DESIGN OF A CENTRALISED RADIO TRAIN CONTROL SYSTEM.

Clive Patrick Barker.

A Dissertation submitted to the Faculty of Engineering, University of the Witwatersrand, Johannesburg, for the Degree of Master of Science in Engineering.

Johannesburg 1985.

SYNOPSIS

The dissertation describes the development of a control system for the safe, reliable and efficient control of all train movements in a railway network of a particular configuration. The network consists of a single main line connecting a number of stations. At each station one or more loops exist which are used as train crossover points. The network can carry trains in both directions and the system aims to manage the movement of trains in such a way as to protect against collisions, whilst maximising the traffic capacity of the network. The system allows for the transit of trains from one station to another, the crossover of trains at crossing points and the shunting of trains at a station.

Traffic management decisions for the entire network are made in a single central control room which is situated in the network. Decisions made by a Traffic Controller are transmitted as 'Train Movement Instructions' to trains via a radio link, and a Movement Instruction, which is displayed to the train driver in his cab, gives the driver authority to proceed as instructed system, called 'Centralised Radio Train Control' or C.R.T.C., does not make use of any lineside signalling equipment or train detection equipment. The radio link is used for the transfer of both speech and data (electronic) messages between train drivers and the control room. Emphasis is placed on the safe operation of the system at all levels of the system design and operation.

DECLARATION

I declare that this dissertation is my own work. It is being submitted for the Degree of Master of Science in Engineering in the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination in any other University.

C.P.BARKER

25th day of March, 1985.

PREFACE

TO ENGINEERS AND IRON FOUNDERS.

THE DIRECTORS of the LIVERPOOL and MANCHESTER RAILWAY hereby offer a Premium of £500 (over and above the cost price) for a LOCOMOTIVE ENGINE, which shall be a decided improvement on any hitherto constructed, subject to certain stipulations and conditions, a copy of which may be had at the Railway Office, or will be forwarded, as may be directed, on application for the same, if by letter, post paid.

HENRY BOOTH, Treasurer.

Railway Office, Liverpool, April 25, 1829.

From 'HERITAGE OF BRITAIN', Readers Digest Association, 1975.

On an October day in 1829 in a field just outside Liverpool, Great Britain, George Stephenson won a prize of 500 pounds after demonstrating his revolutionary 'rocket', the culmination of 20 years of hard work on the challenge of steam locomotion. The 'rocket' achieved what would seem, by modern standards, a modest accomplishment. It covered a distance of 35 miles at an average speed of 15 miles per hour, and it proved to have a maximum unloaded speed of 35 miles per hour. That day, however, heralded the start of a new age of transportation - 'The Railway Age'.

Railway transportation after 1829 very quickly became economical, fast and convenient. It soon reached the stage where it became necessary to regulate train traffic, and as trains became larger, faster and more numerous the traffic control systems became more complicated and more sophisticated, giving rise to a new discipline, that of the Railway Signal Engineer.

In 1876 an event occurred on the Great Northern Line in Great Britain which initiated the concept of 'Fail-Safety'. A train accident was caused by the failure of a number of mechanical signals. The signals had become laden with snow and were stuck showing a 'proceed' indication to trains which should have stopped. In the resultant collision, which involved a coal train and two passenger trains, 13 passengers were killed and many more injured. It was made abundantly clear by this incident that signals would fail from time to time, and that a philosophy of 'Fail-Safety' should be adopted to prevent further occurrences of such tragedies. In this regard, a new mechanical signal was introduced which was guaranteed to automatically display a 'stop', or 'danger' indication in the event of a failure in any of its mechanical components.

Railway Signalling has grown, with Railway Transport itself, into a huge industry and the concept of 'Fail-Safety' has dominated the signalling industry in all aspects of technical design. Today, the microprocessor has become readily and cheaply available and has been identified as having great potential in the implementation of Railway Signalling systems. The microprocessor will inevitably impinge more and more onto the day-to-day control of train traffic, but cheap and convenient though it may be, microprocessor controlled signalling systems have associated problems in terms of safety. The microchip is such a complex device that failures tend to have largely unpredictable results. New philosophies and new design strategies have had to be formulated to achieve the required level of safety.

There are a number of microprocessor-controlled railway signalling installations operative in the world today and the general trend towards the achievement of safety has been to duplicate (or triplicate) hardware to give a degree of redundancy (see Chapter 7). One such system was developed by Dr. J.S.D. Lambrechs of Telkor Signalling (Pty) Ltd. with the original application being for a Fail-Safe Data Transfer System, specifically designed for use in Railway Signalling applications. The control of Single Line Networks was seen as an area of signalling which could benefit from the application of Fail-Safe microprocessor technology, and hence the Centralised Radio Train Control system was devised and designed as described in this dissertation.

The future holds the potential for large scale development work in the field of Railway Signalling. Microprocessor control of vital functions is a concept which is only now becoming acceptable to transport authorities, but after the initial installations of cheap, feasible and safe systems the opportunities will multiply rapidly and the scope for microprocessor based signalling systems will be unlimited.

ACKNOWLEDGEMENTS.

The Author wishes to thank the *following:*

Telkor Signalling *Company* (PTY) Ltd. for promoting the *research.*

All those people from the South African Transport Services who were consulted and who gave freely of their advice.

The University of the Witwatersrand for the use of facilities.

Mr. G.B. Anderson, Managing Director, Telkor Signalling *for* his support and encouragement.

Professor M.G. *Rodd, University* of the Witwatersrand, for his *continued* interest in the Telkor Research Group.

Dr. J.S.D. Lambrechs for his council, guidance and unfailing enthusiasm, and for his confidence in the author.

Mr. N. Frost and the staff of the Computer Aided Draughting Division, Telkor, for help with the production of the drawings.

His colleagues in the Telkor Research *Group* for their patience and cooperation.

His wife for her understanding and inspiration.

CONTENTS

SYNOPSIS .... (i)
DECLARATION .... (ii)
PREFACE .... (iii)
ACKNOWLEDGEMENTS .... (vi)
CONTENTS .... (vii)

INTRODUCTION 1

CHAPTER 1 - Operating Regulations

1.0 What is Single Line Control? 7
1.1 Absolute Working 8
1.2 Permissive Working 8
1.3 Interworking 9
1.4 Subsidiary Working 10
1.5 C.R.T.C. Working 10
1.6 Conclusion 13

CHAPTER 2 - A History of Single Line Control Systems

2.0 Introduction 14
2.1 Timetabling 14
2.2 The Token System 15
2.3 Communication - Radio 17
2.4 Radio and Electronics - Centralised Control 17
2.5 Location Identification and Control 19
2.6 Electronic Token 20
2.7 Safety Considerations - LIC and ET 20
2.8 Centralised Radio Train Control 21
2.9 Conclusion 22

CHAPTER 3 - General Specification for C.R.T.C.

3.0 Introduction 23
3.1 Movement Instructions 26
3.2 The Communication Procedure 27
3.3 The Equipment Used to Implement the System 31
3.4 Conclusion 36

CHAPTER 4 - Detailed Description of C.R.T.C. Equipment

4.0 Introduction 37
4.1 Safety 38
4.2 A Data Entry Interface for th 39
4.3 The Data Entry Interface in C. 42
4.4 The Data Display Interface 44
4.5 The Overview Display Interface 47
4.6 The Train Driver's Interface 51
4.7 C.R.T.C. Operation 53
4.8 Conclusion 56

CHAPTER 5 - C.R.T.C. Transactions

5.0 Introduction 57
5.1 Secondary Transactions 58
5.2 Primary Transactions 60
5.3 The 'ABORT' Function 75
5.4 Language 76
5.5 Conclusion 76

CHAPTER 6 - C.R.T.C. Interlocking

6.0 Introduction 77
6.1 General Rules 78
6.2 Secondary Transactions: Train Enquiry, Setting and Clearing of Blocking Crosses 79
6.3 'Train Initialisation' Transaction 79
6.4 'Exit from C.R.T.C.' Transaction 81
6.5 'Issuing of a Proceed Authority' Transaction 82
6.6 'Train Clear of Station' Transaction 87
6.7 'Entry into Destination Station' Transaction 88
6.8 'Train Complete' Transaction 89
6.9 'Extension of Proceed Authority' Transaction 90
6.10 'Train Stop' Transaction 90
6.11 'Cancel Proceed' Transaction 91
6.12 'Issuing of a Shunt Authority' Transaction 91
6.13 'End of Shunt' Transaction 93
6.14 'Change Train Number' Transaction 93
6.15 The Implementation of Interlocking Rules 94
6.16 Conclusion 95

CHAPTER 7 - The Concept of Safety

7.0 General Safety Principles 96
7.1 Error Sources in C.R.T.C.: Specification Errors 100
7.2 Error Sources in C.R.T.C.: On-Line Errors 101
7.3 Safety in C.R.T.C. 106
7.4 Conclusion 107

CHAPTER 8 - Technical Details of C.R.T.C.

8.0 Introduction 108
8.1 Implementation - Software 108
8.2 The Design Procedure 110
8.3 C.R.T.C. Simulator Software 111
8.4 Modularity and Modification 114
8.5 Software Validation 117
8.6 Conclusion 117

CHAPTER 9 - Conclusion

9.0 The End Users of the System 118
9.1 The Achievements of the Project 119
9.2 A Production Level C.R.T.C. System 119
9.3 The Present Status of the System 121
9.4 Extensions to the System 122
9.5 The Future 122

REFERENCES 123

BIBLIOGRAPHY 126

APPENDICES :

APPENDIX A - Configuration of a C.R.T.C. Layout A.1

APPENDIX B - Description of Modules Linked into C.R.T.C. B.1

APPENDIX C - Major Data Structures C.1

APPENDIX D - Operator's Interface Software - General Status Descriptio D.1

INTRODUCTION.

"Centralised Radio Train Control", C.R.T.C., is concerned with the control of trains travelling in both directions through a Railway Network in which stations are connected by single lines. Single line connections are often used in rural networks where it may be impractical to have two lines, particularly where long distances and low traffic densities are involved. Instead, a single line is used and each station along the route has one or more loops which are used for trains *crossing* thereby allowing bidirectional train travel (refs. 17,20). Figure I.1 represents a single line network, showing a number of stations and crossing loops.

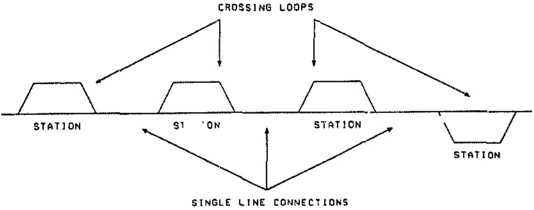


Figure I.1: The Single Line Network.

In single line systems, where it is required for trains to travel in either direction along a single track, it is imperative to manage train traffic so that two trains do not enter the same section simultaneously, giving rise to the

INTRODUCTION.

"Centralised Radio Train Control", C.R.T.C., is concerned with the control of trains travelling in both directions through a Railway Network in which stations are connected by single lines. Single line connections are often used in rural networks where it may be impractical to have two lines, particularly where long distances and low traffic densities are involved. Instead, a single line is used and each station along the route has one or more loops which are used for trains crossing thereby allowing bidirectional train travel (refs. 13,20). Figure I.1 represents a single line network, showing a number of stations and crossing loops.

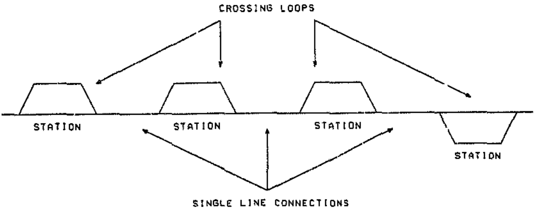


Figure I.1: The Single Line Network.

In single line systems, where it is required for trains to travel in either direction along a single track, it is imperative to manage train traffic so that two trains do not enter the same section simultaneously, giving rise to the possibility of a collision (refs. 13,17). The management of train traffic in this type of network can be achieved by formalising a set of operating regulations, which can be implemented by a control system. The complexity and sophistication of the control system will depend on the traffic density, the number of loops, and the speed of trains in the network.

What essentially must a control system achieve? Figure I.2 shows a single line section connecting two stations, Station A and Station B. Each station in the example has a single crossing loop, and in each station a train is staged on the main line (Train 1 in Station A and Train 2 in Station B.) Train 1 is due to travel to Station B, and Train 2 is due to travel to Station A. In order to avoid a collision, only one of the trains may enter the section, and the other train must wait until the first train has cleared the section. This would constitute a typical operating regulation. A single line control system must decide which train should leave first and ensure that each driver knows whether he should wait or continue.

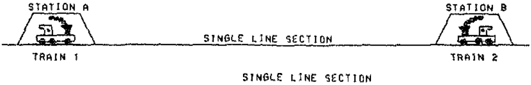


Figure I.2: A Single Line Connection.

Operating regulations for single line traffic, the principles of which are described in Chapter 1, tend to be common to most of the control systems which have evolved throughout the history of single line signalling systems. These regulations generally form the basis for control systems regardless of the sophistication of the control system used to implement them. Chapter 2 outlines the history and development of some of these control systems, placing C.R.T.C. in an historical context, and describes two similar projects which have recently been developed in other parts of the world namely British Rail's "Electronic Token" (ET) and British Columbia Rail's "Location Identification and Control" (LIC) (refs. 3,4,16,20).

The motivation behind recent developments, and that behind the development of C.R.T.C. has been one of cutting operating costs. It has become feasible to use electronics for train control, as exemplified by the fact that solid state systems have been developed, and are in use in several fields where safety is of utmost importance. Cheap, reliable electronic equipment (specifically digital electronics and microprocessors) coupled with the use of radio communication to transmit electronic "messages" have allowed the control of large, single line networks to become centralised (refs. 6,8,11,14,16). This represents a significant saving in running costs by eliminating the need for trackside signalling equipment and lengthy cabling as well as reducing the number of operating personnel required.

Many single line networks, particularly in rural areas where passenger traffic is light, are under threat of closure because of the expense of running them under the present control systems. These can be continued, and run in a safe and efficient control environment by restructuring the train control systems and minimising running costs.

In C.R.T.C. a single operator, working from a control room, is able to communicate by radio with any train driver in the network under his control as shown in Fig I.3. The controller uses the radio link to give each driver his movement instructions as and when the train is to move. Each instruction is given both verbally and by a digitally encoded message, and microprocessor supervisory equipment ensures that no conflicting "Movement Authorities" are issued.

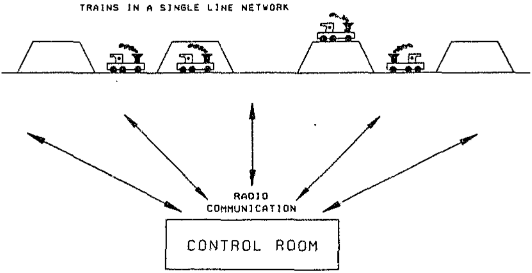


Figure I.3: Radio Communication.

The operator is given various tools to aid in the management of traffic in the network: a microprocessor system which enables the instructions to be digitised and sent in electronic form to the train (where each instruction is displayed to the driver to give him a clear, unambiguous record of the instruction); a specially designed interface which prevents the operator from attempting to set up a movement instruction which would be rejected by the safety check equipment; and a graphic display of the entire network reflecting the position of all stations and crossing loops, the presence of trains, and the current status of each train regarding the driver's movement authority. The structure of the system is summarised in Chapter 3 (a detailed description of the components of C.R.T.C. is given er 4) Chapter 5 enumerates all the possible "instructio which train drivers can be given. These instructions combine to build up a full management system, and are communicated between the C.R.T.C. operator in the control room and train drivers in the field via a fixed protocol which is described in Chapter 3.

Each instruction transmitted to a train driver involves some form of train movement. Once a driver has been given a movement authority, it is necessary that his train be protected, and that no other driver be given an instruction which may result in confliction. Hence each instruction given to any train driver must be interlocked with all existing instructions and the mechanism of this interlocking is discussed in Chapter 6.

C.R.T.C. is a complex system and, like any system which controls the train traffic, must be Fail-Safe: that is a failure of any part of the control system must never lead to the occurrence of a dangerous situation. A failure of the control system can be defined as any deviation from the specified operation of the system, whether it be an equipment failure or human error. A dangerous failure is one which leads to the possibility of a direct infringement of the safety regulations governing train movements. Failures can never be entirely eliminated, but it is the responsibility of the control system to minimise the occurrence of dangerous ones. Malfunctions in C.R.T.C. can arise from several sources: operator error, data corruption during transmission, misinterpretation by the driver etc. Chapter 7 describes all these sources of error in some detail, and shows how the system protects against the errors causing unsafe situations.

The specific aim of the project was to develop a specification for an inexpensive single line control system which was acceptably safe and easy to operate and which would enable the efficient management of train traffic in a network. The resultant system, C.R.T.C., was designed after lengthy consultation with various signalling experts, and part of this consultation involved the demonstration of a physical implementation of C.R.T.C. with a simulation of some of the aspects of the system. Chapter 8 describes the technical details of the simulation, which is in essence an operator's workstation.

Chapter 1: Operating Regulations

1.0 What is Single Line Control?

A single line network consists of a number of stations, or crossing places, as illustrated in Fig 1.1. At each station, two trains (i.e an 'up' train travelling in the 'up' direction and a 'down' train travelling in the 'down' direction) can cross over if one of the trains waits in the loop while the other passes through on the main line (or vice versa). The network can include many such stations, and each station may have one or more crossing loops. The problem lies in regulating the movement of the trains such that collisions are prevented and there are certain basic operating regulations which are used *collectively or individually* to achieve this.

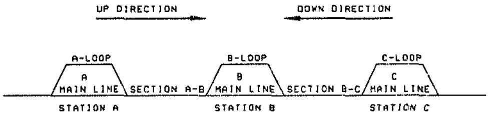


Figure 1.1: Single Line Nomenclature.

Most control systems are based on some or all of the following working procedures described in the following paragraphs (ref. 13).

1.1 Absolute Working.

In this working procedure only one train is ever allowed into a section at a time (see Figure 1.2). This is strictly enforced, and applies to all trains regardless of whether they are shunting, proceeding to a destination, or stationary in a section.

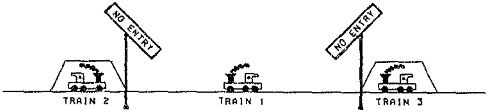


Figure 1.2: Absolute Working.

1.2 Permissive Working.

Here, one or more trains are allowed into a section provided that they are all travelling in the same direction. They must proceed at fixed, regulated time intervals and maintain approximately the same constant speed (see Figure 1.3). Permissive Working (as the name implies) is a method which relaxes the more stringent regulations specified in Absolute Working. Safety is sacrificed to a certain extent in favour of flexibility and Permissive Working is generally used to improve traffic flow where long distances separate crossing places which would tend to become bottlenecks under Absolute Working Regulations.

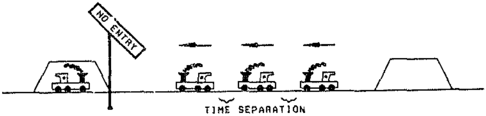


Figure 1.3: Permissive Working.

1.3 Interworking.

This method makes use of 'interloops' to enhance the traffic flow through long sections of line. An interloop is an unmanned crossing place which is normally constructed at some point along a section. When the section between two stations includes an interloop (see Figure 1.4) two trains may enter the same section, one from either end. Each driver is aware that a second train is approaching, and is instructed to stop and wait at the interloop until the crossing has occurred. Like Permissive Working, Interworking is used to enhance traffic flow.

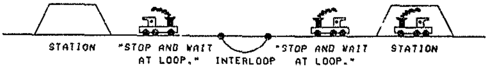


Figure 1.4: Interworking.

1.4 Subsidiary Working.

These regulations cover the situation where the section between two stations includes a siding (see Figure 1.5). This could occur, for example, where a private subscriber is serviced by the single line and trains entering the section may exit the controlled area via the siding. There must be some form of communication between the train driver and operating personnel at one of the stations to ascertain the position of the train so that the section can be 'cleared' for use by another train when the first train has entered the siding.

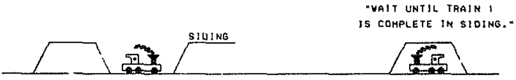


Figure 1.5: Subsidiary Working

1.5 C.R.T.C. Working.

The working regulations described above constitute the rules which must be followed to ensure safe operation for single line networks. C.R.T.C. is an implementation of Absolute Working regulations and strictly enforces the rule of only one train per section. Because C.R.T.C. is based on centralised control, stations are not manned, and all control is initiated from the single control room. This means that 'interloops' and 'sidings' can be catered for - by designating them as 'stations'.

For example, where a single section includes an interloop (or a siding), the interloop would become, under C.R.T.C., a station and the section would be broken into two sections (see Figure 1.6). This means that traffic congestion can be kept to a minimum without sacrificing the inherent safety associated with Absolute Working.

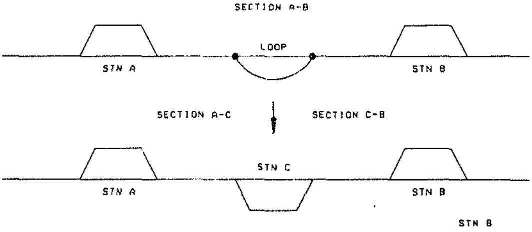


Figure 1.6: "Interworking" in C.R.T.C.

control stations are not manned, and all control is initiated from the single control room. This means that 'interloops' and 'sidings' can be catered for - by designating them as 'stations'.

For example, where a single section includes an interloop (or a siding), the interloop would become, under C.R.T.C., a station and the section would be broken into two sections (see Figure 1.6). This means that traffic congestion can be kept to a minimum without sacrificing the inherent safety associated with Absolute Working.

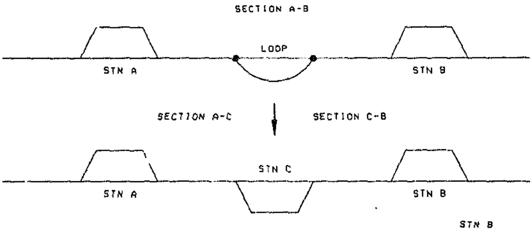


Figure 1.6: "Interworking" in C.R.T.C.

For long sections with no 'interloops', and where traffic levels would normally require the use of Permissive Working, a C.R.T.C. system would break up the long section into two or more sections, with 'stations' in between (see Figure 1.7). Each station would consist of a *main line* but *no crossing loops*. Naturally, crossings could not be made in these apparent stations, but trains following in the same direction would make use of the multiple sections to simulate a 'Permissive Working' situation without deviating from Absolute Working regulations. Each apparent station would be marked in the field, by marker boards which would indicate to the driver that he has reached the *end of a 'section'*, and is required to stop, or to obtain further authority to enter the next section.

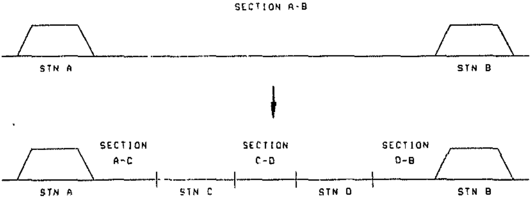


Figure 1.7: "Permissive Working" in C.R.T.C.

1.6 Conclusion.

The operating procedures described in this chapter (Absolute Working, Permissive Working, Interworking and Subsidiary Working) have provided the basis for single line control throughout the development of most control systems. C.R.T.C. is a direct implementation of Absolute Working in principle, but has the inherent flexibility to include some of the advantages which, under conventional control systems, would require the special conditions of Permissive Working, Interworking and Subsidiary Working.

Chapter 2: A History of Single Line Control Systems.

2.0 Introduction.

The *Working Regulations* described in the previous chapter have been the basis of various control systems which have developed throughout the history of train travel. The ever-increasing density of train traffic and the speed of train travel have dictated that these systems be upgraded and changed continually, making use of new technology and techniques. C.R.T.C. is one of the most recent developments along these lines and aims to increase the efficiency of single line control, whilst reducing the cost of maintaining single line networks. Safety has always been of primary concern in all Railway Signalling systems, and safety has been one of the major criteria behind the development of C.R.T.C.

The following Chapter traces aspects of the evolution of single line control systems from which C.R.T.C. is in part derived.

2.1 Timetabling (ref. 1[illegible]).

An early solution to the problem of single line control (and one which is still in use) involves the regulation of train movements by a fixed timetable. The movement of each train is governed by a timetable which dictates details of when the train may proceed, wait or cross over. The timetable is carefully worked out to avoid confliction between different trains. This method, however, has the potential for huge delays which ripple through the whole system if a single train fails to adhere to the timetable. Furthermore, any instance of departure from the timetable (for example if a train breaks down and is delayed in a station) *involves the dynamic* restructuring of the timetable, thereby increasing the potential for error. This method is obviously untenable except for systems having very low traffic levels since with the increase in traffic density comes also an increase in the frequency of errors.

2.2 The Token System (refs. 13,17,20).

Implementation of Absolute Working can be achieved very simply by designating a single, physical 'token' for each single line section. A train may only enter the section if the driver has in his possession the token for that section- thereby ensuring that only one train can be in a section at any one time.

The 'token' was originally represented by a wooden staff which would be carried up and down the section by alternate up and down trains. In the case where two following trains had to travel consecutively through a section, then (for Absolute Working) the staff had to be returned to the original end after the passage of the first train by some other means - usually by a '*runner*' *or a cyclist* employed for that purpose.

Permissive Working was also achieved using the wooden staff token system. Here a paper 'ticket' was issued to the train driver and he was shown the staff as proof that the other end of the section he was about to enter was secure. Only the last of a series of following trains was given the staff for transfer to the other end of the section.

The 'token' system provides the basis for single line control and has been refined over the years where necessary. Telegraphic communication between stations has greatly increased the efficiency of the physical token system and allows for the use of multiple tokens (only one token to be allowed 'out' at a time). This is achieved by regulating the issuing of tokens via 'Token' or 'Block' Instruments which are situated at each end of a section. The release of a token at one end causes electrical signals to be transmitted to the opposing instrument which then locks itself against further token issue until the token has been inserted in the second instrument, signifying that the section is now clear. The Block Instrument is operated by the Station Master, and various methods have been devised whereby the Station Master can exchange tokens with the driver 'on the fly' i.e. without stopping the train. The Station Master thereby receives the token from the previous section, and issues the token for the following section.
control centre. Processing equipment supervised by a human operator at the control centre, prepares 'Control Point Instructions' based on information about all trains. A

2.3 Communication - Radio (ref. 20).

As discussed in the previous section, communication was used to enhance the token system allowing multiple tokens for a section and obviating the need for a 'runner' to carry messages and return tokens from one end of a section to the other. The disadvantages of communication by electrical cable were emphasised on the Inverness-Wick line in the North East of Scotland where traffic was totally disrupted when a severe storm in 1978 destroyed the communication cables. This caused immense delays, and it was decided to replace the cables with a radio network. This gave the added advantage that train drivers could be contacted en-route. The mechanical token system was not replaced, however, but a fail safe microprocessor system was developed by British Rail to interface the existing Block Instruments to Radio Transceivers. Control was not yet centralised and was still essentially implemented between stations.

2.4 Radio and Electronics - Centralised Control (refs. 3,12,20).

The Inverness-Wick 'Radio Block' system illustrates two significant developments in the field of Railway Signalling:

1)Radio communication permits the transfer of information directly to moving trains from a central control area;

2)Microprocessors can be configured to give acceptably safe processing to enhance the efficiency of train control.

These two factors give rise to the feasibility of centralised control and electronic 'Cab Signalling'. The control of single line networks has evolved from the days of blanket timetable operation with little or no communication occurring between different geographical parts of a system "in real time", i.e. as and when events occur during operation, to a more concentrated system with close communication between adjacent stations. The next logical step is towards fully centralised control, in which communication occurs from a central 'operating room' directly to trains in a network with many stations and single line sections. The radio technology facilitates this type of communication - and electronic technology provides the complex decision-making ability required for multiple sections, the high speed processing required for faster, more dense traffic, and the ability to display 'tokens', or 'movement authorities' directly to the driver in his cab - the concept of 'Cab Signalling'.

Technology may not yet be equal to the next step - total automation - where the 'signals' transmitted to the train control automatic 'train driving' equipment, although a number of driverless transit systems do exist in simple, unidirectional systems. Recent projects ,however, are a step in this direction. Some of these projects include British Rail's Electronic Token system, British Columbia Rail's Location Identification and Control system and, the subject of this dissertation, Centralised Radio Train Control, all of which make use of the concept of Cab Signalling.

2.5 Location Identification And Control (ref. 4).

Location Identification and Control (LIC) integrates a complex set of train detection equipment with cab signalling, centralised control, and a microwave communication link into a sophisticated system which can be applied to networks beyond those of single line complexity. The LIC system is presently being developed in British Columbia for areas in which traffic levels are relatively high, and the more conventional 'paper order' system in which drivers are given full instructions on paper (a modified 'timetabling' system) can no longer cope. Detailed information such as speed, location and last carriage detection are transmitted to the control centre through the use of trackside 'transponders'. A transponder is a piece of electronic equipment which, when activated by a passing train, responds with a fixed message (such as its position, distance to the next transponder etc). On- board train processing equipment then processes the message to calculate speed and position, which are transmitted from the train back to the control centre. Processing equipment supervised by a human operator at the control centre, prepares 'Control Point Instructions' based on information about all trains. A

Control Point Instruction is a message which is transmitted to a train driver and displayed in his cab and which constitutes his authority to proceed into a section.

2.6 Electronic Token (refs. 3,12,20).

The 'Electronic Token' system (ET), a project sponsored by British Rail, has been installed on a line in Scotland, and is running in parallel with a mechanical token system. ET is simply a computerised version of the token system with centralised control. Equipment mounted in the cab of the train gives the train driver a 'receive' and a 'send' button which he must actively depress in order to receive or to return tokens. His train position is established by means of verbal communication over the radio with the central despatcher, there being no trackside detection equipment. The despatcher is given a keyboard and monitor interface which gives him a display of the full network and the positions of all the trains, and aids in the management of the traffic.

2.7 Safety Considerations: LIC and ET.

There is little available information concerning safety for LIC, but ET bases its operation on the 'Absolute Working' Method in which a section may only carry one train at a time. The integrity of the token issue procedure is ensured by a computer in which three microprocessors run in parallel, and a majority vote of two out of three microprocessors is required before decisions are allowed. The three 'programs' running in the triple computer are all the same and have undergone an extensive 'validation' procedure before being installed. Through this procedure attempts are made to ensure the correct operation of the system and its compliance with the 'Absolute Working' regulations for all operating conditions.

While the process of validation is an attempt to eliminate software errors, hardware errors are handled by redundancy. Since the hardware is triplicated, a single error can be 'detected' by the other two processors, the offending hardware isolated and notification of a failure indicated.

2.8 Centralised Radio Train Control (refs. 1,2,19).

Since C.R.T.C. is the subject of this dissertation, it is pertinent to discuss it briefly in the context of these other developments.

Unlike LIC, C.R.T.C. does not require complex and expensive trackside equipment. Verbal communication between the traffic controller in the control room and the train drivers serves to establish train locations. In C.R.T.C. on board train processing equipment is used only to receive and transmit messages initiated by the operator and the driver. No train speed information is required as in LIC, making C.R.T.C. a cheaper, less sophisticated system.

C.R.T.C. and ET were developed along similar lines and share certain concepts, particularly those concerned with ensuring safety. However, compared with ET, C.R.T.C. is a more comprehensive management system with a formal, strictly defined procedure for combining the transfer of verbal and data information between the train driver and the controller. Unlike ET, which provides simply for the transfer of tokens, C.R.T.C. allows for the issuing of a full range of movement authorities, allowing shunting, the temporary closure of sections, the cancellation of proceed authorities and the issuing of multiple movement authorities.

2.9 Conclusion.

The traffic control systems described in this chapter have evolved as a result of technological advances (i.e. the telegraph, radio, electronics and the micro processor) which have each enabled improvements to be made in running costs, safety and/or efficiency. In C.R.T.C. a major objective was to incorporate modern technology only where it would provide real, tangible advantages. The following chapters describe the equipment used in C.R.T.C. and illustrate the significant advantages which the use of such equipment facilitated.

Chapter 3: General Specification for C.R.T.C.

3.0 Introduction.

The objective of C.R.T.C. as a control system is the safe, efficient and cost effective management of train traffic in a single line network.

The format of C.R.T.C. is illustrated in Figure 3.1 and comprises a central control room manned by a single operator in radio communication with all trains in the network under his control.

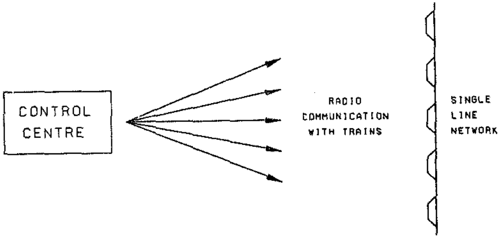


Figure 3.1: C.R.T.C. Format.

The centralisation of control has a number of advantages. Compared with other conventional systems, operating costs in C.R.T.C. are reduced since centralising the control system obviates the necessity for each station to be manned, saving on personnel costs. All traffic control emanates from the single control room, and instructions are transmitted directly to train drivers. With the 'cab signalling' approach, elements such as signals in the field are not required. This represents a saving in both installation and maintenance, and eliminates the need for lengths of trackside cabling.

Efficiency too is significantly enhanced by the centralisation of control. The traffic controller is given reliable, up to date information concerning the trains in the network under his control, and since this information is presented to him in a clear, concise way, he is able to manage train movements, and make his decisions on the basis of the global information available to him. This allows the controller to plan train movements ahead of time and C.R.T.C. is sufficiently flexible to enable the controller to change timetables dynamically to suit traffic conditions which arise during the course of operations. The controller is also able to cater for extraordinary situations, such as broken-down trains, and to reorganise traffic patterns in order to minimise delays to other trains.

C.R.T.C., therefore, allows the controller to handle traffic in the most efficient way possible. Moreover, a further advantage associated with centralised control is that since all decisions emanate from one source, they can be automatically recorded and used to build up a Traffic Flow Summary which can be used as an aid to further planning, or as a 'Log of Events' to be consulted in the event of an accident and used to reconstruct and analyse malfunctions.

In C.R.T.C. the control room is the centre of operations. Train traffic is managed from the control room and from there instructions are transmitted to trains regarding the resulting movement authorities. Train drivers communicate information concerning the status of their trains (i.e. the location of the train, the entry of the train into the network, etc.) from the field to the control room where the operator collates the information and then uses it as a basis for movement decisions.

The system therefore involves the communication of certain messages between the control room and the cab of a train, and both the operator and the driver are provided with equipment to facilitate this communication.

Hence, the general specification for C.R.T.C. includes:

1)A definition of the format of movement instructions;

2)A description of the procedure to be followed for communication between a train driver and the traffic controller;

3)An overview of the equipment used in the control room and the train for the implementation of the system.

3.1 Movement Instructions.

The basis for the operation of C.R.T.C. is Absolute Working, as described in Chapter 1, and all the messages which can be sent to a train driver are based on this premise. The full control of a network is accomplished by using the following types of messages:

1)Permission to enter the network (this is referred to as 'Train Initialisation');

2)Permission to proceed to a destination station (i.e. permission to enter a section or sections);

3)Acknowledgement of the clearance from a section (effectively withdrawing the authority from the driver to be in the section in order to allow a second train to enter the section);

4)Permission to shunt from a station into a section (i.e. permission to enter a section for the purpose of shunting);

5)Acknowledgement of a completed shunt (i.e. clearance of the section used for shunting);

6)Cancellation of a proceed authority (i.e. withdrawing the authority for a section before the driver has entered the section).

There are several other C.R.T.C. instruction types - such as changing a train number - these, along with the types mentioned above, will be discussed in detail in chapter 4.

3.2 The Communication Procedure.

In order to formalise the procedure by which the train driver and controller collaborate in the issuing of movement instructions it is necessary to define the protocol by which they communicate. This protocol is fixed and clearly structured in order to make the format consistent.

The Radio network is configured to allow 'selective calling' whereby the operator can set up a channel between the control room, and a specific train, masking out all other trains. Train drivers are able to request a contact to be made by transmitting a snort, out of band, request burst. A pending request is displayed to the operator, who then sets up a channel to communicate with the requesting train driver.

When a channel has been established there is a communication link across which both speech and data can be transferred between the control room and a train. A C.R.T.C. 'transaction' is the procedure by which a train driver is given a new instruction, and a transaction (or series of transactions) is always initiated by the channel set-up procedure.

The system does not make use of any automatic train detection equipment, and it is the responsibility of the driver to convey verbally any information concerning the locality of his train to the operator. An early specification for C.R.T.C. called for the driver to indicate this information by means of a complex set of pushbuttons in the cab. This, however, would place a burden on the driver, possibly detracting from his primary function - that of the safe driving of the train, and it was decided that all such information should be communicated verbally. The availability of a free speech communication facility further ensures that:

1)Full, unambiguous information is available to the operator;

2)The driver can communicate quickly and easily without the distraction of pushbutton selection;

3)Abnormal or emergency situations can be rapidly communicated.

Communication in C.R.T.C. comprises speech and data, the data communication serving to confirm any instructions transmitted verbally, and also enabling the driver to be given a visual electronic display of any current movement authority. The communication procedure, therefore, combines the transfer of both speech and data information between the control room and the train resulting in the issuing of an instruction to the driver. The driver is given a single pushbutton with which to acknowledge, and give his consent to any transaction, and the controller is required to enter certain information about the train at a data entry terminal interface.

A full transaction consists of the following sequence of activities:-

1)The operator sets up the radio channel to the relevant train (if it is not already set up from a previous transaction);

2)The operator is prompted (at his data entry interface) for information concerning the type of transaction required, and any information about the train (its position etc.) which may be required for the specified transaction;

3)The driver verbally gives the operator the required information;

4)The operator enters the information into the data entry interface;

5)A formal verbal authority is displayed to the operator who is required to read it to the driver over the radio;

6)The driver may, if he agrees, acknowledge the message verbally. If the driver does not accept the transaction, the operator aborts it;

7)If both the driver and the operator are satisfied with the verbal transaction they both press 'execute' pushbuttons (the driver is supplied with a single pushbutton for this purpose and the operator uses his data entry interface);

8)While both buttons are depressed, a data message is transmitted across the radio channel and electronically acknowledged by the receiver equipment on the train;

9)The driver receives an electronic display of the full text of the instruction which now constitutes his movement authority. The authority will be overruled only at the next transaction, and this again must receive the consent of the driver using his pushbutton.

If, at any stage during a transaction, either the driver or the operator is not satisfied that the information is correct, the transaction can be aborted and no authority will be deemed to have been given. The transaction is defined to have been completed only when the data transfer has occurred and authorities are only valid from this point. If, at the conclusion of a transaction, the electronically displayed message does not agree with the verbal message received by the driver, he must immediately inform the controller, and until the error has been rectified he must consider the displayed message invalid.

3.3 The Equipment used to Implement the System.

In order best to implement the specification as generally outlined in this chapter, C.R.T.C. features control room equipment as shown in Figure 3.2.

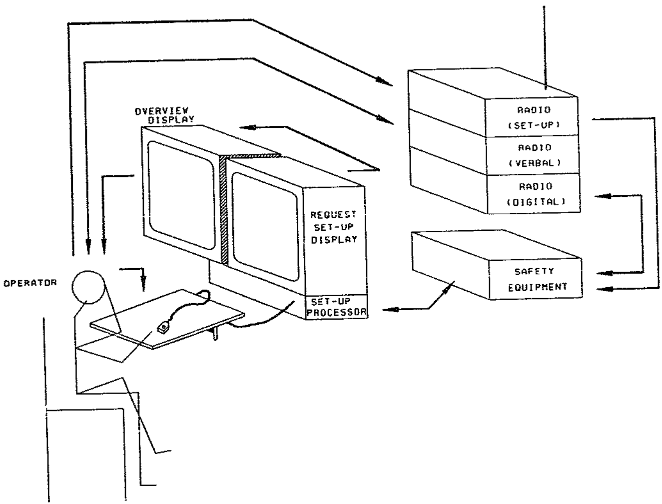


Figure 3.2: Control Room Equipment.

As can be seen from Figure 3.2, the control room equipment comprises:-

1)Radio Equipment with speech and data interfaces;

2)Safety Check Equipment;

3)Overview Display;

4)Data Setup Monitor.

The operator is notified of outstanding requests for communication via an interface with the radio equipment. This interface can also be used to set up communication with any train in the system. The radio, once set up to communicate with a train, allows free speech between the operator and train driver and also has a data interface which allows data to be transmitted to and received from the train. The data is routed through microprocessor 'safety check' equipment which ensures that no unsafe messages are transmitted, and that the correct acknowledgement is received from the train. The data is generated from a monitor and data input device which is used by the controller to set up train movement instructions prior to the data transmission portion of the communication protocol. The entire status of the network, and of all trains in the network is displayed to the operator by a colour overview display monitor, which gives the operator information upon which to base decisions regarding train movements.

Equipment mounted in the cab of the train is shown in Figure 3.3.

The train-borne equipment consists of the following:-

1) Radio with speech and data interfaces;

2) Processing Equipment;

3) Single Line Display Unit;

4) Single Pushbutton.

The radio, which has both speech and data interfaces, enables the driver to communicate with the operator along the lines specified in the communication protocol. Data messages received from the control room are interpreted by a microprocessor which controls the display unit for the display of the current instruction to the driver. This microprocessor equipment also manages the data communication, transmitting back to the control room an acknowledgement of each message, and the state of the driver's 'execute' pushbutton.

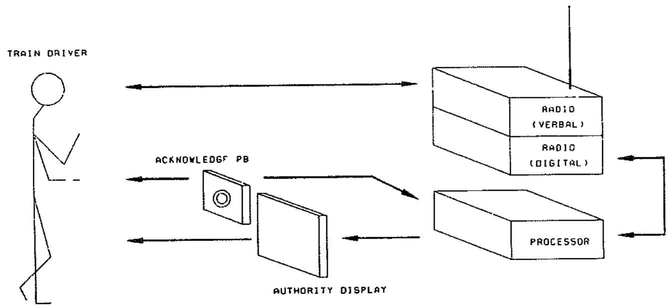


Figure 3.3: Train-Borne Equipment.

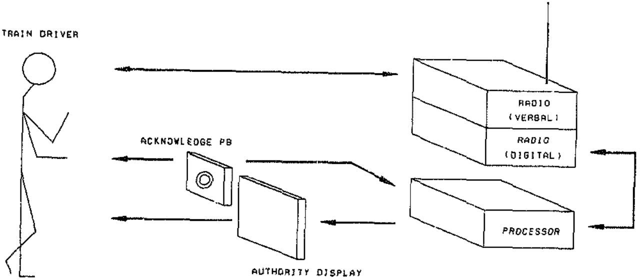


Figure 3.3: Train-Borne Equipment.

3.4 Conclusion.

C.R.T.C. is specified in general by the following major factors:

1)Control is centralised, with movement authorities being transmitted directly to trains by radio;

2)Authorities are transmitted both verbally and electronically;

3)Absolute Working regulations govern all train movement;

4)Traffic movement decisions are made by a traffic controller, but they require the cooperation of the driver of the relevant train before they can be issued as valid instructions.

Chapter 4: Detailed Description of C.R.T.C. Equipment.

4.0 Introduction.

In Chapter 3, a brief overview of the equipment used in C.R.T.C. was given. Chapter 4 will describe in more detail the individual components, particularly with respect to the ergonomics of the human interfaces and the built-in safety at the hardware level of the system.

The operator's interface consists of a monitor and an input device, an overview display and a microprocessor controller. This has been built using an IBM personal computer which was used for the following reasons:-

1) Since a graphics colour monitor and driver is currently available for use with the computer this can be used for the overview display;

2) A monochrome monitor and driver can be controlled by the computer (at the same time as the colour monitor) and the monochrome monitor is used as the operator's data setup monitor;

3) The computer is driven by a fast, powerful microprocessor which is capable of providing the necessary control of the operator's interface;

4)The system includes a keyboard which is used as the data entry interface;

5)Existing hardware and software can be used for interfacing to the safety equipment.

4.1 Safety.

The safety equipment, which accepts requests from the operator's interface, checks the outgoing messages for safety (ensuring that interlocking rules are not transgressed), formats the messages for transfer across the radio link, and ensures that the proper acknowledgements are returned from the train. It is this equipment which is vital to the safe operation of C.R.T.C. and consists of a fail-safe microprocessor system in which three microprocessors run in parallel - all the hardware is triplicated - and a majority 'vote' of two out of the three microprocessors is required before any action is taken (refs. 6,8). The software for this system (each microprocessor runs exactly the same program) is therefore critical, and must be error-free. The major function of the safety equipment is therefore to supervise all the transmissions, ensuring that conflicting authorities are never in force simultaneously and that transmissions are acknowledged correctly. All tasks which are not safety related are undertaken by other equipment in the system. This reduces the amount of software which needs to be validated or tested extensively to remove all errors, since software validation is an extremely labour intensive procedure.

The safety equipment is presented with a 'request' for the release of a train authority, and must decide on the safety of the request (this will depend on authorities that have already been issued). The safety equipment has no effect on, or control of the interaction between the operator and his interface during the generation of the request, it is only 'consulted' in the final phase of a transaction - the data transfer to the train.

4.2 A Data Entry Interface for the Operator.

A man-machine interface for the input of information to a machine must be designed carefully to give fast, simple access to the machine whilst still allowing for the entering of all the possible information which is needed for the running of the system. Various alternatives for the input device were considered in terms of their suitability for the specific aims of C.R.T.C.:

1)Physical Pushbuttons.

In this type of input device, the operator is provided with an array of pushbuttons, each of which is labelled with a command which is understood by the operator to present a valid instruction to the C.R.T.C. equipment. The major disadvantage of this is that, in C.R.T.C. there are a large number of possible 'commands' and the operator will be faced with a large number of pushbuttons from which to select the required pushbutton; he will thus be hampered by the need first to locate the correct pushbutton, then to press it, and this could lead to delays and possible errors.

2) A Full Keyboard.

In this instance, by using a typewriter-type keyboard, the operator builds up a full 'command' by sequentially typing keys, each of which is labelled with a letter, or number. This is a standard approach to data entry, particularly where computer equipment is used. However, it requires that the operator learns the exact structure of each command and the method is time-consuming and error-prone since typing skills are required for the efficient use of a keyboard, and mistakes can be made in typing in the correct structure each time a command is entered.

3) 'Menu and Cursor'.

In this method the operator is given a VDU (Video Display Unit) which displays a number of 'pushbuttons', each one labelled with its function (the pushbutton display is called a 'menu'). Using a 'cursor move' facility, the operator moves an indicator, called a cursor, around on the screen and selects one of the displayed pushbuttons. He then presses a single physical pushbutton which activates the button which he is currently selecting.

The method of 'Menu and Cursor' was chosen as the one most suitable to C.R.T.C. since it has advantages similar to those mentioned for the physical pushbutton method in that an entire instruction can be coded into one pushbutton, and one single entry can be made by the operator to represent a complex command string. It also has the distinct advantage that the labels on the screen, and hence the function of each pushbutton, can be dynamically changed during the course of the data entry procedure. Only those pushbuttons which are relevant to the operator at any particular time need be displayed. This is particularly suitable to C.R.T.C. in which, at any stage, only a limited subset of the full range of pushbuttons needs to be active; for example, at the final stage of a transaction the operator need only choose between 'EXECUTE', and 'ABORT', and hence only these two functions need to be displayed on the 'menu'.

The menu facility can also be used in conjunction with a set of interlocking rules so that unsafe selections can be excluded from the menu. This prevents the operator from making an error and having to re-enter requests. The menu system gives the operator a very fast and efficient data entry interface, significantly reducing the burden on him and hence improving the speed at which he can handle trains. This approach was used in C.R.T.C. for these reasons.

4.3 The Data Entry Interface in C.R.T.C.

As mentioned previously the operator uses the monochrome VDU and the keyboard of the IBM Personal Computer as his data entry and display interface. Part of the screen is dedicated to the display of a list of pushbuttons (the menu) which is dynamically changed during the data entry cycle depending on the type of transaction the operator is attempting to set up. The method of selecting a pushbutton and activating it can be achieved using the keyboard of the IBM Personal Computer. Only three of the keys are used 'Cursor move up', 'Cursor move down', and 'Activate present selection'. This can also be achieved using an even more simple device - a data tablet. In this case the operator uses a blank tablet and he moves the screen indication cursor up and down the 'menu' by moving a physical cursor up and down the tablet. The physical cursor has a pushbutton mounted on it, which the operator uses as his 'activate' pushbutton. Figure 4.1 illustrates the operator input interface, showing the two alternative devices (the keyboard or the data tablet) and the VDU displaying the menu.

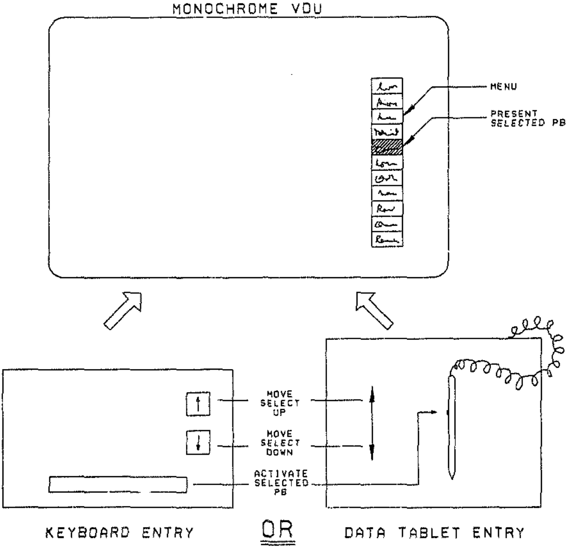


Figure 4.1: The Data Input Interface.

4.4 The Data Display Interface.

The operator is given display which provides him with information regarding the train on which he is currently working (i.e. the train with which he is in radio communication). The display also prompts for any data inputs that are required, reflects the inputs which he makes, and prompts the operator with the verbal message which he must send to the driver during the verbal message transfer portion of the communication protocol. The data display interface is implemented using the remainder of the monochrome display (the rest of which is set aside for the display of the menu). This section of the display, called the 'console', is divided into a number of fields as shown in Figure 4.2.

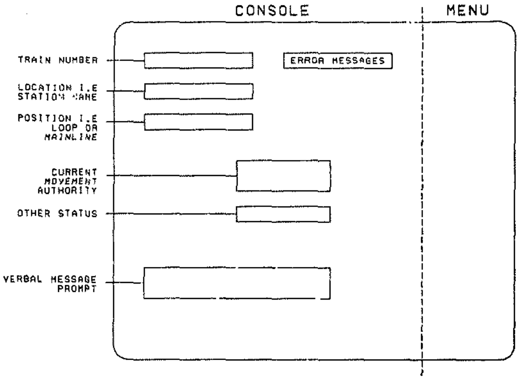


Figure 4.2: The Data Display Interface.

On setting up a communication channel with a train, the console indicates all the information pertaining to that train. This information includes the train number, the present location of the train (i.e. the station at which the train is staged), the position of the train (i.e. the loop which the train is occupying), and any authority with which the train has been issued (i.e. proceed or shunt).

Further information concerning the train may be displayed in the 'other status' field, and this may include a notification that the train is entering a station, or that the train is *'stopped'* in a station (see Chapter 5).

When the operator is required to select a *pushbutton from the* menu, the relevant field of the console is highlighted and a 'question mark' prompt appears. When a transaction reaches the stage at which a verbal authority must be transmitted to the train driver, the full text of the verbal message is displayed in the 'verbal' field of the console for the operator to read out over the radio. When this has been done, the operator and driver continue with the *data transfer phase* of the transaction and if the transaction is rejected by the safety check equipment (i.e. the safety equipment detects an infringement of the interlocking rules, or does not receive the correct acknowledgement from the train) the operator is notified by a message in the 'error' field of the console, and a warning beep sounds. The operator may either abort the transaction, or attempt to retransmit to the driver.

The operator may use the console to display the status of any train in the network without setting up a communications link with the driver. This is termed a 'train enquiry', but in this mode the menu is set up such as to prevent the operator from initiating any train transaction.

4.5 The Overview Display Interface.

The operator is given a graphic display of the entire section under his control to aid in traffic movement decisions, and this is implemented using the colour graphics VDU of the IBM Personal Computer.

The representation of the network consists of a basic layout of the C.R.T.C. area showing Station Names, the single line connections between stations, and the presence of loops within stations as shown in Figure 4.3.

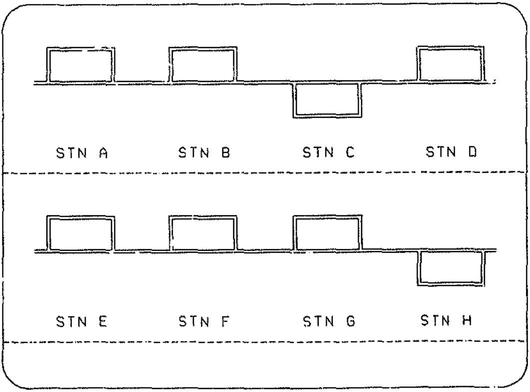


Figure 4.3: The Overview Display Interface.

Using the colour monitor as a display overview, C.R.T.C. can control and display up to 32 *stations*, but due to the physical limitations of the screen size, only 8 stations can be displayed at a time. The operator is given the facility to 'scroll' the VDU display up and down in order to view any group of 8 from the 32 stations. Due to the limitation in width of the VDU, the network is split up and displayed vertically, as shown in Figure 4.3, and in the network *represented in the figure*, Station D is connected to Station E.

Overlaid on the layout diagram *of the colour monitor* is a colour coded representation of all the existing train positions and movement authorities. Figure 4.4 shows the definitions of 'section colours', 'optic contents', and '*optic* colours' which are used to represent the information required by the operator.

DIRECTION ARROW SITUATED ABOVE THE SECTION INDICATES TRAIN MOVEMENT DIRECTION.

COLOUR OF "TRACKS" IS TERMED "SECTION COLOUR."

—68463— NUMERIC DISPLAYED WITHIN TRACKS IS TERMED "OPTIC CONTENTS." COLOUR OF DIGITS IS TERMED "OPTIC COLOUR."

Figure 4.4: Overview Display Nomenclature.

The display represents the trains and movement authorities in the following coded format:-

Optic Contents: The Optic Contents is a numerical figure displayed within a section and indicates the train number of the train which is occupying the section, or has authority to enter the section.

Optic Colour Blue: If the Train number is displayed in blue, then the train is a 'down' train, and is only allowed to proceed in the 'down' direction.

Optic Colour Brown: A Brown train number display signifies that the train is an 'up' train, and may only proceed in the 'up' direction.

Section Colour Red: A section which is displayed in red signifies that the section is occupied. The train which is occupying the section has its train number displayed in the optic of that section.

Section Colour Green: If a section is illuminated with a green colour, this signifies that the section is not occupied, but that a proceed train has been given authority to enter the section (its train number will be displayed in the optic for that section) and hence the section is 'locked out' to any other train.

Section Colour Yellow: A yellow section indicates that a train (whose number appears in the optic) has been given authority to enter the section for shunting.

Section Colour Blue: The section is presently locked but may be released if the train whose number appears in the optic is contacted, and a 'Cancel Proceed' transaction is xecuted.

The operator interprets the information represented to him on the overview display in order to decide on all the transactions required to promote the efficient traffic flow of trains through the network. The graphic nature of the display has been designed in such a way as to give the operator a clear, easily interpreted representation of the status of the network thereby helping him quickly and efficiently to formulate strategies for dealing with the train traffic.

4.6 The Train Driver's Interface.

Train-borne equipment is not triplicated and is not fail-safe. Safety is achieved by other means, as described in Chapter 6. In fact the train-borne equipment has been kept as cheap and simple as possible for two reasons :-

1) In any C.R.T.C. network there is only one set of control room equipment, but many sets of train-borne equipment and it is therefore essential to minimise the cost of train-borne equipment.

2) Train-borne equipment must be portable so that it can be quickly installed aboard trains entering a C.R.T.C. network and removed as the trains leave.

C.R.T.C. has been designed to give the train driver a minimal overhead in terms of 'pushing buttons' while still ensuring that he gives his consent to all transactions before they can be completed. A single pushbutton mounted on the train-borne equipment gives the driver the facility to do this, since no transaction will be completed unless he pushes the button at the correct time (the pushbutton is hence labelled 'ACCEPT', illustrating to the driver that he must depress this button in order to accept a transaction).

The driver is also given a tangible representation of his current movement authority, and this takes the form of an alphanumeric display which gives him a plain language description of his authority. Figure 4.5 shows the driver's interface.

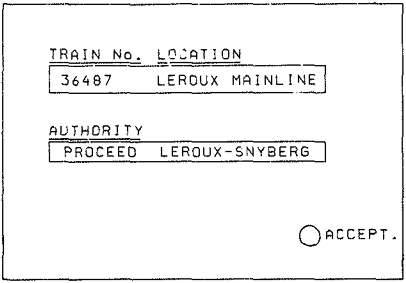


Figure 4.5: The Train Driver's Interface.

As described previously, the driver is given a verbal authority and, on pressing his 'accept' pushbutton, he receives visual confirmation of his authority on the display. The authority can not be removed by the operator except by a further transaction, which must again receive the consent of the driver. This mechanism serves to protect the driver but also means that it is the driver's responsibility to ensure that incorrect transactions are not undertaken. The driver, for example, should refuse to accept a 'Cancel Proceed' authority if he has already entered the section into which he was previously authorised.

4.7 C.R.T.C. Operation.

The C.R.T.C. operator has a radio system with which he can set up a communication link with a train, data entry and display units (i.e. the keyboard and monochrome display) and a network overview display (the colour monitor). A representation of a control room layout is depicted in Figure 4.6, and equipment on board the train will be mounted as shown in Figure 4.7.

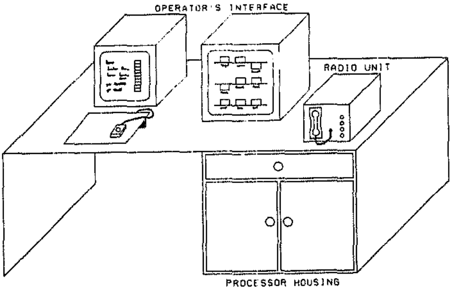


Figure 4.6: A Representation of a Control Room Layout.

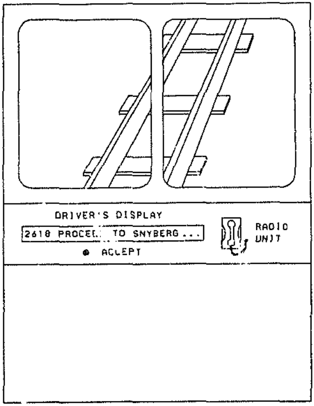


Figure 4.7: The Mounting of the Train-Borne Equipment.

How does this equipment all fit together to allow the operator to manage C.R.T.C. traffic? Assume there are several trains in the C.R.T.C. network, each one staged in one of the stations. The trains and their locations are represented on the overview display, and the operator decides to allow one of the trains to proceed from its present station to another station in the network. The operator requires, then, to execute an 'Issue Proceed Authority'. His first action, in accordance with the laid out protocol for a transaction, is to ectablish a communication link with the relevant train. This causes the console to display all the details of the train including its train number, its present station, and the loop where it is staged. The 'menu' will give the operator the option to initiate a 'PROCEED' transaction, or a 'SHUNT' transaction. The operator, in this case, selects 'PROCEED' with his cursor movement keys, and activates his selection. The console then prompts for new information, by displaying (in the 'proceed' field) "PROCEED TO ---------?", while the new list of pushbuttons on the menu will provide a list of permissible destination stations (permissible stations for a proceed authority are discussed in Chapter 6 under Interlocking).

After selecting a Station Name (from the list of pushbuttons) as the destination for the proceed authority, the verbal transfer stage of the transaction has been reached. The console displays, in the verbal field, the message "TRAIN NNN YOU ARE AUTHORISED TO PROCEED TO STATION AAA" (NNN is the train number and AAA the destination station name) and the menu gives the 'EXECUTE' option. The operator reads the message to the driver, and if they both agree, the operator presses 'EXECUTE' while the driver presses 'ACCEPT' and the message is transferred to the train. Once the procedure has been completed, the proceed authority is displayed in the cab and constitutes the absolute authority for the driver to enter the sections along his route. The new status of the train is illustrated on the overview display [illegible] icates that the train is now authorised to proceed t[illegible]n AAA.

The operator may now either close the communication channel or initiate a new transaction with the same train driver (the new transaction is selected from a new menu, one which now excludes the possibility to proceed or shunt). Full details of all the possible transactions are described in Chapter 5.

4.8 Conclusion.

Chapters 3 and 4 of this dissertation have described the mechanisms by which a traffic controller can manage trains in a single line network. The essential structure of the system is that the controller uses microprocessor equipment to set up 'Movement Authorities' which he then transmits both verbally and electronically to trains in the network. In the following Chapter, Chapter 5, the details of movement authorities are discussed, and a set of 'Interlocking Rules' are set out in Chapter 6. The interlocking rules can be seen as the way in which the existing status of trains in the network will affect the permissible transactions which can occur with other trains, and the objective of the interlocking rules is to enforce Absolute Working Regulations.

Chapter 5: C.R.T.C. Transactions.

5.0 Introduction.

A C.R.T.C. transaction, as described previously, consists of a series of actions whereby a C.R.T.C. traffic controller transmits messages to trains in a C.R.T.C. network.

'Primary' transactions are those transactions which require communication with a train and cooperation from the driver. A C.R.T.C. primary transaction results in the issuing of some authority to a train driver. The following is a list of primary transactions which are operational in C.R.T.C. and which are described in detail later in this section:-

1)Initialise Train into C.R.T.C.;

2)Train Exit from C.R.T.C.;

3)Issuing of Proceed Authority;

4)Train Clear of Station;

5)Train Clear of Penultimate Station and Entry into Destination Station;

6)Train Complete in Destination Station;

7) Extension of Proceed Authority;

8) Cancellation of Proceed Authority;

9) Issuing of a Shunt Authority;

10) Train Ending Shunt;

11) Change of Train Number.

C.R.T.C. also includes a secondary level of transaction which takes place without a communication channel being set up with a train. These transactions allow the operator to set up various functions which do not require any interaction with the train drivers.

5.1 Secondary Transactions.

'Train Enquiry' is one of the 'secondary' transactions available to the operator. This is used purely as an operator aid, and it allows the operator to display information pertaining to a train without communicating with the train. When no communications channel is open the operator is given a list of train numbers on the menu, and each train in the system is listed. Selection of one of these initiates a 'Train Enquiry' transaction for that particular train. The 'Train Enquiry' transaction does not have any effect on the overview display, neither are any messages transmitted to any trains (in fact the operator can only initiate a train enquiry when no channel is open). Details of the train selected by the operator are displayed on the console.

A further option given to the operator in the form of a secondary transaction is that of 'Set Blocking Cross'. Here, the operator can use a blocking cross as a physical reminder that a section is temporarily closed:- the blocking cross may be used, for example, when a section is down for repair. When the operator has set a blocking cross in a particular section, the C.R.T.C. interlocking prevents the issuing of any authority for a train to enter the section. The operator may set a blocking cross in any section, or in any station loop. Once again, this option is only permitted when no channel is open. In order to set a blocking cross, the operator activates a pushbutton labelled "SET BLOCKING CROSS". He then selects the station, and the position (e.g. main line, down end block etc.) for the blocking cross and, on execution of the transaction, the blocking cross is indicated on the display as shown in Figure 5.1. Each Blocking cross which is set is given a number so that the operator can identify it, and this number (XXXØ - XXX9) is displayed in the optic of the section to which it refers. Figure 5.1 shows that there is a blocking cross at Station B on the Main Line, and one on the Down End section.

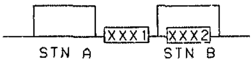


Figure 5.1: Blocking Crosses in Stn. B.

The third type of secondary transaction which the operator may initiate when no communication channel is open, is that of 'Clear Blocking Cross'. The operator uses this transaction to remove a blocking cross when a section has re-opened. It is primarily the operator's responsibility to ensure that a section is workable before he removes the blocking cross. He initiates the removal by selecting the menu option labelled "CLEAR BLOCK CROSS". The operator then identifies the blocking cross which he wishes to remove by selecting the pushbutton which is labelled with the blocking cross number and, on execution of this transaction, the portion of the overview display which indicated the blocking cross is cleared (back to white), and the interlocking for the section is released.

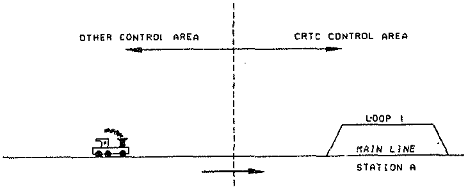
5.2 Primary Transactions.

1) Initialisation of a Train into C.R.T.C.

Each train that enters the C.R.T.C. system must first undergo an initialisation procedure. When a train requires entry into C.R.T.C. the operator first establishes a communication link with the train. This will be the first time that the operator has communicated with that particular train, and when the operator selects a 'radio number' for the train (which is an identification number for the train-borne radio) the processing equipment (in the control room) recognises the train as a new one by virtue of the new radio number.

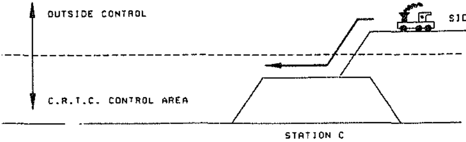
The train must *first be 'Initialised', and the operator* is prompted to enter the information for a 'Train Initialisation'. This information includes a train number (6 digits or less), the name of the station at which the train will enter, and the loop in the station which the train will occupy.

Train Initialisation occurs for a train entering C.R.T.C. either from a private siding within the network, or from another network outside C.R.T.C. control. On execution of the transaction the train driver receives the message "TRAIN NNN YOU ARE AUTHORISED TO ENTER C.R.T.C. AT (STATION), (POSITION)". Figures 5.2 and 5.3 illustrate train initialisation for the case of a train entering C.R.T.C. from another control area (Figure 5.2), and from a private siding into Station C (Figure 5.3).



"TRAIN 666 YOU ARE AUTHORISED TO ENTER C.R.T.C. AT STATION A, MAIN LINE."

Figure 5.2: Train Initialisation (1).



"TRAIN 666 YOU ARE AUTHORISED TO ENTER C.R.T.C. AT STATION C, LOOP 1."

Figure 5.3: Train Initialisation (2).

A newly initialised train will be displayed on the overview display (see Figure 5.4) where the section to be occupied by the train will be displayed in red to show that the section is occupied. The optic for that section will display the train number, and the optic colour will be displayed in blue for a 'down' train and in brown for an 'up' train. The direction of a train is determined by the train number which was entered by the operator during the 'train initialisation' transaction. An even train number designates an 'up' train, and an odd train number designates a 'down' train.

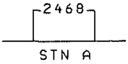


Figure 5.4: Train 2468 Initialised at Stn. A.

The 'train initialisation' transaction occurs only for 'new' trains entering the control area. Once a train has completed this transaction, the C.R.T.C. system recognises the train each time the operator sets up communications with that train throughout the train's sojourn in the network and the *operator's interface will now allow a 'proceed' or 'shunt'* transaction to take place.

2) Exit from C.R.T.C.

The 'Exit from C.R.T.C.' transaction allows the operator to enter the information that a train has left the C.R.T.C. area i.e. the train has entered another contrc or a private siding. The driver of the train must, as . acknowledge the transaction, since at the conclusion of this transaction the driver will have no authority to be in the area. After an 'exit' transaction, the overview display for the section previously occupied by the train is cleared, and the colour returns to white.

3) Issuing of a Proceed Authority.

Once a train has been initialised, the operator may allow the driver to proceed from his present location to a destination station within the network. Referring to Figure 5.5, Train 666 has been initialised into C.R.T.C. and is staged at Station A on the main line.

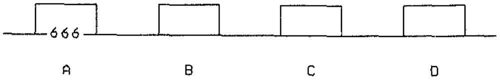


Figure 5.5: Train 666 Staged at Stn. A, Main Line.

The operator initiates a 'proceed' transaction for the train with which he is in communication, in this case Train 666, by activating the "PROCEED" pushbutton on the menu. The console then prompts the operator to enter the destination station name by displaying "PROCEED TO-----?" and the operator must select the station name from the list provided by the menu. Once the operator has selected and activated a destination station name (assume he selects Station D in this example) the verbal part of the transaction has been reached. The console then displays the message "TRAIN 666 YOU ARE AUTHORISED TO PROCEED TO STATION D", and the operator reads this out to the driver. On the execution of this transaction the authority to proceed is transmitted to the train and the message appears on the driver's display.

The Proceed Authority gives the driver in the above example authority to leave Station A, to enter the section A-B, to travel straight through Stations B and C and to approach Station D. The driver is not yet authorised to enter Station D.

After the successful issuing of a proceed authority the overview display highlights those sections which have been allocated to that particular train, thus preventing any other trains from entering those sections, i.e. sections A-B, B-C, and C-D have been allocated solely to Train 666 hence no other train will be allowed to enter any of these sections. Figure 5.6 illustrates the section of the overview display which indicates the fact that Train 666 is proceeding from A to D. The Main Line at A and the section A-B are displayed in red to show that the train may be occupying either of these sections. B main line, the section B-C, C main line and the section C-D are displayed in green, signifying that although the sections have been allocated to Train 666, the train has not yet occupied any of them.

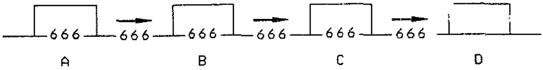


Figure 5.6: Train 666 Authorised to Proceed to Stn. D.

4) Train Clear of Station.

A driver who has been given a proceed authority must always report back to the controller when his entire train is clear of a Station along his route, that is when his train leaves the starting station, and when it passes through any station along the route. The 'clear' transaction serves to 'clear' the sections behind a proceeding train as it leaves the sections so that they can be released to other traffic. The transaction also enables C.R.T.C. to keep a record of the position of a moving train. In the example cited above, Train 666 has authority to proceed from A to D. The driver would report back to the control room as soon as his train leaves Station A. The controller then initiates a 'clear' transaction by activating the pushbutton 'CLEAR' at the data entry interface. Immediately, the verbal phase is reached and the message "TRAIN 666 YOU ARE ACKNOWLEDGED CLEAR OF STATION A" is displayed in the verbal field of the console. On execution of this transaction the message is transmitted to the driver. The overview display is updated to reflect the new situation (see Figure 5.7). Any one of the sections A-B, B main line, and B-C may be occupied by the train from this point, since the driver is not required to report his position again until clearing Station B. Sections C main line and C-D remain green to indicate that these sections are open to Train 666 and locked to other trains.

Figure 5.7: Train 666 Clear of Stn. A.

Upon clearing Station B the driver reports back to the controller and another 'clear' transaction is executed. The overview display is updated, and now sections A-B and B main line are cleared (the section colours return to white). As shown in Figure 5.8 the sections B-C, C main line and C-D indicate that the Train 666 is now in occupation (hence they are illuminated red).

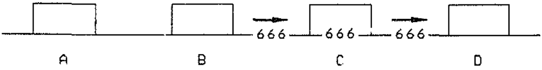


Figure 5.8: Train 666 Clear of Stn. B.

5) Train Clear of Penultimate Station.

In the example of the proceeding train (666) described above, the destination station is Station D; however the driver is not yet authorised to enter the station. This authorisation may be included in the 'clear' transaction when the driver reports clear of his penultimate station (in this case Station C). The operator again activates the "CLEAR" pushbutton, but since the penultimate station has been cleared, the console prompts the controller to select an 'Entry Authority'. The operator may select one of the loops within the station ("MAIN LINE", or "LOOP 1") or he may opt to prevent the train from entering the station by activating a pushbutton labelled "NO ENTRY". The activation of one of the entry authority pushbuttons indicating the start of the verbal phase, and the message transmitted to the driver is "TRAIN 666 YOU ARE ACKNOWLEDGED CLEAR OF STATION C. ENTER STATION D AT MAIN LINE (or LOOP 1)", or "TRAIN 666 YOU ARE ACKNOWLEDGED CLEAR OF STATION C. DO NOT ENTER STATION D". If the driver is given authority to enter the station then the overview display reflects this as shown in Figure 5.9. In this figure loop 1 has been selected by the operator. The sections C-D and D loop 1 are displayed in red to show that the train may be in either one of the sections.

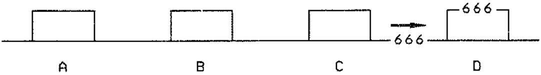


Figure 5.9: Train 666 Entering Stn. D.

If the operator had decided to prevent Train 666 from entering the station (he may have been forced to do this - see interlocking rules) then this would have been represented on the overview display by a break in the section C-D immediately outside Station D as shown in Figure 5.10 and in this case only the section C-D would be shown as being occupied, the station still being clear.

Figure 5.10: Train 666 Unauthorised to Enter Stn. D.

When the operator is ready to allow the train to enter the station (or when the interlocking is released to allow this) he executes an 'entry' transaction by selecting the entry loop. The new authority is then transmitted to the driver in the normal way.

6) Train Complete In Station.

A driver who has been given authority to enter a station must call back to the operator when his train is complete in the station i.e. the train is completely clear of the main line section, and a 'train complete' transaction must be initiated. The operator activates a pushbutton labelled "COMPLETE" and the verbal phase is entered. In the example of Train 666 the message displayed in the verbal field of the console reads "TRAIN 666 YOU ARE ACKNOWLEDGED COMPLETE IN STATION D, LOOP 1". This message is transmitted to the driver and the display reverts to the simple train occupation as shown in Figure 5.11.

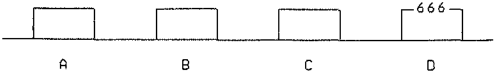


Figure 5.11: Train 666 Complete in Stn. D.

7) Extend Authority.

Once a train has been given a proceed authority (for example "PROCEED FROM A TO C") the controller may decide to extend the authority so that the driver need not be delayed at C by the station entry procedures described above. For example, given the situation in which a driver is authorised to proceed to Station C the driver may have his authority 'extended' so that he can travel straight through Station C to Station D. The 'extend authority' transaction allows the operator to establish this. Having set up communication with the appropriate driver the controller activates a pushbutton labelled "EXTEND AUTHORITY" and subsequently selects a new destination station, in this case Station D. The message sent to the driver is "YOU ARE AUTHORISED TO PROCEED TO STATION D" and the overview display is updated to show the new sections which have been assigned to the train.

8) Cancellation of . Proceed Authority.

For the efficient working of a C.R.T.C. section it may be necessary to cancel a proceed authority - an example of this is shown in Figure 5.12 where Train 666 has the authority "PROCEED FROM A TO C" whilst Train 777 is staged at D (see Figure 5.12). The operator may decide to allow Train 777 to proceed to B and effect the train crossing at that station, rather than forcing 777 to remain at D or allowing 777 to travel only as far as C. It is then necessary that the current proceed authority for Train 666 be cancelled, and a new authority issued:- "PROCEED .O B". The C.R.T.C. equipment cannot be sure of the present location of the Train 666 - the train may have already entered the section A-B. For this reason, C.R.T.C. requires that two transactions be executed for a proceed cancellation:- firstly an instruction to the driver to stop in a station and secondly a cancellation of the proceed authority. The act of cancelling a proceed authority is potentially very dangerous, hence C.R.T.C. demands that two separate transactions be followed. The instruction to the driver to stop in a station serves to ensure that he is absolutely sure of his location (since in a station he can read the station name from the board at the platform). This is necessary to make sure that the driver has not already entered a section for which the operator requires to withdraw the authority.

Figure 5.12: Trains 666 and 777.

In the example illustrated above, the operator must contact the driver of Train 666 and, if the train has not yet left Station A, verbally instruct the driver to remain at the station. If the train has already entered the section, the operator instructs the driver to stop at the next station. Once the driver is stationary in the station as instructed, a 'train stopped' transaction is executed. The operator activates the pushbutton labelled "TRAIN STOPPED" and enters the station and loop at which the train is stopped. On execution of this transaction the driver receives a message such as "TRAIN 666 YOU ARE ACKNOWLEDGED STOPPED AT STATION A, MAIN LINE". The overview display shows that the train is occupying the main line at Station A (see Figure 5.13) but the sections which were previously allocated to the train are not cleared, and these sections are displayed in blue. The colour blue in these sections indicates that the sections have not yet been released, but that they may now be released by a 'cancel proceed' transaction. The operator may now initiate a 'cancel proceed' transaction by activating a pushbutton labelled "CANCEL PROCEED".

The driver receives the message "TRAIN 666 CANCELLED PROCEED AUTHORITY" (which he may accept or reject in the normal way) and the sections are released, the overview display of the sections reverting to white. The operator may now re-issue proceed authorities to the two trains in order to cross-over the trains at Station B.

Figure 5.13: Train 666 'Stopped' at Stn. A.

A 'train stop' and 'cancel proceed' transaction pair may occur at any stage of a train's movement between its source and destination stations.

9) Authority to Shunt.

A shunting train moves backwards and forwards within a station, and has the potential to occupy the main line and all free loops, and to move out of the station onto the section. C.R.T.C. allows a shunting authority to be given to a train driver at a station, and gives access either to the 'up-side section' or the 'down-side section', or both. The operator enters "SHUNT" and then selects one alternative from "UP-SIDE", "DOWN-SIDE" or "BOTH SIDES". The movement authority reads: "TRAIN NNN YOU ARE AUTHORISED TO SHUNT AT (STATION), (SIDE/S)".

On the overview display, shunting is indicated by the tracks being coloured yellow (see Figure 5.14). All the sections which may be occupied by the train are coloured yellow. The Figure shows a station at which Train 246 is shunting on the 'up-side'. A second train, Train 440 is occupying one of the loops, so the loop is displayed in red. The entire Section B-C is coloured yellow, indicating that this section is locked against the issuing of any authorities to other trains.

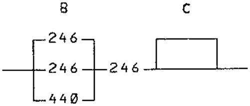


Figure 5.14: Train 246 Shunting.

10) End Shunt.

When the driver of a shunting train has completed a shunting manoeuvre, he reports back to the controller who selects the pushbutton labelled "END SHUNT". The controller selects the pushbutton which represents the section which the train is now occupying (the train may have moved, for example, from the main line out into the section and back into a loop). The message sent to the train is: "TRAIN NNN ACKNOWLEDGED END OF SHUNT AT STATION B, LOOP 1". The yellow indications on the display are cleared and the display reverts to showing the usual 'track-occupied' indication for the section which the train now occupies.

11) Change Train Number.

On train initialisation, the operator enters a train number for the new train and this number is used by the train driver and the operator to identify the train during its movement through C.R.T.C. If, for any reason, the operator requires to change this train number (for example he may want to change the direction of the train) he must follow the 'change train number' transaction. The operator selects the "CHANGE TRAIN NUMBER" pushbutton and then enters the new train number, digit by digit. The message "NEW TRAIN NUMBER NNN" is sent to the train driver, and all the optics on the overview display which concern the relevant train are updated to the new number, in the colour brown for an even-numbered up-train, and in blue for an odd-numbered down-train.

5.3 The 'ABORT' Function.

At any stage during a transaction which has not yet been executed, the operator is given the opportunity to select a pushbutton labelled "ABORT" . This effectively cancels the last selection made by the operator. For example, if he selects "PROCEED" for a newly-initialised train, this will cause a list of possible destination stations to be displayed on the menu together with an "ABORT" pushbutton. On pressing "ABORT", the menu will return to the previous state and allow him instead to re-initialise the transaction as, for example, a 'shunt' transaction. The "ABORT" function can not, however, be used to cancel an authority which has already been executed. For example, in order to stop a train once a proceed authority has been sent, the operator must institute a full 'cancel proceed' transaction.

5.4 Language.

C.R.T.C. is designed for dual-medium operation and can be implemented so that all messages, console prompts, and pushbutton labels appear in English and in Afrikaans. The system can also be modified for any other two languages. At any time the operator may change the current language medium by selecting "ENG/AFR" on the menu. All prompts on the display screen and all messages sent and received are formatted in the currently- set language.

5.5 Conclusion.

The set of transactions which have been decribed in this chapter provide an operating system using which the operator can manage trains, allowing drivers both to proceed and to shunt and providing a number of subsidiary transactions for efficient traffic management. The transactions are restricted by a set of interlocking rules as described in the next chapter which enforce the adherence to Absolute Working regulations.

Chapter 6: C.R.T.C. Interlocking.

6.0 Introduction.

"Interlocking" is a procedure by which the circumstances of a system may cause certain activities to be excluded. Activities which might normally be allowable are disallowed due to the presence of some circumstance or circumstances.

In C.R.T.C. there are two levels of interlocking. The first level is concerned with the 'sensible' working of trains. For example, if a train has no proceed authority it is nonsense to attempt a 'cancel proceed' transaction. The option of proceed cancellation is therefore "interlocked" in such a way that the transaction cannot be initiated for a train with no proceed authority. The second level of interlocking is concerned with safety. For example, the presence, or possible presence of a train in a section, whether stationery, proceeding or shunting, excludes the possibility of allowing a second train into the section. Existing circumstances, ie existing authorities, effectively therefore lock out the possibility of initialising certain new transactions.

Both levels of interlocking are implemented in the operator's interface, both nonsensical and unsafe selections being excluded from the menu. As mentioned previously, a program running in specially designed fail-safe equipment double-checks the safety interlocking by supervising the final stage of a transaction - the data transmission. In C.R.T.C., therefore, there are two levels at which the safety interlocking rules are enforced. The safety equipment keeps a record of all the trains in a network and checks that any new transaction generated at the operator's interface will not cause an infringement of the safety interlocking rules. Secondly, the operator's interface is designed so that it is impossible for the operator to enter an unsafe request, and therefore the safety equipment will only ever reject a transaction if there has been a failure in the operator's interface.

Absolute Working regulations provide the basis for a set of interlocking rules which cover the safety of a network, and these rules (along with the 'sensible', or 'functional' rules) are outlined in detail in the following Chapter. The rules which are specifically safety-related are marked with the symbol **.

6.1 General Rules.

1) The operator can not initiate any primary transaction unless a communication link has been established.

2)** The operator can only execute a primary transaction with the train with which he is in communication.

3)** No transaction is accepted by the C.R.T.C. system as being valid without the proper, timeous acknowledgement of the transaction by the driver (see Communication Protocol, Chapter 3).

4)** The train driver must not consider an authority as being valid if the message displayed in the cab is corrupted, or does not correspond with the verbal authority received from the operator.

6.2 Secondary Transactions: Train Enquiry, Setting and Clearing of Blocking Crosses.

1) The operator can only initiate a secondary transaction when there is no established communication channel.

2)** The interlocking rules treat a section marked with a blocking cross in exactly the same way as if the section is occupied by a train. Setting a blocking cross is therefore constrained by the same rules as those governing the initialisation of a train into a section (see section 6.3).

3) The operator can only 'clear' existing blocking crosses.

6.3 'Train Initialisation' Transaction.

1) The operator can only initialise 'new' trains - that is trains which are about to enter C.R.T.C. and have not yet been initialised.

2) In specifying a train number for a new train the operator can not enter more than 6 digits.

3)** If a train is occupying a section (or a blocking cross has been set in the section) the operator can not initialise a new train into that section (see Figure 6.1).

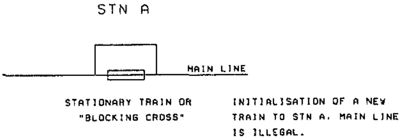


Figure 6.1: Initialisation Rule 3.

4)** The operator can not initialise a train into a station or into any of the loops if a second train has a proceed authority entitling it to pass through the station (see Figure 6.2).

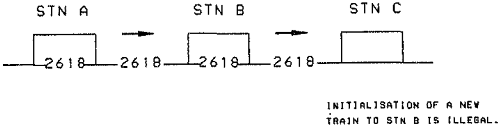


Figure 6.2: Initialisation Rule 4.

5)** If a train is shunting at a station, all the unoccupied loops at the station are locked for shunting and a new train can not be initialised into the station (see Figure 6.3).

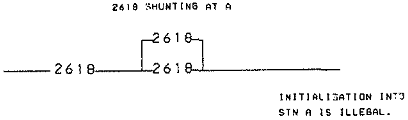


Figure 6.3: Initialisation Rule 5.

6)** If a train is 'entering' a station, all the unoccupied loops at the station are locked and a new train can not be initialised in to the station (see Figure 6.4).

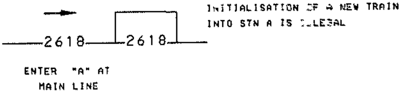


Figure 6.4: Initialisation Rule 6.

<u>6.4 'Exit from C.R.T.C.' Transaction.</u>

1) Only a train which has been initialised into C.R.T.C. can be exited from C.R.T.C.

2)** It is not allowed for the operator to exit a train which has a proceed authority.

3)** It is not allowed for the operator to exit a train which has a shunt authority.

6.5 'Issuing of a Proceed Authority' Transaction.

The following example is used to illustrate the terminology used in the rules for proceeding trains:

A network consists of 5 stations in the order A, B, C, D, E. A train, number 999, has been initialised at A and has been given authority to proceed to E. Initially, A is the 'ORIGIN' station for the train 999; B, C, and D are 'THROUGH' stations and E is the 'DESTINATION' station for the train. When the train clears station A, and a 'clear' transaction has been executed, there are no changes in the station designations (A remains the 'ORIGIN', B, C, and D the 'THROUGH' stations and E the 'DESTINATION'). On clearing station B, however, the proceed authority changes and B becomes the 'ORIGIN' station of the authority, C and D remain as 'THROUGH' stations, and E remains as the 'DESTINATION'. Station A is no longer part of the authority at all. Similarly, as the train proceeds through the network and clears the stations along the route (the stations C and D respectively) change from being 'THROUGH' to 'ORIGIN' and are then removed from the authority. On approaching the 'DESTINATION' station, E, then the driver may be given authority to enter the station, in which case station E is called the 'ENTRY' station for the train.

1)** A train must be initialised before it can be given a proceed authority.

2)The operator can not issue a proceed authority to a train which is already in possession of a movement authority (proceeding or shunting).

3)A proceed authority can span one or more inter-station sections.

4) A train can be given a proceed authority only in the 'up' direction if the train number is even and only in the 'down' direction if the train number is odd (see Figure 6.5).

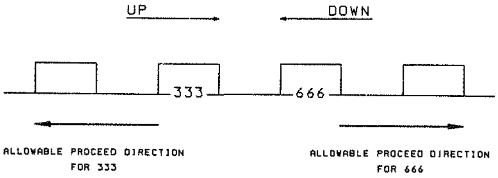


Figure 6.5: Proceed Rule 4.

5)** A proceed authority can not include a THROUGH station which has any occupied loops (see Figure 6.6).

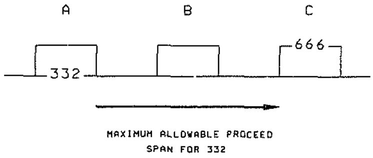


Figure 6.6: Proceed Rule 5.

6)** A proceed authority can not include a 'THROUGH' station which is the 'DESTINATION' station for a train travelling in the opposite direction (see Figure 6.7).

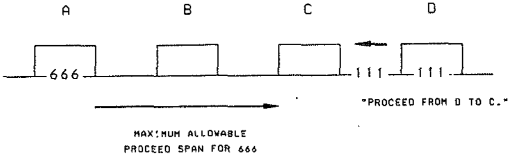


Figure 6.7: Proceed Rule 6.

5)** A proceed authority can not include a THROUGH station which has any occupied loops (see Figure 6.6).

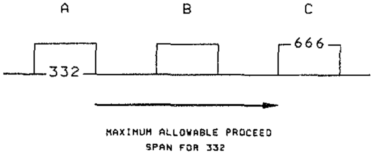


Figure 6.6: Proceed Rule 5.

6)** A proceed authority can not include a 'THROUGH' station which is the 'DESTINATION' station for a train travelling in the opposite direction (see Figure 6.7).

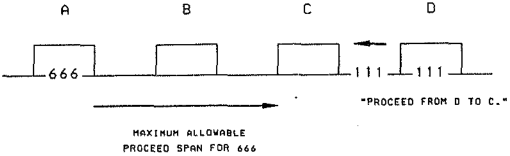


Figure 6.7: Proceed Rule 6.

7)** A proceed authority can not include a 'THROUGH' station which is also the 'ORIGIN' station for a train travelling in the same direction (see Figure 6.8).

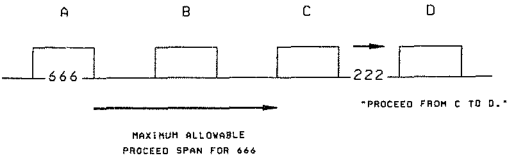


Figure 6.8: Proceed Rule 7.

8)** A train which is stationary in a station can not be given a proceed authority if a second train is shunting at the same station (see Figure 6.9).

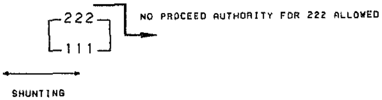


Figure 6.9: Proceed Rule 8.

9)** A train which is stationary in a station can not be given a proceed authority if another train is entering the same station (see Figure 6.10).

7)** A proceed authority can not include a 'THROUGH' station which is also the 'ORIGIN' station for a train travelling in the same direction (see Figure 6.8).

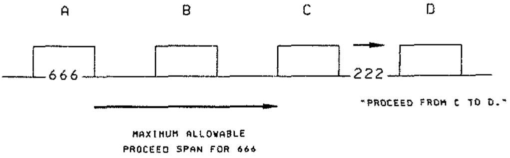


Figure 6.8: Proceed Rule 7.

8)** A train which is stationary in a station can not be given a proceed authority if a second train is shunting at the same station (see Figure 6.9).

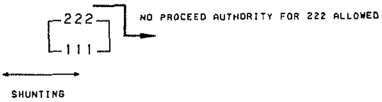


Figure 6.9: Proceed Rule 8.

9)** A train which is stationary in a station can not be given a proceed authority if another train is entering the same station (see Figure 6.10).

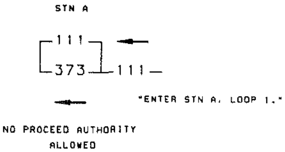


Figure 6.10: Proceed Rule 9.

10)** A station can not constitute the 'DESTINATION' for a proceed authority if the station has no unoccupied loops (see Figure 6.11).

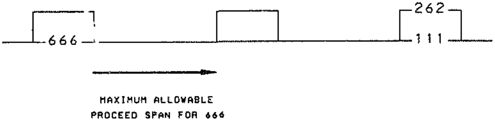


Figure 6.11: Proceed Rule 10.

11)** A train can not be given a proceed authority in the 'up' direction to a destination at which a second train is shunting on the 'down' side (see Figure 6.12) or a proceed authority in the 'down' direction to a destination at which a second train is shunting on the 'up' side (see Figure 6.13).

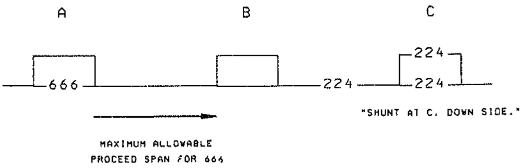


Figure 6.12: Proceed Rule 11.

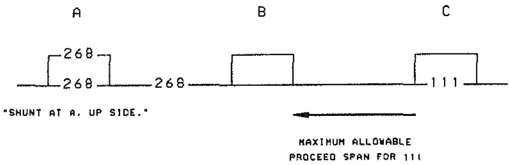


Figure 6.13: Proceed Rule 11.

6.6 'Train Clear of Station' Transaction.

This transaction is only allowed for a train which has been given a proceed authority and has not yet cleared the penultimate station. There are no other interlocking rules governing the 'clear' transaction.

6.7 'Entry into Destination Station' Transaction.

1) The 'entry' transaction can only be initiated with a train in conjunction with a 'train clear' transaction for the penultimate station, or for a train which has previously received a "no entry" instruction.

2)** A train can not be given entry to a station if a second train is entering the station (see Figure 6.14).

Figure 6.14: Entry Rule 2.

3)** A train can not be given entry to a station if a second train is shunting at the station (see Figure 6.15).

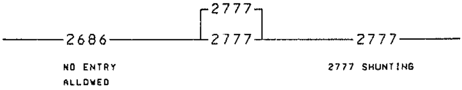


Figure 6.15: Entry Rule 3.

4)** A train can not be given entry to an occupied loop (see Figure 6.16).

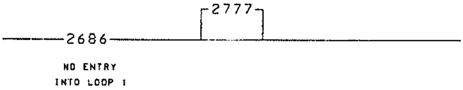


Figure 6.16: Entry Rule 4.

6.8 'Train Complete' Transaction.

This transaction is only allowed for a train which has been given authority to enter a station. There are no other interlocking rules governing this transaction.

6.9 'Extension of Proceed Authority' Transaction.

The operator can only extend the proceed authority for a train which already has a proceed authority and has not yet cleared its penultimate station. The selection of a new destination is constrained by the same rules as those pertaining to the issuing of a proceed authority, except that the operator can not use this transaction to reduce the span of an existing authority. No other interlocking rules govern this transaction.

6.10 'Train Stop' Transaction (preliminary to 'Cancel Proceed' transaction).

The operator can only 'stop' a train which has a proceed authority and which has not yet cleared its penultimate station. The operator must 'enter' the station name and the loop at which the train has stopped.

1)** The train can only be stopped at the station which it would normally have cleared at the next transaction.

2)** If the train has not yet left its starting point the operator can only specify the loop at which the train is staged.

3)** If the train has left its starting station and is stopped en-route, the operator can only specify an unoccupied loop as the stopping place.

6.11 'Cancel Proceed' Transaction.

** A proceed authority can only be cancelled for a train which has undergone a 'stop' transaction. There are no other rules governing this transaction.

6.12 'Issuing of a Shunt Authority' Transaction.

1) A train must be initialised before it can be given authority to shunt.

2) The operator can not issue a shunt authority to a train which is already in possession of a movement authority (proceeding or shunting).

3)** A train can not shunt at a station if a second train is entering the station (see Figure 6.17).

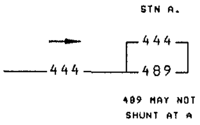


Figure 6.17: Shunt Rule 3.

4)** A train can not shunt at a station where a second train is shunting.

5)** No two trains can shunt from different stations into the same section (see Figure 6.18).

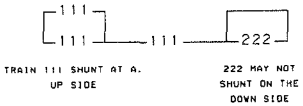


Figure 6.18: Shunt Rule 5.

6)** A train can not shunt into a section which has been dedicated to a proceeding train (see Figure 6.19).

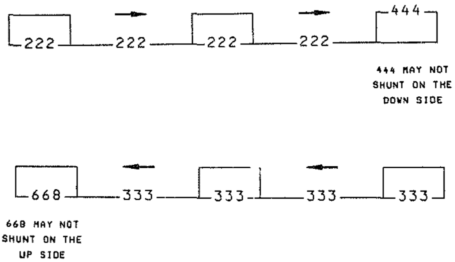


Figure 6.19: Shunt Rule 6.

6.13 'End of Shunt' Transaction.

1) The operator can only initiate an 'end shunt' transaction for a train which has been given a shunt authority.

2)** *The final position of the train (i.e. in the main* line, or in one of the loops) can not be one which is already occupied by another train (see Figure 6.20).

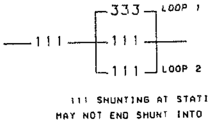


Figure 6.20: End Of Shunt Rule 2.

6.14 'Change Train Number' Transaction.

1) A train must be initialised, but not shunting or proceeding before a 'change train number' transaction can be initiated.

2) In selecting a new train number, the operator can not enter more than 6 digits.

6.15 The Implementation of Interlocking Rules.

The rules described in this chapter serve to ensure that the operation of C.R.T.C. follows a specified format, at the same time as preventing the occurrence of situations which could result in train collisions. The operator is given an aid in following these rules in that, at each stage of a transaction, only those pushbuttons which are valid appear on the menu. In this way the operator is both physically prevented from entering an illegal transaction and relieved of the task of searching through a number of pushbuttons to find a valid one.

All the operating functions are regulated by restricting pushbutton choice, both those which are designed to force consistent operation ("functional rules") and those designed to implement absolute working ("safety rules"). For example, when a train has been initialised and has no movement authority, the operator is given a "PROCEED", a "SHUNT", a "CHANGE TRAIN NUMBER" and an "EXIT C.R.T.C." pushbutton. He is not given the options to "END SHUNT", "TRAIN STOP", "EXTEND PROCEED", or "CLEAR STATION". This ensures the effective implementation of functional rules. If the operator initiates an 'issue proceed' transaction by activating the pushbutton labelled "PROCEED" then he is given a new menu with a list of station names. Only those stations which are permissible (under the interlocking rules) are given to the operator as options for the destination station. This is therefore an implementation of "safety" regulations.

The method of pushbutton exclusion to implement C.R.T.C. operating regulations is used merely as an operator aid. This cannot be considered an acceptable level of safety since the operator's interface is not fail-safe. The safety-related regulations are further supervised by the fail-safe hardware which checks all messages to (and acknowledgements from) trains. This equipment is only concerned with safety:- the limited scope of the equipment allows the software to be minimal and critical safety software is therefore isolated witnin fail-safe hardware physically removed from the operational software running in the operator's interface. The critical software can be validated with extensive error-checking procedures and the validation effort can be confined only to those functions which are safety critical.

6.16 Conclusion.

The interlocking rules defined in this chapter clearly set out the restrictions which are imposed on the operator as a result of the existing circumstances of train positions, current movement authorities and blocking crosses. It is essential that these rules be reliably enforced by the equipment, and various methods used to ensure this are described in the following chapter in which a full discussion of system safety is given.

Chapter 7: The Concept of Safety.

7.0 General Safety Principles.

C.R.T.C. is a railway control system and hence must, as a system prevent the occurrence of dangerous situations with a large degree of integrity. The system must therefore minimise the possibility of a train collision by ensuring that conditions which may lead to a collision have a very small probability of arising.

One must accept that there is a tradeoff between 'safety', in the sense described above, and the 'efficiency' with which traffic is managed and that the achievement of 'absolute safety', with zero risk, is impractical and economically impossible. One must be satisfied with minimising the risk, within a given cost frame and this can be accomplished by the implementation of various safety principles. These principles are enforced in C.R.T.C. within the equipment itself and in the formulation of a regulated operating procedure. It is essential that all operating staff, both traffic controllers and train drivers, are trained in the correct use of the system. The C.R.T.C. system affords every practical aid towards the automatic enforcement of operating regulations, but cannot claim to provide any degree of safety without responsible, competent human operators and drivers (ref. 11).

Clearly, safety can only be achieved if the original specification of the system is acceptably safe, and any 'errors' made at this level cannot be protected against in the running system. Chapter 6 details the specification for C.R.T.C. in terms of a set of 'rules' which govern train movements within a single line network and acceptable safety is achieved in the specification by basing the rules on three major premises:-

1) Never allow more than one train to enter the same section;

2) Never allow more than one train movement (i.e. shunting, entering, or leaving) at the same station;

3) Never allow a train to proceed straight through an occupied station.

'Absolute Safety' would require that these rules are always adhered to regardless of errors, human or otherwise. Practically, however, one can only minimise the possibility of safety violations and this is achieved in C.R.T.C. in several ways:

1) High Reliability.

The reliability of a system is measured by the frequency of failures and high reliability is defined by a low frequency of failures. Human errors, and equipment failures will always occur with some degree of frequency and some percentage of these errors will cause unsafe situations. Minimising the occurrence of all failures serves to reduce the number of critical, unsafe failures.

2) Non-Provocation (ref. 11).

Non-provocation is a standard approach towards the achievement of safety in railway signalling. Each 'component' of a system is configured in such a way that, in the absence of any failure in that component, it can not pass an unsafe condition, or request, to the next component. For example, in C.R.T.C. the operator will be trained to follow a set of clear, unambiguous operating regulations. He is provided with an overview display of the C.R.T.C. network to aid in decision making, and he is aware of the fact that each transaction is being recorded, both the verbal part and the digital part. These factors combine to minimise the frequency at which he will even attempt to set up unsafe situations - hence he does not 'provoke' the system. Furthermore the driver is responsible for ensuring that transactions which are incorrect are not executed - again a case of non-provocation.

As described in Chapters 4,5 and 6 the operator's interface is designed to prevent unsafe requests from being passed to the safety equipment: the operator's interface therefore does *not* 'provoke' the safety equipment. Non provocation is therefore used extensively in C.R.T.C. to minimise the frequency of unsafe failures.

3) Redundancy (refs. 6,8,11,15).

Redundancy is a technique used to minimise the occurrence of unsafe failures. One function of a system is 'backed up' by a separate implementation of the same function. This means that an error in one component can be rapidly identified by the component which is providing the backup. Once the error has been identified, it can be isolated and its effect minimised. An example of the use of redundancy in C.R.T.C. is that each message received by the driver is given to him both verbally and electronically in the form of an electronic display. If the two implementations of the same message are not the same, the driver can identify the fact that an error has occurred and follow the safe procedure - to stop moving and to regard the authority as invalid, thereby preventing the error from causing an unsafe situation. The speech and data information passing provides an example of 'dual redundancy' in which an error can be detected but its source cannot be identified (the driver does not know whether the error was in the verbal message or in the electronic display). Triple redundancy is also used in C.R.T.C. in the fail safe processing equipment. Three microprocessors run in parallel and it an error occurs in one of the microprocessors the other two can isolate the faulty processor and continue as a dual redundant system. In this way the system is not degraded at all and can continue to run normally.

7.1 Error Sources in C.R.T.C.: Specification Errors.

Any implementation of a system can only be guaranteed to enforce the conditions specified in the specification of the system and it is therefore essential in C.R.T.C. to establish a suitable specification.

Extensive consultation with Signal Engineers and with Railway Transport Authorities representing drivers and traffic controllers was undertaken during the design of a specification for C.R.T.C. These consultations continued throughout the design and implementation of the 'operator's interface' which was used to illustrate all aspects of the system, including the essential 'interlocking rules'. The format of transactions and the rules which govern them (as described in Chapters 5 and 6) were found to provide an acceptable level of safety during these consultations.

7.2 Error Sources in C.R.T.C.: On-Line Errors.

Figure 7.1 illustrates the points at which errors could occur throughout the C.R.T.C. system and the following paragraphs describe the defences against such errors causing dangerous situations.

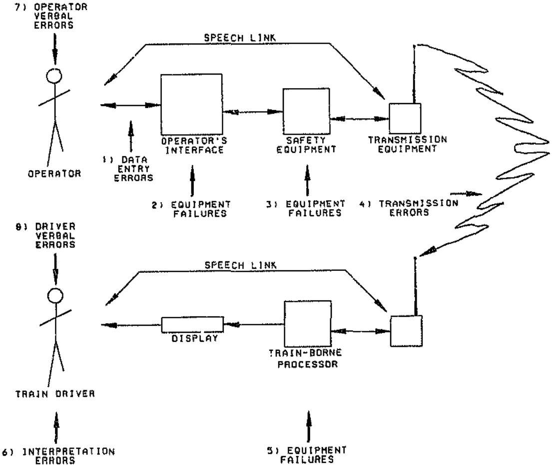


Figure 7.1: Error Sources in C.R.T.C.

1) Operator Data Entry Errors.

If the operator makes a error in entering data at the operator's interface he may immediately recover by aborting the transaction and re-entering the data. If, however, the operator attempts to execute a false transaction (for example executing a 'train clear' transaction before a driver has cleared a section) then it is the responsibility of the driver to refuse to accept the transaction. An operator error of this nature can only be promulgated if the driver makes a similar error, and this is therefore a case of redundancy.

2) Failures in the Operator's Interface.

The operator's interface is non fail-safe equipment and therefore an error in this equipment may cause unpredictable messages to be sent to the safety check equipment. The software for the operator's interface is written in a structured manner and has been extensively tested in an attempt to maximise the reliability of this interface so that the frequency of errors is low. The fail-safe equipment provides a back-up to the operator's equipment, and hence, again, dual redundancy is used to prevent the occurrence of errors from causing dangerous transactions.

3)Failures in the Safety Equipment.

Non-Provocation of the Safety Equipment by the Operator's Interface is used to minimise the frequency at which the Safety Equipment will be used to filter out unsafe requests. The equipment itself makes use of triple redundancy to protect against failures. Three microprocessors run in parallel and each microprocessor monitors all transmissions both to and from trains, and from the operator's interface. All actions taken by the equipment require a two-out-of-three majority decision before they can be executed, and hence a failure in one of the microprocessors will be detected by the other two and the offending microprocessor will be isolated.

All the software in the Safety Equipment is made error free using stringent, approved validation techniques and hardware watchdog timers monitor the execution of the software (by detecting infinite loops etc.) to ensure that the program runs consistently and correctly during the operation of C.R.T.C.

4)Transmission Errors.

All electronic messages which are transmitted to train drivers are sent in 'full text'. For example, an authority for a train to proceed would be built up with the full set of characters: "P-R-O-C-E-E-D- -F-R-O-M- -L-E-R-O-U-X- -T-O- -S-N-Y-B-E-R-G"

(Leroux and Snyberg being the source and destination stations respectively). The possibility of a transmissic distortion being undetected by the receiving equipment is reduced by the use of cyclic redundancy coding (CRC). Messages are sent in the format specified by the standard Synchronous Data Link Control (SDLC) protocol and the particular CRC used has a very low probability for undetectable errors. If an error is detected by the receiving equipment a full message re-transmission is requested. In the unlikely event that an undetected data transmission error passes the receiving equipment error checks the message displayed to the driver would be garbled, and hence ignored. The possibility of a message being corrupted to a sensible, but incorrect message (i.e. S-N-Y-B-E-R-G- being distorted to L-E-R-O-U-X-) as well as passing the CRC check is minute and can be ignored.

5)Train Equipment Failures.

The train-borne equipment is non fail-safe but is designed so that it cannot generate, on its own, a valid movement authority and display it to the driver. The characters used to build up station names, for example, are not stored anywhere in the equipment and hence, under failure conditions, no meaningful display will be given to the driver and he will not move. In the unlikely event of the equipment generating its own valid message the driver will ignore it since he will not have received a verbal authority.

The possibility that train-borne equipment will intercept messages addressed to other trains does exist, but in this case the driver would recognise that the message refers to another train and he would not have received verbal confirmation. Hence he would ignore the message. Furthermore, each set of train-borne equipment has a unique digital address which is transmitted with each message, and the data constituting this address is checked by CRC checking.

6)Driver Interpretation Errors.

One cause of fatal errors could be the driver misinterpreting the sense of an instruction given to him by the operator; however, the driver would have to misinterpret both verbal and visual messages for this to happen. In order to prevent this the wording of all movement authorities has been designed to clearly express the intention of the transaction. Messages are displayed in full text and station names are written out in full rather than giving code numbers or letters.

7) Operator 'Verbal' Errors.

The operator's interface provides him with the full text of any formal verbal instruction which he must give to the driver. If, however, the operator reads the message out incorrectly and assuming the driver 'accepts' the transaction, the driver's written authority will not agree with his verbal authority and the transaction should be considered by the driver to be invalid.

8) Driver 'Verbal' Errors.

One must acknowledge that C.R.T.C. places responsibility on the driver to follow the correct procedures. If the driver reports clear of a section before he has actually cleared it (*this constitutes a driver 'verbal' error*) and the 'clear' transaction is executed then the train will be in a section for which it has no authority. This is equivalent to entering a section without authority and there is no electronic protection against this. Regulations must be followed, and the knowledge that events are recorded (both the data and the verbal portions of transactions) should help to prevent transgressions of these regulations on the part of the driver.

7.3 Safety in C.R.T.C.

General safety in C.R.T.C. therefore relies on the fact that both data and verbal representations of an authority are required by the driver before he can move his train. The interlocking software prevents the inadvertent transmission of conflicting movement authorities, but relies on train drivers to report information correctly and the operator to enter it correctly. Both operator and drivers are relied on to abide by the operating regulations, and to abide by the movement instructions. Enhancements to C.R.T.C. could include Fail-Safe train detection equipment which would effectively remove some responsibility from the operator and drivers. However, both the operator's and the drivers' interfaces are designed to give as much aid in following the regulations as possible and, with suitable training and properly documented, unambiguous operating regulations the human aspect can be made a safe and reliable part of C.R.T.C.

7.4 Conclusion.

Safety in C.R.T.C., as an overall system, was therefore achieved by ensuring that the specification of interlocking rules was correct and complete and thereafter by minimising the probability of 'on-line' dangerous failures through the use of reliability, redundancy and non-provocation. Each error source was identified and isolated and suitable safety mechanisms were incorporated so that errors could be identified and prevented from causing dangerous violations of the operating regulations.

the operator's and the drivers' interfaces are designed to give as much aid in following the regulations as possible and, with suitable training and properly documented, unambiguous operating regulations the human aspect can be made a safe and reliable part of C.R.T.C.

7.4 Conclusion.

Safety in C.R.T.C., as an overall system, was therefore achieved by ensuring that the specification of interlocking rules was correct and complete and thereafter by minimising the probability of 'on-line' dangerous failures through the use of reliability, redundancy and non-provocation. Each error source was identified and isolated and suitable safety mechanisms were incorporated so that errors could be identified and prevented from causing dangerous violations of the operating regulations.

Chapter 8: Technical Details of C.R.T.C.

8.0 Introduction.

In order to create a system which could quickly and effectively demonstrate the system C.R.T.C. an IBM PC was programmed as a full operator's interface (including all the functions described in the previous chapters). The PC was furthermore used to simulate the components of C.R.T.C. which are external to the operator's interface - for example the keyboard of the PC is used to simulate the procedure by which the operator sets up a communications channel with a train, and a dot matrix printer is used to record all the messages which are sent to trains. A pushbutton mounted on the printer simulates the driver's "ACKNOWLEDGE" button.

8.1 Implementation - Software.

The IBM PC with two monitors, a keyboard and a printer provides an ideal hardware vehicle for the implementation of the operator's interface and for the simulation of the external constraints of the C.R.T.C. system.

The PC itself was used as a development tool for the C.R.T.C. software design and the configuration of the PC as an operator's interface was achieved by coding the software in Intel 8086/8 Assembly Language. This language was chosen for the following reasons:

1) The IBM PC uses an 8088 microprocessor as its CPU and hence there exists an 8088 Assembly Program to automate the translation of Assembly Language source software to the correct machine code;

2) Coding software into Assembly Language gives a very large degree of flexibility for driving I/O devices such as monitors, printers, etc. and this flexibility allows the I/O devices to be configured in the specific format suitable to C.R.T.C.;

3) It is possible to 'disassemble' machine code, thereby re-creating the original source Assembly Language. Hence, when de-bugging and testing a running program, the original source code and the disassembled machine code can be compared 'line for line'. This facility was used during the development of C.R.T.C. software to identify and correct errors quickly. It is not possible to 'decompile' machine code to re-create higher language source software;

4) The checking of disassembled machine code against original source code was used to ensure that there were no assembly errors i.e. that the assembler had not translated the source code incorrectly. Again, this cannot be done if high level languages are used, and compiler errors cannot be isolated in this way.

5) Coding software into Assembly Language is generally reputed to be a tedious, time consuming task. However, the author found that the major proportion of the effort spent in implementing the

system was in the design of the software structure. The effort spent in the final coding of the design into a target language proved to be insignificant when compared with the effort spent on software design, and hence the author did not consider this aspect in the choice of a programming language.

8.2 The Design Procedure.

The software implementation of the system 'C.R.T.C.' as it has been described in previous chapters required, first and foremost, that a number of compartmentalised tasks be defined. These tasks were then combined to perform the full system.

The fragmentation of a single, highly-complex system into a number of well-defined tasks served to simplify the design procedure significantly. The first level of fragmentation provided a set of general component tasks which were given the title "Operating System Modules" or OSMs in the C.R.T.C. system. Similarly each operating system module was broken up into a number of "Subroutines" and each subroutine in turn became the sum of a number of "Modules", so that at the level defined by the modules the software consisted of a large number of simple, well-defined tasks, each of which could be readily 'coded' or implemented in the target programming language. This is a standard approach to software design and, as illustrated in Figure 8.1, the shape of the resulting structure resembles an inverted 'tree', hence the name "Tree Structure".

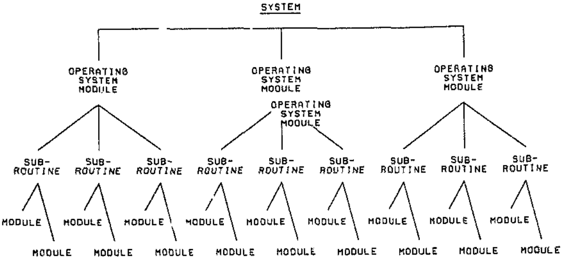


Figure 8.1: Tree Structured Software.

The software design procedure, therefore, involved the isolation of a number of specific activities which combined together to execute the general system called 'C.R.T.C.'

8.3 C.R.T.C. Simulator Software.

Figure 8.2 shows the operating system modules which constitute the operator's interface and C.R.T.C. simulator and the following paragraphs give a brief description of each OSM.

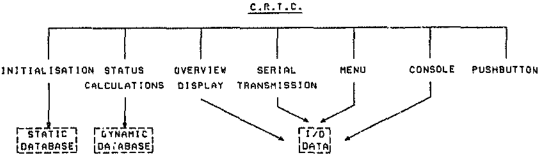


Figure 8.2: C.R.T.C. Structure.

1) Status Calrulations.

This OSM has no Input/Output portion and is responsible for keeping a record of all the trains in the system, the occupation status of each station and each section, the position and current authority of each train and the status of the radio (i.e. channel open or closed). The OSM provides a dynamic data base which is referred to by all the other OSMs to calculate interlocking and to drive the overview display, tne console, the menu and the message transmissions.

2) Menu.

The 'Menu' OSM has two sub-tasks. Firstly, on the basis of information provided by the 'status calculator' module, 'Menu' calculates a list of permissible pushbuttons and this list is used by the 'Pushbutton' module so that the system can identify a pushbutton when it is activated by the operator. Secondly, 'Menu' is responsible for the displaying of the permissible pushbuttons on the monochrome monitor.

3) Console.

'Console', like 'Menu', first calculates what should appear in each field of the console. Then it [illegible] ponsible for the physical outputting of characte [illegible] onsole section of the monochrome monitor in the correct [illegible].

4) Overview Display.

The 'Overview Display' OSM calculates overlay information, such as section colours and optic contents for the display, once an indication has been received from the status calculation program that a transaction has been successfully executed. This module then physically drives the display accordingly.

5) Serial Transmission.

On reception of an indication from the status calculation program that an execution is being attempted, this OSM transmits the message to the fail-safe equipment and awaits a return message signifying whether the transaction was successfully executed or not. If, for example, the driver does not acknowledge the message, then a 'fail' message is returned from the fail-safe equipment.

In the simulator, the 'Serial Transmission' OSM transmits messages to the printer. A 'Printer Ready/ Not Ready' indication is returned to simulate the success or failure message. This allows for the simulation of the case where a driver does not acknowledge the transaction, since the 'Select' pushbutton on the printer is used to simulate the driver's "ACKNOWLEDGE" pushbutton.

6) Pushbutton.

This operating system module performs an input from the keyboard and decodes a pushbutton request by consulting the Menu list. The pushbutton request is then passed to the status calculator which takes the appropriate action. The input portion of this module can readily be modified to allow inputs from an alternative source such as a digitiser pad.

7) Initialisation.

The 'Initialisation' OSM runs only on the starting up of the system, and it initialises all the I/O devices, calculates a static layout of the network to be displayed on the overview display and initialises all dynamic data to reflect that there are no trains in the system, no occupied stations and no radio channel as yet set up. Information such as the number of stations, the name of each station and the number of loops at each station is coded into a 'static' data base, and this data base can be configured very easily to represent the layout for any single line network.

8.4 Modularity and Modification.

C.R.T.C. software was designed, as described above, in a modular structure. In order for the modules to be bound together to form a coherent system, the design procedure also requires the definition of a 'Data Flow Structure'.

The 'Data Flow' defines how the activity performed by each module both depends on the results of another module as well as itself affecting the operation of a further module. This aspect of the design is critical with regard to the ongoing modification of the system, and maximum emphasis was placed on the 'modularity' of the software to allow for changes to be made quickly and simply in the detailed specification. To a certain extent, the software was designed in parallel with the specification and hence changes to certain details were often required. The modular nature of the design facilitated this in that the specific areas to be changed could be extracted (as complete modules), re-designed and re-inserted. The data flow structure was defined in such a way that this process would not disturb or disrupt the flow of data and only relevant modules needed to be changed. The comments made above can be illustrated by the example shown in Figures 8.3 and 8.4.

TASK: CALCULATE THE ALLOWED DIRECTION FOR A TRAIN TO PROCEED.

INPUT DATA: TRAIN NUMBER
OUTPUT DATA: DIRECTION

PROCEDURE: IF THE TRAIN NUMBER IS EVEN THEN THE DIRECTION IS 'UP'.

IF THE TRAIN NUMBER IS ODD THEN THE DIRECTION IS 'DOWN'.

Figure 8.3: Software Module.

In this case the module is simple and well defined. The source data, which is provided by some other module, is the train number. The module in question uses the train number as input data and calculates a result (i.e. the train direction, 'up' or 'down') which can be passed on to another module which uses it (for example) to create a list of allowed destination stations. Assuming that the requirement changes, and that for example a 2, 4, or 6 digit train number defines an 'up' train while a 1, 3, or 5 digit train number defines a 'down' train. The module can be extracted, and the procedure redefined as shown in Figure 8.4. The format of the input and output data is not changed, and so neither the module supplying the input nor the module using the output needs to be re-designed.

TASK: CALCULATE THE ALLOWED DIRECTION FOR A TRAIN TO PROCEED.

INPUT DATA: TRAIN NUMBER

OUTPUT DATA: DIRECTION

PROCEDURE: IF THE TRAIN NUMBER HAS 2, 4 OR 6 DIGITS THEN THE DIRECTION IS 'UP'.

IF THE TRAIN NUMBER HAS 1, 3 OR 5 DIGITS THEN THE DIRECTION IS 'DOWN'.

Figure 8.4: Software Module.

8.5 Software Validation.

Essentially, validating a system involves the testing of the system to prove that it exactly implements the all the requirements laid out in the specification (and only the requirements laid out in the specification) under all circumstances. The structure of the software to be tested affects the task of validation to a large degree. Well defined, loosely coupled modules can be tested and validated individually, and since each module is an entity which defines a certain small function, the validity of the module can be established and proved in isolation from the rest of the system. For this reason the software for C.R.T.C. was written in the structured format described above. Secondary functions (such as the display of colour information on the colour monitor for example) are not critical in terms of the safety of the system as a whole and full validation techniques are not called for. However, in order to produce a reliable system, and one which can be guaranteed to behave to a large extent as specified, it is advantageous to structure the software in a manner which will facilitate the testing of the software.

8.6 Conclusion.

The C.R.T.C. simulator was therefore designed and built to incorporate all the functions enumerated in the specification for C.R.T.C. The IBM PC was programmed in assembly language and the software was designed in a structured, modular fashion to facilitate modification, expansion and error checking.

Chapter 9: Conclusion.

9.0 The End Users of the System.

The System C.R.T.C. originated from a requirement to design and build a safe, efficient, cost-effective system for the control of single line railways. The major part of this project has been to design a full specification for such a system and this was achieved initially by drawing up a series of documents which served to establish the general framework of the system. These documents were used as a means of formally communicating the proposed design of the system both to those people qualified to judge the merits of the system and to those people who were likely to become direct users of the system, such as Transport Authority Personnel. Many of the principles of operation of C.R.T.C. were therefore established through communication with the above groups of people.

Once the general structure of the system had been concluded in this way, it was decided that a full operator's interface should be built so as to present graphically the details of operation of C.R.T.C. to potential customers. In this way, any changes to the details of the system or any specific requirements suggested by the potential client could be illustrated graphically via the actual operator's interface, instead of depending solely upon the verbal description of the effects of such alterations.

The final system specification, as demonstrated using the IBM PC, met with the approval of signal engineers, and personnel from several transport authorities representing drivers and operators.

9.1 The Achievements of the Project.

The C.R.T.C. system operating specification has been completed, and defines a cost-effective single line control system. The inherent measures described in Chapter 7 provide the system with an acceptable level of integrity, and the operator and driver interfaces allow for easy operation. The set of operating procedures described in Chapters 4 and 5 combine to maximise the efficiency with which train traffic in a network can be managed. The resultant system therefore achieves the objectives of cost-effectiveness, safety and efficiency.

9.2 A Production Level C.R.T.C. System.

The IBM Personal Computer running C.R.T.C. software can be seen as a fully operational operator's interface with additional functions effectively implementing a simulation of the full C.R.T.C. System. A full production level system consists of the following components:

1) An Operator's Interface.

This interface performs all the functions described in the preceding chapters of the dissertation, giving the operator a data entry / display device and an overview display.

2) Radio Equipment and a Radio Network.

In general a C.R.T.C. communications network operates as follows:

- the radio network has full range from the control room to all points of a single line network in which Centralised Radio Train Control is operational;

- train-borne and control-room equipment transmit and receive both speech and digitally-encoded signals;

- the train-borne processing equipment provides the driver with a means for sending a request to the control for a communication channel to be opened;

- the control room radio equipment provides the operator with a display of outstanding requests from drivers and a means of selecting a target train for communication. The identification number (or radio number) of the selected train is available for processing by the operator's interface.

3) A Fail-Safe Supervisory Processor.

As described in earlier chapters of this dissertation a fail-safe supervisory processor checks all transactions before they are transmitted to trains, and ensures that the correct communications procedures are followed. Software in the processor is error-free and prevents the violation of any interlocking rules. The software is validated and runs in a fail-safe triple processor system.

4) Train-Borne Processing and Display Equipment.

A single microprocessor system installed in each train receives digital messages and drives a small display unit (two lines of 80 characters). The display is driven in a fail-safe manner i.e. messages are not latched in the display but require the active presence of the microprocessor cyclically to refresh the contents of the display. Failures in the microprocessor therefore result in a blank, or null message display.

9.3 The Present Status of the System.

To date the C.R.T.C. operator's interface has been completed and a full users' specification designed. A fail-safe triple processor system has been designed and built (ref.6) and an error-free operating system for the triple processor system has been completed. It remains to configure the fail-safe equipment specifically for the application of C.R.T.C., to implement a suitable radio network and to design and build train-borne processor and display equipment.

9.4 Extensions to the System.

As described in Chapter 8 the software was designed in such a way as to allow modifications, and future additions. One of the major areas which could be investigated is the implementation of the automatic issue of proceed authorities and the automatic clearance of sections. This would significantly increase the amount of traffic which could be handled by a single operator.

Automation could be achieved by installing trackside transponders at both ends of each station. A reader mounted on board the train would automatically recognise the transponder as the train approaches a station and transmit a request for the issue of an authority to enter the next section. Once a train has entered the next section the reader would recognise a second transponder (mounted at the station exit) and automatically notify the control centre, and the previous section (the section vacated by the train) would be released.

9.5 The Future.

The aim of C.R.T.C. is to save sections which may, with normal signalling systems, be closed due to uneconomical running costs. However, C.R.T.C. may also prove to be a step towards a safe, automated train control system.

REFERENCES.

1. BARKER C.P. (Telkor Signalling (Pty) Ltd.) - "Users' Specification : Centralised Radio Train Control". Telkor Internal Publication, January 1984.

2. BARKER C.P. (Telkor Signalling (Pty) Ltd.) - "Users' Specification : Centralised Radio Train Control". Telkor Internal Publication, February 1984.

3. BIRKIN M.S. - "Low Cost Signalling By Radio". Paper delivered to the Institution of Railway Signal Engineers in London, November 1984.

4. BRITISH COLUMBIA RAILWAYS, GLENAYRE ELECTRONICS LTD., DATAP SYSTEMS - "Location Identification and Control". C&S Division, AAR Committee Reports and Technical Papers, November 1983.

5. DEUTSCH M.S. - "Software Verification and Validation", *Prentice-Hall*, 1982.

6. LAMBRECHS J.S.D. - "The Application of Micro-Electronics in Fail-Safe Systems". Thesis presented for the Degree of Doctor of Philosophy, University of the Witwatersrand, Johannesburg, September 1983.

7. LAMBRECHS J.S.D. - "Proposed Fail-Safe and Reliability Analysis". Document presented at a meeting of Telkor Signalling (Pty) Ltd. and the South African Transport Services, December 1983.

8. LAMBRECHS J.S.D., RODD M.G. - "The Application of Microprocessors in Fail-Safe Equipment". Paper read at the 2nd technical meeting of the South African Section of the Institution of Railway Signal Engineers, Johannesburg, November 1981.

9. LAMBRECHS J.S.D., RODD M.G. - "The use of Computers in Safety Systems". Pulse Magazine, November 1982.

10. LAMBRECHS J.S.D., RODD M.G. - "Highly Reliable Software for use in Fail-safe Control". 3rd IFAC/IFIP Conference on Software for Computer Control, Madrid, October 1982.

11. MILLS J.D. - "Processor Based Safety Systems". Paper delivered to the Institution of Railway Signal Engineers in London, November 1984.

12. "Radio Data Links can Replace Low-Intensity Signalling". Article published in Radio Gazette International, November 1982.

13. REUNERT & LENZ LTD. - "Control of Single Line Railways by the Webb-Thomson Miniature Train Staff Apparatus". Internal publication, 1914.

14. SHORT R.C. - "Sof[illegible] [illegible]alidation for a Railway Signalling System". Paper read at an IFAC conference, Cambridge UK, 1983.

15 SHORT R.C. - "The Design of Fail-Safe Processor Systems". Paper presented at a technical Meeting of the I.E.E., January, 1980.

16. SMITH B.L. - "Vital Microprocessor Interlocking Demonstration". C&S Division, AAR Committee Reports and Technical Papers, 1983.

17. SOUTH AFRICAN RAILWAYS - "Control of Trains over Single Lines, Train Token Method, van Schoor Instruments". Internal Circular no. X.40, November 1949.

18. TELKOR SIGNALLING (PTY) LTD. - Minutes of meetings held to discuss C.R.T.C. proposals. Internal publication, January 1982 - December 1984.

19. TELKOR SIGNALLING (PTY) LTD. - "Preliminary Specification Radio Block Signalling System". Letter sent to the Chief Engineer (S&T), South African Transport Services, May 1983.

20. WHITEHOUSE W.H. - "The Bridge Between Conventional and New Signalling Technology". C&S Division, AAR Committee Reports and Technical Papers, 1983.

BIBLIOGRAPHY.

1. ANDREWS DR. M. - "Human Factors in Train Operation". Paper delivered to the Institution of Railway Signal Engineers in London, November 1979.

2. BARKER C.P. (Telkor Signalling (Pty) Ltd.) - "Users' Specification : Centralised Radio Train Control". Telkor Internal Publication, January 1984.

3. BARKER C.P. (Telkor Signalling (Pty) Ltd.) - "Users' Specification : Centralised Radio Train Control". Telkor Internal Publication, February 1984.

4. BARNARD R.E.B. - "Safety Analysis of Signalling Electronics". Paper delivered to the Institution of Railway Signal Engineers in London, November 1980.

5. BIRKIN M.S. - "Low Cost Signalling By Radio". Paper delivered to the Institution of Railway Signal Engineers in London, November 1984.

6. BRITISH COLUMBIA RAILWAYS, GLENAYRE ELECTRONICS LTD., DATAP SYSTEMS - "Location Identification and Control". C&S Division, AAR Committee Reports and Technical Papers, November 1983.

7. DEUTSCH M.S. - "Software Verification and Validation", Prentice-Hall, 1982.

8. COURDON G., DENIS P. - "Telecommunications on the SNCF". Paper delivered to the Institution of Railway Signal Engineers in London, December 1983.

9. HEWLETT PACKARD - "Optoelectronics Designers' Catalogue", 1983.

10. IBM CORPORATION - "Technical Reference Personal Computer XT" May 1983.

11. IBM CORPORATION / MICROSOFT CORPORATION - "Disk Operating System (DOS)", May 1983.

12. IBM CORPORATION - "Macro Assembler by Microsoft". May 1983.

13. INTEL CORPORATION - "Microprocessor and Peripheral Handbook", 1983.

14. INTEL CORPORATION - "MCS 80/85 Family User's Manual, 1979.

15. INTEL CORPORATION - "MCS-48 Family of Single Chip Microcomputers User's Manual, 1981.

16. JORDAN A.B. - "Simultaneous Speech plus 1200 Baud Data Transmission". Proceedings of the 21st Annual Meeting, Association of American Railroads.

17. KNOWLES T. - "Automatic Train Control for Heavy Mineral Railways". Paper delivered to the Institution of Railway Signal Engineers in London, January, 1984.

18. LAMBRECHS J.S.D. - "The Application of Micro-Electronics in Fail-Safe Systems". Thesis presented for the Degree of Doctor of Philosophy, University of the Witwatersrand, Johannesburg, September 1983.

19. LAMBRECHS J.S.D. - "Proposed Fail-Safe and Reliability Analysis". Document presented at a meeting of Telkor Signalling (Pty) Ltd. and the South African Transport Services, December 1983.

20. LAMBRECHS J.S.D. - "Validation Procedures for Telkor Fail-Safe Microprocessor Software". Document presented at a meeting of Telkor Signalling (Pty) Ltd. and the South African Transport Services, February 1984.

21. LAMBRECHS J.S.D., RODD M.G. - "The Application of Microprocessors in Fail-Safe Equipment". Paper read at the 2nd technical meeting of the South African section of the Institution of Railway Signal Engineers, Johannesburg, November 1981.

22. LAMBRECHS J.S.D., RODD M.G. - "The use of Computers in Safety Systems". Pulse Magazine, November 1982.

23. LAMBRECHS J.S.D., RODD M.G. - "Highly Reliable Software for use in Fail-safe Control". 3rd IFAC/IFIP Conference on Software *for Computer Control*, Madrid, October 1982.

24. LEWIS T.G., SMITH N.Z. - "Applying Data Structures". Houghton Mifflin, 1982.

25. McCORD D.W. - "Electronics and Signals - A Signal Engineer's Point of View". C&S Division, AAR Committee Reports and Technical Papers, 1983.

26. MILLS J.D. - "Processor Based Safety Systems". Paper delivered to the Institution of Railway Signal Engineers in London, November 1984.

27. MORGAN L., WAITE M. - "Graphics Primer for the IBM PC". Osborne/Mcgraw-Hill, 1983.

28. MORSE C.W. - "The Role of Computers in Centralised Train Control". Paper presented to the Annual Meeting, Communication and Signalling Section, Association of American Railroads in Toronto, 1976.

29. MORSE S.P. - "The 8086 Primer", Hayden, 1980.

30. MYERS G.J. - "Software Reliability", Wiley,1976.

31. PALM E. - "Modern Signalling Concepts of Swiss Federal Railways Part 1: Computers and Colour Visual Display Units". Paper delivered to the Institution of Railway Signal Engineers in Zurich, February 1984.

32. POPE R. - "Signalling Projects in East Anglia". Paper delivered to the Institution of Railway Signal Engineers in London, November 1983.

33. "Radio Data Links can Replace Low-Intensity Signalling". Article published in Radio Gazette International, November 1982.

34. REUNERT & LENZ LTD. - "Control of Single Line Railways by the Webb-Thomson Miniature Train Staff Apparatus". Internal publication, 1914.

35. SCANLON L.J. - "IBM PC Assembly Language: A Guide for Programmers". Prentice-Hall, 1983.

36. SHORT R.C. - "Software Validation for a Railway Signalling System". Paper read at an IFAC conference, Cambridge UK, 1983.

37. SHORT R.C. - "The Design of Fail-Safe Processor Systems". Paper presented at a technical Meeting of the I.E.E., January, 1980.

38. SIGNETICS - "TTL Logic Data Manual", 1982.

39. SMITH B.L. - "Vital Microprocessor Interlocking Demonstration". C&S Division, AAR *Committee Reports and Technical Papers*, 1983.

40. SOUTH AFRICAN RAILWAYS - "Control of Trains over Single Lines, Train Token Method, *van Schoor Instruments*". Internal Circular no. X.40, November 1949.

41. SOUTH AFRICAN TRANSPORT SERVICES, Signal Engineering - "Train *Working Rules*". *Internal* Publication.

42. SOUTH AFRICAN TRANSPORT SERVICES, - "Working of Trains by means of Centralised Traffic Control". Internal Publication.

43. "The Challenge of New Technology". Article published in International Railway Journal, November 1983.

44. TELKOR SIGNALLING (PTY) LTD. - Minutes of meetings held to discuss C.R.T.C. proposals. Internal publication, January 1983 - December 1984.

45. TELKOR SIGNALLING (PTY) LTD. - "Specification for a Fail-Safe Radio-Link Cab Signalling System". Internal publication, October 1982.

46. *TELKOR SIGNALLING (PTY) LTD.* - "Preliminary Specification Radio Block Signalling System". Letter sent to the Chief Engineer (S&T), South African Transport Services, May 1983.

47. TURNER J. - "The Structure of Modular Programs". Communications of the ACM, vol. 23, no. 5, pp 272-277, May 1980.

48. WHITEHOUSE W.H. - "The Bridge Between Conventional and New Signalling Technology". C&S Division, AAR Committee Reports and Technical Papers, 1983.

49. WILSON W.G.J. - "Control Panels Versus Video Display Units". Paper delivered to the Institution of Railway Signal Engineers in London, October, 1983.

Appendix A: Configuration of a C.R.T.C. Layout.

The software for the operator's interface includes a module which defines the layout of the network to be controlled, giving the station names, number of loops, etc. This module is called "CRTCLAYT.ASM". The following is extracted from the data module, and illustrates an example layout.

The following diagram is a representation of a Station. Each possible 'position' at which a railway track may exist is numbered as shown in the diagram, i.e. positions 1, 2, 3, 4, 5. The data description of each station includes a numeric value associated with each position shown in the diagram below. For example, a station may have no loops and only a 'main line'. In this case the coding for positions 1, 2, 4, and 5 would be 'non-existent', and the coding for position 3 would be 'main line'. The numeric representations for each of the attributes (non-existent, main line, loop etc.) which a position may be given are defined in the list below the diagram.

EXAMPLE STATION LAYOUT.

```
              -----------------        POSITION 1
              I               I
              I               I
              I               I
              -----------------        POSITION 2
              I               I
              I               I
              I               I
-------------------------------------- POSITION 3
              I               I
              I               I
              I               I
              -----------------        POSITION 4
              I               I
              I               I
              I               I
              -----------------        POSITION 5
```

POSITION DESCRIPTION CODING:

ØØ------NON-EXISTENT (i.e. there is no track in this position in the station)

Ø1------MAIN LINE

Ø2------LOOP 1

Ø3------LOOP 2

Ø4------LOOP 3

Ø5------LOOP 4

Each Station is represented by a string of bytes which gives the position description code number for each position, and the station name. The string of characters giving the station name should be terminated with a $ ('dollar' sign) and each station entry should be filled to 3Ø bytes. The stations should be stored in the following order:

ST1 is the DOWN end station

ST2 is the next station along

etc.

The example given below sets up a network with 18 Stations, 'Vanwykskraal' is the 'down' end Station, and 'Humefield' is the 'up' end Station.

```
ST1     DB      00      ;INDICATES POSITION 1=NON-EXISTENT
        DB      00      ;INDICATES POSITION 2=NON-EXISTENT
        DB      01      ;INDICATES POSITION 3=MAIN LINE
        DB      02      ;INDICATES POSITION 4=LOOP 1
        DB      00      ;INDICATES POSITION 5=NON-EXISTENT
        DB      'Vanwykskraal$'         ;STATION NAME
        DB      12 DUP(?)               ;FILL TO 30 BYTES

ST2     DB      00,00,01,02,00,'Le Roux$',17 DUP(?)
ST3     DB      00,00,01,02,00,'Marevlakte$',14 DUP(?)
ST4     DB      00,02,01,00,00,'Vlakteplaas$',13 DUP(?)
ST5     DB      00,02,01,00,00,'Rooiloop$',16 DUP(?)
ST6     DB      00,02,01,00,00,'Snyberg$',17 DUP(?)
ST7     DB      00,00,01,00,00,'ESCOM PVT #1$',12 DUP(?)
ST8     DB      00,02,01,00,00,'Toorwater$',15 DUP(?)
ST9     DB      00,00,01,02,00,'Vondeling$',15 DUP(?)
ST10    DB      00,00,01,02,00,'Antonie$',17 DUP(?)
ST11    DB      00,00,01,02,00,'Willowmore$',14 DUP(?)
ST12    DB      00,00,01,02,00,'Oven     $',15 DUP (?)
ST13    DB      00,02,01,03,00,'Knoetze  $',15 DUP (?)
ST14    DB      00,00,01,02,00,'Fullarton$',15 DUP (?)
ST15    DB      00,00,01,02,00,'Draaiberg$',15 DUP (?)
ST16    DB      00,02,01,00,00,'Miller   $',15 DUP (?)
ST17    DB      00,00,01,02,00,'Fern     $',15 DUP (?)
ST18    DB      00,00,01,02,00,'Humefield$',15 DUP (?)
```

Appendix B: Description of Modules Linked into C.R.T.C.

The following is a full list of all the files which are assembled and linked together into C.R.T.C. Each programme module is given a level number which indicates its position in the structure of C.R.T.C. and a short description of the function of the module.

1.CRTCLAYT.ASM

This is a data module which defines the geographical layout of the network under control.

2.CRTCCH1.ASM

This is a data module which defines a number of character strings used in driving the console.

3.CRTCCH2.ASM

This is a data module which defines a number of character strings used in driving the menu.

4.CRTCMAC.ASM

This is a module containing definitions of all the macros used in the system. It is 'included' in each module which uses macros.

5.CRTCEQU.ASM

This module defines a number of constants which are used throughout the system. It is 'included' in each module which refers to any of the constants.

6.CRTCEQU2.ASM

This module defines a number of constants which are used throughout the system. It is 'included' in each module which refers to any of the constants.

7.IOINT.ASM

This is an initialisation module which creates a stack area and initialises segment registers in the 8088.

8.CRTCINTX.ASM

This is an initialisation module which initialises the dynamic database i.e. all data which is written by the subroutine PROCCRTC. This data includes a list of stations (which are made all 'clear' on initialisation) and a list of trains (which is made 'empty' on startup).

9.INTBWMON.ASM

This is an initialisation module which initialises the monochrome monitor. The program creates the features which are displayed permanently on the monitor (such as the menu outline).

10.DISPINT.ASM

This is an initialisation module which initialises the colour monitor, and creates the static portion of the overview display.

11.DRINT.ASM

This is an initialisation module which initialises a data buffer which refers to the colour display information. This buffer is called a 'responsibility' buffer and contains records of each train which is responsiole for any overlay on the display.

12.PRNINT.ASM

This is an initialisation module which initialises the printer, causing a title to be printed, and the date and time.

13.PBINT.ASM

This is an initialisaticn module which initialises a list which is used to determine presently selected, and presently activated pushbuttons from the menu list.

14.MAIN2.ASM - LEVEL 1.

This is the main program for running C.R.T.C. which calls programs on lower levels.

15.ST1PROC.ASM - LEVEL 2.

This is one of the Operating System Modules which is executed when the status (value of INST - see Appendix D) is 1.

16.ST2PROC.ASM - LEVEL 2.

This is one of the Operating System Modules which is executed when the status (value of INST - see Appendix D) is 2.

17.ST3PROC.ASM - LEVEL 2.

This is one of the Operating System Modules which is executed when the status (value of INST - see Appendix D) is 3.

18.ST4PROC.ASM - LEVEL 2.

This is one of the Operating System Modules which is executed when the status (value of INST - see Appendix D) is 4.

19.ST5PROC.ASM - LEVEL 2.

This is one of the Operating System Modules which is executed when the status (value of INST - see Appendix D) is 5.

20.VDUSCRL.ASM - LEVEL 2.

This is a special routine which is called from MAIN2 to scroll the colour monitor. The module directly accesses hardware VDU drivers to maximise the speed at which this is done.

21.CRTCTEST.ASM - LEVEL 2.

This is a special module called from MAIN2 which sets up a simulated run (simulating the operator with random keystrokes).

22.INFLD.ASM - LEVEL 3.

This module may be called from any of the 5 OSMs and it creates an 'instruction' field which is displayed on the console to tell the operator which keys he is allowed to select at any time. The instruction field appears at the bottom of the console.

23.SERIAL.ASM - LEVEL 3.

This module is called by the OSMs ST4PROC and ST5PROC. The module implements serial transmission of messages to the printer.

24.RAND.ASM - LEVEL 3.

This is a module which is called by the CRTCTEST module to provide a random number.

25.CONIO.ASM - LEVEL 3.

This module takes the lists of characters which have been assembled for display on the console, and outputs them to the device.

26.MENIO.ASM - LEVEL 3.

This module takes the lists of characters which have been assembled for display on the menu, and outputs them to the device.

27.PROCCRTC.ASM - LEVEL 3.

This module calls all the routines used to keep track of the trains, stations, and current status (in the variable MST - see Appendix D).

28.DISPLAY.ASM - LEVEL 3.

This module calls all the routines used to calculate and output information to the colour monitor.

29.CONSOLE.ASM - LEVEL 3.

This module calls all the routines used to calculate information concerning the characters to be displayed on the console.

30.MENU.ASM - LEVEL 3.

This module calls all the routines used to calculate information concerning the pushbuttons which should be active, and the pushbutton labels to be displayed on the menu.

31.PROROP.ASM - LEVEL 3.

This module assembles data to be passed to the PROCCRTC status calculation module in the event of a radio channel being opened.

32.PRORC.ASM - LEVEL 3.

This module assembles data to be passed to the PROCCRTC status calculation module in the event of a radio channel being closed.

33.PBRQ.ASM - LEVEL 3.

This module assembles data to be passed to the PROCCRTC status calculation module in the event of a pushbutton being selected.

34.PROARR.ASM - LEVEL 3.

This module is called when the operator instigates a 'select up' or 'select down' of the menu and keeps an internal record of the currently selected Pb.

35.PROSCR.ASM - LEVEL 3.

This module is called when the operator instigates a 'scroll menu' and keeps an internal record of the currently displayed range of the menu.

36.CRTC711.ASM - LEVEL 4.

This is the first of the status control calculation modules called by PROCCRTC which decides which of the following modules should react.

37.CRTC721X.ASM - LEVEL 4.

Reacts only when a successful execute has been achieved. Calculates a new status for the train, and all the new information concerning the train (location, position, destination etc.)

38.CRTC722X.ASM - LEVEL 4.

Reacts only when a successful execute has been achieved. Puts the new train values calculated by CRTC721 into the trains list which is a buffer starting at BUF722.

39.CRTC723X.ASM - LEVEL 4.

Reacts only when a successful execute has been achieved. Calculates new information concerning the stations (occupation, through stations, destination stations etc) and puts the information into the stations list (BUF723).

40.CRTC724X.ASM - LEVEL 4.

Reacts only when a secondary transaction execution has been achieved. Keeps a record of blocking cross numbers, locations and positions.

41.CRTC731X.ASM - LEVEL 4.

Reacts only when a change in the status of the radio has occurred, either a channel has been opened or it has been closed. In the case of a channel opening the module extracts train information from the trains list into a temporary working list.

42.CRTCFL.ASM - LEVEL 4.

Reacts only when there has been an attempt at executing a transaction which has failed.

43.CRTC741X.ASM - LEVEL 4.

Only reacts when a pushbutton has been depressed. Keeps a record of information associated with pushbutton entries (i.e. location numbers, position numbers etc.) and calculates new status values for MST (see Appendix D).

44.CRTC751X.ASM - LEVEL 4.

This module is responsible for gathering information generated by the previous 7 modules and collating it into a temporary working database, giving train status of the current train, location, position, destination, etc.

45.CRTC760X.ASM - LEVEL 4.

This module simply extracts the value of the status MST751 into MST760 which is used in calculating a new menu, new console and new display.

46.CRTC770.ASM - LEVEL 4.

This module calculates and stores the addresses at which the character strings for station names and position names are stored, starting with a station number, and a position number for location, position, destination location, destination position etc.

47.FLDCLR.ASM - LEVEL 4.

This module is called from CONSOLE, and it clears all the character strings for each console field prior to new character strings being calculated by the following modules.

48.CRTCCON1.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is Ø1.

49.CRTCCON2.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is Ø2.

5Ø.CRTCCON3.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is Ø3.

51.CRTCCON4 ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is Ø4.

52.CRTCCON5.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is Ø5 or Ø6.

53.CRTCCON7.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 07.

54.CRTCCON8.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 08.

55.CRTCCON9.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 09 or 0A.

56.CRTCCONB.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 0B,45,46,47,48,55 or 56.

57.CRTCCONC.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 0C or 15.

58.CRTCCOND.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is ØD or 16.

59.CRTCCONE.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is ØE or 17.

6Ø.CRTCCONF.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is ØF.

61.CRCONAA.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is AA.

62.CRCONAB.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is AB.

63.CRCONAC.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is AC.

64.CRCONAD.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is AD or AE.

65.CRCONAF.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is AF.

66.CRCONBØ.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is BO or B1.

67.CRCON1Ø.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 1Ø or 11.

68.CRCON12.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 12 or 13.

69.CRCON14.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 14,49,4A,4B,4C,57, or 58.

70.CRCON18.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 18 or 19.

71.CRCON1A.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 1A.

72.CRCON1E.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 1E or 1F.

73.CRCON20.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 20.

74.CRCON24.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 24 or 25.

75.CRCON27.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 27 or 30.

76.CRCON28.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 28 or 31.

77.CRCON29.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 29 or 32.

78.CRCON35.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 35 or 36.

79.CRCON26.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 26,4D,4E,4F,50,59,5A.

80.CRCON2A.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 2A.

81.CRCON2B.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 2B or 2C.

82.CRCON2D.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 2D or 2E.

83.CRCON2F.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 2F,51,52,53,54,5B, or 5C.

84.CRCON33.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 33.

85.CRCON37.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 37.

86.CRCON38.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 38 or 39.

87.CRCON40.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 40.

88.CRCON41.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 41.

89.CRCON42.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 42 or 43.

90.CRCON45.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 45,49,4D, or 51.

91.CRCON46.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 46,4A,4E OR 52.

92.CRCON47.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 47.

93.CRCON48.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 48,4C,50, or 54.

94.CRCONSTP.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores overlay character strings for the extra status field when the status value of MST (see Appendix D) is 0B,14,26, or 2F AND the stopped flag is set for the train, i.e. the train has been stopped.

95.CONCPV.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 55,57,59, or 5B.

96.CONCPX.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 56,58,5A, or 5C.

97.CRCON60.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 60.

98.CRCON61.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 61.

99.CRCON62.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores the character strings for each field when the status value of MST (see Appendix D) is 62.

1ØØ.CRCONER.ASM - LEVEL 4.

This module is called from CONSOLE. It calculates and stores overlay character strings for the error field when an execute attempt has failed.

1Ø1.CRMENAA.ASM - LEVEL 4.

This module is called from MENU. It calculates and stores the code numbers and character strings for each pushbutton when the status value of MST (see Appendix D) is AA.

1Ø2.CRMEN1.ASM - LEVEL 4.

This module is called from MENU. It calculates and stores the code numbers and character strings for each pushbutton when the status value of MST (see Appendix D) is Ø1,Ø2,6Ø, or 61.

103.CRMEN3.ASM - LEVEL 4.

This module is called from MENU. It calculates and stores the code numbers and character strings for each pushbutton when the status value of MST (see Appendix D) is Ø3 or AB.

104.CRMEN4.ASM - LEVEL 4.

This module is called from MENU. It calculates and stores the code numbers and character strings for each pushbutton when the status value of MST (see Appendix D) is Ø4.

105.CRMEN5.ASM - LEVEL 4.

This module is called from MENU. It calculates and stores the code numbers and character strings for each pushbutton when the status value of MST (see Appendix D) is Ø5,Ø9,ØD,1Ø,12, 16,18,1C,1E,22,24,28,2B,2D,31,33,35,38,42,4B,4F,53,55,57,59, 5B, or 62.

106.CRMEN6.ASM - LEVEL 4.

This module is called from MENU. It calculates and stores the code numbers and character strings for each pushbutton when the status value of MST (see Appendix D) is Ø6,ØA,ØE,11,13,17, 19,1D,1F,23,25,29,2C,2E,32,34,36,39,43,48,4C,5Ø,54,56,58,5A, 5C,AE,B1, or 63.

107.CRMEN7.ASM - LEVEL 4.

This module is called from MENU. It calculates and stores the code numbers and character strings for each pushbutton when the status value of MST (see Appendix D) is Ø7.

108.CRMEN8.ASM - LEVEL 4.

This module is called from MENU. It calculates and stores the code numbers and character strings for each pushbutton when the status value of MST (see Appendix D) is 08.

109.CRMENB.ASM - LEVEL 4.

This module is called from MENU. It calculates and stores the code numbers and character strings for each pushbutton when the status value of MST (see Appendix D) is 0B,26,2F, or 14.

110.CRMENC.ASM - LEVEL 4.

This module is called from MENU. It calculates and stores the code numbers and character strings for each pushbutton when the status value of MST (see Appendix D) is 0C,27,30, or 15.

111.CRMENF.ASM - LEVEL 4.

This module is called from MENU. It calculates and stores the code numbers and character strings for each pushbutton when the status value of MST (see Appendix D) is 0F,20, or 2A.

112.CRMEN1A.ASM - LEVEL 4.

This module is called from MENU. It calculates and stores the code numbers and character strings for each pushbutton when the status value of MST (see Appendix D) is 1A.

113.CRMEN37.ASM - LEVEL 4.

This module is called from MENU. It calculates and stores the code numbers and character strings for each pushbutton when the status value of MST (see Appendix D) is 37.

114.CRMEN40.ASM - LEVEL 4.

This module is called from MENU. It calculates and stores the code numbers and character strings for each pushbutton when the status value of MST (see Appendix D) is 40.

115.CRMEN41.ASM - LEVEL 4.

This module is called from MENU. It calculates and stores the code numbers and character strings for each pushbutton when the status value of MST (see Appendix D) is 41.

116.CRMEN45.ASM - LEVEL 4.

This module is called from MENU. It calculates and stores the code numbers and character strings for each pushbutton when the status value of MST (see Appendix D) is 45,49,4D, or 51.

117.CRMEN46.ASM - LEVEL 4.

This module is called from MENU. It calculates and stores the code numbers and character strings for each pushbutton when the status value of MST (see Appendix D) is 46,4A,4E, or 52.

118.CRMENAC.ASM - LEVEL 4.

This module is called from MENU. It calculates and stores the code numbers and character strings for each pushbutton when the status value of MST (see Appendix D) is AC.

119.CRMENAD.ASM - LEVEL 4.

This module is called from MENU. It calculates and stores the code numbers and character strings for each pushbutton when the status value of MST (see Appendix D) is AD, or BØ.

12Ø.CRMENAF.ASM - LEVEL 4.

This module is called from MENU. It calculates and stores the code numbers and character strings for each pushbutton when the status value of MST (see Appendix D) is AF.

121.CRMENFIN.ASM - LEVEL 4.

This module is called from MENU. It calculates the menu for the case where the channel is closed and there is a train enquiry, overriding any menu calculation from the above modules.

122.DISPCLR.ASM - LEVEL 4.

This module is called from DISPLAY. It receives information as to which train is the subject of the successful execute, and by referring to the Display Responsibility buffer it knows which colours, optics, and arrows have been caused by that train. It then clears all the colours, optics, and arrows for which the train was previously responsible. If the channel is not open then DISPCLR clears ALL the blocking crosses from the display (they are re-calculated and re-issued to the colour monitor in CRDISPAA).

123.CRDISP1.ASM - LEVEL 4.

This module is called by DISPLAY. It calculates and outputs to the colour monitor for a train which occupies a position in a station, i.e. if the status reached by the train during the transaction was Ø7,ØB, or 14.

124.CRDISP2.ASM - LEVEL 4.

This module is called by DISPLAY. It calculates and outputs the colour monitor for a train which is still in the origin but has a proceed authority in the UP direction, i.e. if the status reached by the train during the transaction was ØB, or 14 and the direction is UP.

125.CRDISP3.ASM - LEVEL 4.

This module is called by DISPLAY. It calculates and outputs the colour monitor for a train which is still in the origin but has a proceed authority in the DOWN direction, i.e. if the status reached by the train during the transaction was 0B, or 14 and the direction is DOWN.

126.CRDISP4.ASM - LEVEL 4.

This module is called by DISPLAY. It calculates and outputs the colour monitor for a train which has status 1A and is proceeding in the UP direction.

127.CRDISP5.ASM - LEVEL 4.

This module is called by DISPLAY. It calculates and outputs the colour monitor for a train which has status 1A and is proceeding in the DOWN direction.

128.CRDISP6.ASM - LEVEL 4.

This module is called by DISPLAY. It calculates and outputs the colour monitor for a train which has status 20 and is proceeding in the UP direction.

129.CRDISP7.ASM - LEVEL 4.

This module is called by DISPLAY. It calculates and outputs the colour monitor for a train which has status 20 and is proceeding in the DOWN direction.

130.CRDISP8.ASM - LEVEL 4.

This module is called by DISPLAY. It calculates and outputs the colour monitor for a train which has status 26 or 2F and is proceeding in the UP direction.

131.CRDISP9.ASM - LEVEL 4.

This module is called by DISPLAY. It calculates and outputs the colour monitor for a train which has status 26 or 2F and is proceeding in the DOWN direction.

132.CRDISPA.ASM - LEVEL 4.

This module is called by DISPLAY. It calculates and outputs the colour monitor for a train which has status 40 (shunting) and is responsible for the positions within the station.

133.CRDISPB.ASM - LEVEL 4.

This module is called by DISPLAY. It calculates and outputs the colour monitor for a train which has status 40 (shunting) and is responsible for the down end section if the train has a 'down end shunt', or a 'both sides shunt'.

134.CRDISPC.ASM - LEVEL 4.

This module is called by DISPLAY. It calculates and outputs the colour monitor for a train which has status 40 (shunting) and is responsible for the up end section if the train has an 'up end shunt', or a 'both sides shunt'.

135.CRDISPAA.ASM - LEVEL 4.

This module is called by DISPLAY. It calculates and outputs the colour monitor when the status is AA, i.e. a secondary transaction. The module outputs ALL the blocking crosses from the blocking cross list.

136.PROC1.ASM - LEVEL 5.

This is a special procedure which is called whenever necessary to write a character string to the monochrome monitor. The procedure defined in this module is called CHWRITE.

137.PROC2.ASM - LEVEL 5.

This is a special procedure which is called wnere necessary. Its function is to insert the variable part of the information which is embedded in the verbal field, and it does this on the basis that it reads the characters for the verbal message from fixed memory until it encounters a # followed by a code number. The code number tells it whether to insert a train number, station name, position name, etc.

138.BINASC.ASM - LEVEL 5.

This is a special procedure which is called where necessary (from SERIAL when it requires to print numbers on the printer). Its function is to translate a binary coded number into the ASCII representation of the number.

Appendix C: Maj ata Structures.

This Appendix gives a brief description of two of the main data structures used in the software for the Operator's interface: the Trains List and the Stations List.

1. The Trains List.

This is stored in a table which is located by the label BUF722. The module CRTC722X is the only module which is allowed to write to the trains list. The table is organised as follows:

```
BUF722:      |
             |
             |
             |       Block of 30 bytes for the zeroth
             |       entry in the list.
             |
             |
             |----------------------------------------
             |
             |
             |
             |       Block of 30 bytes for the first
             |       entry in the list.
             |
             |
             |----------------------------------------
             |
             |
             |
             |       Block of 30 bytes for the second
             |       entry in the list.
             |
             |
             |----------------------------------------
             |
             |             ETC.
             |
```

The trains list allows for 21 blocks of 30 bytes. The zeroth block is not used. Each block contains all the information pertinent to a train, therefore allowing a maximum of 20 trains in the system at any time.

A block in the trains list is called a BERTH. Each time a radio channel is opened with a new train the trains list is searched from the bottom up, and the new train is assigned the first free berth. When a train leaves C.R.T.C. the berth which was occupied by that train is cleared.

The information in each berth is organised as follows:

BYTE NUMBER:	NAME:	COMMENT.
00	OCC	Boolean variable indicating whether the berth is occupied or not.
01	RN	Radio number of the train.
02	MST	Status of the train after the last successful execute (see Appendix D).
03	-	Not used.
04	-	Not used.

BYTE NUMBER:	NAME:	COMMENT.
Ø5	DIR	Boolean variable indicating the direction of the train - up or down.
Ø6	LOC	Station number of the station at which the train is situated (if the train is stationary).
Ø7	POS	Position number which the train is occupying (if the train is stationary).
Ø8	DL	Destination station number (for a proceed train).
Ø9	DP	Destination position number (for a train which has been given entry authority).
1Ø	FL	The station number immediately in front of a proceeding train if the train is in a section.
11	BL	The station number immediately behind a proceeding train if the train is in a section.

BYTE NUMBER:	NAME:	COMMENT.
12	SL	The source station of a proceeding train i.e. the station number from which the train was given the original proceed authority.
13,14	AD	A 16 bit binary number giving the address of the train-borne equipment.
15,16, 17,18, 19,20, 21.	TN	A 7 character string allowing for 6 train number digits (ASCII) and a terminator ($).
22	END	The 'end' of the station at which the train is allowed to shunt. 01H=UP END, -01H=DOWN END, 55H=BOTH ENDS.
23	STP	A flag indicating whether or not the train has been STOPPED.
24	STPLC	The number of the station at which the train is stopped.

BYTE NUMBER:	NAME:	COMMENT.
25	STPS	The number of the position at which the train is stopped.

Data stored in the trains list is only updated when a transaction has been successfully executed. When a channel is opened with a train the information from the train's berth is extracted by the module CRTC731X and stored under a list of variables with subscript 731 (i.e. MST731, LOC731, etc.). This data is then stored by CRTC751X in a Temporary Database with the subscripts 751 (i.e. MST751, LOC751 etc.). As the operator selects pushbuttons etc. which change the values of, for example MST, only the temporary values are changed. After the successful execution of a transaction the new values are inserted into the trains list proper by the module CRTC722X.

1. The Stations List.

This is stored in a table which is located by the label BUF723. The module CRTC723X is the only module which is allowed to write to the stations list. The table is organised as follows:

```
BUF7_3:           |
                  |
                  |
                  |       Block of 20 bytes for the zeroth
                  |       entry in the list.
                  |
                  |
                  |----------------------------------------
                  |
                  |
                  |
                  |       Block of 20 bytes for the first
                  |       entry in the list.
                  |
                  |
                  |----------------------------------------
                  |
                  |
                  |
                  |       Block of 20 bytes for the second
                  |       entry in the list.
                  |
                  |
                  |----------------------------------------
                  |
                  |              ETC.
                  |
```

The stations list allows for 1 block of 20 bytes for each station in the network plus one extra block of 20 bytes for the zeroth block which is not used. Each block contains all the information pertaining to a station (the first block is for the first, or down end station etc.).

The information in each block of the stations list is organised as follows:

BYTE NUMBER:	NAME:	COMMENT.
00	UTAL	If this value is non zero then the station constitutes the Authority Limit for an UP train. The value of UTAL is the berth number of the train which is responsible.
01	DTAL	If this value is non zero then the station constitutes the Authority Limit for a DOWN train. The value of DTAL is the berth number of the train which is responsible.

BYTE NUMBER:	NAME:	COMMENT.
Ø2	UTT	If this value is non zero then the station constitutes a through station for an UP train. The value of UTT is the berth number of the train which is responsible.
Ø3	DTT	If this value is non zero then the station constitutes a through station for a DOWN train. The value of DTT is the berth number of the train which is responsible.
Ø4	RP1	If position 1 is not clear and does exist then this variable stores the berth number of the responsible train.
Ø5	RP2	If position 2 is not clear and does exist then this variable stores the berth number of the responsible train.

BYTE NUMBER:	NAME:	COMMENT.
06	RP3	If position 3 is not clear and does exist then this variable stores the berth number of the responsible train.
07	RP4	If position 4 is not clear and does exist then this variable stores the berth number of the responsible train.
08	RP5	If position 5 is not clear and does exist then this variable stores the berth number of the responsible train.
09	P1	This variable stores a code number giving position 1 a 'status' as described below.
10	P2	This variable stores a code number giving position 2 a 'status' as described below.
11	P3	This variable stores a code number giving position 3 a 'status' as described below.

BYTE NUMBER:	NAME:	COMMENT.
12	P4	This variable stores a code number giving position 4 a 'status' as described below.
13	P5	This variable stores a code number giving position 5 a 'status' as described below.

POSITION STATUS CODING.

Value:	Constant Name:	Description:
00	NON	Position does not exist.
01	FULL	Position is occupied.
02	BKD	Position has been given to an entering train.
03	CLR	Position is clear, or free.

The stations list is only updated when a transaction has been successfully executed. The module CRTC731X is the only module which is allowed to modify the stations list. When a transaction has been executed, the module CRTC731X makes a note of the berth number of the train with which the transaction was executed. It then clears all the entries in the stations list which were caused by that train, i.e. which have the berth number as responsibility. The module then inserts information into the stations list according to the new status of the train.

Appendix D: Operator's Interface Software -

General Status Description.

The Variable named INST ('Instruction Status') reflects the overall status of the system in the following manner:

VALUE:	MEANING.
1	Channel Closed.
2	Channel Opening (i.e. the operator has pressed 'O').
3	Channel Open (i.e. the operator has selected a radio number).
4	Executing a Primary Transaction.
5	Executing a Secondary Transaction.

A second 'status' variable is defined in the subroutine CRTC, this variable is called MSTXXX (where XXX refers to the module number which is responsible for writing to the variable i.e. MST751 is written by CRTC751 etc.). The values of MST reflect more closely the status of the system in the following manner:

Uninitialised Train.

Ø1 Newly opened channel, no Pbs have been pushed yet.

Ø2 1 or more Train Number Digits have been pressed.

Ø3 6 TN Digits have been pressed, or the operator has pressed 'ENTER'.

Ø4 Location name has been pressed.

Ø5 Position name has been pressed.

Ø6 'EXECUTE' has been pressed for the train initialisation.

Initialised Train with No Authority.

07	Newly opened channel, no Pbs have been pushed yet.
08	'PROCEED' has been pressed.
09	Destination Station has been selected.
0A	'EXECUTE' has been pressed for the proceed issuing.
35	'EXIT CRTC' has been pressed.
36	'EXECUTE' has been pressed for the exit.
37	'SHUNT' has been pressed.
38	Shunt side has been selected.
39	'EXECUTE' has been pressed for the shunt issuing.
60	'CHANGE TRAIN NUMBER' has been pressed.
61	1 or more TN digit has been entered.
62	6 TN Digits have been pressed, or the operator has pressed 'ENTER'.
63	'EXECUTE' has been pressed for the change train number transaction.

Train is in the Origin, has a Proceed Authority, and the next Station is the Authority Limit.

0B Newly opened channel, no Pbs have been pushed yet.

0C 'EXTEND AUTHORITY' has been pressed.

0D New destination location has been entered.

0E 'EXECUTE' has been pressed for the extend transaction.

0F 'CLEAR' has been pressed.

10 Entry position has been selected.

11 'EXECUTE' has been pressed for the entry transaction.

12 'NO ENTRY' has been pressed.

13 'EXECUTE' has been pressed for the no entry transaction.

45 'REQUEST STOP' has been pressed.

46 Stop location has been selected.

47 Stop position has been selected.

48 'EXECUTE' has been pressed for the stop transaction.

55 'CANCEL PROCEED' has been pressed.

56 'EXECUTE' has been pressed for the cancel transaction.

Train is in the Origin, has a Proceed Authority, and the next Station is a through Station.

14 Newly opened channel, no Pbs have been pushed yet.

15 'EXTEND AUTHORITY' has been pressed.

16 New destination location has been entered.

17 'EXECUTE' has been pressed for the extend transaction.

18 'CLEAR' has been pressed.

19 'EXECUTE' has been pressed for the clear transaction.

49 'REQUEST ST[illegible]' has been pressed.

4A Stop loca[illegible] been selected.

4B Stop positi[illegible]as been selected.

4C 'EXECUTE' has been pressed for the stop transaction.

57 'CANCEL PROCEED' has been pressed.

58 'EXECUTE' has been pressed for the cancel transaction.

Train is in the Origin and has a Shunt Authority.

40 Newly opened channel, no Pbs have been pushed yet.

41 'END SHUNT' has been pressed.

42 New position has been selected.

43 'EXECUTE' has been pressed for the end shunt transaction.

Train is approaching the Authority Limit, entry has been given.

1A Newly opened channel, no Pbs have been pushed yet.

1E 'COMPLETE' has been pushed.

1F 'EXECUTE' has been pushed for the complete transaction.

Train has left the Origin, and the next station is the penultimate one.

26 Newly opened channel, no Pbs have been pushed yet.

27 'EXTEND AUTHORITY' has been pressed.

28 New destination location has been entered.

29 'EXECUTE' has been pressed for the extend transaction.

2A 'CLEAR' has been pressed.

2B Entry position has been selected.

2C 'EXECUTE' has been pressed for the entry transaction.

2D 'NO ENTRY' has been pressed.

2E 'EXECUTE' has been pressed for the no entry transaction.

4D 'REQUEST STOP' has been pressed.

4E Stop location has been selected.

4F Stop position has been selected.

50 'EXECUTE' has been pressed for the stop transaction.

59 'CANCEL PROCEED' has been pressed.

5A 'EXECUTE' has been pressed for the cancel transaction.

Train has left the Origin, and the next Station is a through Station.

2F Newly opened channel, no Pbs have been pushed yet.

30 'EXTEND AUTHORITY' has been pressed.

31 New destination location has been entered.

32 'EXECUTE' has been pressed for the extend transaction.

33 'CLEAR' has been pressed.

34 'EXECUTE' has been pressed for the clear transaction.

51 'REQUEST STOP' has been pressed.

52 Stop location has been selected.

53 Stop position has been selected.

54 'EXECUTE' has been pressed for the stop transaction.

5B 'CANCEL PROCEED' has been pressed.

5C 'EXECUTE' has been pressed for the cancel transaction.

The Channel is closed, and no train enquiry is active.

AA No Pbs have been pressed.

AB 'SET BLOCK X' has been pressed.

AC Block X location has been selected.

AD Block X position has been selected.

AE 'EXECUTE' has been pressed for the set block X transaction.

AF 'CLEAR BLOCK X' has been pressed.

BØ Block X number has been selected.

B1 'EXECUTE' has been pressed for the clear block X transaction.