Automated Warehouse

Project Overview

Objective: The goal of this project is to design and implement an automated warehouse system using **Ladder Logic programming in TIA Portal**. The system is designed to automate the movement of objects from a conveyor belt to a storage rack, leveraging **programmable logic controllers (PLCs)** and **sensors** to control and monitor the operations. The programming is primarily done using **Ladder Logic** (LD) to handle the sorting, transferring, and placement of objects in the warehouse, optimizing efficiency and reducing manual intervention.

The automation system aims to:

- **Increase efficiency** in warehouse operations by automating the transfer of objects.
- **Reduce manual labor** by allowing the system to independently sort and place objects.
- **Enhance accuracy** in storing objects by using sensors to ensure correct placement.

Scope: The project focuses on automating the process of transferring objects from a conveyor system to specific positions on a storage rack. The automation is controlled using a Siemens PLC with **TIA Portal** as the development platform. The system uses **Ladder Logic** to implement the control sequences for various components, such as:

Conveyor System: The conveyor transports objects through the system, where sensors detect their presence.

- **Sensors**: Used to detect the position of objects on the conveyor, as well as to monitor when objects reach their designated placement location on the rack.
- **Actuators**: These are used to move the objects from the conveyor to their correct rack position. Actuators can include motors, robotic arms, or other mechanical devices.
- **Racking System**: A static structure where objects are placed after being transferred from the conveyor.
- **PLC Control**: The Siemens PLC, programmed in **Ladder Logic**, controls all aspects of the system, including sensor data acquisition and actuator control.

Key Features:

- **Ladder Logic Control**: The entire automation process is controlled by Ladder Logic programming, which mimics electrical relay logic for easy troubleshooting and modification.

- **Real-Time Control**: The PLC reads data from sensors and issues control commands to actuators in real-time, ensuring objects are moved smoothly from the conveyor to the rack.
- **Efficiency & Safety:** The automated system enhances efficiency by reducing manual labor and minimizes downtime with built-in safety features, such as emergency stops and fault detection.

System Architecture

Sensors

Figure 1. Automated Warehouse's Sensors

- **At Entry Sensor**: Detects whether an object is present at the entry point of the conveyor.
- **At Exit Sensor**: Detects whether an object is present at the exit point of the conveyor.
- **At Load Sensor**: Detects whether an object is positioned at the loading point.
- **Forklift Position Sensors**:

    - **At Left Sensor**: Detects whether the forklift is positioned at the left side.
    - **At Right Sensor**: Detects whether the forklift is positioned at the right side.
    - **At Middle Sensor**: Detects whether the forklift is positioned at the middle.
- **Stacker Crane Movement Sensors**:
    - **Moving X Sensor**: Detects movement of the stacker crane in the X direction (horizontal movement).
    - **Moving Z Sensor**: Detects movement of the stacker crane in the Z direction (vertical movement).

Actuators

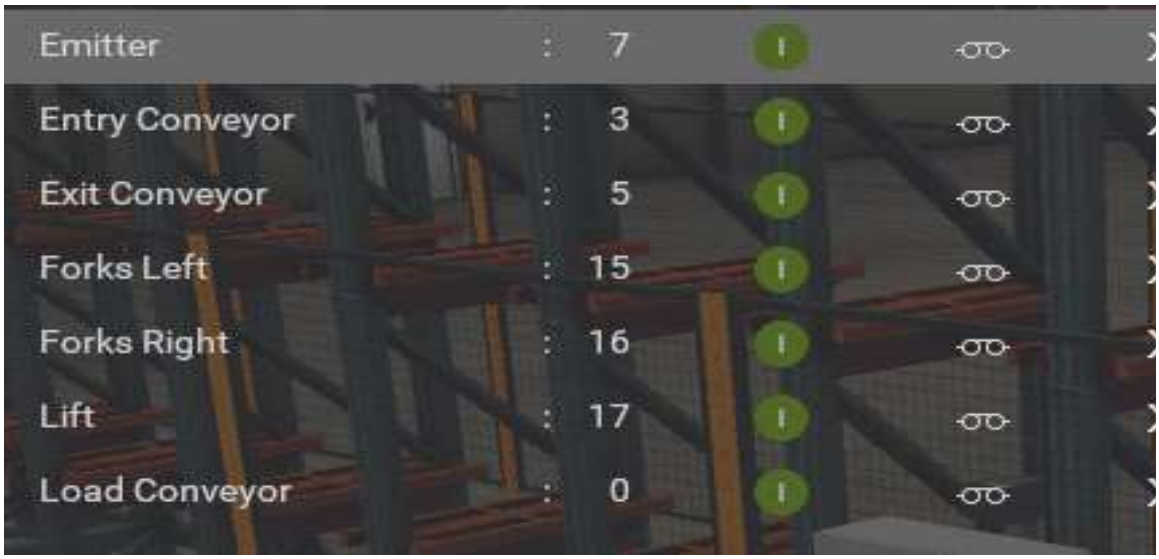


Figure 2. Automated Warehouse's Actuators

- **Entry Conveyor**: Conveyor at the entry point that transports objects into the system.
- **Forks Left**: Position of the forklift at the left side of the system for loading/unloading.
- **Forks Right**: Position of the forklift at the right side of the system for loading/unloading.
- **Lift:** The lifting mechanism that raises or lowers objects to the appropriate rack height.
- **Load Conveyor**: Conveyor at the loading point used to transfer objects from the conveyor to the rack or stacker crane.

Hardware components

PLC (Programmable Logic Controller)

Model: SIMATIC S7-1500

I/O Configuration: 1) DI 16x24VDC BA 2) DQ 16x24VDC/0.5A BA

Figure 3. Siemens SIMATIC S7-1500

Sensors

Retroreflective Sensor and Reflector

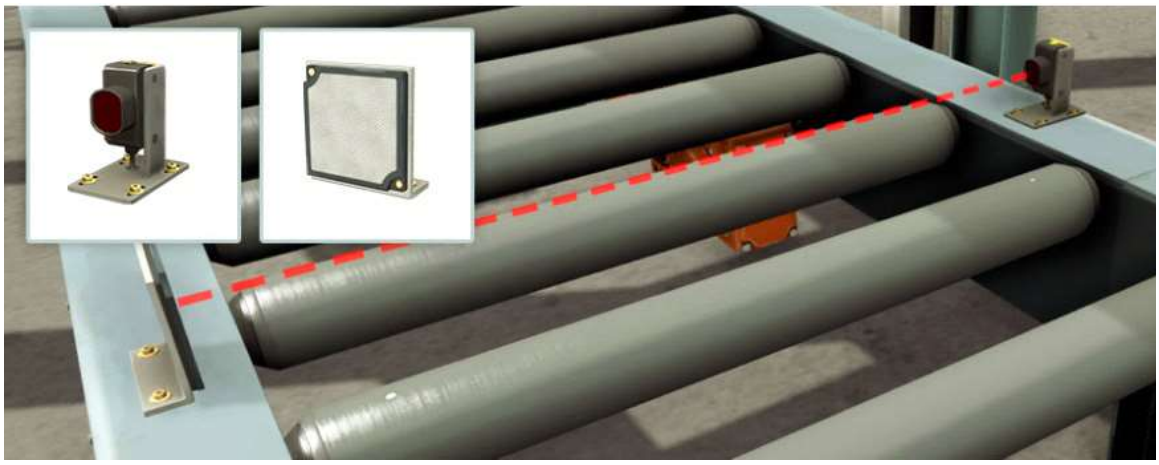


Figure 4. Factory IO Retroreflective Sensor and Reflector

To work properly, it must be aligned with the reflector. It is equipped with two LEDs which indicate the correct alignment (green) and detection status (yellow).

- Green LED: aligned with the reflector
- Yellow LED: light beam not interrupted
- Detectable materials: solids
- Sensing range: 0 - 6 m

Actuators

Stacker Crane and Rack

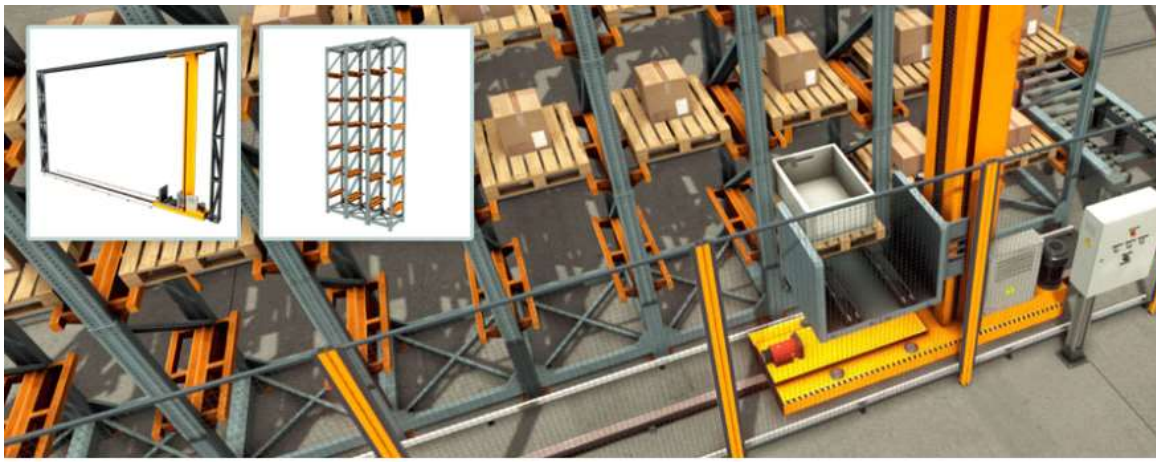


Figure 5. Factory IO Stacker Crane and Rack

Rail-mounted stacker crane used to stock heavy cargo. Includes a cart, a vertical platform, and two forks which may slide to both sides.

Two laser rangefinders, placed on the cart and the platform, measure the horizontal and vertical position of the platform. Racks are upright steel frames connected by horizontal steel beams with the purpose to store loads. The available rack is a single-deep rack type, also known as selective rack, which only allows loads to be stored one pallet deep. Loads can be stored from both sides of the rack.

Each rack must be aligned with one of the rail ends, making the stacker crane stop at the correct position. The stacker crane can be controlled by Digital, Numerical and Analog values, according to the selected configuration.

Numerical: the target cell can be defined by an integer value between 1 and 54. If this value is set to zero, the stacker crane stops at the current position. However, if it is higher than 54, it will move to the rest position (55).

Conveyors

Heavy-duty roller conveyor: can be controlled by digital and analog values according to the selected configuration.

Figure 6. Factory IO Heavy-duty roller conveyor

Loading conveyor: Heavy-duty conveyor, mostly used to load/unload cargo onto a Stacker Crane. Can be controlled by digital or analog values.

Program Components

Inputs and Outputs

This Factory I/O configuration includes inputs (I0.0–I1.6) for sensors and control signals like position detection, movement status, and system controls (e.g., Start, Stop, Emergency Stop). Outputs (Q0.0–Q1.1) manage actuators such as conveyors, forks, lift mechanisms, and indicator lights. QW30 represents an integer value for the target position.

Figure 8. Factory IO input output configuration for Siemens S7-PLCSIM

Code Structure

The program logic for this project can be found under the Program Blocks tab in TIA Portal. To ensure clarity and maintainability, the code is organized into five program blocks

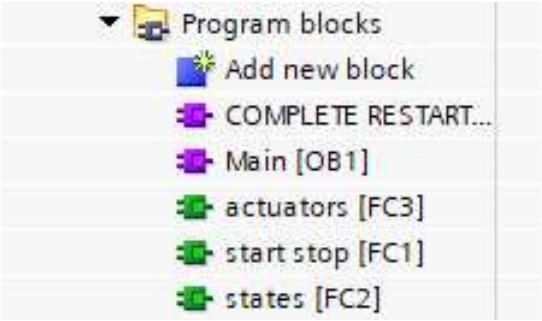


Figure 9. TIA Portal Program Blocks

Program blocks

- **Main (OB1):**
  - Serves as the primary organization block for the program.

- Responsible for initiating the following three key functions:
    - Actuators (FC3): Manages the operation of warehouse actuators.
    - Start/Stop (FC1): Handles the warehouse start and stop functionality.
    - States (FC2): Manages state transitions within the system.

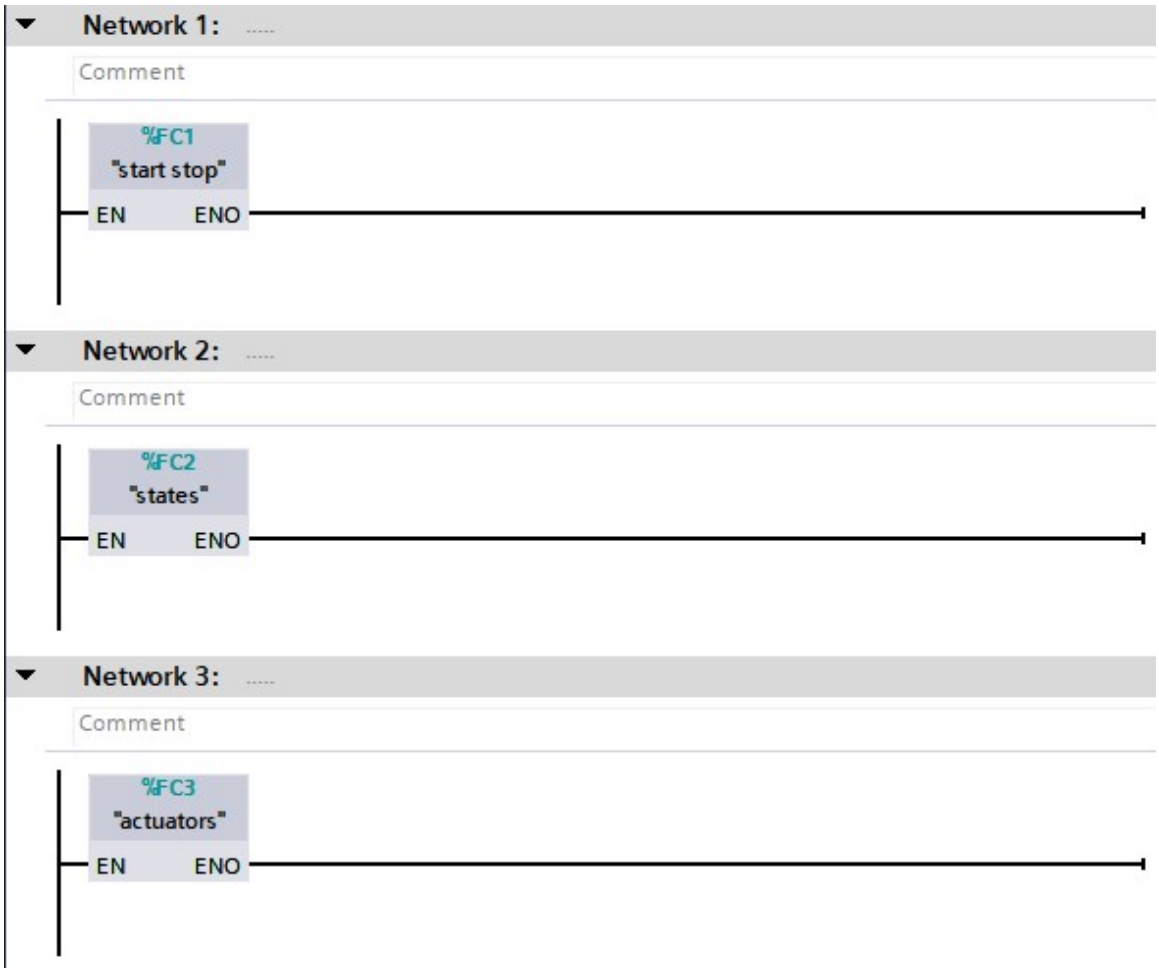


Figure 10. Main (OB1) Code

Complete Restart:

- This block is triggered during the program's initialization.
- Ensures the system starts at **State 0**, preparing it for subsequent operations.

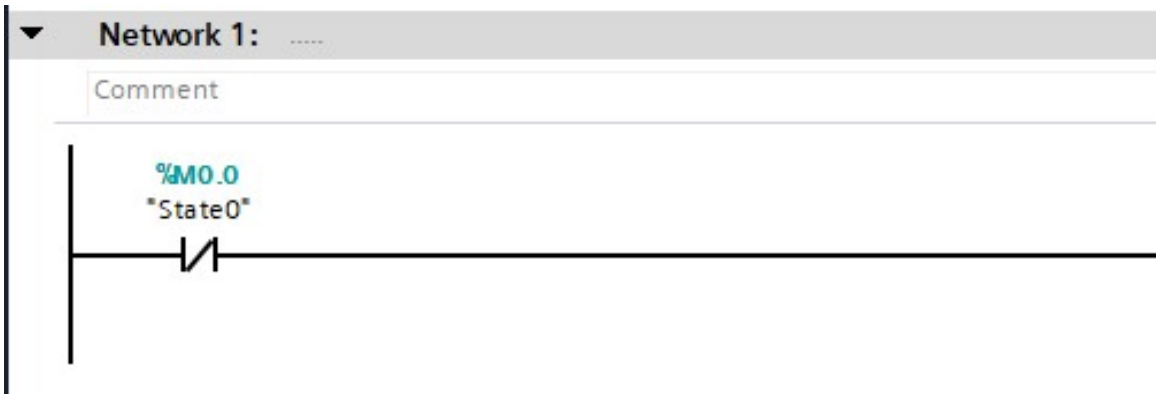


Figure 11. Complete Restart Code

- **Start/Stop (FC1):**
    - Governs the initiation of the warehouse operation when the **Start** button is pressed.
    - Activates the **Start Light** to indicate the system is running.
    - Automatically switches the **Stop Light** on when the system is inactive.

The **Start/Stop function** ensures smooth operation of the system by toggling between idle and active states. The functionality is implemented in two networks as shown:

Network 1 (Start Logic):

- Condition: When the Start button (%I1.1) is pressed, and the Stop button (%I1.3) is not pressed:
    - The Start Light (%Q0.7) is activated, signaling the system is operational.
    - A latching mechanism is used: the Start Light is placed in parallel with the Start Button. This keeps the Start Light on even after the Start Button is released.
- Interruption: If the Stop button is pressed, the Start Light turns off, halting the system.

Network 2 (Stop Logic):

- Condition: When the Start Light (%Q0.7) is off:
    - The Stop Light (%Q1.1) is automatically activated, indicating the system is in an idle state.

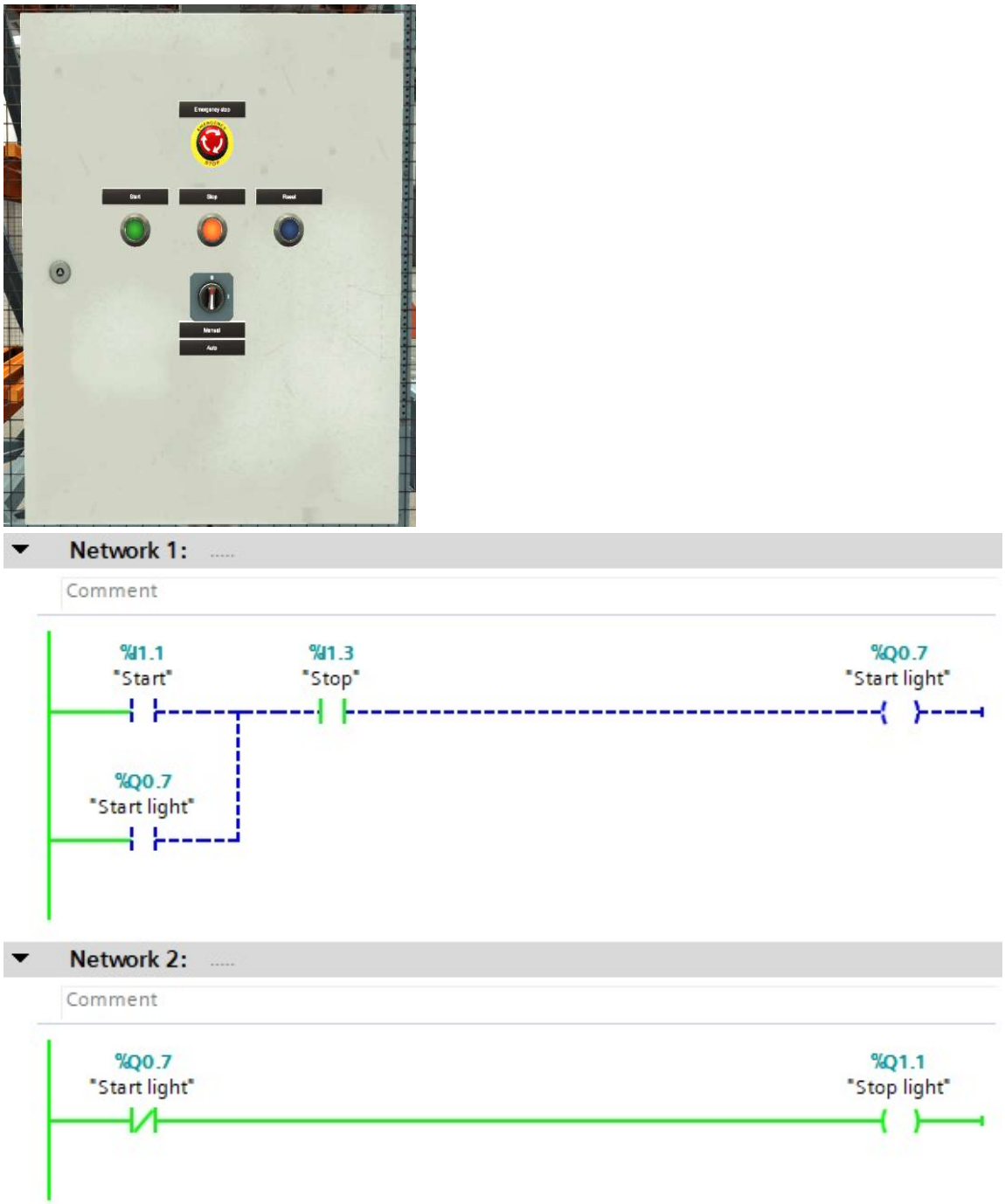


Figure 13. Start-Stop (FC1) Idle

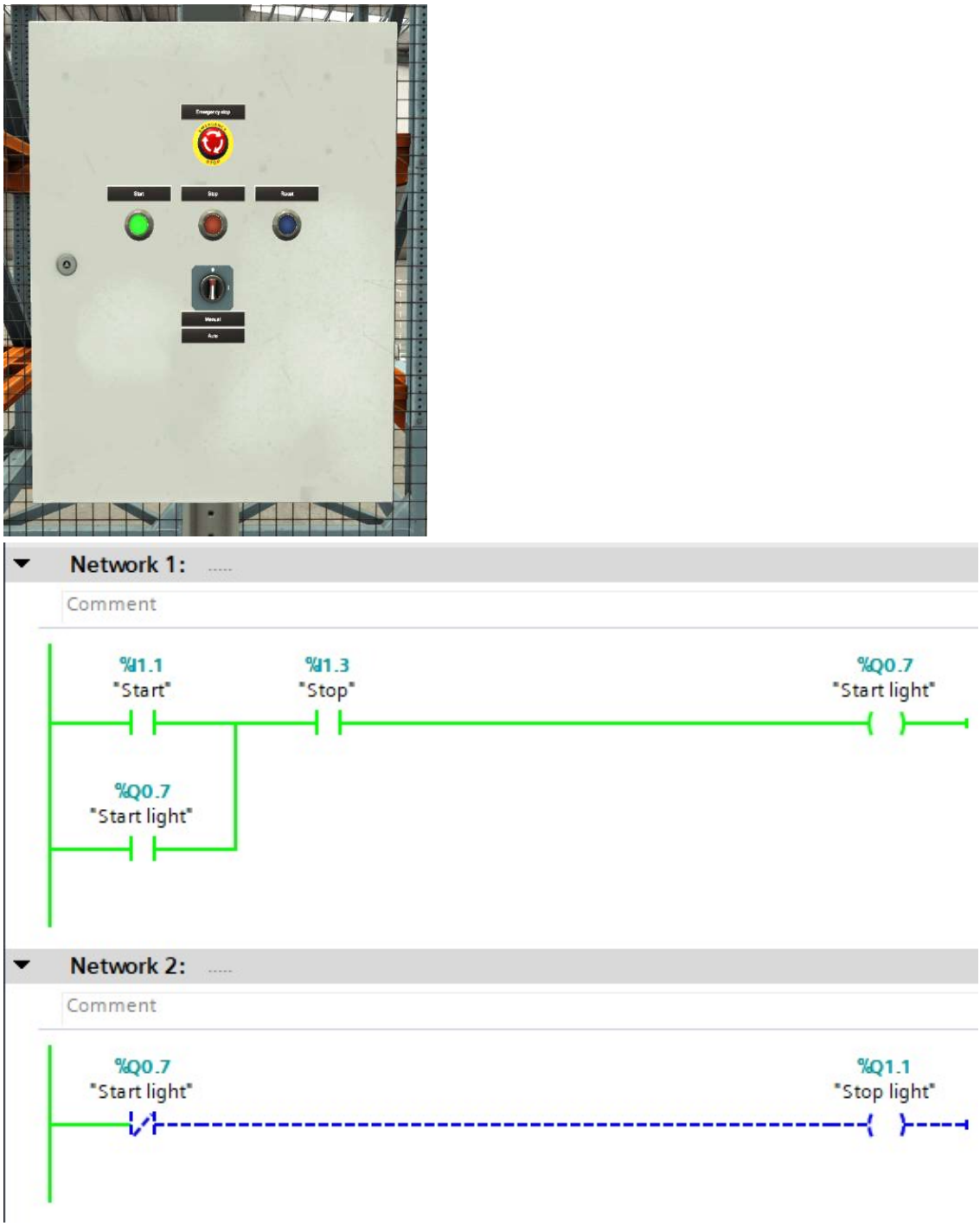


Figure 15.Start-Stop (FC1) Start button is pressed

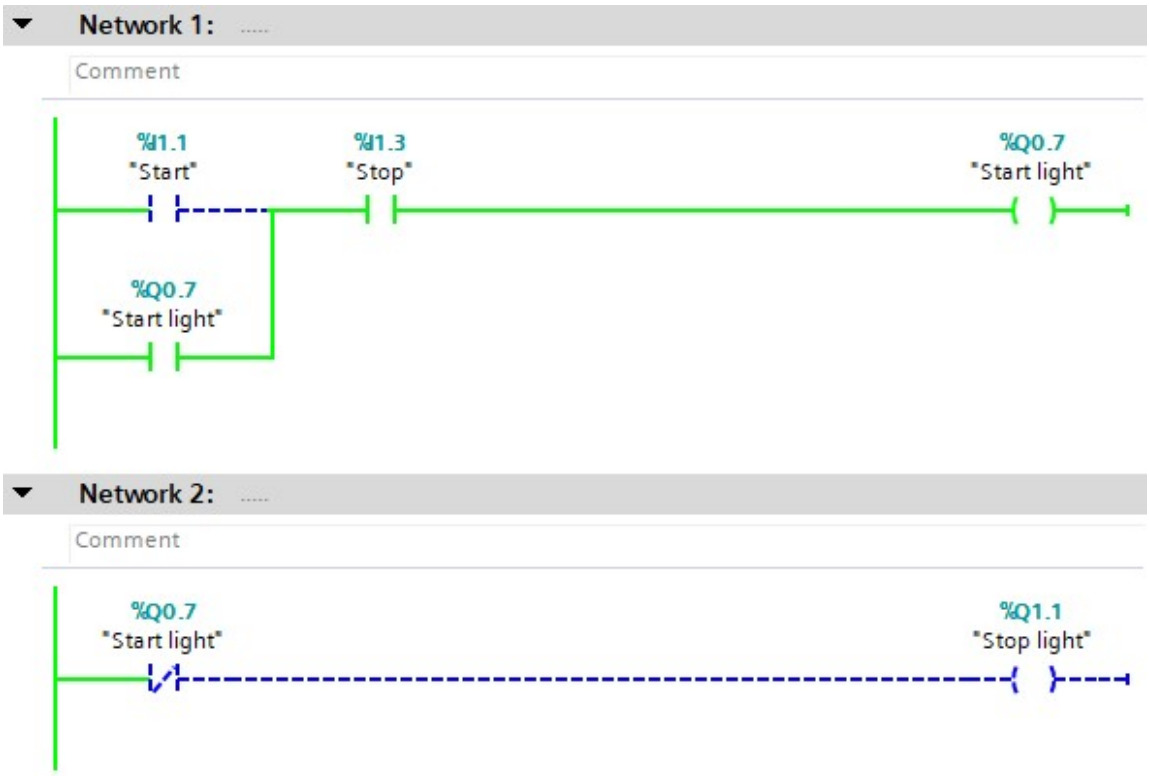


Figure 16. Start-Stop (FC1) Start button is released

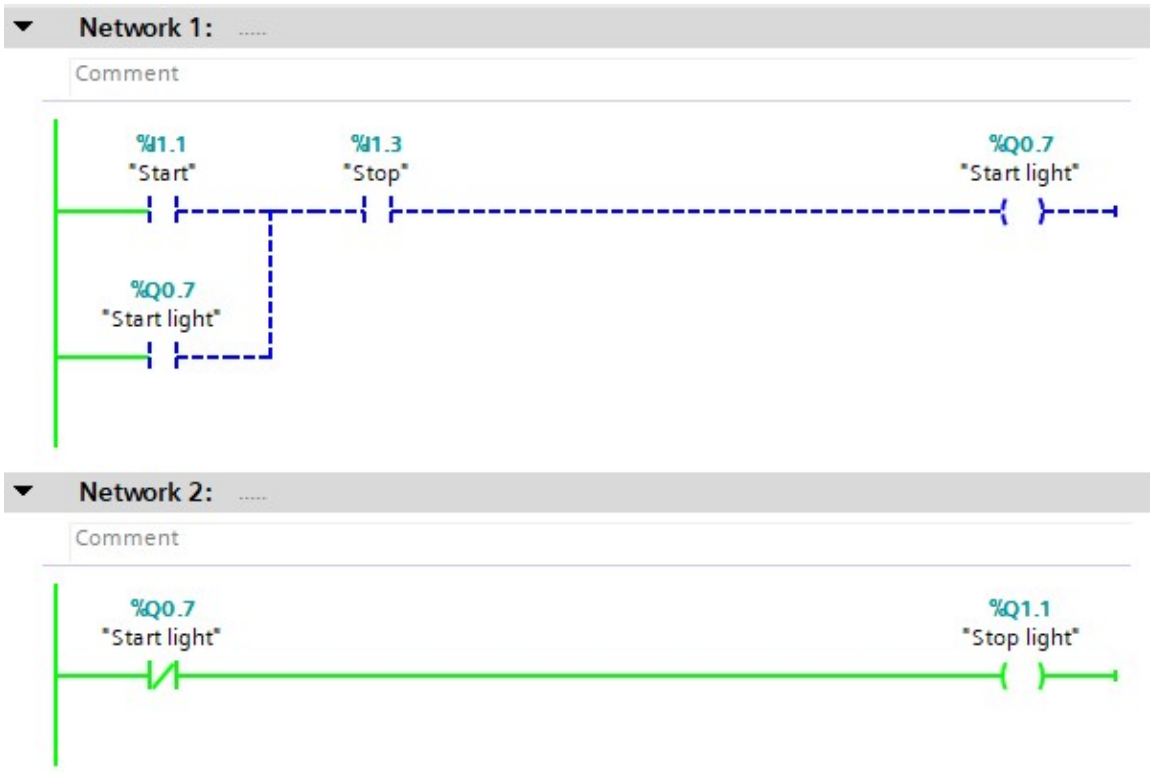
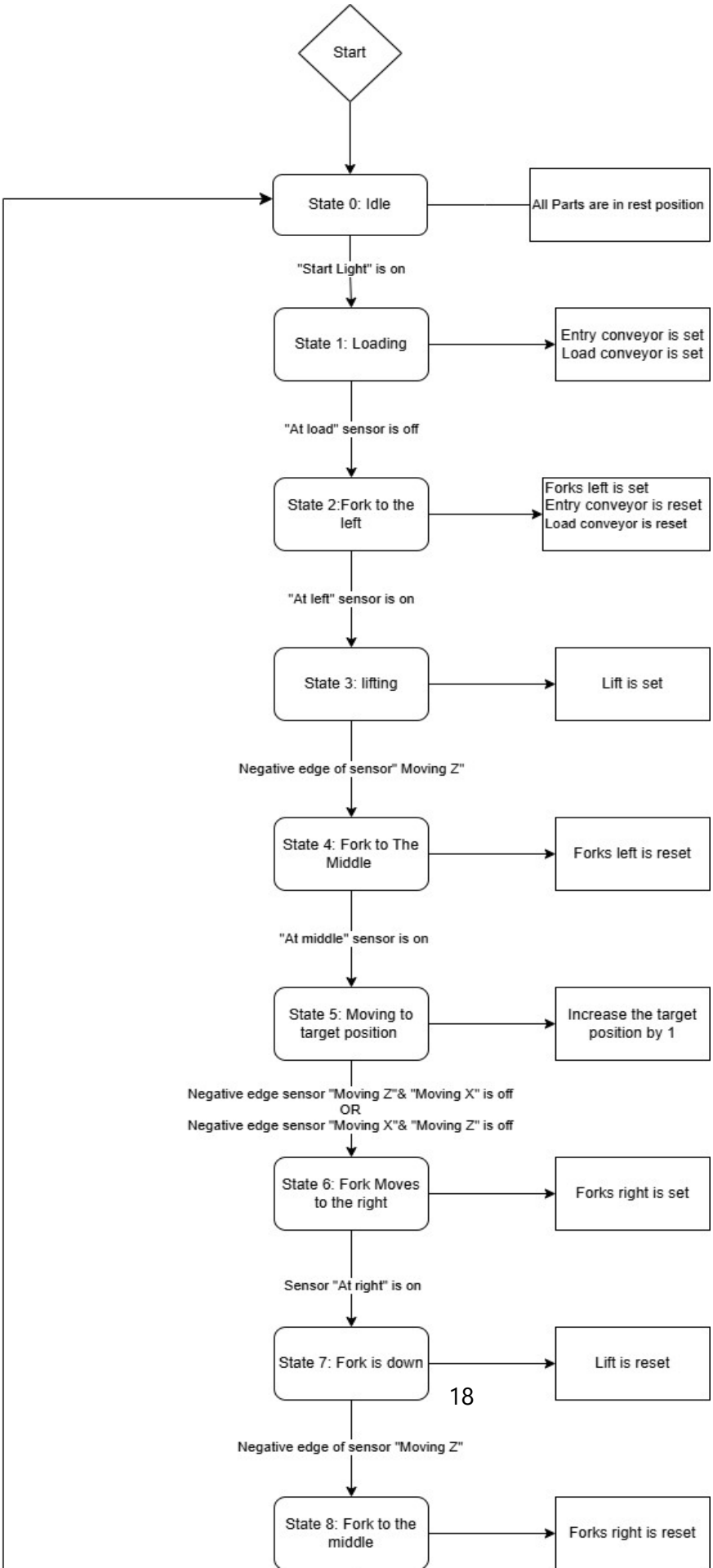


Figure 17. Start-Stop (FC1) Stop button in pressed

States (FC2) and Actuators (FC3):

- These two blocks collaboratively operate the warehouse in the correct sequence.
- States (FC2): Maintains the logical progression of warehouse operations by tracking system states.
- Actuators (FC3): Controls the physical devices in accordance with the current state.


Start
State 0: Idle
All Parts are in rest position
"Start Light" is on
State 1: Loading
Entry conveyor is set
Load conveyor is set
"At load" sensor is off
State 2:Fork to the left
Forks left is set
Entry conveyor is reset
Load conveyor is reset
"At left" sensor is on
State 3: lifting
Lift is set
Negative edge of sensor" Moving Z"
State 4: Fork to The Middle
Forks left is reset
"At middle" sensor is on
State 5: Moving to target position
Increase the target position by 1
Negative edge sensor "Moving Z"& "Moving X" is off
OR
Negative edge sensor "Moving X"& "Moving Z" is off
State 6: Fork Moves to the right
Forks right is set
Sensor "At right" is on
State 7: Fork is down
Lift is reset
Negative edge of sensor "Moving Z"
State 8: Fork to the middle
Forks right is reset

System Operation Overview
The system operates using a state-based process, as outlined in the flowchart. Each state performs specific actions, and transitions to the next state are determined
by conditions represented on the arrows. Below is a detailed explanation of each state:

State 0: Idle

- **Condition to Enter State 0:**
    - The system transitions back to State 0 when the negative edge of the "Moving Z" sensor or the Stop Button is pressed.
- **Actions:**
    - All system components are reset to their initial rest positions.
    - The Start Light is turned on to indicate readiness.

Figure 18. Automated Warehouse Idle state

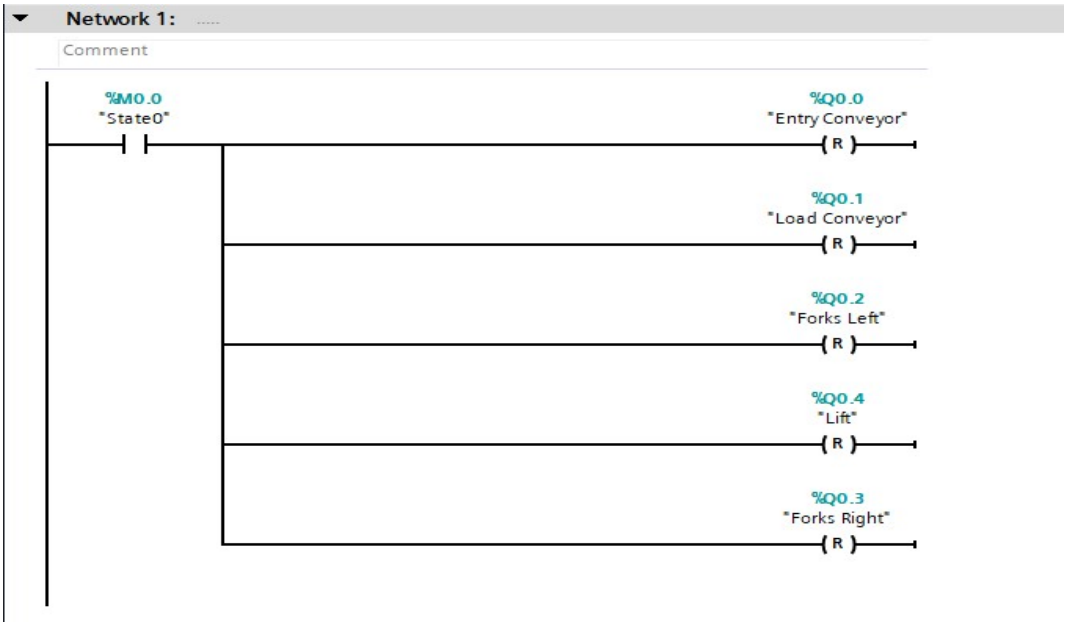


Figure 19. Actuators (FC3) Network 1

State 1: Loading

- **Condition to Enter State 1:**
  - The system enters State 1 when the Start Light is turned on from State 0.
- **Actions:**
  - The Entry Conveyor is activated to transport the part into position.
  - The Load Conveyor prepares the part for loading.

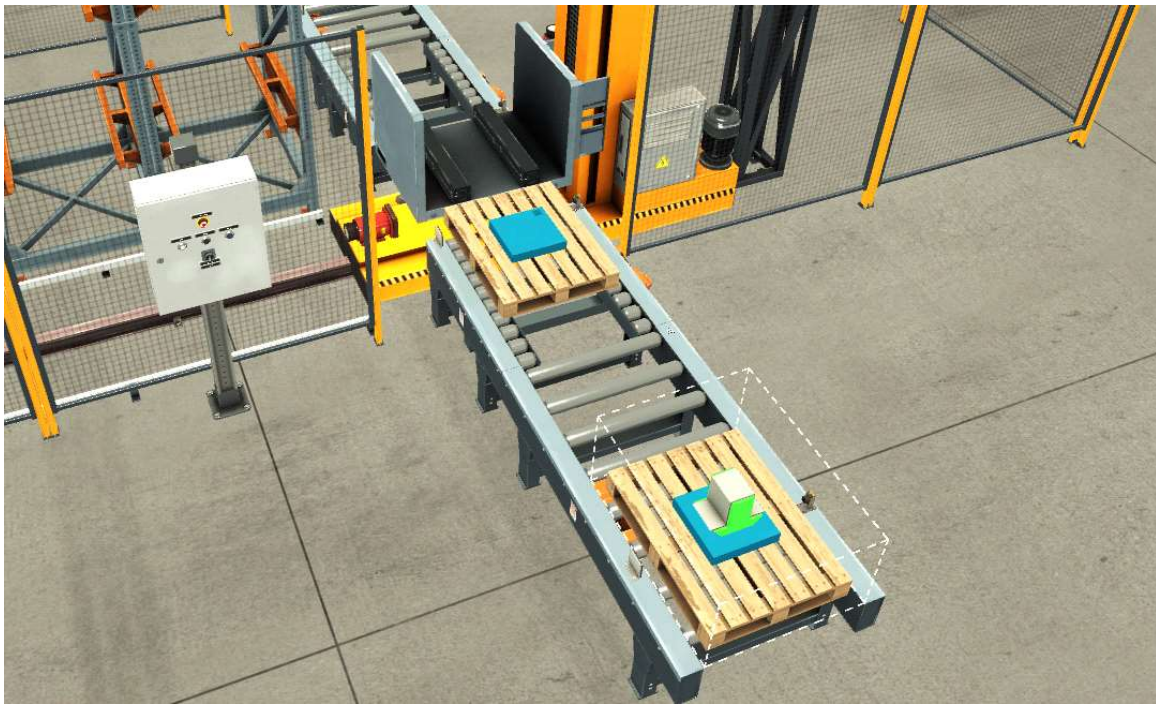


Figure 20. Automated Warehouse Loading state

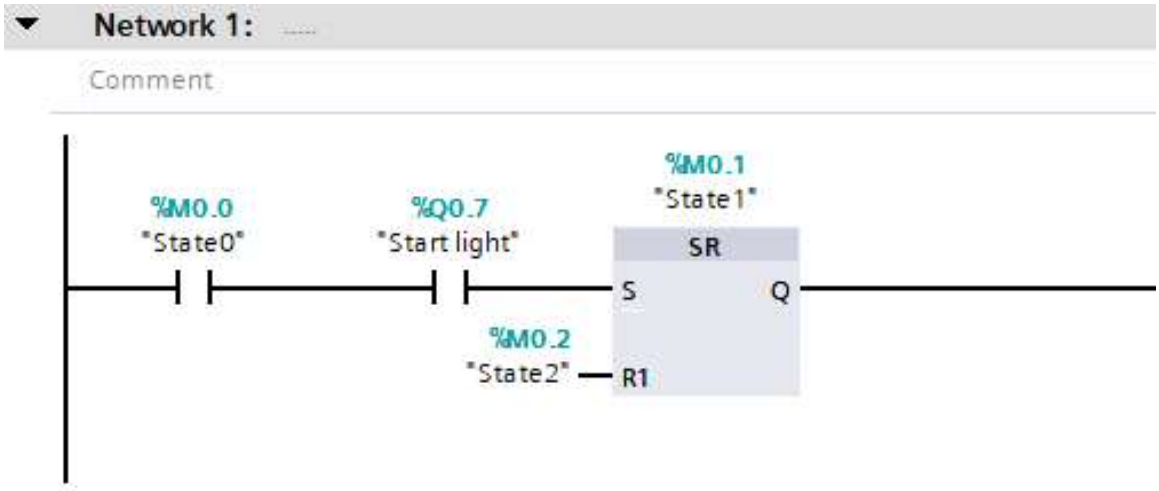


Figure 21. States (FC2) Network 1

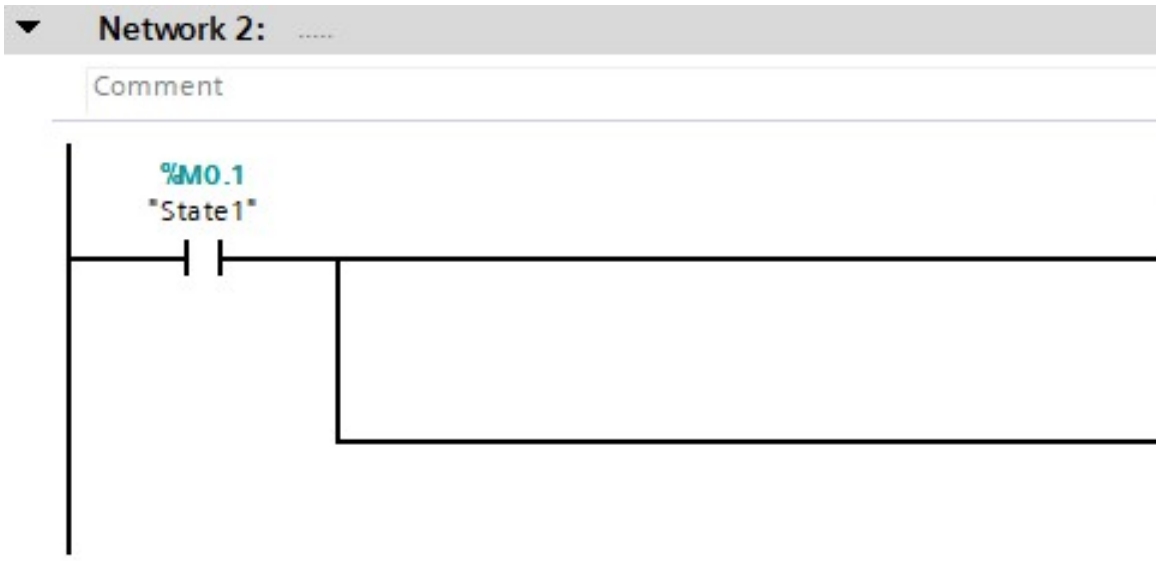


Figure 22. Actuators (FC3) Network 2

State 2: Fork to the Left

- **Condition to Enter State 2:**
  - The system enters State 2 when the "At Load" sensor is off in State 1.
- **Actions:**
  - The Fork Left actuator is activated to move the fork to the leftmost position.
  - The Entry Conveyor and Load Conveyor are reset to their initial states.

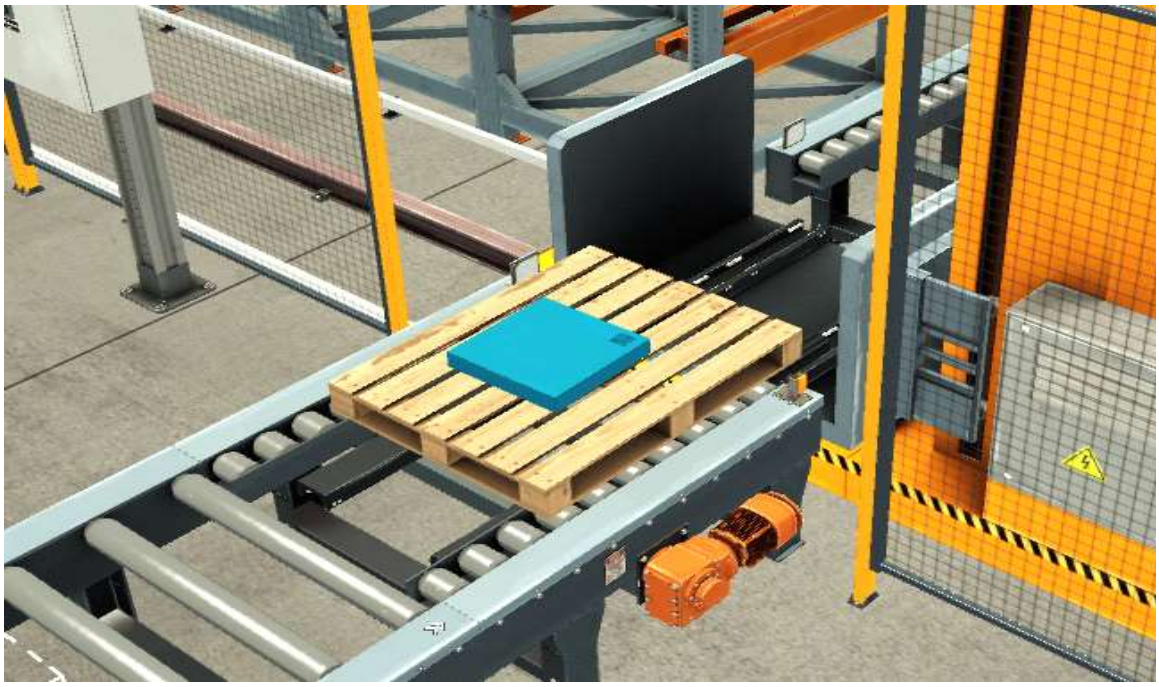


Figure 23. Fork moved to the left

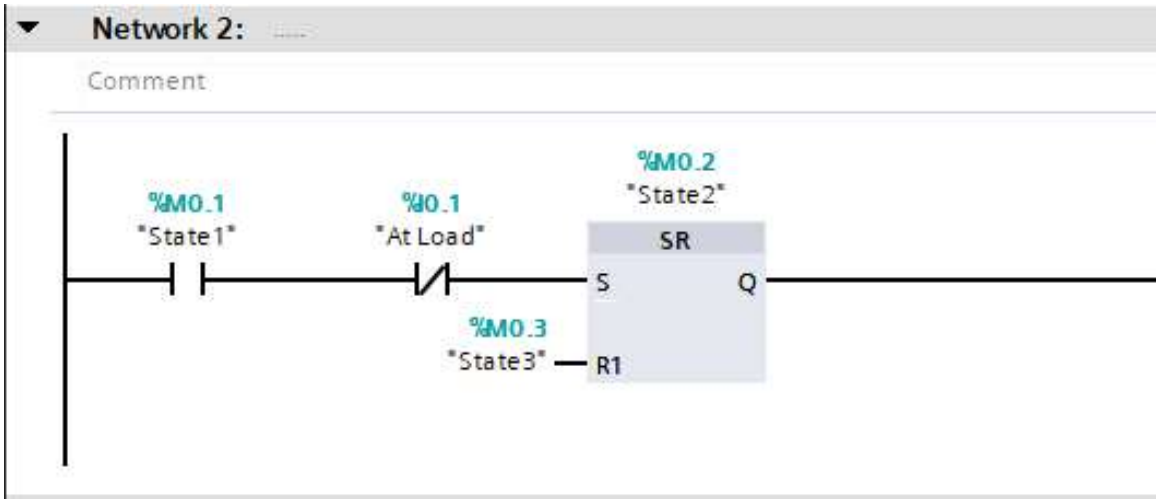


Figure 24. States (FC2) Network 2

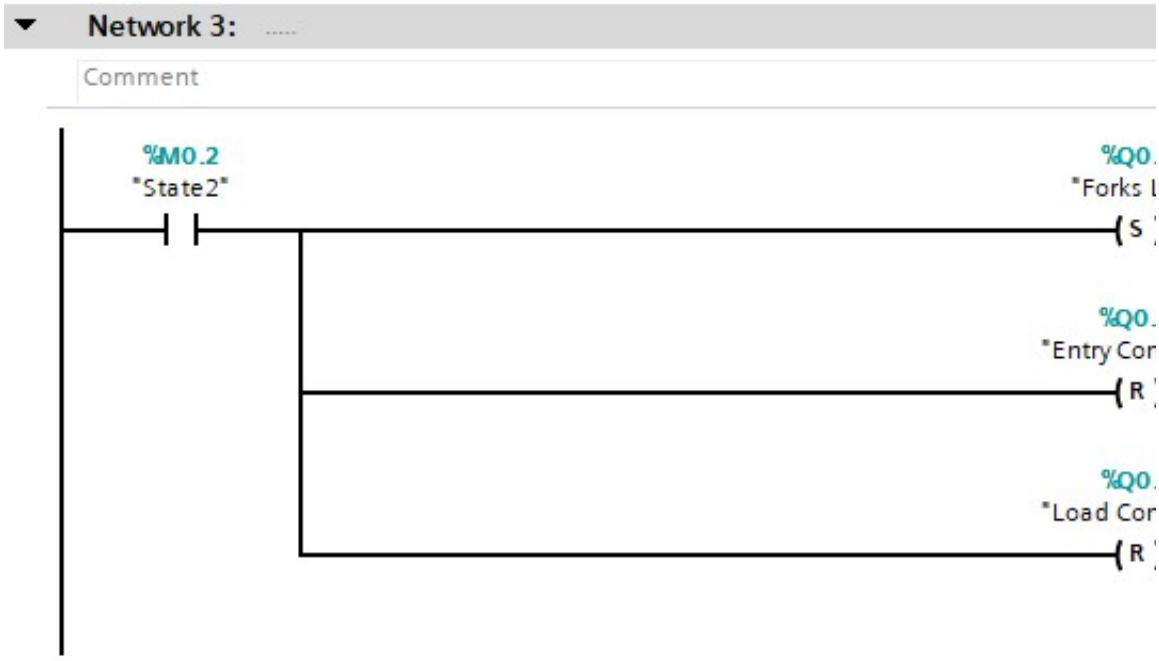


Figure 25. Actuators (FC3) Network 3

State 3: Lifting

- **Condition to Enter State 3:**
  - The system enters State 3 when the "At Left" sensor is on in State 2.
- **Actions:**
  - The Lift actuator is activated to raise the fork to the required height.

Figure 26. Fork in lift position

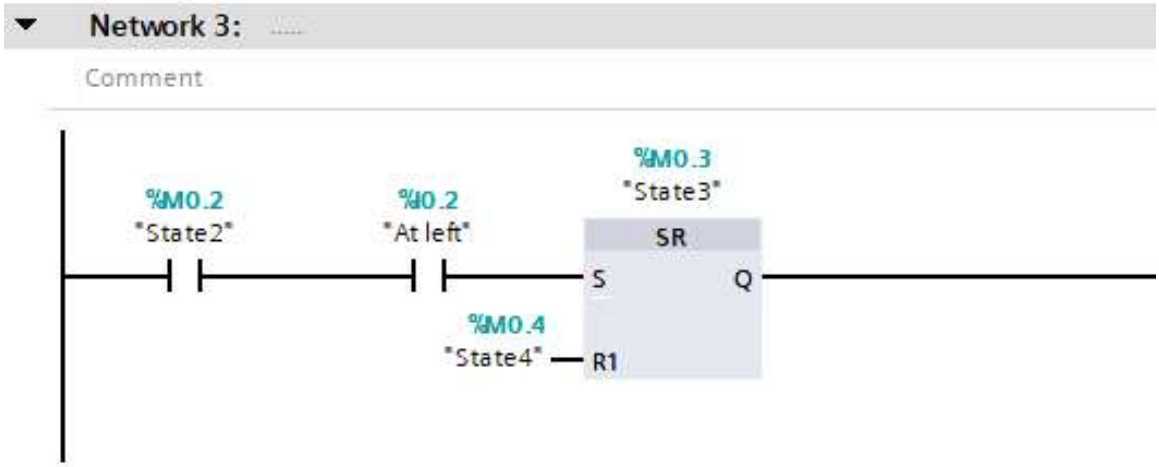


Figure 27. States (FC2) Network 3

Figure 28. Actuators (FC3) Network 4

State 4: Fork to the Middle

- **Condition to Enter State 4:**
  - The system enters State 4 when the negative edge of the "Moving Z" sensor is detected in State 3.
- **Actions:**
  - The Fork Left actuator is reset to move the fork back to the middle position.

Figure 29. Fork at middle

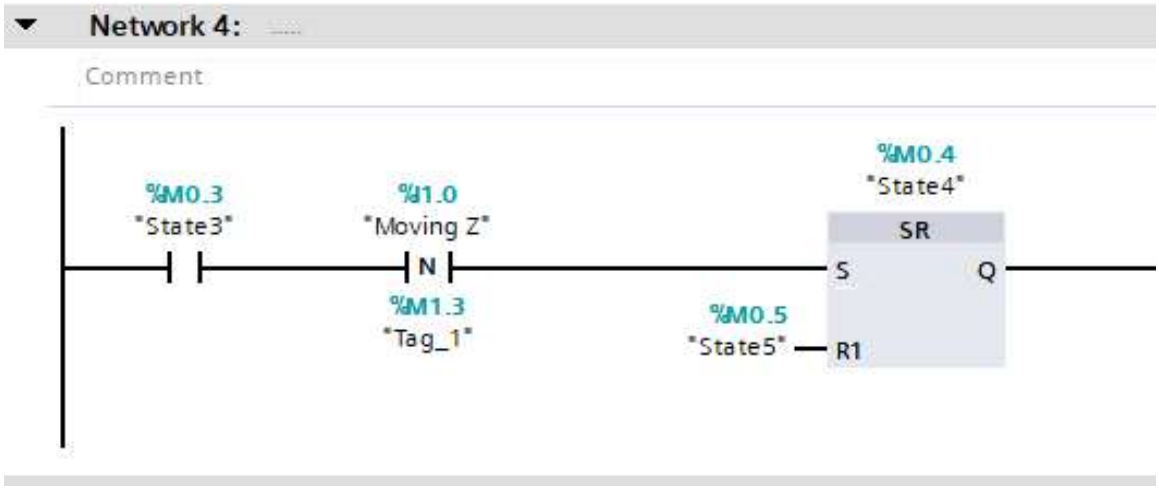


Figure 30. States (FC2) Network 4

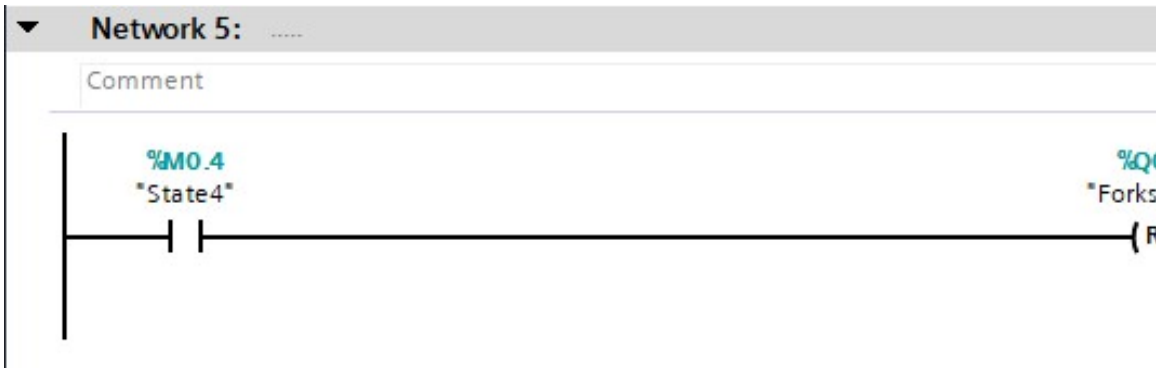


Figure 31. Actuators (FC3) Network 5

State 5: Moving to Target Position

- **Condition to Enter State 5:**
  - The system enters State 5 when the "At Middle" sensor is on in State 4.
- **Actions:**
  - The system increments the Target Position by 1, guiding the fork to the next designated delivery location.
  - To ensure proper incrementation, an additional variable (T) is introduced:
    - **Why Variable T is Used:**
      - At the end of the process (in State 9), the Target Position is reset to 55 to return the fork to its initial loading position.
      - If the Target Position itself were used for incrementation, it would start the second cycle with the value 55, instead of the correct next position (2).
      - Therefore, Variable T is used to incrementally track the cycles independently.
    - **Positive Trigger Logic:**
      - A Positive Trigger Block is utilized to increment Variable T only once per cycle. This ensures the addition operation happens exactly once during each pass through the state.

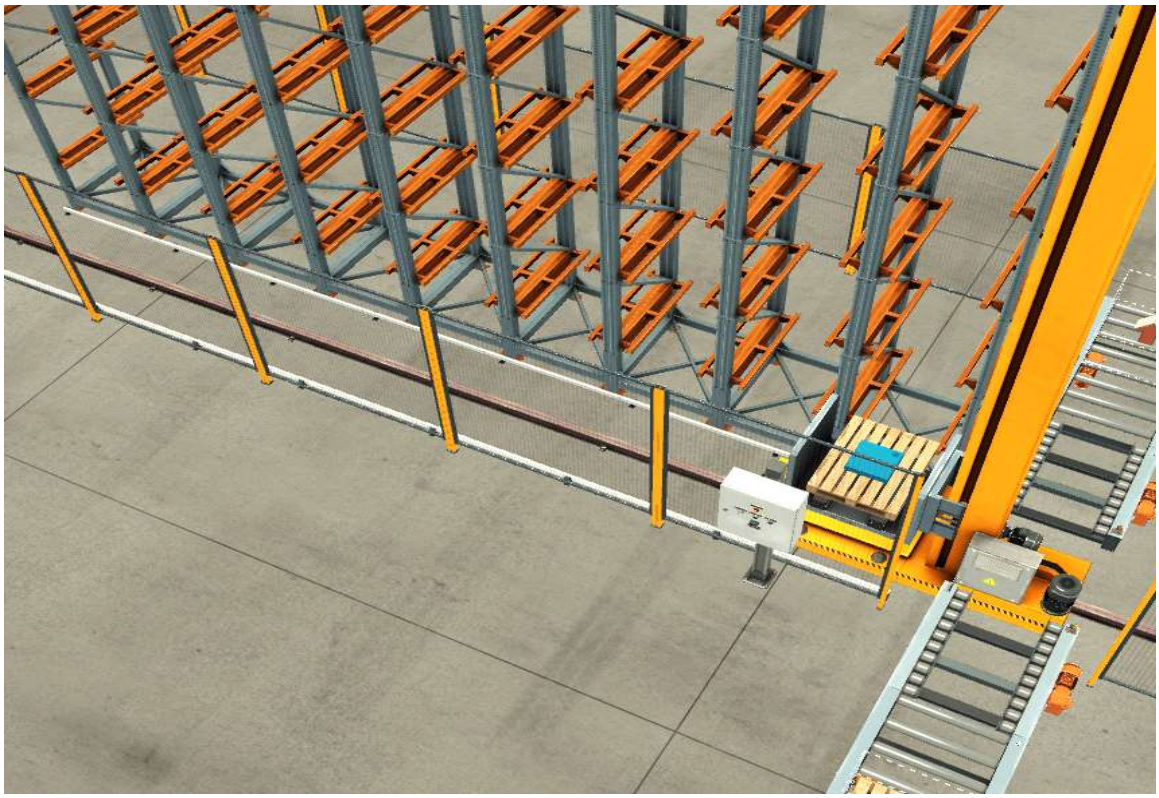


Figure 32. Fork moved to target postion

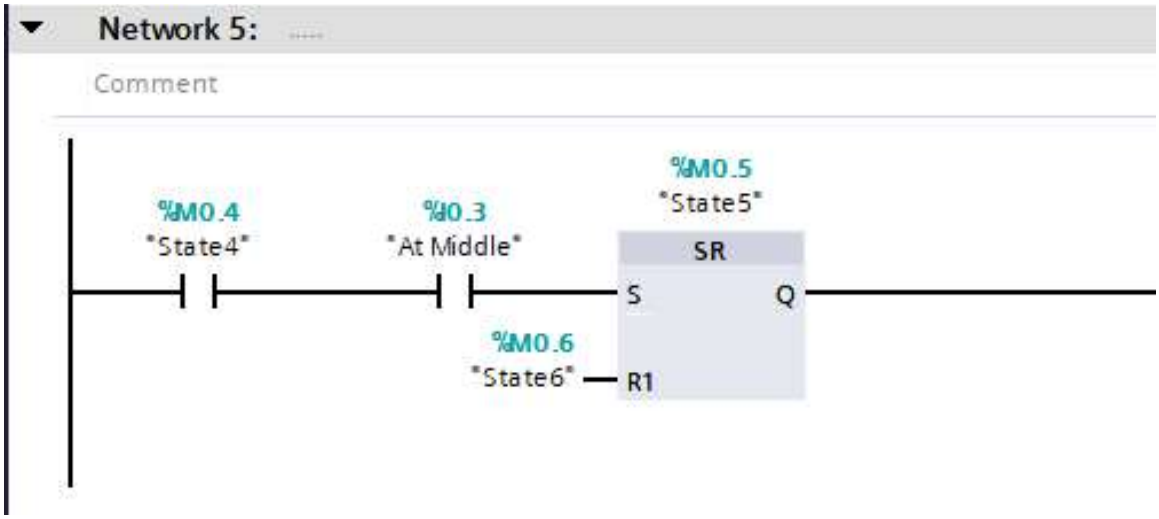


Figure 33. States (FC2) Network 5

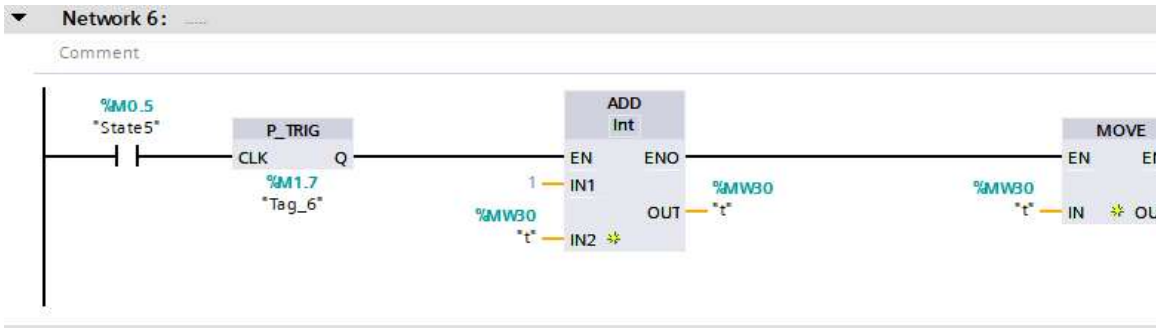


Figure 34. Actuators (FC3) Network 6

State 6: Fork Moves to the Right

- **Condition to Enter State 6:**
    - The system enters State 6 when the negative edge of "Moving Z" and "Moving X" sensors is detected in State 5.
- **Actions:**
    - The Fork Right actuator is activated to align the fork for part release.

Figure 35. Fork in Right postion

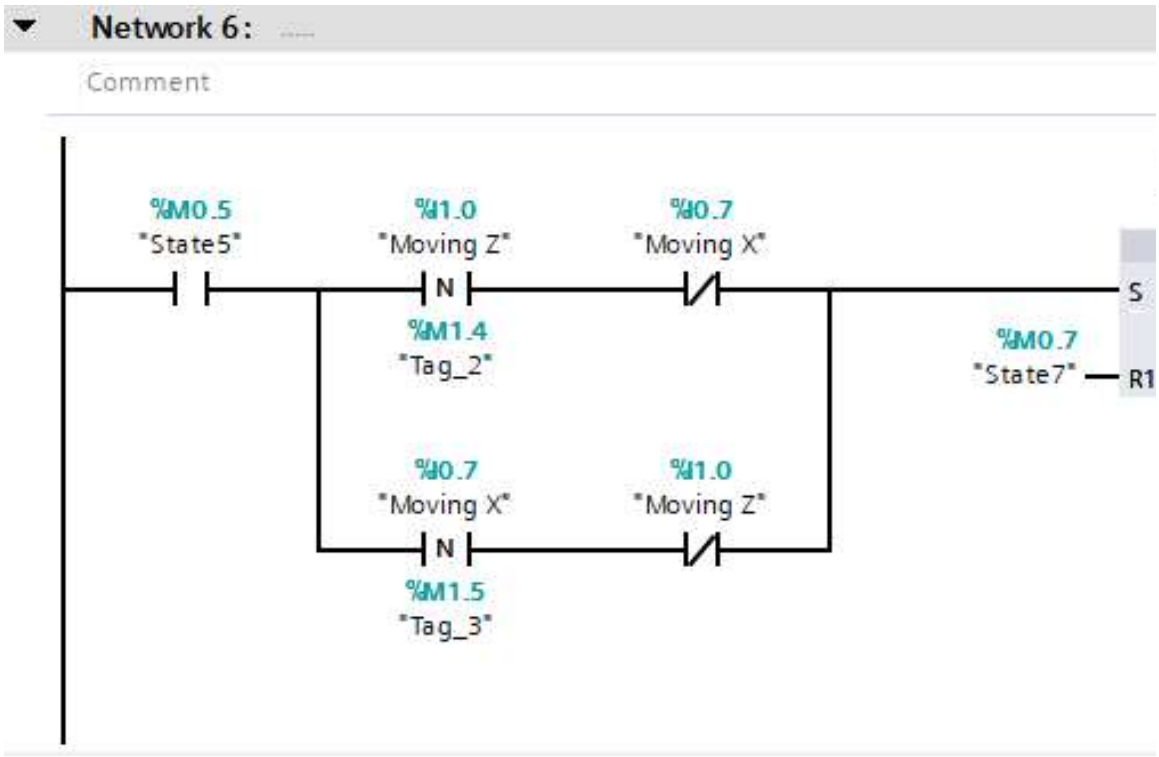


Figure 36. States (FC2) Network 6

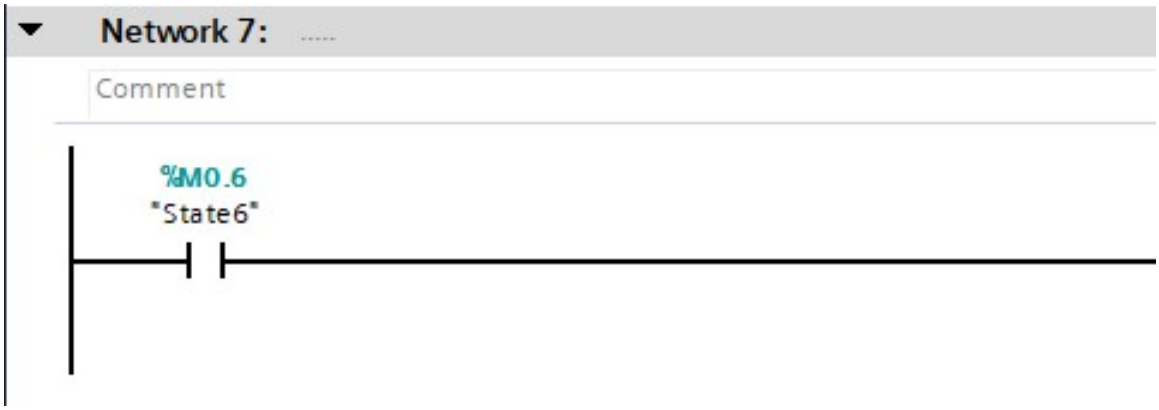


Figure 37. Actuators (FC3) Network 7

State 7: Fork Down

- **Condition to Enter State 7:**
  - The system enters State 7 when the "At Right" sensor is on in State 6.
- **Actions:**
  - The **Lift** actuator is reset to lower the fork, releasing the part at the target

position.

Figure 38. Fork is in lower postion

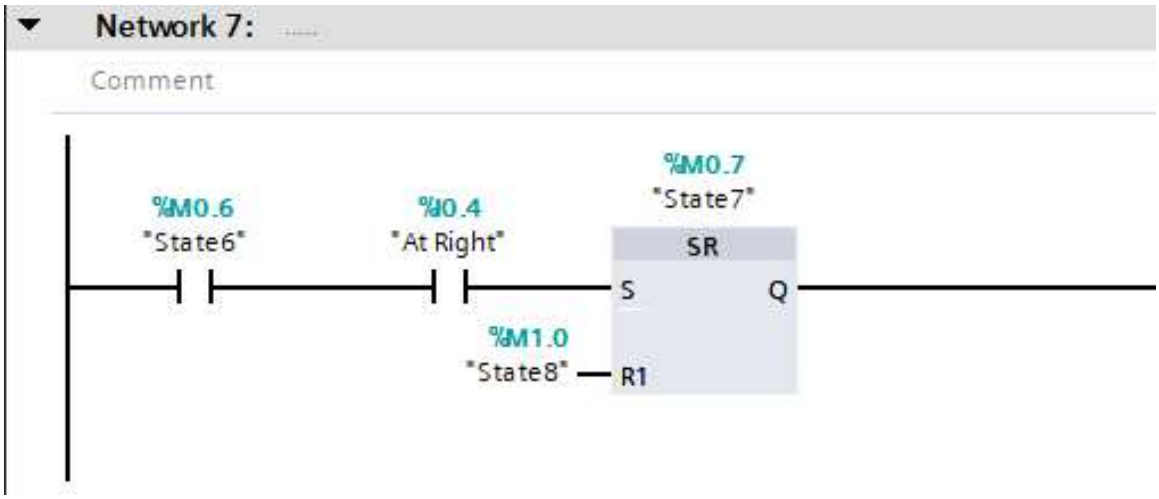


Figure 39. States (FC2) Network 7

Figure 40. Actuators (FC3) Network 8

State 8: Fork to the Middle

- **Condition to Enter State 8:**
  - The system enters **State 8** when the **negative edge of the "Moving Z" sensor** is detected in **State 7**.
- **Actions:**
  - The **Fork Right** actuator is reset to return the fork to the center position.

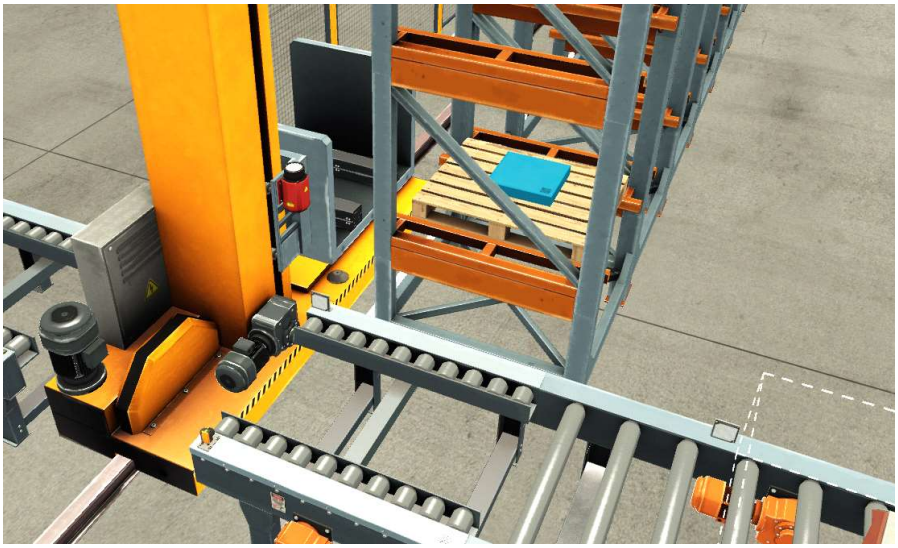


Figure 41. Fork at middle position

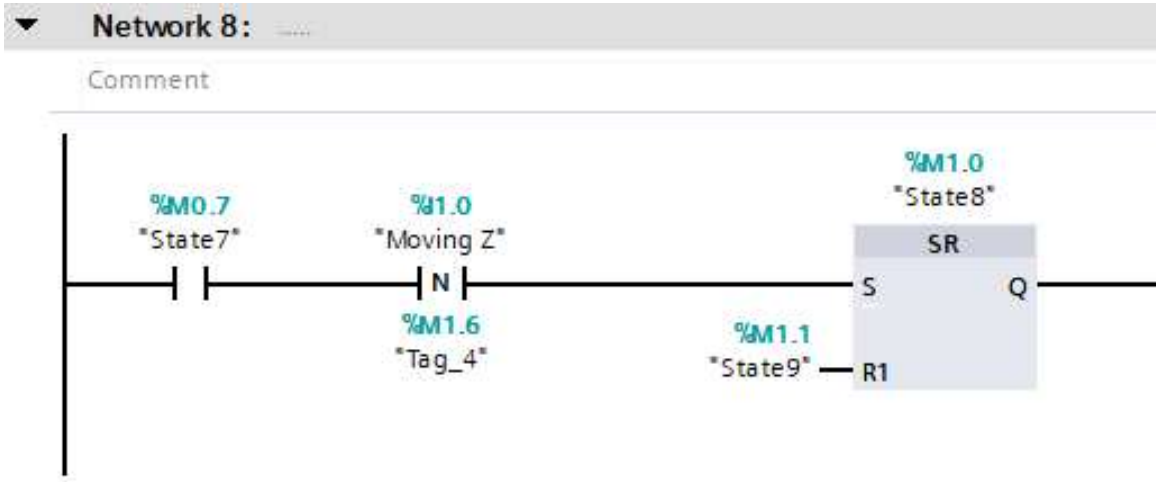


Figure 42. States (FC2) Network 8

Figure 43. Actuators (FC3) Network 9

State 9: Return to Loading Position

- **Condition to Enter State 9:**
  - The system enters State 9 when the "At Middle" sensor is on in State 8.
- **Actions:**
  - The Target Position is reset to 55.
  - All system components return to their initial positions.

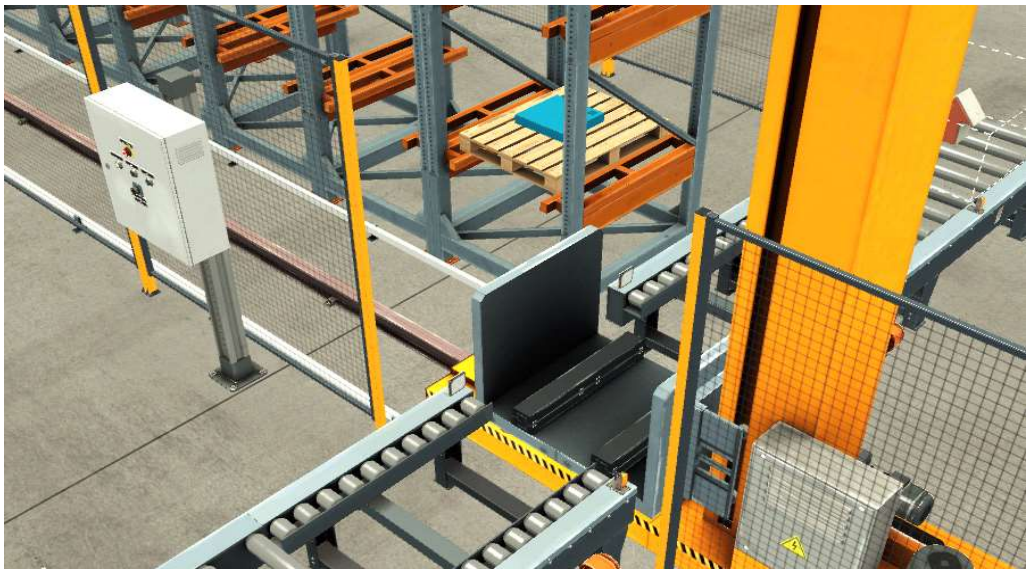


Figure 44. Fork returned to initial position

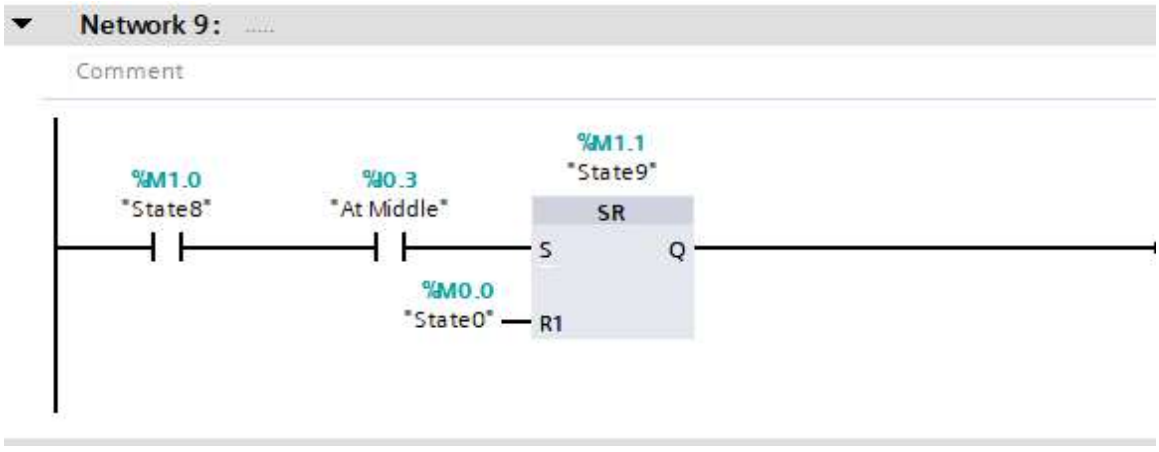


Figure 45. States (FC2) Network 9

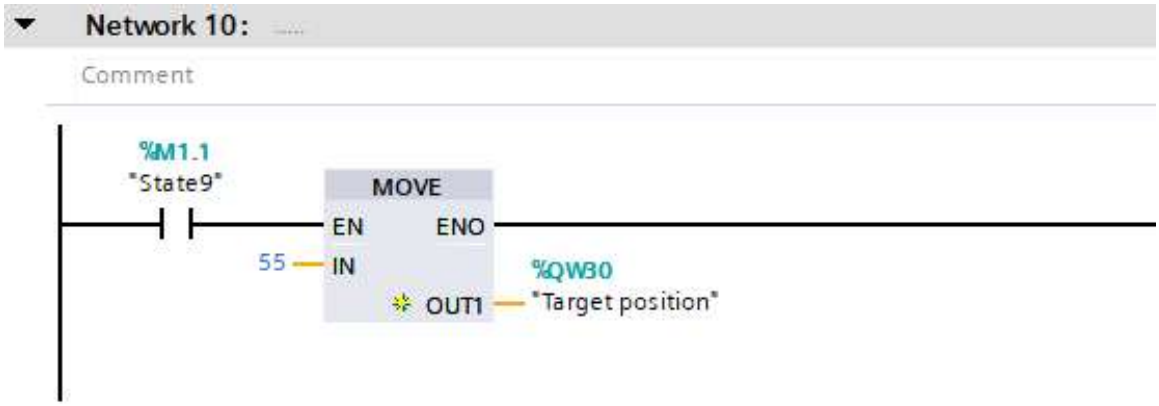


Figure 46. Actuators (FC3) Network 10