A Village House Energy-Supply System: Fundamentals of Energy Conversion

Article *in* IEEE Antennas and Propagation Magazine · September 2009

DOI: 10.1109/MAP.2009.5338752 · Source: IEEE Xplore

2 authors:

S. Kurtulan
Istanbul Technical University
26 PUBLICATIONS **140** CITATIONS
SEE PROFILE

Levent Sevgi
ATLAS University
291 PUBLICATIONS **3,675** CITATIONS
SEE PROFILE

Automata-Based Railway Signaling and Interlocking System Design

Emre Dincel, Oytun Eris, and Salman Kurtulan

Control Engineering Department
Istanbul Technical University, Istanbul, Turkey
E-mail: [dincele, erisoy, kurtulans]@itu.edu.tr

Abstract

Railway signaling has become more important over the years, with the increase in railway traffic and the use of high-speed trains. Since the most important aspect of a railway signaling system is safety, the decision-making "interlocking system" is the critical element of a railway-signaling system. To satisfy the safety standards set for railway-signaling systems, the interlocking software design must be taken into account, using formal methods. In this study, an automata-based railway signaling and interlocking system is introduced and implemented, using a fail-safe programmable logic controller (PLC) on a scale railway model.

Keywords: Railway signalization; automata; interlocking system design; fail-safe system; PLC

1. Introduction

Timetables and flag officers were used in the early days of railway signaling. However, with the increase in the number of accidents caused by human-oriented errors, automatic signaling became essential. The decision-making unit in railway systems is called the interlocking system. The responsibility of the interlocking system is to evaluate the route request in order to prevent potential accidents, so that trains travel safely on railways [1].

With the first implementation of a mechanical interlocking system in 1843 [2], the history of automatic signaling began. Parallel to the improvement of the technology and electronic systems, mechanical interlocking systems left their place to electronic interlocking systems. Nowadays, programmable logic controllers (PLC), which are specialized for industrial-automation systems, are used for interlocking systems.

Safety is the most important criterion for all transportation systems. Especially in railway systems, even a small mistake can result in fatal consequences [3, 4]. In Turkey alone between 2001 and 2008, 1118 people were killed in accidents caused by train collisions, passage collisions, and derailments [5, 6]. To ensure the safety of such systems, in most countries, the umbrella standard IEC 61508 for critical safety systems, and EN 50128 for railway applications, are required.

It is obvious that a decisive interlocking system is directly responsible for avoiding any possible collisions in a railway system [7]. The interlocking system must satisfy both the hardware and software requirements of the standards. While it is easy to satisfy the hardware aspect by using certified COTS (commercial off-the-shelf) products, safe software design needs more effort and formal approaches. One of the main reasons for the famous Gare de Lyon rail accident was a flaw in the design of the interlocking systems' emergency procedure, which directed the train to the busy platform instead of the empty platform [8]. A high-speed train accident in China with 39 fatalities was caused by a flawed signal showing green after a lightning strike [9]. Another accident in Tamil Nadu (India) was also known to be an interlocking-system error, called a "signal flew back error," which means the system showed the wrong signal and turned back to red after a while [10].

Interlocking-system design can be performed by different methods. Graph-based and interlocking-table-based methods can be given as examples [11-13]. In addition, there are other studies that are based on automata [14, 15] and Petri nets [16-18].

In this study, the design steps of an interlocking system intended to provide safe operation for a scale railway model by using automata theory are discussed, and interlocking software was developed. The software was then implemented in a Siemens CPU 317F-2 model Fail-Safe PLC, in order to show the success and the applicability of the method.

2. Railway Signalization Systems

2.1 The Traffic Control Center

Train movements and all other processes in a railway line are monitored and carried out by the Traffic Control Center (TCC). Positions of switches, states of signal lights, locations of trains, and other information can be monitored and controlled [19].

There is a bidirectional data exchange between the Traffic Control Center and the interlocking system. Requests sent by the Traffic Control Center are evaluated and applied to the field equipment by the interlocking system. At the same time, answers for the requests and feedback information taken from the field are transferred to the Traffic Control Center.

2.2 The Interlocking System

The interlocking system is a safe bridge between the field and the Traffic Control Center. The interlocking system checks the suitability of the commands requested from the Traffic Control Center with the help of feedback signals and approves these commands in order to send them to the field, if they are suitable. Otherwise, pending requests are rejected, and necessary notifications are also transferred to the Traffic Control Center by the interlocking system.

2.3 Track Circuits and Axle Counters

A track circuit is a simple electrical component that is used to detect the existence of trains in a railway system. There are two types of track circuits: ac and dc. As the train passes over the track circuit, it is short circuited: detection of the train on the track is thereby possible [16]. Axle counters are also used to detect train movements. They are positioned at the ends of railway blocks, and count the number of incoming and outgoing axles. If the number of incoming axles is higher than the number of outgoing axles, it is said that the railway block is occupied [20].

2.4 Switches

Switches are the components that provide passage of trains from one track to another, and are generally controlled by motors. Switches have two positions: a normal position, and reverse. A switch is considered to be in its normal position if the train continues directly on the path, and is in the reverse position if train deviates from the path.

2.5 Signal Lights

Signal lights give information to the machinist about the next routes, such as permission for passing to the next route, the necessary speed of the train, and the state of the next signal lights. Signal lights are positioned on the right side of the railway, according to the direction of movement. There are four types of signal lights – two-aspect dwarf signals (red and green), three-aspect dwarf signals (red, green, and yellow), three-aspect tall signals (red, green, and yellow), and four-aspect tall signals (red, green, and two yellow) – which are used in Turkish railways. Table 1 shows the meanings of the signal-light notifications.

3. Interlocking System Design

As explained earlier, interlocking system design is an important subject, due to its vital role. Procedures such as are applied in the interlocking system design can be explained as follows. When a route request comes from the Traffic Control Center, tracks which are included in the requested route are first checked and, if there is an occupation, the route request is rejected by the interlocking system. If there is no occupation, these tracks are electronically locked. After that, if there is a switch on the route, the switch position is changed to the appropriate position, and feedback (position-indication information) is awaited for confirmation. If any problem occurs, the route request is rejected, and necessary notifications are sent to the Traffic Control Center. Signal lights are then properly set up, and again feedback (signal-light-indication information) is awaited for confirmation. If there is no problem, the route request is accepted. Figure 1 shows a simple flow chart for the route acceptance/rejection process.

While the train is moving on its reserved route, track circuits are checked by interlocking system to see if the train is entering the tracks in the correct order. If there is a switch, its position is always checked until the route is released to ensure safety. Signal lights are also checked to be sure whether or not they show the right notification. Finally, when the train reaches the end of the route, all track circuit occupations are cleared, and the route is released for the next request by the interlocking system.

In order to ensure proper operation of the system, both for electronic and relay-based interlocking systems, a well-organized interlocking table is needed. The interlocking table should consist of information such as track circuits, switches, signal lights, and their notifications, depending on the route. With the help of this table, some logical operations and function blocks are easily created to implement the interlocking system in hardware. Figure 2 shows a simple railway yard, and Table 2 shows a part of the interlocking table that was created for the railway line given in Figure 2.

Table 1. Signal lights and their meanings.

Signal Lights	Definition
Red	The next railway block is occupied (Stop)
Yellow	The next railway block is free but the second to the next one is occupied (Continue with a predefined speed)
Green	The next two railway blocks are free (Continue)
Yellow–Red	The next block is occupied and there is a track deviation (Continue as stop at any moment)
Yellow–Yellow	The next block is free and there is a track deviation (Continue with a predefined speed and track deviation)
Yellow–Green	The next two blocks are free and there is a track deviation (Continue with track deviation)

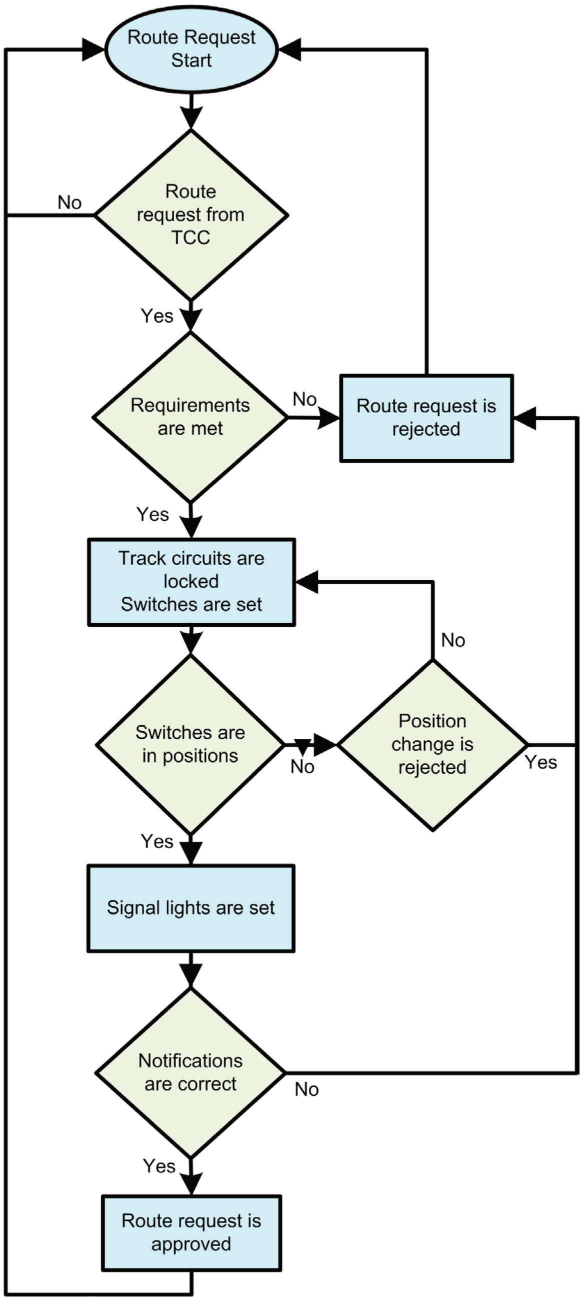
Route Request Start
Route request from TCC
No
Yes
Requirements are met
No
Route request is rejected
Yes
Track circuits are locked Switches are set
No
Switches are in positions
No
Position change is rejected
Yes
Yes
Signal lights are set
Notifications are correct
No
Yes
Route request is approved

Figure 1. A flowchart for the route request.

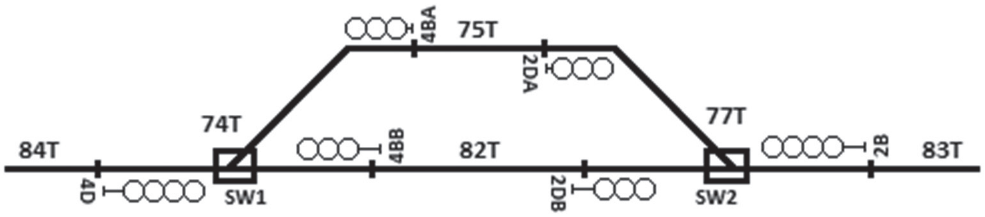

Figure 2. A simple railway yard.

Table 2. A part of an interlocking table for the railway line.

Route Tracks	Signal Light		Preceding Signal Light	Switch Position
84T-74T-82T	4D	Y	2DB – R	SW1 - N
		G	2DB – G, Y	
		R		
84T-74T-75T	4D	YY	2DA – R	SW1 - R
		YG	2DA – G, Y	
		YR		
83T-77T-82T	2B	Y	4BB – R	SW2 - N
		G	4BB – G, Y	
		R		
83T-77T-75T	2B	YY	4BA – R	SW2 - R
		YG	4BA – G, Y	
		YR		

4. Automata-Based Design

Automaton – also known as the state-transition graph – is a graphical representation that consists of an initial state and other states, state transitions, and events that provide the transition between states. In order to obtain the state-transition graph of a system, states and events first need to be determined. The state-transition graph is then drawn, and logical expressions are generated by using this graph. Figure 3 illustrates a state-transition graph.

In the next step, state-transition functions have to be generated. These functions are determined depending on conditions that bring automaton to the concerned state, and also take automaton of the concerned state. For automaton that consists of m states, the state-transition functions can be expressed as follows [21]:

$$Q_i = \sum_{j=1}^{m} q_j T_{j,i} + q_i \prod_{k=1}^{m} T_{i,\text{k}}, \quad i \neq j, i \neq k . \qquad (1)$$

Here, the Q_i are the next states, q_i and q_j are the current states, $T_{j,i}$ are conditions that bring the state from the current state (q_j) to the (Q_i) next state, and $T_{i,k}$ are conditions that bring the state from the current state (q_i) to the (Q_k) next state. The initial state expression is given as follows:

$$Q_1 = \prod_{k=2}^{m} \overline{q}_k . \qquad (2)$$

This means that if the automaton is not in any state, it should be in the first state. The output functions can also be described depending on the current states as follows:

$$z_j = f_j\left(q_1, q_2, \ldots, q_m\right), \quad j = 1, 2, \ldots, n . \qquad (3)$$

In automata theory, signals that determine the circuit behavior are instantaneous signals. An instantaneous signal is sufficiently long-term in order to cause state transition; however, it is ineffective after the state's transition. These types of signals can be produced by positive and negative edge detectors in programmable logic controllers.

The state variables on the left side of the state equations are $Q_k = Q_k\left(t+T\right)$, and on right side of the state equations are $q_k = Q_k\left(t\right)$. For an automaton that consists of m states, this assignment is given as [21]

$$q_k = Q_k, \quad k = 1, 2, \ldots, m. \tag{4}$$

State equations for the example state-transition graph given in Figure 3 are

$$\begin{aligned}
Q_1 &= \overline{q}_2 \overline{q}_3 \overline{q}_4, \\
Q_2 &= q_1 e_1 + q_4 e_3 + q_2 \overline{e}_2, \\
Q_3 &= q_2 e_2 + q_3 \overline{e}_4, \\
Q_4 &= q_3 e_4 + q_4 \overline{e}_2 \overline{e}_4, \\
q_1 &= Q_1, \\
q_2 &= Q_2, \\
q_3 &= Q_3, \\
q_4 &= Q_4.
\end{aligned} \tag{5}$$

5. A Case Study

5.1 Representation of the Hardware Simulator

In order to test the interlocking system, a hardware simulator that was a scale model of an existing railway station was made. The sample station, Mithatpaşa (Turkey), consisted of 14 signals, 10 track circuits, 10 switches, and pme level crossing. It can be seen in Figure 4.

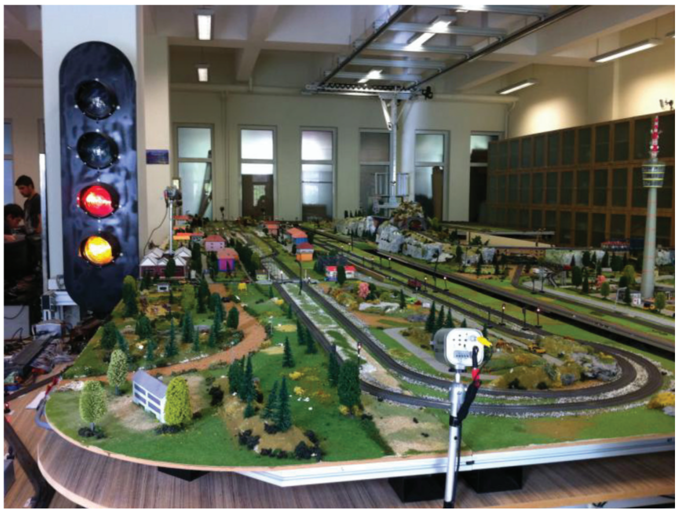
Figure 4a. The hardware simulator (general view).

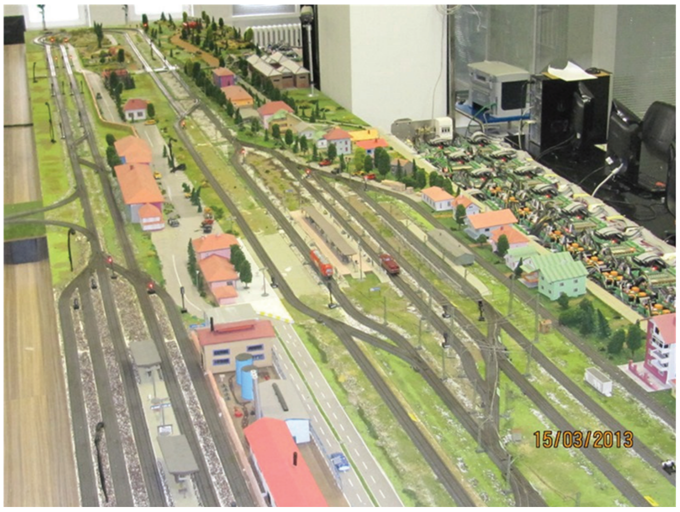

Figure 4b. The hardware simulator (Mithatpaşa station).

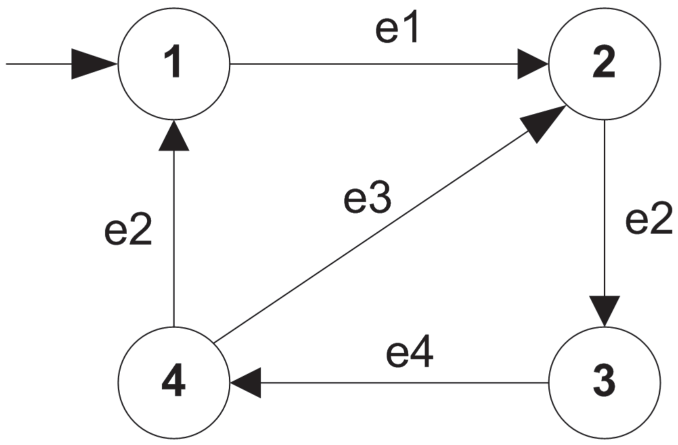

Figure 3. A state-transition graph.

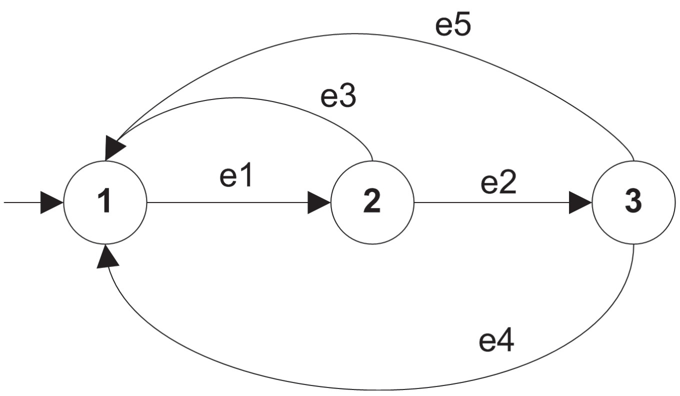

Figure 5. An automaton for a switch element.

5.2 Interlocking-System Design

5.2.1 Modeling of the Field Equipment by Automata

In the automata-based interlocking-system design, field equipment, such as switches, signal lights, and level crossings, should first be modeled. Security precautions should also be taken into account while creating states and events in automata. Once their state-transition graphs are obtained, it is possible to write logical functions using basic logical expressions, such as AND, OR, and NOT. The logical functions obtained are ready to be implemented in a programmable logic controller using a programming language. To this end, in this section, state-transition graphs are drawn and state-transition functions are obtained for switches, signals, and a level crossing.

Figure 5 shows the state-transition graph that was created for a switch element. When the position of the switch was requested to be changed from the Traffic Control Center, if all requirements were met (i.e., there was no locking operation or fault), then commands were sent by the interlocking system to the field. After that, it is expected that the switch will arrive in the desired position in seven seconds. Finally, if the correct position indication arrives in this time, the operation is completed and the automaton goes to the initial state. Otherwise, necessary notifications are sent to the Traffic Control Center by the interlocking system.

States are given in Table 3 and event descriptions are given in Table 4 for the automaton in Figure 5. In addition, the state-transition functions were obtained as

$$\begin{aligned} Q_1 &= \overline{q}_2\overline{q}_3 \\ Q_2 &= q_1e_1 + q_2\overline{e}_2\overline{e}_3 \\ Q_3 &= q_2e_2 + q_3\overline{e}_4\overline{e}_5 \,. \end{aligned} \tag{6}$$

In signaling systems, all signals should light up red unless otherwise required. When a signal is needed to light up a color other than red, this request is sent to the interlocking system. If all requirements are met, then commands are sent by the interlocking system to the field. After that, if the correct indication arrives in the specified time, the operation is completed; if not, the necessary notifications are sent to the Traffic Control Center by the interlocking system.

The automaton for signal lights is given in Figure 6, the states and their descriptions are given in Table 5, and the events are given in Table 6. The state-transition functions that were obtained using the automaton given in Figure 6 were also given as

$$\begin{aligned} Q_1 &= \overline{q}_2\overline{q}_3 \\ Q_2 &= q_1e_1 + q_2\overline{e}_2\overline{e}_3 \\ Q_3 &= q_2e_2 + q_3\overline{e}_4 \,. \end{aligned} \tag{7}$$

Level crossings are the points where roads and railways intersect. For this reason, level crossings are critical points, at risk for accidents. The designed interlocking system should close the level crossing when necessary by sending a signal to the field. The road traffic should thereby stop, and the train can continue its movement. After the passage of the train, the level crossing should be opened by the interlocking system so that road traffic continues to flow.

The automaton for the level crossing is given in Figure 7, the states and their descriptions are given in Table 7, and the events are given in Table 8. The state-transition functions are also given as

$$\begin{aligned} Q_1 &= \overline{q}_2\overline{q}_3\overline{q}_4 \\ Q_2 &= q_1e_1 + q_2\overline{e}_2\overline{e}_3 \\ Q_3 &= q_2e_2 + q_4e_6 + q_3\overline{e}_4 \\ Q_4 &= q_3e_4 + q_4\overline{e}_3 \,. \end{aligned} \tag{8}$$

5.2.2 Automaton Design to Evaluate Route Requests

A route request can be described as an event that comes from the Traffic Control Center so that a train can start to move. When a route request is sent by the Traffic Control Center, the interlocking system checks the suitability of the request as explained earlier, and performs the necessary operations. It is possible to create an automaton model to evaluate route requests, in accordance with Figure 1.

Table 3. The states and their descriptions for the switch automaton.

State	Description
1	Initial state
2	Request to change switch position has arrived
3	All requirements are met

Table 4. The events of the switch automaton.

Name of the Event	Event
e_1	Request to change switch position
e_2	Requirements are met
e_3	Requirements are not met
e_4	Switch has arrived to the desired position
e_5	Timer has expired

Table 5. The states and their descriptions for the signal-light automaton.

State	Description
1	Initial state
2	Request to change signal notification has arrived
3	Correct indication has arrived

Table 6. The events of the signal-light automaton.

Name of the Event	Event
e_1	Request to change signal notification
e_2	Indication is correct
e_3	Timer has expired
e_4	Notification request is terminated

Table 7. The states and their descriptions for the level-crossing automaton.

State	Description
1	Initial state (level crossing is open)
2	Request to close level crossing has arrived
3	Indication has arrived (level crossing is closed)
4	Request to open level crossing has arrived

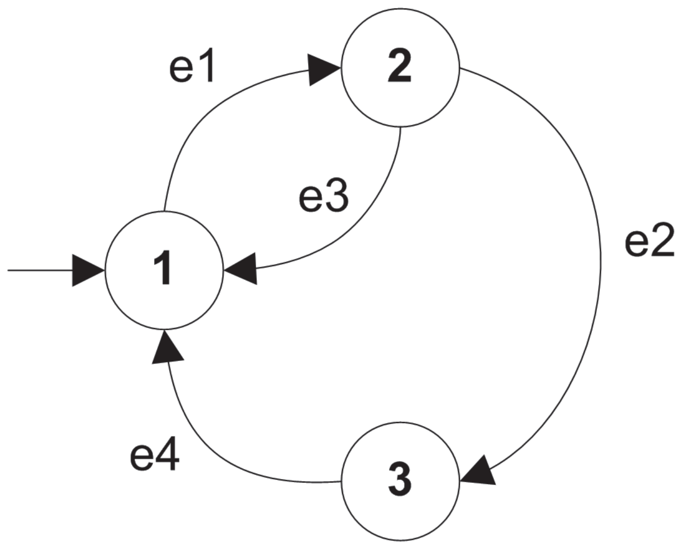

Figure 6. An automaton for a signal-light element.

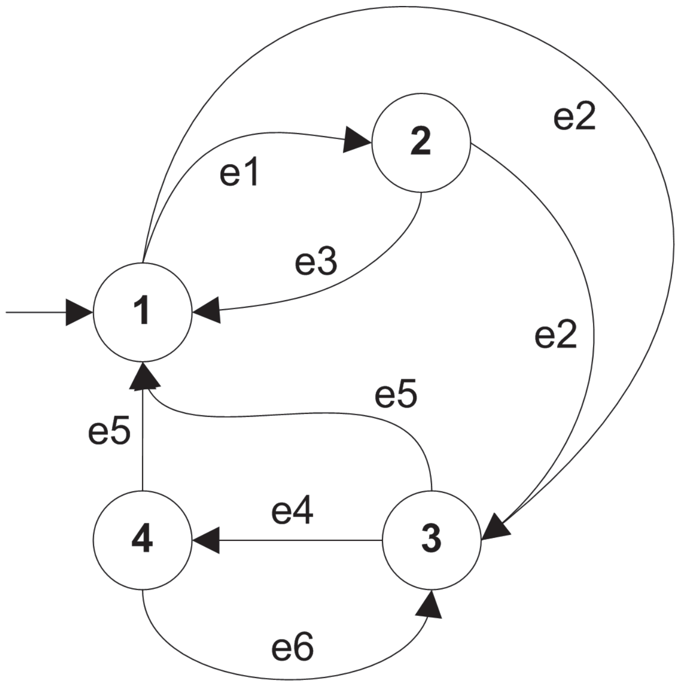

Figure 7. An automaton for a level-crossing element.

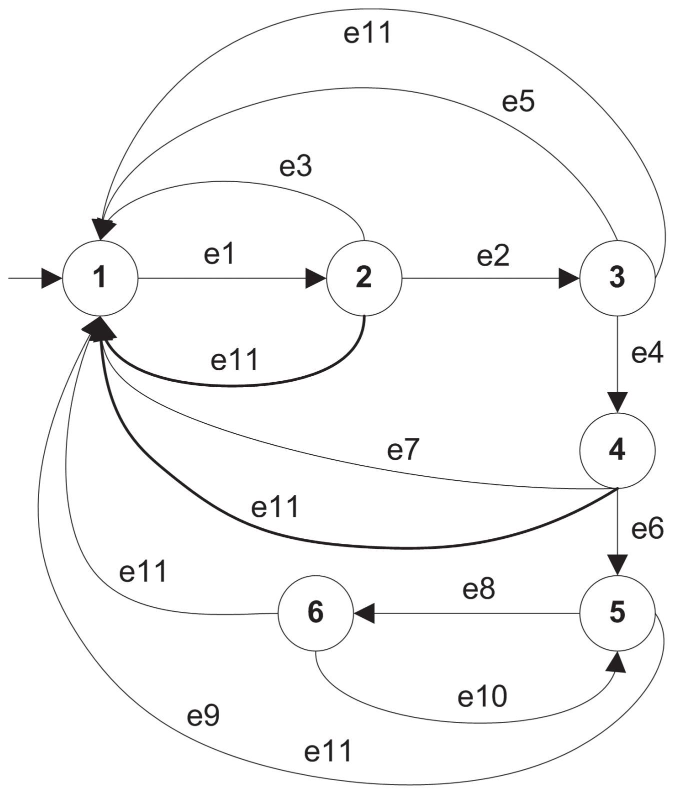

Figure 8. An automaton to check route requests.

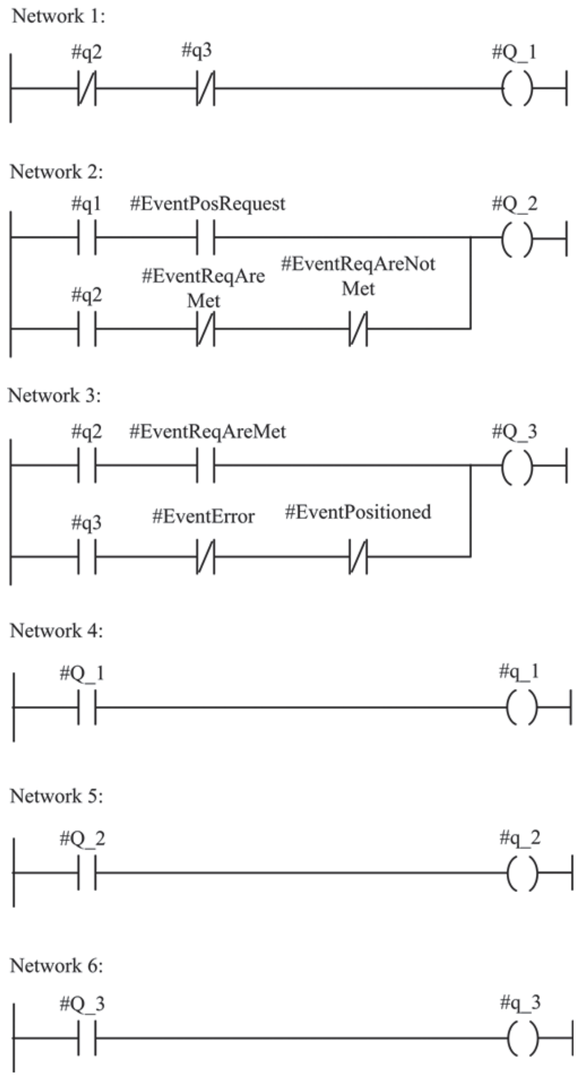

Figure 9. Part of the programmable logic controller program for the state functions of a switch element.

The automaton for the incoming requests is given in Figure 8, the states and their descriptions are given in Table 9, and the events are given in Table 10. The state-transition functions for the created automaton are given by

$$
\begin{aligned}
Q_1 &= \bar{q}_2\bar{q}_3\bar{q}_4\bar{q}_5\bar{q}_6 \\
Q_2 &= q_1 e_1 + q_2\bar{e}_2\bar{e}_3\bar{e}_{11} \\
Q_3 &= q_2 e_2 + q_3\bar{e}_4\bar{e}_5\bar{e}_{11} \\
Q_4 &= q_3 e_4 + q_4\bar{e}_6\bar{e}_7\bar{e}_{11} \\
Q_5 &= q_4 e_6 + q_6 e_{10} + q_5\bar{e}_8\bar{e}_9 \\
Q_6 &= q_5 e_8 + q_6\bar{e}_{10}\bar{e}_{11} \,.
\end{aligned}
\qquad (9)
$$

5.3 Implementation of the Interlocking System on a PLC

Since the hardware and software requirements should be satisfied according to defined safety standards in the implementation phase, on the hardware side, fail-safe programmable logic controllers are used. Therefore, a Siemens CPU317F-2 model fail-safe programmable logic controller was used in this study. In the fail-safe programming, it is allowed to use only a limited set of programmable-logic-controller commands. User-defined data types, or other complex data types, such as `REAL`, `ARRAY`, `BYTE`, are also mpy allowed to be used in the fail-safe program. Only basic data types, such as `WORD`, `INT`, `BOOL`, and `TIME` are allowed to be used. The fail-safe programmable-logic-controller program can be written only in ladder-diagram (LD) or function-block-diagram (FBD) languages. All of these limitations lead to difficulty in programming. However, with all these precautions, safe operation is guaranteed.

It is easy to implement the designed interlocking system on the programmable logic controller using the basic logical expressions if the state-transition functions are obtained. For instance, the state-transition functions of the switch element, the automaton for which was given in Figure 5, were obtained as in Equation (5). Implementation of these functions on the programmable logic controller using a ladder diagram is given in Figure 9.

In this program section, events of e_1, e_2, e_3, e_4, and e_5 are represented by the variable names `#EventPosRequest`, `#EventReqAreMet`, `#EventReqAreNotMet`, `#EventPositioned`, and `#EventError`, respectively. These events were also properly programmed by ladder logic. Figure 10 shows a part of the programmable-logic-controller program for the events e_4 (`#EventPositioned`) and e5 (`#EventError`).

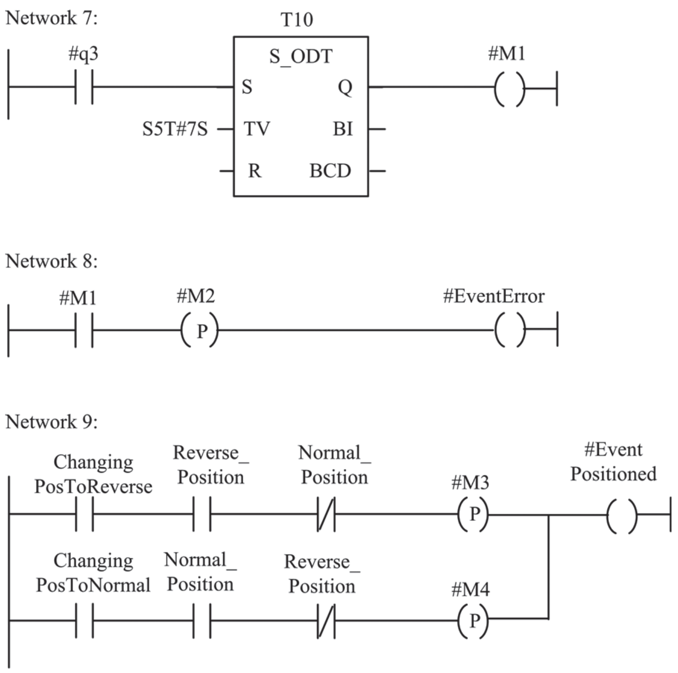

Figure 10. A program section producing event signals e_4 and e_5.

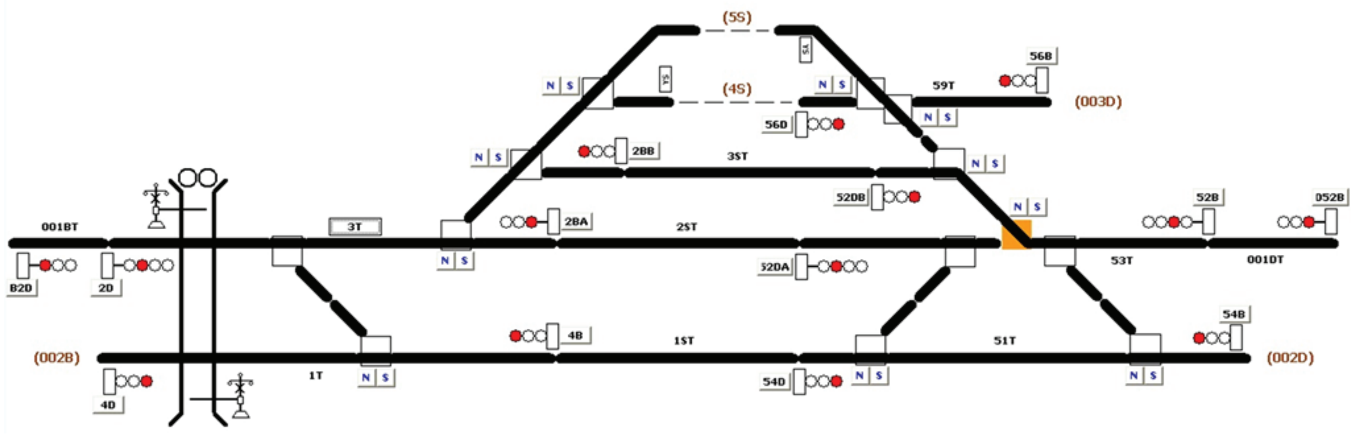

Figure 11. The supervisory control and data acquisition interface (SCADA) for the given railway station.

Table 8. The events for the level-crossing automaton.

Name of the Event	Event
e_1	Request to close level crossing
e_2	Indication (Level crossing is closed)
e_3	Timer has expired
e_4	Request to open level crossing
e_5	Indication (Level crossing is opened)
e_6	Timer has expired

Table 9. The states and their descriptions for the automaton that checks route requests.

State	Description
1	Initial state
2	Route request has arrived
3	All requirements are met
4	Switch is in position
5	Signal lights give correct notification
6	Train monitoring continues

Table 10. The events for the automaton that checks route requests.

Name of the Event	Event
e_1	Route request
e_2	Requirements are met
e_3	Requirements are not met
e_4	Correct switch indication
e_5	Request to change switch position has rejected
e_6	Correct signal light indication
e_7	Incorrect signal light indication
e_8	Train monitoring have not finished
e_9	Train monitoring finished
e_{10}	Route request is not cancelled
e_{11}	Route request is cancelled

5.4 Supervisory Control and Data Acquisition

The supervisory control and data acquisition (SCADA) system is designed for the railway simulator in order to monitor train positions, field equipment states and unoccupied-occupied routes, and to also manually control field equipment, such as switches. Figure 11 shows the supervisory control and data acquisition interface for the given railway station.

6. Conclusion

In this study, an automata-based railway signaling system was presented, and an interlocking system was designed for a railway station. In order to satisfy hardware and software requirements, a fail-safe programmable logic controller was used, and the programmable-logic-controller program of the designed interlocking system by automata was written in accordance with the fail-safe programming rules. Finally, all necessary tests were performed on the railway hardware simulator. It was shown that interlocking software worked without any problem. In addition, the designed supervisory control and data acquisition interface was taken advantage of during both the test and operation phases.

7. References

1. A. Svendsen, G. K. Olsen, J. Endresen, T. Moen, E. Carlson, K. J Alme, O. Haugen, "The Future of Train Signaling," *Lecture Notes in Computer Science*, 5301, 2008, pp. 128-142.

2. S. Hall, *Modern Signalling Handbook*, England, Ian Allan Publishing, 2008.

3. G. J. Kuepper, "150 Years of Train-Disasters – Practical Approaches for Emergency Responders," *9-1-1 Magazine* September/October, 1999, pp. 30-33.

4. Wikipedia, "Rail Accidents and Disasters," http://en.wikipedia.org/wiki/List_of_accidents_and_disasters_by_death_toll#Rail_accidents_and_disasters.

5. Turkish Railways, http://www.tcdd.gov.tr/Upload/Files/ContentFiles/2010/istatistik/tcddistatistik2005.pdf (in Turkish), 2010.

6. Turkish Railways, http://www.tcdd.gov.tr/Upload/ Files/ Content Files/2010/istatistik/2004_2008.pdf (in Turkish), 2010.

7. O. Eriş and İ. Mutlu, "Design of Signal Control Structures Using Formal Methods for Railway Interlocking Systems," The 11th International Conference on Control, Automation, Robotics and Vision, December 2010, Singapore.

8. Wikipedia, "Gare de Lyon Rail Accident," http://en.wikipedia.org/wiki/Gare_de_Lyon_train_accident.

9. "China Train Crash Caused by 'Signal Design Flaw'," http://www.guardian.co.uk/world/2011/jul/28/china-train-crash-signal-flaw, Jul 2011.

10. "India, Signal Error Could Have Caused Train Crash," http://www.toonaripost.com/2011/09/world-news/india-signal-error-could-have-caused-train-crash/, Sep 2011.

11. X. She, Y. Sha, Q. Chen, J. Yang, "The Application of Graph Theory on Railway Yard Interlocking Control System," *Proc. of the IEEE Intelligent Vehicles Symposium*, 2007, pp. 883-888.

12. M. Banci, A. Fantechi, and S. Ginesi, "The Role of Formal Methods in Developing a Distributed Railway Interlocking System," Proceedings of the 5th Symposium on Formal Methods for Automation and Safety in Railway and Automotive Systems, 2004, Braunschweig, Germany.

13. M. Banci, A. Fantechi, and S. Ginesi, "Some Experiences on Formal Specification of Railway Interlocking Systems using Statecharts," Train International Workshop at SEFM2005, 2005, Koblenz, Germany.

14. S. Türk, A. Sonat, A. Kuzu, M.T. Söylemez, Ö. Songüler, and T. Taralp, "Automated Interlocking Algorithm Generation from Interlocking Tables for Railway Signalization Systems," International Conference on Mechatronics, 2011, Istanbul, Turkey.

15. M. T. Söylemez, M. S. Durmuş, U. Yıldırım, S. Türk, and A. Sonat, "The Application of Automation Theory to Railway Signalization Systems: The Case of Turkish National Railway Signalization Project," 18th World Congress, IFAC'11, 2011, Milano, Italy.

16. U. Yıldırım, M. S. Durmuş, and M. T. Söylemez, "Fail-Safe Signalization and Interlocking Design for a Railway Yard: An Automation Petri Net Approach," The 7th International Symposium on Intelligent and Manufacturing Systems, 2010, Sarajevo, Bosnia Herzegovina.

17. A. Giua and C. Seatzu, "Modeling and Supervisory Control of Railway Networks Using Petri Nets," *IEEE Transactions on Automation Science and Engineering*, **5**, 2008, pp. 431-445.

18. A. M. Hagalisletto, J. Bjork, I. C. Yu, and P. Enger, "Constructing and Refining Large-Scale Railway Models Represented by Petri Nets," *IEEE Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews*, **37**, 2007, pp. 444-460.

19. M. S. Durmuş, U. Yıldırım, and M. T. Söylemez, "Signalization and Interlocking Design for a Railway Yard: A Supervisory Control Approach by Enabling Arcs," The 7th International Symposium on Intelligent and Manufacturing Systems, 2010, Sarajevo, Bosnia Herzegovina.

20. J. Palmer, "The Need for Train Detection," The 11th IET Professional Development Course on Railway Signalling and Control Systems, 2006, York, UK.

21. S. Kurtulan, PLC ile Endüstriyel Otomasyon (in Turkish), Birsen, İstanbul, 2010.

5.4 Supervisory Control and Data Acquisition

The supervisory control and data acquisition (SCADA) system is designed for the railway simulator in order to monitor train positions, field equipment states and unoccupied-occupied routes, and to also manually control field equipment, such as switches. Figure 11 shows the supervisory control and data acquisition interface for the given railway station.

6. Conclusion

In this study, an automata-based railway signaling system was presented, and an interlocking system was designed for a railway station. In order to satisfy hardware and software requirements, a fail-safe programmable logic controller was used, and the programmable-logic-controller program of the designed interlocking system by automata was written in accordance with the fail-safe programming rules. Finally, all necessary tests were performed on the railway hardware simulator. It was shown that interlocking software worked without any problem. In addition, the designed supervisory control and data acquisition interface was taken advantage of during both the test and operation phases.

7. References

1. A. Svendsen, G. K. Olsen, J. Endresen, T. Moen, E. Carlson, K. J Alme, O. Haugen, "The Future of Train Signaling," *Lecture Notes in Computer Science*, 5301, 2008, pp. 128-142.

2. S. Hall, *Modern Signalling Handbook*, England, Ian Allan Publishing, 2008.

3. G. J. Kuepper, "150 Years of Train-Disasters – Practical Approaches for Emergency Responders," *9-1-1 Magazine* September/October, 1999, pp. 30-33.

4. Wikipedia, "Rail Accidents and Disasters," http://en.wikipedia.org/wiki/List_of_accidents_and_disasters_by_death_toll#Rail_accidents_and_disasters.

5. Turkish Railways, http://www.tcdd.gov.tr/Upload/Files/ContentFiles/2010/istatistik/tcddistatistik2005.pdf (in Turkish), 2010.

6. Turkish Railways, http://www.tcdd.gov.tr/Upload/ Files/Content Files/2010/istatistik/2004_2008.pdf (in Turkish), 2010.

7. O. Eriş and İ. Mutlu, "Design of Signal Control Structures Using Formal Methods for Railway Interlocking Systems," The 11th International Conference on Control, Automation, Robotics and Vision, December 2010, Singapore.

8. Wikipedia, "Gare de Lyon Rail Accident," http://en.wikipedia.org/wiki/Gare_de_Lyon_train_accident.

9. "China Train Crash Caused by 'Signal Design Flaw'," http://www.guardian.co.uk/world/2011/jul/28/china-train-crash-signal-flaw, Jul 2011.

10. "India, Signal Error Could Have Caused Train Crash," http://www.toonaripost.com/2011/09/world-news/india-signal-error-could-have-caused-train-crash/, Sep 2011.

11. X. She, Y. Sha, Q. Chen, J. Yang, "The Application of Graph Theory on Railway Yard Interlocking Control System," *Proc. of the IEEE Intelligent Vehicles Symposium*, 2007, pp. 883-888.

12. M. Banci, A. Fantechi, and S. Ginesi, "The Role of Formal Methods in Developing a Distributed Railway Interlocking System," Proceedings of the 5th Symposium on Formal Methods for Automation and Safety in Railway and Automotive Systems, 2004, Braunschweig, Germany.

13. M. Banci, A. Fantechi, and S. Ginesi, "Some Experiences on Formal Specification of Railway Interlocking Systems using Statecharts," Train International Workshop at SEFM2005, 2005, Koblenz, Germany.

14. S. Türk, A. Sonat, A. Kuzu, M.T. Söylemez, Ö. Songüler, and T. Taralp, "Automated Interlocking Algorithm Generation from Interlocking Tables for Railway Signalization Systems," International Conference on Mechatronics, 2011, Istanbul, Turkey.

15. M. T. Söylemez, M. S. Durmuş, U. Yıldırım, S. Türk, and A. Sonat, "The Application of Automation Theory to Railway Signalization Systems: The Case of Turkish National Railway Signalization Project," 18th World Congress, IFAC'11, 2011, Milano, Italy.

16. U. Yıldırım, M. S. Durmuş, and M. T. Söylemez, "Fail-Safe Signalization and Interlocking Design for a Railway Yard: An Automation Petri Net Approach," The 7th International Symposium on Intelligent and Manufacturing Systems, 2010, Sarajevo, Bosnia Herzegovina.

17. A. Giua and C. Seatzu, "Modeling and Supervisory Control of Railway Networks Using Petri Nets," *IEEE Transactions on Automation Science and Engineering*, **5**, 2008, pp. 431-445.

18. A. M. Hagalisletto, J. Bjork, I. C. Yu, and P. Enger, "Constructing and Refining Large-Scale Railway Models Represented by Petri Nets," *IEEE Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews*, **37**, 2007, pp. 444-460.

19. M. S. Durmuş, U. Yıldırım, and M. T. Söylemez, "Signalization and Interlocking Design for a Railway Yard: A Supervisory Control Approach by Enabling Arcs," The 7th International Symposium on Intelligent and Manufacturing Systems, 2010, Sarajevo, Bosnia Herzegovina.

20. J. Palmer, "The Need for Train Detection," The 11th IET Professional Development Course on Railway Signalling and Control Systems, 2006, York, UK.

21. S. Kurtulan, PLC ile Endüstriyel Otomasyon (in Turkish), Birsen, İstanbul, 2010.