Software:

CIROS simulation can be controlled using external PLC software to enable real-time interaction between virtual and physical automation components.

- **PLC SIM Advanced:** Enables Siemens PLC program simulation and signal exchange with CIROS.
- **CODESYS (Soft-SPS):** Provides software-based PLC environment for control and logic testing.
- CIROS can function as simulation software when linked using the **EasyPort USB** interface.
- Data transfer between process models in CIROS and an external PLC (or PLC simulator like **S7-PLCSIM** or **CODESYS Soft-SPS**) is typically achieved using the **EzOPC server**. EzOPC acts as a virtual switchboard connecting the signal ranges of the communicating partners, allowing synchronized data exchange and real-time process control.

5.4.2 Information Sheet 2: Transfer Program into the Robot Controller

Programs and position lists are transferred to the physical robot controller. Transfer steps often include:

1. Connecting the PC (running CIROS or RT ToolBox3) to the control unit (e.g., CR800-O2VP-S25) via a **network cable** (LAN socket on the front of the control unit).
2. Establishing a connection (e.g., using **TCP/IP** and entering the IP address of the robot controller).
3. Using a program management tool (such as **RCI Explorer** in CIROS) to upload files from the controller (if needed for backup or modification) or **download** programs and position lists to the robot controller.
4. Alternatively, programs can be written to the robot controller via RT ToolBox3 or backed up/restored using a USB memory stick via the Teaching Box (R56TB).

CIROS Studio is a sophisticated software system designed for the simulation and planning of automated processes, particularly those involving industrial robotics. This documentation outlines its features, requirements, and provides a detailed application example.

5.4.3 Job Sheet - Programming and Simulation of Pick-and-Place Robot in CIROS

Section	Content
Job Title	Programming and Simulation of Pick-and-Place Operation using CIROS and MELFA Basic
Personal Protective Equipment (PPE)	None required (software simulation environment).
Materials Required	CIROS Studio software; Mitsubishi RV-2FL robot model; Vacuum gripper; Conveyor belt; Replicator (Source); TrashCan (Sink); Light barrier sensor; Custom product (Cylinder) object.
Tools and Equipment	PC with CIROS Studio installed; Mouse and keyboard; MELFA Basic V programming environment for robot control.
Procedure	1. Configure Work Cell - Open CIROS Studio and load the robot work cell. - Add Mitsubishi RV-2FL robot, vacuum gripper, conveyor, and sensors to the scene. - Position components accurately using orthographic and 3D views. 2. Define Product and Material Flow - Create or import a custom 3D product model (Cylinder). - Register the model as a template and assign grip points (GP_Pick, GP_Place). - Add the Replicator (Source) near the conveyor input and the TrashCan (Sink) at the output. - Insert the ConveyorBelt and Light Barrier sensor along the transport path. 3. Configure I/O Connections - Map digital I/O channels between robot and external devices. - Assign output for vacuum ON/OFF control and inputs for sensor feedback. - Test manual I/O operation in CIROS to confirm functionality.

Continued on next page

Section	Content
	4. Teach Robot Positions - Use the robot's jog mode to move the TCP to pick and place points. - Save positions in the Position List (P1-P5) for different motion stages. - Verify reachability and clearance around the work area.
	5. Write and Execute MELFA Basic Program - Open MELFA Basic V editor and write logic for the pick-and-place sequence. - Include signal control, motion commands, and sensor interlocks. - Run the simulation and monitor part transfer from Source → Conveyor → Sink.
	6. Verify Operation and Record Simulation - Ensure full cycle execution without collisions. - Confirm correct synchronization between robot motion and I/O logic. - Save the simulation video using <i>Simulation → Record</i>.
Notes	- Confirm correct Tool Center Point (TCP) orientation before teaching pick/drop positions. - Use Manual Mode to debug programs before automation runs. - Save incremental versions to preserve working configurations. - This workflow serves as a base for multi-robot and sensor-based control systems.
Key Technical Points	- CIROS integrates robot, I/O, and process simulation within one platform. - MELFA Basic V enables precise sequencing using digital I/O and motion commands. - Source and Sink simulate realistic material inflow and outflow. - Light barrier sensors automate conveyor control via event-based triggers. - Taught robot positions define accurate motion paths for pick-and-place automation.

5.4.4 Self-Check Questions

No.	Question
1	Identify the two main types of output files generated by CIROS Studio that are necessary for executing a program on a physical Mitsubishi robot controller.
2	What is the primary role of the EzOPC server and EasyPort USB interface when transferring signals between a CIROS simulation and an external PLC simulator (e.g., S7-PLCSIM)?
3	Name two specific physical interfaces that CIROS Studio utilizes to establish a connection with a physical Mitsubishi robot controller for program transfer.
4	What common program management tool within CIROS Studio facilitates the transfer of robot programs and position lists between the PC and the physical controller?
5	Describe the process of loading a robot program onto a physical Mitsubishi robot controller using the Teaching Box (R56TB) as an alternative to CIROS Studio's RCI Explorer.
6	What is the purpose of the robot program file (.mb5), and how does it relate to the position list file (.pos)?
7	Explain the function of RCI Explorer when used for online debugging of a robot system.
8	Why is it important to synchronize the data exchange between a CIROS process model and an external PLC simulator?

5.4.5 Answer Key

No.	Answer / Guidance
1	The robot program file (.mb5) and the position list file (.pos) are the required files for execution on the physical Mitsubishi robot controller.
2	The EzOPC server acts as a virtual switchboard that utilizes the EasyPort USB interface to connect the signal ranges of the communicating partners (the CIROS model and the PLC simulator). This allows synchronized data exchange and real-time process control.
3	CIROS Studio can communicate with physical Mitsubishi robot controllers through either a serial interface (RS-232) or a network interface (TCP/IP).
4	The RCI Explorer (Robot Controller Interface) is the tool within CIROS Studio that enables the upload or download of robot programs and position lists between the PC and the physical robot controller.
5	Programs can be backed up or restored using a USB memory stick via the Teaching Box (R56TB).
6	The robot program file (.mb5) contains the motion logic and I/O commands, while the position list file (.pos) contains the actual taught coordinate data that the .mb5 program references.
7	The RCI Explorer manages the connection properties, allowing users to upload or download files and enabling online debugging capabilities.
8	Synchronized data exchange is necessary to ensure real-time interaction and process control between the virtual robotic process model in CIROS and the logic running on the external PLC controller or simulator.

Module-6: Perform Robotic Program

Module Description

This unit covers the knowledge, skills and attitudes required to perform robotic program. It specifically includes the tasks of identifying programming syntax, developing program for robot arm movement, developing program for controlling robot input and output and operating robotic arm.

Minimum Time: 50 Hours

Learning Outcomes

1. Programming syntax is identified.
2. Program for robot arm movement is developed.
3. Program for controlling robot input and output is developed.
4. Robotic arm is operated.

6.1 Learning Outcome: Identifying Robot Programming Syntax

6.1.1 Information Sheet: Core Terminology in Robotics

Payload: The maximum weight a robot can lift and move while maintaining its specifications such as accuracy and speed. This includes the weight of the workpiece or the end-of-arm tooling (EOAT).

Accuracy: The difference between the robot's commanded position and its actual attained position. It represents how close the robot gets to a specified target without correction.

Range / Reach: The maximum distance the robot can extend from its base within its work envelope, determined by link lengths and joint configurations.

Repeatability: The ability of a robot to return to the same programmed position multiple times. It measures consistency (random error) and is often more important than absolute accuracy in industrial settings.

Degrees of Freedom (DOF): The number of independent motions or axes a robot can perform. A typical industrial robot has six degrees of freedom.

Work Envelope: The three-dimensional space within which the robot can operate, defined by its structure and joint limits.

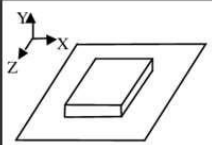
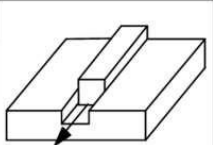
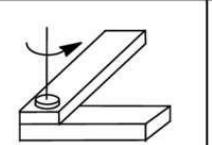
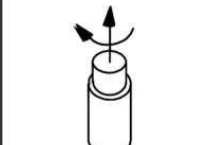
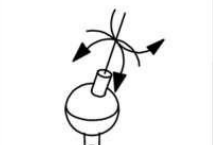
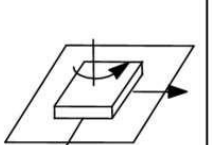
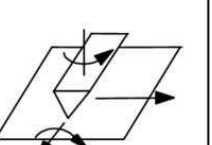
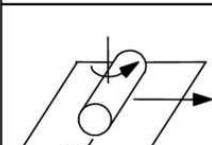
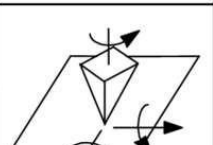
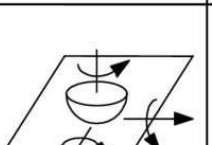
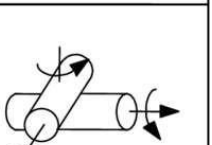
			
Rigid (no motion)	Prismatic (1)	Revolute (1)	Parallel Cylinders (2)
			
Cylindrical (2)	Spherical (3)	Planar (3)	Edge Slider (4)
			
Cylindrical Slider (4)	Point Slider (5)	Spherical Slider (5)	Crossed Cylinders (5)

Figure 6.1: Degrees of freedom in a kinematic system

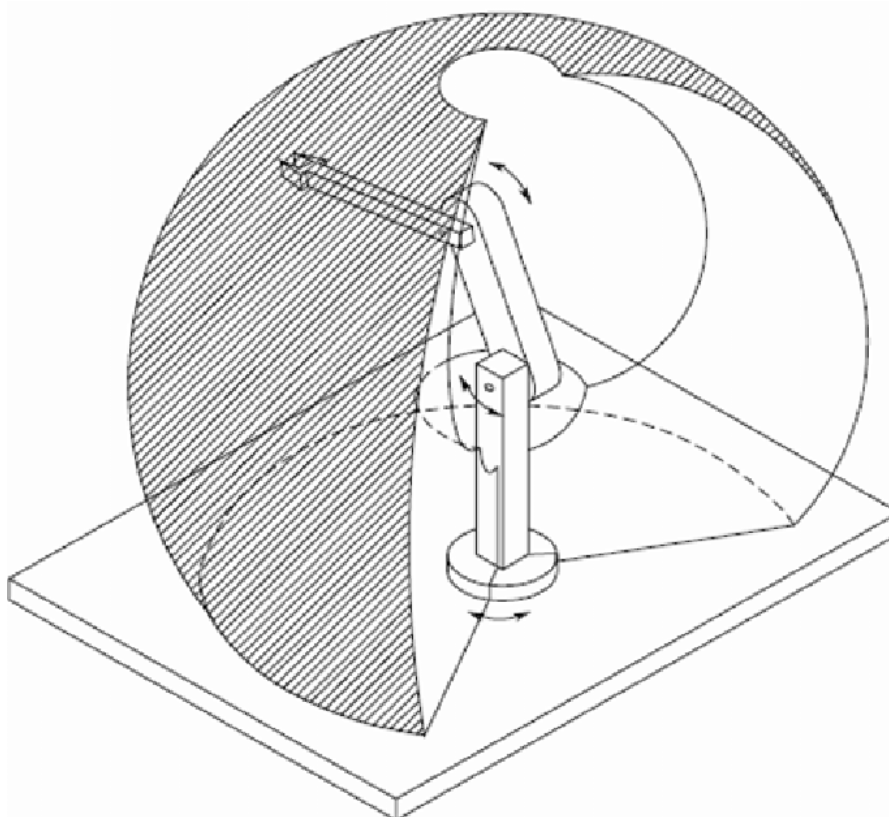


Figure 6.2: Work envelope of a robotic arm

End-Effector (EOAT): The tool or device attached to the robot's wrist to perform a specific task (e.g., gripper, welding torch, paint gun).



Figure 6.3: Different types of end-effectors

Controller: The “brain” of the robot system, consisting of the hardware and software that direct motion and respond to feedback.

Actuator: A motor or mechanism that produces motion in the robot's joints or links in response to control signals.

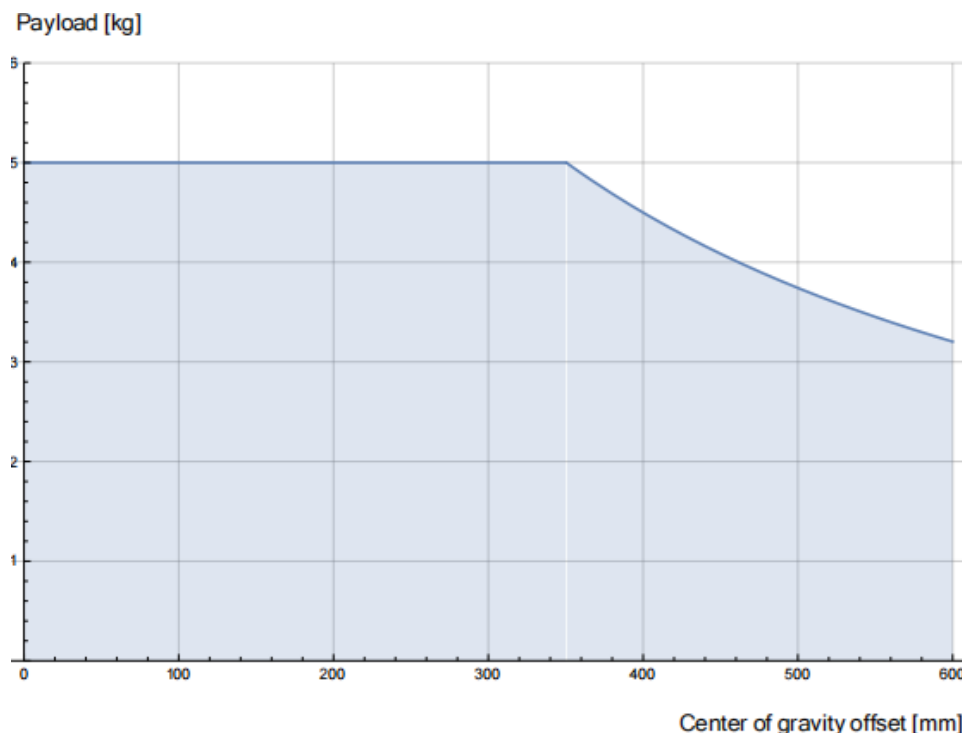


Figure 6.4: Relationship between Payload and Center of Gravity (CoG) for Robots

Forward and Inverse Kinematics

Forward kinematics (FK) calculates the end effector's position and orientation from the joint angles, while inverse kinematics (IK) calculates the required joint angles to reach a desired end effector position and orientation. A robotic arm controller solves inverse

kinematics (IK) by mathematically determining the joint angles required for the robot's end-effector (hand or tool) to reach a desired position and orientation in space.

Forward kinematics is simpler to compute but less practical for tasks like trajectory planning, whereas inverse kinematics is more complex but essential for tasks where the end effector's target position is known.

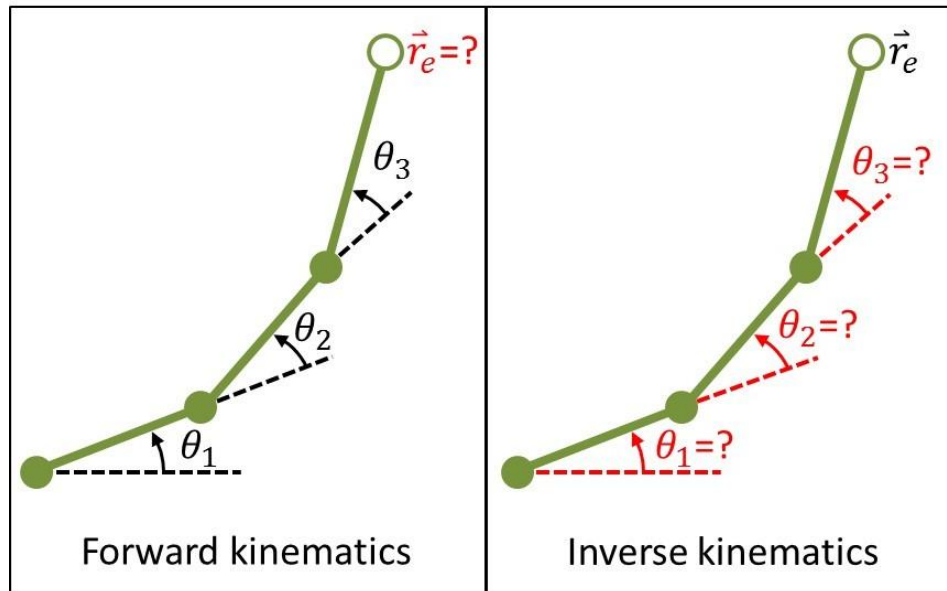


Figure 6.5: Forward Kinematics vs Inverse Kinematics

6.1.2 Information Sheet - Mitsubishi Robot Programming Language and Versions

The primary language used for programming Mitsubishi industrial robots, particularly with controllers like the CR800-D/R/Q series, is **MELFA-BASIC VI**. The specific language can be selected or confirmed using the parameter RLNG.

Syntax Rules of MELFA-BASIC (V)

MELFA-BASIC V programs are built from **statements**, which are the minimum configuration unit.

Syntax Element	Description & Rules
Statement Format	A statement is generally configured as: Step No. Command word Data [Appended statement].
Execution Flow	Steps are executed in ascending order based on the Step No.
Commands per Step	Only one command statement is permitted per step.
Direct Execution	A statement entered on the T/B without a Step No. is executed immediately upon input and is not saved in memory.
Command Words	Instructions specifying robot movement or tasks (e.g., Mov, Mvs, HOpen).
Appended Statements	Optional commands added to movement instructions (e.g., Wth, Wthlf).

6.1 Learning Outcome: Identifying Robot Programming Syntax

Naming Conventions	Variable names must begin with an alphabetical character; length limited to 32 characters.
Labels	Jump destinations are indicated by an asterisk (e.g., *LBL).
Comments	Use Rem or a single quotation mark (').
Component Access	A period (.) references subcomponents (e.g., P2.X).
Character Usage	Uppercase (A-Z) and numerals (0-9) are always valid; lowercase (a-z) may be used in variable names but not as the first letter of a program name.

6.1.3 Information Sheet - Variable Types and Characteristics

Variables store values, positions, or strings and are classified by data type (identifier).

Variable Type	Identifier	Data Contained	Notes on Use
Numeric Value	M (or others except P, J, C, W)	Integers, single/double-precision reals	Declared with Def Inte, Def Float, Def Double.
Position (XYZ/Work)	P or W	X,Y,Z coords & A,B,C posture data + flags (FL1, FL2)	Refers to robot coordinates in World CS.
Joint	J	Axis data (J1-J6 etc.)	Used for joint interpolation.
Character String	C	Text data	-
Input/Output	M_In, M_Out etc.	External signal references	Pre-defined system variables.

6.1.4 Information Sheet - Command Structures for Programming the Robotic Arm

Command Category	Command Word(s)	Function / Purpose
Movement Control	Mov	Joint interpolation move.
Movement Control	Mvs	Linear interpolation (straight-line path).
Movement Control	Mvc, Mvr, Mvr2, Mvr3	Circular interpolation movements.
Tool / Hand Control	HOpen, HClose	Opens / closes designated hand.

Module-6: Perform Robotic Program

Speed Control	Ovrd	Overall speed override (%).
Pallet Operation	Def Plt, Plt	Define / calculate pallet positions.

6.1 Learning Outcome: Identifying Robot Programming Syntax

I/O Operations	M_Out(), M_OutB()	Output signals to external devices.
Timing	Dly	Waits for designated time (seconds).

6.1.5 Information Sheet - Operators and Logical Conditions

MELFA-BASIC VI supports standard arithmetic, comparison, and logic operators.

Operator Class	Example	Meaning / Usage
Substitution	M1 = 1	Assign value.
Arithmetic	M1 = M1 + 1	Add, subtract, multiply, divide, power.
Integer Math	M1 = M1\3	Integer division or remainder.
Comparison	If M1>9 Then *L3	Equality/inequality tests.
Logical	And, Or, Not, Xor	Logical or bitwise operations.
Bit Shift	M1 = M1«2	Shift bits left/right.
Position Ops	P10 = P1 + P2	Coordinate arithmetic for positions.

6.1.6 Information Sheet - Continuous and Conditional Statements

Continuous Movement Cnt 1 enables continuous path; Cnt 0 ends it. Optional distances: Cnt 1, 100, 200.

Conditional and Repetition Structures

Structure	Usage
If-Then-Else	Multi-line conditional branching.
Select Case	Multi-way branching by expression.
For-Next	Counter-based loop (16 levels max).
While-WEnd	Condition-based loop (16 levels max).
GoTo	Unconditional jump to label.
On GoTo	Indexed conditional jump.

6.1.7 Self-Check Questions

No.	Question
1	What is the primary programming language utilized for Mitsubishi industrial robots, specifically with controllers like the CR800-D/R/Q series?
2	Identify the four key performance criteria that define Learning Outcome 6.1 (Programming syntax is identified).
3	In MELFA-BASIC V programming, what does the syntax element P typically represent, and in which coordinate system is it often defined?
4	Contrast the two ways a statement can be executed on the Teaching Box (T/B) in MELFA-BASIC: with and without a Step No..
5	What is the MELFA-BASIC V naming convention rule regarding the first character of a Variable Name?
6	Name three Command Words used in MELFA-BASIC for Movement Control.
7	What symbol is required in MELFA-BASIC V to indicate a Jump Destination (Label)?
8	In MELFA-BASIC V, what is the function of the Substitution Operator (=), and what is an example of its usage?
9	Besides Position and Joint variables, name two other Variable Types identified by an Identifier in MELFA-BASIC V.
10	How can a programmer include explanatory text in a MELFA-BASIC V program that does not execute during runtime?

6.1.8 Answer Key

No.	Answer / Guidance
1	The primary language is MELFA-BASIC VI.
2	The four performance criteria are: Correct syntax for defining variables is identified; Required command structures for programming the robotic arm are recognized; Proper use of operators and logical conditions is identified; and Structure for continuous and conditional statements is defined.
3	P represents Position variables (X,Y,Z coords & A,B,C posture data + flags). It refers to robot coordinates in the World Coordinate System.
4	A statement entered without a Step No. is executed immediately upon input and is not saved in memory. A statement with a Step No. is configured as Step No. Command word Data and is part of the program.
5	Variable names must begin with an alphabetical character.
6	Mov (Joint interpolation move), Mvs (Linear interpolation), and Mvc, Mvr, Mvr2, or Mvr3 (Circular interpolation movements) are used.
7	Jump destinations are indicated by an asterisk (e.g., *LBL).
8	The Substitution Operator (=) is used to assign a value to a variable. An example is M1 = 1.
9	Other variable types include Numeric Value (M or others except P, J, C, W), Character String (C), and Input/Output References (M_In, M_Out etc.).
10	The programmer can use the Rem command or a single quotation mark (') to define comments.

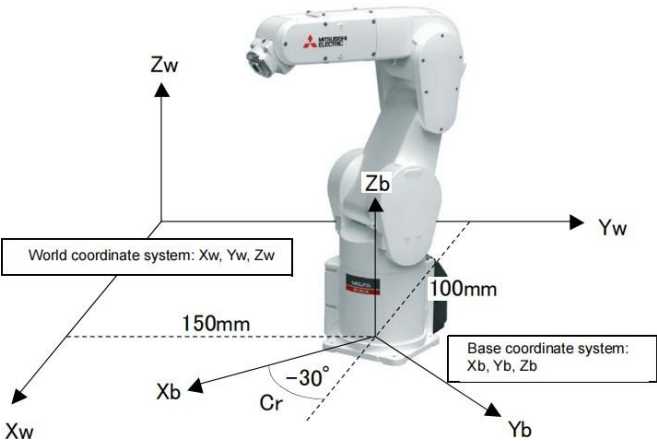
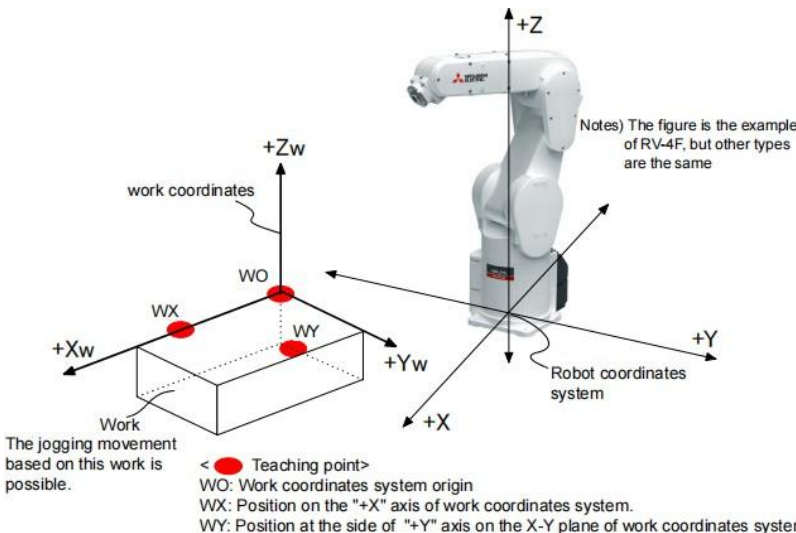
6.2 Learning Outcome: Develop Program for Robot Arm Movement

6.2.1 Information Sheet - Understanding Robot Coordinate Systems

The robot's movement and position teaching rely on several different coordinate systems. The selection of a coordinate system, typically done via the Teaching Box (T/B) jog mode selection, determines how the robot axes move when manually commanded.

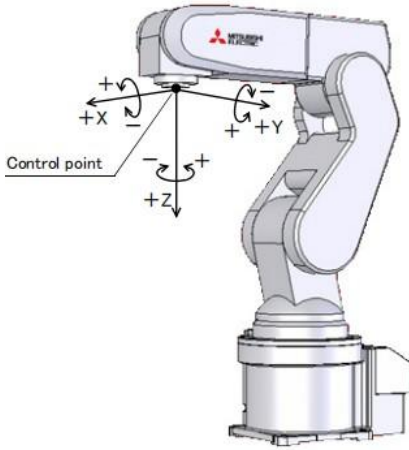
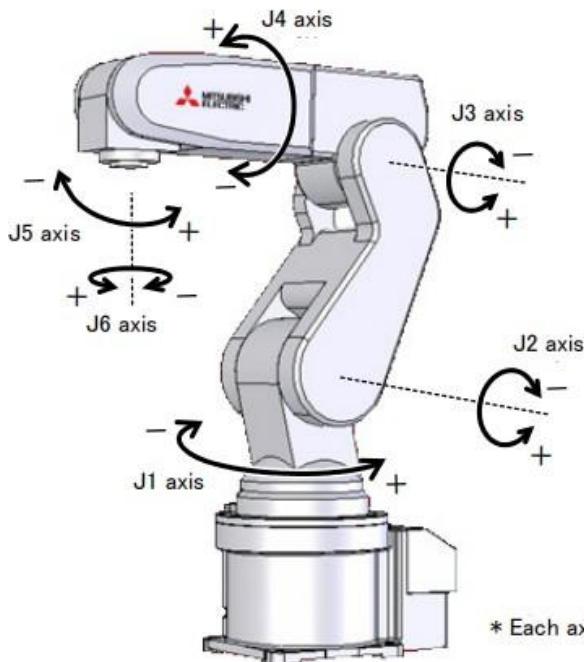
6.2 Learning Outcome: Develop Program for Robot Arm Movement

Table 6.8: Robot Coordinate Systems and Their Functions

Coordinate System	Purpose / Description (with Illustration)
World Coordinate System (X_w, Y_w, Z_w)	Base system for position control. In this mode, XYZ jog moves the tool or flange linearly relative to the global floor-fixed frame. 
Base Coordinate System (X_b, Y_b, Z_b)	Defines the robot installation orientation and reference relative to the mounting surface. All other systems are derived from this frame. The position of the world coordinate system is set to zero (0) before leaving the factory; therefore, the base coordinate system (robot's installation position) coincides with the world coordinate system.  Notes) The figure is the example of RV-4F, but other types are the same The jogging movement based on this work is possible. < Teaching point > WO: Work coordinates system origin WX: Position on the "+X" axis of work coordinates system. WY: Position at the side of "+Y" axis on the X-Y plane of work coordinates system. [Supplement] : The coordinate values which use all three teaching points for setting of the work coordinates system are each only X, Y, and the Z-axis. Although the coordinate value of A, B, and C axis is not used, positioning will get easy if the XYZ jog or TOOL jog movement is effected with the same value. (The direction of the hand is the same)

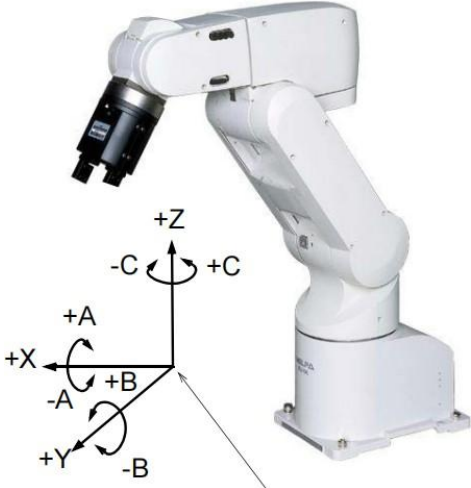
Continued on next page

Table 6.8 – continued from previous page

Coordinate System	Purpose / Description (with Illustration)
Tool Coordinate System (X_t , Y_t , Z_t)	Defined at the Tool Center Point (TCP). Jogging in TOOL mode moves linearly along or about the tool's local axes, useful for welding, painting, or assembly.  * While maintaining the flange surface posture, the axis moves straight along the tool coordinate system. Also, while maintaining the flange surface position, the flange surface posture changes.
Joint Coordinate System (J1 – J6)	Moves each robot joint independently. Each axis rotates about its mechanical center, useful for home position checks or maintenance.  * Each axis moves independently.

Continued on next page

Table 6.8 – continued from previous page

Coordinate System	Purpose / Description (with Illustration)
Work Coordinate System (X_w , Y_w , Z_w)	User-defined frame aligned to the workpiece, fixture, or pallet. Simplifies program reuse by offsetting motions relative to the part setup.  Work coordinate system: Coordinate system squared with the work, the working table, etc.

6.2.2 Information Sheet -Hand Jog via Teach Pendant Buttons



1. Teach mode switch (turn ON before teaching, programming)
2. Hand Open/Close indicator text
3. Hand jog button
4. +/- button controls hand open/close

Figure 6.6: Hand jog and manual open/close via teach pendant

6.2.3 Information Sheet - Defining the Robot's Starting (Home) Position

The designated initial position for the MPS system is defined by position variable j_Home.

- Ensure no collisions with station components.
- Axes should not be near end stops.
- Standard posture: Axis A = 180°, B = 0°, C = 0°.
- Update JSAFE parameter when j_Home changes.

Step-by-Step: Return to Home via T/B

1. Activate [TEACH] switch on T/B.
2. Press [MENU] → open main menu.
3. Select “PROGRAM” icon.
4. Mark program “UBP” → press “OPEN”.

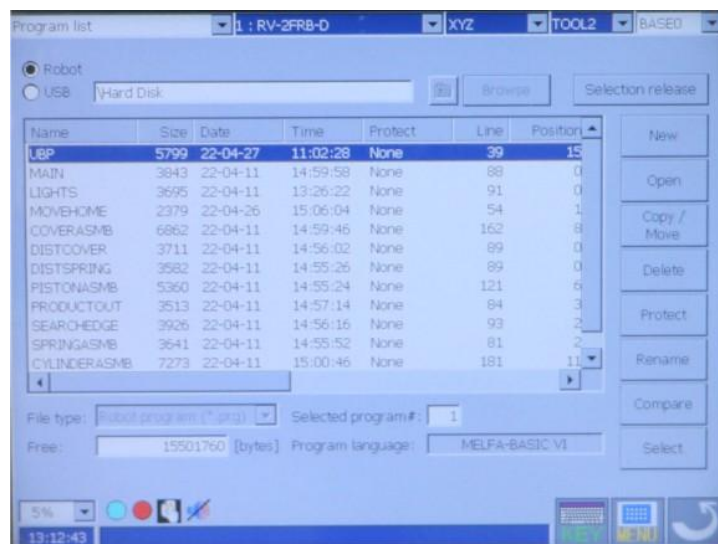


Figure 6.7: Navigating to UBP program on Teaching Box

5. Press “XYZ” (or Joint) to show positions.

6.2 Learning Outcome: Develop Program for Robot Arm Movement

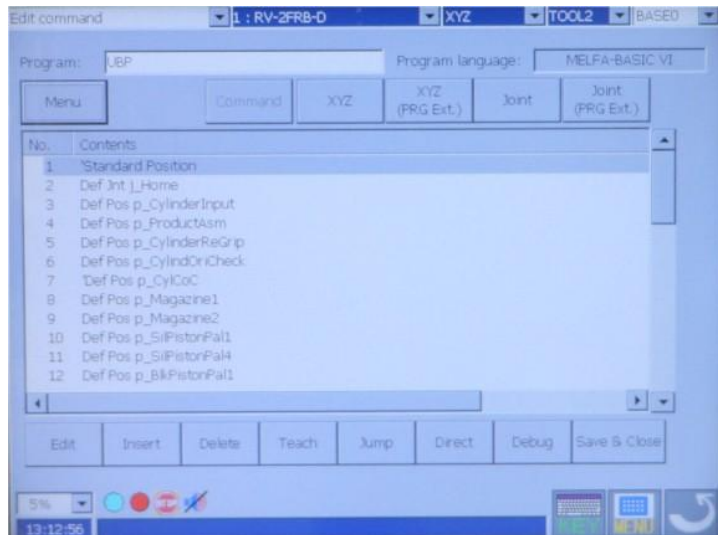


Figure 6.8: UBP program

6. Highlight variable **j_Home**.



Figure 6.9: UBP file pre-taught positions

7. Select correct TCP (TOOL1–TOOL4).
8. Hold enable button mid-position and press **[SERVO]** to turn servos ON.
9. Hold **[EXE]** to move robot to selected position.

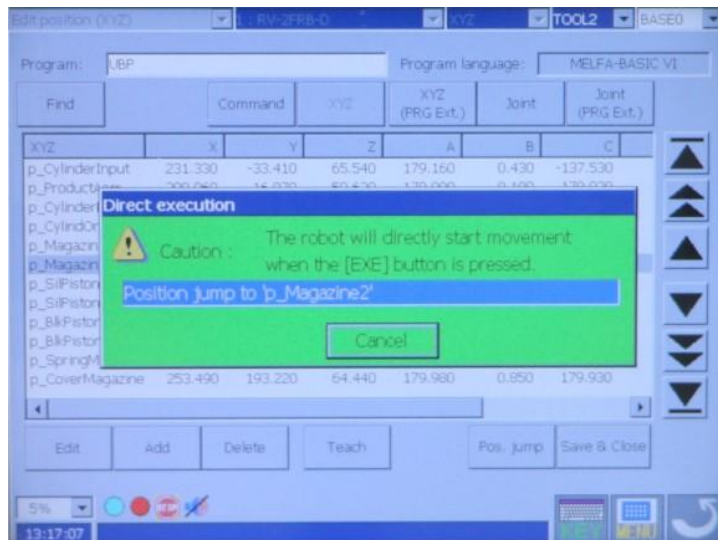


Figure 6.10: Position jump to any selected position

6.2.4 Information Sheet - Robot Jogging Using Teaching Box

Jogging is manual robot movement using the teach pendant (T/B) for position teaching and testing.

Jog Modes

- **Joint:** Moves each axis independently.
- **XYZ:** Moves TCP linearly in world coordinates.
- **Tool:** Moves TCP relative to tool orientation.
- **Work:** Moves relative to defined work coordinate frame.

Jog Operation Steps

1. Switch robot to **Teach Mode**.
2. Hold the enabling switch at mid-position.
3. Turn **Servo ON**.
4. Select jog mode (Joint/XYZ/Tool/Work).
5. Press arrow keys to move along selected axis.
6. Observe current coordinates in the display window.

6.2.5 Information Sheet - Defining Tool Center Point (TCP)

The Tool Coordinate System defines the control point (TCP) at the tool tip, required for Tool Jog and linear moves.

- Tool conversions configured via parameters (MEXTL1-4) or Tool command.

6.2 Learning Outcome: Develop Program for Robot Arm Movement

- Factory default = zero length (flange plane).

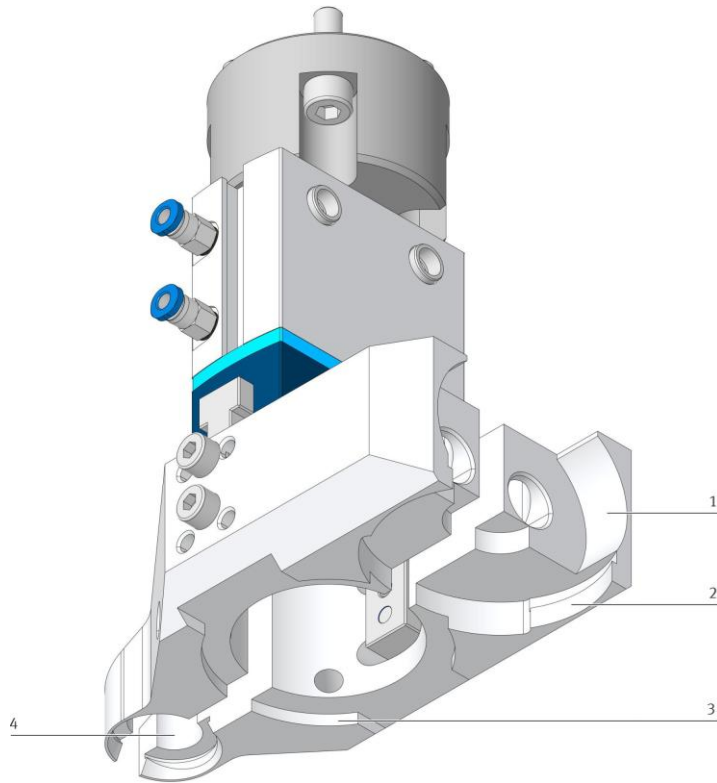


Figure 6.11: Teaching Box interface for returning robot to Home position

Tool Data for MPS Multifunction Gripper

No.	Designation	Coordinates (X,Y,Z,A,B,C)	Description
1	Center	(0.0, 0.0, 115.0, 0.0, 0.0, 0.0)	Gripping basic body / cover.
2	Outer	(40.0, 0.0, 110.5, 0.0, 0.0, 0.0)	Gripping outer faces.
3	Spring / Piston	(-40.0, 0.0, 117.5, 0.0, 0.0, 180.0)	Gripping pistons and springs.

Programming Example

Tool(0, 0, 115.0, 0, 0, 0) ' Sets TCP 115 mm along Z-axis of flange

6.2.6 Job Sheet - Perform Robot Jogging

Module-6: Perform Robotic Program

Section	Content
Job Title	Perform Robot Jogging using Teaching Box (T/B)
Objective	To operate the industrial robot arm using the Teaching Box (T/B), focusing on manual jogging, safety procedures, and verifying fundamental control mechanisms such as I/O activation and position control.
Personal Protective Equipment (PPE)	Safety gloves, safety glasses, and protective shoes must be worn inside the robot operation area. Ensure the safety fence and emergency stop are active before operation.
Tools and Equipment	Mitsubishi RV-2FL robot and CR800 controller; Teaching Box (T/B); Operator panel; Safety fence and emergency stop unit; Power supply and communication cables. Pneumatic air source for gripper actuation.
Procedure	1. Preparation and Safety Check - Ensure the work area is clear of any obstacles. - Power ON the robot controller and Teaching Box. - Verify that the emergency stop circuit is functional and safety fence is closed. - Confirm controller indicators show the system is ready.
	2. Manual Jogging Control - Set the key to Manual. - Turn the mode selector switch on the T/B to Teach. - Press Servo ON to activate the robot motion. - Use jog keys to move the robot to the home position (PHOME). - Jog the robot in small increments, especially when near objects, to prevent collisions.

Continued on next page

Section	Content
	3. Shutdown Procedure
	- Press Servo OFF to remove power from robot drives.
	- Power OFF the controller and secure the robot arm in its rest position.

Specification Sheet

Tasks and Required Sequences for Jogging Practice

Sequence Name	Description / Design Requirement
Safety Verification	All safety interlocks (fence, E-Stop, mode switch) must be confirmed functional before operation.
Jogging Modes	Jog the robot manually using the Teaching Box (T/B). This must include movement commands in both Joint and XYZ Coordinate Systems.
Home Position Return	Jog the robot back to a designated home position (PHOME) using the T/B jog keys.
End-Effector Control	Program/Test the I/O for controlling the end-effector (gripper). Verify tool hand open and closing functions.
Speed Adjustment	Vary the jogging speed setting via the T/B controls.
TCP Variation Analysis	Define and set different Tool Center Points (TCPs) for the robotic arm. Observe the resulting movement paths and operational differences when jogging in specific coordinate modes (e.g., Cartesian/XYZ) with varied TCP settings.

6.2.7 Information Sheet - Setting Joint Angles or Coordinates (Position Teaching)

Position variables (P, W, J) must be taught. Use JOINT Jog for coarse, XYZ/TOOL Jog for fine movement.

Register Current Position via T/B

1. Jog robot to desired point.
2. Open Program / Position Edit screen.
3. Display target variable (e.g., PGET).
4. Press **[F2 TEACH]** → confirm registration.

Programming Examples

Command	Purpose	Example
Position Constant	Define XYZ + ABC directly.	P1=(300,100,400,180,0,180)(7,0)
Joint Constant	Define J1-J6 angles.	J1=(0,10,80,10,90,0)
SetPos	Set position variable via args.	P2=SetPos(P1.X,P1.Y,M2)
SetJnt	Set joint variable (radians).	J3=SetJnt(M0,M0,M90,M0,M90,M0)

6.2.8 Information Sheet - Teaching Pre-Programmed Positions

Procedure

1. Jog robot to desired position.
2. In the position edit screen, select a position variable.
3. Press [**F2 TEACH**] to register the current coordinates.
4. Verify saved position by viewing the Position Data screen.

Recommended Positions

- **PWAIT:** Waiting position.
- **PGET:** Pick position.
- **PPUT:** Place position.
- **PHOME:** Safe home position.

6.2.9 Information Sheet - Programming Motion Paths

A. Joint Interpolation (Mov) Moves robot with joint-axis interpolation.

- 1 Mov PWAIT
- 2 Mov PGET, -50

B. Linear Interpolation (Mvs) Straight-line motion of TCP.

- 3 Mvs PGET
- 4 Mvs PGET,+20

C. Continuous Movement (Cnt)

Cnt 1	'Start continuous mode
Cnt 0	'End continuous mode

Cnt 1,100,200 'Set approach/exit distances

Module-6: Perform Robotic Program

6.2.10 Information Sheet - Program to Control Speed and Acceleration

Speed Commands

Command	Unit	Use	Example
Ovrd	%	Overall speed override	Ovrd 50
Spd	mm/s	Linear/circular speed	Spd 120
JOvrd	%	Joint interpolation speed	JOvrd 70

Acceleration Control

Command	Function	Example
Accel	Set accel / decel as %.	Accel 70, 70
Accel (no arg)	Restore to default.	Accel

6.2.11 Information Sheet - I/O Programming for End-Effector (Parallel Gripper)

The MPS station uses a parallel pneumatic gripper controlled by robot I/O.

Cmd	Function	Target	Example
HClose	Close hand	1-4	HClose 1
HOpen	Open hand	1-4	HOpen 1
Dly	Delay (s)	-	Dly 1.0
M_Out()	Output bit	Signal No.	M_Out(10)=0

6.2.12 Information Sheet - Example Pick-and-Place Program

Defined Positions

Variable	Description	Taught?
PWAIT	Wait position	Yes
PGET	Pick position	Yes
PPUT	Place position	Yes

Program Sequence

Step	Command	Description
1	Mov PWAIT	Move to wait position.

6.2 Learning Outcome: Develop Program for Robot Arm Movement

2	Mov PGET,+20	Approach pick position (+20 mm).
3	Mvs PGET	Move to grasp position.
4	HClose 1	Grasp workpiece.
5	Dly 1.0	Wait 1 s for grip.
6	Mvs PGET,+20	Retreat 20 mm.
7	Mov PPUT,+20	Approach place position.
8	Mvs PPUT	Place workpiece.
9	HOpen 1	Release workpiece.
10	Dly 1.0	Wait 1 s for release.
11	Mvs PPUT,+20	Retreat upward.
12	Mov PWAIT	Return to wait position.
13	End	End program.

6.2.13 Information Sheet - Executing a Pre-Written Robot Program

Program Execution Steps

1. Load the desired program (e.g., PickPlace.mbp).
2. Turn the **Servo ON** switch.
3. Switch to **AUTO Mode**.
4. Press **START** on the operator panel or pendant.
5. Observe robot motion and verify end-effector operation.
6. Press **STOP** or **Emergency Stop** if an abnormal condition occurs.

6.2.14 Information Sheet - Manual Execution for Debugging

- Use **STEP** mode to execute line by line.
- Monitor I/O status using the **Monitor** → **I/O Signal** screen.
- Verify correct position variables and tool selection.
- Modify commands in T/B as needed.

6.2.15 Information Sheet - Error Handling and Alarms

Typical Error Codes

Error Code	Meaning / Description
1010	Servo power supply fault.

6.2 Learning Outcome: Develop Program for Robot Arm Movement

2001	Overtravel limit triggered.
2020	Position data not found.
3050	I/O communication fault.
6001	User program syntax error.

Steps for Recovery

1. Read alarm number on T/B display.
2. Check associated cause in the robot's alarm list.
3. Remove cause (e.g., clear obstruction, check wiring).
4. Reset alarm using **[RESET]** key.
5. Re-enable servo and resume operation.

6.2.16 Job Sheet - Program Robot to Pick and Place Workpiece

Section	Content
Job Title	Program Robot to Pick and Place Workpiece using Teaching Box (T/B)
Objective	To operate a robotic arm, teach necessary positions, and execute a program to perform an integrated pick-and-place motion, demonstrating the control of robot movement and gripper actuation.
Personal Protective Equipment (PPE)	Safety gloves, safety glasses, and protective shoes must be worn inside the robot operation area. Ensure the safety fence and emergency stop are active before operation.
Tools and Equipment	Mitsubishi RV-2FL robot and CR800 controller; Teaching Box (T/B); Computer with MELFA Basic software; Pneumatic air source for gripper actuation.

Continued on next page

Section	Content
Procedure	1. Preparation and Safety Check 1. Ensure the work area is clear of any obstacles. 2. Power ON the robot controller and Teaching Box (T/B). 3. Verify that the emergency stop circuit is functional and safety fence is closed. 4. Confirm controller indicators show the system is ready.
	2. Manual Jogging and Position Teaching 1. Turn the mode selector key on the controller to Manual. 2. Turn the mode selector switch on the T/B to Teach. 3. Press Servo ON to activate robot motion. 4. Use jog keys (Joint, XYZ, Tool, or Work mode) to move the robot arm. 5. Move the robot to the Home position (PHOME). 6. Jog the robot to the required points and press [F2 TEACH] to register three positions: PWAIT, PGET, and PPUT. 7. Store these positions in the robot's memory.
	3. Program Development for Tool Control and Motion 1. Define the Tool Center Point (TCP) in the controller memory using the coordinates for the multifunction gripper (e.g., Tool No. 1). 2. Write the MELFA Basic program sequence using commands such as Mov (Joint interpolation) and Mvs (Linear interpolation) to move the robot between taught positions. 3. Integrate Hand I/O control using HClose 1 (to grasp the workpiece) and HOpen 1 (to release the workpiece). 4. Insert the Dly command after gripper actuation to allow time for the mechanical action to complete.

Continued on next page

Section	Content
	4. Program Execution and Verification 1. Load the program (e.g., PickPlace.mbp) using the Teaching Box or PC. 2. Execute the program in AUTO mode. 3. Observe the robot's pick-and-place motion and gripper I/O activation. 4. Monitor for any alarms or abnormal behavior.
	5. Shutdown Procedure 1. Stop the program and press Servo OFF. 2. Reset alarms if present. 3. Power OFF the controller and secure the robot arm in its rest position.
Notes	Never enter the robot's motion range while it is powered or in auto mode. Jog the robot in small increments near objects to prevent collisions.

Specification Sheet - Pick and Place Program Requirements

Key Technical Requirements and Parameters

Parameter	Requirement / Configuration
Required Positions	PHOME (Safe home position); PWAIT (Waiting position); PGET (Pick position); PPUT (Place position).
Gripper Control	Hand 1 is used for gripper actuation (HClose 1 to grasp, HOpen 1 to release).
Tool Definition (TCP)	TCP must be defined (e.g., Tool No. 1) for the gripper based on its coordinates (e.g., X=0.0, Y=0.0, Z=115.0).
Pick/Place Speed	Approach and final movements must be controlled. Linear speed (Spd) can be specified (e.g., Spd 120 mm/s). Joint override (Ovrd) sets overall speed percentage.
Motion Path for Approach	Linear interpolation (Mvs) is used to move straight to the pick/place position from an approach offset (e.g., Mvs PGET,+20).
Motion Path for Retreat	After grasp/release, use linear movement (Mvs) with a positive offset to retreat safely (e.g., Mvs PGET,+20).
Program Flow Control	Use Dly 1.0 (Delay 1.0 second) after gripper commands (HClose/HOpen) for reliable mechanical action completion.

6.2.17 Job Sheet - Debugging and Verification of MELFA Basic Program

Section	Content
Job Title	Perform Debugging on MELFA Basic Pick and Place Program using Teaching Box (T/B)
Objective	To debug and verify a robot program written in MELFA Basic, ensuring correct step-by-step execution, precise position alignment, synchronized I/O control for picking and placing a workpiece, and demonstrating proficiency in manual operation commands like jumping to taught positions.
Personal Protective Equipment (PPE)	Safety gloves, safety glasses, and protective shoes must be worn inside the robot operation area. Ensure the safety fence and emergency stop are active before operation. Never enter the robot's motion range while it is powered or in auto mode.
Tools and Equipment	Mitsubishi RV-2FL robot and CR800 controller; Teaching Box (T/B); Computer with MELFA Basic software; Pneumatic air source for gripper actuation.

Continued on next page

Section	Content
	1. Preparation and Safety Check
	1. Power ON the robot controller and Teaching Box. 2. Verify that the emergency stop circuit is functional and safety fence is closed. 3. Ensure all necessary positions (PWAIT, PGET, PPUT, PHOME) have been accurately taught and stored in the robot's memory.
	2. Program Loading (Sample Pick and Place Program)
	<i>Use the Teaching Box or PC to load a sample MELFA Basic program (e.g., PickPlace.mbp). A typical sequence involves movement, gripper actuation, and delays.</i>
	3. Step-by-Step Debugging and Verification (STEP Mode)

Procedure

1. Turn the mode selector switch on the T/B to **Teach**.
2. Press **Servo ON** to activate robot motion.
3. Switch the execution mode on the T/B to **STEP** mode.
4. Use the execution button (e.g., START/EXE) to execute the program one line at a time.
5. **Verify Motion Steps (Mov/Mvs):** For each movement instruction, confirm the robot follows the desired path (joint or linear interpolation). Jog the robot in small increments near objects to prevent collisions.
6. **Verify I/O Control (HClose/HOpen):** Observe the gripper actuation and monitor the I/O status using the Monitor → I/O Signal screen in the T/B to ensure the expected signals are active. The delay (Dly) commands must be sufficient for the mechanical action to complete.

Continued on next page

Section	Content
	4. Position Jump and Manual Adjustment
	1. If debugging reveals incorrect alignment, use the Manual Jogging function to move the robot arm slightly. 2. To jump to a predefined position, navigate to the Program / Position Edit screen on the T/B, highlight the variable (e.g., PGET), and hold [EXE] to move the robot to the selected position, ensuring the enabling button is held at mid-position. 3. If the current position needs adjustment, jog the robot to the desired point and press [F2 TEACH] to register the new coordinates to the corresponding position variable. 4. If needed, use the Forced output function via MENU → Monitor → Signal monitor → General-purpose signal to manually overwrite output status for testing.
	5. Program Execution (AUTO Mode)
	1. Once debugging is complete, turn the mode selector switch to AUTO. 2. Press START on the operator panel or T/B to execute the full program. 3. Observe the robot's motion and monitor for alarms or abnormal behavior.
	6. Shutdown Procedure
	1. Stop the program and press Servo OFF. 2. Reset alarms if present. 3. Power OFF the controller and secure the robot arm in its rest position.

6.2.18 Self-Check Questions

No.	Question
1	Identify the four key performance criteria that define Learning Outcome 6.2 (Develop Program for Robot Arm Movement).
2	What MELFA-BASIC V instruction is used for programming linear interpolation (straight-line) motion of the Tool Center Point (TCP)?
3	Contrast the movement path generated by the Mov command versus the Mvs command in MELFA-BASIC V.
4	Which MELFA-BASIC command is used to set the overall speed override as a percentage?
5	Describe the function of the Cnt command and its purpose when programming a sequence of movements.
6	In MELFA-BASIC V, what is the purpose of the TCP Data Configuration, and what specific information does it define?
7	When programming a motion stage to approach a pick position (PGET) but stopping 20 mm away, how would this be written using MELFA-BASIC V?
8	Name two coordinate systems used for motion teaching and movement control in a Mitsubishi robot, typically selected via the Teaching Box (T/B).
9	What is the function of the HOpen command, and what should immediately follow it in the program sequence?
10	How can a programmer define Position Constants (P) directly within a MELFA-BASIC V program without using the Teaching Box?

6.2.19 Answer Key

No.	Answer / Guidance
1	The four key performance criteria are: Robot's starting position is defined in the program; Appropriate joint angles or coordinates are set for arm movement; Motion paths for linear or rotational movements are programmed; and Speed and acceleration parameters are specified for each movement.
2	The MELFA-BASIC V instruction used for programming linear interpolation (straight-line TCP motion) is Mvs.
3	The Mov command performs Joint interpolation, resulting in a movement path defined by joint axes (a curved path). The Mvs command performs Linear interpolation, resulting in a straight-line motion of the TCP.
4	The command used to set the overall speed override as a percentage is Ovrđ.
5	The Cnt command enables Continuous Movement (Cnt 1 starts it) or ends it (Cnt 0 ends it). This allows the robot to move without stopping precisely at intermediate points, often used with approach/exit distances for smoother motion.
6	TCP Data Configuration defines the Tool Coordinate System at the tool tip, which is required for Tool Jog and linear moves. It specifies the X, Y, Z coordinates and A, B, C posture data relative to the robot flange.
7	This motion stage would be written as Mov PGET, +20 or Mvs PGET, +20. The +20 specifies an offset of 20 mm from the target position.
8	Two coordinate systems used for motion control are the World Coordinate System (XYZ) and the Joint Coordinate System (JOINT). (Also accepting Tool or Work Coordinate System).
9	The HOpen command is used to Open the designated hand (gripper). It should be followed by a Dly command to ensure a time delay for the gripper action to complete.
10	Position Constants (P) can be defined directly by specifying XYZ and ABC posture data within the program, such as P1 = (300,100,400,180,0,180)(7,0).

6.3 Learning Outcome: Develop Program for Controlling Robot Input and Output

6.3 Learning Outcome: Develop Program for Controlling Robot Input and Output

6.3.1 Information Sheet - Understanding Input and Output Devices in MPS Robot Station

- **Controller Capacity:** Mitsubishi CR800 has up to 256 Inputs and 256 Outputs. Optional I/O cards can expand this capacity (e.g., 32/32).
- **I/O Signal Types:** Dedicated I/O, General-purpose I/O, and Hand I/O.
- **Physical Interface:** Devices connect via the Robot Interface Box (RIA box), distributing digital I/O.
- **Gripper Dedicated I/O:** 6 Inputs and 6 Outputs (900-907) are reserved for gripper control.
- **MPS Components:** Conveyor, Robot Handling Module, and Operator Panel are I/O-mapped.

Input Device		Signal Name	Controller No.	Input
Workpiece Sensor	Orientation	m_SensOriCheck	I1	
Conveyor Belt Start Sensor		m_SensBeltFront	I2	
Spring Present at Pickup		m_SprAtPickup	IA	
Start Button (Panel)		m_StartButton	I3	
Output Device		Signal Name	Controller No.	Output
Start Button LED		m_LedStart	Q0	
Conveyor Motor Forward		-	Q1 (typical)	
Status Light (Panel)		m_LedQ1	Q2	
Open Gripper (Solenoid Valve)		GR1-1	900 (Hand Output)	

6.3.2 Information Sheet - Controlling Input and Output Devices

Controlling Input Devices

Short Notes

- Input signals are read using robot status variables (M_In()).
- M_In(bit_no) returns 1 for ON, 0 for OFF.
- Use Wait instruction for synchronization.

Module-6: Perform Robotic Program

- Advanced functions include Get-Position-Quick (Def Gps, GpsChk).

6.3 Learning Outcome: Develop Program for Controlling Robot Input and Output

Programming Example: Detect Switch / Proximity Sensor

Step	Command	Description
10	*WAIT_SENSOR	Label for looping.
20	Wait M_In(2)=1	Wait until input 2 ON.
30	Mov P_Pick_Approach	Move to approach position.
40	If M_In(2)=1 Then *PROCESS_OK	Jump if still ON.
50	GoTo *WAIT_SENSOR	Loop until detected.
60	*PROCESS_OK	Detection confirmed.

Controlling Output Devices

Short Notes

- Outputs are written using M_Out().
- ON = 1, OFF = 0.
- Combine with Dly for pulse signals.

Example: Control Conveyor, Cylinder, and LEDs

Step	Command	Description
10	M_Out(0)=1	Turn ON Start Button LED.
20	Dly 1.0	Wait 1 second.
30	M_Out(0)=0	Turn OFF LED.
40	M_Out(1)=1	Start Conveyor Motor.
50	Wait M_In(15)=1	Wait for end sensor.
60	M_Out(1)=0	Stop Conveyor.
70	M_Out(50)=1	Extend Cylinder.
80	Dly 0.5	Wait for motion completion.
90	M_Out(50)=0	Stop Cylinder Extend.

Force ON/OFF Output Devices via Teaching Box

Step-by-Step: Forcing General-Purpose Output

1. Set mode to **MANUAL**. Open **MENU** → **Monitor**.
2. Select “**Signal monitor**”.

3. Choose “General-purpose signal”.

6.3 Learning Outcome: Develop Program for Controlling Robot Input and Output

4. Tap “**Forced output**”.
5. Enter head signal number, press **Set**.
6. Toggle desired ON/OFF bits.
7. Execute using **Bit Forced Output** or **Port Forced Output**.

Monitor I/O Status in T/B

Step-by-Step: Monitoring General Signals

1. Press **MENU** → **Monitor**.
2. Select **INPUT** or **OUTPUT**.
3. Input starting signal number (head).
4. Observe ON/OFF status (black = ON, white = OFF).

6.3.3 Information Sheet - MPS Multifunction Gripper I/O Programming

Key Notes

- Controlled by HOpen and HClose.
- Requires hand number (1-4).
- Always follow with Dly for timing.
- Hand1 Open = Output 900; Hand1 Close = Output 901.
- Feedback via M_HndCq(HandNo).

Gripper Operation Procedure

1. Use HOpen/HClose.
2. Add delay Dly <time>.
3. Continue motion command.

Example: Color Detection with Gripper Diffuse Sensor

Step	Command	Description
10	HClose 1	Grasp workpiece.
20	Dly 0.5	Wait 0.5s.
30	Mvs , -50	Lift workpiece.
40	Mov PPUT_Approach	Move toward placement.
50	Mvs PPUT	Move to placement position.
60	HOpen 1	Release workpiece.

70	Dly 0.5	Wait 0.5s.
80	Mvs , -50	Retreat upward.

6.3.4 Information Sheet - Programming Example: Detect Product Color and Place Piston

Color Detection Logic

```
10 M_Tool = m_Center
20 Mvs pCylindColCheck
40 If M_In(i_SensColCheck)=1 Then
50   mBlack% = 0
70 Else
80   mBlack% = 1
90   pColorOffset = (+0.00,+0.00,-2.00,+0.00,+0.00,+0.00)
100 EndIf
```

Piston Selection and Placement

```
110 M_Tool = m_Piston
120 HOpen 1
130 If mBlack% Then
140   Mvs (Plt 2,m_SilCnt), -50
160   HClose 1
170   Dly 0.5
190   m_SilCnt = m_SilCnt + 1
200 Else
210   Mvs (Plt 1,m_BlkJnt), -50
230   HClose 1
240   Dly 0.5
260   m_BlkJnt = m_BlkJnt + 1
270 EndIf

280 Cnt 1
290 Mov p_ProductAsm,-50
300 Mvs p_ProductAsm
310 HOpen 1
320 Dly 0.5
340 Cnt 0
```

6.3 Learning Outcome: Develop Program for Controlling Robot Input and Output

6.3.5 Self-Check Questions

No.	Question
1	Identify the four key performance criteria that define Learning Outcome 6.3 (Develop Program for Controlling Robot Input and Output).
2	What is the designation for the dedicated I/O signals used for gripper control in the Mitsubishi robot system, and what is the typical range of inputs and outputs supported by the controller?
3	How does a MELFA-BASIC V program read the status of an external sensor connected to the robot controller? Provide the relevant command or syntax.
4	A programmer wants the robot to pause its execution until a specific Input signal No. 2 turns ON. Write the appropriate MELFA-BASIC V command for this condition.
5	Describe the process and commands required in a MELFA-BASIC V program to synchronize robot motion with the closing action of a Hand (Gripper).
6	What is the purpose of the M_Out command, and how is it used to control an Output Device (e.g., turning an LED ON)?.
7	Explain the necessity of using the Dly command immediately after an HClose command in a MELFA-BASIC program.
8	What Teaching Box function allows a user to manually overwrite the status of a General-purpose Output for testing and debugging purposes?
9	When implementing conditional logic based on a sensor reading, what MELFA-BASIC V structure is typically used to jump to a specific part of the program if the sensor condition is met?
10	The MPS Robot Station uses a Hand I/O signal to open the gripper. What Output No. and corresponding MELFA-BASIC V command typically achieve this action for Hand 1?

6.3.6 Answer Key

No.	Answer / Guidance
1	The four key performance criteria are: Input devices are configured to trigger robot actions; Input data is read and decisions are made based on defined conditions; Output devices are activated based on logic conditions; and Communication protocols for input/output devices are established and connected.
2	The gripper dedicated I/O signals use the range 900-907. The Mitsubishi CR800 controller has capacity for up to 256 Inputs and 256 Outputs.
3	Input signals are read using the robot status variable M_In(bit_no). For example, M_In(2) returns 1 for ON or 0 for OFF.
4	Wait M_In(2)=1.
5	The program must use the HClose command to close the hand (gripper), immediately followed by a Dly command to enforce a time delay (e.g., Dly 1.0) to allow the mechanical action of grasping to complete before the robot continues motion.
6	The M_Out(bit_no) command is used to write outputs to external devices. To turn an LED ON, you would write M_Out(0) = 1 (assuming LED is Output 0).
7	The Dly command is necessary immediately following an HClose command to create a time delay (in seconds). This delay ensures that the physical action of the gripper or end-effector has time to complete before the robot executes the next motion command.
8	The Teaching Box function is Forced output, which is accessed via MENU → Monitor → Signal monitor → General-purpose signal.
9	The If-Then GoTo structure is used, which performs a comparison and executes an unconditional jump (GoTo) to a Label (marked by *) if the condition is met.
10	The command is HOpen 1, which corresponds to Output 900 for Hand 1 Open.

6.4 Learning Outcome: Operate Robotic Arm**6.4.1 Information Sheet - Safety Guidelines**

When operating the robotic arm, strict safety measures must be followed at all times:

- Read and understand the robot's operation manual thoroughly before use.
- Ensure all emergency stop switches are functional and accessible.
- Verify that all guards and safety fences are in place and operational.

- Keep the robot cell free from unnecessary materials or personnel.
- Never enter the operating range while the robot is in automatic mode.
- Always perform maintenance or adjustments in **Teach** mode with low speed.

6.4.2 Information Sheet - Subroutine Programming

Short Notes

- GoSub / Return: Local subroutines; returns after Return.
- CallP: Calls separate program files.
- Up to 16 parameters via FPrm.
- Subprograms may nest.

Example: Using CallP for Cylinder Assembly

Main Program Snippet	Subprogram Snippet (PistonAsmb)
59 CallP "PistonAsmb", mBlack%	27 Def Inte mBlack
61 CallP "CoverAsmb", mBlack%	28 FPrm mBlack%
57 If m_Return<0 Then GoTo *Loop	71 If mBlack% Then ... EndIf

6.4.3 Information Sheet - User Base Program (UBP) Usage

Short Notes

- Defines global variables and positions.
- Register UBP file in PRGUSR parameter.
- Includes Def Pos, Def Jnt, constants, and tools.
- Consists of .MB5 (instructions) and .POS (coordinates).

Access Steps

1. Press **[MENU]** → Program.
2. Select program **UBP**.
3. Open and view definitions.

6.4.4 Information Sheet - Palletizing Programming

Short Notes

- Use Def Plt to define pallets; Plt to calculate coordinates.
- Patterns: Zigzag (1), Same direction (2), Arc (3).
- MPS uses two 4-point pallets for black/silver pistons.

Example: Pallet Definition and Calculation

```
' Pallet Definition
33 Def Plt 1,p_BlkJistonPal1,p_BlkJistonPal4,p_BlkJistonPal1,,4,1,2
34 Def Plt 2,p_SilPistonPal1,p_SilPistonPal4,p_SilPistonPal1,,4,1,2

' Pallet Calculation
72 Mvs (Plt 2,m_SilCnt), -50
P10=(Plt1,M1)
```

6.4.5 Interrupt Programming

An interrupt is a program control function used to monitor signals or other conditions concurrently with the main program execution. If predetermined conditions are met, the currently executing process can be paused, and control is branched to a designated interrupt routine for immediate execution.

When to Use Interrupts

Interrupts are essential when the robot program requires a fast and precise response to asynchronous external events or internal status changes, without necessarily stopping the robot entirely. Specific scenarios include:

- **Safety and Collision Detection** Interrupts can be defined using commands like Def Act to specify an immediate response if a safety condition is violated. For instance, a movement command can be interrupted if collision detection is activated ($M\_ColSts(1)=1$) to execute a defined escape procedure and stop the operation by generating a user-defined error. This is typically used when the collision detection function is enabled in "NOErr" mode (error non-occurrence mode).
- **Interlocking and Process Control** They are effective for monitoring signals, such as workpiece drops during transport. An interrupt can be defined to execute a subroutine if a specific input signal turns ON.
- **Fast System Response** In general programming, hardware interrupts provide a fast system response to events, such as rising or falling edges of digital input signals.
- **Multitasking Synchronization** Interrupts are used for interlock purposes, such as communicating with external devices. For example, the On Com GoSub command defines a subroutine to be called when an interrupt is generated from a communication line after data is input from an external device.

Programming Example of Using an Interrupt in MELFA-BASIC VI

A common example of using an interrupt is handling a stop button, ensuring that if an operator releases a safety switch or presses an E-stop button, the program halts gracefully.

The following example defines and enables an interrupt for a stop button input ($M\_In(i\_StopButton)$), which, when detected, branches to a subroutine labeled *Stopped:

```
35 ' Declaration Of Interrupts
36 '#####'
```

```
37 Def Act 1, M_In(i_StopButton) = 0 GoSub *Stopped 'Define interrupt 1
38 '
54 Act 1 = 1 'enable interrupt 1
... (Main Program Execution)
67 ' Interrupts
68 '#####'
69 '
70 '>>>>> Stop button <<<<<<<'
71 *Stopped
72 m_StartLight% = 1 'blink LED at START button
73 Wait M_In(i_StartButton) = 1 'wait for START button pressed
74 m_StartLight% = 0 'set LED at START button to steady
75 Return 0 'Returns control to the step where the interrupt occurred
```

*Explanation of Commands Used:

- **Def Act 1, M_In(i_StopButton) = 0 GoSub *Stopped:** Defines Interrupt 1. The condition `M_In(i_StopButton) = 0` is monitored (assuming `i_StopButton` is normally closed, indicating that the button is released or pushed). If the condition is met, the program jumps to the subroutine at `*Stopped`.
- **Act 1 = 1:** Enables (allows) the defined interrupt 1.
- **Wait M_In(i_StartButton) = 1:** Pauses the program execution until the Start button is pressed (input signal is 1).
- **Return 0:** Returns control to the original line where the interrupt occurred.

*Interrupt Stop Types: When defining an interrupt using `Def Act`, the stopping behavior can be specified by a '<Type>' argument:

- **Omitted (Default): Stop type 1** The robot stops at the stop position, considering 100% external override. The robot will always stop at the same position, though the time required might be longer if the external override is small.
- **S (Stop type 2)** The robot decelerates and stops in the shortest time and distance possible, independent of the external override.
- **L (Execution complete stop)** Interrupt processing occurs only after the movement instruction being executed is completed.

For example, defining an interrupt to stop in the shortest time and distance possible would include the 'S' parameter: `Def Act 1, M_In(17) = 1 GoSub *L100, S.`

6.4.6 Information Sheet: Programming Robot for Assembling Process with Screwing

This information sheet outlines the procedure and required programming commands for the robot assembly task involving picking a cover (cap) from a magazine, checking its orientation, and screwing it onto the workpiece at the product assembly position, utilizing MELFA-BASIC VI commands. This process is typically handled by a subprogram called `CoverAsmb`.

Overview of the Assembly Process (CoverAsmb)

The overall task involves distributing a cover from a magazine, picking it up, checking its orientation using a sensor, placing it on the cylinder (workpiece) at the assembly position (p_ProductAsm), and screwing it on,. If the orientation check fails, the part may be discarded.

Required Positions (Global Positions to Teach)

These positions must be defined beforehand, usually as global position variables (UBP):

- j_Home: Home position,.
- p_ProductAsm: Product assembly position (where the red cylinder is assembled),.
- p_CoverMagazine: Cover pickup position (blue cover magazine).
- p_CoverOriCheck: Cover orientation check position (blue cover).
- p_Magazine2: Magazine 2 for bad covers (for discarding covers if orientation is incorrect),,,.

The program may also utilize local position variables such as pCoverHeight (a Z offset for cover height) and pCoverWithAngle (adjusted assembly position),.

Key MELFA-BASIC VI Commands Used

The assembly logic relies on movement, hand control, timing, and logical branching,:

- CallP: Program call; used here to call the SearchEdge function to check orientation,.
- Mov: Moves the robot using joint interpolation,.
- Mvs: Moves the robot using linear interpolation (move straight),.
- HOpen/HClose: Orders the gripper to open or close,,.
- Dly: Delays the robot process for a specified time (used for waiting for hand operations),,.
- If...Then...Else...EndIf: Conditional branching,.
- Cnt 1 / Cnt 0: Designates the start and end of continuous movement.
- JOvr/Spd: Commands for controlling movement speed (Joint Override or linear Speed),.
- M_Tool = m_Center (or similar): Sets the active Tool conversion data (TCP),.
- M_Run(Slot No.): Status variable indicating if a task slot is running (1: executing, 0: not executing), often used with Wait or XRun.

The Cap Assembly Programming Sequence Example

The core sequence involves distributing the cap, picking it up, verifying its position, calculating the final placement, and performing the screwing motion:

1. Initial Setup and Magazine Access

- The robot moves to the approach position above the cover magazine.
- The cover is transported out of the magazine using the subprogram DistCover (not detailed here, but necessary prior step).
- The program waits for external output signals related to the cover distribution to stabilize.
- Mov p_CoverMagazine, -50: Moves the robot via joint interpolation to an approach position 50 mm above the cover magazine,.
- Mvs p_CoverMagazine: Moves straight down to the pickup position,.
- HClose 1: Closes Hand 1 (gripping the cover),.
- Dly 0.5: Wait for secure grip.
- Mvs p_CoverMagazine, -50: Moves straight up 50 mm (lifting the cover).

2. Orientation Check

- Mov p_CoverOriCheck, -50: Moves to the approach position above the orientation check position.
- Mvs p_CoverOriCheck: Moves straight down to the check position.
- CallP "SearchEdge", p_CoverOriCheck, 240: Calls the subroutine to search for the edge or recess (hook of the bayonet closure) using a sensor,.
- If the sensor detects no hook, the cover is dropped into p_Magazine2 and the process is repeated.

3. Calculate Placement Position and Move

- Based on the orientation check result (m_EdgeDir%), the target assembly position (pCoverWithAngle) is calculated, adjusted for color offset and required angular offset.
- Mov pCoverWithAngle, -50: Moves to the approach position 50 mm above the target assembly location.
- Mvs pCoverWithAngle: Moves straight down to the placement position.

4. Screwing Operation

- The final joint position (J_Curr) is saved in jCoverScrewedOn.
- The 6th axis (J6) is rotated by a predetermined angle (fScrewAngle) to simulate screwing the cap onto the cylinder.

Speed Control Considerations

The program uses velocity control commands like JOvrd m_OvrdFast (Joint Override Fast) for rapid movements and Spd m_SpdSlow (Speed Slow) for approaching or detailed manipulation, ensuring precision during pick and place and assembly.

6.4.7 Job Sheet - Operate Robotic Arm using Teaching Box (T/B)

Section	Content
Job Title	Operate Mitsubishi RV-2FL Robotic Arm using Teaching Box (T/B)
Personal Protective Equipment (PPE)	Safety gloves, safety glasses, and protective shoes must be worn inside the robot operation area. Ensure the safety fence and emergency stop are active before operation.
Materials Required	Mitsubishi RV-2FL robot and CR800 controller; Teaching Box (T/B); Operator panel; Safety fence and emergency stop unit; Workpiece (cylinder); Power supply and communication cables.
Tools and Equipment	Computer with MELFA Basic software; Pneumatic air source for gripper actuation; Multimeter for signal testing and verification.
Procedure	1. Preparation and Safety Check - Ensure the work area is clear of any obstacles. - Power ON the robot controller and Teaching Box. - Verify that the emergency stop circuit is functional and safety fence is closed. - Confirm controller indicators show system ready. 2. Manual Jogging and Teaching - Turn the mode selector switch on the T/B to Teach. - Press Servo ON to activate robot motion. - Use jog keys to move the robot to the home position (PHOME). - Teach and register three positions: PWAIT, PGET, and PPUT. - Store these positions in the robot's memory. 3. Program Execution - Load the pre-written program PickPlace.mbp using the Teaching Box or PC. - Execute the program in AUTO mode. - Observe the robot's pick-and-place motion and gripper I/O activation. - Monitor for any alarms or abnormal behavior. 4. Shutdown Procedure - Stop the program and press Servo OFF. - Reset alarms if present. - Power OFF the controller and secure the robot arm in its rest position.

Continued on next page

Section	Content
Notes	- Never enter the robot's motion range while it is powered or in auto mode. - Always confirm the emergency stop and safety fence functionality before operation. - Jog the robot in small increments near objects to prevent collisions. - Document taught positions for future verification.
Key Technical Points	- The Teaching Box (T/B) allows manual control, position teaching, and program execution for Mitsubishi robots. - Servo ON enables power to robot drives, allowing motion commands. - Each taught position is stored in the controller memory and used in MELFA Basic programs. - The system's safety interlocks (fence, E-Stop, mode switch) must be verified before any operation. - Pre-written programs such as PickPlace.mbp demonstrate integrated control of robot motion and gripper actuation.

6.4.8 Self-Check Questions

No.	Question
1	Identify the four key performance criteria that define Learning Outcome 6.4 (Operate Robotic Arm).
2	What must be done before entering the robot operating range, especially while the robot is in automatic mode?
3	How can an operator initiate manual movement of the robot arm using the Teaching Box (T/B)? Name the two essential safety steps required.
4	Name four distinct jog modes available on the Teaching Box and describe the corresponding movement reference for each.
5	What procedure must be performed using the Teaching Box to save the current coordinates of the robot end-effector, and what key is used?
6	Describe the robot's immediate reaction if the Emergency Stop (E-Stop) button on the Teaching Box is pressed.
7	What must be done after an E-Stop alarm is triggered before the robot can resume motion?
8	When is it appropriate to use the STEP mode for execution, and what is the primary benefit of this mode during programming?
9	What are the recommended Position Variables (P) that should be taught and stored in the robot controller's memory for a typical pick-and-place operation?
10	Which Teaching Box screen should be monitored to verify the real-time status of the robot's I/O signals during program execution?

6.4.9 Answer Key

No.	Answer / Guidance
1	The four performance criteria are: Safety guidelines are followed as per safety manual; Robot peripherals are installed and connected; Robot Tool Center Point (TCP) is configured correctly as per instruction manual; Robot jogging is performed using teach pendant; Robot axes are aligned using teach pendant; and Pre-programmed positions are taught using teach pendant. (Accepting four of these or noting the full list).
2	Never enter the operating range while the robot is in automatic mode.
3	The operator must switch the robot to Teach Mode and hold the enabling switch at mid-position (or hold the enable button mid-position) before pressing Servo ON and using the jog keys.
4	Joint: Moves each axis independently; XYZ: Moves TCP linearly in world coordinates; Tool: Moves TCP relative to tool orientation; Work: Moves relative to defined work coordinate frame.
5	The operator must jog the robot to the desired point, select the position variable on the edit screen, and press [F2 TEACH] to register the current coordinates.
6	Pressing [EMG. STOP] on the T/B immediately turns servo power OFF.
7	The alarm must be reset using the [RESET] key and servo re-enabled before resuming operation.
8	STEP mode is used to execute the program line by line. The primary benefit is to debug the program and verify logic flow.
9	Recommended positions include PWAIT (Waiting position), PGET (Pick position), PPUT (Place position), and PHOME (Safe home position).
10	The operator should monitor the Monitor → I/O Signal screen (General-purpose signal) to observe the ON/OFF status of the signals in real-time.

Module-7 : Operate Collaborative Robot (COBOT)

Module Description

This unit covers the knowledge, skills and attitudes required to operate collaborative robot (COBOT). It specifically includes the tasks of understanding collaborative robot principles, features and safety protocols, setting up and operating collaborative robots, integrating COBOT with CNC machine, installing end-effectors with COBOT, programming tasks for collaborative robots and troubleshooting and optimizing robot performance.

Minimum Time: 25 Hours

Learning Outcomes

1. Understand Collaborative Robot Principles, Features and Safety Protocols
2. Set up and Operate Collaborative Robots
3. Integrate COBOT with CNC Machine
4. Program Tasks for Collaborative Robots
5. Troubleshoot and Optimize Robot

7.1 Learning Outcome: Understand Collaborative Robot Principles, Features and Safety Protocols

7.1.1 Information Sheet - Introduction to Collaborative Robots (Cobots) and Their Features

Learning Objective

At the end of this session, the trainee will be able to describe the principles, features, and capabilities of collaborative robots (cobots).

What is a Collaborative Robot (Cobot)?

A collaborative robot is a type of industrial robot designed to work safely alongside humans without physical barriers. Unlike traditional robots, cobots are equipped with advanced sensors, lightweight designs, and force-limiting features that make them inherently safe for human-robot collaboration.



Figure 7.1: Cobot and human working together in a shared workspace

Advantages of Collaborative Robots

- Easy to program and reconfigure
- Compact and lightweight for mobile use
- Lower safety barrier requirements
- Quick integration with existing systems
- Cost-effective for small and medium enterprises (SMEs)

7.1.2 Information Sheet - Principles and Applications of Collaborative Robots (Cobots)

Learning Objective

At the end of this session, the trainee will be able to explain the operating principles of collaborative robots and their applications in industry.

Principles of Collaborative Robotics

Cobots operate based on the following principles:

1. **Human-Robot Collaboration (HRC):** Cobots share workspace with humans, assisting in repetitive or precise tasks.
2. **Force and Torque Limiting:** Embedded sensors limit the force applied, ensuring the robot stops if it contacts a person.
3. **Speed and Separation Monitoring:** Sensors detect human presence; the robot slows down or stops automatically.

4. **Power and Force Limiting (PFL):** Regulates the amount of force output to prevent injury.
5. **Hand-Guiding Mode:** Allows the operator to manually move the robot arm to teach positions.

Applications of Collaborative Robots

Cobots are versatile and used across various industrial applications:

- **Assembly:** Tightening screws, inserting parts, small component fitting.
- **Packaging and Palletizing:** Product sorting, box stacking.
- **Quality Inspection:** Vision-based measurement and sorting.
- **Material Handling:** Loading/unloading CNC machines.
- **Pick and Place:** Fast repetitive item transfer.
- **Gluing, Welding, Polishing:** Assisted material joining and surface finishing.

7.1.3 Information Sheet - Safety Standards and Considerations for Collaborative Robots (Cobots)

Learning Objective

At the end of this session, the trainee will be able to identify safety standards, explain safety considerations, and carry out risk assessments related to collaborative robots.

International Safety Standards

The safety features of the collaborative robot, such as the UR5e, encompass a wide range of hardware and software functions designed to ensure safe operation, especially in collaborative applications where humans and robots work in proximity. These safety functions are constructed to achieve ****Performance Level d (PLd)**** using a ****category 3 architecture**** in compliance with standards.

Standard Code	Title	Description
ISO 10218-1 & 2	Robots and Robotic Devices - Safety Requirements	General safety guidelines for industrial robots and integration systems.
ISO/TS 15066	Collaborative Robot Safety Requirements	Defines force, speed, and power limits for human-robot collaboration.
ANSI/RIA R15.06	Safety Standard for Industrial Robots	North American equivalent to ISO 10218.

Table 7.1: International Safety Standards for Robotics

Safety System Architecture and Stop Categories

The robot's safety system continuously monitors for exceeded safety limits, Emergency Stops, or Safeguard Stops. The actions taken in response to these triggers fall into distinct stop categories defined according to IEC 60204-1:

Stop Category	Description
0	Stop the robot by immediate removal of power.
1	Stop the robot in an orderly, controlled manner. Power is removed once the robot is stopped.
2	*Stop the robot with power available to the drives, while maintaining the trajectory. Drive power is maintained after the robot is stopped.

*Universal Robots robots' Category 2 stops are further described as SS1 or as SS2 type stops according to IEC 61800-5-2.

Figure 7.2: Stop Categories for Robot Safety System Responses

- **Stop Category 0:** This is the robot's reaction to a ****Limit Violation**** or a ****Fault Detection****. It involves the immediate removal of power, resulting in an uncontrolled stop where the robot may deviate from the programmed path as joint brakes engage rapidly.
- **Stop Category 1:** This is the reaction to an ****Emergency Stop****. It is a controlled stop where power remains available to the drives to achieve the stop, but power is removed once the robot is stopped and standing still.
- **Stop Category 2:** This is the reaction to a ****Safeguard Stop****. It is a controlled stop where drive power is maintained after the robot has stopped, and the safety-related control system monitors that the robot stays at the stopped position.

Configurable Safety Functions

The Universal Robots e-Series includes several configurable safety functions, particularly relevant for addressing specific risks identified during the integrator's mandatory risk assessment for collaborative applications. These features are configurable through the safety configuration settings.

Limitation Functions:

- **Force and Power Limiting:** These reduce clamping forces and pressures exerted by the robot, particularly in collision scenarios between the robot and an operator. This function limits the maximum force exerted by the tool/end effector (e.g., 25N) and limits the mechanical work performed by the robot (e.g., Power Limit 10W).
- **Momentum Limiting:** This is used to reduce high transient energy and impact forces in collisions by restricting the robot's speed. The maximum momentum is limited (e.g., 3 kg m/s).
- **Speed Limitation (Speed Limit):** This function limits the maximum robot speed, which is restricted at the elbow, tool/end effector flange, and the center of user-defined tool/end effector positions (e.g., 50mm/s).

Stopping Time and Distance:

- **Stopping Time Limit and Stopping Distance Limit:** These functions ease the application risk assessment process by dynamically reducing the robot's motion speed so that it can always be stopped within the configured limits (e.g., Time Limit 50ms and Distance Limit 40mm).

Position and Orientation Limits:

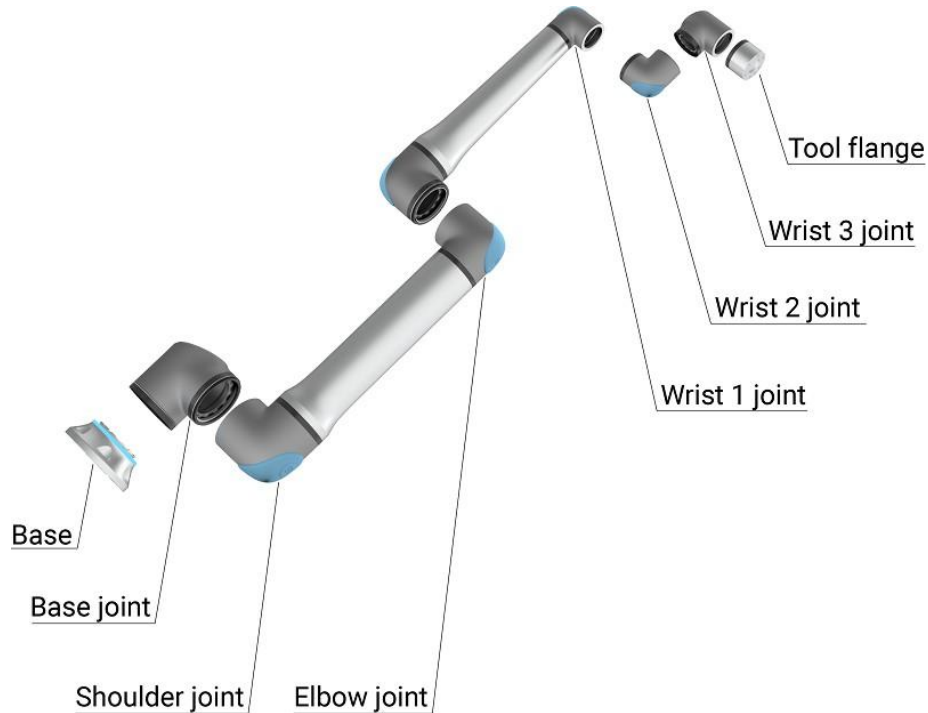


Figure 7.3: Cobot joint exploded view

- **Joint Position Limit:** Sets upper and lower limits for the allowed rotational positions of individual joints.
- **Joint Speed Limit:** Sets an upper limit for the angular speed of individual joints.
- **Safety Planes:** Defines up to eight planes in space to restrict the robot's workspace, limiting either the tool/end effector alone or both the tool/end effector and the elbow. These planes can also be configured to **Trigger Reduced Mode**.
- **Tool Orientation Limiting:** Defines allowable orientation limits for the tool, often used to reduce risks associated with tool features like sharp edges.
- **Safe Home Position:** A user-defined return position for the robot arm which can be defined as a safety feature.

Safety Modes and Interfaces

The safety system features different operating modes, each associated with a set of safety limits.

Safety Modes:

- **Normal Mode:** The safety mode that is active by default.
- **Reduced Mode:** This mode is active when the robot is triggered by a specific safety input or when the Tool Center Point (TCP) is positioned beyond a ****Trigger Reduced Mode plane****. When in Reduced Mode, safety limits for speed and momentum are generally more restrictive than in Normal Mode.
- **Recovery Mode:** This special mode is automatically entered if a safety limit is exceeded (initiating a Stop Category 0) or violated at startup. It permits the robot arm to be manually moved back within limits using Freedrive Mode or the Move tab, although limits for joint position, tool position, and tool orientation are disabled in this mode.

Safety I/O and Control Signals: The robot is equipped with safety I/O and control signals, which are crucial for connecting to external machines and additional protective devices. All safety-related I/O are constructed redundantly (two independent channels) to ensure that a single fault cannot lead to a loss of the safety function (Category 3 architecture).

- **Permanent Inputs** include the ****Robot Emergency Stop**** (Stop Category 1) and the ****Safeguard Stop**** (Stop Category 2).
- **Configurable Inputs** include the System Emergency Stop, Safeguard Reset, Reduced Mode, ****Three-Position Enabling Device**** (required for movement in manual mode when configured), and Operational Mode switch.
- **Configurable Outputs** include the System Emergency Stop signal (to share the E-stop state with other machines), Robot Moving, Robot Not Stopping, Reduced Mode status, Not Reduced Mode status, and Safe Home status.
- **OSSD Signals** can be used for configured safety inputs and outputs, detecting the ability of the Control Box to make safety outputs active/low.

7.1.4 Self-Check Questions

No.	Question
1	Define a Collaborative Robot (Cobot) and explain the fundamental safety distinction between a Cobot and a traditional industrial robot.
2	Identify four key physical features used to describe the capabilities of a Cobot, such as the UR5e.
3	What specific ISO safety standard defines the force, speed, and power limitations required for safe human-robot collaboration?
4	Name the two Stop Categories defined by IEC 60204-1 and describe the primary difference in their execution, particularly concerning power removal.
5	Describe the operational characteristics of the Cobot when it is functioning in Reduced Mode.
6	Why must safety-related I/O signals on a Cobot be constructed redundantly, utilizing two independent channels?
7	What specific safety feature allows the operator to manually move the robot arm to teach positions, and how is it used?
8	Name three common industrial applications where Cobots are frequently utilized due to their versatility and ease of integration.
9	If a Cobot initiates a Stop Category 0 due to a safety limit violation, which special mode does the robot enter, and what restriction is placed on program execution in this mode?
10	The robot's safety configuration is represented by a unique numerical indicator displayed in PolyScope. What is this indicator called, and what is its purpose?

7.1 Learning Outcome: Understand Collaborative Robot Principles, Features and Safety Protocols

7.1.5 Answer Key

No.	Answer / Guidance
1	A Cobot is a type of industrial robot designed to work safely alongside humans without physical barriers. Unlike traditional robots, Cobots are inherently safe due to features like advanced sensors and force-limiting capabilities.
2	Four key physical features are Payload, Range, Joint Speed, and Accuracy and Repeatability.
3	ISO/TS 15066 defines the force, speed, and power limits for human-robot collaboration.
4	Stop Category 0 (STO) involves the immediate removal of power (uncontrolled stop). Stop Category 1 (SS1) involves a controlled stop where power remains available to the drives to decelerate the robot, but power is removed once the robot is stopped.
5	When in Reduced Mode, safety limits for speed and momentum are generally more restrictive than in Normal Mode. This mode is triggered by a specific safety input or when the TCP is beyond a designated Trigger Reduced Mode plane.
6	Safety-related I/O must be constructed redundantly (dual-channel) to ensure that a single fault cannot lead to a loss of the safety function (Category 3 architecture).
7	The Hand-Guiding Mode (or Freedrive Mode) allows the operator to manually move the robot arm to teach positions. It is activated by pressing and holding the Freedrive Button located at the back/top of the Teach Pendant.
8	Three applications are Assembly (tightening screws, inserting parts), Packaging and Palletizing (sorting, stacking), and Material Handling (loading CNC machines). (Also accepting Quality Inspection, Pick and Place, Gluing, Welding, Polishing).
9	The robot enters Recovery Mode. In this mode, programs cannot be run, but the robot arm can be manually moved back within limits.
10	The indicator is the Safety Checksum. It reflects the currently applied safety configuration and is used to verify the settings.

7.2 Learning Outcome: Set up and Operate Collaborative Robots

7.2.1 Information Sheet - Installation and Mounting of Collaborative Robots (Cobots)

The installation and mounting process for the Universal Robots e-Series collaborative robot (cobot), such as the UR5e, involves strict adherence to mechanical requirements, electrical interface specifications, and comprehensive safety measures dictated by the integrator's risk assessment. The process is divided into two primary sections in the manual: Hardware Installation and PolyScope Manual usage.

Integrator Responsibility and Pre-Installation Safety

The UR e-Series robot is classified as partly completed machinery, requiring the integrator to be responsible for the overall system safety. Before the robot arm is powered on for the first time, the integrator must fulfill several critical safety obligations:

1. ****Risk Assessment:**** A risk assessment for the complete robot system must be performed, taking into account all work tasks throughout the lifetime of the application, including teaching, maintenance, and normal operation. The integrator is responsible for eliminating significant hazards in the complete application.
2. ****Safety Configuration:**** Appropriate safety settings must be configured in the software as determined by the risk assessment. Unauthorized access to these safety settings must be prevented, typically using password protection.
3. ****Pre-Use Assessment:**** Before initial use or after any modifications, the integrator must verify the correct functioning of all installed safety features, including Emergency Stop buttons and Safeguard inputs.

Mechanical Mounting of Robot Arm and Components

The physical installation involves securing the robot arm and placing the associated control equipment:

- **Robot Arm Mounting:** The Robot Arm is mounted using four M8 bolts (strength class 8.8) inserted into four 8.5mm mounting holes at the base. The bolts must be tightened with 20Nm torque. The mounting surface must be sturdy, vibration-less, and capable of withstanding at least ten times the full torque of the base joint and at least five times the weight of the Robot Arm. The Robot Arm, Control Box, and Teach Pendant must only be operated in environments suited to their respective IP ratings (IP54 for the robot arm).
- **Tool Mounting:** The tool flange has four M6 threaded holes for attachment. M6 bolts must be tightened with 8Nm (strength class 8.8). Bolts must not extend beyond 8mm to prevent damage and short-circuiting.
- **Control Box and Teach Pendant:** The Control Box (IP44) can be hung on a wall or placed on the ground, requiring 50mm clearance on each side for airflow. The Teach Pendant (IP54) can be hung on the Control Box or a wall, ensuring the cable does not cause a tripping hazard.

Electrical Interface and Connections

The electrical setup involves connections for power and control signals:

- **Mains Power:** The Control Box must be connected to the mains using a standard IEC plug/cord. The input power must be protected by a Residual Current Device (RCD) and a correct fuse. A protective earth/ground connection is mandatory.
- **Robot Connection:** The cable connecting the robot arm to the Control Box is plugged into the bottom connector and must be locked by twisting it twice before the Robot Arm is powered on.
- **Safety I/O:** The robot provides redundant, dual-channel safety I/O which must be kept separate so a single fault does not lead to loss of the safety function. Safety signals must strictly use the safety-related I/O.
- **Controller I/O:** The Control Box contains various I/O terminals (digital, analog, configurable) for interfacing with peripherals, PLCs, and emergency stop buttons.

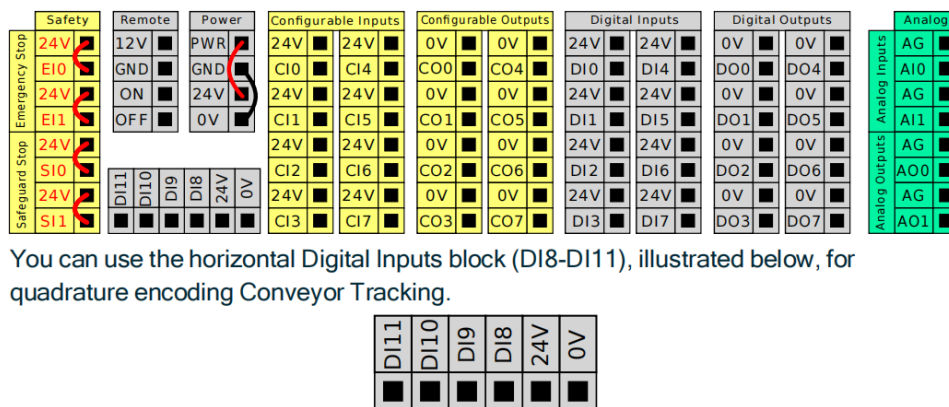


Figure 7.4: Universal Robots e-Series Controller I/O Terminals

Initialization and Software Configuration

Once physical installation is complete, the system startup proceeds with software configuration and initialization:

- **Power Up:** The Control Box is turned on via the Teach Pendant power button. After the system starts, the user must verify the correct installation settings, including Robot mounting angle, mass in TCP, and TCP offset.
- **Initialization:** If the Robot Arm is in the Emergency Stopped state, unlocking the emergency stop button changes the state to Power off. Tapping the ON button on the Initialize Robot screen changes the state to Idle. Tapping Start releases all joint brakes, making the Robot Arm ready for normal operation.
- **Safety Checksum:** After making and applying any changes to the safety configuration, the system generates a Safety Checksum, which is displayed in PolyScope. This checksum reflects the currently applied safety configuration.

- **Mounting Angle Configuration:** If the robot is wall-mounted, ceiling-mounted, or mounted at an angle, this configuration must be adjusted on the Mounting screen under the Installation tab to accurately inform the controller about the direction of gravity. Failure to set the mounting correctly can lead to Protective Stops or unpredictable behavior in Freedrive Mode.

7.2.2 Information Sheet - Introduction to the Universal Robots Cobot Programming Interface

The collaborative robot (cobot) system, such as the Universal Robots e-Series, consists of two main parts: the **Robot Arm** and the **Controller with Teach Pendant**. The Teach Pendant hosts the control interface, while the Control Box houses the main electronics and I/O connections.

The primary user interface for controlling the robot system is the **PolyScope Graphical User Interface (GUI)**, which is displayed on the Teach Pendant's touch screen. PolyScope is used to operate the robot arm and control box, and to create, load, and execute robot programs.

PolyScope Interface Overview

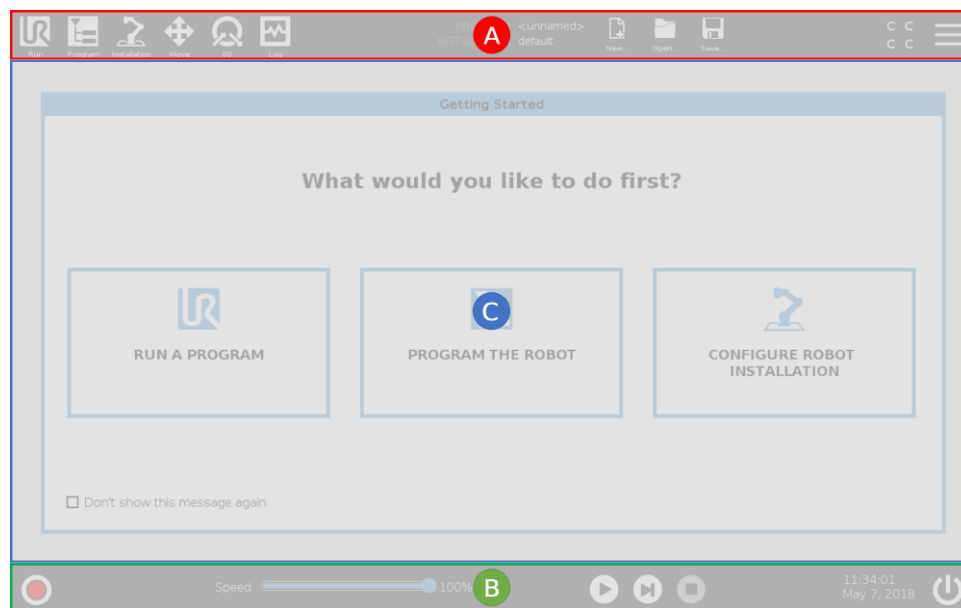


Figure 7.5: PolyScope GUI layout (Header, Footer, Screen).

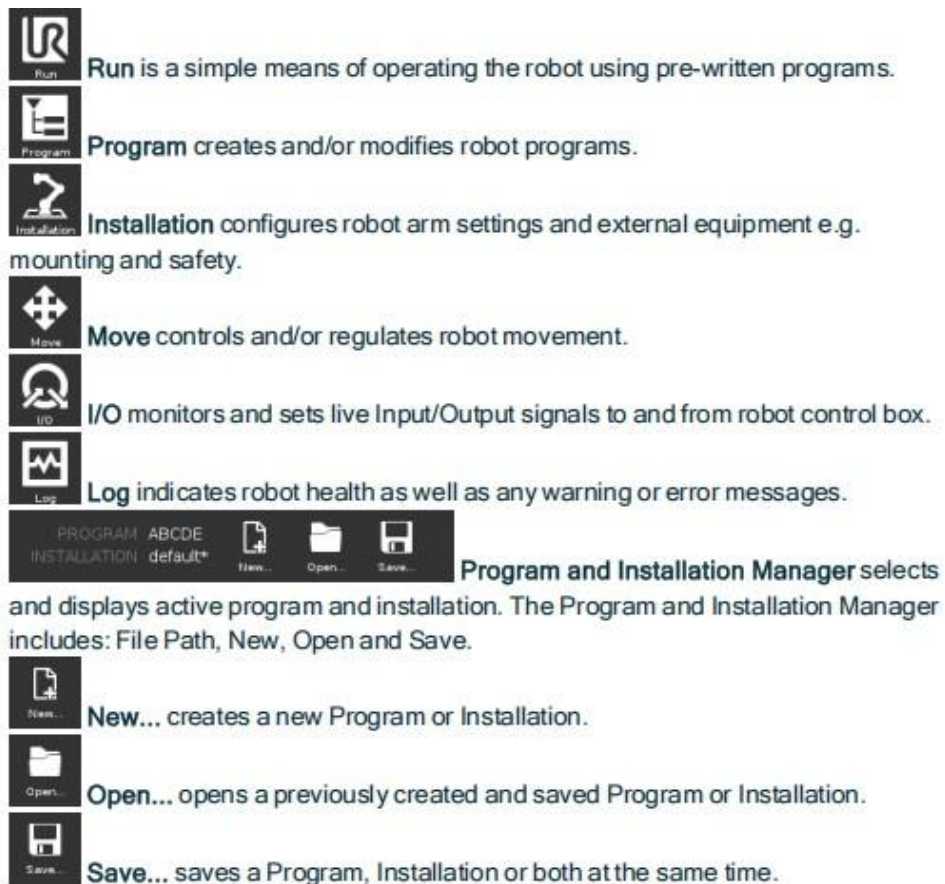
The PolyScope GUI is divided into three main zones:

- **A: Header** - Contains tabs/icons that provide access to interactive screens and system information.
- **B: Footer** - Contains buttons that control loaded program(s).
- **C: Screen** - Displays fields and options used to manage and monitor robot actions.

Header Icons/Tabs

The Header provides access to various functionalities and displays the robot's current state and configuration:

Header Icons / Functions



Operational modes

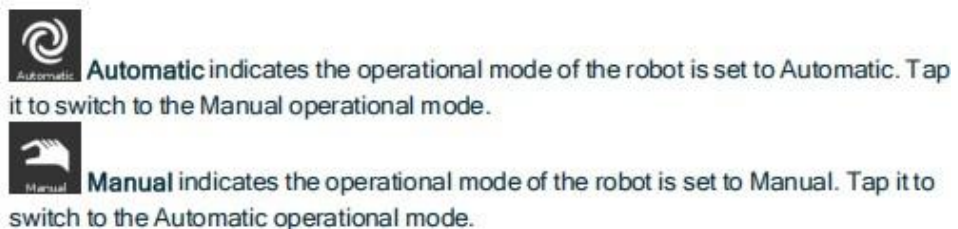
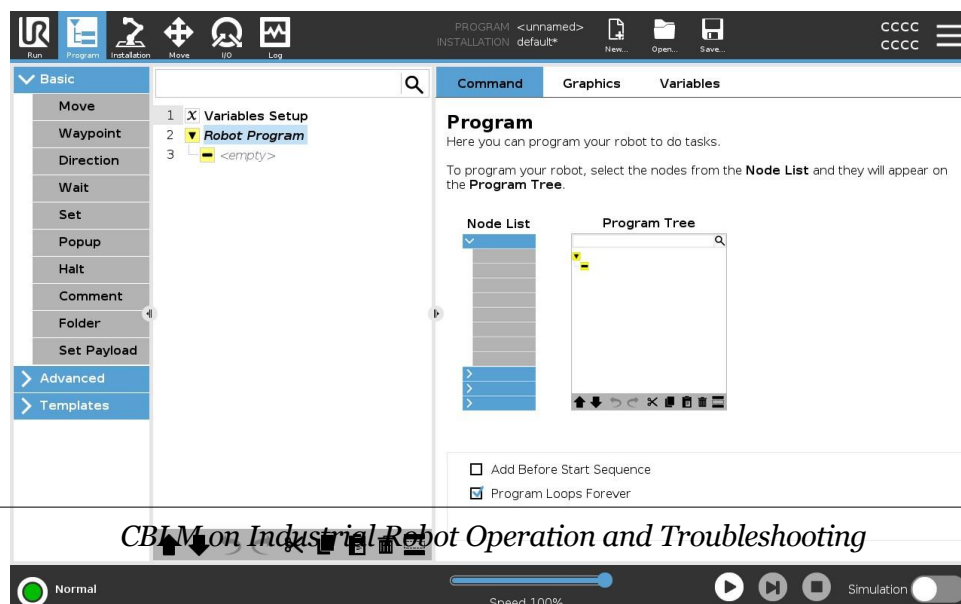


Figure 7.6: PolyScope Header Icons and Functions

Icon/Tab	Function/Description
Run	A simple means of operating the robot using pre-written programs. This tab also displays program status and variables.
Initialize	Manages the robot's state. The icon color indicates the robot arm's running state: Green (normal operation), Yellow (powered on, idle), and Red (stopped state).
Program	Used to create and/or modify robot programs.
Installation	Configures robot arm settings and external equipment, such as mounting, Tool Center Point (TCP), and safety configurations.
Move	Controls and/or regulates robot movement, allowing the user to "jog" the robot arm directly.
I/O	Monitors and sets live Input/Output signals to and from the robot Control Box.
Log	Indicates robot health and displays any warning or error messages, including information about the Control Box and robot joint load.
Program and Installation Manager	Includes File Path (displays active Program and Installation), New..., Open..., and Save... functionalities for managing files.
Operational Modes	Icons display if the robot is set to Automatic or Manual operational mode (if a password is set). They allow switching between modes.
Safety Checksum	Displays the active safety configuration, represented by four or eight digits, to verify the settings.
Hamburger Menu	Provides access to PolyScope Help, About (version, legal, general data), and Settings (preferences, passwords, system management).

Table 7.4: PolyScope Header Icons and Functions



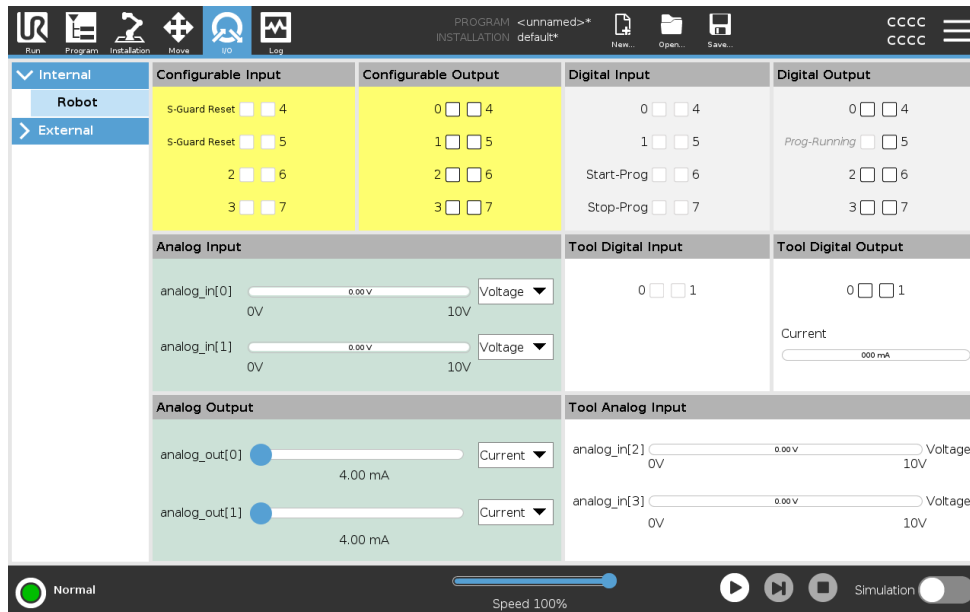


Figure 7.8: Polyscope I/O Tab Interface

Footer Buttons and Functions

The Footer primarily contains buttons for program execution control.

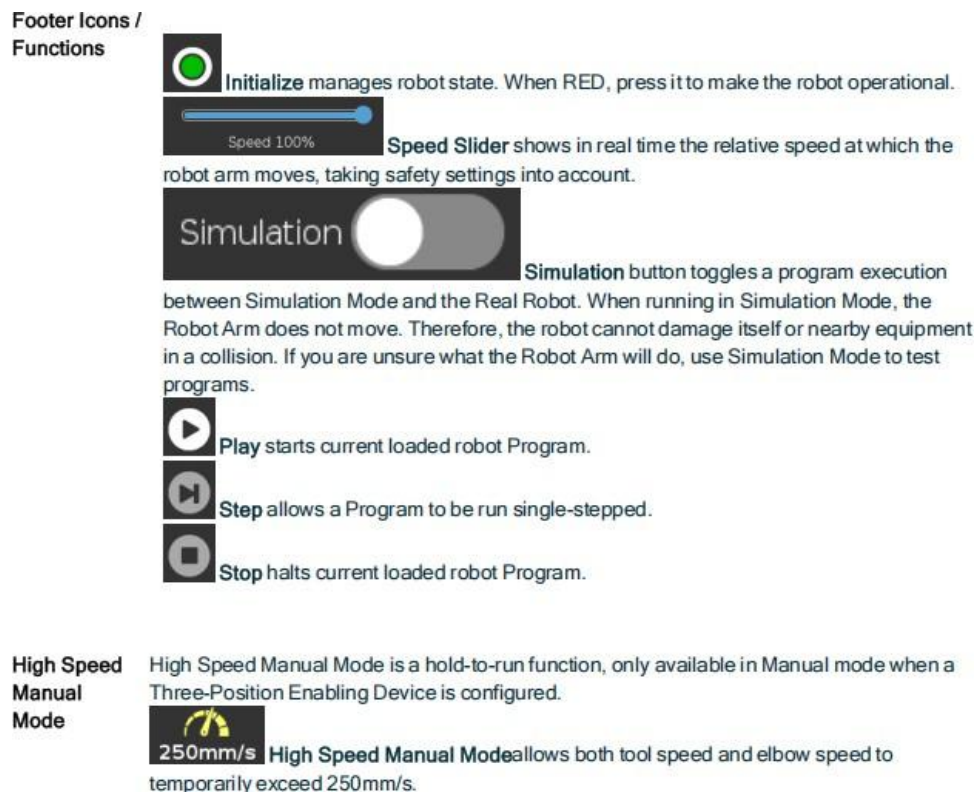


Figure 7.9: PolyScope Footer Buttons and Functions

Button/Function	Description
Speed Slider	Shows the relative speed at which the robot arm moves in real time, taking safety settings into account.
Simulation button	Toggles program execution between Simulation Mode (robot arm does not move) and the Real Robot .
Play/Pause/Resume	Play starts the loaded program. Pause pauses a program. The Play button changes to Pause when running, and Pause changes to Resume when paused.
Step	Allows a program to be run one node at a time (single-stepped). Only available in Manual mode when the program is paused.
Stop	Halts the current loaded robot program. A stopped program cannot be resumed .
Shutdown	Allows the robot system to be powered off or restarted.

Table 7.5: PolyScope Footer Buttons and Functions

Button	Function
Play 	To play a program 1. Under Control, tap Play to start running a program from the beginning.
Resume 	To resume a paused program 1. Tap Resume to continue running the paused program.
Stop 	To stop a program 1. Tap Stop to stop the running program You cannot resume a stopped program. You can tap Play to restart the program.
Pause 	To pause a program 1. Tap Pause to pause a program at a specific point. You can resume a paused program.

Figure 7.10: PolyScope Play, Pause, Resume Buttons

Teach Pendant Specific Features

The Teach Pendant (TP) is the handheld panel containing the PolyScope GUI and physical controls.

1. Power button
2. Emergency Stop button
3. USB port (comes with a dust cover)
4. 3PE buttons

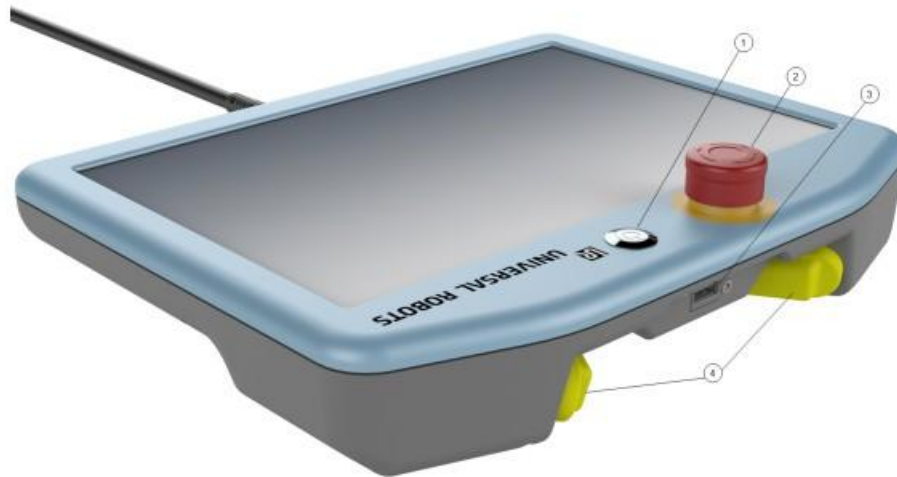


Figure 7.11: Universal Robots e-Series Teach Pendant

Feature	Function/Description
Freedrive Button	Located at the back/top of the Teach Pendant, pressing and holding this button allows the robot arm to be manually pulled into desired positions/poses.
Emergency Stop Button (E-Stop)	A physical push-button on the Teach Pendant used to immediately stop all robot motion. It initiates a Stop Category 1 .
Power Button	Used on the Teach Pendant to turn on the Control Box.
3-Position Enabling Device (3PE)	If configured, this device must be pressed and held in the center-on position for the robot to move in Manual Mode.
Stylus Pen with Laser	Included with the Control Box and Teach Pendant.

Table 7.6: Teach Pendant Physical Features

Control Box Features

Feature	Function/Description
Controller I/O	Includes dedicated safety signals (yellow/red), configurable I/O (yellow/black), general purpose digital I/O (gray/black), and analog I/O (green/black). Contains DI0–DI7, DO0–DO7, CI0–CI7, CO0–CO7, AI0–AI1, AO0–AO1.
Safety I/O	Paired (redundant) terminals for Robot Emergency Stop and Safeguard Stop. Configurable safety I/O can support System Emergency Stop or Reduced Mode.
Mains Connection	IEC C20 plug at the bottom for mains power, protected by RCD and fuse.
Robot Connection Port	Connector at the bottom to plug and lock the robot arm cable.
Ethernet Interface	Used for MODBUS, EtherNet/IP, PROFINET, remote access, and control.
Remote ON/OFF Control	12V, ON, OFF, GND terminals allow remote ON/OFF without Teach Pendant (often via PLC).

Table 7.7: Control Box Features and Connections

7.2.3 Information Sheet - Initializing and Configuring the Universal Robots Cobot System

Objective

To perform COBOT calibration and basic configuration.

Power ON and Initialize the System

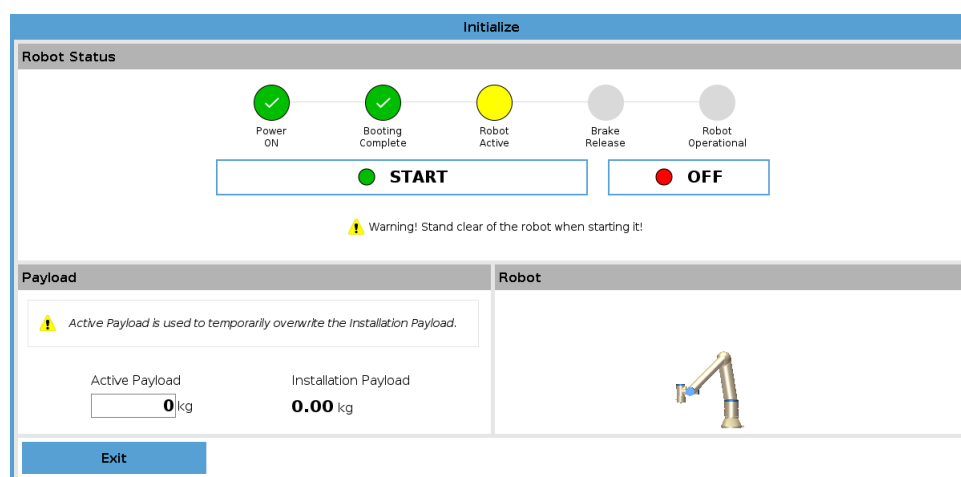


Figure 7.12: PolyScope Initialize screen.

1. **Press Emergency Stop (E-Stop):** On the Teach Pendant, press the emergency stop button.

2. **Power On Control Box:** Press the power button on the Teach Pendant to turn on the Control Box. Wait for the system text and buttons to appear.
3. **Access Initialization Screen:** When a "Cannot Proceed" or similar dialog appears on the touch screen, tap the button to access the initialization screen.
 1. On your Teach Pendant, press the power button to turn on the control box.
 2. Wait as text from the underlying operating system, followed by buttons, appear on the screen.
 3. A Getting Started screen can appear, prompting you to begin programming the robot.

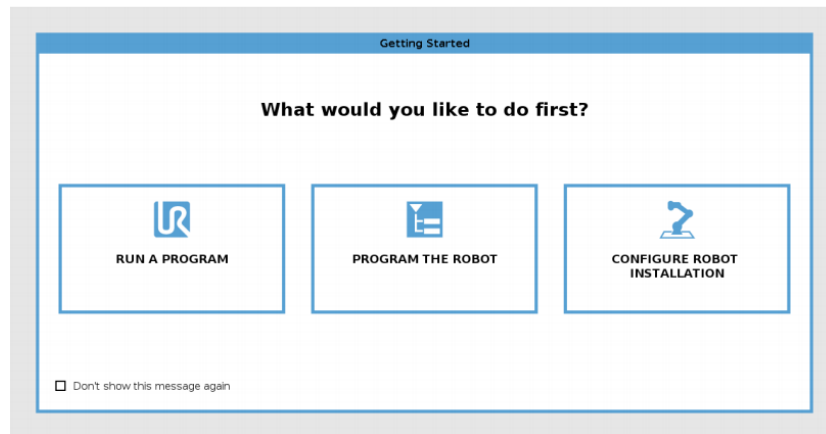


Figure 7.13: Powering ON the universal cobot

4. **Confirm Safety Configuration:** If prompted by the "Confirmation of applied Safety Configuration" dialog, press the Confirm Safety Configuration button to apply the initial set of safety parameters.
5. **Unlock E-Stop:** Unlock the physical emergency stop button on the Teach Pendant. This changes the robot state from "Emergency Stopped" to **Power off**.
6. **Step Outside Workspace:** Step outside the reach (workspace) of the robot.
7. **Power On Robot Arm:** On the Initialize Robot screen, touch the ON button. The robot state will change to **Idle** (Yellow LED indicator).
8. **Verify Configuration:** In the Payload field (under Active Payload), verify that the payload mass is correct. Also, verify the mounting position in the Robot field. If the detected mounting (based on sensor data) does not match the selected mounting (with a tolerance of 30°), the next button will be disabled.
9. **Start Robot:** On the Initialize Robot screen, touch the Start button. The robot will release all joint brakes, accompanied by clicking sounds and slight movements, indicating it is ready for normal operation (**Green LED indicator**).

Execute Joint Homing to Default Positions

Homing often refers to moving the robot arm to a known, predefined position (Home Position).

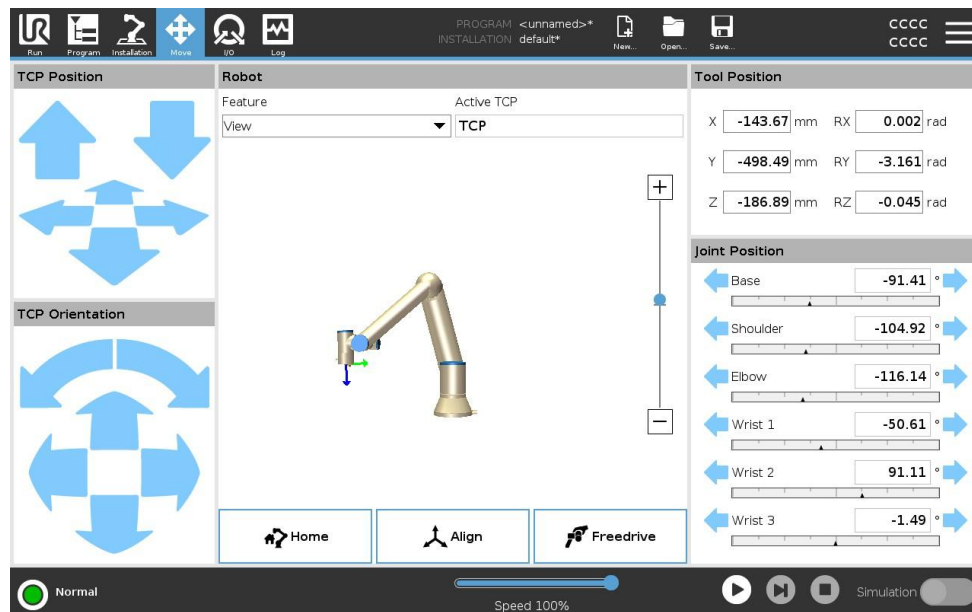


Figure 7.14: Polyscope move tab interface.

1. Define Home Position (If not already set):

- Navigate to the Installation Tab.
- Under General, select Home.
- Tap Set Position.
- Teach the robot arm to the desired return position using either Freemove or the Transition buttons.
- *Note:* If you tap Zero Position on this screen, the Robot Arm is returned to an upright position.

2. Execute Homing:

- Navigate to the Move Tab.
- Press the Home tab (or Home button). This accesses the Move Robot into Position screen.
- Hold down the Auto tab to automatically move the robot arm to its predefined start position.

Set Tool Center Point (TCP)

The TCP is the point on the robot tool where the robot interacts with the world. It includes a translation and a rotation relative to the center of the tool output flange.

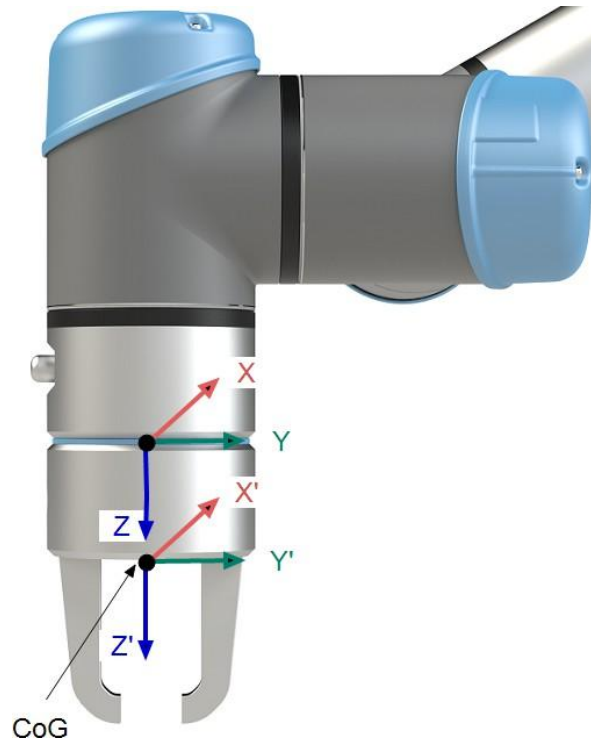


Figure 7.15: Tool center point co-ordinates and center of gravity.

1. **Access TCP Configuration:** In the PolyScope Header, tap the Installation Tab, and then select General > TCP Configuration.
2. **Add/Select TCP:** Tap the New button (plus icon) to define a new TCP, which automatically receives a unique name.
3. **Teach TCP Position (Offset):**
 - Tap Measure (or Position) to automatically calculate the position coordinates.
 - Choose a fixed point in the workspace.
 - Use the position arrows or manual manipulation (if allowed) to move the TCP from at least three different angles and save the corresponding positions.
 - Tap the Set button to apply the verified coordinates to the selected TCP.
4. **Set TCP Orientation (Rotation):**
 - Tap Measure (or Orientation).
 - Select a Feature from the drop-down list.
 - Tap Set point and use the Move tool arrows to position the tool so its orientation coincides with the selected Feature's coordinate system.
 - Tap Set to verify and apply the calculated orientation.
5. **Set Payload and Center of Gravity (CoG):** Specify the **Payload** (weight) of the tool and the **Center of Gravity (CoG)** coordinates (CX, CY, CZ) in the lower part of the screen. Incorrect settings here can result in unwanted robot movement during Freedrive.

Set Work Coordinate System (WCS) / Features

Features represent objects (such as tables or workpieces) and define reference frames relative to the robot base.

1. **Access Features Screen:** In the Installation Tab, select Features.
2. **Add Feature:** Select the desired feature type: Point, Line, or Plane. For high precision WCS definition (e.g., a table), Plane Feature is recommended.
3. **Teach Feature Position:** Use the on-screen guide to teach the feature pose using the current TCP pose by moving the robot (via jog or Freedrive) to the required positions/orientations.
 - **For a Plane Feature:** Define the Origo, then the direction of the positive X-axis, and finally the positive Y-axis.
4. **Make Joggable (Optional):** Ensure the Joggable option is enabled if you plan to use this Feature as the reference frame in the Move Tab for manual jogging.

Create Program and Teach Points via Free Drive

The Freedrive function allows for manually moving the robot arm to quickly set fixed waypoints.

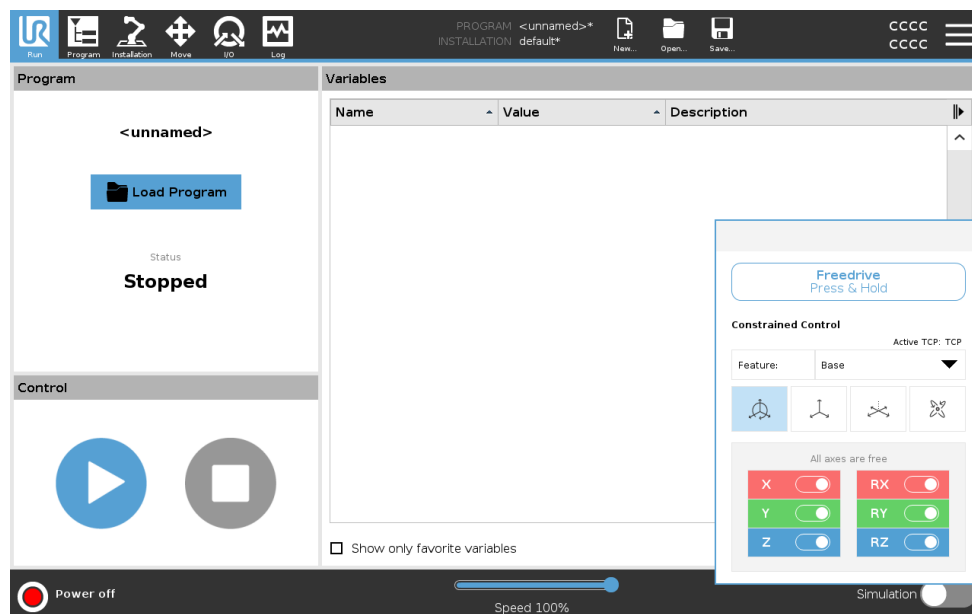


Figure 7.16: Freedrive interface in polyscope

1. **Enter Program Mode:** Navigate to the Program Tab.
2. **Add Move Node:** Under Basic commands, tap Move to add a movement node, which automatically includes a Waypoint node.
3. **Activate Freedrive:** Press and hold down the **Freedrive Button** located at the back of the Teach Pendant.

 All axes are free	Movement is allowed through all axes.
 Plane	Movement is only allowed through the X-axis and Y-axis.
 Translation	Movement is allowed through all axes, without rotation.
 Rotation	Movement is allowed through all axes, in a spherical motion, around the TCP.

Figure 7.17: Freedrive icons explanation

4. **Move Robot Arm:** Manually pull the robot arm to the desired position/pose.
5. **Set Waypoint:** Once the arm is in position, tap the OK button on the screen to save the position as the waypoint (e.g., Waypoint_1).

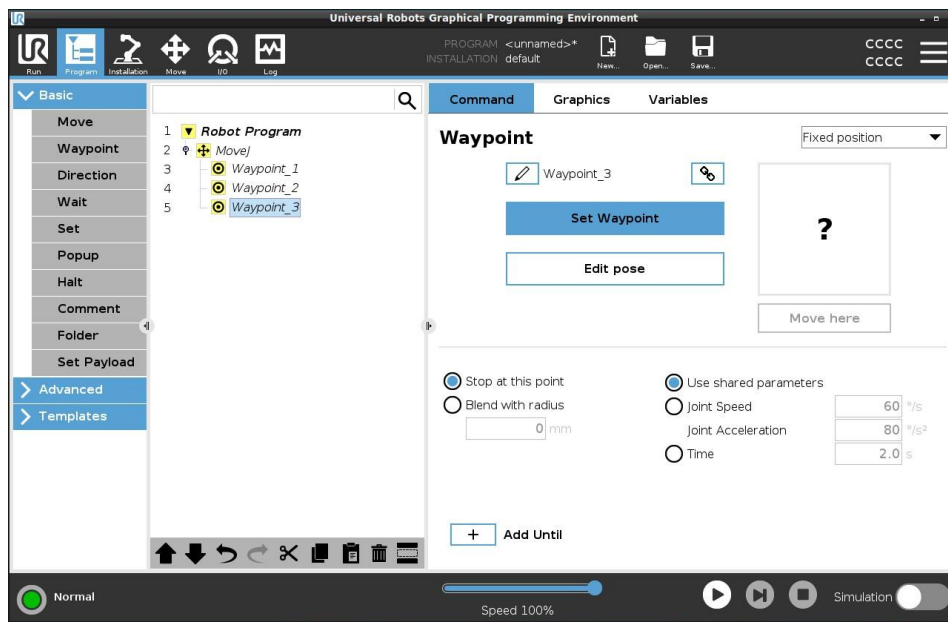


Figure 7.18: Set waypoint in UR programming

6. **Release Freedrive:** Release the Freedrive button.
7. **Repeat:** Repeat steps 2 through 6 to define subsequent waypoints (e.g., Waypoint_2).

Run Program and Verify Repeatability

To verify that the robot reliably returns to the taught points, execute the created program.

1. **Define Movement:** Ensure the Waypoints are placed under a Move command (e.g., MoveJ or MoveL), defining the desired path, speed, and acceleration.
2. **Stand Clear:** Step outside the reach (workspace) of the robot.

3. **Execute Program:** In the PolyScope Footer, press the Play button for the Robot Arm to move between the defined Waypoints.
4. **Observe Motion:** The robot moves through the planned trajectory. Pose Repeatability for the UR5e is specified at **0.03** mm. Observing consistent movement to the desired points validates the repeatability of the taught positions.

Save and Back-up Configuration Data

Save Program and Installation Files

The program and installation files are saved using the File Manager. Saving a program automatically saves the current active installation.

1. **Access Save Options:** In the Header File Path, tap Save....
2. **Save:** Choose one of the following options:
 - **Save All:** Saves the current Program and the current active Installation immediately.
 - **Save Program As...:** Saves the Program with a new name/location (and saves the Installation with its existing name).
 - **Save Installation As...:** Saves the Installation with a new name/location (and saves the Program with its existing name).
3. **Name File:** If using Save As..., assign a file name and tap Save.

Create a Full System Backup

For disaster recovery, you can save a full copy of the system, including configuration, programs, and error logs, to a USB drive.

1. **Insert USB Drive:** Insert an external media into a USB port (it is recommended to use one of the ports inside the Control Box).
2. **Access Backup Screen:** In the Header, tap the Hamburger menu and select Settings. Under the System menu, tap Backup Restore.
3. **Perform Backup:** Select the Location on the USB drive to store the backup and press Backup.
4. **Reboot:** Tap OK for a full system reboot to complete the process.

7.2.4 Information Sheet - Installation and Configuration of End-Effectors

Objective

To install and configure end-effectors for correct attachment and operation.

End-Effector Types

- **Gripper:** Mechanical or pneumatic for pick-and-place.
- **Vacuum Cup:** For flat surfaces like panels or cartons.
- **Welding Torch:** For spot or arc welding tasks.
- **Screwdriver Spindle:** For automated fastening.

Procedure

1. Mount the tool on the robot flange.
2. Connect pneumatic or electrical interface.
3. Register tool data in software (**TCP & weight**).
4. Test tool activation and verify signal feedback.

Safety

- Disconnect power before mounting tools.
- Ensure cables do not restrict joint motion.
- Verify tool response in manual mode before automation.

The installation and configuration of a proprietary end-effector, such as an OnRobot gripper or vacuum cup, typically requires three main phases: installing the necessary controlling software (**URCap**), connecting the end-effector using the tool interface, and configuring the physical properties (**Tool Center Point** and **Payload**) within the PolyScope interface.

Phase 1: Installing the URCap Plugin

URCaps are proprietary accessories and peripherals that extend the functionality of the Universal Robots (UR) software.

1. **Access URCap Management:** On the Teach Pendant, tap the Hamburger Menu and select Settings. In the menu on the left, under System, select URCaps.
2. **Install the File:** Tap the + button, select the required .urcap file (which should be stored on a USB drive), and press Open.
3. **Restart the System:** To proceed with the installation, press Restart. The URCap will be installed and ready for use once the system reboots.
4. **Note:** Once the URCap is installed, if it controls the end-effector (like a gripper), the URCap requires control of the Tool I/O Interface.

Phase 2: Physical and Electrical Connection

The end-effector (End of Arm Tooling / EoAT) is the device at the end of the robotic arm, which could be a vacuum cup, gripper, or magnetic tool.

1. **Mechanical Mounting:** Attach the end-effector to the robot tool flange (Wrist 3). The tool flange has **four M6 thread holes** for attaching a tool, and the bolts must be tightened with **8Nm** torque (strength class 8.8).
2. **Tool I/O Connection:** Connect the end-effector cable to the **eight-pinned connector** adjacent to the tool flange on Wrist #3. This connector provides power and control signals.
3. **Set Tool Power Supply:** The internal power supply for the tool connector can be set to **0V, 12V, or 24V** in the I/O tab of the GUI.
 - The supply voltage in 24V mode is typically 24V (min 23.5V, max 24.8V).
 - The supply voltage in 12V mode is typically 12V (min 11.5V, max 12.5V).
 - The average supply current must not exceed **600 mA** in both modes.
4. **Assign I/O Interface Control (If using URCap):** If the end-effector is controlled by the newly installed URCap, you must assign control:
 - Navigate to the Installation Tab and select General > Tool I/O.
 - Under I/O Interface Control, select the name of the URCap to allow it to control the Tool Analog Inputs and/or Digital Output Mode settings.
5. **Enable Tool Communication Interface (TCI):** If the tool communicates via RS485 or similar protocol, enable the Communication Interface:
 - In the Installation Tab, under General, select Tool I/O.
 - Select Communication Interface to edit TCI settings. Enabling TCI allows the robot to communicate with the attached tool via the robot tool analog input, removing the need for external cabling.
 - You can then select the desired baud rate, parity, and stop bits from the drop-down menus.

Phase 3: Configuring the End-Effector in PolyScope

Configuration is performed in the Installation Tab and is crucial for kinematic accuracy and safety.

A. Defining TCP (Tool Center Point) and Payload

The **TCP** is the point on the robot tool where the robot interacts with the world, defined by a translational and rotational offset from the tool output flange.

1. **Access TCP Configuration:** Navigate to the Installation Tab > General > TCP Configuration.

2. **Define a New TCP:** Tap the New button to define a new TCP, which receives a unique name.
3. **Teach TCP Position (Offset):**
 - Tap Measure (or Position).
 - Choose a fixed point in the robot's workspace.
 - Move the TCP from at least three different angles (using position arrows or Freedrive) and save the corresponding positions.
 - Tap the Set button to apply the calculated coordinates to the TCP.
4. **Configure Tool Payload (Mass):**
 - Specify the **Payload (weight)** of the tool/end-effector in the fields on the screen.
 - The maximum allowed payload depends on the center of gravity offset.
5. **Set Center of Gravity (CoG):** Specify the **CoG** using the fields CX, CY, and CZ. If the CoG is not specified, the TCP is assumed to be the CoG.
6. **Verify Configuration:** Always verify the payload mass, CoG, and robot mounting angle are correct, as incorrect settings may result in frequent robot stops.

B. Configuring Safety Tool Position Restriction

If safety configuration requires setting constraints based on the tool geometry, the end-effector must be defined as a tool sphere within the safety settings.

1. **Access Safety Settings:** Navigate to the Installation Tab > Safety. Enter the safety password and tap Unlock.
2. **Configure Tool Position Restriction:** Select Tool Position from the side menu.
3. **Add Tool:** Tap Add Tool.
4. **Define Tool Geometry:**
 - Specify the Position (x, y, z) of the tool sphere relative to the robot tool flange.
 - Set the Radius of the tool sphere (range 0 – 300 mm). The radius defines the sphere that the safety system uses to ensure the tool does not pass a defined safety plane.
 - *Optional:* You can use an existing defined TCP to copy the position values.

7.2.5 Self-Check Questions

No.	Question
1	After initial power-on of the Universal Robots Cobot system, what critical safety action must be performed on the Teach Pendant before proceeding to the initialization screen?
2	What is the function of the General → Home setting in the Installation Tab, and how is the Home position typically executed after configuration?
3	Define the Tool Center Point (TCP) in the context of UR programming, and identify the required settings that must be specified in the configuration screen.
4	Describe the procedure for setting the TCP position (Offset) using the PolyScope interface (e.g., using Measure), including the minimum number of angles required.
5	When setting up a Plane Feature for a Work Coordinate System (WCS), what three geometric references must be defined sequentially?
6	What physical action on the Teach Pendant is used to quickly save a Waypoint position, and what must the operator do simultaneously?
7	What is the purpose of the Payload and Center of Gravity (CoG) settings in the Installation Tab, and what consequence can occur if they are set incorrectly, particularly during Freedrive?
8	Why is it critical to Set Mounting Angle correctly in the Installation Tab if the robot is wall-mounted or ceiling-mounted?
9	When configuring an End-Effector, what two physical properties must be defined in the software to ensure kinematic accuracy and safety?
10	Which PolyScope menu provides access to Backup Restore functions for creating a full system backup onto a USB drive?

7.2.6 Answer Key

No.	Answer / Guidance
1	The Emergency Stop (E-Stop) button on the Teach Pendant must be pressed.
2	The Home setting defines a known, predefined position for the robot arm. The Home position is executed by navigating to the Move Tab and holding down the Auto tab (or pressing the Home button).
3	The TCP is the point on the robot tool where the robot interacts with the world, defined by a translation and rotation relative to the center of the tool output flange. The configuration must specify the Payload (weight) and Center of Gravity (CoG).
4	To set the TCP Offset, the user accesses TCP Configuration → Measure (or Position) and moves the TCP from at least three different angles to a fixed point in the workspace.
5	For a Plane Feature (WCS), the user must define the Origo (origin), then the positive X-axis, and finally the positive Y-axis.
6	To save a Waypoint position, the operator must press and hold the Freedrive Button (located on the back of the Teach Pendant) while manually pulling the robot arm to the desired position; then tap the OK button on the screen to save the position.
7	Payload and CoG settings are crucial for kinematic accuracy and preventing unwanted robot movement. If set incorrectly, the robot arm may move (fall) unexpectedly when the Freedrive button is pressed.
8	Failure to correctly Set Mounting Angle can lead to Protective Stops or unpredictable behavior in Freedrive Mode, as the controller needs this information to accurately calculate the direction of gravity.
9	The Tool Center Point (TCP) and the Payload (weight and Center of Gravity) must be correctly configured in the Installation Tab.
10	The Backup Restore function is accessed via the Hamburger menu → Settings → System → Backup Restore.

7.3 Learning Outcome: Integrate COBOT with CNC Machine

7.3.1 Information Sheet - Identifying CNC I/O Terminals

Objective To identify CNC input and output terminals for specific machine operations.

Key Points

- Study the CNC electrical diagram and identify digital I/O terminals.
- Typical signals:
 - **Input signals (to CNC):** Cycle Start, Reset, Feed Hold.
 - **Output signals (from CNC):** Door Open, Door Close, Cycle Complete.
- Use a multimeter to verify voltage levels (commonly 24 V DC).
- Label each wire and terminal according to the function for traceability.

Operation	CNC Terminal	Signal Type	Description
Door Open	X101	Output	Sends signal to COBOT to open the CNC door.
Cycle Start	Y102	Input	COBOT triggers machine start.
Air Blow	X103	Output	Activates air jet for chip removal.

7.3.2 Information Sheet - Performing Signal Wiring and I/O Configuration

Objective To perform signal wiring between CNC and COBOT systems and configure inputs/outputs using the teach pendant.

Key Steps

1. Plan the wiring layout based on both systems' I/O diagrams.
2. Use relay isolation or an optocoupler to protect circuits from voltage mismatch.
3. Connect CNC output terminals to COBOT input ports (for feedback).
4. Connect COBOT output ports to CNC input terminals (for control actions).
5. Test signal continuity using a multimeter before powering on.
6. Label all cables with tags for maintenance reference.

Teach Pendant Configuration

1. Open **I/O Configuration Menu**.
2. Assign each robot input/output to corresponding CNC function.
3. Example:
 - Input 1 → “Cycle Complete”
 - Output 2 → “Door Open”
4. Save mapping and test signals using simulation mode.

Verification Test

Run Manual Test Cycle:

- COBOT sends Door Open → CNC door opens.
- COBOT sends Cycle Start → CNC begins machining.
- CNC sends Cycle Complete → COBOT proceeds with unloading.

7.3.3 Information Sheet - Simulate Robot Programs Using URSim in a Virtual Environment

This guide details the process of installing, setting up, and launching the URSim simulation software within a virtual environment, specifically using Oracle VM VirtualBox on a Microsoft Windows operating system.

Part 1: Download and Install Oracle VM VirtualBox

URSim is designed for the Linux operating system. To run it on another operating system like Windows, a virtual machine (VM) is required. Oracle VM VirtualBox is chosen because it is currently free of charge.

1. **Download Oracle VM VirtualBox:** Go to the specified Oracle VM download page and download the latest Oracle VM VirtualBox installer.
2. **Install VirtualBox:** Launch the downloaded installer and follow the suggested installation procedure.

Part 2: Download the URSim Virtual Disk Image (VDI)

The Virtual Disk Image (VDI) contains the Linux operating system and the Universal Robots software necessary to simulate PolyScope and physical robots.

1. **Navigate to Download Page:** Go to the Universal Robots download page (universal-robots.com)
2. **Select Options:** Select the following options from the menu:
 - eSeries (or CB3 for the previous robot version)
 - Software

Module-7 : Operate Collaborative Robot (COBOT)

- Offline Simulator
- Non-Linux
- URSim for non-Linux version 5.8 (or newer)

3. **Download VDI:** Click on the URSim_VIRTUAL-5.8.0.10253.rar file (or newer version) at the bottom of the page to start the download.
4. **Extract Files:** Once downloaded, extract the contents of the .rar file to a directory of your choice (A RAR extractor like 7-zip may be needed).

Part 3: Set Up the Virtual Environment in VirtualBox

After downloading the necessary software and files, Oracle VM VirtualBox must be configured to launch the URSim VDI.

1. **Launch VirtualBox:** Open Oracle VM VirtualBox.
2. **Add New VBoxImage:** Click the 'New' button to add a new VBoxImage.

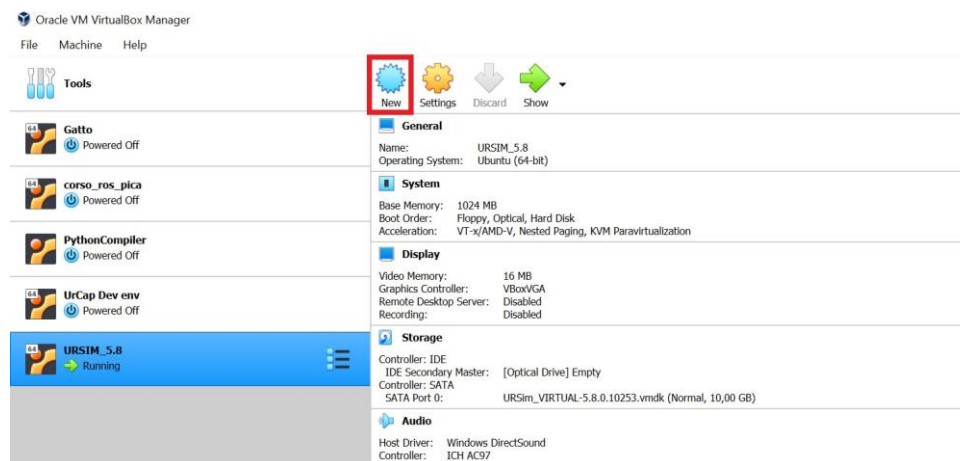


Figure 7.19: Starting polyscope virtual box image

3. **Name and OS Selection:**
 - Choose a name, e.g., URSIM.
 - Set the Type entry to Linux.
 - Set the Version to Ubuntu 64 Bit.
4. **Select RAM:** Select a sufficient amount of RAM memory. A minimum of 768 MB should be selected; if possible, reserve more, specifying a value equal to a power of 2 (e.g., 1024, 2048). Click 'Next'.
5. **Use Existing Hard Disk:** In the next window, select 'Use an existing virtual hard disk file'.
6. **Select VDI File:** Click the folder icon with the green arrow, then click 'Add'. Navigate to the folder where the URSIM files were extracted and select the first element (the VDI file).

7. **Create VM:** Click 'Open', then 'Choose', and finally 'Create'. A new entry with the selected name (e.g., URSIM) should now appear on the left side of the Oracle main page.

Part 4: Configure VDI Settings for Correct Execution

Specific settings must be adjusted to ensure the VDI runs correctly.

1. **Access Settings:** Select the newly installed VDI and click on 'Settings'.
2. **Change Display Controller:**
 - Click on 'Display' on the left side.
 - Change the Graphics Controller to VBoxSVGA. (Note: Ignore the warning "Invalid settings detected." This is done because the default VMSVGA option sometimes causes screen size issues).
3. **Set Up Shared Folder (Optional):** This step is only necessary if file transfer between the virtual machine and the Windows system is planned.
 - Select 'Shared Folder' on the left side.
 - Click on the folder icon with a green cross.
 - Specify the Folder path (the Windows folder for transfers), choose a Folder Name, check Auto Mount, and specify the Mount Point in the Linux system (remember to start with a " ", which can be typed as Alt + 126 on the keypad).
 - Click 'OK' twice to close the Settings window.

Part 5: VDI First Launch and Shared Folder Access

1. **Launch VDI:** In Oracle VirtualBox, select the new VDI and click on 'Start'. The Linux OS should appear.
2. **Adjust Screen Size:** If the screen is small, expand the window by clicking the top right square and dragging the icons (URSim UR16 and Programs UR16) to visualize them correctly.
3. **Access Shared Folder (If Configured):** If a shared folder was created in Part 4, open a terminal and type the following line:
 - `'sudo adduser ur vboxsf'` (Refer to Part 7 if the keyboard layout needs adjustment).
4. **Restart VM:** Press Enter, then close and restart the virtual machine.
5. **Locate Shared Folder:** Once restarted, the shared folder is located in the /media directory. Access it by clicking the bottom-left icon, navigating to accessories, and opening File Manager PCManFM. Click the up-arrow twice to reach the parent directory, then double-click media to find the shared directory.



Figure 7.20: URSim Linux OS launch screen.

Part 6: Launching PolyScope

Launching PolyScope and starting the robot simulation is straightforward.

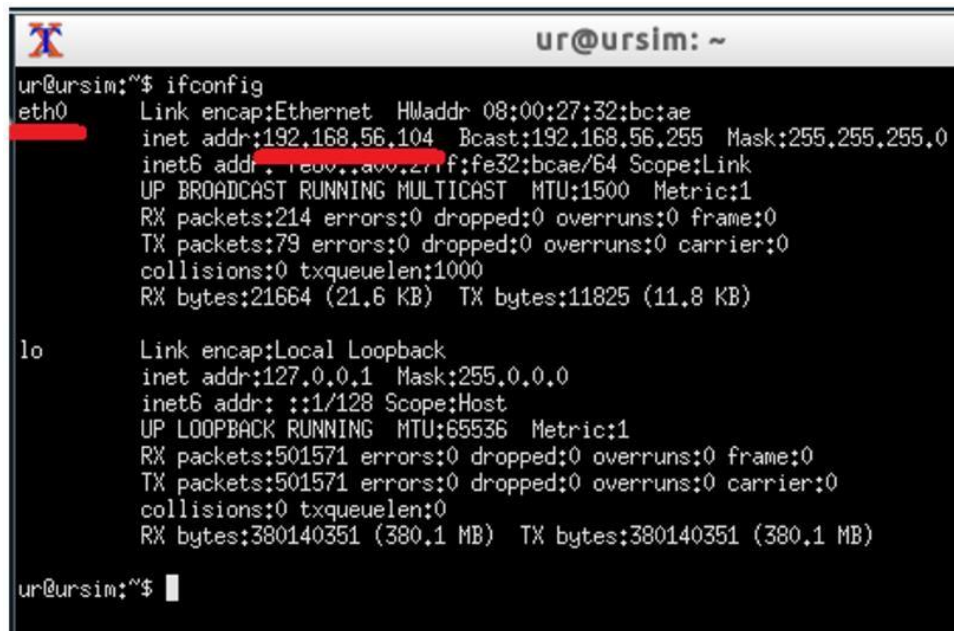
1. **Select Robot Type:** Choose to simulate one of the four types of UR Cobots: UR3e, UR5e, UR10e, or UR16e.
2. **Launch Simulation:** Double click on the desired robot icon from the Virtual Machine desktop. A window simulating the Teach Pendant will appear.
3. **Program Location:** The folders labeled 'Programs URx' contain .urp programs, .scripts, and .urcap files used by the simulator. This is where you will place files received from a real robot or via email to run them in the simulator.

Part 7: Change Keyboard Layout (Optional)

The default keyboard layout is US. To change it, click the US symbol on the bottom right corner of the screen, click "add" to add a new layout, and use the Up/Down buttons to set the priority, making the top one the default.

Part 8: Set Up Network Adapter (Advanced Use)

This step is only necessary if communication with the Windows environment or external devices is required.

A terminal window titled 'ur@ursim: ~' showing the output of the 'ifconfig' command. The output displays details for two network interfaces: 'eth0' (Ethernet) and 'lo' (Local Loopback). The 'eth0' interface has an IP address of 192.168.56.104. The 'lo' interface has an IP address of 127.0.0.1. The terminal text is as follows:

```
ur@ursim:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:32:bc:ae
          inet addr:192.168.56.104  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27:f:fe32:bcae/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:214 errors:0 dropped:0 overruns:0 frame:0
          TX packets:79 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:21664 (21.6 KB)  TX bytes:11825 (11.8 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:501571 errors:0 dropped:0 overruns:0 frame:0
          TX packets:501571 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:380140351 (380.1 MB)  TX bytes:380140351 (380.1 MB)

ur@ursim:~$
```

Figure 7.21: Finding IP addresses for URSim and Windows environment.

1. **Choose Adapter:** In Settings –Network, choose either "Host Only Adapter" or "adapter with bridge," depending on whether communication is needed with the host software or an external device.
2. **IP Addresses:** The Windows environment and the URSIM virtual machine will have two different IP addresses for communication.
3. **Retrieve Windows IP (for VM to Windows communication):**
 - Open the Windows command prompt (cmd) and type ipconfig.
 - Search for the "Ethernet Adapter Virtual Box Host Only network." This IP address is used when communicating from the virtual machine to the Windows environment.
4. **Retrieve VM IP (for Windows to VM communication):**
 - Start the virtual machine.
 - Open the Linux terminal (e.g., UXTerm via the bottom right icon → System Tools) and type the command ifconfig.
 - The IP address shown inside eth0 is the virtual machine's IP address.
5. **Verify Connection:** The connection can be verified by using the ping command from the Windows command prompt to the Virtual Machine IP, and vice versa from the Virtual Machine terminal to the Windows IP.

7.3.4 Self-Check Questions

No.	Question
1	When integrating a Cobot with a CNC Machine, what is the primary safety purpose of using relay isolation or an optocoupler in the signal wiring?
2	What two types of signals are typically exchanged between the CNC machine (Output) and the Cobot Controller (Input)? Give an example of a CNC output signal.
3	Describe the function of URSim and explain why a Virtual Machine (VM) like VirtualBox is required to run it on a Windows PC.
4	In the URSim setup, what operating system is the Cobot simulation software designed for, and which specific Linux distribution is typically used?
5	When configuring the VirtualBox display settings for URSim, which Graphics Controller should be selected, and why might this setting deviate from the default option?
6	In a CNC-Cobot integration test cycle, if the Cobot sends a "Door Open" signal, what two subsequent actions must occur to verify correct communication?
7	What specific file type, containing the Linux OS and the Universal Robots software, must be downloaded from the Universal Robots website to set up URSim?
8	When connecting the CNC to the Cobot, what is the general rule regarding the CNC Output terminals and the Cobot Input ports?
9	What are two examples of signals the Cobot sends (Output) to the CNC Machine (Input) for controlling the machining process?
10	After launching the URSim VDI in VirtualBox, which desktop icon is selected to start the simulation of a specific Cobot model (e.g., UR5e)?

7.3.5 Answer Key

No.	Answer / Guidance
1	Relay isolation or an optocoupler must be used to protect circuits from voltage mismatch between the CNC and Cobot systems.
2	CNC Output terminals send signals or data to COBOT Input ports. An example of a CNC output signal is "Door Open" or "Cycle Complete".
3	URSim is simulation software used to simulate PolyScope and physical robots in a virtual environment. A Virtual Machine (VM) like VirtualBox is required because URSim is designed for the Linux operating system, and the VM allows it to run on a Windows PC.
4	URSim is designed for the Linux operating system, and VirtualBox settings typically specify the version as Ubuntu 64 Bit.
5	The Graphics Controller must be changed to VBoxSVGA. This is necessary because the default VMSVGA option sometimes causes screen size issues.
6	The CNC door must open, and the Cobot must wait for the CNC to confirm the Door Open state (if configured) before proceeding to the next step, which may involve the Cobot sending a "Cycle Start" signal.
7	The Virtual Disk Image (VDI) file, such as URSim_VIRTUAL-5.8.0.10253.rar (or newer), must be downloaded.
8	CNC Output terminals should be connected to Cobot Input ports to provide feedback from the CNC to the Cobot.
9	The Cobot sends signals such as "Cycle Start" (to begin machining) and "Door Open" (to open the CNC door).
10	The user double-clicks on the desired robot icon (e.g., UR5e) from the Virtual Machine desktop to launch the Teach Pendant simulation window.

7.4 Learning Outcome: Program Tasks for Collaborative Robots

7.4.1 Information Sheet - Programming Node Functions in the PolyScope Graphical User Interface

The PolyScope programming interface provides a variety of nodes that can be used to create robot programs. These nodes are categorized into basic program nodes, advanced program nodes, templates, specialized functions, and nodes provided by URCaps for specific applications.

Basic Program Nodes

The basic program nodes serve as the fundamental building blocks for robot programs:

- **Move:** This command controls the robot's motion through its underlying way-points and defines the speed and acceleration between those points. It includes movement types such as **MoveJ** (movements calculated in joint space, resulting in a curved path), **MoveL** (linear Tool Center Point (TCP) motion), and **MoveP** (linear motion with constant speed and circular blends, intended for process operations like gluing or dispensing).

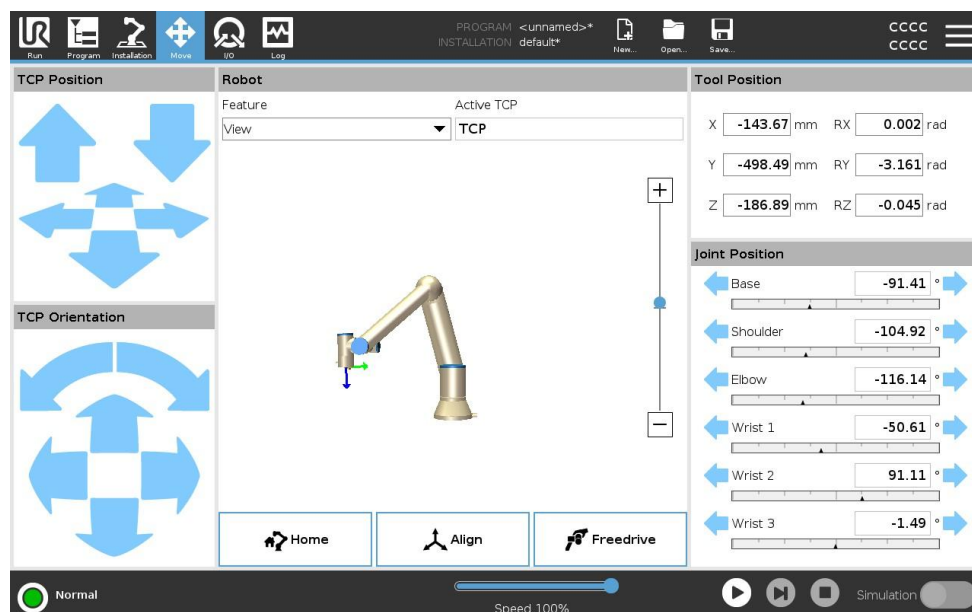


Figure 7.22: Manual jogging, aligning and homing of UR cobot

- **Direction:** This node specifies a linear motion relative to feature axes or TCPs and continues until an **Until** condition is met.
- **Wait:** This function pauses the program for a given amount of time, an I/O signal, or an expression.
- **Set:** This node is used to set digital or analog outputs to a specified value, or to send a single pulse for digital outputs. It can also be used to adjust the payload mass of the robot arm to prevent protective stops, or to modify the active TCP.

- **Popup:** This displays a message on the screen, pausing the program until the user presses the "OK" button. If the "Halt program execution" option is selected, the program halts at this point.
- **Halt:** This command explicitly stops the program execution at its point of insertion.
- **Comment:** This allows the programmer to add a line of text to the program for reference; it does not execute any action during runtime.
- **Folder:** This node is utilized to label and organize specific segments of a program, which helps to simplify the Program Tree structure and improve readability.

Advanced Program Nodes

The advanced nodes introduce flow control and external communication capabilities:

- **Loop:** This node controls the repetition of underlying program commands, allowing for infinite loops, a specified number of repetitions, or looping as long as a given condition remains true.
- **If:** This function allows the robot's behavior to change based on the evaluation of sensor inputs or variable values (expressions). If the condition is determined to be True, the statements within the *If* command are executed.
- **SubProgram:** This holds program parts that are required in multiple locations. A call to a SubProgram runs the lines within it and then returns to the following line in the main program.
- **Assignment:** This node assigns the computed value of the right-hand side expression to the variable specified on the left-hand side.
- **Script:** This provides options to write a single line of URscript code (Line) using the Expression Editor, or to load and execute external URscript files (File). Functions and variables declared in a URscript file are available throughout the program in PolyScope.
- **Event:** This is utilized to monitor an input signal and trigger an action or set a variable when that signal transitions to high. Events are checked every control cycle (2ms).
- **Thread:** This creates a parallel process to the main robot program, enabling the control of an external machine independently of the robot arm's movements.
- **Screwdriving:** This specialized node facilitates the programming of tightening or loosening actions for an attached screwdriver or nutrunner, including defining forces, speed limits, and error conditions.
- **Switch:** This allows the robot to change its behavior based on sensor inputs or variable values by evaluating a base condition against several defined cases. If a case matches, the lines within that case are executed.

- **Timer:** This measures the execution time for specific program segments, storing the elapsed time in a program variable.
- **Home:** This moves the robot arm to a user-defined Home position using joint angles.

Templates and Specialized Functions

PolyScope also offers specialized templates to simplify complex applications:

- **Palletizing:** A template designed to easily program palletizing and depalletizing operations, managing repeatable pick-and-place actions across multiple layers and patterns.
- **Seek:** This function employs a sensor (such as a pressure sensor or push button) to accurately determine when the robot has reached the correct position to handle items, often used for stacks with varying item thickness or unknown positions.
- **Force:** Used in the robot workspace to enable compliance and force application along selectable axes. The robot automatically adjusts its position to maintain a required force against an environment or workpiece.
- **Conveyor Tracking:** This enables the robot arm to track the movement of up to two configured conveyors. All movements under this node are relative to the conveyor's motion.

Remote TCP and Toolpath URCaps Nodes

For applications requiring specialized TCP handling or complex trajectories, the Remote TCP and Toolpath URCap provides these motion nodes:

- **RTCP_MoveP:** Similar to a standard MoveP, this node defines the tool speed and acceleration for motion executed relative to a Remote TCP.
- **RTCP Circle move:** This node is added to an RTCP_MoveP to facilitate circular movements.
- **RTCP Waypoint:** These waypoints enable linear movement with constant speed and circular blends relative to the fixed Remote TCP.
- **Toolpath Moves (Remote TCP or Regular TCP):** These nodes automatically generate robot motions based on imported G-code toolpath files, allowing the robot to follow complex trajectories accurately.

7.4.2 Information Sheet - Detailed Guide for Programming Palletizing in PolyScope

The PolyScope GUI utilizes a specialized template for programming palletizing and depalletizing operations, which allows the robot to handle items across multiple layers and patterns.

Step-by-Step Guide to Creating a Palletizing Program

To successfully configure a palletizing program, you must first define the actions, patterns, and item handling sequences:

1. **Choose Reference Plane:** Determine whether the program will use a manually taught Feature or the robot Base as the reference plane.
2. **Insert Template:** In the Program Tab, locate Templates and tap **Palletizing** to add the node to the Program Tree.
3. **Select Action:** On the Palletizing screen, choose the desired action:
 - Select **Palletizing** to organize items onto a pallet.
 - Select **Depalletizing** to remove items from a pallet.
4. **Configure Pallet Properties:** Under Pallet Properties, specify the **Name**, **Feature**, **Object Height**, and **Item Counter** name. Optionally, select the **Remember last item location** box if the robot needs to restart at the last handled item when the program stops.
5. **Add Sequence Actions:** Under Actions, you can add additional actions to be performed:
 - **Add Action Before Palletizing:** Actions performed before starting to palletize.
 - **Add Action After Palletizing:** Actions performed after finishing palletizing.
6. **Designate Patterns:** In the Program Tree, tap the **Patterns** node to designate the required patterns for your layers. Available patterns are **Line**, **Grid**, or **Irregular**. This screen also allows selection of whether to include a separator between layers.
7. **Teach Pattern Positions:** Tap the specific pattern node(s) on the Program Tree to teach the layer-specific positions. All positions must be taught at the bottom of the pallet.
 - **Line Pattern:** Teach **Start_Item_1** and **End_Item_1**. Insert the number of items in the sequence using the Items text box.
 - **Grid Pattern:** Teach **Corner_Item_1**, **Corner_Item_2**, **Corner_Item_3**, and **Corner_Item_4**. Insert the number of rows and columns.
 - **Irregular Pattern:** Teach positions such as **Item_1**, **Item_2**, etc. Use **Add Item** to include new items in the sequence.
8. **Configure Layers:** Tap the **Layers** node to configure the layers of the palletizing sequence. Use the Choose Pattern drop-down menu to select the pattern for each layer. Layers must be added in the correct order as they cannot be reordered later.
9. **Define Item Handling:** Tap the **At Each Item** node. Choose between the default **At Each Item Wizard** or **Manually Configure At Each Item** to define the sequence of actions for handling each piece.

Configuring Item Handling (At Each Item)

The process specifies crucial waypoints for movement during item handling: the ReferencePoint, the Approach Waypoint, ToolActionPoint Waypoint, and Exit Waypoint.

A. At Each Item Wizard Sequence

If you select the At Each Item Wizard, follow these steps:

1. Start by tapping the At Each Item node and selecting **Next** on the subsequent screen.
2. Tap **Move Here**. Hold the **Auto** button or use the **Manual** button to move the robot to the **ReferencePoint**. Tap **Continue**, then **Next**.
3. Tap **Set Waypoint** to teach the **Approach Waypoint**.
4. Repeat the movement step (Step 2).
5. Tap **Set Waypoint** to teach the **Exit Waypoint**.
6. Tap **Finish**.
7. Add appropriate gripper action nodes within the **Tool Action** folder in the Program Tree.

B. Manual Configuration of At Each Item

If you choose Manual Configuration:

1. On the At Each Item start screen, tap **Manual Configuration**.
2. Select a **Pattern** and a **ReferencePoint** item from the drop-down menus. Tap the **Use this ReferencePoint** button.
3. Move the robot to the ReferencePoint by tapping **Move Here**.
4. Tap the **Approach** node to teach the Approach Waypoint (this waypoint keeps the same orientation and direction regardless of different items' orientations).
5. Tap the At Each Item node, and repeat the movement step (Step 3).
6. Tap the **Exit** node to teach the Exit Waypoint.
7. Add appropriate gripper action nodes inside the **Tool Action** folder.

Adding Separators Between Layers

If you wish to place separators (such as Styrofoam or paper) between layers:

1. **Define Separator Properties:** Select the **Patterns** node in the Program Tree, select **Separator**, and define the **Separator Height**. The program will not run if the height is undefined.
2. **Assign Separator to Layers:** Select **Layers** in the Program Tree to choose which layers require separators (they are automatically placed between each layer).
3. **Teach Separator Position:** Tap the **Separator** node, then tap **Set Separator** to teach the Separator Position.
4. **Configure Sequence:** Choose between the default **Separator Wizard** or **Manually Configure the Separator** sequence.

The resulting Program Tree template under the Separator Action node includes folders for **Pick Up Separator**, **Drop Off Separator**, and **Tool Action**.

7.4.3 Information Sheet - Integration and Operation of the AG-95 Gripper with Universal Robots

The integration of the AG-95 gripper involves mechanical and electrical installation, UR-Cap software setup, activation through the PolyScope interface, and control of parameters such as position, force, and speed using program nodes or URScript commands.

I. Mechanical and Electrical Connection (Hardware Installation)

The robot system consists of the robot arm, Control Box, and the Teach Pendant. The AG-95 is a two-finger adaptive electric gripper designed for industrial use.

Mechanical Installation

1. Install the standard flange onto the end of the robot arm. Accurate positioning of the flange and manipulator end can be achieved using $\Phi 6 \times 10$ positioning pins.
2. Mount the AG-95 gripper onto the flange and secure it using screws on both sides. $\Phi 3 \times 10$ positioning pins are used to align the clamping jaws.

Electrical Connection The AG-95 gripper supports several communication modes, including Modbus RTU, IO control, and others via protocol converters such as TCP/IP, USB, RS485, CAN2.0A, and EtherCAT.

- **UR e-Series Tool I/O (TCI):** The DH gripper supports E-Series Tool I/O communication control and connects to the UR tool communication interface (TCI). For dual-pin power supply, use connector 03.05.0136.
- **Direct Terminal Connection:** When connecting directly to the robot terminal aviation socket, ensure a 24 V supply with at least 1 A current capacity. The UR Tool I/O ports near the tool flange provide power (0 V/12 V/24 V on the gray wire, and GND on the red wire). The 24 V mode current must not exceed 2 A from the Control Box.
- **Control Box Connection:** Power wires (24 V and GND) are drawn from the Control Box's DC regulator output. For RS485 communication, connect 485A and 485B to the T/R+ and T/R- inputs of a 485-to-USB converter, which interfaces with the Control Box.

II. URCap Software Installation

The installation procedure depends on the robot series.

UR E-Series (Minimum Software Version 5.4.3)

1. Insert the DH USB stick into the Teach Pendant.
2. Open **Settings** → **System** → **URCaps**.
3. Click the “+” icon, select the URCap file (e.g., DH-Grippers-PGC-2.4.6.urcap), and click **Open**.
4. Click **Restart** to finalize installation.

III. Gripper Activation and Configuration

After installation, the gripper must be configured and activated in PolyScope.

Configuration (E-Series Specific)

1. Click **Installation**.
2. Select **General** → **Tool I/O**.
3. Choose the controller “DH-PGC”.

Connecting and Activating

1. In PolyScope, open **URCaps** → **DH-PGC**.
2. Select the number of grippers and the connection method under “Relate to” (e.g., TCI or USB).
3. Click **Connect** and wait for the status to show “Connected”.
4. Click **Activate** and wait until activation completes.

Status Feedback The AG-95 provides feedback through indicator LEDs:

- **Uninitialized/Inactivated:** Red light blinks.
- **Initialized/Activated:** Blue light stays constant.
- **Command Received:** Red light flashes briefly (appears purple if blue is also on).

Scripted Auto-Connection and Initialization To automatically initialize the gripper, insert the following URScript commands in the *Before Start* section:

```
dh_pgc_scan()  
dh_pgc_relate_device(1, "TCI", 1)  
dh_pgc_connect(1)  
dh_pgc_set_activate(1)  
dh_pgc_wait_until_activated(1)
```

IV. Setting Position, Force, and Speed

The AG-95 can be controlled either through URCap nodes or URScript.

Using URCap Program Node

1. In the **Program** tab, go to **URCaps** → **DH-PGC Move**.
2. Click **Edit** to adjust position, force, and speed; then click **Save**.
3. Click **Move Position** to execute the gripper action.

Controlling via URScript Use the following script functions to control the gripper by index (Index):

- **Position Control:**

- Set position: `dh_pgc_set_position(Index, Position)`
- API version: `DhSetPosition(position)` (range: 0-1000%)
- Read position: `var = dh_pgc_get_position(Index)`
- Open/Close: `DhOpen(time)` or `DhClose(time)` (time in ms)

- **Force Control:**

- Set force: `dh_pgc_set_force(Index, Force)`
- API version: `DhSetForce(force)` (range: 20-100)
- Read force: `var = dh_pgc_get_force(Index)`

- **Speed Control:**

- Set speed: `dh_pgc_set_speed(Index, Speed)`
- Read speed: `var = dh_pgc_get_speed(Index)`

Script commands can also be used to synchronize motion by waiting for activation or object gripping events.

7.4.4 Self-Check Questions

No.	Question
1	Identify the four key performance criteria that define Learning Outcome 7.4 (Program Tasks for Collaborative Robots).
2	What is the primary motion distinction between the MoveJ and MoveL nodes in PolyScope programming?
3	Describe the function of the Wait Node in PolyScope, and name two conditions it can pause the program for.
4	Contrast the purpose and execution of the If node and the Switch node in control flow programming.
5	What essential Waypoint positions must be defined for an item when manually configuring the At Each Item step in a Palletizing template?
6	Why would a programmer use the Script node, specifically the Line or File options, in PolyScope?
7	What mechanism must be used to execute a program segment independently of the main robot arm movements, for tasks such as controlling an external machine?
8	When programming the Palletizing template, what must be defined using the Patterns node for a Grid Pattern?
9	What command node is used to assign a computed value from an expression (e.g., $1 + 1$) to a variable in PolyScope?
10	The Force node is used to enable compliance along selectable axes. What safety feature might be triggered if the speed of motion is too high when using the Until Tool Contact node?

7.4.5 Answer Key

No.	Answer / Guidance
1	The four key performance criteria are: Appropriate programming nodes are selected; Logical flow is established using loops and conditional statements; Positioning waypoints and safe home positions are set; and Input/output data is used to control peripheral devices.
2	MoveJ calculates motion in joint space, resulting in a curved path, while MoveL results in linear motion of the Tool Center Point (TCP).
3	The Wait Node pauses the program flow. It can pause for a given amount of time or until an I/O signal is received (or an expression is met).
4	The If node allows robot behavior to change based on the evaluation of sensor inputs or variable values, executing statements if the condition is True. The Switch node changes behavior by evaluating a base condition against several defined cases, executing the lines within the matching case.
5	The manual configuration of the At Each Item process requires defining the ReferencePoint, the Approach Waypoint, the ToolActionPoint Waypoint, and the Exit Waypoint.
6	The Script node is used to write a single line of URScript code (Line option) or to load and execute external URScript files (File option).
7	The Thread node creates a parallel process to the main program, enabling the control of an external machine independently of the robot arm's movements.
8	For a Grid Pattern, the user must teach the four corner positions (Corner_Item_1 through Corner_Item_4) and insert the number of rows and columns.
9	The Assignment node assigns the computed value of the right-hand side expression to the variable specified on the left-hand side.
10	If the speed of motion is too high when using the Until Tool Contact node, it may trigger a protective stop before the contact condition takes effect.

7.5 Learning Outcome: Troubleshoot and Optimize Robot

7.5.1 Information Sheet- Safety Instructions for Universal Robots Systems

The safety instructions for Universal Robots (UR) systems cover mandatory integration requirements, general warnings and cautions, electrical warnings, operational precautions, and maintenance procedures. Compliance with these instructions is mandatory, and a comprehensive risk assessment must be performed for every robot application.

I. Mandatory Safety Assessments and Integrator Responsibilities

The UR robot is classified as partly completed machinery, requiring the integrator to ensure safety measures are in place before operation.

1. **Risk Assessment (Mandatory):** The integrator must perform a risk assessment for the complete robot system prior to powering the robot on for the first time. This assessment must cover all tasks, including teaching, troubleshooting, maintenance, and normal operation.
2. **Safety Configuration and Validation:** The integrator is responsible for setting the correct safety settings in the software. The robot application must be validated to ensure it is correctly designed, installed, and integrated.
3. **Documentation:** The integrator must retain all documentation, including the application risk assessment and the robot manual.
4. **Preventing Unauthorized Changes:** The integrator must ensure that safety measures are not modified, and unauthorized access to the safety configuration must be prevented by enabling and setting password protection.

II. Pre-Use and Operational Safety Checks

Before operating the robot application for the first time or after any modifications, specific tests must be conducted:

1. **Connection Verification:** Verify that all safety inputs and outputs are correctly connected and functioning as intended, including devices common to multiple machines.
2. **Emergency Stop Test:** Test all emergency stop buttons and inputs to verify the robot stops and the brakes engage. If System Emergency Stop outputs are used, test their ability to bring the whole system to a safe state.
3. **Safeguard Test:** Test safeguard inputs to verify robot motion stops. If configured, check that Safeguard Reset must be activated before motion resumes.
4. **Mode and Enabling Device Test:**
 - Examine the initialization screen to verify that Reduced Mode can switch the safety mode.

- Test that the operational mode switches correctly, observing the icon change in the user interface.
 - Test the 3-position enabling device (3PE) to verify it must be pressed to enable motion in Manual Mode, and that the robot is under reduced speed control.
5. **Installation Settings:** Always verify the actual payload and installation settings (e.g., mounting angle) are correct before starting up the robot arm. Incorrect settings can lead to hazardous operation or damage.

III. General Warnings and Cautions

1. **Mounting and Tooling:** Verify the robot arm and tool/end effector are properly and securely bolted in place. Ensure the tool is constructed so it cannot create a hazardous situation by dropping a part unexpectedly.
2. **Workspace and Movement:** Verify the robot application has ample space to operate freely. Keep outside the reach (workspace) of the robot, especially when it is about to start-up.
3. **Clothing and Hair:** Avoid wearing loose clothing or jewelry when working with the robot; tie back long hair.
4. **Physical Condition and Damage:** Avoid using the robot if it is damaged. Do not expose the robot to permanent magnetic fields.
5. **Heat Warning:** The robot arm and Control Box generate heat during operation. Do not handle or touch the robot during or immediately after operation; allow it to cool down by powering it off and waiting one hour. Check the temperature on the log screen before handling or touching the robot.
6. **Emergency Stop Use:** Inform users of the location of the emergency stop button(s) and how to activate them in case of emergency or abnormal situations.
7. **Modifications:** Never modify the robot or alter e-Series end caps, as modifications can create unforeseen hazards.
8. **Unintended Use:** Only use the industrial robot in accordance with its intended use and specifications. Misuse (e.g., medical use, use in explosive environments, or using the robot outside stated specifications) is prohibited.

IV. Electrical and Power Safety Warnings

1. **Isolation (Lockout-Tagout):** Before performing any cleaning, maintenance, or repair, remove the mains input cable from the Control Box to ensure complete power isolation. Ensure other equipment does not supply power to the robot I/O when the system is locked out.
2. **Control Box Hazards:** Avoid sticking fingers behind the internal cover of the Control Box. Avoid disassembling the power supplies inside the Control Box, as high voltages (up to 600V) can be present for several hours after switching off.

3. **Grounding:** Ensure the robot is grounded correctly (electrical connection to ground). Use the M6-size screw connections marked with earth symbols inside the Control Box for Protective Earth (PE). Check the earth connection before re-powering the system.
4. **Cabling:** Ensure all cables are connected correctly before the Control Box is powered. Only use the original cables supplied with the robot. Do not extend or modify the original Robot Cable.
5. **Liquid Contact:** Ensure the Control Box, Teach Pendant, and cables do not come into contact with liquids. Place the Control Box (IP44) and Teach Pendant (IP54) in environments suited for their IP ratings.
6. **Safety Wiring:** Never connect safety signals to a PLC that is not a safety PLC with the correct safety level. All safety-related signals must be constructed redundantly (two independent channels) and kept separate so a single fault cannot lead to loss of the safety function.
7. **Tool Power:** Ensure tools and grippers are connected correctly so that if there is an interruption of power, no hazards occur.

V. Maintenance and Cleaning Safety

1. **Maintenance Execution:** Maintenance, inspection, calibration, and repair must be performed according to the latest Service Manual. Only authorized system integrators or Universal Robots should perform repairs.
2. **Component Replacement:** Replace faulty components only with new components bearing the same article numbers or equivalent components approved by Universal Robots.
3. **Post-Maintenance Checks:** After maintenance, the correct functioning of all safety functions must be tested. If any safety configuration is changed, the overall safety approval process, including the risk assessment, must be updated.
4. **Cleaning Agents:** Use a cloth with water, 70% isopropyl alcohol, or 10% ethanol alcohol for cleaning the robot arm or Teach Pendant. Do not use bleach.
5. **Cleaning Methods:** Never use compressed air to clean the robot arm, as this can damage components.
6. **Grease Handling:** During grease work, ensure the area is well ventilated, wear protective gloves (oil resistant), use eye protection, and ensure an eye wash station is nearby.

7.5.2 Information Sheet-Troubleshooting and Maintenance Steps for Universal Robots Systems

Troubleshooting a Universal Robots (UR) system involves rigorous safety validation checks, diagnosis during operation via the PolyScope interface, and regular physical inspection and maintenance. It is mandatory for the integrator to conduct a risk assessment before configuring safety settings or operating the robot for the first time.

I. Pre-Use Assessment and Safety Configuration Checks

The following tests must be performed before initial use or after any modification to the robot application:

1. **Verify Connections and Functionality:** Verify that all safety inputs and outputs are correctly connected. Test all connected safety inputs and outputs, including devices shared across multiple machines, to ensure they function as intended.
2. **Emergency Stop Test:** Test emergency stop buttons and inputs to confirm they stop the robot and engage the brakes. If System Emergency Stop outputs are utilized, test that the connected system can be brought to a safe state.
3. **Safeguard Test:** Test safeguard inputs to verify that robot motion stops. If a safeguard reset is configured, verify that it must be activated before motion resumes.
4. **Mode and Speed Control Test:**
 - Examine the initialization screen to test that Reduced Mode can successfully switch the safety mode.
 - Test that the operational mode switches correctly, observing the icon change in the top-right corner of the user interface.
 - Test the 3-position enabling device (3PE): verify that it must be pressed to enable motion in Manual Mode, and that the robot operates under reduced speed control.
5. **Output Signal Detection:** Test that external systems connected to outputs such as Robot Moving, Robot Not Stopping, Reduced Mode, or Not Reduced Mode can detect and respond to these output changes.
6. **Installation Settings Verification:** Before starting the robot arm, verify that the actual payload and installation settings (e.g., mounting angle) are correct. Incorrect settings can lead to hazardous operation or damage.

II. Operational Troubleshooting and Error Handling

Responding to Errors and Stops:

- **Immediate Action:** If the software prompts an error, immediately press the emergency stop button, record the conditions leading to the error, locate the corresponding error codes on the log screen, and contact the supplier.
- **Safety System Violation:** If the robot detects a fault or violation in the safety system (e.g., a wire cut in the Emergency Stop circuit or a safety limit exceeded), a Stop Category 0 (immediate power removal) is initiated.
- **Recovery Mode Activation:** If a safety limit is violated or if the system is outside a safety limit at start-up (e.g., joint position limit violation), the robot enters Recovery Mode. In this mode, programs cannot be run, but the robot arm can be moved manually (using Freedrive or the Move tab) back within safety limits. Caution is necessary as limits for joint positions, safety planes, and tool orientation are disabled in Recovery Mode.

- **Tool Contact Detection:** If using the Until Tool Contact node, the default speed of motion might be too high, potentially triggering a protective stop before the contact condition takes effect. To prevent this, the speed of motion should be lowered (e.g., to 100 mm/s). Note that vibrations from mounted tools (like vacuum grippers with embedded pumps) may interfere with Until Tool Contact.

Troubleshooting Freedrive Issues:

- **Unexpected Motion during Freedrive:** If the robot mounting or payload settings are incorrect, the robot arm may move (fall) unexpectedly when the Freedrive button is pressed. Ensure payload and mounting settings are configured correctly before operating Freedrive.
- **Structural Issues:** If the stand on which the robot is mounted lacks sufficient static stiffness, it can deflect during motion, causing the robot to miss waypoints or trigger protective stops. Lack of dynamic stiffness can also cause the system to resonate or result in protective stops.

III. Data Logging and Error Reporting

The PolyScope system provides methods to access diagnostic information:

- **Log Tab Monitoring:** The Log tab displays real-time robot health indicators, warnings, and error messages, categorized by severity (Info, Warning, Violation, Fault).
- **Error Reports:** A detailed status report, known as an Error Report, is associated with a log line when a paper clip icon appears.
- **Exporting Reports:** To save the report, select the log line and tap the *Save Report* button to export the data to a USB drive. The exported report contains the user program, history log, installation settings, and a list of running services.
- **Tracked Errors:** Errors that can be tracked and exported include Faults, Internal PolyScope exceptions, Protective Stops, Unhandled exceptions in URCap, and Violations.

IV. Maintenance and Repair Procedures

Safety Precautions for Service:

- **Power Isolation (Lockout–Tagout):** Before performing any cleaning, maintenance, or repair work on the robot or gripper, ensure the robot and gripper power supply are turned off. Remove the mains input cable from the Control Box to ensure complete power isolation (lockout–tagout all power for the complete installation). Ensure no other equipment supplies power to the robot I/O.
- **Electrical Hazards:** Avoid disassembling power supplies inside the Control Box, as high voltages (up to 600 V) can remain present for several hours after switching off. Check the earth connection before re-powering the system. Observe ESD regulations.

- **Component Replacement:** Replace faulty components only with new components bearing the same article numbers or equivalent parts approved by Universal Robots.
- **Post-Maintenance Validation:** After maintenance, safety functions must be immediately revalidated, and their correct functioning must be tested. If any safety configuration is changed, the entire robot system safety approval process (including the risk assessment) must be updated.

Physical Inspection and Cleaning:

- **Robot Inspection:** Inspect the cable connecting the Control Box and robot arm for damage. Check that base mounting bolts are properly tightened. Inspect joint components such as flat rings and blue lids for wear, damage, or cracks.
- **Cleaning:** Do not use compressed air to clean the robot arm. Use a cloth with cleaning agents such as water, 70% isopropyl alcohol, or 10% ethanol; do not use bleach.
- **Gripper Inspection:** Check that finger movement is symmetrical and smooth by testing the stroke and observing whether the claws are stuck or shaken during operation. Check the wear of finger rubber and all screws for rust or damage; replace if necessary.

V. Software Update and URCap Management

Software Management:

- **Installing URCaps:** To install a URCap, access the Hamburger Menu **Settings** **System** **URCaps**. Press the “+” button, select the .urcap file, and press **Open**. Press **Restart** to complete installation.
- **Post-Update Check:** Always test existing programs after a software upgrade, as updates might change trajectories.

7.5.3 Information Sheet- Communication Protocols of Universal Robots (UR) Systems

Universal Robots (UR) systems utilize various electrical interfaces and industrial communication protocols for control, external interaction, and safety signaling. These communications happen primarily via the Control Box I/O, Tool I/O, and network interfaces (Ethernet/Fieldbus).

I. Fieldbus Communication Protocols

UR robots support several industrial computer network protocols for real-time distributed control, configured via the Installation Tab → Fieldbus setup in PolyScope.

A. MODBUS MODBUS is supported for external communication and control.

- **Setup:** The robot acts as a MODBUS client (master). MODBUS client signals (registers or digital) can be set up via specified IP addresses connected to MODBUS servers (slaves).

- **I/O Signals:** The registers include Digital standard general purpose, configurable, tool I/O, Analog standard general purpose, tool I/O, and General purpose registers (Boolean, Integer, Float).
- **Function Codes:** Supported Modbus Function Codes include Read Holding Registers (0x03), Read Multiple Registers (0x04), Write Single Register (0x06), and Write Multiple Registers (0x10).
- **Access/Control via Registers:** Users can read and write data into specific registers, which include basic control registers (initialization, force setting, position, speed, and states) and configuration registers (Modbus parameters and I/O parameters).
- **Network Configuration:** The IP address of the MODBUS unit can be set in the I/O Setup. The robot controller can connect to MODBUS units via Ethernet or RS485 (using a converter).
- **Advanced Options:** Users can configure advanced options like Update Frequency, Slave Address, and Sequential Mode, where the client waits for a response before sending the next request. Update Frequency can be set from 0 to 125 Hz.

B. EtherNet/IP and PROFINET These protocols are supported for connecting the robot to industrial controllers (Scanner Device or IO-Controller).

- **Setup:** The robot can enable connection to an EtherNet/IP Scanner Device or use PROFINET.
- **Connection Loss Action:** If connection is lost, users can select an action: None (program continues), Pause (program pauses), or Stop (program stops).

C. PROFIsafe PROFIsafe (version 2.6.1) allows the robot to communicate with a safety PLC according to ISO 13849, Cat 3 PLd requirements.

- **Safety Functions:** PROFIsafe provides a safe, network-based alternative to physical wiring for safety I/O pins, enabling the safety PLC to trigger safety functions like Emergency Stop or transition to Reduced Mode.
- **State Information:** The robot transmits safety state information to the PLC, including Stop Categories 0, 1, and 2.

II. Tool Communication Interface (TCI)

The tool flange, specifically on Wrist #3, includes an eight-pinned connector providing power and control signals for connected tools, such as grippers and sensors.

- **RS485 Communication:** The Tool Communication Interface (TCI) enables the robot to communicate with an attached tool via RS485 signals. This removes the need for external cabling.
- **Configuration:** TCI settings, including baud rate, parity, stop bits, and idle character parameters, can be enabled and edited in PolyScope → Installation Tab → General → Tool I/O → Communication Interface.

- **I/O Conflict:** When the TCI is enabled, the tool analog inputs become unavailable for I/O Setup or program use.
- **Data Latency:** The latency of messages sent via the tool connector ranges from 2ms to 4ms (from the time the message is written on the PC to the start of the message on the RS485).
- **Baud Rates:** Valid baud rates include 9.6k, 19.2k, 38.4k, 57.6k, 115.2k, 1M, 2M, and 5M.

III. Low-Level Control and Scripting

A. URScript The underlying real-time script language, URScript, controls the robot.

- **Purpose:** URScript includes variables, types, flow control statements, and built-in variables and functions that monitor and control I/O and robot movements.
- **Execution:** URScript programs are executed in real-time in the URControl Runtime Machine (RTMachine), which communicates with the robot at 500Hz.
- **Access:** Script programming can be done via the Script node in PolyScope. Instructions can be found in the Script Manual.

B. Remote Procedure Call (RPC) RPCs allow the robot controller to execute functions defined and running remotely (e.g., on an external PC) using the XMLRPC standard.

- **Functionality:** The controller waits for the remote function to complete during an RPC call. RPC is useful for integrating external devices like vision systems, where the script creates an RPC connection using `rpc_factory`.

C. Dashboard Server The Dashboard Server allows commands to be sent externally, such as changing the Operational Mode between Automatic and Manual, or sending simple commands like starting or loading programs.

IV. Communication for DH Robotics Grippers (General)

The AG-95 gripper supports multiple communication modes, including Modbus RTU and IO mode control. Other protocols like USB, TCP/IP, RS485, CAN2.0A, and EtherCAT can be supported through an optional protocol converter.

- **Data Format:** Communication through non-CAN/I/O interfaces (TCP/IP, USB, RS485) typically uses a 14-byte frame format including a fixed frame header, Gripper ID, an 8-byte Data Segment (the actual command), and a fixed Frame End.

7.5.4 Self-Check Questions

No.	Question
1	According to safety instructions, what must the integrator perform for the complete robot system prior to initial power-on?
2	What must be done with all Emergency Stop buttons and inputs before operating the robot, and what is the expected robot reaction during this test?
3	If a safety limit is violated during operation, what specific Stop Category is initiated, and what is the primary consequence regarding power?
4	A programmer is using the Until Tool Contact node for a precise operation, but the robot triggers a protective stop before contact. What corrective action is recommended?
5	Describe the operational status of the robot when it enters Recovery Mode due to a safety limit violation, and state one restriction on movement in this mode.
6	What two Installation Settings must be verified before starting the robot arm, as incorrect settings can lead to hazardous operation or damage?
7	What specific component should never be used for cleaning the robot arm, and why might this be hazardous to the components?
8	Name two industrial protocols, besides MODBUS, that Universal Robots support for connecting to external controllers (IO- Controller or Scanner Device), configured via the Fieldbus setup.
9	If the robot arm moves (falls) unexpectedly when the Freedrive button is pressed, what two configuration issues are the most likely cause?
10	The Teach Pendant provides access to diagnostic information. Which specific tab displays real-time robot health indicators, warnings, and error messages?

7.5.5 Answer Key

No.	Answer / Guidance
1	The integrator must perform a risk assessment for the complete robot system prior to initial power-on. This assessment must cover all tasks, including teaching, troubleshooting, maintenance, and normal operation.
2	All E-Stop buttons and inputs must be tested to verify the robot stops and the brakes engage. If System Emergency Stop outputs are used, their ability to bring the whole system to a safe state must be tested.
3	A Limit Violation or Fault Detection triggers Stop Category 0, which involves the immediate removal of power and results in an uncontrolled stop.
4	The speed of motion should be lowered (e.g., to 100 mm/s) because a speed that is too high may trigger a protective stop before the contact condition takes effect.
5	In Recovery Mode, programs cannot be run. The robot arm can be manually moved back within limits using Freedrive Mode or the Move tab.
6	The actual payload and the installation settings (e.g., mounting angle) must be verified before starting the robot arm. Incorrect settings can lead to hazardous operation or damage.
7	Compressed air should never be used to clean the robot arm, as this can damage components.
8	EtherNet/IP and PROFINET are supported protocols for connecting the robot to external industrial controllers.
9	If the robot arm moves unexpectedly during Freedrive, it indicates that the robot mounting or the payload settings are incorrect.
10	The Log Tab displays real-time robot health indicators, warnings, and error messages.