VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI - 590018

"Auxiliary Relief Panel"

This is submitted in partial fulfillment of the curriculum prescribed for the award of the degree of Bachelor of Engineering in Computer Science & Engineering by

1CR14CS001 A Sri Bhagya Lakshmi

Under the Guidance of

Mrs. Daminderjit Sunner
Assistant Professor
Department of CSE, CMRIT, Bengaluru

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
#132, AECS LAYOUT, IT PARK ROAD, BENGALURU - 560037

2017-18

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI - 590018

Certificate

This is to certify that the project entitled "**Auxiliary Relief Panel**" is a bonafide work carried out by **A Sri Bhagya Lakshmi** in partial fulfillment of the award of the degree of Bachelor of Engineering in **Computer Science & Engineering** of Visvesvaraya Technological University, Belgaum, during the year 2017-18. It is certified that all corrections / suggestions indicated during reviews have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

-------------------------	-------------------------	-------------------------
Signature of Guide	Signature of HoD	Signature of Principal
Mrs.Daminderjit Sunner	**Dr. Jhansi Rani P**	**Dr. Sanjay Jain**
Assistant Professor	Professor & Head	Principal
Department of CSE	Department of CSE	CMRIT,
CMRIT, Bengaluru - 37	CMRIT, Bengaluru - 37	Bengaluru - 37

External Viva

	Name of the Examiners	Institution	Signature with Date
1.	-------------------------	----------------------------	----------------------------
2.	-------------------------	----------------------------	----------------------------

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI - 590018

Declaration

We **Ms.A Sri Bhagya Lakshmi**, bonafide students of CMR Institute of Technology, Bangalore, hereby declare that the dissertation entitled "**Auxiliary Relief Panel**" has been carried out by us under the guidance of **Mrs. Daminderjit Sunner**,Assistant Professor of CSE Department, CMRIT, Bangalore, in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in **Computer Science Engineering**, of the Visvesvaraya Technological University, Belgaum during the academic year 2017-2018.The work done in this dissertation report is original and it has not been submitted for any other degree in any University.

A Sri Bhagya Lakshmi

Acknowledgement

The satisfaction and euphoria that accompany a successful completion of any task would be incomplete without mentioning people who made it possible, success is the epitome of hard work and perseverance, but steadfast of all is encouraging guidance.

So,with gratitude we acknowledge all those whose guidance and encouragement served as beacon of light and crowned our effort with success.

We would like to thank **Dr. Sanjay Jain**, Principal, CMRIT, Bangalore for providing excellent academic environment in the college and his never-ending support for the B.E program.

We would like to thank **Dr. Jhansi Rani**, Professor & HOD, Department of Computer Science, CMRIT, Bangalore who shared their opinions and experiences through which we received the required information crucial for the project.

We would also like to thank **Mr. Sudhakar K N** , Associate Professor & Project Coordinator, Department of Computer Science, CMRIT, Bangalore who shared his opinions and experiences through which we received the required information crucial for the project.

We consider it a privilege and honour to express my sincere gratitude to my internal guide **Mrs. Daminderjit Sunner**, Assistant Professor, Department of Computer Science & Engineering, for his valuable guidance throughout the tenure of this project work.

We would also like to thank all the faculty members who have always been very co-operative and generous. Conclusively, we also thank all the non-teaching staff and all others who have done immense help directly or indirectly during my project

A Sri Bhagya Lakshmi

Table of Contents

List of Figures

List of Tables

Abstract

One of the major cause for railway accidents, delay in train arrival time and mishaps is due to failure of Interlocking System or absence of an alternate solution or system to replace the original system onsite before any major accidents takes place when the system fails to function.

Hence Auxiliary Relief Panel / Emergency Panel (ARP/EP) shall be used as standby interlocking for small and big stations as the failure management mechanism. The failure situations are ranging from equipment failure up to the building damage. Based on the situation and the requirement of Indian Railways, ARP/EP is designed.

The ARP/EP (Auxiliary Relief Panel / Emergency Panel) is a microprocessor based equipment used for the operation of points, signals, level crossing gates, track circuits for manual operation of route setting and route releasing if Electronic Interlocking System (EI) is failed. ARP shall have limited buttons control for route setting (only main line in fixed configuration) and releasing. It shall be capable of detecting the EI failure from standard contact (Relay contact which shall indicate failure of EI), record for health monitoring purpose.

Hence, Auxiliary Relief Panel controls the train routes and prevents from rail accidents.

Chapter 1

PREAMBLE

1.1 Introduction

Indian Railways (IR) is India's national railway system operated by the Ministry of Railways. It manages the fourth-largest railway network in the world by size, with 121,407 kilometres (75,439 mi) of total track over a 67,368-kilometre (41,861 mi) route. Forty nine percent of the routes are electrified with 25 KV AC electric traction while thirty three percent of them are double or multi-tracked. IR runs more than 13,000 passenger trains daily, on both long-distance and suburban routes, from 7,349 stations across India.

Railway signalling is a system used to direct railway traffic and keep trains clear of each other at all times. In railway signalling, an interlocking is an arrangement of signal apparatus that prevents conflicting movements through an arrangement of tracks such as junctions or crossings

In the early days of the railways, signalmen were responsible for ensuring any points (US: switches) were set correctly before allowing a train to proceed. Mistakes, however, led to accidents, sometimes with fatalities. The concept of the interlocking of points, signals and other appliances was introduced to improve safety. This prevents a signalman from operating appliances in an unsafe sequence, such as clearing a signal while one or more sets of points are not set correctly for the route. Early interlocking systems used mechanical devices both to operate the signalling appliances and to ensure their safe operation. Beginning around the 1930s, electrical relay interlocking were used. Since the late 1980s, new interlocking systems have tended to be of the electronic variety. Nowadays, Indian Railways use Electronic Interlocking (EI) system which is the main component for every station used to control the movements of train. In Electronic Interlocking, the points and signals are worked from one integrated mechanism in a signal cabin which features a display of the entire track layout with indications of sections that are occupied, free, set for reception or dispatch, etc. The interlocking is accomplished not by mechanical devices but by electrical circuitry – relays and switches in older electrical or electro pneumatic systems, and computerized circuits in the newer electronic systems.

But badly side of Indian Railways are Railway accidents. Over 290 accidents have occurred till date. Railway accidents may be classified by their effects, e.g.: head-on collisions, rear-end collisions, side collisions, derailments, fires, explosions, wrong-side failure etc. They may alternatively be classified by cause, e.g.: driver and signalman error; mechanical failure of rolling stock, tracks and bridges. A wrong-side failure describes a failure condition in a piece of railway signalling equipment that results in an unsafe state. A typical example would be a signal showing a 'proceed' aspect (e.g. green) when it should be showing a 'stop' or 'danger' aspect, resulting in a "false clear". (The converse is a right-side failure, where even with any reduction the resulting state is safe overall.).

1.2 Problem Statement

In case of Failure in Interlocking system,
→ It affects the normal rail traffic movements.
→ Interlocking equipment failure/equipment room Disaster during non-Interlock period.
→ Impact of which is over utilization of Railway Staffs in re- installation of equipment within a short duration, delay in the entire traffic network.
To overcome the problem, standby requirement is mandatory. A Successful Solution would be to provide a stopgap interlocking with ARP/EP.

1.3 Proposed System

Auxiliary Relief Panel / Emergency Panel (ARP/EP) shall be used as standby interlocking for small and big stations as the failure management mechanism. The failure situations are ranging from equipment failure up to the building damage. Based on the situation and the requirement of Indian Railways, ARP/EP is designed.

The ARP/EP (Auxiliary Relief Panel / Emergency Panel) is a microprocessor based equipment used for the operation of points, signals, level crossing gates, track circuits for manual operation of route setting and route releasing if Electronic Interlocking System (EI) is failed. ARP shall have limited buttons control for route setting (only main line in fixed configuration) and releasing. It shall be capable of detecting the EI failure from standard contact (Relay contact which shall indicate failure of EI), record for health monitoring purpose. Failure shall be confirmed by standard contact pickup (Fail Relay indication from EI) and manual button pickup (EI Fail button from panel. System shall have Serial / Ethernet Ports for future purpose. This system facilitates limited seamless operation of train movement in case of EI failure. It will be in idle mode by default and shall be active only when EI failure is confirmed and shall be put back to idle condition once EI is restored.

1.4 Advantages

1. No copper cable required
2. Can be applied for small and big yards
3. Customized application logic can be stored in memory and can be used as applicable.
4. Standalone and easy to install.
5. IOs are configurable hence depending on the size of the Station number of interface signals can be added in the system configuration.
6. As the unit is modular it can be easily transported to the required location.
7. All the components of the system are off the shelf electronic components hence service and maintenance of the interface units is simple and cost effective.

1.5 Phase Description

Phase	**Task**	**Description**
Phase1	Analysis	Analyzing the core of the IEEE paper and provide Literature review based on analysis.
Phase2	Literature Survey	Collect raw data and elaborate on literature surveys.
Phase3	System Description	Gives the details of the complete system.
Phase4	System Design	Object designing and Functional description .
Phase5	Implementation	Implement the code based on the object specification .
Phase6	Testing	Test the project according to Test Specification .
Phase7	Documentation	Prepare the document for this project with conclusion and future enhancement.

Table 1.1: Phase Description

1.6 Organization of the project report

The project report is organized as follows:

Chapter 2: **Literature Review** - Gives a brief overview of the survey papers and the research sources that have been studied to establish a thorough understanding of the project under consideration.

Chapter 3: **Theoretical Background** - Establishes groundwork for the proposed project by giving a detailed analysis of the project topic, existing research relevant to the project, arguments in favor and against the existing solutions and finally explores the motivation behind the proposed solution.

Chapter 4:**System Requirement Specification** - Discusses in details about the different kinds of requirements needed to successfully complete the project.

Chapter 5: **System Description** - Gives the details of the complete system.

Chapter 6: **System Design** - Gives the design description of the project, conceptual and detailed design well supported with design diagrams.

Chapter 7: **Implementation** - Discusses the implementation details of the project and reasons the use of the programming language and development environment.

Chapter 8: **Testing** - Briefs the testing methods used for testing the different modules in the project.

Chapter 9: **Results and Performance Analysis** - Gives the snapshots and graphs of the proposed protocols.

Chapter 10: **Conclusion and Future Scope** - Gives the concluding remarks of the project, throwing light on its future aspects.

Chapter 2

LITERATURE SURVEY

Literature survey is mainly carried out in order to analyze the background of the current project which helps to find out flaws in the existing system and guides on which unsolved problems we can work out. So, the following topics not only illustrate the background of the project but also uncover the problems and flaws which motivated to propose solutions and work on this project.

2.1 LITERATURE SURVEY

Literature survey is the documentation of a comprehensive review of the published and unpublished work from secondary sources data in the areas of specic interest to the researcher. The library is a rich storage base for secondary data and researchers used to spend several weeks and sometimes months going through books, journals, newspapers, magazines, conference proceedings, doctoral dissertations, masters theses, government publications and nancial reports to nd information on their research topic. Reviewing the literature on the topic area at this time helps the researcher to focus further interviews more meaningfully on certain aspects found to be important is the published studies even if these had not surfaced during the earlier questioning .So the literature survey is important for gathering the secondary data for the research which might be proved very helpful in the research. The literature survey can be conducted for several reasons.

[1] Muhammed Ali Nur Oz and Ibrahim Sener in their paper "A Tool for Automatic Formal Modeling of Railway Interlocking Systems"

Muhammed Ali Nur Oz and Ibrahim Sener in their paper A Tool for Automatic Formal Modeling of Railway Interlocking Systems provides information about interlocking system and its uses and view of signaling components and their role in interlocking system. It also provides a new approach to graphical user interface where user can easily create and modify a single line diagram of the station without the need for an additional tool. This interface is composed of three main parts; a toolbox, an information panel and a layout editor. The toolbox includes blocking blocks of railway systems, which represent station topology components such as tracks, un-signaled tracks, 3-connection points and 4-connection points. The Information panel is used to notify the user on how to use the interface and to warn the user of unauthorized actions. The layout editor is the third part of the interface, where the single line diagram is graphically created.

[2] G Pavan Kumar, IRSSE, Director(Signal), RDSO in their article "Fail-Safety Requirement for Railway Signalling Equipment"

G Pavan Kumar, IRSSE, Director(Signal), RDSO in their article Fail-Safety Requirement for Railway Signalling Equipment defines requirements for acceptance and approval of safety related electronic systems in the Railway Signalling field. Safety related electronic systems for signalling include hardware and software aspects. This article is concerned with the evidence to be presented for the acceptance of safety related systems, it specifies those life-cycle activities which shall be completed before the acceptance stage, followed by additional planned activities to be carried out after the acceptance stage. Safety justification for the whole of the life-cycle is therefore required.

Chapter 3

THEORETICAL BACKGROUND

Theoretical background highlighting some topics related to project work. The description contains several topics which are worth to discuss and also highlight some of their limitation that encourage going on finding solution as well as highlights some of their advantages for which reason these topics and their features are used in this project.

3.1 Electronic Interlocking

In railway signalling, an interlocking is an arrangement of signal apparatus that prevents conflicting movements through an arrangement of tracks such as junctions or crossings. The signalling appliances and tracks are sometimes collectively referred to as an interlocking plant. An interlocking is designed so that it is impossible to display a signal to proceed unless the route to be used is proven safe.

A minimal interlocking consists of signals, but usually includes additional appliances such as points (switches) and derails, and may include crossings at grade and movable bridges. Some of the fundamental principles of interlocking include:
→ Signals may not be operated to permit conflicting train movements to take place at the same time on set route.
→ Switches and other appliances in the route must be properly 'set' (in position) before a signal may allow train movements to enter that route.
→ Once a route is set and a train is given a signal to proceed over that route, all switches and other movable appliances in the route are locked in position until either.
→ The train passes out of the portion of the route affected, or
→ The signal to proceed is withdrawn and sufficient time has passed to ensure that a train approaching that route has opportunity to come to a stop before passing the signal.

3.2 Point Machines

Point machines have an electric motor and gears to convert the rotational motion of the motor into the linear motion required to switch the points. The gear assembly also provides required transmission ratio so that it can generate necessary force to move switch blades. The machine performs following functions:
→ Moving switch blades.
→ Locking the blades.

→ Detection and proving the position of blades.

Figure 3.1: Point Machine

3.3 Signals

Signalling is one of the most important parts of the many components which make up a railway system. Train movement safety depends on it and the control and efficient management of trains depends on them.
Over the years many signalling and train control systems have been evolved. The journey started with very simple systems such as simple coloured flags and semaphore arms to that today a highly technical and complex electrical and electronic systems.

Why Signalling is required?
The signalling has the following basic functions:-
→ Arranging safe reception and dispatch of trains onto required lines at stations.
→ Ensuring that trains are not received on occupied lines.
→ Ensuring that two trains do not enter the same part of the track between two stations (Also called block signalling)
→ Optimizing the utilization of track and other assets by allowing the dealing of maximum number of trains at highest speed permitted by track and train vehicles safely.
→ Ensuring that no part of the train is left over in Block section between two stations.

Achieve all the above in a manner called Fail safe which makes signalling a unique field of Engineering as every component and particularly the combination shall not fail to an unsafe end result at any cost.

Figure 3.2: Railway Signals

3.3.1 Track Circuit

A track circuit is an electrical circuit that uses the rails as a conductor. It consists of an electrical source and a regulating resistance at one end and a low resistance safety relay at the other end. Its purpose is to prove the absence of a train from a section of line so that points may be operated and signals cleared to show a proceed aspect. It also provides an indication to the signalman as to the position of a train or vehicles on each section of line where track circuits are fitted.

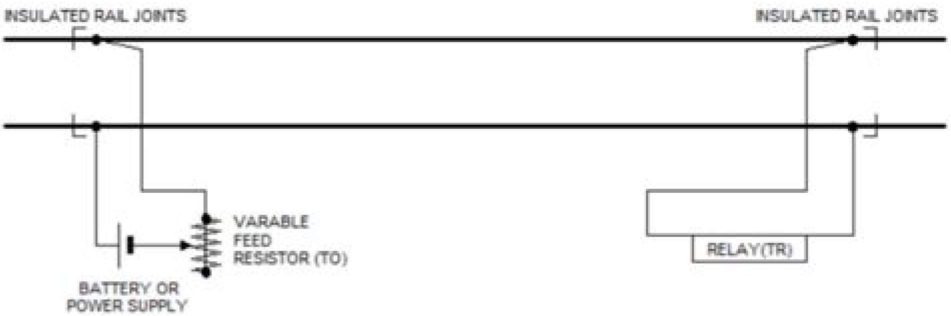

Figure 3.3: Track Circuit mechanism

A feed is applied to the rails at one end, with each rail being of opposite polarity. At the other end a relay is connected across the rails. The current flows down through one rail track to energise the relay and returns through the other rail track. When tracks are clear, the track relay (TR) is energised. When a vehicle wheelset is placed across the rails, it causes a short circuit across the feed which drops the TR and sends status of track occupied.

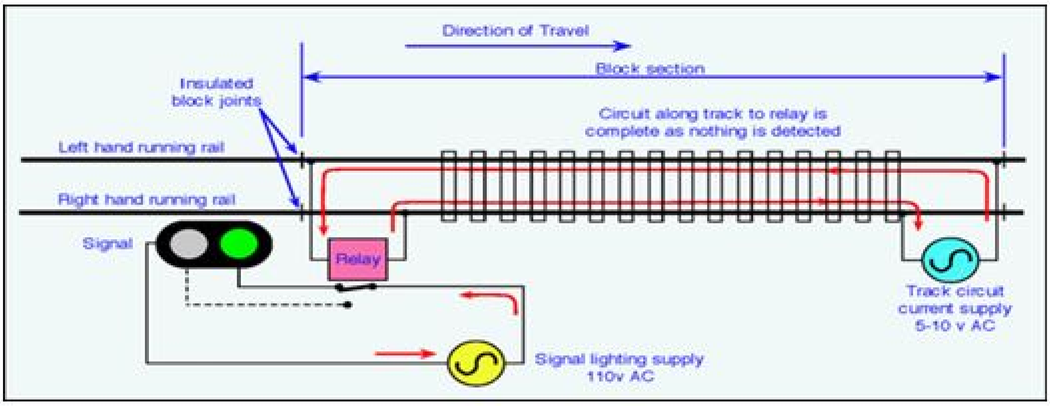

Figure 3.4: Track Circuit - Block Unoccupied

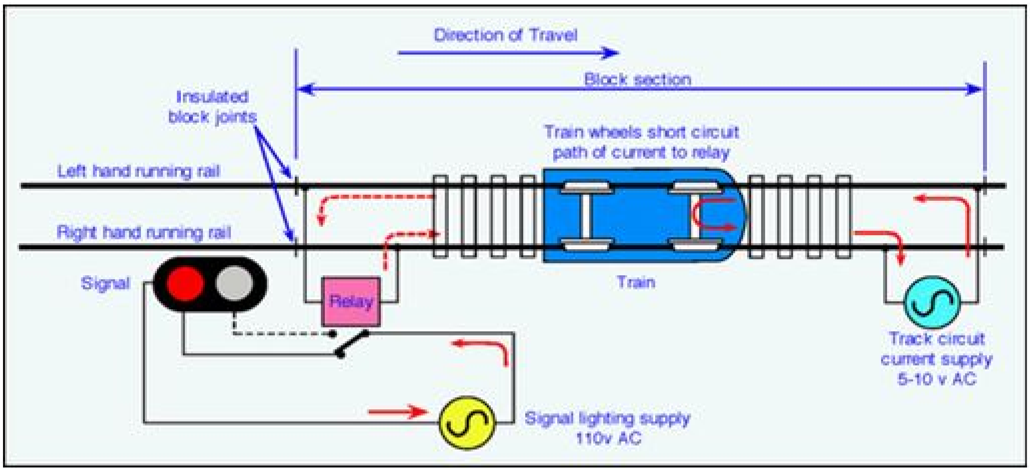

Figure 3.5: Track Circuit - Block Occupied

3.3.2 Single Section Digital Axle Counter

3.6 SINGLE SECTION DIGITAL AXLE COUNTER Axle Counter is a train detection equipment used in Railways for monitoring a defined track section to provide occupancy / clear status. The System detects the presence of a train in any specified track section. The track section can be platform lines, yard lines, block section between two stations. The System consists of SSDAC units, TX/Rx coil Axle detectors and vital relays. The SSDAC Unit is designed with High Frequency tuned circuits; pulse shaping circuits and Micro Controllers. Two units (one pair) constitute one system for monitoring single-track section. The units are to be installed near the trackside at the beginning and end of the track section i.e. outer limits of section. The system is easy to install, Commission and maintain. The system is designed using Micro controller along with other electronic circuits and programmed using dedicated software.

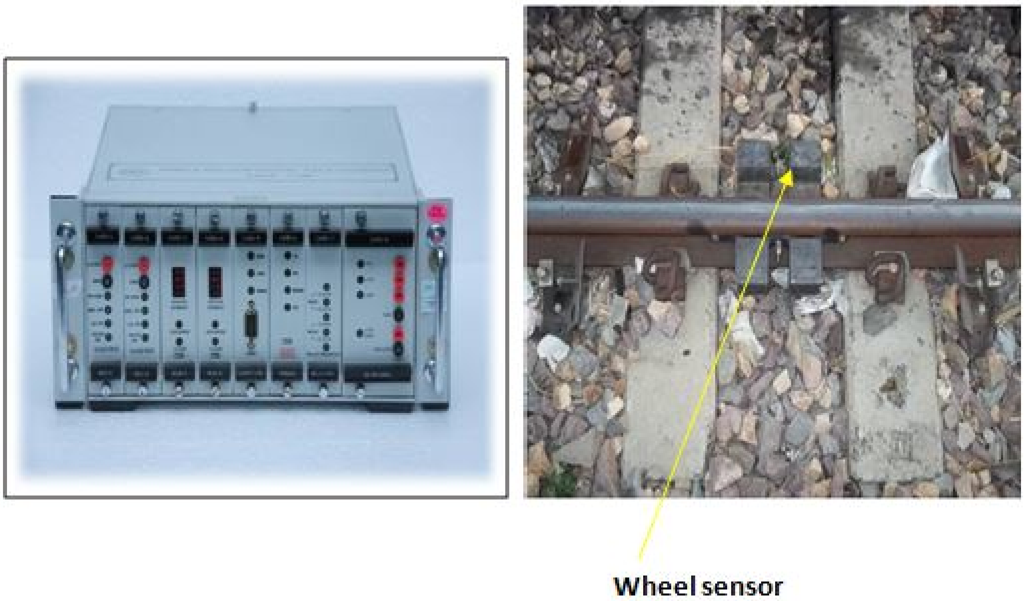

Figure 3.6: Single Section Digital axle counter

3.4 Relays

A relay is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relays. Relays are used to control a circuit by a separate low-power signal, or where several circuits must be controlled by one signal. Signalling relays are basically electro mechanical switches working on DC supply. All signalling relays have two types of contacts - Front Contacts (which close when the relay is energized) and alternatively, Back contacts (which close when the relay is de-energized). Physically, both the contacts are exactly similar. These contacts are arranged in many configurations such as 12F/4B, 8F/8B etc. The singularity of railway signalling relays is that they operate on low voltages and currents. Also they can work in any specified manner, while contributing to speed & accuracy in operations.

Figure 3.7: Relay

Summary

This chapter mainly concentrates on the basic theoretical background related to the topic of focus. It gives information about the platform on which this application has been developed in this chapter.

Chapter 4

SYSTEM REQUIREMENT SPECIFICATION

Software requirement Specification is a fundamental document, which forms the foundation of the software development process. It not only lists the requirements of a system but also has a description of its major feature. An SRS is basically an organization's understanding (in writing) of a customer or potential client's system requirements and dependencies at a particular point in time (usually) prior to any actual design or development work. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time.

The SRS also functions as a blueprint for completing a project with as little cost growth as possible. The SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it. It is important to note that an SRS contains functional and nonfunctional requirements only.
Thus the goal of preparing the SRS document is to
→ To facilitate communication between the customer, analyst, system developers, maintainers.
→ To serve as a contrast between purchaser and supplier.
→ To rm foundation for the design phase.
→ Support system testing facilities.
→ Support project management and control.
→ Controlling the evolution of the system.

4.1 Functional Requirements

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. In this system following are the functional requirements:-

→ Input test case must not have compilation and runtime errors.
→ The application must not stop working when kept running for even a long time.
→ The application must function as expected for every set of test cases provided.
→ The application should generate the output for given input test case and input parameters.
→ The application should generate on-demand services.

4.2 Non-Functional Requirements

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviors. They may relate to emergent system properties such as reliability, response time and store occupancy. Non-functional requirements arise through the user needs, because of budget constraints, organizational policies, the need for interoperability with other software and hardware systems or because of external factors such as:-

→ Product Requirements.
→ Organizational Requirements.
→ User Requirements.
→ Basic Operational Requirements.

In systems engineering and requirements engineering, a non-functional requirement is a requirement that species criteria that can be used to judge the operation of a system, rather than specic behaviours. This should be contrasted with functional requirements that define specific behaviour or functions. The plan for implementing non-functional requirements is detailed in the system architecture. Broadly, functional requirements dene what a system is supposed to do and non- functional requirements define how a system is supposed to be. The Systems overall properties commonly mark the dierence between whether the development project has succeeded or failed. Non-functional requirements of our project include:

→ **Response time** The time the system takes to load and the time for responses on any action the user does.
→ **Processing time** - How long is acceptable to perform key functions or export / import data?
→ **Throughput** The number of transactions the system needs to handle must be kept in mind.
→ **Storage** - The amount of data to be stored for the system to function.
→ **Architectural Standards** The standards needed for the system to work and sustain.

4.2.1 Product Requirements

→**Correctness:** It follows a well-defined set of procedures and rules to compute and also rigorous testing is performed to confirm the correctness of the data.

→**Ease of Use:** The front end is designed in such a way that it provides an interface which allows the user to interact in an easy manner.

→**Modularity:** The complete product is broken up into many modules and well dened interfaces are developed to explore the benet of exibility of the product.

→**Robustness:** This software is being developed in such a way that the overall performance is optimized and the user can expect the results within a limited time with utmost relevancy and correctness.

Where as evolution quality involves testability, maintainability, extensibility or scalability.

4.2.2 Organizational Requirements

Process Standards: Indian Railway standards are standards used to develop the application which is the standard followed to develop any standalone for Electronic Interlocking System. Design Methods: Design is one of the important stages in the software engineering process. This stage is the rst step in moving from problem to the solution domain. In other words, starting with what is needed design takes us to work how to satisfy the needs.

4.2.3 User Requirements

The user requirements document (URD) or user requirements specication is a document usually used to software engineering that species the requirements the user expects from software to be constructed in a software project. Once the required information is completely gathered it is documented in a URD, which is meant to spell out exactly what the software must do and becomes part of the contractual agreement. A customer cannot demand feature not in the URD, whilst the developer cannot claim the product is ready if it does not meet an item of the URD. The URD can be used as a guide to planning cost, timetables, milestones, testing etc. The explicit nature of the URD allows customers to show it to various stakeholders to make sure all necessary features are described. Formulating a URD requires negotiation to determine what is technically and economically feasible. Preparing a URD is one of those skills that lies between a science and economically feasible. Preparing a URD is one of those skills that lies between a science and an art, requiring both software technical skills and interpersonal skills.

4.2.4 Basic Operational Requirements

Operational requirement is the process of linking strategic goals and objectives to tactic goals and objectives. It describes milestones, conditions for success and explains how, or what portion of, a strategic plan will be put into operation during a given operational period, in the case of commercial application, a scal year or another given budgetary term. An operational plan is the basis for, and justication of an annual operating budget request. Therefore, a ve-year strategic plan would typically require ve operational plans funded by ve operating budgets. Operational plans should establish the activities and budgets for each part of the organization for the next 1-3 years. They link the strategic plan with the activities the organization will deliver and the resources required to deliver them. An operational plan draws directly from agency and program strategic plans to describe agency and program missions and goals, program objectives, and program activities. Like a strategic plan, an operational plan addresses four questions:

→ Where are we now?
→ Where do we want to be?
→ How do we get there?

The customers are those that perform the eight primary functions of systems engineering, with special emphasis on the operator as the key customer. Operational requirements will dene the basic need and, at a minimum, will be related to these following points:

Mission profile or scenario: It describes about the procedures used to accomplish mission objective. It also finds out the effectiveness or efficiency of the system.

Performance and related parameters: It points out the critical system parameters to accomplish the mission

Utilization environments: It gives a brief outline of system usage. Finds out appropriate environments for effective system operation.

Operational life cycle: It defines the system lifetime.

4.3 Hardware Requirements

→ Internal memory 512KB flash for code storage.
→ Internal SRAM 96KB for data storage.
→ External memory Flash of size 32Mbit based on SPI interface.
→ Serial COM port for PC interface with max. Baud rate of 115200 baud.
→ RTC Clock with battery.
→ 48 Relay drive channels and each channel can source 24VDC@500mA max.
→ The board consist of 96 input reading channels for knowing the status of the relay.
→ Each output relay drive has internal shot circuit protection.

4.4 Software Requirements

→ Operating System: Windows 7/10 and Ubuntu.
→ Coding Language : C sharp.
→ Tools : Visual Studio Dot Net Framework, External USB to RS-232 convertor drivers.

4.5 Software Quality Attributes

→**Functionality:** the capability of the software to provide functions which meet stated and implied needs when the software is used under specied conditions.
→**Reliability:** the capability of the software to maintain its level of performance when used under specied conditions.
→**Usability:** the capability of the software to be understood, learned, used and liked by the user, when used under specied conditions.
→**Eciency:** the capability of the software to provide the required performance, relative to the amount of resources used, under stated conditions.
→**Maintainability:** the capability of the software to be modied. Modications may

include corrections, improvements or adaptation of the software to changes in environment, and in requirements and functional specications.
→**Portability:** the capability of software to be transferred from one environment to a

Summary

This chapter gives details of the functional requirements, non-functional requirements, resource requirements, hardware requirements, software requirements etc. Again the non-functional requirements in turn contain product requirements, organizational requirements, user requirements, basic operational requirements etc.

Chapter 5

SYSTEM DESCRIPTION

5.1 Introduction

The ARP/EP (Auxiliary Relief Panel / Emergency Panel) is a microprocessor based equipment used to control the operation of points, signals, and level crossing gates, track circuits for manual operation of route setting and route releasing when Electronic Interlocking System (EI) fails. A railway electronic interlocking (EI) system is a kind of control system, which is developed to control railway traffic securely. The basic objective of interlocking system is to prevent trains from colliding and derailing by controlling field equipment such as points, signals and track circuits in a railway stations. Hence, when EI system fails due to system error or human error, Emergency Panel is installed and used to control train movements .In this system, it focuses the train movements on two main routes and on two platforms. The Station Master interacts with graphical user interface of video display unit to control and set the routes for train at the state of emergency.

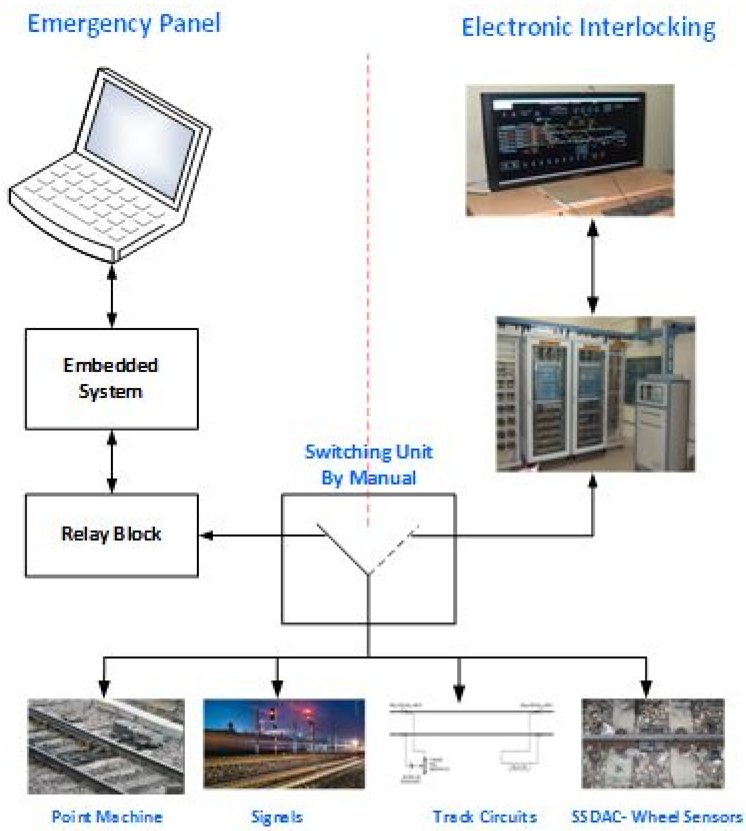

Figure 5.1: Emergency Panel Implementation

5.2 Theory Of Operation

The above block diagram consist of two main system
→ Electronic Interlocking System
→ Emergency panel.
Both systems are used to control railway traffic and prevent trains from colliding and derailing by controlling field equipment such as points, signals and track circuits in a railway stations.
Emergency panel is implemented through Manual Switching unit when Electronic interlocking system fails to function.
Components of Emergency panel are:
→ Application Software
→ Embedded System
→ Relay block
→ Manual Switching Unit

5.2.1 Application Software

User sets commands through application software. The PC is connected to embedded system through RS232 communication interface. Station Master is the user of the system executes commands from the application software.
These commands are:
→ To connect to platform track
→ To dispatch the stationed train.
These commands are sent to the embedded system through RS232 communication. The communication implements using ICD (Interface Control Document) which contains information about command structure. The Embedded system interrupts the request and executes the command.
The Application software consists of the GUI (Graphical User Interface) screen which contains soft switches, lamps, tracks, LEDs and many more. The user operates train routes using GUI controls. The GUI screen also displays the status of the field units information to confirm the settings.

5.2.2 Embedded System

The embedded system is developed using ARM controller and other necessary hardware circuits to perform its function. Embedded software is used to dictate the field units such as points, etc. to implement the command received from application

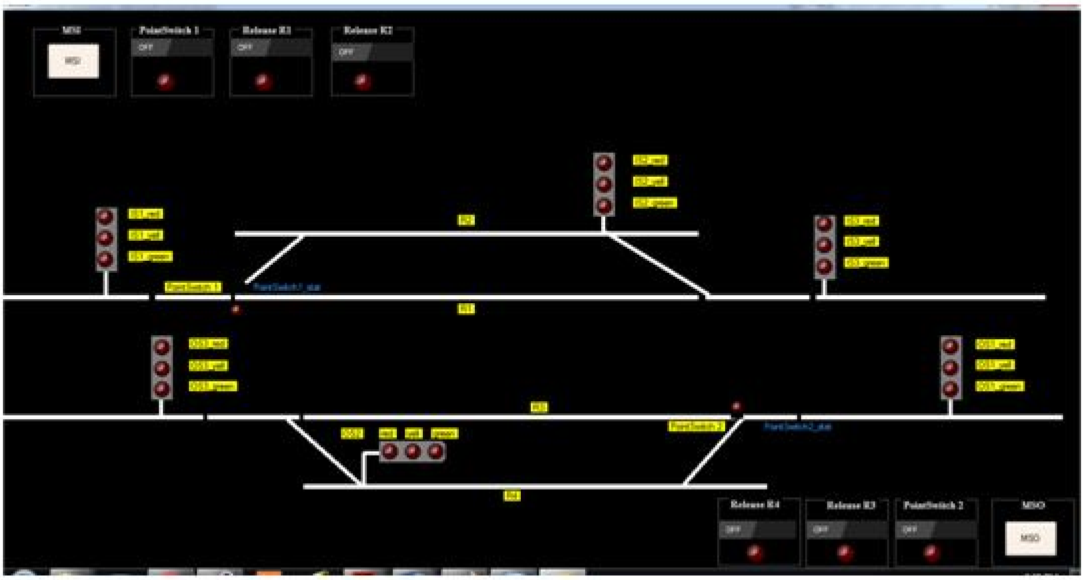

Figure 5.2: Graphical user interface of application software

software and read the feedback of field units and send back to application software.

5.2.3 Relay Block

Relay block contains more number of standard relays which is used to switch states in high current devices. The embedded hardware directly drives the relay to act as switch and the front contact is used to apply required electrical signals to field units.

5.2.4 Manual Switching Unit

The manual switching unit is required to switch control signals between Electronic Interlocking and Emergency Panel. Whenever Electronic Interlocking system fails due to human error or system error, the Station Master switches all signals to Emergency Panel to route the trains before any mishaps occur. Now the Emergency Panel has the control over field units and train routes. Once Electronic Interlocking is restored back after required services performed, then all control signals are switched back to Electronic Interlocking System.

Chapter 6

SYSTEM DESIGN

Overview

Design is a meaningful engineering representation of something that is to be built. It is the most crucial phase in the developments of a system. Software design is a process through which the requirements are translated into a representation of software. Design is a place where design is fostered in software Engineering. Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. Design is the perfect way to accurately translate a customers requirement in the finished software product. Design creates a representation or model, provides details about software data structure, architecture, interfaces and components that are necessary to implement a system. The logical system design arrived at as a result of systems analysis is converted into physical system design.

6.1 System development methodology

System development method is a process through which a product will get completed or a product gets rid from any problem. Software development process is described as a number of phases, procedures and steps that gives the complete software. It follows series of steps which is used for product progress. The development method followed in this project is waterfall model.

6.1.1 Model phases

The waterfall model is a sequential software development process, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Requirement initiation, Analysis, Design, Implementation, Testing and maintenance.

Requirement Analysis: This phase is concerned about collection of requirement of the system. This process involves generating document and requirement review.

System Design: Keeping the requirements in mind the system specifications are translated in to a software representation. In this phase the designer emphasizes on:-algorithm, data structure, software architecture etc.

Coding: In this phase programmer starts his coding in order to give a full sketch of product. In other words system specifications are only converted in to machine readable compute code. Implementation: The implementation phase involves the actual coding or programming of the software. The output of this phase is typically the library, executables, user manuals and additional software documentation .

Testing: In this phase all programs (models) are integrated and tested to ensure that the complete system meets the software requirements. The testing is concerned with verification and validation.

Maintenance: The maintenance phase is the longest phase in which the software is updated to fulfill the changing customer need, adapt to accommodate change in the external environment, correct errors and oversights previously undetected in the testing phase, enhance the efficiency of the software.

6.1.2 Reason for choosing Waterfall Model as development method

1.Clear project objectives.
2.Stable project requirements.
3. Progress of system is measurable.
4.Strict sign-off requirements.
5.Helps you to be perfect.
6. Logic of software development is clearly understood.
7.Production of a formal specification.
8.Better resource allocation.
9.Improves quality. The emphasis on requirements and design before writing a single line of code ensures minimal wastage of time and effort and reduces the risk of schedule slippage.

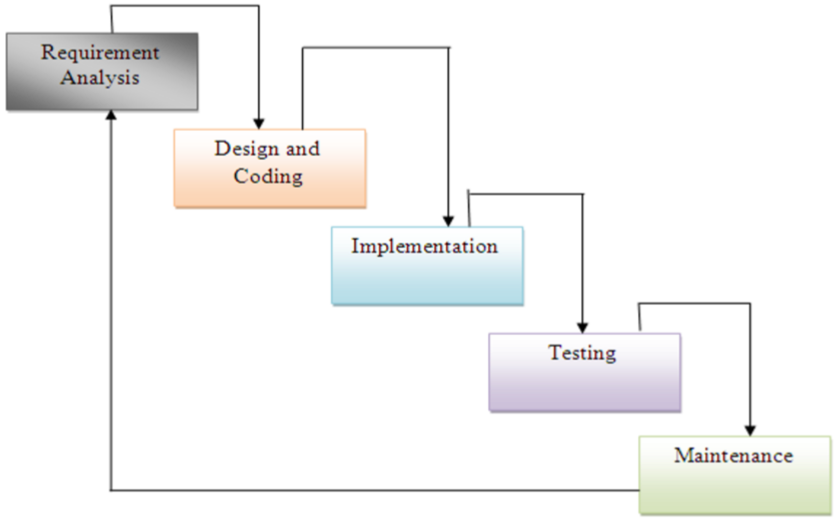

Figure 6.1: Waterfall model

6.2 Use Case Diagram

A use case defines a goal-oriented set of interactions between external entities and the system under consideration. The external entities which interact with the system are its actors. A set of use cases describe the complete functionality of the system at a particular level of detail and it can be graphically denoted by the use case diagram.

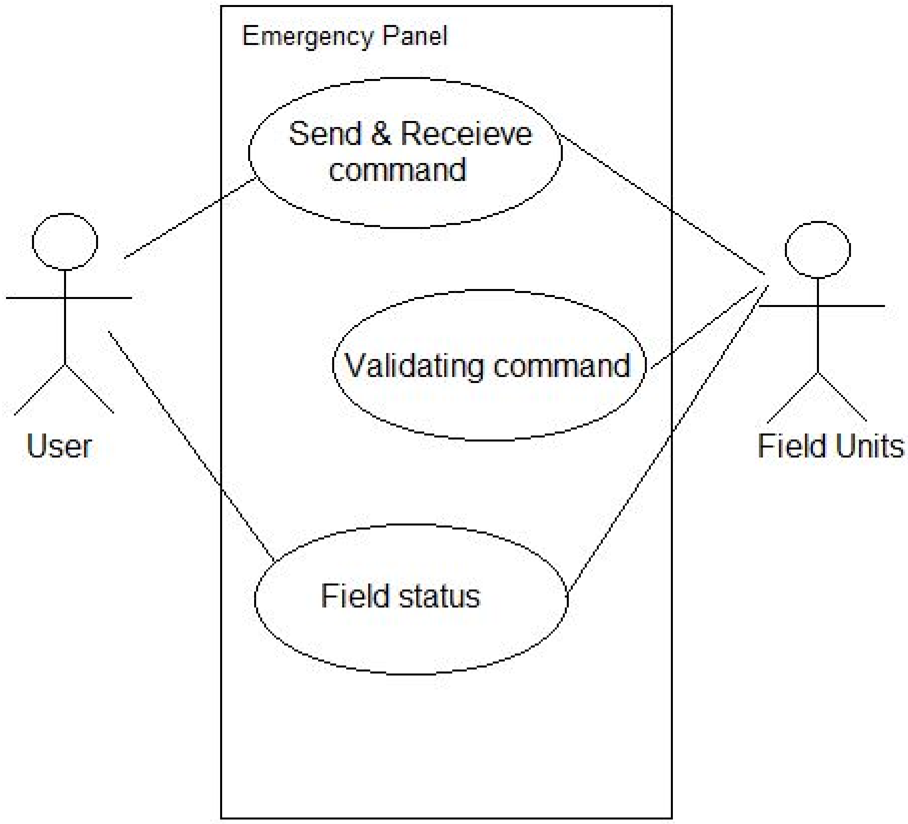

Figure 6.2: Use Case Diagram

6.3 Sequence Diagram

Sequence diagram are an easy and intuitive way of describing the behavior of a system by viewing the interaction between the system and the environment. A sequence diagram shows an interaction arranged in a time sequence. A sequence diagram has two dimensions: vertical dimension represents time, the horizontal dimension represents the objects existence during the interaction.
Basic elements:

- Vertical rectangle: Represent the object is active (method is being per-formed).
- Vertical dashed line: Represent the life of the object.
- X: represent the life end of an object. (Being destroyed from memory)
- Horizontal line with arrows: Messages from one object to another.

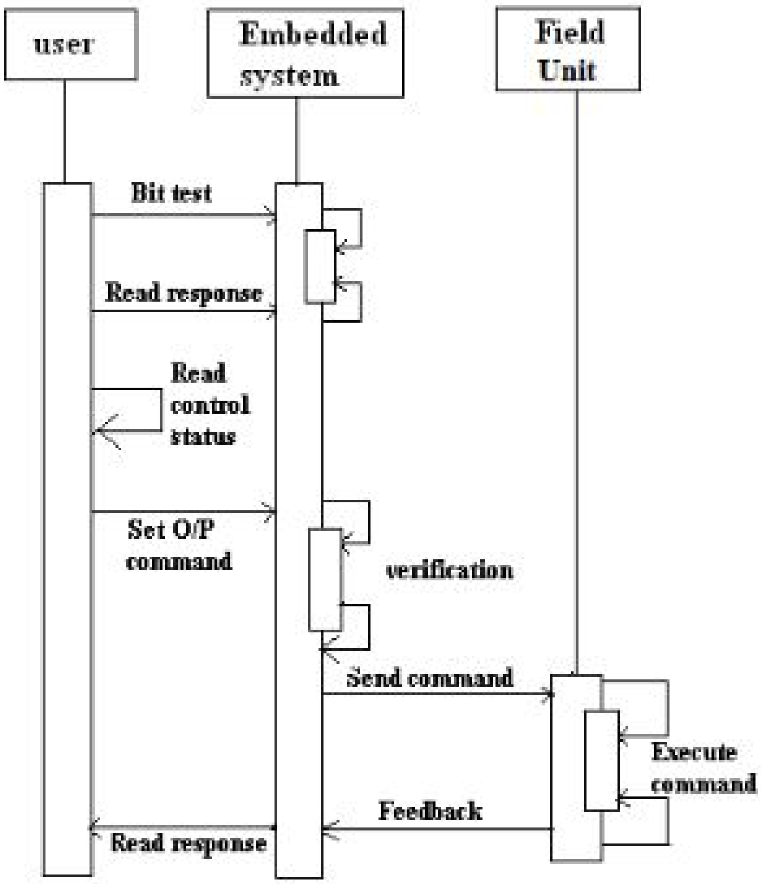

Figure 6.3: Sequence Diagram

Chapter 7

IMPLEMENTATION

The implementation phase of the project is actually carrying out ,execution or practice of a plan, method, or any designed idea, model specification, standard, or policy for doing some task. Aim of this phase is to translate the design into a best possible solution in a suitable programming language. This chapter covers the implementation aspects of the project giving details of the programming language and the development environment used. This stage requires following tasks:

→ Careful planning.
→ Investigation of system and constraints.
→ Design of methods to achieve the changeover.
→ Evaluation of the changeover method.
→ Correct decisions regarding selection of the platform.
→ Appropriate selection of the language for application development.

7.1 Video Display Unit Code

```
Code to send command from PC to embedded system through
RS-232 serial communication
  Public void serial_send (Byte [] tran_buffer)
        {

             serial_p1.Write (tran_buffer, 0, tran_buffer.Length);

        }
Code to receive data from hardware unit through
RS-232 serial communication
       Public void serial_recv (Byte [] rec_buffer)
       {
           Int i = serial_p1.Read (rec_buffer, 0, rec_buffer.
           Length);
           if (i <= 0)
           {
               MessageBox.Show ("fail");
           }
       }
Code to establish RS-232 Serial communication
static SerialPort serial_p1 = new SerialPort("COM4",9600,
```

```
Parity.None, 8, StopBits.One);
Private void Form1_Load (object sender, EventArgs e)
        {
form.FormBorderStyle = FormBorderStyle.None;
Form .WindowState = FormWindowState.Maximized;
                              form.FormBorderStyle =
                              FormBorderStyle.Sizable;
                              form.WindowState =
                              FormWindowState.Maximized;
                 Try
                   {
                        serial_p1.Open ();
                   }
                 Catch (Exception ex)
                   {
                        MessageBox.Show ("Error" + ex);
                   }
        if (serial_p1.IsOpen==true)
                 {

                          MessageBox.Show ("open");
            Byte sum_value, recv_csum ;
                          Byte[] transmit_buffer = new Byte[5];
                        Byte[] receive_buffer = new Byte[6];
//This command shall be used to perform built in test of
hardware unit.
                        //Command Code: 0x01
transmit_buffer[0] = 0x5A;
transmit_buffer[1] = 0x01;
transmit_buffer[2] = 0x00;
transmit_buffer[3] = 0x01;
sum_value = (Byte)(transmit_buffer[0] + transmit_buffer[1] +
transmit_buffer[2] +        transmit_buffer[3]);
                   transmit_buffer[4] = two_complement(sum_value);
                          //transmiting data
                            serial_send(transmit_buffer);
                          System.Threading.Thread.Sleep(100);

                      //receive data
```

```
                    serial_recv(receive_buffer);
recv_csum = (Byte)(receive_buffer[0] + receive_buffer[1] +
receive_buffer[2] + receive_buffer[3] + receive_buffer[4]);
                     recv_csum = two_complement(recv_csum);

                 if(receive_buffer[5]==recv_csum)
                   {
                     //2nd condition -command byte of transmit
                     buffer and recieved buffer are equal
                         if(transmit_buffer[3]==receive_buffer[3])
                       {
                         //check status
                           if(receive_buffer[4]==0x00)
                             {
                                 result.Text = "pass";
                                 class1 c = new class1(this);
                                 c.callINput();
 }
                             else
                             {
                                 result.Text = "fail";

                             }
                       }
                         else
{
MessageBox.Show("command does not match");
 }


                  }
                 else
                 {
                         MessageBox.Show("error1");

                 }
              }
         else
            {
                 MessageBox.Show("connection not established");
```

```
            }


    }
```

7.2 Command to set O/P (Relay)

```
public int SwitchONOFF()
      {
          Byte[] receive_buffer = new Byte[6];
          Byte recv_csum;
              Byte command5 = 0x05;
            //To read status of display unit on PC
              Read_status(command5, this);
              System.Threading.Thread.Sleep(100);
              //receive data
          serial_recv(receive_buffer);
       recv_csum = (Byte)(receive_buffer[0] + receive_buffer[1]
   + receive_buffer[2] + receive_buffer[3] + receive_buffer[4]);
             recv_csum = two_complement(recv_csum);
          if (receive_buffer[5] == recv_csum)
          {
            if (command5 == receive_buffer[3])
              {
                  if (receive_buffer[4] == 0x00)
                  {
                      MessageBox.Show ("command executed ");

                  }
                  else
                  {
                      MessageBox.Show ("fail");
                      return -1;
                  }
              }
              Else
 {
                  MessageBox.Show("command does not match5");
 }
```

```
            }
            else
            {
MessageBox.Show("error5"); }
              return 0;

           }
```

7.3 Graphical User Interface (GUI) Module

GUI allows the use of icons or other visual indicators to interact with electronic devices, rather than using only text via the line. Designing the visual composition and temporal behaviour of a GUI is an important part of software application programming in the area of humancomputer interaction. Its goal is to enhance the efficiency and ease of use for the underlying logical design of a stored program. Methods of user-centred design are used to ensure that the visual language introduced in the design is well-tailored to the tasks.

The GUI module depicts the tracks and status of tracks on a display screen, which is used by station master to monitor and issues command accordingly.GUI module continuously interacts with hardware in a pre-set timer of 500ms/400ms to receive updates of the track. Hence, station master can get hold of real time monitoring and issues command accordingly.

The commands issued by station master are:
→ **Master Switch IN (MSI)**-An incoming train trigger event.
→ **Point Switch 1**-To connect R2 track to the main track.
→ **Release R1**-To dispatch train stationed on Track R1.
→ **Release R2**- To dispatch train stationed on Track R2.
→ **Master Switch OUT (MS0)**- An incoming train trigger event.
→ **Point Switch 2**- To connect R4 track to the main track.
→ **Release R3**- To dispatch train stationed on Track R3.
→ **Release R4**- To dispatch train stationed on Track R4.

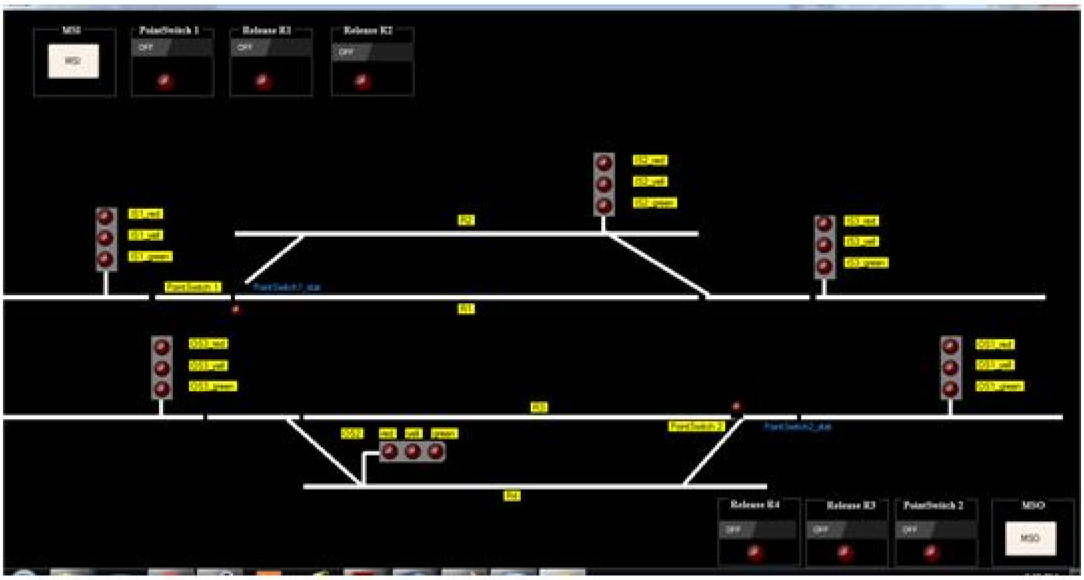

Figure 7.1: Graphical user interface display

7.4 Electronic Interlocking Rules Implementation Module

In Indian Railways operating rules are common for all zonal railways of Indian Railway and can be amended only by the Railway Board.

These rules are based on data received by embedded system from certain signalling components. Embedded System receives input from hardware unit called field status sent by some fundamental components such as SSDAC (single section digital axle counter), track circuit, points.

→ **A Railway point** is a mechanical tool which allows trains to be guided from one track to another on railway intersection according to the desired route. There are two positions named as normal and diverging.

→ **Railway signals** are a system used to control railway traffic safely so that collision of the trains can be prevented. Signals transmit coloured light (green, red, yellow) notice, which notifies the trains regarding the proceeding of the trains and feed up until the next signal. There are three types of signal; 3-aspect long signals, 4-aspect long signals and 3-aspect ground signals. 3-aspect long signals include red, green and yellow lights. There is also an additional yellow light in 4-aspect long signals. This extra yellow light indicates whether there is a deviation from the route or not. 3-aspect ground signals are used inside of the stations. It is understood to be a deviation from the colour of the notification of 3- aspect ground signals.

Indicator	**Meaning**
Red	Stop at signal proceeding track is busy.
Yellow	Slow down at signal
Green	Proceeding track and the next track is empty.

Table 7.1: Signal indicators and their meaning

→ **Track circuit** is a simple electrical circuit designed to detect the absence or presence of a railway vehicle in a certain part of a railway track. They provide information whether the route is available or occupied by a railway vehicle.
→ **Axle counter** is a device for monitoring a specified section of track for the presence of vehicle. Now Axle Counters are designed using Micro-Controllers and Software program, and these are called as Digital Axle Counters. Axle counters are generally used to monitor a track section i.e. there is only one entry point and one exit point for a particular track section. Digital data is transmitted directly between two axle counter field equipment by means of modem communication. There are counting heads (Axle Detectors) at the beginning and end of each track section. These units are connected to an evaluation computer (Central Evaluator) which processes the information generated by the counting heads. If the number of axles counted in matches that counted out, the respective track section is indicated as being clear.

With respect to these equipment output the status of the track can be analysed.

Hence, according to the predefined rules and output of these equipment sent through communication module the issued command from station master can be verified. This module confirms whether to send the issued command to hardware unit or reject it.

7.5 Communication Module

Communication module acts as interface between software unit and embedded system/hardware unit. It helps to communicate with hardware unit.

Messages are sent and received on communicating systems to establish communication. Protocols should therefore specify rules governing the transmission. Communication protocol is a system of rules that allow two or more entities of a communications system to transmit information via any kind of variation of a physical quantity.

The protocol defines rules syntax, semantics and synchronization of communication and possible error recovery methods. Protocols are implemented by hardware, software, or a combination of both.

The protocols used in this system is RS-232 protocol.

7.5.1 RS-232 Protocol

The RS-232 serial communication protocol is a standard protocol used in asynchronous serial communication. The process of sending data sequentially over a computer bus is called as serial communication, which means the data will be transmitted bit by bit. It is used to send command issued by station master to hardware unit and receive data sent by hardware unit.

Characteristics of Serial Communication:
→ **Baud rate** is used to measure the speed of transmission. It is described as the number of bits passing in one second. For example, if the baud rate is 200 then 200 bits per Sec passed. In telephone lines, the baud rates will be 14400, 28800 and 33600.
→ **Stop Bits** are used for a single packet to stop the transmission which is denoted as T. Some typical values are 1, 1.5 & 2 bits.
→ **Parity Bit** is the simplest form of checking the errors. There are of four kinds, i.e., even odd, marked and spaced. For example, If 011 is a number the parity bit=0, i.e., even parity and the parity=1, i.e., odd parity.

An RS-232 serial port was once a standard feature of a personal computer, used for connections to modems, printers, mice, data storage, uninterruptible power supplies, and other peripheral devices. In modern personal computers, USB has displaced RS-232 from most of its peripheral interface roles Many computers no longer come equipped with RS-232 ports (although some motherboards come equipped with a COM port header that allows the user to install a bracket with a DE-9 port) and must use either an external USB-to-RS-232 converter or an internal expansion card with one or more serial ports to connect to RS-232 peripherals. In this system external USB-to-RS-232 converter is used to connect PC and ARP/EP .In RS232 there are two data lines RX and TX. TX is the wire in which data is sent out to embedded system. RX is the line in which hardware unit put the data it need to send to the PC.

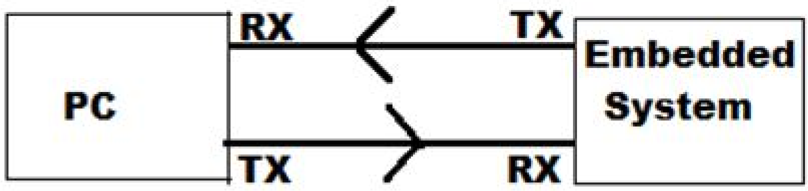

Figure 7.2: RS-232 serial communication

7.5.2 ICD

Interface Command Document contains the List of all commands, its format and function of the command.

Message format

Following message format shall be implemented on top of UDP protocol.

Request from host

Host [PC] shall send request message in following formats.

MESSAGE FORMAT					
	HEADER		DATA		CHECKSUM
No of bytes	1	2	1	0-1000	1
	Start of message	Length	Command	parameters(optional)	Checksum
Example	0X5A	0X01	0X00		0xXX

Figure 7.3: Message Request Format

→ Start of message (SOM): 0x5A

→ Length (Size): Number of bytes in the data field (excluding header and checksum bytes)

→ Command: First byte in the data field, which tells which command is being requested

→ Parameters: Optional data field, containing additional bytes of information for the command

→ Checksum: Twos complement checksum of preceding bytes includes DATA and Header bytes. The 8 bit sum of all bytes (including the checksum) is zero if packet contents have not been corrupted. A non-zero sum indicates that packets have been corrupted.

Response from ARP/EP

EMP shall send response message in following formats.

MESSAGE FORMAT					
	HEADER		DATA		CHECKSUM
No of bytes	1	2	1	1	1
	Start of message	Length	Command	Result	Checksum
Example	0X5A	0X01	0X00	0X00	0xXX

Figure 7.4: Message Response Format

→ Start of message (SOM): 0x5A

→ Length (Size): Number of bytes in the data field (excluding header and checksum bytes)

→ Command: First byte in the data field, which tells which command is processed

→ Status: Second byte in the data field, which tells the status of requested command. A non-zero indicates the requested command is failed. Refer error code table for more detail about status information.

→ Parameters: Optional data field, containing additional bytes of information for the processed command

→ Checksum: Twos complement checksum of preceding bytes includes DATA and Header bytes. The 8 bit sum of all bytes (including the checksum) is zero if packet contents have not been corrupted. A non-zero sum indicates that packets have been corrupted.

Error Code

Error code shall be assigned in status byte of response message. The following table for contains error code and its description

Error Code	Desceiption
0x00	Test Fail.
0x01	No Error.
0x02	Invalid Length.
0x03	Invalid Command.
0x04	Invalid End Packet.
0x05	Checksum Error.
...	...
...	...
TBD	TBD

Table 7.2: Error code

Command List

Command code shall be assigned in command byte of request / response message. Refer following table for commands.

Command Code	Command Name
01	BIT.
05	Set O/P(Relay or System OP.
06	Read Input.
07	MSI Trigger Event.
08	MSO Trigger Event.

Table 7.3: Commands

BIT / SELFTEST Request

This command shall be used to perform built in test of hardware unit.

Command Code: 0x01

Parameters: None

Byte Order(Index)	**Byte Data**	**Description**
1	0x5A	SOM
2	0x00	Length MSB
3	0x04	Length LSB
4	0x02	Command
5	0x00	Paramater (dummy data)
6	0xXX	Checksum

Table 7.4: BIT Request Format

BIT / SELFTEST Response

PC shall receive this command response as a result of built in test.

Byte Order(Index)	**Byte Data**	**Description**
1	0x5A	SOM
2	0x03	Length LSB
3	0x00	Length MSB
4	0x01	Command
5	0x01/00	Status
6	0xXX	Checksum

Table 7.5: BIT Response Format

Set Output

This command shall be used to send command to hardware unit

Command Code: 0x05

Parameters: Relay no (1-128), Relay State (0/1)

Set Output Request

Byte Order(Index)	Byte Data	Description
1	0x5A	SOM
2	0x07	Length LSB
3	0x00	Length MSB
4	0x05	Command Byte
5	0x01/00	Status of Point Switch 1
6	0x01/00	Status of Point Switch 2
7	0x01/00	Status of Track 1
8	0x01/00	Status of Track 2
9	0x01/00	Status of Track 3
10	0x01/00	Status of Track 4
11	0xXX	Checksum

Table 7.6: Set Output Request Format

Set Output Response

Byte Order(Index)	Byte Data	Description
1	0x5A	SOM
2	0x03	Length LSB
3	0x00	Length MSB
4	0x05	Command
5	0x01/00	Status
6	0xXX	Checksum

Table 7.7: Set Output Response Format

Read Input

This command shall be used to read status of actual relay.

Command Code: 0x06

Read Input Request

Byte Order(Index)	Byte Data	Description
1	0x5A	SOM
2	0x01	LSB Length Of Message
3	0x00	MSB Length Of Message
4	0x06	Command Byte
5	0xXX	Checksum

Table 7.8: Read Input Request Fotmat

Read Input Response

Byte Order(Index)	Byte Data	Description
1	0x5A	SOM
2	0x52	LSB Length Of Message
3	0x00	MSB Length Of Message
4	0x06	Command Byte
5	0xXX	Status
6	0xXX	Status
....	....	....
52	0xXX	Checksum

Table 7.9: Read Input Response Fotmat

Chapter 8

TESTING AND RESULTS

8.1 Introduction

Testing is an important phase in the development life cycle of the product this was the phase where the error remaining from all the phases was detected. Hence testing performs a very critical role for quality assurance and ensuring the reliability of the software. Once the implementation is done, a test plan should be developed and run on a given set of test data. Each test has a dierent purpose, all work to verify that all the system elements have been properly integrated and perform allocated functions. The testing process is actually carried out to make sure that the product exactly does the same thing what is supposed to do. Testing is the nal verication and validation activity within the organization itself. In the testing stage following goals are tried to achieve:
• To arm the quality of the project.
• To nd and eliminate any residual errors from previous stages.
• To validate the software as the solution to the original problem.
• To provide operational reliability of the system. During testing the major activities are concentrated on the examination and modication of the source code.

The test cases executed for this project are listed below. Description of the test case, steps to be followed; expected result, status and screenshots are explained with each of the test cases.

8.2 Testing Methodologies

There are many dierent types of testing methods or techniques used as part of the software testing methodology. Some of the important types of testing are:

8.2.1 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level. Using white box testing we can derive test cases that:
• Guarantee that all independent paths within a module have been exercised at least once.
• Exercise all logical decisions on their true and false sides.
• Execute all loops at their boundaries and within their operational bounds.

• Execute internal data structure to assure their validity.

8.2.2 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a denitive source document, such as specication or requirements document, such as specication or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot see into it. The test provides inputs and responds to outputs without considering how the software works. It uncovers a dierent class of errors in the following categories:
• Incorrect or missing function.
• Interface errors.
• Performance errors.
• Initialization and termination errors.
• Errors in objects.

Advantages:
• The test is unbiased as the designer and the tester are independent of each other.
• The tester does not need knowledge of any specic programming languages.
• The test is done from the point of view of the user, not the designer.
• Test cases can be designed as soon as the specications are complete.

8.3 TEST CASE 1

Input: MSI button is clicked ,where
• R1 track is unoccupied
Output: The train is stationed on track R1,R1 status changes from green to red specifying it is occupied

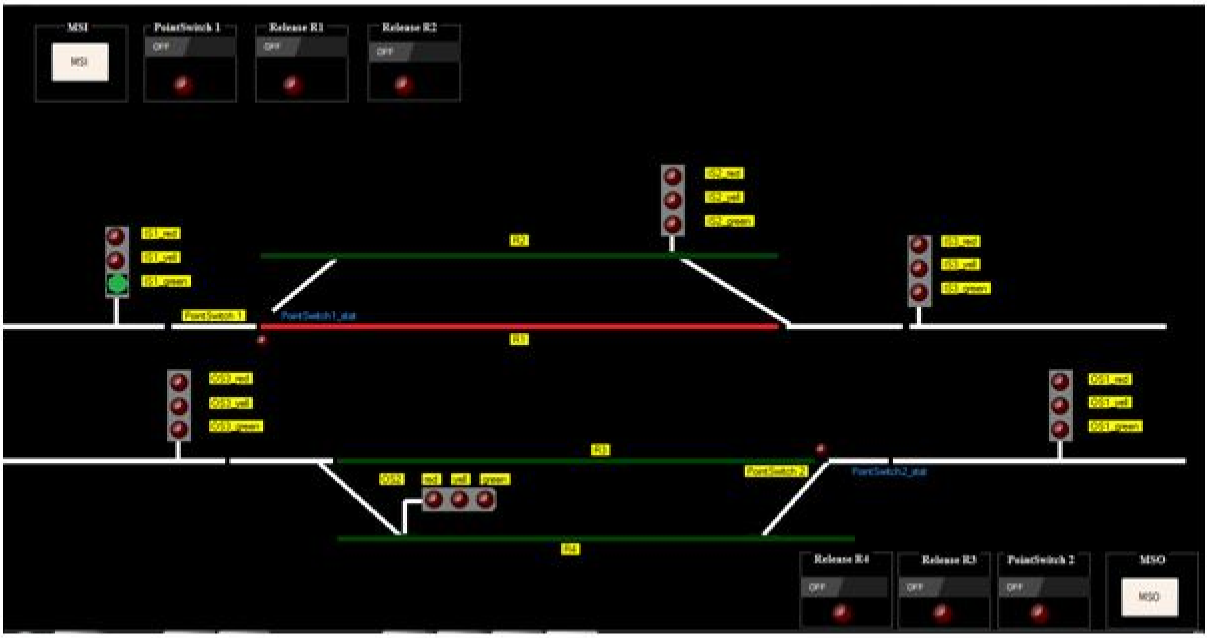

Figure 8.1: Test Case 1

8.4 TEST CASE 2

Input: MSI button is clicked where ,

- R2 track is unoccupied
- point switch 1 is in ON state

Output: The train is stationed on track R2 and R2 changes status from green to red specifying it is occupied

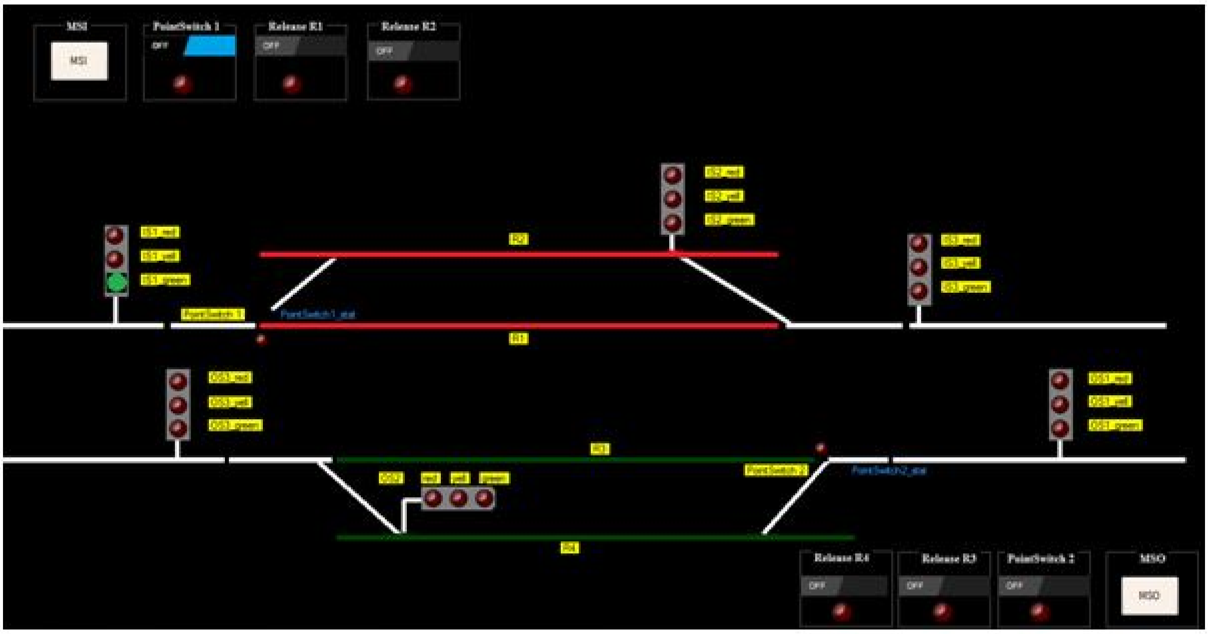

Figure 8.2: Test Case 2

8.5 TEST CASE 3

Input: MSI button is clicked when both R1 and R2 track are occupied Output: The train is waiting for its position IS1(incoming signal 1) indicates red(stop).

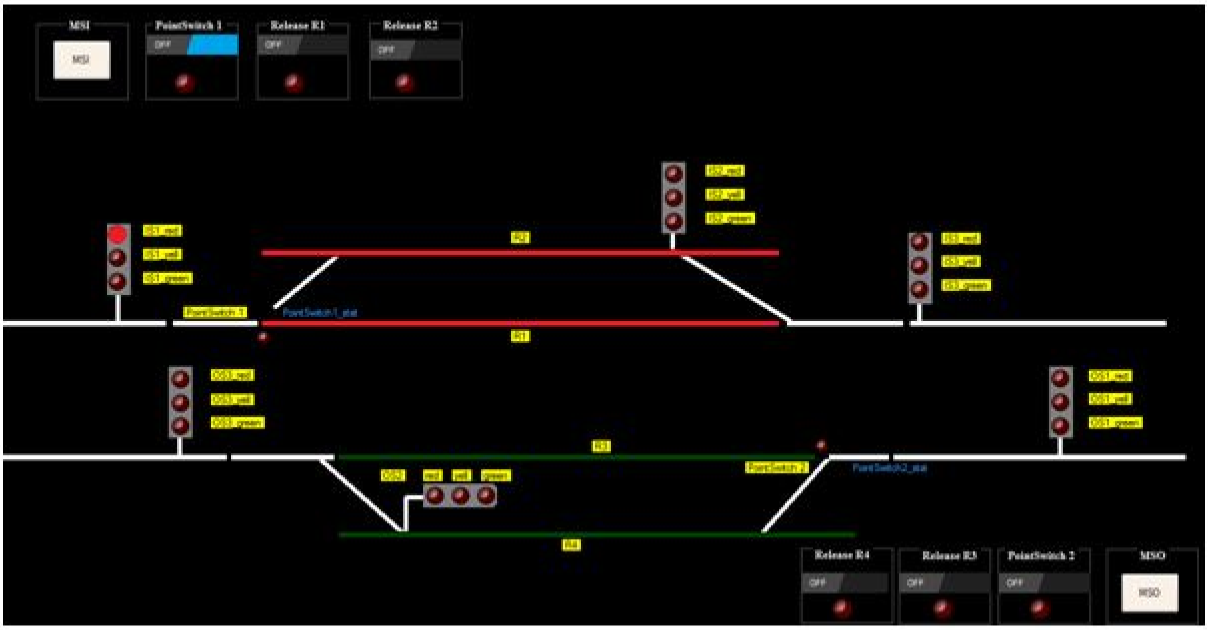

Figure 8.3: Test Case 3

Summary

The chapter discusses the tests that are done on the system to check its functionality. Testing is carried out at three different levels from the module level to the system level checking for errors at each stage. The remarks have also been documented.

Chapter 9

CONCLUSION & FUTURE SCOPE

9.1 Conclution

Considering the need to operate train network with full capacity same as in existing interlocking system when system fails to function is achieved by Auxiliary relief panel/Emergency panel. This system acts as a standby interlocking for small and big stations as the failure management mechanism. It is also capable of detecting the electronic interlocking system failure from standard contact (Relay contact which shall indicate failure of EI) and record for health monitoring purpose .Hence rail accidents, delays can be avoided and it minimizes the amount spent in terms of time and cost required for immediate re-installation during disasters.

9.2 Future Scope

In this section we list the things that were either left open by this project or were opened by the analysis performed and the lessons learned during our interaction with the subject.

→ In the developed system, Auxiliary Relief Panel is used to control two main routes but the ARP/EP can be extended to control all routes available in station .In some of large station, where there is major requirement to control more number of routes when electronic interlocking fails, can be achieved by adding additional hardware module to current system with their corresponding software changes.

→ Hence, Routes on Graphical Interface can also be created dynamically according to number of routes specified by user.

→ The current ARP/EP uses RS-232 communication between PC and embedded system .Other communication protocol such as Ethernet can be implemented to provide noise free and wireless communication.

REFERENCES

[1] A Tool for Automatic Formal Modeling of Railway Interlocking Systems, Muhammed Ali Nur Oz, Ibrahim Sener, Yildiz Technical University Istanbul / Turkey

[2] About Interlocking on Wikipedia https://en.wikipedia.org/wiki/Interlocking

[3] About serial communication RS-232 in C Sharp Winforms-https://msdn.microsoft.com/ en-us/library/ system.io.ports. serial-port(v=vs.110) .aspx

[4] About RS-232 communication basics- https://www. electronicshub.org /rs232-protocol -basics/

[5] Handbook on Railway Signalling in Indian Railways http://www.rdso. indianrailways. gov.in

[6] About Signalling in Indian Railways- http://www.railway- technical.com/ signalling/

[7] About Indian Railway-https://en.wikipedia .org/wiki /Indian_Railways