THE APPLICATION OF AUTOMATION THEORY TO RAILWAY SIGNALING SYSTEMS: THE TURKISH NATIONAL RAILWAY SIGNALING PROJECT

OTOMASYON TEORİSİNİN DEMİRYOLO SİNYALİZASYON SİSTEMLERİNE UYGULANMASI: ULUSAL DEMİRYOLO SİNYALİZASYON PROJESİ

Mustafa Seçkin DURMUŞ *, Uğur YILDIRIM *, Mehmet Turan SÖYLEMEZ *

1Control Engineering Department, Istanbul Technical University, TR-34469, Istanbul, Turkey.
durmustu@itu.edu.tr, yildirimu@itu.edu.tr, soylemez@itu.edu.tr

Received/Geliş Tarihi: 04.01.2013, Accepted /Kabul Tarihi: 28.02.2013
doi: 10.5505/pajes.2013.14633

Abstract

When compared with the other transportation systems, railway systems are economical, safer and more environment friendly. Despite all these features, investments on railways in Turkey remained quite restricted in comparison to other European countries. The initial construction costs of railways are high and developing its signaling system also requires a lot of attention. Signaling system is the vital part of a railway system which ensures safe travel and transportation on railways. One of the main reasons that make a signaling system critical and hard to develop is to satisfy the recommendations of related functional safety standards such as EN 50128 and EN 61508-3. In this study, functional safety requirements used in railway signaling systems are stated and the use of recommended methods in the Turkish National Railway Signaling Project (TNRSP) are explained. Especially, the software development process and the use of several programming techniques in determination of the software architectures are discussed.

Keywords: Railway signaling system, Railway interlocking system, Functional safety, Petri nets.

1 Introduction

Railways provide more economical and environmental solutions [1] even their initial costs are relatively high. Among different transportation alternatives, for a passenger who travels from Rome to Paris, traveling by train causes less carbon dioxide emission and energy consumption as illustrated in Figure 1. Researches also prove that the lifetime risk of death caused by railways is very low according to different transportation alternatives. For example in the UK, the risk of death for 100 million vehicles per kilometer for travelling by car is 6,780 while it is 520 in railways. Similarly, odds of dying in average year for travelling by airplane are 1 in 4,082,474 while it is 1 in 9,848,485 for railways [2]. As a result of these advantages and the rise in population, the demand on railways increases day by day.

![Typical emission and energy consumption](image1.png)

Figure 1. Typical emission and energy consumption [1].

Former railway systems where railway traffic and density were not too much as today did not need any signaling systems. Train drivers are warned about the obstruction in front of their way by railway guards who show red flag or mechanical interlocking. In order to eliminate all these problems, the first interlocking system was established in 1843 in UK [3]. The need of reliable and safe signaling system is much more today than in the past because sometimes failures may result in fatal accidents [4]. In addition to mechanical interlocking systems where the railway traffic operations were realized by signalboxes manually [3], railway control systems such as SMILE [5], STERNOL [6], ELEKTRA [7] and microprocessor-based systems [8] were also used. Many studies on modeling, design and verification of railway signaling and interlocking systems can be found in the literature [9]-[18].

Nowadays, as an alternative way, already certified COTS (Commercial off-the-shelf) solutions are also available in the market provided by several companies [19]-[24]. On the other hand, signaling systems are expected to satisfy safety related standards such as CENELEC (European Committee for Electrotechnical Standardization) standards in Europe. The European standard IEC 61508 [25] is developed for functional safety requirements of all kinds of electronic and programmable devices. In addition to the umbrella standard
IEC 61508, EN 50126 describes the functional safety requirements related with all kinds of railway applications where Reliability, Availability, Maintenance and Safety analysis (RAMS) are determined. EN 50128 (similar to EN 61508-3) determines methodologies for building software for railway control and applications and EN 50129 (similar to EN 61508-2) defines requirements for the hardware of electric, electronic and programmable devices. A well-known definition arises within these standards known as Safety Integrity Level (SIL). SIL is the probability for the system to execute the safety functions required in all specified input conditions within a specified time interval [26]. For railway interlocking systems, safety integrity level is expected to be at least SIL 3 [27], which will be explained in detail in section 3.

International agreements such as deployment of European Rail Traffic Management System (ERTMS), Turkish State Railways and so development of railway systems in Turkey accelerated. Construction of the railways compatible with ERTMS still continues and the Turkish Government has already planned new investments for future extensions. Within this perspective, developing original, reliable and fail-safe interlocking systems, which constitute one of the most vital parts of the railway signaling systems, is of great importance for Turkey.

The paper is organized as follows. Brief descriptions of railway components are given in the next section. Section 3 describes the determination of SIL and the software design process is explained in section 4. Software tests are explained in section 5 and finally, conclusions are given in section 6.

2 Components of Railway Signaling System

In this section, the definitions of the components that are used in the TNRSP are explained.

2.1 Traffic Command Center (TCC)

All operations on railways such as monitoring the railway traffic and situation of the railway field components (e.g. positions of switches, colors of signals) and scheduling of trains are in the responsibility of the TCC. All railway traffic is managed by the help of the officers (the dispatchers) who request routes for incoming and outgoing trains. In case of an emergency situation, the dispatchers can also talk directly with the train drivers.

2.2 The Interlocking System

The interlocking system can be considered as decision-making software because it checks the incoming requests from the TCC and compares these requests with the actual situation of the railway field equipments. An incoming request is accepted, if all safety criteria are met, or rejected if there are any conflicts or the request is improper. Instead of relay-based interlocking systems (where vital relays are used as interlocking fail-safe components) fail-safe Programmable Logic Controllers (PLCs) [28] and computer-based solutions [22] are already in use for railway signaling systems.

2.3 Signals

Railway signals (or optical wayside signals) are established along the rail line on specific points to inform the train driver about the occupation of the next railway block. Train drivers have to pay attention to the signals on the right side with respect to their direction of movement.

Similar to road traffic lights, red color means that the next block is occupied, so the train must stop. Yellow color means that the next block is free but not the block after that (i.e. the next signal is red, so the train should stop at the next signal). Green color means at least the next two blocks are free, so the train can proceed within speed limits. Unlike road signal lights, however, some railway signals can have four aspects especially near station areas. The fourth aspect, which is a bottom yellow color for the Turkish State Railways, designate a line change ahead (so the train should proceed with reduced speed on switch regions). On the station exits where there is always a line change ahead, dwarf signals are used instead of four aspect signals.

2.4 Switches (Point Machines)

Since railway vehicles do not have any steering mechanism like the road vehicles, they need switches in order to change track when maneuvering in a railway area. Switches have two location indications named normal and reverse. In case of a route request or a manual request from the TCC, switches are controlled by the interlocking system. An example schematic representation of a scissor crossover switch (or scissor crossing) is given in Figure 2.

2.5 Track Circuits

Track circuits are used to detect trains on railways. They inform the TCC about the absence or the presence of the train. A railway block may contain one or more track circuits that are connected in series depending on the length of the rail line. By the entrance of a train onto a railway block as given in Figure 3, the track circuit is short-circuited by the axles of the train. The track circuit then sends appropriate indication signals to the interlocking system.

2.6 Level Crossing

Level crossings are intersection points of road traffic and railway traffic. Normally when there is no route reservation or route request, the interlocking system send inactive signal to level crossing barriers to keep it open and allow road traffic. After a route reservation, inactive signal is ended by the interlocking system to close the barriers of the level crossing and to activate the flashing red lights of the level crossing. More explanations and related fail-safe Petri models of the level crossing can be found in [29].
3 Influence of Functional Safety Requirements on the Interlocking Software Development

In order to design a fail-safe interlocking system, railway related safety standards developed by CENELEC should be considered by the designers. While developing a railway application in accordance with the CENELEC standards, as a first step, required SIL have to be determined. The required SIL for a given system can be determined by using the Figure D.1, the Figure D.2 and the Table D.1 in EN 61508-5. SIL determination is realized by using different parameters such as consequence of risk (C), frequency and exposure time of risk (F), possibility of failing to avoid hazardous event (P) and probability of the unwanted occurrence of risk (W) [27].

The consequence risk parameter (C) can be chosen as C= level for the interlocking systems since system failures can result in death of several people. Similarly, the risk parameter (F) is chosen as frequent to permanent exposure in the hazardous zone (F0). Since avoiding from hazardous events are possible under certain conditions, possibility of falling to avoid hazard risk (P) is chosen as P3 and lastly, due to high probability of unwanted risk occurrence, probability of the unwanted occurrence of risk (W) is chosen as W3. By the help of these observations, it is possible to claim that the required SIL should be at least 3 for the interlocking systems. SIL determination process is given in Figure 4 and the relation between the expected range of failure per hour (λ) and SIL can be seen from Table 1 [27].

Table 1. Relationship between SIL, λ and MTTF.

<table>
<thead>
<tr>
<th>SIL Level</th>
<th>High Demand or Continuous Operation Mode (Range of λ)</th>
<th>Low Demand Operation Mode Range of MTTF (Mean Time to Failure)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>10^{-9} ≤ λ &lt; 10^{-8}</td>
<td>100000 ≥ MTTF ≥ 10000</td>
</tr>
<tr>
<td>3</td>
<td>10^{-8} ≤ λ &lt; 10^{-7}</td>
<td>10000 ≥ MTTF ≥ 1000</td>
</tr>
<tr>
<td>2</td>
<td>10^{-7} ≤ λ &lt; 10^{-6}</td>
<td>1000 ≥ MTTF ≥ 100</td>
</tr>
<tr>
<td>1</td>
<td>10^{-6} ≤ λ &lt; 10^{-5}</td>
<td>100 ≥ MTTF ≥ 10</td>
</tr>
</tbody>
</table>

Figure 4. SIL determination process for the interlocking software according to EN 61508-5 [27].

On high demand or continuous operation mode, the λ value for a SIL3 system has to be between 10^{-8} ≤ λ < 10^{-7}. Similarly on low demand mode of operation, the railway interlocking system is expected to work minimum 1000 years without falling into a dangerous failure state. In order to satisfy the values given in Table 1, development of hardware and software for the system need to be considered as a whole. Different architectures such as 1oo2, 2oo2, 2oo3 or parallel redundant structures can be used as a solution [30]. Instead of hardware development process, more effort was made on the software development and testing process in the TNRSP.

3.1 Interlocking Software Architecture

The EN 50128 safety standard proposes several techniques for forming the architecture of safety critical software. Defensive Programming (DefP), Failure Assertion Programming (FAP), and Diverse Programming (DivP) techniques have been used to achieve a SIL 3 software in the TNRSP [27].

The aim of DefP is to detect abnormal control flow, data flow or data values during their execution and react to these in a predetermined and acceptable manner [31]. In other words, designer has to put several extra control points to check the validity of results and/or variables of the program. For instance, if a railway block is occupied, the entrance signal of this block cannot be yellow or green. Similarly, movement of a switch must not be allowed when there is a train on the same railway block. Such so called general rules can be checked independently, regardless of the current state of the program, just before sending output signals to the railway field.

In FAP, the main idea is to check pre-conditions (checking the initial conditions for validity before execution of a command) and post-conditions (checking all results after the execution of a command) in the execution of safety critical commands. For example, position indication of a switch before and after a movement request should be checked to ensure correct functioning of the switch. Similarly, a signal cannot be red and green at the same time. If red and green color indications are received from a signal at the same time, this must be considered as a failure by the interlocking software.

Finally, the aim of DivP is to detect and mask software design faults during execution of a program in order to prevent safety critical failures of the system and continue operation with high reliability [31]. In this technique, at least two independent groups develop the software with the same specifications using different techniques. The outputs of algorithms provided by separate groups are compared online to make a safety critical decision. For a safety critical signal (e.g. a position change command for a particular switch) to be sent to the railway field, all algorithms must agree. If a disagreement occurs then the system remains in the predetermined safe-state (the switch is not allowed to be moved). The architecture of the interlocking system is given in Figure 5.

Figure 5. The architecture of the interlocking system.

Two independent groups from Istanbul Technical University (ITU) have developed interlocking software using automatons and Petri nets in the TNRSP. Both of these algorithms are expected to be used for diverse programming purposes. Although it increases safety integrity level of a program, the main disadvantage with diverse programming is that the system can fall into safe state unexpectedly (causing operation distractions), when the algorithms do not produce the same
results at the same time. Possible ways to overcome this shortcoming is documenting software specifications very dearly and using a careful synchronization between controllers.

4 Interlocking Software Development Process

The steps of the software development process used in the TNRSP are explained in this section. For this purpose, a recommended software development model named V-model is used (see Figure 6).

![Figure 6. The V-Model [31].](image)

As it is obvious from the Figure 6, as an initial step, the designer should obtain system specifications along with the software specifications. Actually, forming an interlocking table is the first formal step towards modeling an interlocking system. Software specifications are defined by Turkish State Railways. An example railway yard and a part of its interlocking table are given in Figure 7.

![Figure 7. The interlocking table, (G-Green, Y-Yellow, R-Red, YY-Yellow-Yellow, YR-Yellow-Red).](image)

Some of the possible route requests (e.g. 001DT-1ST, 001DT-2ST), related signal colors with these requests and the required conditions of the other railway field components (to be able to reserve the route) are defined in this table. For example, if route 001DT-2ST is requested switches 51, 53 and 55 have to be in normal position, signals 52DB, 52DA have to be red and entrance signal of this route 52B can be green (if the next signal 2BA is yellow), yellow (if the next signal 2BA is red) or yellow-red (if the block 2ST is occupied by another train).

The second step in the V-model is to choose the appropriate combination of the software architectures. The chosen architecture was described in section 3.1.

After this process, related system sub models (the modules) have to be obtained. In order to maximize the diversity, the components of the railway signaling system explained in section 2 are modeled separately by two different workgroups using two different methods: Automatons [32] and Petri nets [33]. Both methods are mentioned in IEC 61508 and EN 50128 safety standards as semi-formal modeling tools and highly recommended to obtain SIL 3 software.

error tracking, makes the verification and validation stages much easier and provides flexible programming. Therefore the whole railway field is modeled as a combination of parallel working sub-models. The Petri net models will be given in section 4.2. Additionally, for more information about automata models, the reader is referred to [41] and [42].

4.1 Petri Nets

Petri nets are defined in the literature by [33].

\[
PN = P, T, F, W, M_0
\]

where

- \( P \) is a set of places, \( P = \{ P_1, P_2, ..., P_n \} \),
- \( T \) is a set of transitions, \( T = \{ t_1, t_2, ..., t_m \} \),
- \( F \subseteq (P \times T) \cup (T \times P) \) is a set of arcs,
- \( W : F \rightarrow \{ 1, 2, 3, ... \} \) is a weight function,
- \( M_0 : P \rightarrow \{ 0, 1, 2, 3, ... \} \) is the initial marking,
- \( P \cap T = \emptyset \) and \( P \cup T \neq \emptyset \).

4.1.1 Basic Properties of Petri nets

- A transition \( t \) is said to be enabled if each input place \( P \) of \( t \) is marked at least \( W \{ P, t \} \) tokens, where \( W \{ P, t \} \) is the weight of arc from place \( P \) to transition \( t \).

\[
x(P) \geq W(P,t)\quad \text{for all } P \in I(t_j)
\]

where \( x(P) \) is the number of tokens on ith place and \( I(t_j) \) is the set of input places of transition \( t_j \).

- An enabled transition may or may not fire (depending on whether or not the event actually takes place).

- A firing of an enabled transition \( t \) removes \( W(P,t) \) tokens from each input place \( P \) of \( t \) and adds \( W(t,P) \) tokens to each output place \( P \) of \( t \), where \( W(t,P) \) is the weight of the arc from \( t \) to \( P \).

\[
x'(P) \geq x(P) - W(P,t) + W(t,P)
\]

where \( x'(P) \) is the number of tokens on ith place after the firing of transition \( t \). Petri nets can also be represented in matrix form known as incidence matrix.

\[
A = \begin{bmatrix} a_{ij} \end{bmatrix} \Rightarrow a_{ij} = a_{ij}^+ - a_{ij}^- = W(P,j,t) + W(t,P)
\]

In this matrix form, the new marking after a transition is calculated as follows:

\[
M_k = M_{k-1} + A^T u_k
\]

where \( u_k \) is the kth firing vector, \( M_k \) is the new marking which is an \( n \times 1 \) column vector.

A PN is said to be pure if it has no self-loops and a PN is said to be bounded if all of its arc weights are 1. For more details about PNs and its specification, features and types, [33] can be seen.

4.2 Modeling of the Railway Field Components

Basic Petri net models of the railway field components are given in the following figures from Figure 8 to Figure 13 and definitions of the Petri net models can be found in Table 2 and Table 3. All Petri net models given in figures are safe (or 1-bounded) Petri nets since there exists one, and only one, token available in the Petri net at any given time.

Since the routes given in Figure 7 intersect with each other, only one route reservation can be made in order to prevent...
collisions. More than one route reservation can be made when there is no intersection between the routes. At the beginning, switches are assumed to be on initial position ($P_0$). After an incoming position command or a route request from TCC, switch moves to the desired position (normal or reverse) and stays there until a new position command or route request is received (the interlocking software locks the switch electronically). If the switch did not reach to desired position in a predetermined time, which is assumed as 7 sec for Turkish State Railways, it is considered as an error and the token moves to place $P_6$. Similar error states are also taken into account for the other component models but due to space restrictions, only the simplified models are given here.

![Figure 8. Petri net model for the route reservations given by the interlocking table in Figure 6.](image)

![Figure 9. Petri net model for the switch.](image)

![Figure 10. Petri net model for the three aspect tall signal.](image)

![Figure 11. Petri net model for the four aspect tall signal.](image)

![Figure 12. Petri net model for the three aspect dwarf signal.](image)

![Figure 13. Petri net model for the track circuits.](image)

Railway track circuits are modeled just single places with one input and one output because only one train can occupy a track circuit at the same time. If a train enters onto a track circuit than related transition is fired and a token put on its related place. In addition to this, while train is moving on a reserved route, the interlocking system controls the movement of trains on to track circuits which also allows detecting unexpected occupation of track circuits. More explanations on modeling of the railway field components can be found in [39]-[42]. Please note that, the Petri net models given in this paper only explains the basic operation principles of the related railway field components.

Depending on the route reservation and the occupation of the next block, signal colors are adjusted to inform the train drivers. If a train passes a signal when it is red, then the train is stopped by Automatic Train Stop (ATS) or Automatic Train Protection (ATP) systems, which are out of the scope of this paper.

<table>
<thead>
<tr>
<th>Place</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_1$</td>
<td>No reservation for the railway yard</td>
</tr>
<tr>
<td>$P_2$</td>
<td>001DT - 1ST route is reserved</td>
</tr>
<tr>
<td>$P_3$</td>
<td>001DT - 2ST route is reserved</td>
</tr>
<tr>
<td>$P_4$</td>
<td>001DT - 3ST route is reserved</td>
</tr>
<tr>
<td>$P_5$</td>
<td>Switch starting position</td>
</tr>
<tr>
<td>$P_6$</td>
<td>Switch is moving from Normal to Reverse position</td>
</tr>
<tr>
<td>$P_7$</td>
<td>Switch is moving from Reverse to Normal position</td>
</tr>
<tr>
<td>$P_8$</td>
<td>Switch is on Reverse position</td>
</tr>
<tr>
<td>$P_9$</td>
<td>Switch is on Normal position</td>
</tr>
<tr>
<td>$P_{10}$</td>
<td>Signal is Red</td>
</tr>
<tr>
<td>$P_{11}$</td>
<td>Signal is Yellow</td>
</tr>
<tr>
<td>$P_{12}$</td>
<td>Signal is Green</td>
</tr>
<tr>
<td>$P_{13}$</td>
<td>Signal is Yellow-Yellow</td>
</tr>
<tr>
<td>$P_{14}$</td>
<td>Signal is Yellow-Red</td>
</tr>
<tr>
<td>$P_{15}$</td>
<td>Signal is Yellow-Green</td>
</tr>
<tr>
<td>$P_{16}$</td>
<td>Signal is Red</td>
</tr>
<tr>
<td>$P_{17}$</td>
<td>Signal is Yellow</td>
</tr>
<tr>
<td>$P_{18}$</td>
<td>Signal is Green</td>
</tr>
<tr>
<td>$P_{19}$</td>
<td>Signal is Red</td>
</tr>
<tr>
<td>$P_{20}$</td>
<td>Signal is Yellow</td>
</tr>
<tr>
<td>$P_{21}$</td>
<td>Signal is Green</td>
</tr>
<tr>
<td>$P_{22}$</td>
<td>Signal is Yellow-Red</td>
</tr>
<tr>
<td>$P_{23}$</td>
<td>Track circuit 1ST</td>
</tr>
<tr>
<td>$P_{24}$</td>
<td>Track circuit 2ST</td>
</tr>
<tr>
<td>$P_{25}$</td>
<td>Track circuit 3ST</td>
</tr>
<tr>
<td>$P_{26}$</td>
<td>Track circuit 51T</td>
</tr>
<tr>
<td>$P_{27}$</td>
<td>Track circuit 53T</td>
</tr>
<tr>
<td>$P_{28}$</td>
<td>Track circuit 001DT</td>
</tr>
<tr>
<td>$P_e$</td>
<td>Error State</td>
</tr>
</tbody>
</table>
Table 3. Definition of the transitions given in Figure 8 to Figure 13.

<table>
<thead>
<tr>
<th>Transition</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_1 (t_3, t_5) )</td>
<td>Route 001DT - 1ST (001DT - 2ST, 001DT - 3ST) is reserved</td>
</tr>
<tr>
<td>( t_2 (t_4, t_6) )</td>
<td>Route 001DT - 1ST (001DT - 2ST, 001DT - 3ST) is released</td>
</tr>
<tr>
<td>( t_7 (t_9) )</td>
<td>Normal (Reverse) position request for switch</td>
</tr>
<tr>
<td>( t_9 (t_{10}) )</td>
<td>Switch arrives on Reverse (Normal) position</td>
</tr>
<tr>
<td>( t_{11} (t_{12}) )</td>
<td>Switch is moving from Reverse (Normal) to Normal (Reverse) position</td>
</tr>
<tr>
<td>( t_{13} (t_{15}, t_{16}, t_{18}, t_{21}) )</td>
<td>Yellow (Green, Yellow-Red, Yellow-Green) color request for signal</td>
</tr>
<tr>
<td>( t_{14} (t_{16}, t_{18}, t_{20}, t_{22}) )</td>
<td>Red color request for signal</td>
</tr>
<tr>
<td>( t_{22} (t_{23}) )</td>
<td>Yellow (Green) color request for signal</td>
</tr>
<tr>
<td>( t_{24} (t_{26}) )</td>
<td>Red color request for signal</td>
</tr>
<tr>
<td>( t_{25} (t_{26}, t_{27}, t_{28}, t_{29}) )</td>
<td>Yellow (Green, Yellow-Red) color request for signal</td>
</tr>
<tr>
<td>( t_{26} (t_{28}) )</td>
<td>Red color request for signal</td>
</tr>
<tr>
<td>( t_{27} (t_{28}, t_{31}) )</td>
<td>Track circuit 1ST (2ST, 3ST, 51T, 53T, 001DT) is occupied</td>
</tr>
<tr>
<td>( t_{32} (t_{25}, t_{27}, t_{28}, t_{42}, t_{43}) )</td>
<td>Track circuit 1ST (2ST, 3ST, 51T, 53T, 001DT) is unoccupied</td>
</tr>
<tr>
<td>( t_{44} )</td>
<td>Switch indication error</td>
</tr>
</tbody>
</table>

4.3 Generation of PLC Codes from Models

Converting the Petri net models to a useful PLC code is simpler by using Sequential Function Charts (SFCs) which is one of the five languages defined by IEC 61131-3 standard. Instead of widely used techniques like Ladder Diagrams (LD) or Function Block Diagrams (FBD), SFC is directly related with safe Petri nets and automatons. Besides, several formal conversion techniques are also available in the literature to convert Petri nets to PLC codes [43]-[46]. A snapshot of a part of a SFC code is given in Figure 14.

4.4 An Example (Route Reservation Procedure)

Assume that route request for 001DT-3ST is made from TCC. If there is no route reservation interacting with the route 001DT-3ST or existence of another conflicting condition, the interlocking software will accept the request. In this case, in the first step, transition \( t_5 \), which moves the token on place \( P_1 \) to place \( P_4 \), will be fired. In the second step, the switches related with route 001DT-3ST have to be arranged in accordance with the interlocking table given on Figure 7. Switches 51 and 55 have to be on normal position, whereas switches 53 and 57 have to be on reverse position. If all these conditions are satisfied, the interlocking software sends related signal (which is 52B for this route) appropriate color information given on the interlocking table (signal 52B is assumed to be yellow for this example). Finally, the train movement is monitored via track circuits by the interlocking system. After route reservation is made, trains have to pass the track circuits one by one in a predetermined order. The block diagram of this example is given in Figure 15.

Figure 15. Route reservation procedure (Y-Yes, N-No).

5 Software Tests

Testing interlocking software is at least as important as developing it. The standards (EN 61508-3 and EN 50128) provide several guidelines for testing such crucial software. Testing of the interlocking software is achieved in five steps.

5.1 Interlocking Software Module Tests (Verification)

Module tests (testing program modules separately as mentioned in V-model given in Figure 6.) are realized by using the Interlocking Test Program (ITP) developed by the Department of Control Engineering of Istanbul Technical University (ITU). In this program, a basic simulator is used to imitate the signals in the field. An automatic testing procedure (2-3 weeks) was applied to each interlocking module.

5.2 Interlocking Software Simulator Tests (Validation)

Validation was realized by using a complex software simulator which is connected to the interlocking software. Simulator was connected to the interlocking software through a second PLC set using digital I/O modules, and all possible signals in the field are imitated by this second PLC. Complex train movements with several stress tests can be performed by the software simulator [47].

5.3 Hardware Simulator Tests (Commissioning)

A hardware simulator, which is 1:87 scaled model of the real railway field, is used to provide an even more realistic simulation. The signals collected from the model field are delivered to the interlocking system using a set of PLCs in this simulation model [48]. The hardware simulator was...
established in the Industrial Automation Laboratory of the Control Engineering Department of ITU.

5.4 Factory Acceptance Tests (FAT)
These tests are also realized in the Industrial Automation Laboratory of the Control Engineering Department of ITU, under the observation of a group of delegates from Turkish State Railways. The whole system (including the interlocking and TCC software, which was developed by TÜBİTAK-BİLGEM) has been validated in these tests.

5.5 Field Acceptance Tests
The final field acceptance tests are realized by Turkish State Railways in Mithatpaşa railway station in Sakarya, Turkey in January, 2012. All the tests are passed successfully. It should also be noted that the tests are executed by independent groups and documented carefully as suggested by the CENELEC standards.

6 Conclusion
In this study, the development of a railway interlocking system was explained. The railway related functional safety requirements of CENELEC standards, especially EN 61508 and EN 50128, are also considered. The voting system, which is a part of the developed interlocking system, consists of two fail-safe PLCs with the running codes determined by using Automata and Petri net methods regarded as semi-formal methods by EN 50128. Communication between the Traffic Control Center, the railway field and the interlocking PLCs are achieved by a master controller where all decisions are compared, considered and executed. Obtained interlocking software is tested and verified by both using a simulator called Interlocking Test Program (ITP) and a small-sized hardware simulator of railway yard established in Istanbul Technical University Industrial Automation Laboratory.

Automatic construction of the interlocking tables, automatic generation of supervisors for the control of railway field component models, automatic PLC code generation for the interlocking software that is fully compatible with European Rail Traffic Management System (ERTMS) Level 1-3 issues can be classified as possible future works.

7 Acknowledgement
This work is supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) project number 108G186 – The National Railway Signaling Project.

8 References


