CREST Gold Award Project Report

Design and Development of a PLC Train Level Crossing Simulator

Author: Patryk Derlukiewicz
Host Company: Proeon Systems Ltd, Norwich (part of RSK Group)

Programme: T Level Mechatronic Student Industry Placement

Award Sought: CREST Gold Award (British Science Association)

Date: July 2026

Project hours: 50 to 70+ hours

Contents

Abstract

This report documents the design, development, and testing of a PLC based train level crossing simulator, produced as part of an industry placement at Proeon Systems Ltd, a Norwich based control systems and industrial automation company operating across the energy, nuclear, offshore wind, oil and gas, chemicals, food and beverage, and manufacturing sectors. The simulator reproduces the control sequence of a road level crossing, using push buttons in place of physical train detection sensors to simulate train arrival and departure events. The control logic was developed in Siemens TIA Portal V20, running on a virtual machine, and includes a debounce function block to prevent a single button press from being registered as multiple, unintended inputs.

The project was chosen because a level crossing control system draws together several strands of engineering practice in a single, manageable scope: sequential control logic, digital input and output handling, timer based operation, and consideration of safety critical behaviour. This makes it well suited to demonstrating the planning, technical understanding, problem solving, and reflective practice required for a CREST Gold Award. The report covers the background research behind the project, the planning process, the system design and programming approach, the testing carried out, the problems encountered (in particular, false triggering caused by mechanical switch bounce) and how these were resolved, an evaluation of the finished system, and a personal reflection on what was learned.

1. Introduction and Project Aims

Level crossings are one of the more safety critical interfaces in the rail industry, where a road and a railway line cross at the same level. Failures or ambiguous signalling at a crossing can have serious consequences, so crossing control systems are built around clear, well defined logical sequences, redundancy in detection, and fail safe behaviour. Studying and building a simplified version of this kind of system is a practical way to engage with core industrial control engineering principles that are directly relevant to the sectors Proeon works in, including rail adjacent infrastructure, energy, and manufacturing, where PLC controlled sequential systems are common.

1.1 Aim

The aim of this project was to design, build, and test a functioning PLC based simulation of a road level crossing, using push button inputs to represent train detection, and to demonstrate reliable, repeatable control of the barriers and warning lights through a correctly sequenced and debounced control program.

1.2 Objectives

- Research how real level crossing control systems operate, and identify the core stages of a crossing sequence that the simulator should reproduce.
- Select an appropriate development platform and simulate the required inputs and outputs.
- Design a control sequence covering train arrival, warning activation, barrier operation, and the safe return to normal running.
- Program the sequence in Siemens TIA Portal V20 using ladder logic, including a debounce function block for the push button inputs.
- Test the system against a range of operating scenarios, including rapid or repeated button presses, and identify and correct any faults.
- Evaluate the finished simulator against the original aim and reflect on what could be improved.

1.3 Wider Purpose

Beyond the immediate goal of building a working simulator, the project has a wider purpose in demonstrating how sequential, safety oriented control logic is developed in an industrial setting. Proeon designs and integrates control systems for clients across energy, nuclear, and process industries, where the same underlying principles apply: physical inputs must be filtered and validated before being trusted, control sequences must move through clearly defined states, and the system must default to a safe condition if something goes wrong. A level crossing is a widely understood, easily visualised example of this pattern, which makes it a useful vehicle for learning and for communicating engineering ideas to a non specialist audience.

2. Background Research

2.1 How Real Level Crossings Operate

Road level crossings in the UK are controlled through a sequence that begins with train detection, typically using track circuits, axle counters, or treadles positioned a set distance from the crossing, calculated from the maximum permitted line speed so that road users receive adequate warning time. Once a train is detected, audible and visual warnings are activated, followed a short time later by the lowering of barriers, and the crossing remains in a locked state until the train has fully cleared a second detection point beyond the crossing. Only then do the barriers raise and the warnings stop. This staged approach, with a clear separation between detection, warning, locking, and clearance, was the basis for the simulator's control sequence.

2.2 Programmable Logic Controllers and Ladder Logic

A programmable logic controller, or PLC, is an industrial computer designed to control machinery and processes through digital and analogue inputs and outputs, typically programmed using ladder logic, a graphical language modelled on relay wiring diagrams. Ladder logic is standardised, in outline, under IEC 61131-3, which defines several programming languages for programmable controllers, including ladder diagram. Siemens TIA Portal is one of the main development environments used for Siemens S7 family PLCs, and it was chosen for this project because it is the platform used within Proeon's own control system work, giving direct relevance to the placement.

2.3 Switch Bounce and Debouncing

Mechanical push buttons and switches do not produce a single, clean transition between open and closed states when they are pressed or released. Instead, the contacts physically bounce for a short period, typically a few milliseconds, producing a rapid series of on and off transitions before settling. In a digital control system scanning inputs at high speed, this bounce can be misread as several separate button presses rather than one, leading to unintended repeated triggering of the control logic. This is a well known issue in digital electronics and PLC programming, and is normally solved either in hardware, using an RC filter or a Schmitt trigger circuit, or in software, using a debounce timer that ignores further changes of state for a short period after the first detected transition. For this project, the debouncing was implemented in software, within the PLC program itself, as a function block.

2.4 Health and Safety Considerations in Crossing Design

Level crossing risk management in the UK is overseen by the Office of Rail and Road, and crossing operators are required to carry out risk assessments and follow guidance on crossing design and maintenance. While this project is a simplified educational simulation rather than a real crossing, the background research into fail safe design, clear sequencing, and the consequences of a missed or delayed warning informed the safety conscious approach taken in the control logic, discussed further in Section 7.

3. Planning and Methodology

3.1 Considering Different Approaches

Several approaches were considered before settling on the final method. A purely simulated system running entirely in software, with no physical inputs, was considered but rejected as it would not demonstrate practical input handling or debouncing. Building a fully physical model with real sensors, motors, and barriers was also considered, but this was not practical within the time and equipment available during the placement, and it would have shifted the focus toward mechanical construction rather than control logic. The approach finally chosen used physical push buttons, wired or simulated as digital inputs, to represent train detection events, combined with a fully developed PLC program including debounce logic. This struck a balance between a practical, testable, hands on system and a project that could realistically be completed within the placement.

3.2 Project Plan

The project was broken down into the following stages, planned before development began:

- Research level crossing operation and PLC debouncing techniques.
- Define the input and output list and the sequence of operation.
- Build the ladder logic program in Siemens TIA Portal V20 on a virtual machine.
- Add and configure the debounce function block for the push button inputs.
- Simulate and test each stage of the sequence individually.
- Test the complete sequence end to end, including edge cases such as rapid repeated presses.
- Identify and resolve any faults found during testing.
- Evaluate the finished system and write up the report.

3.3 Materials and Support Used

The main resources used for this project were access to Siemens TIA Portal V20 running on a virtual machine, provided through the placement at Proeon Systems Ltd, and informal guidance from colleagues at Proeon when questions arose about PLC programming conventions and good practice. Background research was carried out using Siemens product documentation and general engineering references on PLC programming and level crossing operation.

4. System Design

4.1 Sequence of Operation

The simulator was designed around the following sequence, using a button to represent train arrival and a second button to represent the train clearing the crossing:

Stage	Description
1	Idle state: road open, barriers up, no warning lights active.
2	Train arrival button pressed: amber warning lights activate.
3	After a short timed delay, the barriers begin to lower.
4	Once the barriers are down, red warning lights flash continuously while the crossing is occupied.
5	Train clearance button pressed, representing the train passing the crossing.
6	Barriers raise and the warning lights deactivate.
7	System returns to the idle state, ready for the next cycle.

4.2 Inputs and Outputs

The digital inputs and outputs used in the program were defined as follows: two push button inputs, one for train arrival and one for train clearance, each passed through the debounce function block before being used elsewhere in the program; and four outputs, representing the amber warning lamp, the red flashing warning lamp, and the barrier motor's raise and lower commands. Along with this all the inputs were put into memory stores rather than using raw inputs.

4.3 Debounce Function Block

Each push button input is routed through a dedicated debounce function block before it is allowed to affect the sequence logic. On detecting a change of state at the input, the block starts a short timer and ignores any further changes of state until that timer has elapsed, at which point it outputs a single, clean pulse representing one genuine press. This prevents the contact bounce described in Section 2.3 from being interpreted as multiple separate train arrival or clearance events or wildlife interference.

5. Hardware and Software

5.1 Development Environment

The control program was developed entirely in Siemens TIA Portal V20, running on a virtual machine, which allowed the PLC program to be written, compiled, and simulated without requiring a physical PLC to be available for the whole duration of development. This is a common approach in industrial control system development, where offline simulation is used to test and refine logic before it is deployed to physical hardware, reducing the risk of errors being introduced directly on site.

5.2 Justification for Using a Virtual Machine and Simulation

Using a virtual machine running TIA Portal, rather than requiring dedicated physical hardware for the full development period, made it possible to iterate on the program quickly, test edge cases repeatedly without risk to physical equipment, and work flexibly around the availability of shared hardware within the placement. This mirrors real industrial practice, where control logic is very often developed and validated in simulation before commissioning on site.

6. PLC Program Development

6.1 Programming Approach

The program was developed using ladder logic, building up the sequence in stages rather than attempting to write the complete logic in one pass. The idle state and the train arrival trigger were implemented first, followed by the timed delay before barrier lowering, the barrier down and warning light logic, and finally the clearance and reset logic. Building the program incrementally in this way made it possible to test each stage in isolation before integrating it with the rest of the sequence, which made faults easier to locate.

6.2 Timers and Sequential Control

Timers were used at two points in the sequence: a short delay between the warning lights activating and the barriers beginning to lower, intended to give simulated road users adequate warning before the crossing becomes obstructed, and a timer within the debounce function block itself. Sequential control was managed using a simple state based structure, where the current stage of the crossing cycle determines which outputs are active and which inputs are being monitored, preventing, for example, the clearance button from having any effect while the system is still in its idle state.

7. Testing and Results

7.1 Test Strategy

Testing was carried out in two phases. The first phase tested each stage of the sequence individually, confirming that each output responded correctly to its associated input and timer condition. The second phase tested the complete sequence end to end, including deliberately unusual scenarios such as pressing a button rapidly several times in quick succession, pressing both buttons close together, and pressing the clearance button while the system was still in its idle state, to check that the sequence logic correctly ignored inputs that should have no effect at that stage.

7.2 Results

The staged sequence, when tested one stage at a time, behaved as designed: the warning lights, barrier movement, and reset logic all activated in the correct order and returned the system to idle correctly at the end of a cycle. During full sequence and edge case testing, however, rapid or repeated pressing of the train arrival button revealed false triggering caused by mechanical switch bounce, which is discussed in detail, along with the fix applied, in Section 8.

8. Problems Encountered and Solutions

8.1 False Triggering from Switch Bounce

The main problem encountered during development was false triggering of the train arrival and clearance inputs, caused by mechanical contact bounce in the push buttons. Because the PLC scans its inputs many times per second, a single physical button press was, in some cases, being registered as several separate presses within a few milliseconds, causing the sequence logic to behave unpredictably rather than moving cleanly from one stage to the next.

8.2 Solution: Software Debounce Function Block

The solution adopted was to add a debounce function block ahead of the main sequence logic, so that every button input passes through debouncing before it is allowed to trigger a change of stage also known as power up timers. Once a change of state is detected, the block holds that state for a short period, ignoring any further transitions, before passing through a single clean signal. This removed the false triggering observed during testing and made the sequence behave reliably even when a button was pressed rapidly or repeatedly. Additionally, the debounce blocks were stated as single function and not multifunction, therefore the raw signal wasn't activating the debounce blocks not allowing the sequence to continue. This was spotted due to a naming error as there is a difference between titles of these blocks, and was adjusted accordingly to work

8.3 Other Issues

Other issues included one button being wired as a normally closed rather than a normally open, therefore this was adjusted within the code by setting the raw value into a memory storage; which was then used as a normally open contact into a output of a separate memory store.

9. Evaluation

9.1 Evaluation Against the Original Aim

The finished simulator meets the original aim set out in Section 1: it reliably represents the core stages of a level crossing cycle, from idle through warning, barrier operation, and reset, and does so using debounced push button inputs that behave predictably even under repeated or rapid pressing. The project also met its secondary objectives of researching real crossing operation, selecting an appropriate development platform, and testing the system against a range of scenarios rather than only the expected, straightforward case.

9.2 Limitations

As a simulation built around push buttons rather than genuine train detection sensors, the system does not address the additional complexity of real world detection, such as verifying that a train has fully cleared the crossing before releasing the barriers, or handling sensor faults and redundant detection, which are essential features of a real level crossing control system. The project deliberately scoped these aspects out in order to focus on sequential control logic and input debouncing within the time available.

9.3 Wider Implications

The core lesson from this project, that raw digital inputs from mechanical switches cannot be trusted without filtering, and that safety related sequential logic must be built around clearly defined, mutually exclusive states, applies well beyond level crossings. The same principles underpin the design of PLC based control systems across the sectors Proeon works in, including energy, nuclear, and process industries, where an unfiltered or misread input could lead to an unintended and potentially unsafe change of state in a real system.

10. Conclusions

This project set out to design, build, and test a PLC based train level crossing simulator using debounced push button inputs, developed in Siemens TIA Portal V20 during an industry placement at Proeon Systems Ltd. The finished system successfully reproduces the core sequence of a road level crossing and reliably rejects false inputs caused by mechanical switch bounce, following the addition of a dedicated debounce function block. The project demonstrated practical application of ladder logic programming, sequential and timer based control, and a safety conscious approach to input handling, and provided a concrete, testable example of principles that are directly relevant to industrial control engineering.

11. Personal Reflection

11.1 Tasks completed and how successful the project was and what I learnt

I consider the project to have been a complete success. I successfully developed the railway level crossing simulator, with both the PLC program and the HMI operating together correctly. The sequence functioned as intended throughout testing, and evidence of the completed project has been provided in the attached documentation and supporting media.

This project significantly improved my understanding of Siemens TIA Portal, as it was my first time using the software. I gained experience creating and using function blocks to produce more organised and reusable code, implemented debounce blocks to prevent unwanted multiple inputs from push buttons, and learned how different timers can be used to control sequencing and delays within an automated process. Alongside the technical knowledge, I also developed a better understanding of how to structure PLC programs and troubleshoot issues systematically.

11.2 What impact this might have on others or the wider world

Although this project was a simulator, it reflects principles used in real industrial automation systems. Reliable input handling, accurate sequencing and clear operator interfaces are essential in applications where safety is critical, such as railway crossings and manufacturing processes. Working on this project gave me a better appreciation of how careful PLC programming contributes to safe and dependable control systems.

11.3 What I would have done to improve my work

If I were to complete the project again, I would organise my PLC tags, function blocks and program folders in a more systematic way from the beginning. A clearer structure would make the program easier to navigate, debug and maintain, particularly as the project becomes more complex or when other engineers need to understand the code.

11.4 What I would do to develop the project in the future

To further develop the project, I would implement a genuine manual override system that could be used in the event of a fault or emergency. This would better reflect how industrial control systems are designed, allowing an operator to safely take control if the automatic sequence were to fail. Additional improvements could include more fault diagnostics, alarms and extra safety monitoring to make the simulator even more realistic.

12. References

- British Science Association, CREST Awards, Gold Award Student Guide, available at www.crestawards.org
- International Electrotechnical Commission, IEC 61131-3, Programmable controllers, Part 3: Programming languages
- Office of Rail and Road, guidance on level crossing safety and risk management, available at www.orr.gov.uk
- Siemens AG, SIMATIC STEP 7 (TIA Portal) documentation, Siemens Industry Online Support
- East of England Energy Group (EEEGR), CREST Award information for Coastal Energy host companies