TECHNOLOGY

Siemens S 7 1200 Plc Programming Examples

PUBLISHED
November 25, 2026

REFERENCE
#1795581503036697

Mastering Siemens S7-1200 PLC Programming Through Real-World Examples

Programmable Logic Controllers (PLCs) form the backbone of modern industrial automation, and the Siemens S7-1200 remains one of the most widely adopted controllers for small to medium-sized applications. Whether you are a student entering the world of automation or an experienced engineer expanding your skillset, understanding how to build practical logic with this powerful device is essential. The best way to gain proficiency is by working through concrete, real-world **Siemens S 7 1200 PLC programming examples**that illustrate fundamental concepts and common automation tasks.

This article walks you through several hands-on programming examples built using TIA Portal (Totally Integrated Automation Portal), the standard engineering environment for the S7-1200. Each example is designed to teach you a specific technique – from simple logic gates to more advanced function blocks and data handling – while using natural, readable logic structures. By the end, you will have a solid toolkit of practical **PLC programming examples for Siemens S7-1200** that you can adapt to your own projects.

Why Work with Practical Siemens S7-1200 PLC Programming Examples?

Theoretical knowledge is valuable, but automation engineering demands hands-on competence. When you study **Siemens S 7 1200 PLC programming examples**, you bridge the gap between abstract instructions and the physical behavior of machines. These examples help you understand scan cycles, memory addressing, timer behavior, and the interaction between hardware inputs and software outputs. They also reveal the debugging mindset needed when a system does not behave as expected.

Moreover, the S7-1200 offers multiple programming languages under the IEC 61131-3 standard – Ladder Logic (LAD), Function Block Diagram (FBD), and Structured Control Language (SCL). By exploring diverse examples, you learn which language suits which scenario. The following examples are presented primarily in Ladder Logic, as it is the most intuitive for beginners and widely used in maintenance and troubleshooting environments.

Example 1: Two-Hand Safety Control for a Pneumatic Press

Problem Description

In many manufacturing environments, operators must use both hands to start a machine cycle, ensuring their hands are away from hazardous moving parts. This example creates a simple two-hand anti-tiedown circuit using the S7-1200.

Hardware Mapping

- Input I0.0 – Left Hand Button (normally open)
- Input I0.1 – Right Hand Button (normally open)
- Output Q0.0 – Actuate Cylinder (start press cycle)
- Input I0.2 – Cylinder Home Position (limit switch)
- Input I0.3 – Cylinder Extended Position (limit switch)

Programming Logic

The core requirement is that both buttons must be pressed within 0.5 seconds of each other. If one button is pressed alone, the output remains off. Once both are recognized simultaneously (or with a small timing window), the cylinder extends. Releasing either button during extension retracts it immediately.

In Ladder Logic, we implement a simple AND condition using normally open contacts for I0.0 and I0.1 in series. However, to prevent a single button from being tied down, we add a cross-check: both inputs must transition to 1 in the same scan cycle (or within a short time). Using the built-in TON (On-Delay Timer) on each input creates a “simultaneity” check. If both timers are active within 500 ms, the output is set. The rung also includes a latch for the output until the cylinder reaches the extended position, after which the cycle resets automatically.

Key learning points: Input debouncing, safety logic with timers, and state-based control using set/reset coils. This **Siemens S 7 1200 PLC programming example**demonstrates how to handle operator safety in a compact and reliable manner.

Example 2: Conveyor Belt Sorting System with Proximity Sensors

Objective

Automated sorting lines require the PLC to track items on a conveyor belt and divert them based on a sensor reading. This example simulates a simple sorting system where a prox sensor detects the presence of a metal part, and a pneumatic diverter pushes it into a chute.

I/O Assignment

- I0.4 – Belt Start/Stop Pushbutton (momentary)
- I0.5 – Proximity Sensor at detection point
- I0.6 – Diverter Home Limit Switch
- I0.7 – Diverter Extended Limit Switch
- Q0.1 – Belt Motor Contactor
- Q0.2 – Diverter Solenoid (extend)

Ladder Logic Implementation

The belt runs continuously once started. When the proximity sensor (I0.5) detects a part, the PLC triggers a timer (T1) set to the conveyor travel time from sensor to diverter. After T1 elapses, the diverter solenoid (Q0.2) is energized, pushing the part off. The diverter remains extended for 1 second (T2) to ensure the part clears, then retracts automatically. A retract coil is energized when the diverter home limit switch is not active and the timer has expired.

This example introduces **sequencing using timers**and the importance of handling cylinder end positions. It also shows how to create a simple state machine in Ladder Logic using internal memory bits (M bits) to represent states like RUNNING, WAITING_DIVERT, and RETRACTING.

Variation for advanced learners: Add a counter to track the number of parts sorted, and display the count on an HMI (Human Machine Interface) using data blocks. This transforms the basic **PLC programming example** into a production monitoring tool.

Example 3: Three-Tank Level Control with PID

Scenario

Process control often involves maintaining a fluid level in a tank. Here we program a three-tank system where Tank 1 feeds Tank 2, and Tank 2 feeds Tank 3. Each tank has a level sensor (analog input) and a pump (digital output). The goal is to maintain the level in Tank 2 at a setpoint by adjusting the speed of the pump from Tank 1, while Tank 3 is emptied cyclically.

Hardware Setup

- AI0 (analog input 0-10V) – Level sensor Tank 2
- AQ0 (analog output 0-10V) – Variable frequency drive for Pump 1
- DI I1.0 – Level switch high Tank 1
- DI I1.1 – Level switch low Tank 1
- DI I1.2 – Level switch high Tank 3
- DO Q0.3 – Drain valve Tank 3 solenoid
- DO Q0.4 – Fill valve Tank 3 from Tank 2 (on/off)

Creating a PID Loop in TIA Portal

The S7-1200 includes a powerful PID compact instruction embedded in the technology objects. Instead of writing custom logic, we add a “PID_Compact” block from the instruction library. The configuration wizard allows us to set the setpoint (e.g., 50% of tank level), the input (AI0 scaled to a real value in mm), and the output (AQ0 scaled to 0-100%). We then manually tune the PID parameters (proportional gain, integral time, derivative time) either in automatic tuning mode or by using the available trend view.

While the PID block handles the continuous control, the discrete logic manages the fill and drain operations for Tank 3. When Tank 3 high switch activates, the drain valve opens until the low switch signals empty, then closes. The fill valve from Tank 2 operates only when Tank 2 is above a minimum level and Tank 3 is below its high level. This interlocking prevents pump starvation.

What this Siemens S7-1200 PLC programming exampleteaches: Analog signal scaling, PID loop configuration and tuning, and integration of continuous and discrete control in a single program. It also highlights the importance of cycle time because analog processing uses floating-point arithmetic.

Example 4: Sequential Batch Process with Step Control

Process Overview

Many industrial recipes follow a fixed sequence of operations: fill, mix, heat, drain, rinse. This example implements a four-step batch sequence using a sequencer (step ladder) pattern. The machine has three agitators, a heater, a fill valve, and a drain valve.

Step Control with Shift Register

Using a shift register made from internal bits (M10.0 to M10.3), each step is active only when the previous step completes. The first step (Fill) is triggered by a start pushbutton (I0.0). When the fill level sensor reaches the high limit, the step bit shifts to Mix. The Mix step runs a timer for 10 minutes, then shifts to Heat. The Heat step runs until temperature sensor (AI1) reaches 80°C, then shifts to Drain. The Drain step opens the drain valve until the low level switch clears, then resets the sequencer and returns to step 0 (idle).

This pattern is extremely reusable. By replacing the completion conditions with different sensors or timers, you can adapt the sequence to any production line. The code is self-documenting when you add comments in TIA Portal for each step.

Advanced addition: Store recipe parameters in a DB (Data Block) – such as target temperatures, mixing

times, and fill levels – and allow the operator to select a recipe via HMI. This turns the example into a configurable batch controller, a common task in food and pharmaceutical industries.

Example 5: Motor Control with Soft Start and Star-Delta Transition

Context

Large induction motors often require reduced voltage starting to limit inrush current. A star-delta starter switches the motor windings from star (low voltage) to delta (full voltage) after a short delay. The S7-1200 can control this transition precisely using timers and interlocks.

Program Structure

- I1.3 – Start Pushbutton
- I1.4 – Stop Pushbutton
- Q0.5 – Star Contactor
- Q0.6 – Delta Contactor
- Q0.7 – Main Power Contactor

The logic: On start, the main contactor and star contactor energize simultaneously. A timer (T3) is set to 3 seconds (star period). When the timer expires, the star contactor is de-energized, and after a 100 ms delay (to prevent short circuit), the delta contactor energizes. A stop pushbutton de-energizes all contactors. Safety interlocks ensure both star and delta are never on together – implemented by writing the output coils with normally closed contacts of the opposite contactor.

This **Siemens S 7 1200 PLC programming example** demonstrates **interlocking logic**, which is crucial for any motor control circuit. It also introduces the concept of transitional delays to avoid electrical arcs.

Best Practices When Learning from Programming Examples

To get the most out of any **Siemens S7-1200 PLC programming example**, follow these guidelines:

Related Articles

- **Apexvs Learning Answers**
  URL: https://signage.openstack.org/apexvs-learning-answers.pdf
- **Learning The Art Of Helping 7th Edition**
  URL: https://signage.openstack.org/learning-the-art-of-helping-7th-edition.pdf
- **Learning The Art Of Helping 5th Edition**
  URL: https://signage.openstack.org/learning-the-art-of-helping-5th-edition.pdf
- **Prosuite Allscripts Cloud Login**
  URL: https://signage.openstack.org/prosuite-allscripts-cloud-login.pdf

TOPIC INDEX

#siemens • #1200 • #plc • #programming • #examples

DIGITAL EDITION

siemens-s-7-1200-plc-programming-examples

DIGITAL EDITION

siemens-s-7-1200-plc-programming-examples

DIGITAL EDITION

siemens-s-7-1200-plc-programming-examples

DIGITAL EDITION

siemens-s-7-1200-plc-programming-examples