STUDY AND IMPLEMENTATION OF LOAD MONITORING ON AN 11 kV UNDERGROUND CABLE DISTRIBUTION NETWORK

presented to

THE DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
UNIVERSITY OF CAPE TOWN

in partial fulfilment
of the requirements of the

M.Sc. (Eng.) Degree

by

BRENT GARY HAMPTON

April 1987

ACKNOWLEDGEMENTS

The writer would like to record grateful thanks to the following persons, without whose assistance this report would not have been possible.

Professor K.M. Reinek for his invaluable assistance as supervisor.

Messrs H. du Toit, N Johnson and P. Nell of Escom for their help in fomulating the subject of this report.

Messrs H. Fram and J. Davies for assistance in implementing software aspects of the project.

Mr M. Fielding of the Professional Communications Unit for assistance in structuring the report.

Alta Roussouw for her extreme patience and diligence in typing the report.

I certify that, except as noted above, the report is all my own work and all references are accurately reported.

BRENT GARY HAMPTON

SYNOPSIS

Operating and Planning staff of an electricity supply utility have expressed the opinion that a load monitoring system needed to be installed at protected substations* forming part of an 11 kV underground cable reticulation network. Load information was currently only available from feeders connected to the 132/11 kV stepdown substations which provide infeeds to the system. The staff felt that the availability of feeder load information from the protected substation level of the 11 kV ring network would assist in more efficient system planning and more accurate system operation.

This request for additional load data constituted the main motivation for the project described in this report. The two main objectives of the project can thus be stated as follows:

(i) To prove conclusively that there were benefits to be derived from implementing load monitoring at protected susbstations on the 11 kV reticulation network.

(ii) To implement a suitable load monitoring system <u>if</u> the study resulting from (i) proved this was necessary.

* Protected substation - next level of distribution point after 132/11 kV stepdown substation. No stepdown of voltage occurs at protected substations which are equipped with circuit breakers and current transformers.

Background information

The network under consideration provides power to mainly domestic and commercial consumers in the areas immediately surrounding the Cape Town municipal areas. There are three main levels of distribution associated with the 11 kV network, viz.

(i) 132 kV or 66 kV infeeds to the 11 kV network via stepdown substations. A telemetry system is installed at all stepdown substations. This system provides data on feeder loadings, busbar voltages, breaker indications and alarms via a radio data link to the control centre.

(ii) 11 kV Solkor* ring feeders which form the backbone of the reticulation system. These feeders are operated in continuous ring configuration to ensure reliability of supply. The rings loop through protected substations and normally back to the original stepdown substation busbars. The ring configuration can be changed to suit network conditions by operating breakers at the protected substations.

(iii) 11 kV Overcurrent ring+ feeders which are fed from the Solkor Rings via the busbars of the protected substations. Overcurrent rings are always operated with an open point in the ring. They service minisub rings which in turn supply commercial and domestic consumers.

The proposed load monitoring system would be installed at the protected substation level on the network. A system has already been installed at stepdown substations (higher level) while it would not be possible to install load monitoring at substations on overcurrent rings (lower level) because they are not equipped with current transformers. Overcurrent protection equipment is located only at the protected substations which feed into the overcurrent ring.

* Solkor - refers to type of cable protection used on continuous feeder rings. It is a unit protection system and operates on the principle of detection of change (outside defined limits) between power into and power out of a device.

\+ Overcurrent rings - so called because of overcurrent protection equipment installation at the protected substation where the ring originates.

Study on benefits of load monitoring

A literature search indicated that no previous study appears to have been done on the benefits of load monitoring at substations on an 11 kV cable ring network.

As a starting point a study was made on the extent to which underground cables could be overloaded without resultant damage. Cyclic loading factors were investigated as they are of importance when investigating a cable network supplying mainly domestic and commercial consumers. Conclusions reached from this study indicated that an amount of uncertainty exists with regard to the overload capability of underground cables under cyclic conditions. For this reason a decision was reached to adopt a conservative approach and to avoid situations causing overloading of the network.

The next step in the process was to do a critical analysis on "operations" aspect of the reticulation network. A typical section of the network was selected and three operating conditions considered when cable overloading could occur. These conditions were as follows:

- Peak load on a cold winter day
- Change in network configuration for maintenance purposes
- Change in network configuration because of faulty conditions

In all three cases it was shown that considerable overloading could occur. It became clear that the availability of feeder load data from the protected substation in question would help determine the optimum switching configuration which would distribute the load as evenly as possible. Overloading would thus be avoided.

An investigation into "planning" considerations centred on the determination of diversity factors as they are an important input to the load flow programme which is used to plan network extensions and reinforcements. The study proved the availability of feeder load data from protected substations would assist in more accurate determination of diversity* factors. This would in turn facilitate more accurate specification of cable sizes. Economic benefits could clearly result from this more accurate specification.

$$\text{* Diversity factor} = \frac{\text{max recorded feeder load}}{\text{sum of individual max demands of loads on feeder}}$$

Implementation of load monitoring on the network

The previous studies clearly indicated the benefits to be derived from load monitoring. It thus became necessary to implement a suitable system to meet the following requirements (determined in conjunction with the user):

(i) The system had to provide for storage of information on cassette tape or disc at a central point. This requirement thus excluded the possible use of chart recorders.

(ii) Acquisition of data had to be on a digital basis in order to enable data transmission via a radio channel to be implemented.

(iii) Continuous load monitoring was not required. Load data would only be required for limited periods ranging from one day to one week.

In order to determine whether a suitable system was available for implementation a literature search and market survey was carried out. The results of this study indicated that a system would have to be designed and implemented specifically for this application.

A system design philosophy was developed. The main points were as follows:

(i) Digital techniques would be used to acquire the data.

(ii) The system would scan a maximum of 8 analogue quantities.

(iii) An overall system accuracy of 5% would be sufficient.

(iv) The outstation would not be microprocessor based.

(v) Data would be sampled every 15 minutes.

(vi) Data would initially be transferred to the master station by means of cassette tape. Radio transmission of data would only be introduced in the future.

(vii) Final presentation of the data would be in the form of graphs and tables.

(viii) Protection of the outstation against noise and spikes would have to be provided.

Motivation for this philosophy is provided in the main body of the report.

The main constraint on the development of the system was that a total budget of R40 000 for implementation at ten outstations was available.

System design was done in a top down manner. Firstly a block diagram of the entire system was developed followed by block diagrams of the outstation and master station respectively. Finally the detailed electronics design of the system was done with an emphasis on using readily available components. The design was implemented on breadboard, veroboard and finally double sided printed circuit board. Accuracy and correctness of operation was tested at all stages of the design process.

The system software was demarcated into two distinct areas viz

(i) Data Input Programme which provided for data to be read from the cassette tape through a demodulator into the serial port of a microcomputer.

(ii) Macro-driven spreadsheet model which produced tables and graphs of load data.

The Data Input Programme was designed in a top down manner with the aid of Jackson Charts. Maximum flexibility regarding periods of data collection was allowed for, while user friendliness was a top priority. Final implementation was done using Turbopascal. The programme was tested using known samples of recorded data.

The second section of software was implemented by means of a macro-driven Lotus spreadsheet model. A considerable amount of programming was necessary to automate the production of graphs and tables. The end user thus requires almost no working knowledge of the software in order to produce tables and graphs of data. Operation of the system is on a menu driven basis.

Finally the outstation layout was designed and a prototype unit assembled. The system was then implemented with only one outstation and a radio link to trigger the data collection process. Examples of load data contained in this report were captured using this prototype unit.

Certain improvements to the prototype system have been proposed:

(i) Installation of a Selective Calling System so that outstations can be selectively triggered.

(ii) Installation of Battery Standby at the outstation.

(iii) Implementation of Data Transmission using a Radio Link.

The first two proposals will shortly be implemented. The third proposal will only be implemented in the longer term should it prove necessary.

The report finally concluded that there were definite benefits to be derived from monitoring feeder loads at protected substations in areas of high load on an underground cable 11 kV ring network. As a result of this conclusion a suitable system has been designed and implemented. This system is currently in the prototype stage. Research objectives have been met except for the implementation of a data transmission link to the control centre using radio.

TABLE OF CONTENTS

1.0 INTRODUCTION

Overview

This section provides a brief background to the study and implementation of load monitoring on an underground cable network. Motivation for the project is given and the objectives outlined. The scope and limitations of the project are discussed and finally an outline of the report is provided.

1.1 Background to the Project

This study involves the investigation and subsequent implementation of load monitoring on an 11 kV underground cable urban reticulation network. Prior to this study telemetry facilities were only available at the 132 or 66/11 kV stepdown substations on the network and not at any subsequent distribution points.

Distribution staff expressed the opinion that the availability of loading information from protected substations (next level of distribution point after stepdown substation) would assist in improving the standard of system operation as well as contribute towards optimisation of system planning.

This subjective opinion formed the basis of motivation for a full investigation into the possible benefits of load monitoring on underground cable reticulation networks. The ensuing study proved that additional load monitoring on the network would result in definite benefits.

This in turn provided motivation for the design and implementation of a suitable load monitoring system. At present field tests are being carried out on a prototype system at one particular substation. A system improvement that will enable selective scanning of outstations from the control centre will shortly be implemented.

1.2 Motivation

Distribution personnel have stated that in their opinion the availability of loading information from cable feeders would assist in optimizing current planning and operating techniques.

With regard to operating procedures they believed that cable damage could occur due to overloading occurring on the network during abnormal or peak load conditions. The operators in the control room would be unaware of such overloading taking place unless the feeders were connected to a stepdown substation where a telemetry system was already installed.

The majority of the cable reticulation system is operated in a ring configuration. It is thus possible to change feeder loadings by switching breakers, thus spreading the load more evenly. Such switching procedures cannot be optimized however, unless a facility exists which can measure feeder loadings.

With regards to planning network extensions or reinforcements of present systems, staff expressed the opinion that overspecification of cable sizes had been occurring because system diversity factors could not be determined with sufficient accuracy. It appeared likely that provision of additional load data would contribute towards more exact specification of cable sizes. This would in turn have definite economic advantages.

A thorough investigation into the reasons for the requirement of additional load data was clearly necessary.

This investigation proved conclusively that it would be beneficial to monitor cable loading at protected substations in the network.

This in turn provided the motivation for implementing load monitoring on the system.

The proposed system had to meet the following requirements which were determined in conjunction with the user:

(i) Provide for digital storage of information at the control centre. In this way the information could be accessed easily for planning and operating purposes. Clearly this requirement excluded the possible use of chart recorders which would necessitate storage of cumbersome paperwork.

(ii) The system did not have to monitor loads continually at protected substations. For this reason the system had to provide the facility of selective remote triggering of outstations.

(iii) Acquisition of data had to be on a digital basis to enable transmission over a radio link to take place.

The above requirements have been included as they are the main reasons for adopting the particular design philosophy which is explained in Section 6.0.

1.3 Objectives of the Project

The main objective of this project comprises two aspects, viz.

(1) To prove that the provision of feeder load data from protected substations had definite benefits with regard to planning and operating the network.

(2) If (i) above is proved correct, to implement a suitable load monitoring system on the network.

The main objective in turn gave rise to certain secondary objectives. With regard to (1) above these were as follows:

(i) To determine the overload capability of underground cable.

(ii) To assess the effect of cyclic loading on cables.

(iii) To critically evaluate operating procedures currently in use.

(iv) To critically evaluate planning procedures currently in use particularly with regard to diversity factors.

(v) To determine whether additional network loading information would be of use in optimizing current planning and operating techniques and if so, to specify the information in detail.

With regard to (2) above the secondary objectives were as follows:

(i) To determine whether a data acquisition system suitable for this specifc application has already been designed.

(ii) If a suitable system does not already exist to design and implement a system with the following objectives in mind:

- to design the system with sufficient flexibility to be used at any location on reticulation system.
- to design the systems to be user friendly.
- to allow for ease of maintenance.
- to provide for digital storage of information at a central point.
- to provide well laid out and clear information to the user.
- to allow for data to be transmitted from outstations to the control centre on a communications link.

1.4 Scope and Limitations

The project involved on examination of the need for load monitoring at distribution point on an urban underground cable network. An investigation concerning overload capabilities of cables and effects of cyclic loading was done in order to determine possibilities of cable damage during worst case conditions.

The report focuses on shortfalls in operating and planning techniques currently in use and proves that provision of additional information would assist in alleviating these shortfalls.

Interviews with personnel and literature reviews were used for this part of the report.

A limitation of this study is that it applies chiefly to reticulation networks which are operated in a ring configuration.

The second part of the project was concerned with the design, assembly, test and implementation of a suitable load monitoring system. Certain constraints had to be considered viz:

- A financial limit of R40 000 to develop and implement the system at ten substations.

- Requirement for storage of load data on tape or disc at central point.

- Users with very limited computer knowledge.

- Maintenance personnel with no microprocessor knowledge.

The report covers all aspects of the development and implementation process and includes full motivation on why a particular design philosophy was adopted. The final system could be used in a related data acquisition system although it does have the limitation of only being able to scan up to a maximum of eight analogue quantities. In addition it could not be used to acquire large volumes of data because of fairly limited storage capability.

1.4 Outline of the Report

Section two provides background information on the underground cable reticulation network which forms the basis of the study.

This covers inter alia geographical details, configuration of the network and control of network operations. In particular different levels of distribution on the network as well as Solkor rings are discussed in detail.

Section three contains an outline of the procedure followed during the project.

Section four reviews the issue of possible damage occurring because of overloading. A critical analysis is done of current operating and planning procedures. The chapter concludes that a definite requirement exists for load monitoring at protected substations in areas of high load. The requirements are outlined for a load monitoring system which will provide the necessary data.

Section five contains both a literature review as well as a market survey on systems likely to meet the requirements. It concludes that no suitable system that will meet the requirements is available and recommends the development of a custom designed system.

Section six outlines the design philosophy of the system. Factors which influenced the design are discussed and motivation is provided for configuring the system in a particular manner.

Section seven provides a top down description of the system design. Block diagrams of the overall system, master stations and outstations are discussed. The detailed electronic design of the hardware is provided and a description of testing procedures given.

Section eight provides a description of the design approach adopted for the data input software. A brief programme specification is outlined followed by a conceptual overview. System constraints and requirements are discussed. The design is explained with the aid of Jackson charts followed by a discussion of validation procedures.

Section nine describes the development of a macro-driven spreadsheet model. The purpose of this software is to display data in a clear and well laid out manner to the user. Reasons for developing a highly automated system are outlined.

Section ten describes the implementation of the system and field tests. The layout design of the outstation is described and field testing procedures are discussed.

Section eleven outlines three possible improvements that could be made to the system.

Section twelve concludes on the need for implementing load monitoring on underground cable reticulation networks. In addition the extent of fulfilment of research objectives is evaluated.

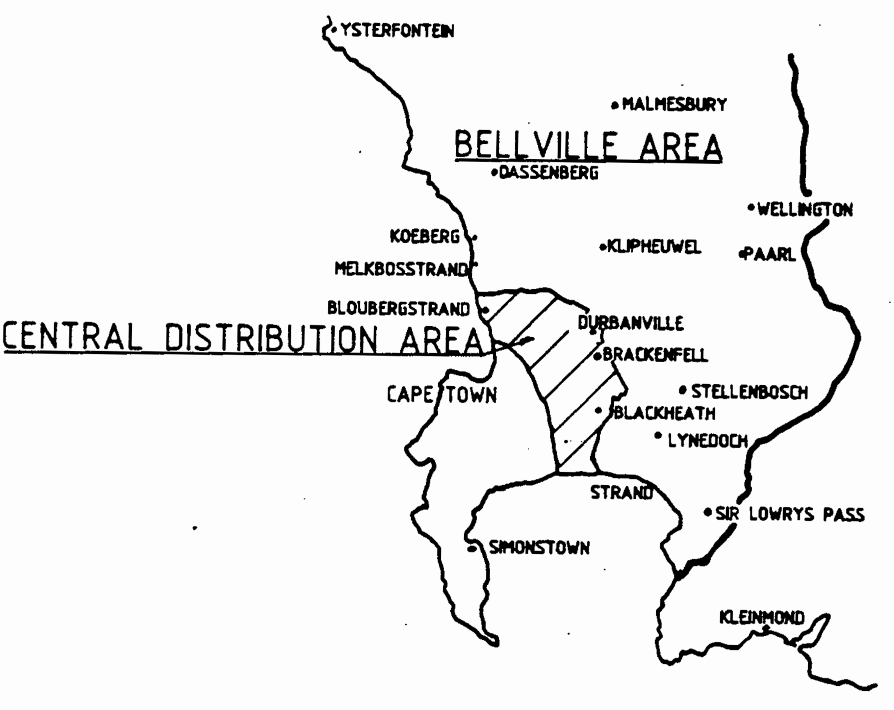
2.0 BACKGROUND INFORMATION

Overview

This chapter is intended to provide the reader with relevant information on the Central Distribution Area of Escom. This information should assist in providing an understanding of the reasons for developing the system discussed later in the report.

2.1 Geographical details

The Electricity Supply Commission provides power to its customers through a total of 12 regions. The section of network under discussion (Central Distribution Area) forms part of the Western Cape Region. The map shown in fig. 1 shows the geographical boundaries of the area.



GEOGRAPHICAL BOUNDARIES OF CENTRAL DISTRIBUTION AREA

Fig. 1

The geographical area covered by Central Distribution Area is only 2 225 km^2 compared with 57 025 km^2 for the Region. The density of consumers, however, is considerably greater than that of the rural areas.

2.2 Function of Central Area

Escom provides supply to all individual domestic and commercial customers as well as the usual industrial users. The area is chiefly urban and hence has a majority of domestic customers. Central Area is unique within Escom as it is the only section which reticulates supply at 380 V directly into households.

2.3 Configuration of the Network

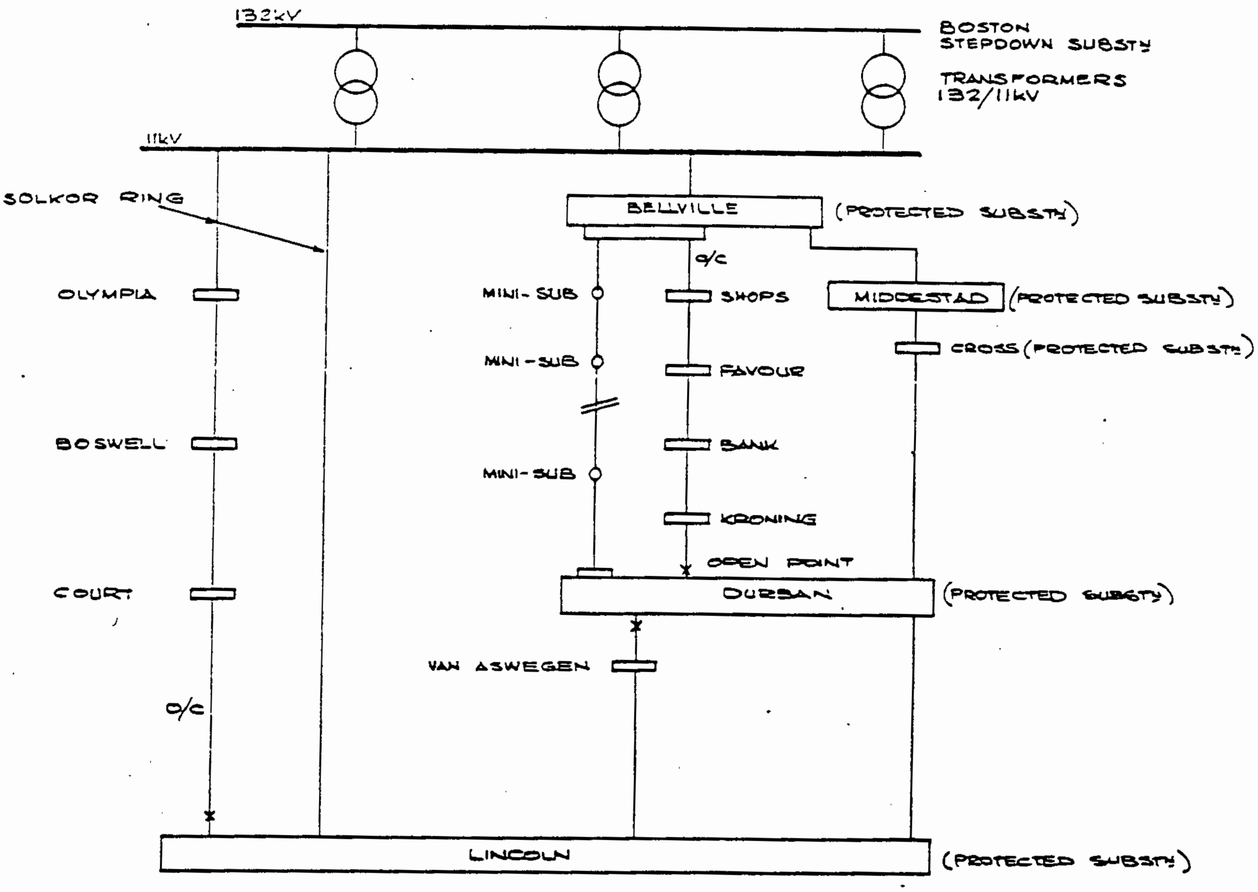
An example of a section of the 11 kV reticulation network is shown in fig. 2.

2.3.1 Stepdown Substations

There are a total of 18 stepdown substations in the area. The main purpose of these substations is to step the voltage from 132 kV or 66 kV down to 11 kV which is the voltage at which the reticulation network is operated. Referring to fig. 2 Boston is the stepdown substation in this particular example.

The majority of the reticulation network uses underground cables although a small length of overhead lines is also used.

An Intrac outstation is installed at each stepdown substation for telemetry purposes. This system will be discussed briefly at a later stage but is completely independent of the monitoring system discussed in the main body of this report.



REPRESENTATIVE SECTION OF 11 KV RETICULATION NETWORK

Fig. 2

2.3.2 Continuous "Solkor" Rings

Protected substations are connected together by continuous ring feeds which normally loop back to the same stepdown substation busbars. The feeders forming part of these continuous rings are protected by Solkor protection and are therefore referred to as Solkor rings. Solkor protection only provides protection against feeder faults. The majority of Solkor circuits have no overcurrent protection. This is an important fact in considering the requirement for a load monitoring system at protected substations.

Appendix 1 indicates the advantages of using continuous ring feeds. This explanation is based on the theory proposed by Cotton (1). The protected substations ("brickbuilts") act as distribution points for overcurrent ring feeders. No stepdown of voltage occurs at these substations. 185mm^2 3 core cable is used on all Solkor rings in the reticulation network.

The network is configured so that in the event of a Solkor ring feeder being isolated, other breakers can be closed in the system and the ring will be completed via another path to a stepdown substation.

2.3.3 Overcurrent rings

These rings are protected by overcurrent and earth fault protection and are fed from protected substations. 95mm^2 3 core cable is used on these rings which loop through either ring main units or extensible switchgear. Extensible switchgear provides bulk supply (11 kV) to factories while ring main units feed onto minisub rings which in turn transform the voltage to 380 V for reticulation to individual houses.

Overcurrent rings and minisub rings are always operated with an open point. The position of this open point can be altered for maintenance or fault isolation purposes but this can change relative loadings on the protected substations.

This information has been provided to illustrate the various levels of 11 kV reticulation and the philosophy adopted for operating the network. This has a direct bearing on the need for load information at specific points in the network as will be shown in section 4.0.

2.4 Relevant Functional Areas

There are two areas of responsibility of the Distribution Area Manager, viz Line Function and Staff Function.

(i) Line Function - The staff are responsible directly for the operation and maintenance of the electrical network. The area is divided into two districts. In each of these districts there are a number of depots strategically positioned to ensure continuity of supply to the customer.

(ii) Staff Function - The staff in this function are responsible for the following technical functions:

- Network Planning
- Network Operation

Section 4.0 outlines in detail how the implementation of a Load Monitoring System at certain Protected Substations will assist in improving efficiency in the above 2 functional areas.

2.5 Control of the Network

Operating procedures on the entire 11 kV reticulation network are controlled from the "Urban Control Centre" situated in Bellville. No remote operating is done from this centre. Operators in the field perform this function although they are under the absolute authority of the control centre staff.

A telemetry system (Motorola Intrac) has been implemented to provide analogue, alarm and breaker return information on stepdown substations to control centre staff. (Details of the Motorola Intrac System are contained in Appendix 9). Information concerning loading of feeders at the protected substation level is not available except from meter readings taken annually during peak periods.

3.0 PROJECT PROCEDURE

Overview

This section has been included to illustrate the approach followed during the execution of the project.

3.1 Setting Project Objectives

Distribution personnel were of the opinion that network operating and planning techniques currently in use were inadequate because of a lack of loading data from protected substations. The objectives of the project were then established viz

- To critically evaluate planning and operating techniques.
- To develop/purchase a load monitoring system for protected substations should this prove necessary.

3.2 Investigation into current techniques used for network planning and operations

An investigation was initiated to critically evaluate current network planning and operating techniques. A comprehensive literature survey was done to determine the parameters of interest. Distribution staff were consulted concerning current practice.

The results of the investigation showed that if load data could be obtained from protected substations in areas of high load density, current planning and operating techniques could be improved. It became clear that there was a need for a Load Monitoring System to be installed at selected protected substations.

3.3 Investigation of currently available systems

A literature search was done to establish whether a suitable system for this application had already been developed. A market survey was carried out for the same purpose. Several companies including Telkor, Motorola, Hewlett Packard and Servitek were approached. It soon became clear that systems had been developed for related applications. Use of these systems in this particular load monitoring application would be uneconomical because of the provision of unnecessary features and general unsuitability. The conclusion of this investigation was that a suitable system would have to be designed and implemented.

3.4 Establishment of Design Philosophy and System Configuration

The main objective was to develop and implement a low cost reliable load monitoring system to monitor feeder loadings at protected substations. A budgetary limit for the project was established in conjunction with Distribution personnel. A suitable system configuration was then established based on environmental factors and system constraints. A dedicated hardwired approach was chosen for the outstation design while data transferral between outstation and master station would be done using cassette tape. An IBM compatible microcomputer would be used to process the data. A radio link to transfer data from outstations to the Urban Control Centre could be implemented at a later stage.

3.5 Design of the Outstation

Firstly a block diagram of the outstation board was developed. Various devices were then selected and the detailed design was developed. A suitable system clock rate was selected and system timing diagrams were produced.

After completion of this stage of the design a test circuit was built up using Veroboard. A power supply of +12 V, +5 V and -5 V was designed and built. This supply was used throughout the development process. The design process continued and various modifications were made to the design as testing of the veroboard circuit proceeded.

As the veroboard system is by nature inconvenient and clumsy it was decided to design a prototype Printed Circuit Board. This was done using a PC development software package on the Zenith Computer. This method proved to be an efficient alternative to the normal method of PC Board design. The PC Board circuit was assembled and detailed testing continued. Further circuit design modifications were done to ensure correct operation of the board. Various analogue inputs were connected to the board and in this way operation of the various functions of the circuit were checked. Batches of test data were stored in FSK form on a tape ready for test use at the master station.

3.6 Design of Master Station Hardware

At this stage an FSK demodulator was designed and built for use at the master station end. A circuit for conversion of the data stream to RS 232 mode was designed and built. The output of the circuit was connected to a 24 pin connector ready to input to the serial port of the Zenith computer. The operation of this part of the hardware was tested using the sample data recorded on the tape.

3.7 Design of the Master Station Software

3.7.1 Software to read data from serial port

Pascal was used to develop this section of the software. The purpose of the software is to read the data from the serial port, transform it into a form suitable for use in a Lotus 1-2-3 worksheet and then store the transformed data on a floppy disc. The software was tested using taped sample data.

3.7.2 Lotus Model

At this stage a macro driven Lotus model was developed to process the data into a form suitable for producing graphs and reports. A master worksheet was developed and tested using sample data stored on disc by the software.

3.8 Design of the Outstation Layout

A Class B meter box was selected as a suitable housing for the prototype outstation board. Various modifications were made to the box in order to house the board, power supply, radio receiver, tag blocks and cabling.

3.9 Field Testing

A second PC Board incorporating the modifications made during the development phase was produced. The components were soldered onto the board and testing was carried out. The board was mounted in the outstation housing together with the power supply. The UHF radio receiver was also installed and the necessary tag block connections made.

The outstation was installed at Burger substation and the necessary power, transducer and antenna connections were made. At the master station (Urban Control Centre) the UHF transmitter and timer was installed. The system was ready for operation and the field test commenced. The tests had been in progress for approximately one month when this report was produced.

4.0 INVESTIGATION INTO POSSIBLE BENEFITS OF LOAD MONITORING

Overview

This section verifies the opinion that definite benefits are to be derived from implementing load monitoring on the networks. A literature survey revealed that no previous investigation of this nature appears to have been done. Theory on overload capability of cable systems is outlined and conclusions reached. A critical evaluation is made of current planning and operating techniques.

4.1 Theory of High Voltage Cable Operation

As discussed in section 2.0 Escom uses underground cable almost exclusively for 11 kV reticulation in the Central Distribution Area. The current carrying capacity of cables is very restricted because of the difficulty of dissipating the heat produced. In many cases several cables per circuit are needed plus larger conductor cross sections in order that the same power be transmitted as by an overhead line at the same voltage (2). Unfortunately the electrical requirements for cables result in the use of insulation the resistance of which is very good thermally as well as electrically. In addition in this application the cables are buried directly in the ground which is also of high thermal resistivity.

4.1.1 Calculation of the Temperature Rise and Load Capability of Cable Systems

Considerable research has been carried out on the subject of Load Capability of Cable Systems. For 25 years since 1932 the work of D.M. Simmons (3) was the basis of calculations on load capability. Neher and McGrath (4) presented an updated version of this research in 1957. More recently research carried out by Katz (5) indicates that thermal limits prescribed in the past have been too conservative particularly in the case of cross linked polyethylene (XLPE) insulated cables. At present the IEC standard 287 on "Calculation of the continuous current rating of cables" (1974) appears to be the basis for manufacturer's ratings. A summary of the relevant formulae used for calculation is given below.

The reticulation system under discussion uses 3 core paper insulated cable with a lead sheath and steel armouring. Only formulae relevant to this particular configuration will be discussed.

The permissable current rating of a cable can be derived from the expression for the temperature rise above ambient:

$$\Delta\theta = (I^2R + \tfrac{1}{2}Wd)\,T_1 + [I^2R\,(1 + \lambda_1) + Wd]\,nT_2 + [I^2R\,(1 + \lambda_1 + \lambda_2) + Wd]\,n\,(T_3 + T_4) \qquad (1)$$

where:

I =	current flowing in one conductor	A
$\Delta\theta$ =	conductor temperature rise above the ambient temperature	deg C
R =	alternating current resistance per unit length of the conductor at maximum operating temperature	ohm/cm
Wd =	dielectric loss per unit length for the insulation surrounding the conductor	W/cm
T_1 =	thermal resistance per unit length between one conductor and the sheath	deg C cm/W
T_2 =	thermal resistance per unit length of the bedding between sheath and armour	deg C cm/W
T_3 =	thermal resistance per unit length of the external serving of the cable	deg C cm/W

T_4 = thermal resistance per unit length between the cable surface and the surrounding medium, as derived from Clause 9 — deg C cm/W

n = number of load-carrying conductors in the cable (conductors of equal size and carrying the same load)

λ_1 = ratio of losses in the metal sheath to total losses in all conductors

λ_2 = ratio of losses in the armouring to total losses in all conductors

Taking into account the effect of solar radiation on a cable, the permissable current rating is given by the formula:

$$I = \frac{\Delta\theta - W_d[0.5T_1 + n(T_1 + T_3 + T_4)] - \sigma D_e H T_4}{RT_1 + nR(1+\lambda_1)T_2 + nR(1+\lambda_1+\lambda_2)(T_3+T_4)} \qquad (2)$$

where:

σ = absorption coefficient of solar radiation for the cable surface,

H = intensity of solar radiation which should be taken as 0.1 W cm-2 for most latitudes,

T_4 = external thermal resistance of the cable in free air, but protected from solar radiation.

Methods for calculating the different quantities required for the evaluation of equation (2) are included in Appendix 2.

Study of this theory indicates that permissible steady state load limits are well defined.

4.1.2 The Calculation of Cyclic Rating Factors for Cables

In practice cables are not loaded continuously. The disadvantage of the method proposed by the IEC Standard 287 is that it assumes continuous loading. Weedy (2) states that real life calculations are often based on an idealized daily load case of full load current for 8 hours and no-load for 16 hours. This results in a lower temperature rise than when permanently on full load which is allowed for in practice by the use of Cyclic Rating Factors.

This factor gives the amount that a cable can be operated above its rated full load current in cyclic loading conditions.

H. Goldenberg (6) has proposed a method for calculating cyclic loading factors. Weedy (2) states that for directly buried cables of $20mm^2$ - $200mm^2$ diameter a factor of 1,13 can be used.

4.1.3 Overloadability of Cable Systems

C. van Hoewe (7) has investigated to what extent cables can be overloaded without damages. He distinguishes two cases that can occur during an overload situation:

(a) Overload which is subsequent to a preceding load below the maximum capacity and during which the maximum conductor temperature of the cable is not exceeded (65°C for PILC (1)). This type of overload may be repeated as often as desired without damage since the lifetime of the cable depends exclusively on the conductor temperature and not on the maximum current intensity occurring.

(b) Overload which is subsequent to a preceding load which leads to a conductor temperature exceeding the maximum conductor temperature of the cable. Van Hoewe addresses the question of assessing the reduction of cable lifetime by means of measuring the level and period of a temperature rise above maximum conductor temperature. He states, however that it is uncertain which cable properties should be measured to assess reduction in lifetime.

Certain overload tests were carried out and it is concluded by van Hoewe that a cable can be loaded for several hours with a current far exceeding the rated current, without a permissable conductor temperature being exceeded provided the preceding load was below a certain limit. It is also concluded that even higher values can be used but that this will result in a reduction of cable lifespan. A quantitative assessment is however not possible because of the difficulty in evaluating the influencing parameters.

4.1.4 Conclusions reached from study of theory

The items discussed above viz

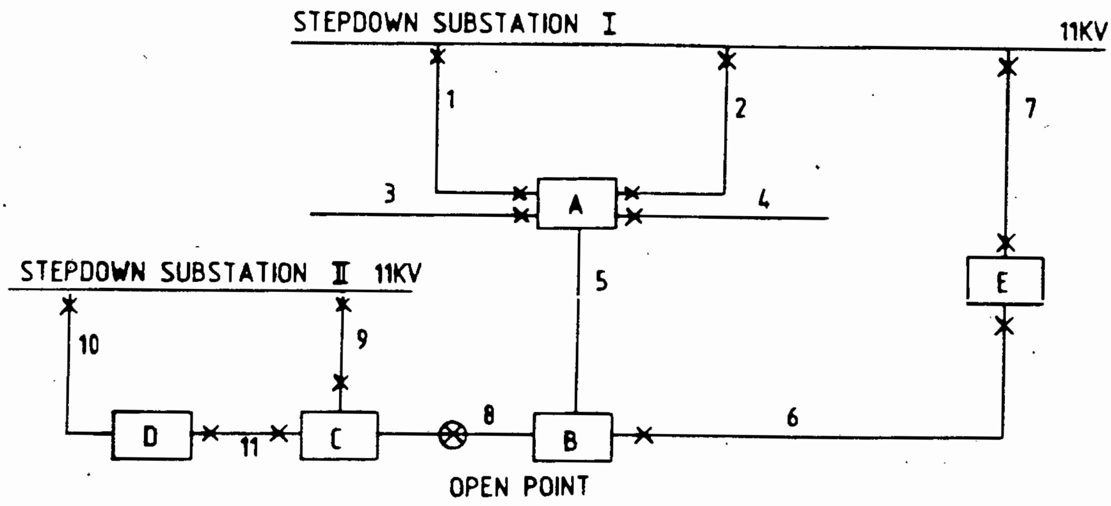
- Load capability of a cable system
- Cyclic Loading factors
- Overloadability of cable systems

have a direct bearing on the discussion of operating and planning considerations which follows. Although considerable research has been done on aspects of cable performance, a degree of uncertainty still exists particularly in the area of overloadability of cable systems. For this reason we feel it is necessary to adopt a conservative approach when operating and planning the system.

4.2 Discussion of Operating Considerations

A brief introduction to the configuration of the 11 kV reticulation network was given in section 2.0. The system is completely interconnected and integrated in order to ensure maximum availability of power to the consumer.

In order to illustrate certain shortfalls in the current approach to operating the network the simplified section of the network shown in fig. 3 is used.



SIMPLIFIED SECTION OF RETICULATION SYSTEM

Fig. 3

In fig. 3 I and II are stepdown substations while A - E are protected substations connected in Solkor ring configuration.

Network Information presently available:

An Intrac system has been installed at stepdown substations as discussed in section 2.0. The information supplied by the Intrac system which is relevant to the operation of the section of reticulation system shown in figure 3 is as follows:

(i) Busbar volts at I and II
(ii) 11 kV feeder loadings in Amps ie loadings on feeders 1, 2, 7, 9 and 10 in the figure

No information is supplied concerning loadings of feeders 3, 4, 5, 6 and 8. These feeders are connected to protected substations which have no load monitoring equipment.

Three situations will now be discussed where the lack of load information from protected substations reduces overall efficiency of system operation.

(i) Peak on a cold winter day

Consider that substation A is at a point of high consumer density and that loading is known historically (by manual reading of meters) to be approaching limits. In this situation loadings on feeders 1, 2 and 7 are known. The operator does not know what the loadings are on feeders 3, 4 and 5 connected to substation A.

Consider the situation where loadings on feeders 1 and 2 are each 3/4 of maximum cable rating. Feeders 3 and 4 (overcurrent circuits) could be drawing minimal load for instance. In this case the majority of the sum of feeder 1 and 2 loadings will be taken by feeder 5 which is considerably overloaded ($\pm$ 1,5 times maximum rating). This is an undesirable situation since damage could occur on the feeder 5 cable.

In practice this situation could easily occur since no load information is available at substation A. The provision of data for a sample of winter peak days would prevent this situation occurring, since an alternative switching arrangement could then be implemented, closing the breaker between substations B and C.

(ii) Changes in network configuration for maintenance purposes

Routine maintenance is often required on the reticulation system. Certain sections need to be switched out and other breakers closed to maintain the continuity of the Solkor ring. In figure 3 consider the situation where feeder 9 is switched out for maintenance. The open breaker between substations B & C is now closed to ensure that D and C are part of a ring. The additional load imposed by closing the breaker is known as it is the same as that of feeder 9 which is monitored by the Intrac system at stepdown substation II.

Loading on feeder 5 prior to the switchover is unknown. When the switchover occurs for maintenance purposes feeder 5 could be overloaded causing cable damage.

Provision of a load monitoring system at substation A or B would enable load readings to be taken during the day prior to maintenance to ensure that the maximum rating of feeder 5 cable would not be exceeded. If it is clear that the rating would be exceeded by a significant amount (say 20%) an alternate switching arrangement could be established.

(iii) Change in network configuration because of fault conditions

Consider the situation where a cable fault occurs on feeder 9. The Solkor protection operates isolating the feeder. The open point is closed to ensure that substation D and C form part of a ring.

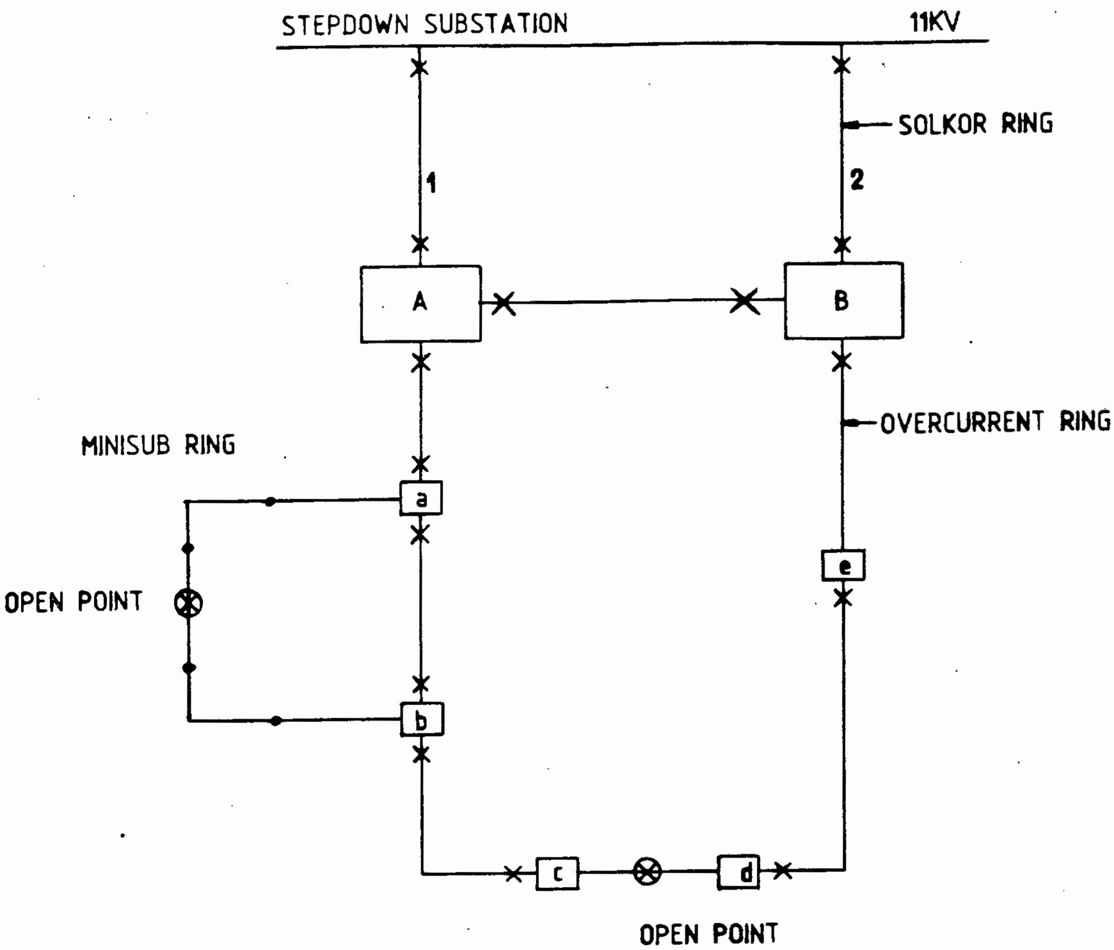
In this case it would be useful to be able to monitor feeder 5's load which could be over the permissable load limit. Provision of load data at protected substations would thus aid in drawing up operating plans for fault conditions in such a manner that cable damage will not occur.

It can be clearly seen that the provision of load data from protected substations in areas of high load will greatly increase the operator's ability to operate the system efficiently. Such a system could be off-line and need not run on a continuous basis.

4.3 Discussion of Planning Considerations

Sarikas and Thacker (8) have stated that "Since the sole purpose for the existence of an electric distribution network lies in supplying the requirements of the consumer's utilisation devices, it is fundamental that a knowledge of these load requirements is necessary for sound planning of the distribution system. The economic importance of the knowledge of load characteristics has became more and more recognized with the growth of load on utility systems".

The simplified network diagram shown in fig. 4 will serve as an aid in discussing shortfalls in the current method of planning caused by lack of load information.



SIMPLIFIED SECTION OF 11 KV NETWORK

Fig. 4

The planning department currently uses a program "Powsys" to perform fault studies and load flows on the network. The load flow aspect is of particular importance in this case. Load flows are performed on different sections of the network to check whether the cable system needs to be uprated in the future in different areas assuming a specific growth rate.

The Intrac Telemetry System provides maximum demand data for feeders 1 and 2 in figure 4. For a load study to be done on this section of the network, the network parameters are used as input to Powsys as well as the maximum demand current for feeders 1 & 2. The Powsys program compares the connected load on the ring with the input currents and scales the loads accordingly. This takes into account the overall diversity of the Solkor ring. The load flow then provides the loadings of all feeders in the circuit down to the minisub ring level.

The problem with this approach is that equal diversity is assumed for all parts of the circuit. This is probably not the case in reality. For example the load drawn by the part of the overcurrent ring supplied by substation A may be considerably greater than that drawn by the part supplied by substation B. Thus assuming equal diversity throughout the Solkor ring could easily give rise to an erroneous conclusion regarding the need to install extra cable capacity particularly on overcurrent rings. Cable capacity may need to be increased in a certain area even though this is not indicated by the load flow study.

This situation occurs because of the lack of load information from protected substations. If load information on overcurrent ring feeders were available the diversity of different parts of the circuit could be used as input to Powsys and a more realistic load flow study obtained.

4.4 Conclusions of the Investigation and Definition of the Need for a Load Monitoring System

Load information at the stepdown substation level is provided by the Intrac Telemetry System. The investigation described in preceding paragraphs shows clearly that the provision of feeder load data from

protected substations would clearly improve the network operating and planning techniques currently in use.
It should be noted that it is not possible to monitor load at minisub and extensible switchgear level as these stations are not equipped with current transformers. The Load Monitoring System would thus be installed only at protected substations.

Such a Load Monitoring System has the following requirements:

(i) The system must provide for digital storage of information at a central point. This requirement thus excludes the possible use of chart recorders.

(ii) Acquisition of data should be on a digital basis. This will enable data communication links using radio to be established between the outstations and the master station. Since the data will be used on an off-line basis, an alternative method could initially be used for data transferral should the implementation of radio data communication links prove too costly.

(iii) The system would only be installed at selected protected substations. Distribution staff have selected 10 substations which are in areas of high load where damage to cables could easily occur. It is clearly unnecessary to install the system in areas where the loads are well below the maximum rated loadings of cables.

(iv) Load data will only be required from particular substations for limited periods ranging from one day to one week. The information will be required during network conditions defined in 4.2 and 4.5. Unnecessary data would clearly result if the system were run continuously at all 10 substations.

5.0 SURVEY OF CURRENTLY AVAILABLE LOAD MONITORING SYSTEMS

Overview

This section gives an indication of load monitoring/telecontrol systems that have recently been implemented. The first two items of equipment were sourced from a literature survey while the remaining items were examined during a market survey. The equipment has been selected for discussion because of similarity to our requirements. In each case the reasons for not selecting a particular approach are outlined.

5.1 On Line Production of Planning Load Data in Substations (9)

Schroder and Kwong state that a need exists for additional load data which can be used to optimize network planning. Over a period of years they developed a load monitoring system which in its final form produced the following information.

(i) Selectable quarter/half hour summated maximum demands of power, reactive power and volt amps and the time and power factor when each occurred.

(ii) Load factors, loss load factors and summated I^2h value.

(iii) Load duration curve.

(iv) Average week-day daily load curve.

(v) Average weekend load curve.

(vi) Lowest and highest voltage recorded and total time outside the $\pm$ 5% voltage limit.

(vii) Times of all power interruptions.

(viii) A record of 5 consecutive days of substation quarter/half hour summated data.

The system is microprocessor based and carries out the data acquisitions, statistical calculations and data analysis locally in the substation on a continuous basis. There is no communications link between the outstation and control centre. Data collection personnel visit the remote site with a cassette recorder and a printer terminal. By using a "dump" push button on the front panel formatted load data can be transferred to the printer terminal and cassette tape simultaneously. The authors reduced costs of the system by using commercial grade tape recorders for data transfer and archive storage. Test results showed that overall system accuracy was better than 0,5%.

Program development was carried out using an HP 3000 computer and software development took 12 months.

This system appears to have been developed for use at major substations in the system. It provides more information than is needed at protected substations. Since VT's are not installed at these substations it would not be possible to monitor power factors and voltage limits.

The successful use of a commercial grade tape recorder for transferring data was noted. Overall costs were not mentioned but they would probably exceed the limit allowed for our particular application.

5.2 Data Acquisition in a Distribution System using the MV Transmission Network as the Transmission Path (10)

The authors have developed and implemented a data acquisition system using the cable network as the transmission path.

Unlike the overhead H.V. power lines, signal blocking devices cannot be used at cable ends because of the great number of branches in the cable network. This fact results in vagueness of line parameters such as impedance and attenuation. Tests were carried out on an 11 kV distribution network. Signals were injected using the same LC coupling device used in overhead systems. Results of the tests were positive. Details can be obtained from the reference (10).

A data acquisition system was then developed using the principle of transmission caused by an interrogation signal sent from the central station. Due to the peculiar characteristics of the cable network, a specific modem was designed. The main difference from conventional modems is the frequency range (between 80 kHz and 300 kHz) and the input and output impedances which are high and low respectively to minimize the effect of the low and irregular impedance of the M.V. network.

The data acquisition system was based on microprocessor technique and consisted of three outstations and one central station. The outstations are polled from the central station and send information about changes that have occurred since the last interrogation. The outstations can monitor up to 16 analogue and 16 digital quantities.

The results of tests on this equipment show that data transmission over the M.V. cable network is both possible and reliable. It is not exempt from problems and care has to be taken that transmission is possible whatever the network configuration.

The concept of using the cable network for transmission could be applied to the Central Distribution 11 kV Network. From the paper it appears that this work is still in the research stages. For our application the greatest problem would appear to be ensuring a transmission path for all possible network configuration. It was thus decided not to use the cable network for data transmission. If a data communication link is required at a later stage a radio link would provide a simpler solution.

5.3 Hewlett Packard Data Logger (3421A)

Hewlett Packard produces a range of Data Loggers and Data Acquisition Systems. The HP 3421A system came closest to meeting our particular requirements.

The HP 3421A must be used in conjunction with the HP 71 B terminal which acts as a controller for the data logging device. The system can scan up to 30 channels and can measure the following:

- dc volts
- ac volts
- 2 wire and 4 wire ohms
- frequency
- temperature

The system can only store up to 30 readings at a time so a printer or data storage device would be required in this application. The system is provided with battery backup and has a self test facility.

Measurement integrity is a strong feature of this system. It has a full scale accuracy of 0,01% while the integrating A/D converter ensures high noise reduction. There features are provided because this system is intended for use in an industrial environment for high accuracy measurement applications.

The HP 3421A provides digital inputs annd outputs in addition to measurement facilities. The system can also measure thermocouplers.

The HP 3421A system could be used for our application but was rejected for the following reasons.

1. Excessive cost - to equip 10 substations would cost ± R90 000.

2. Unnecessary accuracy (0,01%). This accuracy is not required for measuring feeder loadings in this application.

3. Interface circuitry would be required as the system cannot have a current input.

4. Unnecessary features - digital input and output are not required as alarms are not being monitored. Only 8 channels are required whereas the system provides 30 channels.

5.4 Telkor Antelope System

Telkor produces a range of telemetry and telecontrol equipment. The system which came closest to meeting our requirements is the Antelope telemetry system. This system is briefly described below.

The Antelope system is a point to point analogue multiplex system consisting of a transmitter, receiver and a hand held terminal or teleprinter. The transmitter is the remote unit and will scan up to 32 analogue inputs in the range -7,5 mA to +7,5 mA. The data is sent out in serial form via an FSK transmitter onto a bearer channel.

The receiver decodes the incoming FSK signal into digital data. This data represents the individual input values. Data is then produced as output at up to 32 output channels in the ~~form~~ of a forced current in the range -5 mA to +5 mA.

The microprocessor, power supply and configuration data cards used in the transmitter and receiver are identical except for the firmware resident on the microprocessor cards.

The cost of implementing a single link, ie equipping one protected substation with this system would be of the order of R15 000.

This system, although appearing to be suitable at first, was rejected for the following reasons:

1. Excessive cost - to equip 10 substations would cost $\pm$ R150 000.

2. Use of a point to point configuration - this system adds excessive hardware to the system by requiring a master station for each outstation.

3. Non availability of graph producing facility - the output of this system drives a meter.

4. No data storage facilities available on the system.

5.5 Servitek

Servitek markets a telecontrol system which could be implemented in a power network context. The system consists of a number of outstations (series 8200) installed at substations and a single microcomputer based master station.

The outstation is microprocessor based and is capable of the following:

1. 8 dual control outputs

2. 16 analogue inputs

3. 16 alarm inputs

This means that the system is capable of executing remote controls (eg. opening and closing breakers), reporting on alarm conditions (eg. battery charger fail) as well as scanning analogue inputs (eg. feeder loadings).

An FSK modem is used in the outstation. Transmission speed is selectable for 75, 110, 200, 300, 600 or 1200 Baud operation.

The master station comprises a microcomputer using IBM PC software. This machine is linked to the front-end processor using a standard RS 232 line.

Features provided by the system include the following:

1. Control of the system from the console - Real time mimic drawings of particular sections of the network can be displayed at any time.

2. Storage of analogue data - Production of graphs and tables of data.

3. Comprehensive alarm detection.

This system, although well engineered, was rejected for our application for the following reasons:

1. Excessive cost - approximately R300 000 to implement.

2. Unnecessary sophistication - only analogue information is required in this case. Control and alarm facilities are not required.

5.6 Conclusions of the Survey

From the results of both the literature search and the market survey it appears that no system has been designed to date for this particular application. Systems are available which could be used but they are not entirely suitable and provide unwanted features. For this reason it is necessary to develop a system specifically for this Load Monitoring application.

6.0 DESIGN PHILOSOPHY OF SYSTEM

Overview

This section defines the function that the Load Monitoring System is required to perform. Factors which influenced the design during the initial stage are then discussed. Reasons for configuring the system in a particular manner are explained.

6.1 Required Function of the Load Monitoring System

The investigation into optimisation of the network planning and operation functions clearly showed that there was a requirement for feeder load data from protected substations positioned in areas of high load. After the survey of available systems revealed that no suitable existing system was available the decision was taken to design and implement a custom built system. The function of the system would be to collect feeder load data at protected substations as outlined in 4.4. This load data would then have to be processed in the form of graphs and tables for analysis purposes.

6.2 Factors which influenced the design

6.2.1 Geographical Considerations

As stated in Section 2.0 Central Distribution covers a relatively small area (2 225km^2). Weather conditions are reasonably uniform in this area. It was thus considered unnecessary to include temperature sensing in the Load Monitoring Outstation. This would only be necessary if the system covered say the whole Western Cape Region where weather conditions vary widely and it would be necessary to link temperature data to a particular load curve.

6.2.2 Budgetary Constraint

The total cost of developing, installing and commissioning the system at 10 protected substations could not exceed R40 000. This figure did not include cost of a microcomputer which was already available.

6.2.3 Maintenance Requirement

The system will be maintained by personnel who are responsible for all Telecommunications equipment in the area. They have no specific uprocessor training as there is currently no other μprocessor based equipment in the area. Ease of maintenance is thus an important requirement and components must be readily available locally.

6.3 System Configuration Design and Philosophy

This section outlines certain decisions taken in respect of system configuration. The reasons for taking these decisions are explained.

6.3.1 Use of digital techniques

It was decided to use digital techniques to acquire the load data for the following reasons:

- Data can conveniently be stored in a digital form at the control centre
- A data transmission link can be implemented at a later stage if required

These requirements could not be met if a chart recorder were used for gathering data.

6.3.2 Number of data input channels required

The maximum number of feeders connected to a protected substation is 8. In most cases the number varies between 2 and 6. It is only possible to measure current at these stations as no voltage transformers exist. No temperature sensing is required as discussed in 6.2. It was thus decided to configure the system with a maximum of 8 input channels. This is also particularly convenient from a digital design viewpoint.

6.3.3 Required accuracy

The load data will be used on an off-line basis for analysis purposes. No controlling functions are involved. In the planning context load growth figures are only estimates with varying degrees of accuracy. Extreme accuracy of load data is thus not required. With regard to the operations application, the theory outlined in section 4.0 has shown that the current carrying capacity of a cable is not known to a great degree of accuracy. Hence great accuracy is not required from the protected substation load data.

For these reasons it was decided that an overall system accuracy of 5% or better would be sufficient for this application.

6.3.4 Requirement for a microprocessor based Outstation

In reaching a decision whether a microprocessor based outstation was required the following guidelines were consulted (11). A microprocessor based data gathering outstation is considered unnecessary if:

(i) The number of input channels is between 1 and 100.

(ii) The main requirement is to collect raw data for future off-line analysis.

(iii) The budget is less than R5 000 per outstation.

Based on these considerations the outstation in this application need not be microprocessor based.

Other factors were considered before reaching a decision. A microprocessor development system is not available in Escom (Western Cape Region) at this stage. Development of a microprocessor based system would thus have involved extra cost in purchasing or hiring a system. Maintenance staff would have to undergo specialized microprocessor training at high cost to be able to maintain the system.

When considering the overall function of the outstation, ie scanning and storing 8 analogue values at a prescribed instant in time, we do not feel that intelligence is necessary. If it had been decided to implement alarm monitoring, a microprocessor would have streamlined the system by reporting by exception.

For these reasons it was decided to use a dedicated hardwired approach in designing the outstation as this best fulfilled the system requirements.

6.3.5 Selection of a suitable data scanning interval

A suitable time interval was selected on the basis of theory presented in a paper by Willis, Vismor and Powell (12). This paper discusses sampling rates for load data on distribution systems. As the number of households increases, the choppiness on the load curve decreases. A sample load curve for 125 households shows a smooth curve when the load is monitored continuously. Distribution personnel have stated that the number of supply points fed from a protected substation usually exceeds 150 so little choppiness can therefore be expected on the feeder load curve. The paper states that for the measured data to be valid, sampling must be done at twice the rate of the highest frequency in the signal. For load measurements taken on sections of the network involving a large number of consumers, a valid sampling rate would be 15/30 minutes. For this reason a sampling rate of 15 minutes was chosen for the system.

6.3.6 Data Storage Capability

The investigation into the load data requirement showed that data would be required from a particular substation for a period of up to 2 days at a stretch. The data would not be required continuously but only when required by control centre and planning staff.

Considering 8 sets of feeder data scanned every 15 minutes for 48 hrs a total of 1536 bytes of memory are required. 2K bytes of memory was thus considered sufficient memory for the outstation. Use of an 8K E^2PROM for memory was considered to enable flexibility for gathering ± 1 week's data. In view of the extra cost it was decided to provide 2K memory on the outstation board but provide for flexibility in the master station software by allowing up to 3 sets of consecutive 2K blocks of data to be processed by the microcomputer. In this way one week's load data can be displayed graphically should this prove necessary.

6.3.7 Transmission of data

The load data will initially be used on an off line basis for analysis purposes. For this reason it was decided not to relay the data on a communications link back to the master station in the initial development of the system. Implementation of data communication links would involve considerable cost which would cause the budget to be exceeded. At a later stage the system could be modified to incorporate a data communications link as data is acquired digitally at the outstation.

It was decided to use a commercial grade cassette tape to transfer data from the outstation to the master station. This technique is cheap to implement and has been shown to work very successfully (10). Maintenance personnel will read the data onto cassette tape and transport the tape to the control centre.

6.3.8 Recording of Date and Time Information

Since data would be recorded using a commercial grade tape recorder, it was decided to simply record the start and end dates and times on the tape prior to recording the data. The maintenance personnel responsible for collecting the data would record the date and time information. This method has been successfully used in the system described in reference 9. Dates and times of readings would also be kept in a log book at the substation for record purposes.

6.3.9 Processing of load data

In order for analysis to be done the load data must be presented finally in the form of graphs and tables. A software package had to be obtained to read the data from the cassette tape into computer memory. Thereafter the data had to be processed to produce graphs and tables.

No suitable software could be found that would read the data from the tape through an RS 232 port into microcomputer memory and provide for the flexibility of reading in 3 sets of data consecutively with necessary validation. For this reason a program was written specifically for the requirement.

Use of the Lotus 1-2-3 package would greatly simplify processing of the data into a suitable format. It was thus decided to develop a macrodriven Lotus model for this purpose.

All software developed would have to be user-friendly and have validation checks as distribution staff have little computer knowledge. In order to make use of the Lotus package it was decided to use a Zenith microcomputer (IBM compatible).

6.3.10 Protection against spikes and noise

The outstation would be situated in a noisy environment being close to high voltage switchgear and cables. For this reason adequate protection against noise and spikes would have to be provided on the input circuits. A proven system similar to that used on the Intrac outstation was finally selected.

This section only outlines the reasons why particular approaches were adopted in the design of the overall system. Detail design points are discussed in the sections on hardware and software design.

7.0 DESIGN OF SYSTEM HARDWARE

Overview

This section provides a top down description of the system design. The block diagram of the overall system is discussed followed by explanations of the outstation and master station block diagrams. The detailed electronics design of the outstation and the master station is then provided. Finally a description of testing procedures is given.

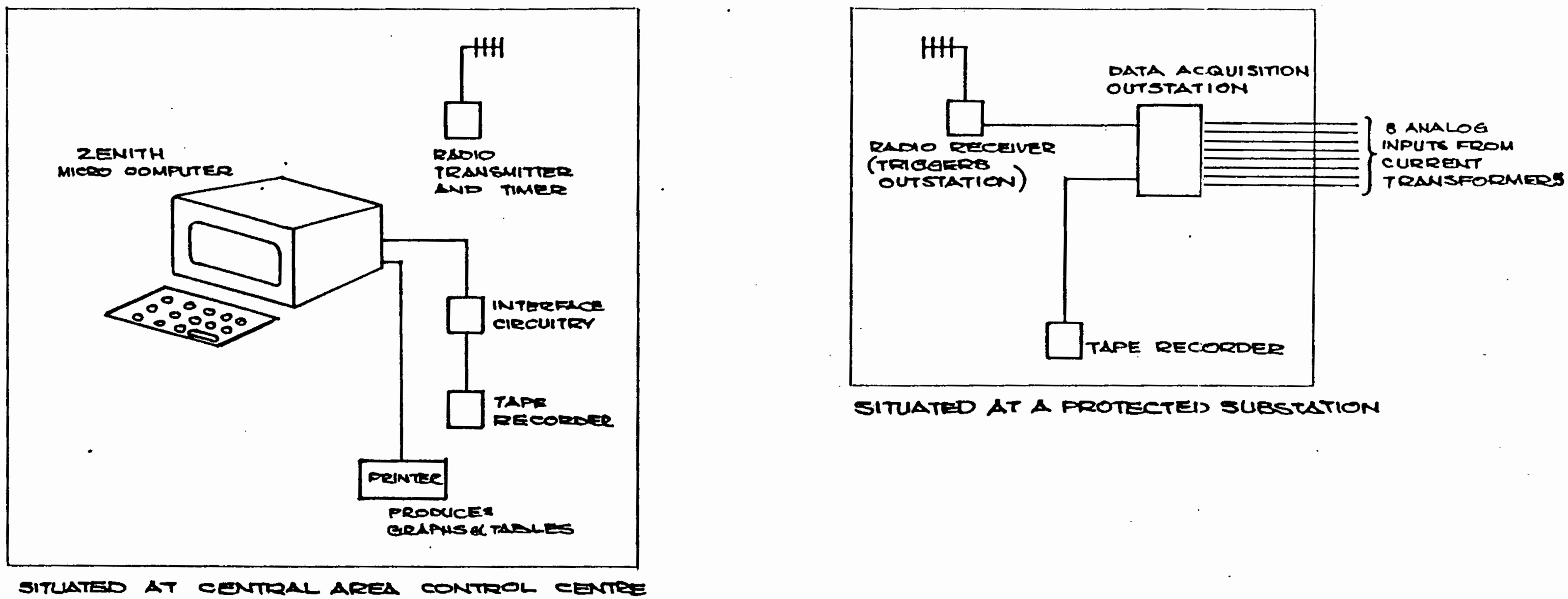
7.1 Discussion of System Block Diagram

7.1.1 Figure 5 indicates the overall system block diagram. Only one protected substation has been shown although it is intended to equip 10 of these stations with the data acquisition system.

The operation of the system is as follows:

A radio transmitter situated at the Urban Control Centre sends a short signal (unmodulated carrier) to the receiver at the protected substation. The receiver provides a pulse to the data acquisition system which proceeds to scan up to a maximum of eight analogue quantities. The system stores the values in an E^2PROM. Scanning of the analogue quantities takes place every 15 minutes (set on the timer situated at the transmitter at Urban Control Centre).

After 2 days of scanning information the E^2PROM will be filled to capacity (2K). A maintenance official will change a switch to "read" position on the system and the data will be read from the E^2PROM onto a tape via a UART. The switch is then returned to the normal position, a new tape inserted in the recorder and the system is reinitialised if further data is required. The tape containing 2 K of data is then taken to the Urban Control Centre where the data will be read into the computer which will process all necessary reports and graphs.



OVERALL BLOCK DIAGRAM OF LOAD MONITORING SYSTEM

Fig. 5

7.1.2 Discussion of the Data Acquistion Outstation Block Diagram

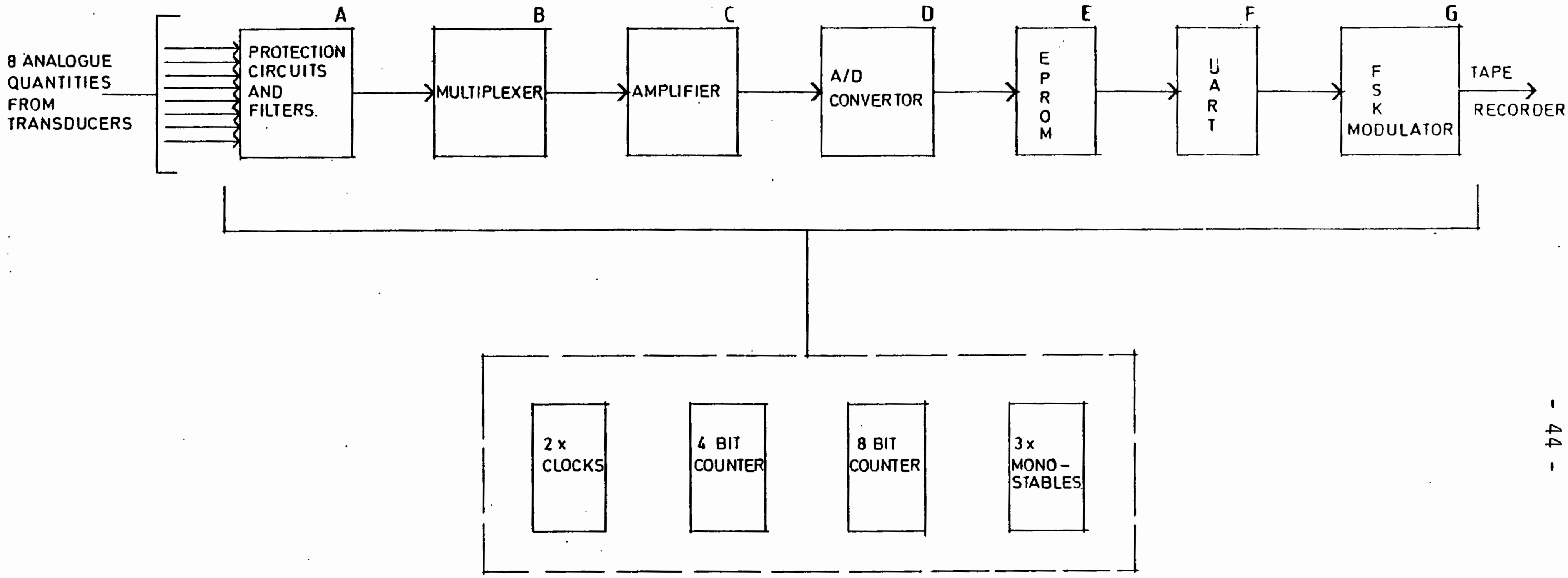
The main purpose of the outstation is to record current readings on feeders connected to the protected substation busbar. Each of these feeders is equipped with a current transformer (CT) for protection purposes. Each current transformer has an associated ratio eg. 400/5. For a loading of 400 A say on the line, the CT will give an output of 5A. 400 A is the maximum loading that should occur on the line.

The reading from the CT is also used for input to the Load Monitoring System. The current is first passed through a transducer which reduces the value to a corresponding value in mA. Please refer to Appendix 4 for a specification of the transducers used in this project. The outstation board can scan up to a maximum of 8 inputs from transducers.

Each feeder is provided with a protection and filter circuit as shown in fig. 6. The purpose of these circuits is to protect the remainder of the board from overcurrents/voltages and to change the current input value to a voltage level suitable for input to the next stage.

When the "trigger" signal is received, the system clock is enabled and the outputs of the 8 protection and filter circuits are scanned by the multiplexer. (point A in fig. 6)

The eight voltage inputs are thus multiplexed onto a single output line. (point B in fig. 6) The clock and counter required to drive the multiplexer will be discussed in detail in section 7.2.



BLOCK DIAGRAM OF LOAD MONITORING OUTSTATION

Fig. 6

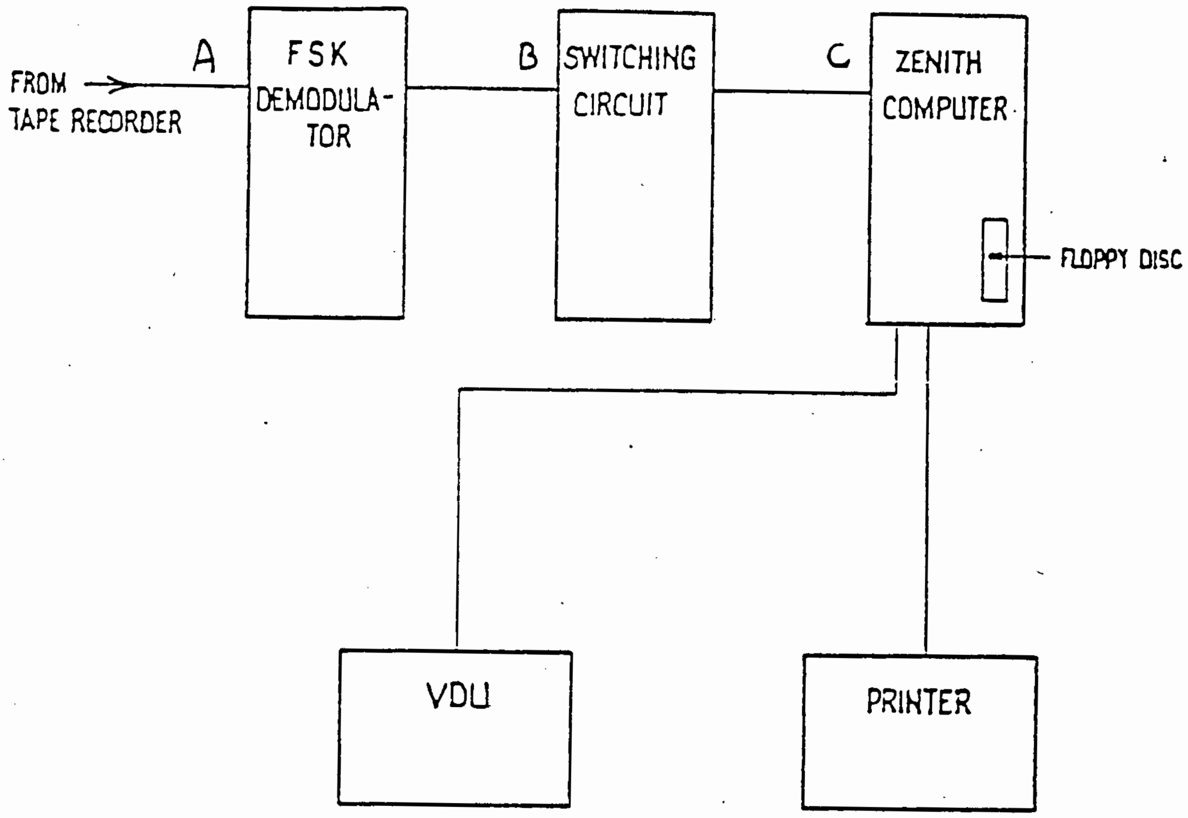
The information is then amplified to a suitable level for input to the A/D converter (point C on the diagram). The analogue information is passed through an 8 bit Analogue/Digital Converter to give a corresponding digital output on 8 lines (point D in figure 6). Synchronisation for writing information into and reading information out of the A/D converter is provided by 3 monostable circuits which are discussed in paragraph 7.2.10.

The information is then stored in an E^2PROM with 2048 locations (2 K). Synchronisation for this circuit is provided by the same monostable unit mentioned in the previous paragraph. A 12 bit counter provides addressing for the E^2PROM. When all the locations in the E^2PROM contain data, the counter resets and no further information is stored.

A switch is then changed manually from "Normal" to "Read" position and the information is read out from the E^2PROM into a Universal Asynchronous Receiver Transmitter (UART) (point E on fig. 6). The UART converts the parallel data to serial format. The data stream (300 Bd) then passes into an FSK modulator (Point F on fig. 6) The modulated data (at standard CCITT freq) is then passed into a tape recorder for storage on a conventional cassette tape. (point G on fig. 6) This completes the description of the outstation block diagram.

7.1.3 Discussion of the Master Station Block Diagram

Fig. 7 shows the block diagram of the Load Monitoring System Master Station. This station is situated at the Urban Control Centre.



BLOCK DIAGRAM OF LOAD MONITORING MASTER STATION

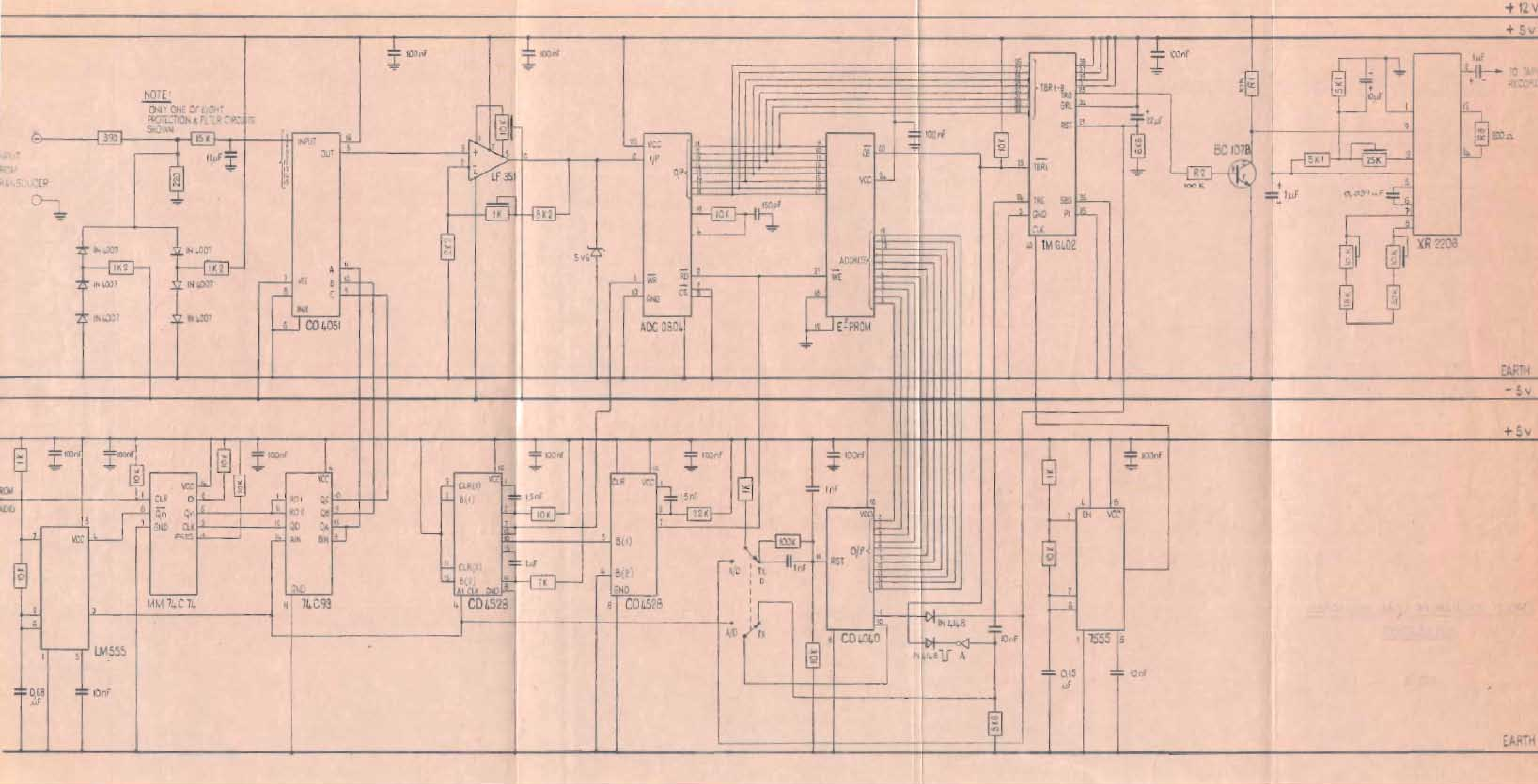
Fig. 7

The maintenance official deposits the cassette tape at the control centre. Data is read into the Zenith computer by the following method. Control Centre staff prepare the computer for receiving data through its RS 232 port. The software for doing this is discussed at a later stage in this report. Data (FSK) is transferred from the tape into an FSK demodulator. (point A in fig. 7)

The FSK demodulator converts the FSK data into a square wave serial data stream (0 - 12 V). RS 232 protocol requires a -12 V to +12 V swing so the data is passed through a transistor circuit to achieve this (Point B on fig. 7). Once in RS 232 protocol the data is fed into the Zenith computer ready for storage on floppy disc. (Point C on fig. 7).

CIRCUIT DIAGRAM OF LOAD MONITORING OUTSTATION

FIG. 8



+12v
+5v
NOTE:
ONLY ONE OF EIGHT PROTECTION & FILTER CIRCUITS SHOWN
INPUT FROM TRANSDUCER
CD 4051
LF 351
ADC 0804
E²PROM
TM 6402
BC 107B
XR 2206
LM555
MM 74C74
74C93
CD4528
CD4528
CD4040
7555
EARTH
-5v
+5v
EARTH

The same software which reads the data into the computer performs manipulations on the data and stores it on disc in a form suitable for use by the Lotus spread sheet system. When graphs and reports are required the data is read off the disc and Lotus is utilized to give the necessary tables and graphs. An explanation on how Lotus was used in this application is included in section 9.0.

7.2 Detailed Electronics Design of Load Monitoring Outstation

The detailed circuit diagram of the outstation is shown in fig. 8. The design of each circuit on the diagram will be discussed in detail in this section. Please refer to figure 8 while reading this section to see how the circuits interface with each other.

7.2.1 Protection and filter circuit

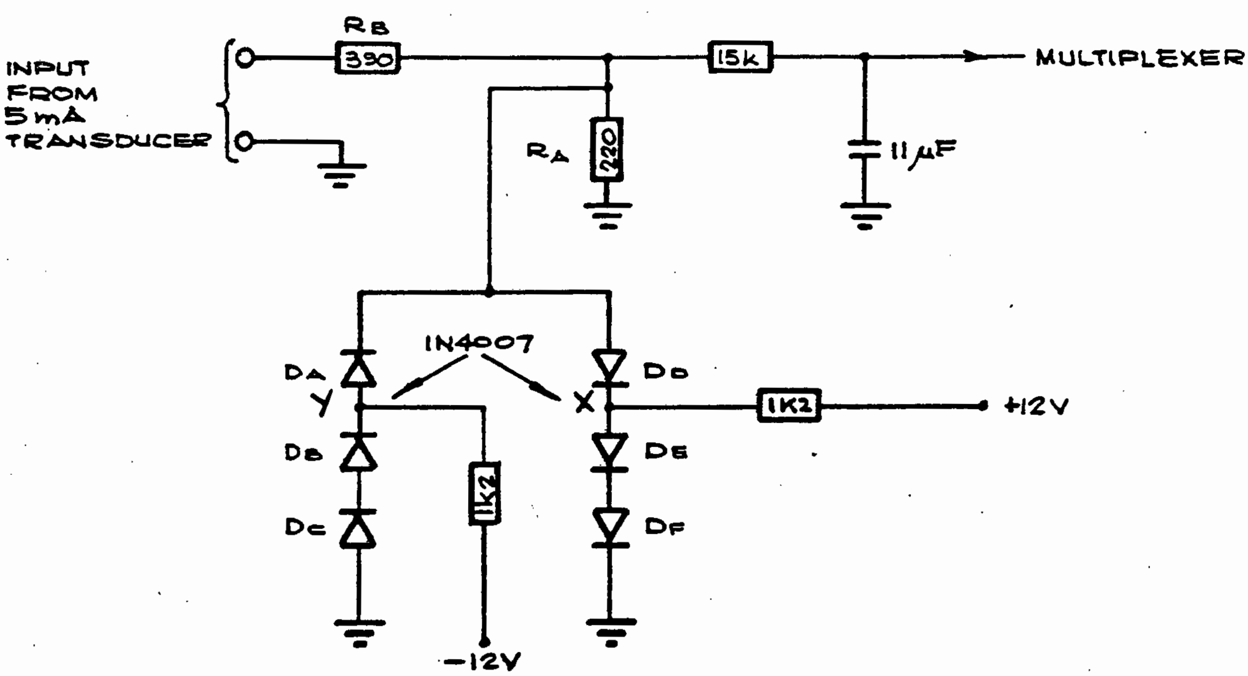


DIAGRAM SHOWING AND PROTECTION AND FILTERING CIRCUIT

Fig. 9

The main objective of this circuit is to provide protection against overcurrent and overvoltage and filtering of high speed transients for the remainder of the outstation circuitry. The outstation is situated in a confined area together with indoor switchgear operating at 11 kV so it was considered necessary to include protection and filtering circuitry at the input stage to the outstation board.

The transducers provide a current output which flows through R_B and R_A to ground. The main function of R_A is to provide a voltage value corresponding to the current input. This voltage is then applied to the multiplexer. A value of 220 Ohm was chosen as this provides a usable voltage level to the multiplexer.

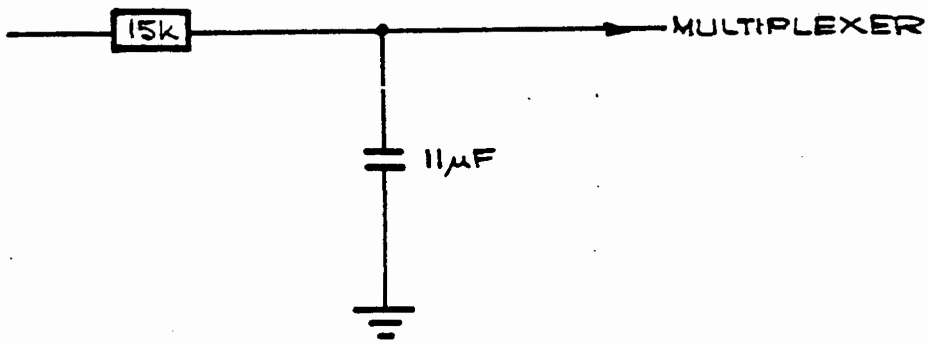
eg. a maximum input current of 5 mA results in the following voltage:

$5 \times 10^{-3} \times 220 = 1,1$ V

It is necessary to use a high tolerance (5%) resistor to ensure high accuracy. The initial resistor R_B (390 Ohm) acts as a current limiter during over current conditions.

The section of circuit using the diodes and 1,2 K resistors was designed to limit the voltage across the 220 Ohm resistor ie. the input voltage to the multiplexer. Under normal operating conditions the diodes D_e, D_f are switched on. At point X in figure 9 for example the voltage will be 1,2 V owing to the 0,6 V drop across each of diodes D_e + D_f. Diodes D_a and D_d remain switched off (0,6 V across each one) unless the voltage at point Y in fig. 11 rises above 1,8 V. At this point the Diode D_d turns on and the voltage is clamped at 1,8 V. If the voltage were to rise above 1,8 V damage would occur further along the circuit after amplification has taken place. The same reasoning applies with respect to negative voltages for diodes D_a, D_b + D_c although this condition is never likely to occur.

The circuitry shown in Fig. 10 was designed to act as a low pass filter to filter out noise and high speed spikes.



CIRCUIT DIAGRAM OF LOW PASS FILTER

Fig. 10

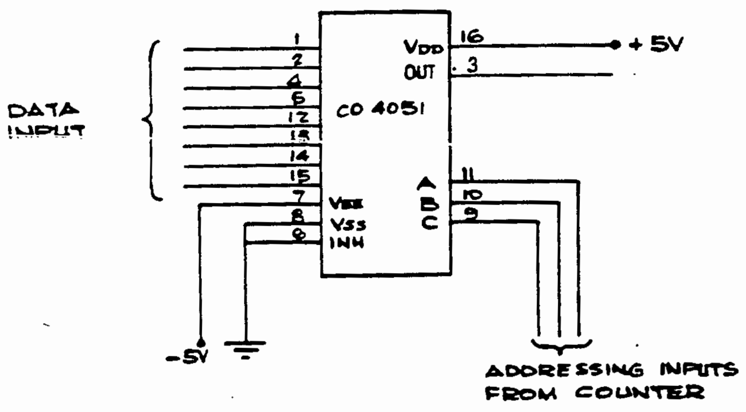
The time constant of this circuit is:

$$RC = 15 \times 10^{3} \times 11 \times 10^{-6}$$

$$= 0,165 \text{ seconds}$$

Design of this part of circuitry was taken from the Intrac outstation. It is a proven design and has adequately protected the Intrac equipment in stepdown substations. No impulse test equipment was available to check the operation of this part of the circuitry.

7.2.2 Multiplexer Circuit



SINGLE 8-CHANNEL ANALOGUE MULTIPLEXER - CD4051

Fig. 11

The integrated circuit shown in fig. 11 is the 8 channel analogue multiplexer (CD 4051) selected for use in the outstation design. It was decided to use a separate multiplexer integrated circuit rather than a combined multiplexer and analogue to digital converter. If the combined unit were used, eight amplifier circuits would be necessary. This would add to cost as well as cause unnecessary complication of the Printed Circuit Board design. In the design approach used only one amplifier is necessary.

The multiplexer has three binary inputs A, B, C (pins 9, 10, 11) and an input inhibit (pin 6) which is not used in this case. The three binary signals select one of eight channels in the multiplexer to be turned 'on' and hence connect the input to the output.

Pin 16 (V_{dd}) is connected to +5 V

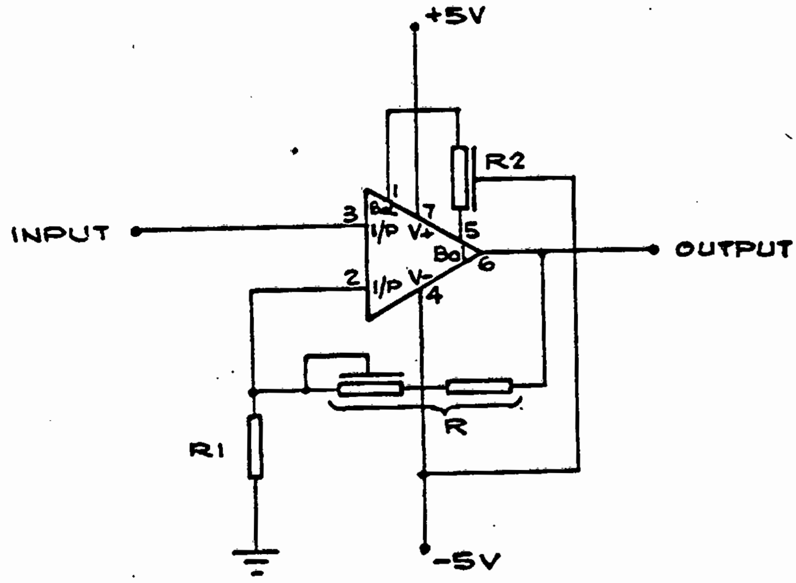
Pin 8 (V_{ss}) is connected to ground

Pin 7 (V_{ee}) is connected to -5 V

Pins 9, 10 + 11 (A, B + C) are connected to the outputs of a 4 bit binary counter (74C93) which will be discussed later. A is connected as the Least Significant Bit (LSB).

Pin 3 is the single output line to the amplifier while pins 1, 2, 4, 5, 12, 13, 14, 15 are the 8 input lines from the protection and filter circuits.

7.2.3 Amplifier Circuit



AMPLIFIER CIRCUIT USING LF351 OPERATIONAL AMPLIFIER

Fig. 12

In order to provide the Analogue to Digital Converter with the required input of 0 - 5 V, an amplification stage is necessary.

Figure 12 shows the configuration of the amplifier circuit. The central component in this circuit is the LF 351 Wideband JFET Operational Amplifier. This is a high speed low cost operational amplifier with an internally trimmed input offset voltage. The device requires a low supply current and yet maintains a large gain bandwidth and a fast slew rate. This is important in this design as the amplifier must follow input from the multiplexer accurately and quickly. The device also has low noise and offset voltage drift. Overall this device was most suitable for use in this design where accuracy is of importance.

As was discussed in paragraph 7.2.1 on Protection and Filter circuits, a maximum input of 5 mA produces a voltage of 1,1 V at the output of the multiplexer. This voltage must be amplified to a value of 5 V to give a full scale reading (255) on the output of the Analogue to Digital Converter.

$$\text{Gain of amplifier} = R/R_1 + 1$$
$$= 5/1,1$$
$$= 4,5$$

Choose R_1 to be 2,2 kOhm

$$\frac{5}{1,1} = \frac{R}{2,2 \times 10^3} + 1$$

$$R = 7,8 \text{ kOhm}$$

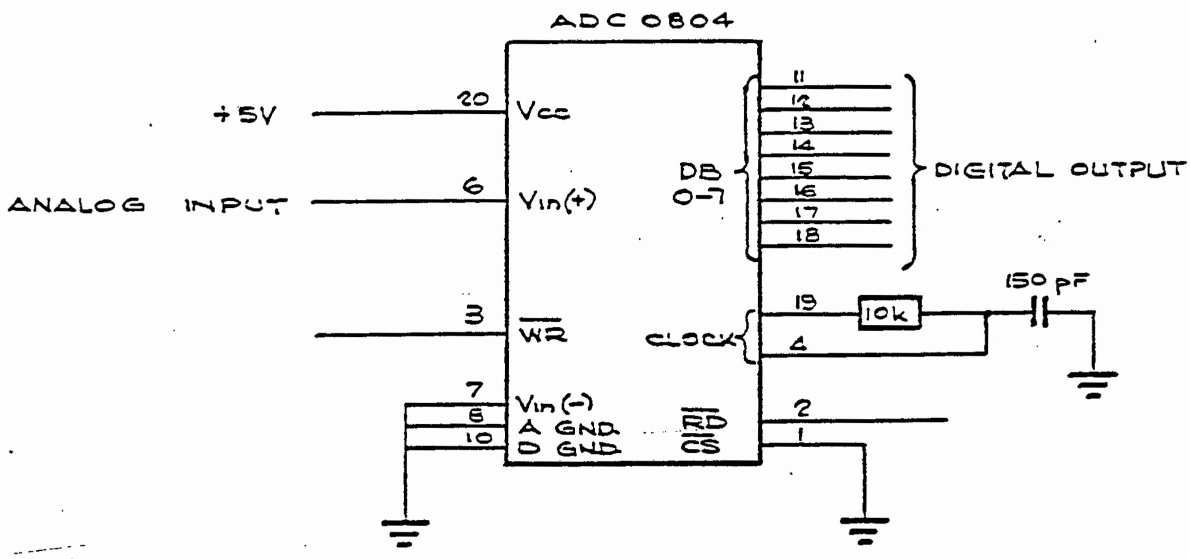
It is most convenient to divide up the resistor into a fixed component and a variable component for adjustment purposes.
Choose a fixed resistor of 6,8 k Ohm and a variable resistor of 2,2 k Ohm (maximum).

It is now possible to adjust gain so that an input current of 5 mA produces an output of 5 V from the amplifier.

The 10 kOhm variable resistor R_2 is used to zero offset voltage.

Note that the amplifier is used in non inverting mode.

7.2.4 Analogue to Digital Converter



ANALOGUE TO DIGITAL CONVERTER CIRCUIT - ADC 0804

Fig. 13

The circuit shown in figure 13 is a CMOS 8-bit successive approximation A/D Converter. Although 12 bit A/D converters are available, it was decided that 8 bit resolution was sufficent for this application.

Considering that an input of 5 V gives an output of 111111 or 255 each step is 5/255 = 19,6 μV.

This gives rise to an accuracy of:

$$\frac{19,6 \times 10^{-3}}{5} = 1\%$$

which was considered sufficiently accurate.

The conversion time of this A/D converter is 100 μs. (ie 100 kHz conversion rate). Converters are available with faster conversion times eg. ZN 427 with a conversion time of 10 μS.

The system clock operates at 1 kHz hence it was decided that a conversion time of 100 μs was quite adequate.

The ADC 0804 is a reasonably priced readily available component. Its use in the design therefore contributes to the overall objectives set out in section 6.0.

The pins of the ADC 0804 are connected as follows:
Input data from the amplifier is provided at pin 6. A zener diode is connected across the input to the A/D Converter to limit input voltage to 5 V.
Pin 8 (A GND) and Pin 10 (D GND) as well as Pin 7 (Vin -) are connected to ground. Vin + (Pin 6) is thus the input from the amplifier.
Pin 9 (span adjustment) is not connected in this case as an internal reference of Vref/2 of 4,5 is provided. This is correct in this case and it is not necessary to force V_{ref} to another voltage.
Pins 19 and 4 are connected to a 10 kOhm resistor and 150 pF capacitor as shown in figure 15. These components are essential to the correct operation of the on board clock generator.
Pin 20 is connected to the +5 V supply rail. A decoupling capacitor of 100 nF is provided.

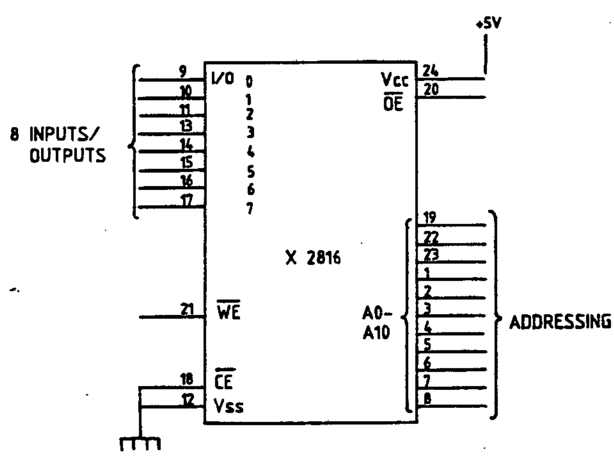
Pins 11 - 18 provide the 8 bit parallel line digital output of the A/D converter. The output on pin 18 is the least significant bit (LSB) while that on pin 11 is the most significant bit (MSB). Pin 1 (Chip select) is permanently tied to ground ie. this circuit is always ready to operate.

Pin 2 $\overline{\text{(Read)}}$ is connected to the output of a monostable (discussed in paragraph 7.2.10). The negative pulse allows the value on the input (pin 6) to be read into the A/D converter.

Pin 3 $\overline{\text{(Write)}}$ is connected to the output of the 3rd monostable (discussed in paragraph 7.3.11). The negative pulse generated by this monostable enables the converted digital value of the input to appear on the 8 output lines DBO - DB7 of the A/D Converter. The write pulse delays the read pulse by approximately 0,3 milliseconds to enable the A/D conversion to take place. The same write pulse enables the digital output to be read into the E^2PROM storage device.

Pin 5 is the Interrupt pin. It is not connected as the A/D converter is not used in conjunction with a microprocessor.

7.2.5 E^2PROM



2K 8 BIT ELECTRICALLY ERASABLE PROM CIRCUIT

Fig. 14

The device shown in fig. 14 is the 2 K E^2PROM used for storage of analogue data on the outstation board. The E^2PROM was chosen as the storage device instead of the simpler RAM despite greater cost. The reason for this decision is that the E^2PROM will retain stored data in the event of power failure whereas all data is lost from the RAM during the same conditions.

In order to write data into the E^2PROM and read data from it the following conditions are required:

$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	MODE	I/O
L	L	H	Read	Dout
L	H	L	Write	Din

The $\overline{CE}$ (Chip Enable pin 18) must be low in both conditions, hence it is tied to ground. During the write condition $\overline{WE}$ ($\overline{\text{Write Enable}}$ pin 21) must be low. The same negative pulse which enables data to be read from the A/D converter is applied here. The $\overline{OE}$ (output enable pin 20) is held high during this stage. $\overline{OE}$ is connected to the $\overline{TBRL}$ ($\overline{\text{Transmitter Buffer Register Load}}$) pin of the UART which is held high when the UART is not in operation. During the Read condition (ie. when data is read out of the E^2PROM into the UART) the system clock is not in operation. This means that the monostable output is continuously high and $\overline{OE}$ on the E^2PROM is held high. The UART is operating during this stage, however, and a negative pulse provided by $\overline{TBRL}$ on the UART is applied to $\overline{OE}$ on the E^2PROM which enables data to be read from a particular memory location of the E^2PROM.

During the Write phase data is applied to pins 9, 10, 11, 13, 14, 15, 16 and 17 from the A/D converter. During the Read phase data is provided as output from the same pins on the E^2PROM. During this stage the system clock is not operational and there is no data on the output of the A/D Converter to create confusion.

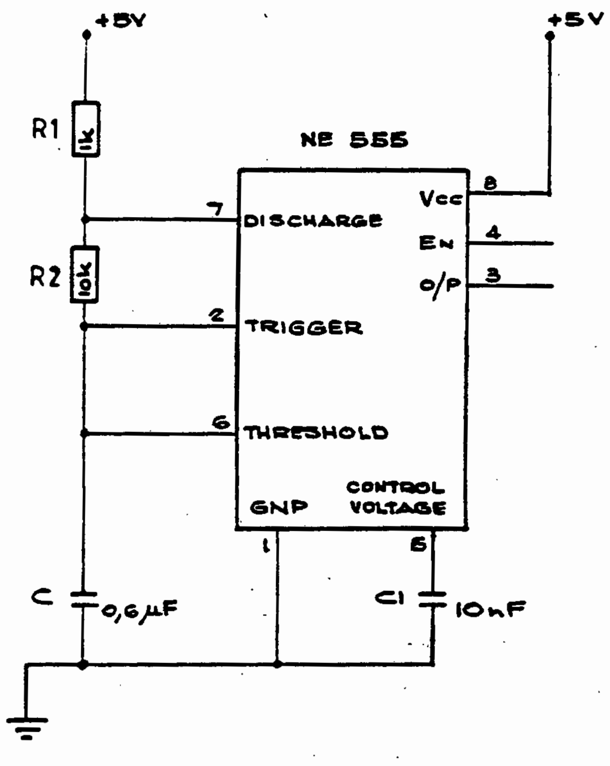
Addressing for the E^2PROM is provided by an 11 bit counter (CD 4040).

Pins 1, 2, 3, 4, 5, 6, 7, 8, 19, 22 and 23 are the address pins of the E^2PROM.

Pin 24 is connected to +5 V and

Pin 12 is connected to ground.

7.2.6 System Clock



SYSTEM CLOCK CIRCUIT INCORPORATING NE 555 TIMER

Fig. 15

The device shown in fig. 15 is referred to as the System Clock. This name is slightly misleading as the device provides clock pulses for the majority of the circuitry on the board but only for the stage up to and including storage of data in the E^2PROM. When data is read out of the E^2PROM into the UART, clock pulses are provided for the UART by a second clock running at 4800 Hz while the System Clock is switched off. The reason for this will become clear if the reader refers to paragraphs 7.3.10 and 7.3.12 which explain the design of the UART and its associated circuitry.

The basis of the system clock circuit is the NE 555 CMOS counter. This is a highly stable device used for generating accurate time delays or oscillation. Terminals are provided for triggering and resetting. For astable operation as an oscillator the free running frequency and duty cycle are accurately controlled with two external resistors and one capacitor. These are indicated as R_1, R_2 and C in fig. 15.

A frequency of 1 kHz was selected for the system clock. The A/D converter used in the design has a conversion time of 1 µS. The system clock period is one order slower (1 ms) and hence no errors will occur in the A/D conversion.

<u>Calculation of resistors $R_1 + R_2$ and capacitor C.</u>

The following formulas are used:

t_h = $0{,}693\ (R_1 + R_2)\ C$ ....(1)

t_l = $0{,}693\ R_2C$ ....(2)

where t_h = time for which output is high

t_l = time for which output is low

In our case $t_l = t_h$

Consider R_1 to be 10 times smaller in value than R_2. It can thus be neglected.

Select $R_2 = 10\ k\Omega$

Let t = 0,5 ms (since $t_h + t_1 = 1$ ms
for f = 1 kHz)

Using formula (2): $C = \dfrac{0{,}5 \times 10^{-3}}{0{,}693 \times 10 \times 10^3}$

$= 0{,}72\ \mu F$

Select values as follows:

$C = 0{,}68\ \mu F$

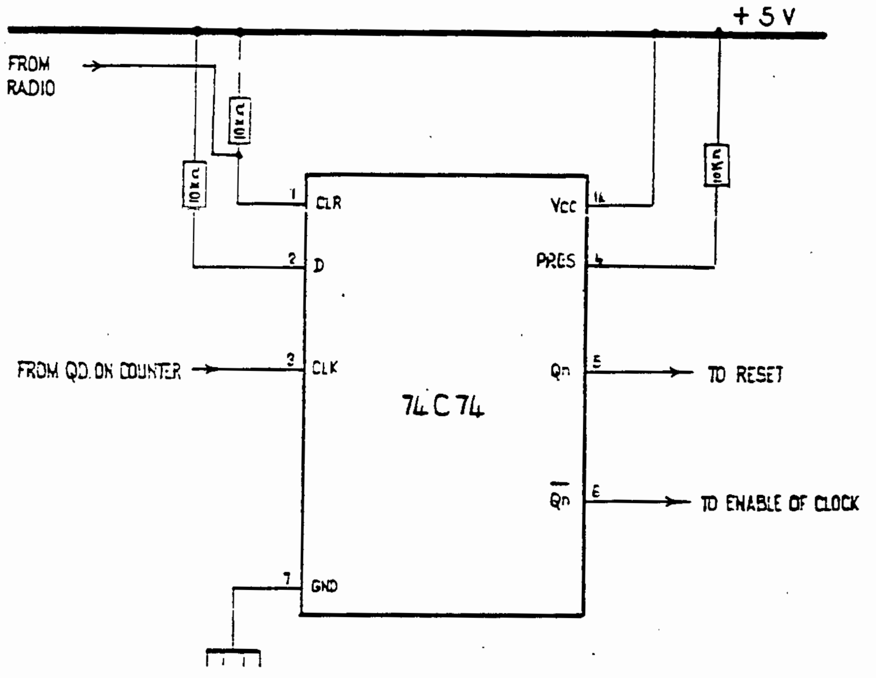
$R_1 = 1\ k\Omega$

$R_2 = 10\ k\Omega$

$C_1 = 10$ nF as recommended by the CMOS handbook.

Referring to fig. 15 pin connections are as follows:
Pin 1 is the ground connection while Pin 8 is connected to +5 V.
The Supply is decoupled using a 100 nF capacitor.
Pin 3 is the output of the oscillator circuit.
Pin 4 is used to enable the clock. This pin is normally low (clock not enabled) and is connected to the Q_n output of a set reset flip flop. The flip flop provides a high pulse for a time period being enough to enable the clock to provide 8 output pulses which enables 8 analogue inputs to be scanned.
Pin 2 (Trigger), Pin 6 (Threshold) and pin 7 (discharge) are connected as shown in fig. 17.
The control voltage pin (5) was not used in this case.

7.2.7 Clock Enable Circuit



DUAL D FLIP FLOP CIRCUIT MM 74C74

Fig. 16

Only one of the Flip Flop circuits contained in the package shown above is used in the design of the outstation board. The circuit is implemented into the design as follows:

Pin 14 is connected to +5 V supply.

Pin 7 is connected to ground.

Pin 1 (Clear) is normally held high but is connected to the output from the radio receiver which provides a negative pulse of 2 ms duration when a "scan signal" is received.

Pin 3 (Clock) is connected to the Q_d (4th count) output of the 74C93 4 bit binary counter. This pin goes high momentarily when the 4th count is reached ie. after 8 clock pulses have been produced.

Pin 2 (Data) is tied to the +5 V rail via a 10 kOhm resistor and so is permanently at a high level.
Pin 4 (Preset) is held high.

The truth table of the flip flop is included here so the reader can further understand the purpose of the flip flop.

Truth Table

	Preset	Clear	Q_n	$\overline{Q}_n$
	0	0	0	0
	0	1	1	0
A	1	0	0	1
	1	1	*Q_n	*$\overline{Q}_n$

*No change in output from previous state.

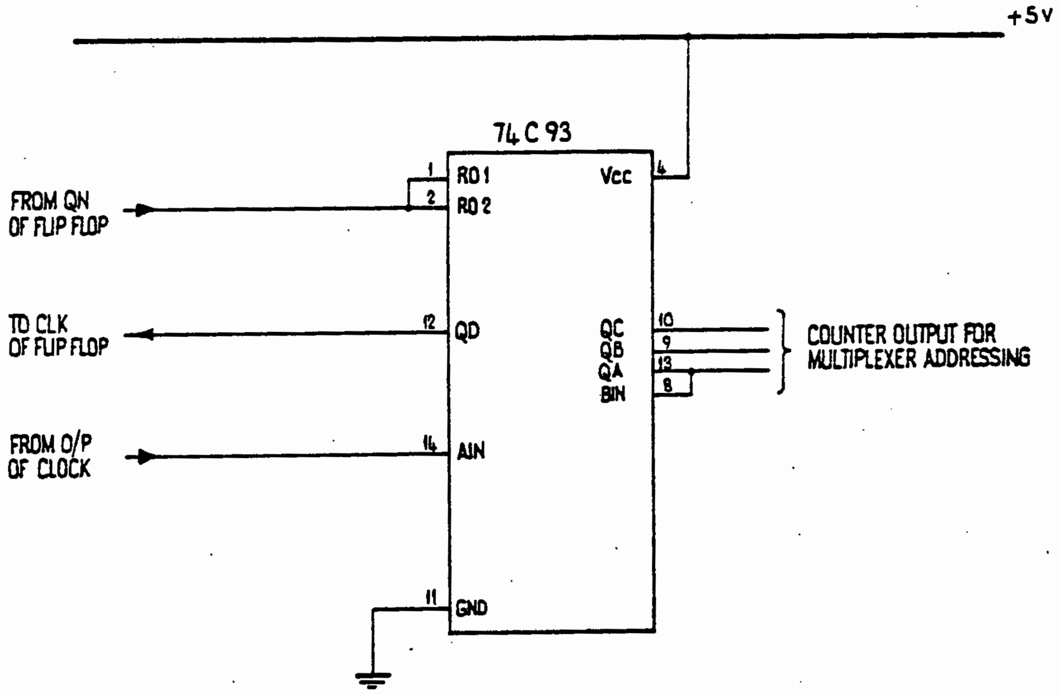
TRUTH TABLE FOR D FLIP FLOP

Fig. 17

The pulse applied to the 'clear' input of the input lasts only 2 ms and is considerably less than 8 clock pulses. During the application of the negative pulse at Clear, the preset input is permanently tied to 1. This gives rise to the condition marked A in the Truth Table. The output $\overline{Q_n}$ (pin 6) goes high and the system clock is enabled. After 2 ms the radio pulse disappears but the output is latched and the clock remains enabled until the fourth count is received from Q_d of the counter at the clock (pin 3). The high level at the data input (pin 7) is clocked through to Q_n which causes $\overline{Q_n}$ to become 0.

This causes the system clock to stop. The high level at Q_n in turn causes the counter to reset. The system remains in this condition until the next radio pulse is received and the process repeats itself.

7.2.8 4-Bit Binary Counter



4-BIT BINARY COUNTER 74C93

Fig. 18

The purpose of the counter illustrated in fig. 18 is to provide the inputs A, B, C to the multiplexer CD 4051 (pins 9, 10 & 11) in order to enable scanning of the eight analogues to take place. Refer to paragraph 7.3.2 for a discussion on this multiplexer.

As the 74C93 is a 4 bit counter it is possible to count to 2^4 (16) before resetting. In this case a count to 2^3 (8) is required and the 4th count is used at the Clock input of the 74C74 D flip flop which in turn resets the clock after 8 pulses.

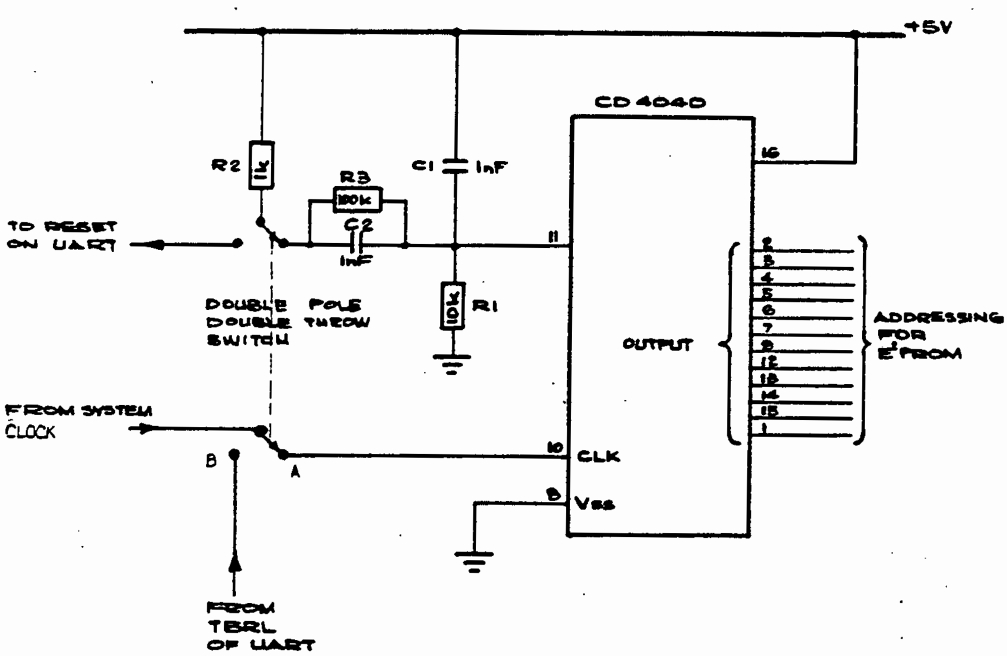
Pins 1 + 2 are the reset pins for the counter and are connected to the Q_n output of the D flip flop which resets the counter after 8 counts.

Pin 4 is connected to +5 V.

Pin 11 is connected to ground.

Pin 14 is the clock input to the counter. The divide by 2 count appears on pin 13 and while this is used as the A input to the multiplexer, it is also connected to Bin (pin 8) of the counter to act as input to the second stage. Outputs B + C appear at pins 9 and 10 respectively. The 4th count, D (pin 12) is used as input to the clock of the D flip flop as discussed previously.

7.2.9 <u>12-Stage Ripple Carry Binary Counter</u>



12-STAGE RIPPLE CARRY BINARY COUNTER - CD4040 AND ASSOCIATED CIRCUITRY

Fig. 19

The purpose of the counter illustrated in figure 19 is to provide addressing for the E^2PROM discussed in section 7.2.5. A total of 2048 locations in the E^2PROM must be addressed. The CD4040 can count to a total of 2^{12} ie. 4096. It is thus unnecessary to use the 12th count for addressing purposes.

The pins on the counter are used as follows:
Pin 16 is connected to +5 V while
pin 8 is connected to ground.
Pins 2, 3, 4, 5, 6, 7, 9, 12, 13, 14, 15 are the counter outputs and are connected to the address pins of the E^2PROM.
Pin 11 is the reset pin and is connected as shown in fig. 19. Whenever the power supply is switched on the 1 nF capacitor (C_1) and 10 K resistor (R_1) provide a pulse of duration:

$RC = 10 \times 10^3 X 1 X 10^{-9} = \underline{0{,}01 \text{ ms}}$ to pin 11.

This causes the counter to reset. In addition when the double pole double throw switch S1 is switched from A to B position a similar pulse is applied to pin 11 and the counter is reset. When the switch is in the A position (data being scanned) the system clock is able to drive the counter whereas when it is in the B position (data being read out through UART) the counter is driven by clock pulses from the UART. The counter unit must be reset before this process starts to ensure data is read from location 1 in the E^2PROM, hence the inclusion of R_2 and C_2. Resistor R_3 is included to ensure that C_2 does not hold a residual charge.

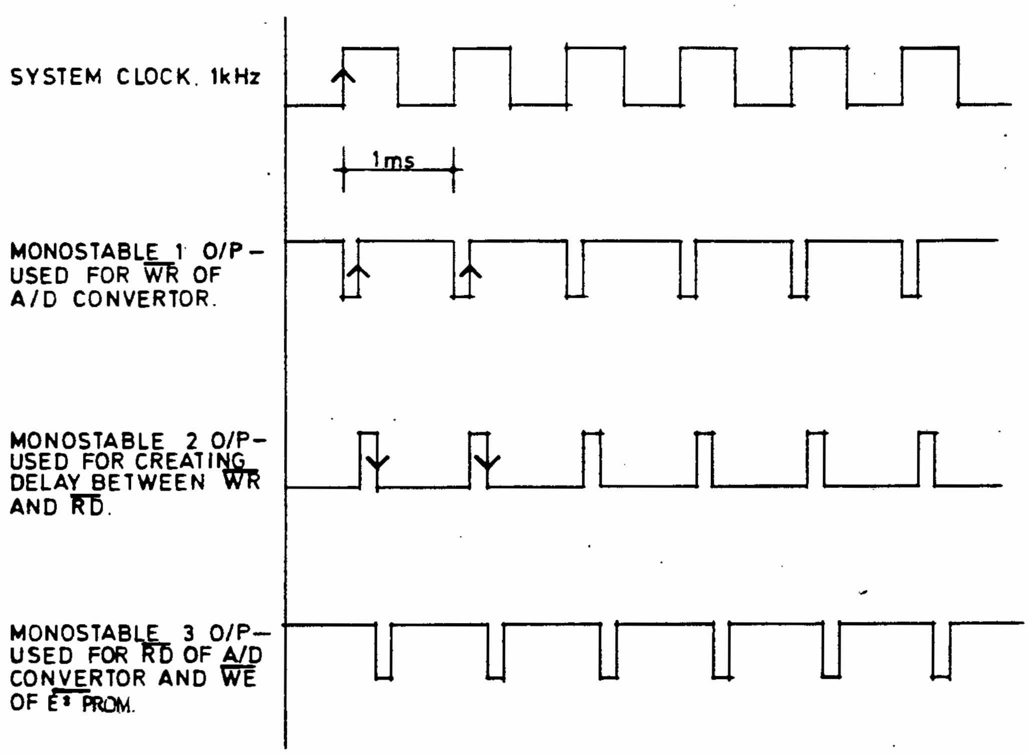
The output at Pin 1 is the 12th count provided by the counter and is not used for address purposes. During the phase of reading the data out of the E^2PROM into the UART the transistion from low to high on this pin is used to reset the UART as all the data has been read out of the E^2PROM at that stage.

7.2.10 Timing Diagram of System and related monostables

The 1 kHz system clock discussed in paragraph 7.2.6 synchronizes the operation of the outstation during the stage when data is being read into the E^2PROM. The system clock controls three monostable circuits which provide synchronisation pulses for data to be converted from Analogue to Digital form and then stored in the E^2PROM.

During the second stage of operation when the system clock is not in operation the UART provides synchronisation pulses for the system. This aspect will be discussed later in paragraph 7.2.11 which describes the UART in detail.

The Timing Diagram associated with the monostables is given in fig. 20.

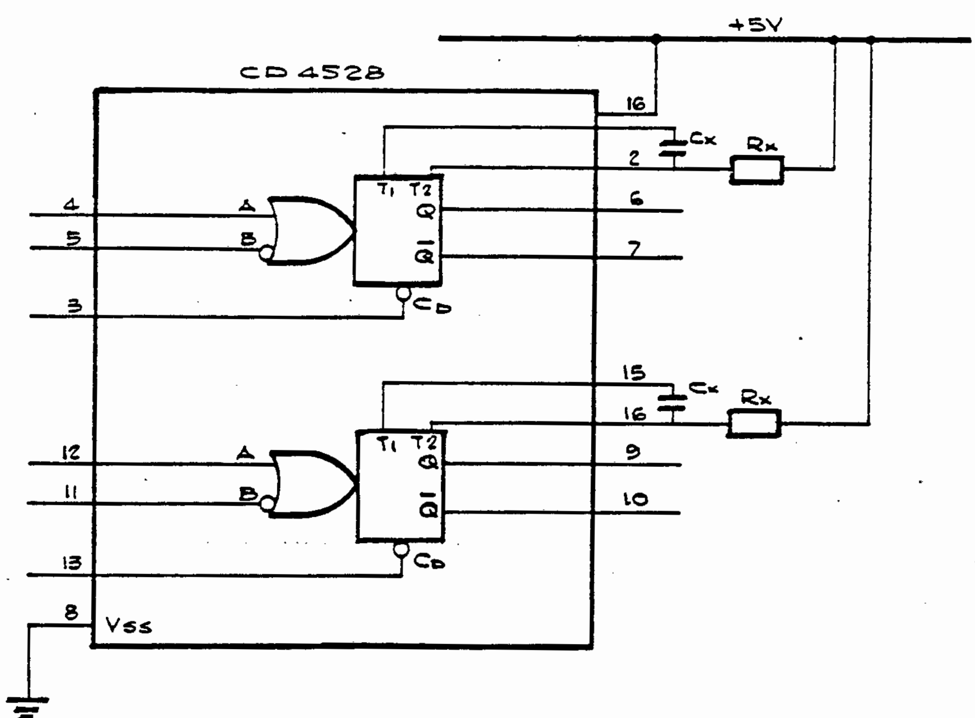


TIMING DIAGRAM SHOWING MONOSTABLE OUTPUT PULSES USED TO SYNCHRONIZE THE OPERATION OF GATHERING AND STORING DATA

Fig. 20

From the timing diagram it is obvious that three monostable circuits are required. The output of the 1st monostable is used for $\overline{WR}$ on the A/D converter while the output of the 3rd monostable is used for $\overline{RD}$ on the A/D converter. The 2nd monostable is used as an intermediate stage to introduce a delay between the $\overline{RD}$ and $\overline{WR}$ pulses.

The pin layout of the monostable circuits which provide the pulses is shown in fig. 21.



PIN LAYOUT OF DUAL MONOSTABLE MULTIVIBRATOR CIRCUIT - CD 4528

Fig. 21

Note that only a general diagram is shown in figure 21. Please refer to fig. 8 for detail on the implementation of the monostable in the overall system. The Truth Table for the monostable circuit is shown in fig. 22.

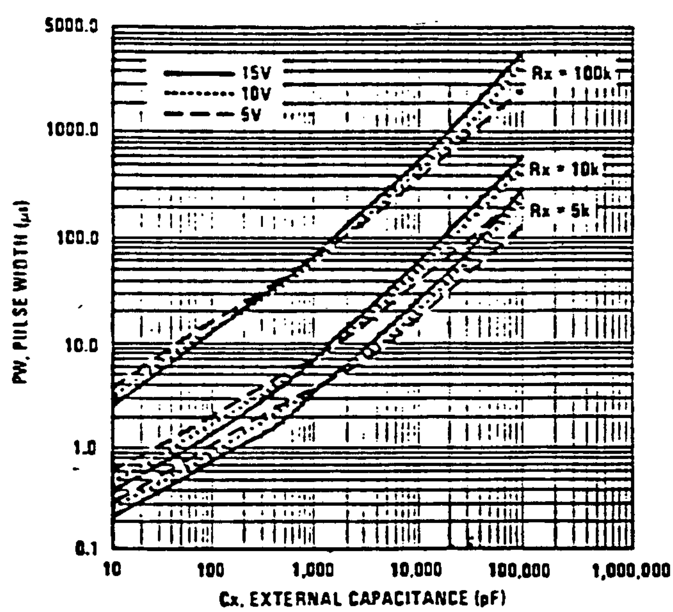
INPUTS			OUTPUTS	
CLEAR	A	B	Q	$\overline{Q}$
L	X	X	L	H
X	H	X	L	H
X	X	L	L	H
H	L	↓	⊓	⊔
H	↑	H	⊓	⊔

H = High level
L = Low level
↑ = Transition low to high
↓ = Transition high to low
⊓ = One high level pulse
⊔ = One low level pulse
X = Irrelevant

TRUTH TABLE FOR MONOSTABLE CD 4528

Fig. 22

The graph used for determining the external values Rx and Cx is shown in fig. 23.



GRAPH OF PULSE WIDTH VERSUS EXTERNAL CAPACITANCE Cx FOR VARYING VALUES OF Rx

Fig. 23

The CD 4528 dual monostable circuit was chosen since it incorporates two monostables in one IC. package. In order to produce the outputs shown in the timing diagram in fig. 20 the Truth Table in fig. 22 and the graphs shown in fig. 23 were used as follows:

Requirements for Monostable 1

(i) Negative pulse
(ii) Triggers on positive edge of clock pulse
(iii) Pulse width should be approximately 10 μs.
Pin 4 (A) is connected to the clock input.
Pin 5 (B) and 3 (Clear) are tied to +5 V to trigger the monostable on the positive edge of the clock pulse.
Pin 16 is tied to +5 V and Pin 8 is tied to ground.
To establish a pulse width of 10 μs a value of R_X = 10 K and C_X = 1500 pF was chosen.
A negative pulse is required as the output is taken from pin 7. These pulses are connected to the RD (pin 2) of the A/D converter and enable data to be read into the converter.

The same pulses are used as input to the 2nd monostable. The purpose of the 2nd monostable is to introduce a time delay between pulses produced by the first monostable (used for $\overline{RD}$ on A/D) and those produced by the third monostable (used for $\overline{WR}$ on the A/D). This delay enables the Analogue to Digital Conversion to take place as explained in paragraph 7.2.4.

Requirement for Monostable 2

(i) Positive pulse
(ii) Triggers on positive edge of output from 1st monostable.
(iii) Pulse width should be approximately 0,5 ms.
The output of the 1st monostable at pin 7 is used as the input to the 2nd monostable on pin 12.

Pins 11 and 13 are tied to +5 V to meet the triggering requirement.
To establish a pulse width of 0,5 ms a value of R_x = 100 kOhm and C_x = 20 000 pF was chosen.
A positive pulse is required so the output is taken from pin 10 and used as input to the 3rd monostable.

Requirements for Monostable 3

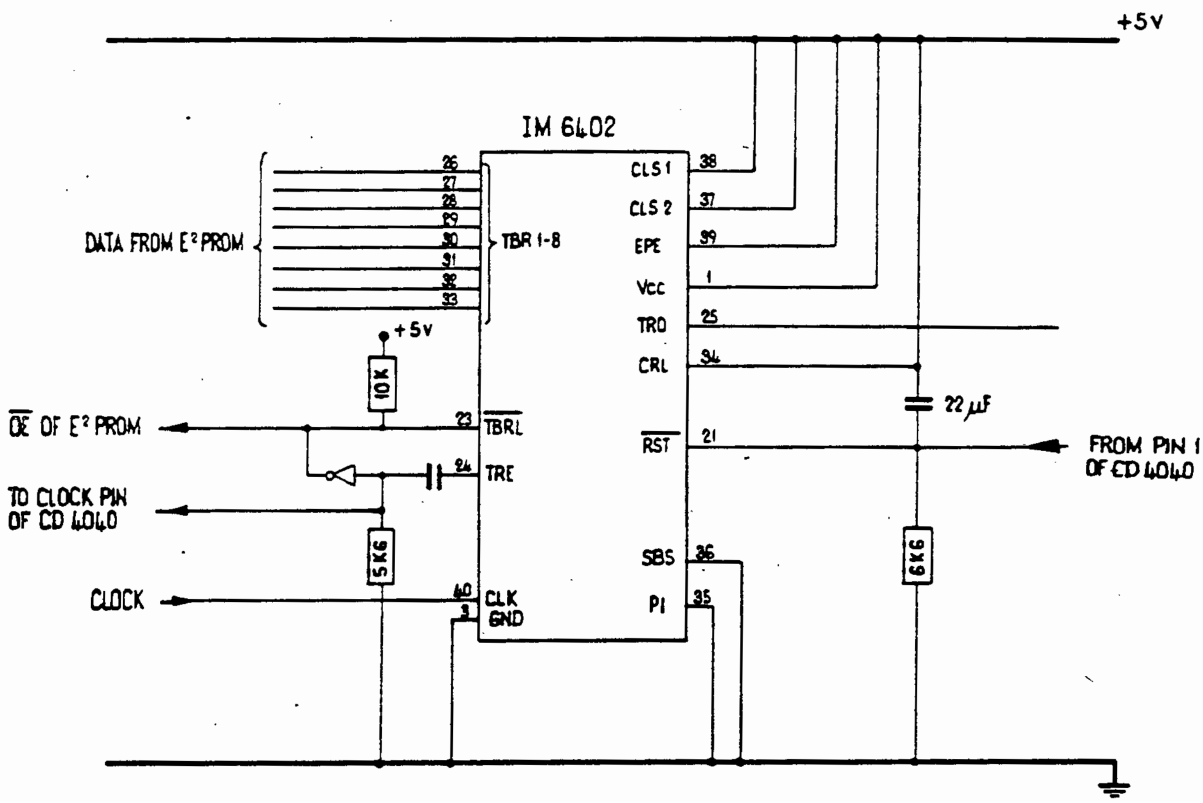
(i) Negative pulse
(ii) Triggers on negative edge of output from monostable 2
(iii) Pulse width should be approximately 10 μs.

The output from monostable 2 is connected to pin 5 (B).
Pin 4 (A) is connected to ground and Pin 3 (Clr) is connected to +5 V to satisfy the triggering requirement.
To establish a pulse width of 10 μs a value of R_x = 10 K and C_x = 1500 pF was chosen.
The output from pin 7 of this monostable is connected to the $\overline{WR}$ of the A/D Converter.

As the result of the implementation of monostable 2 in the design a delay of $\pm$ 0,5 ms exists between the $\overline{RD}$ and $\overline{WR}$ pulses which allows sufficient time for the A/D Conversion to take place.

7.2.11 Universal Asynchronous Receiver Transmitter (UART)

The functional pin definition of this UART is given in Appendix 9.10 Only the specific functions of the UART used in this particular design are illustrated in fig. 24 in order to keep the diagram as clear and simple as possible.



IMPLEMENTATION OF UART INTERSIL IM 6402 IN OUTSTATION DESIGN

Fig. 24

The 6402 UART is used in a variety of applications using asynchronous data channels. CMOS technology permits clock frequencies up to 4,0 MHz. The device is particularly suitable for this application as it is reasonably priced, readily available and adds start, stop and parity bits.

The UART is used in the second stage of operation of the outstation board ie. when data is being read out of the E^2PROM and stored on a tape. The purpose of the UART in this case is to convert the 8 lines of parallel digital data from the E^2PROM into a serial bit stream which can be converted into FSK and stored on a tape.

A baud rate of 300 was chosen as this has proved successful for this type of application. A clock running at 16 X the baud rate is required for pin 40 of the UART. For this reason a clock operating at 4800 Hz is included in the outstation design. The design of this clock is identical to that of the system clock and is not described in this report.

It was decided to configure the serial word with even parity and 1 stop bit. The word has eight data bits to correspond with the output of the A/D converter.

The 6402 UART can be hard wired to produce a variety of serial word configurations. For this application the following programming was necessary:

CLS1	CLS2	P1	EPE	SBS	Data Bits	Parity	Stop Bit
H	H	L	H	L	8	Even	1

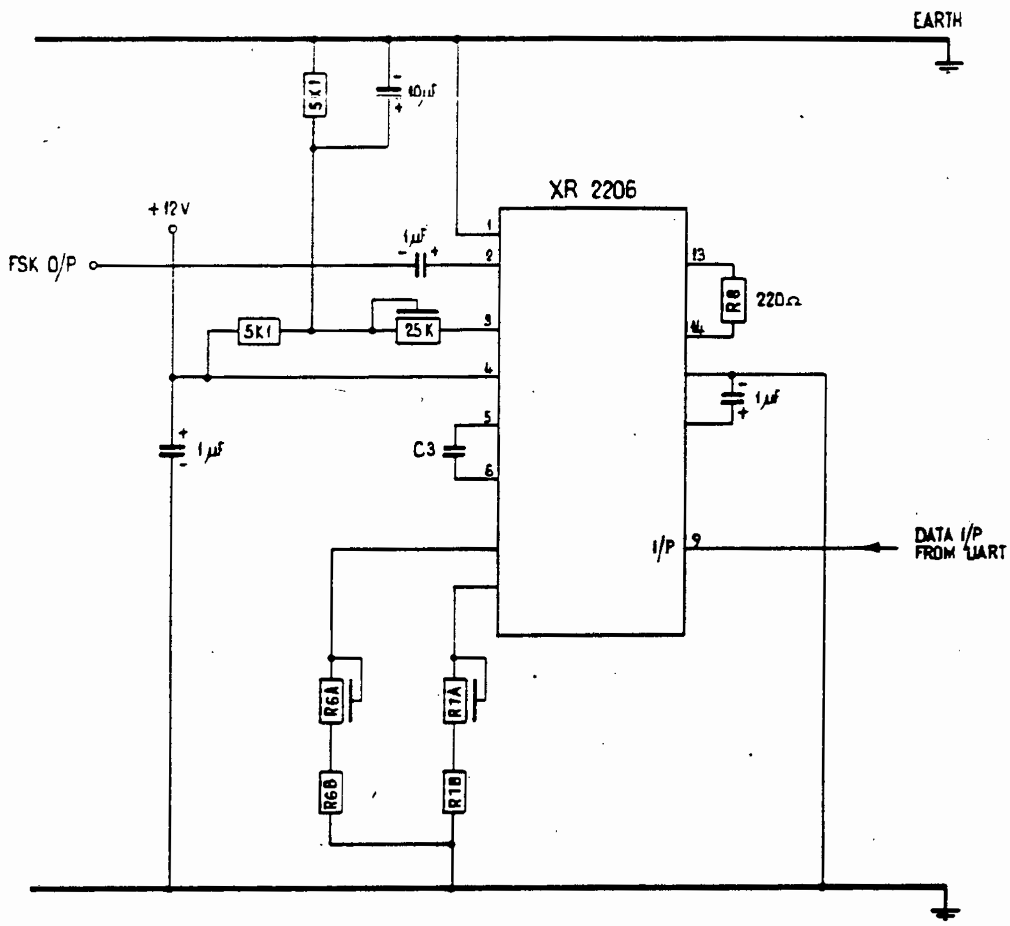
The diagram in figure 24 shows that the pins have been connected accordingly.

Pin 1 is connected to +5 V and pin 3 to ground.
Pins 26 - 33 (Transmitter Buffer Registers 1 - 8) are connected to the 8 parallel data bits from the E^2PROM.

A high level on pin 24 indicates Transmitter Register empty ie. completion of transmission of a character including stop bits. Pin 24 is normally tied to ground via the 5 k 6 resistor shown in figure 24. The 10 μF capacitor is included to remove any DC voltage level on the line. The pulse from TRE is then fed through an inverter to produce a negative pulse and is then connected as an input to pin 23 $\overline{\text{TBRL}}$ (Transmitter Buffer Register Load) which is normally tied to +5 V through a 10 K resistor. The resulting low level on $\overline{\text{TBRL}}$ transfers data from inputs TBR1 - TBR8 into the transmitter buffer register. The following low to high transition on $\overline{\text{TBRL}}$ requests data transfer to the transmitter register. If the transmitter register is busy, transfer is automatically delayed so that the two characters are transmitted end to end.

The negative pulse applied to $\overline{\text{TBRL}}$ is also used as the input to pin 20 (Output Enable) of the E^2PROM to enable data to appear on the output lines.
Pin 21 (reset) is connected via a 22 μF capacitor to +5 V and a 6k6 resistor to earth as shown in figure 24. As soon as the switch is switched to B position (to read data out of the E^2PROM into the UART) the UART is reset and parallel to serial conversion occurs. Once all data has been read out of the E^2PROM, pin 1 on the CD 4040 goes high. This high level passes through a diode 1N 4148 (for isolation purposes) and is connected to pin 21 (reset) of the UART. This condition also has the effect of resetting the UART which is necessary once all the data has been read from the E^2PROM.

7.2.12 Frequency Shift Key Modulator



SINUSOIDAL FSK GENERATOR XR-2206

Fig. 25

The serial bit stream created by the UART must be FSK modulated in order to be able to stored on a magnetic tape.

The XR-2206 shown in fig. 25 is ideal for FSK applications such as this one requiring the spectral purity of a sinusoidal waveform. It offers both TTL and CMOS compatibility, excellent frequency stability and ease of application. The XR-2206 can provide a 3V peak to peak sinewave output. Untrimmed harmonic distortion is approximately 2,5%. In addition the circuit is reasonably priced compared with say the AM 7910 and so was considered suitable for this application.

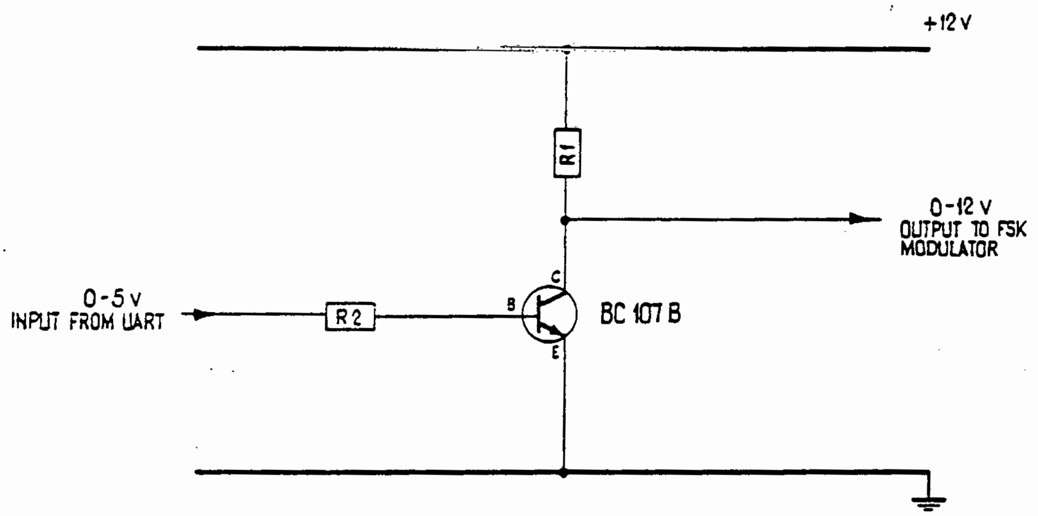
The circuit connection for the FSK Generator is shown in fig. 25. The data input is applied to pin 9. A high level selects the frequency $(1/R_7C_3)$ Hz; a low level selects the frequency $(1/R_6C_3)$ Hz, (resistors in ohms and capacitors in farads). The voltage applied to pin 9 should be selected to fall between ground and V+ (12V). In this application minimum distortion is unnecessary and pins 15 + 16 are left open-circuited while R_8 is replaced with a 220 Ohm resistor.

It was decided to use a standard FSK Band of f_l = 1070 Hz and f_h = 1270 Hz. The manufacturer recommends the following component values for this frequency band:

FSK BAND				XR 2206			
Baud rate	f_l	f_h	R6a	R6b	R7a	R7b	C3
300	1070	1270	10	18	10	20	0,039

Units : Frequency - Hz; Resistors - kOhm; capacitors - µF.

The output from the UART varies between 0 and 5V. The FSK generator, however requires an input varying between 0 and 12 V. Thus the switching circuit shown in Fig. 26 was inserted between the UART and the FSK modulator.



SWITCHING CIRCUIT

Fig. 26

The amplifier circuit was designed as follows:

A standard BC 107B npn transistor was selected for this circuit.

Choose I_c = 1 mA

R_1 = V/1 = 12V/1 mA = 12 kOhm

select R_1 = 10 kOhm

for BC 107 B $h_{fe} = 200$

$$Ib = \frac{Ic}{hfe}$$

$$= \frac{1 \times 10^{-3}}{200}$$

$$= 0,5 \times 10^{-5} \text{ A}$$

The input voltage is 0 - 5 V max. At least 0,7 V is needed to switch the transistor on.

Thus the voltage drop across R_2 must be 4,3 V.

Thus R_2 can be calculated as follows:

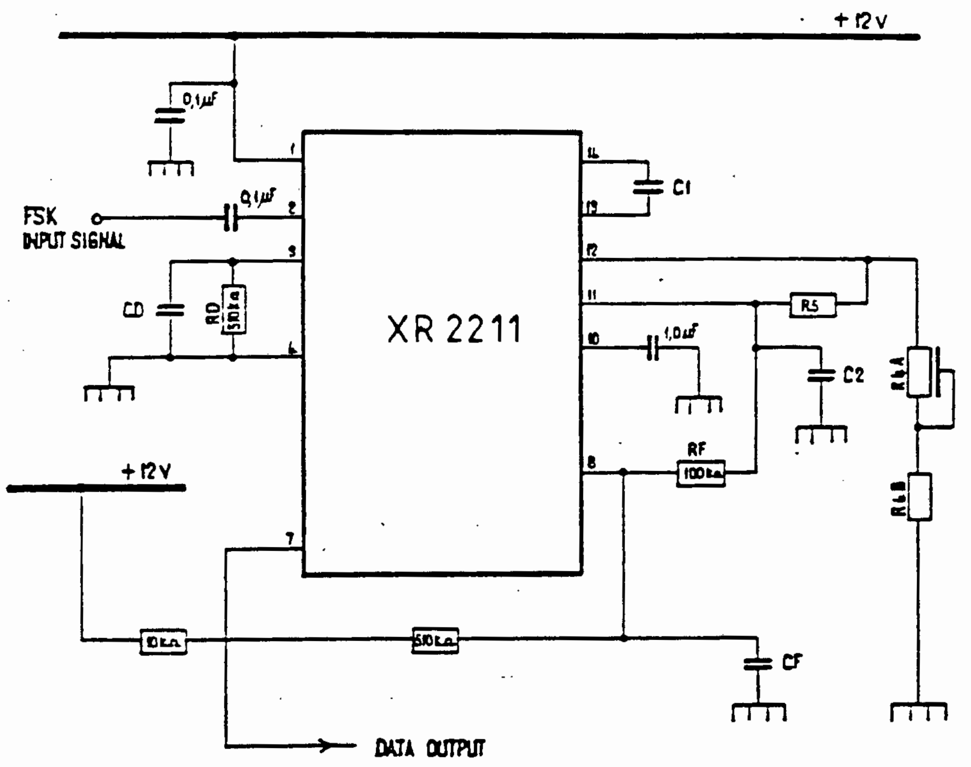
$$R_2 = \frac{4,3}{0,5 \times 10^{-5}} = 860 \text{ kOhm}$$

In order to ensure switch on of transistor make R_2 approximately 10 times smaller than above value.

$\therefore$ Select R_2 = 100 kOhm

7.3 Design of the Load Monitoring Master Station and the Interface with the Zenith Microcomputer

7.3.1 Frequency Shift Key Demodulator



FREQUENCY SHIFT KEY DEMODULATOR XR 2211

Fig. 27

At the master station data is read from the cassette tape through a demodulator into the RS 232 port of the microcomputer. It was decided to use the XR 2211 FSK demodulator for this application. This demodulator is the obvious choice as it has a relatively low cost, is readily available and is compatible with the XR 2206 modulator used at the outstations.

The XR 2211 demodulator operates on the phase locked loop principle. Its performance is virtually independent of input signal strength variations over the range of 2 mV to 3 V rms.

Please refer to fig. 27 for the circuit connections of the XR 2211. The centre frequency is determined by $fo = 1/C_1R_4$ Hz where capacitance is in Farads and resistance is in Ohms. F_o should be calculated to fall halfway between the mark and the space frequencies.

The tracking range ($\pm\Delta_f$) is the range of frequencies over which the phase locked loop can retain lock with a sinusoidal input signal. This range is determined by the formula :

$$\Delta_f = (R_4\ fo/R_5)\ \text{Hz}.$$

Δ_f is chosen to be slightly less than the difference between the mark and space frequencies.

The capture range ($\pm\Delta_{fc}$) is the range of frequencies over which the phase locked loop can acquire lock. It is always less than the tracking range. The capture range is limited by C_2, which in conjunction with R_5 defines the loop filter time constant. In most designs $D_{fc} = (80\% - 99\%)\ D_f$.

The loop damping factor (ζ) determines the amount of overshoot, undershoot or ringing present in the phase locked loop's response to a step change in frequency. This factor is determined by :

$$\zeta = \frac{1}{4}\sqrt{C_1/C_2}$$

For modern applications ζ is chosen to be 1/2.

The FSK output filter time constant (τ_F) removes chatter from the FSK output. The formula is:

$$\tau_F = R_f C d_f$$

and is normally calculated to be $\frac{0,3}{\text{baud rate}}$ seconds.

The lock-detect filter capacitor C_d removes chatter from the lock-detect output. With R_d = 510 kOhm, the minimum value of C_d can be determined by: C_d (μF) = 16/capture range in Hz.

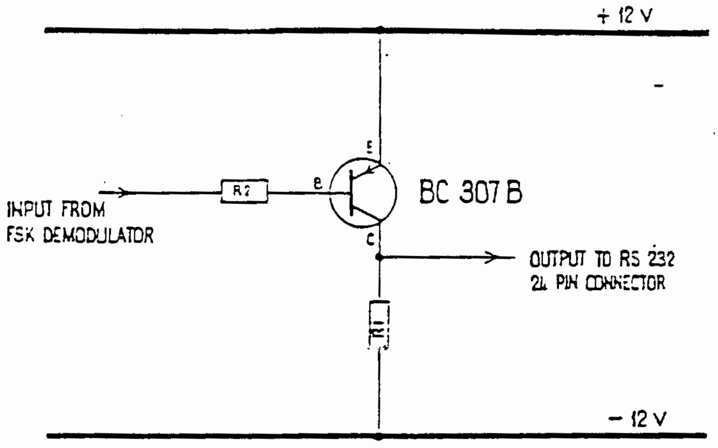
The demodulator has been designed to detect the standard FSK frequencies discussed in paragraph 7.2.12 (design of the FSK modem). These frequencies are : f_l = 1070 Hz
f_h = 1270 Hz

The manufacturer recommends the following component values for this frequency band:

FSK BAND			XR 2211						
Baud Rate	F_l	F_h	R4	R4	R5	C1	C2	Cf	Cd
300	1070	1270	10	18	100	0,039	0,01	0,005	0,05

Units: Frequency - Hz
Resistors - kOhm
Capacitors - μF

The output of the demodulator is a square wave bit stream varying between 0 and 12 V. The RS 232 interface standard requires that the bit stream varies in level between -12 V and +12 V. For this reason the following switching circuit was inserted between the demodulator output and the RS 232 interface plug.



SWITCHING CIRCUIT

Fig. 28

A standard BC 307 B transistor was selected for this circuit.

Choose Ic = 1 mA

R = V/I = 12V/1mA = 12 kOhm

Select R = 10 kOhm

for BC 307 B h_{fe} = 200

$$I = Ic/hfe = \frac{1x10-3}{200} = 0,5 \times 10^{-5}\ A$$

The input voltage is 0 - 12V. At least 0,7 V is needed to switch the transistor on

thus the voltage drop across R_2 must be 11,3 V

thus R_2 can be calculated as follows:

$$R_2 = \frac{11,3}{0,5 \times 10^{-5}}$$

$$= 2,26 \times 10^6\ \text{Ohm}$$

$$= 2260\ \text{kOhm}$$

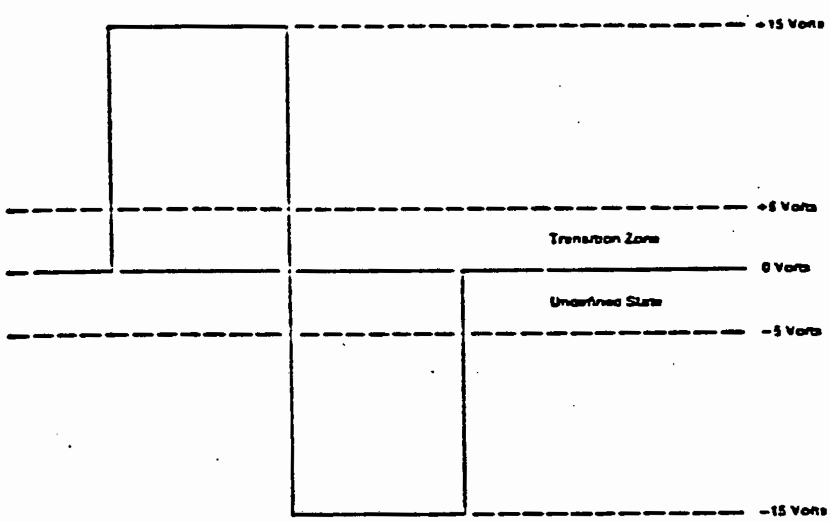
In order to ensure the transistor switches on make the resistor ten times smaller let R_2 = 270 kOhm.

7.3.2 RS 232 Standard Interface

Voltage Levels

A 24 pin connector is used to connect the output of the demodulator to the cable which is connected to the serial port of the microcomputer.

CCITT recommends that none of the pin connector voltages must be lower than + or - 3 volts or more than + or - 15 volts as shown in fig. 29. In this design the majority of pins are unused as no handshaking occurs. It is also recommended that the maximum length of cable between pieces of equipment must not exceed 15 m. The characteristics of the cable affects the speed and timing of the signals. In this design the length of cable is less than 1 meter.



RS 232 INTERFACE VOLTAGE LEVELS

Fig. 29

Pin Connections

Please refer to Appendix 6 for a complete listing of all 24 pins of the RS 232 connector.

Only the pins that are relevant in this design are discussed here. The connections of pins are as follows:

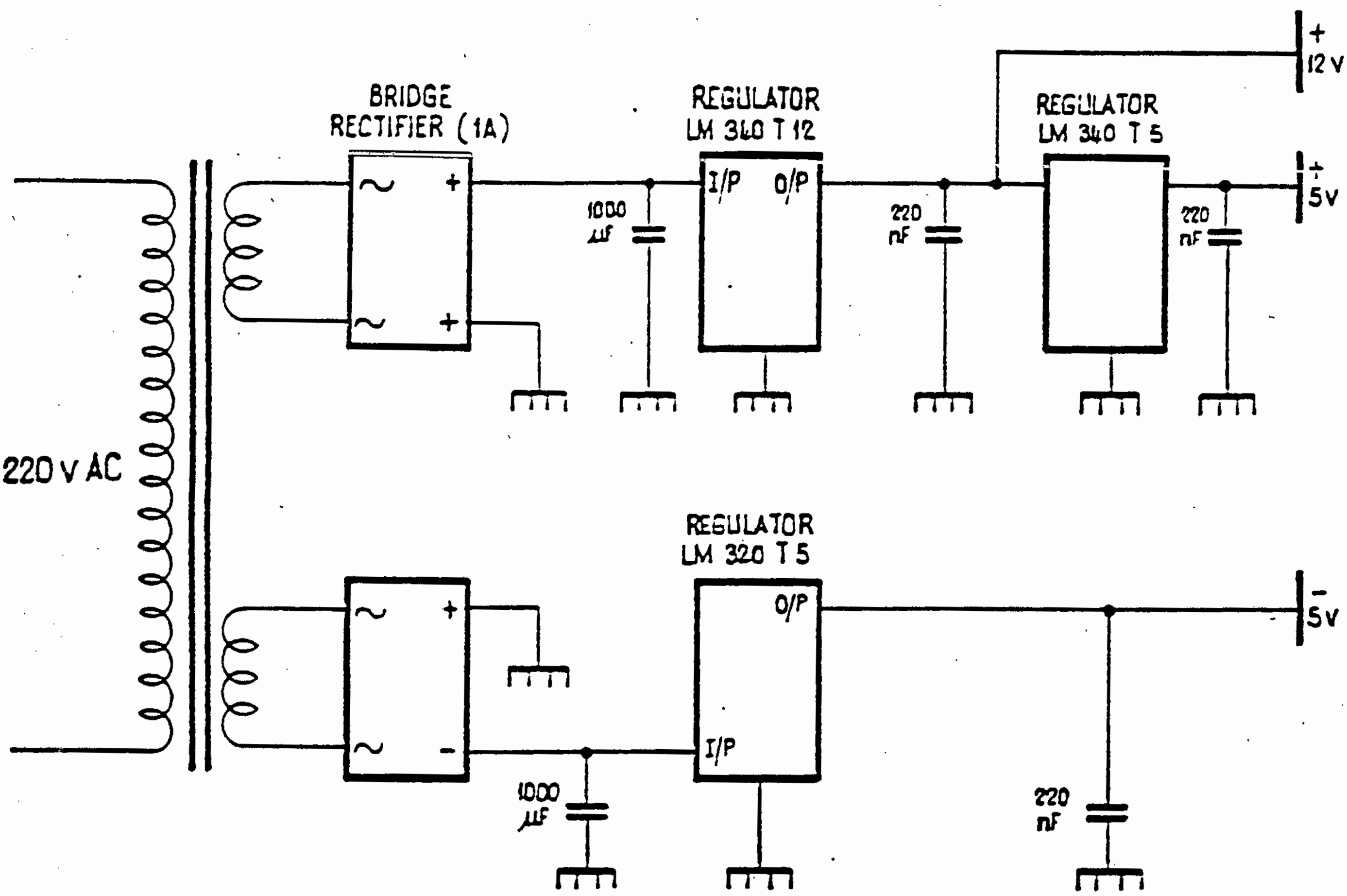
PIN NUMBER	FUNCTION	CONNECTION
3	Received data	To output of demodulator
7	Signal ground	To system ground
4	Request to send	To pin 5
5	Clear to send	To pin 4

Pins 4 and 5 are connected together as indicated by the above table.

The microcomputer transmits a 'request to send' signal to the peripheral. The computer then awaits a signal 'clear to send' before data can be received.

Since the peripheral in this case is not intelligent and cannot provide the required signal, pins 4 and 5 are connected together and the 'request to send' signal is sent back to the microcomputer as the 'clear to send' signal. The system is now ready to receive data.

7.4 Power Supply Design



CIRCUIT DIAGRAM OF POWER SUPPLY AT OUTSTATION

Fig. 30

The circuit diagram for the power supply used at each outstation is shown in figure 30. The following voltages need to be supplied:

+5 V

+12 V

-5 V

A multiturn 220V - 12V 1A transformer is used to transform the mains voltage to 12V AC. One ampere is more than sufficient to cater for the system requirements as CMOS is used extensively. Referring to the top half of the diagram this AC voltage is passed through a 1 amp bridge rectifier which results in a 12 V dc voltage.

A 1000 μF capacitor to ground is included at this stage to minimize ripple effects. The DC voltage is then applied to a LM 340 T12 regulator connected as shown in the diagram. This results in a +12 V DC output which is used by the system. A 220 nF capacitor is connected between the output line and ground for decoupling purposes.

The 12 V DC output is also connected through to a LM 340 T5 regulator which results in a +5 V DC output which is required by the system. A 220 nF capacitor is included for decoupling purposes.

Referring to the bottom half of fig. 30 the AC voltage is passed though a 1 A bridge rectifier as shown. In this case a negative voltage is required hence the negative output of the rectifier is connected to the LM 320 T5 regulator which produces -5 V output which is required by the system. Capacitors are included in this section of the circuit for the same reasons given earlier.

7.5 Hardware Testing

Individual sections of the circuits were tested at the breadboard level to ensure that the design work was correct. A variable DC power supply, high resolution Multimeter and storage oscilloscope were used for this purpose. Corrections of component values were made at this stage.

The outstation was then implemented on Veroboard where individual tests and finally a system test was performed. The accuracy of the system was found to be better than 2%. This excludes the 1% accuracy of the transducers. Accuracy and correctness of operation was obtained by using known current inputs and producing processed tables of data.

Correctness of operation up to the A/D converter was checked by using 8 identical known constant inputs and checking the output on the A/D converter.

The design was then implemented on a P.C. Board and the testing process repeated. Problems were constantly encounted due to lack of contact between IC pins and sockets. This problem continued to the Field Testing stage.

8.0 DATA INPUT PROGRAM

Overview

This section provides a brief program specification followed by a paragraph on user instructions. A conceptual overview follows which outlines the reasons for using this particular approach in developing the program. System constraints and requirements which had an influence on the program design are outlined in a separate paragraph. The design is then explained with the aid of Jackson charts. Finally validation procedures are discussed. The program listing is contained in Appendix 7.

8.1 Program Specification

This program forms part of the Load Monitoring System which has been developed for the Central Distribution Area of Escom.

At the master station the recorded load data must be read from the cassette tape through a demodulator into the Zenith microcomputer ready for processing. The main purpose of this program is to scan the serial port of the microcomputer, manipulate the data in memory and store the resultant file on a floppy disc. This file can then be accessed by the Lotus 1-2-3 package for further processing.

8.2 Instructions for use

8.2.1 Input Specifications

(1) User input - The first screen which appears when the program is run prompts the user to enter substation name; and start/end date and time information for data. Validation of input data is done and the following error messages may appear:

- Typing error or substation not on list
- Numeric expression out of bounds (relating to date/time input)

The substation name must be one of a list of 10 predefined substations names. With respect to date and time information, checks are done on the day, month, hour and time information to check they are within predetermined bounds eg. months $\ngtr$ 12.

The user can then re-enter corrected information.

A check is done to determine whether more than 200 hours elapse between the start and finish times entered by the user. This check is done because of the maximum no of points (200) that can be plotted on a Lotus graph. If more than 200 hours elapse a message appears : "Start and End Dates not within limits".

The program then checks whether the particular file (identified by the substation's name entered) already exists and if so whether it is suitable for further use. The following messages may then appear:

(i) "Loading existing file into memory" if file is suitable for further use.

(ii) "The existing file '______' is to large to allow new information to be added to it. Please ensure you have a hard copy of the data before you continue".

(iii) "More than one fifteen minute time interval has elapsed since the last set of data. Please ensure you have a hard copy of the data before you continue".

(iv) "'______' does not exist."

In the case of (ii) - (iv) the user is asked whether he wishes to continue. If so a message appears "New file '______' will be created".

The system is now ready to receive load data.

(2) Load data input - This data is read into the serial port of the microcomputer from a cassette tape through a demodulator at 300 Baud (asynchronous). Configuration of this data is as follows:

- 1 Start bit
- 1 Stop bit
- Even Parity

Once the initial information has been provided by the user a message appears instructing the user to connect up the tape recorder to the serial port and press the space bar to prepare the system for receiving data. The tape recorder can then be switched on and data is read into the microcomputer. In order to indicate to the user that the system is operating correctly input data in the form of ASCII characters is scrolled across the screen.

8.2.2 Transmission Error Messages

Once the data has been read into the port a summary of data transmission errors which have occurred will appear in the top right hand corner of the screen. The user is asked whether he wishes to retransmit the data. This should be done if an excessive number of errors is reported.

8.2.3 Output Specifications

The user can then exercise the option of storing the load information on a disc for later processing by Lotus 1-2-3. If he chooses to do this, the program writes the data from the relevant arrays to a file according to a strictly defined structure. Lotus 1-2-3 must then be used to produce graphs and tables of the load data.

8.3 Programming Approach

A dedicated custom written software approach was adopted in this case. This was done because of the specific system requirements and constraints (8.4). No standard software package is available that will perform the required functions.

Turbopascal was chosen as the programming language because of the advantages of using a structured design, viz more efficient programming and easy interpretation of the final program. Turbopascal was chosen instead of conventional Pascal because of the additional feature of screen cursor control characters which have been used to improve the user friendliness of the system.

The program has also been well documented to aid with interpretation and maintenance.

8.4 System Constraints and Requirements

(1) Time Period of Load Readings

Graphs and tables of feeder loadings will normally be required for a specific two day period. In order to provide flexibility, however, data must also be able to be displayed continuously over a period of a week (Section 6.0).

(2) The E^2PROM mounted on the outstation board can only store 2K bytes of data ($\pm$ 2 days).

(3) The Lotus system can plot a maximum of 200 points on a graph.

These factors have a direct bearing on the program design. Since the memory device can only store 2K bytes of data it is obvious that three recordings have to be made to gather a week's data (taken at 15 min intervals). The program had to be developed so that three tapes of data could be loaded into the microcomputer consecutively and one output file obtained. Various error checks were necessary to ensure correct final data.

In addition it was necessary to make the program as user friendly as possible since the users have little computer background.

8.5 Program Design

A top-down description of the program has been given enabling the reader to understand the various levels of detail.

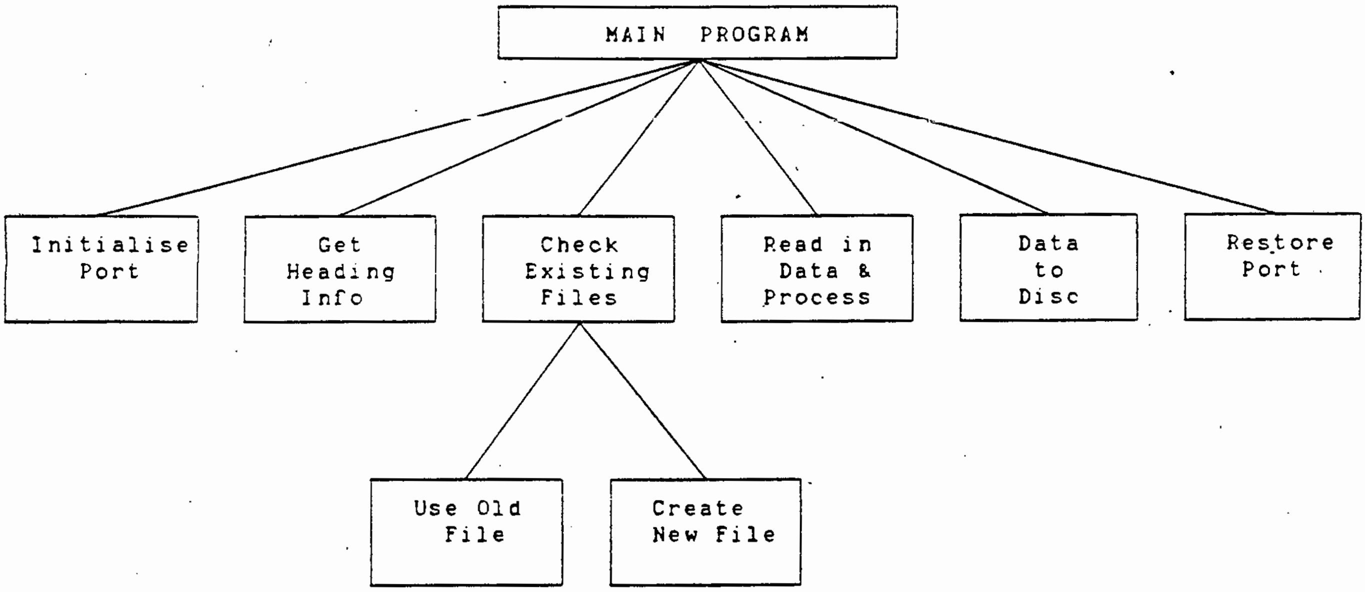
Procedures contained in the package have been demarcated into one or more of the following functional areas:

- Terminal controller
- Interface with the user
- File management section

This demarcation is useful from the point of view of program maintenance and that of understanding the program's operation.

The program design will be explained with the aid of Jackson charts.

8.5.1 Main Program



JACKSON CHART OF MAIN PROGRAM

Fig. 31

The Jackson Chart in fig. 31 indicates the procedures called by the main program. Lower levels of procedure are indicated later in the discussion of individual procedures.

Input

Input to the overall program is a 300 Bd serial data stream in RS 232 mode as well as user entered substation and date/time information.

Process

The program then manipulates the data and stores it in various arrays.

Output

The processed data is stored as an output on a disc at the option of the user. This file can be accessed by the Lotus 1-2-3 system.

8.5.2 Initialize Procedure

The serial port COM1 of the Zenith microcomputer is used to read data from the cassette tape. This procedure initializes COM1 and does an initial read of the port to clear the receive buffer.

Technical Details of Serial Port

The Zenith microcomputer uses an INTEL 8251 A serial port or USART (Universal Synchronous Asynchronous Receiver Transmitter). The 8251 A is used as a peripheral device and is programmed by the CPU to operate using virtually any serial data transmission technique currently in use.

In this particular application the USART receives a serial stream and converts it into parallel data characters for the CPU. The USART signals the CPU whenever it has received a character. The USART can operate from DC to 19,2 K Baud in Asynchronous mode.

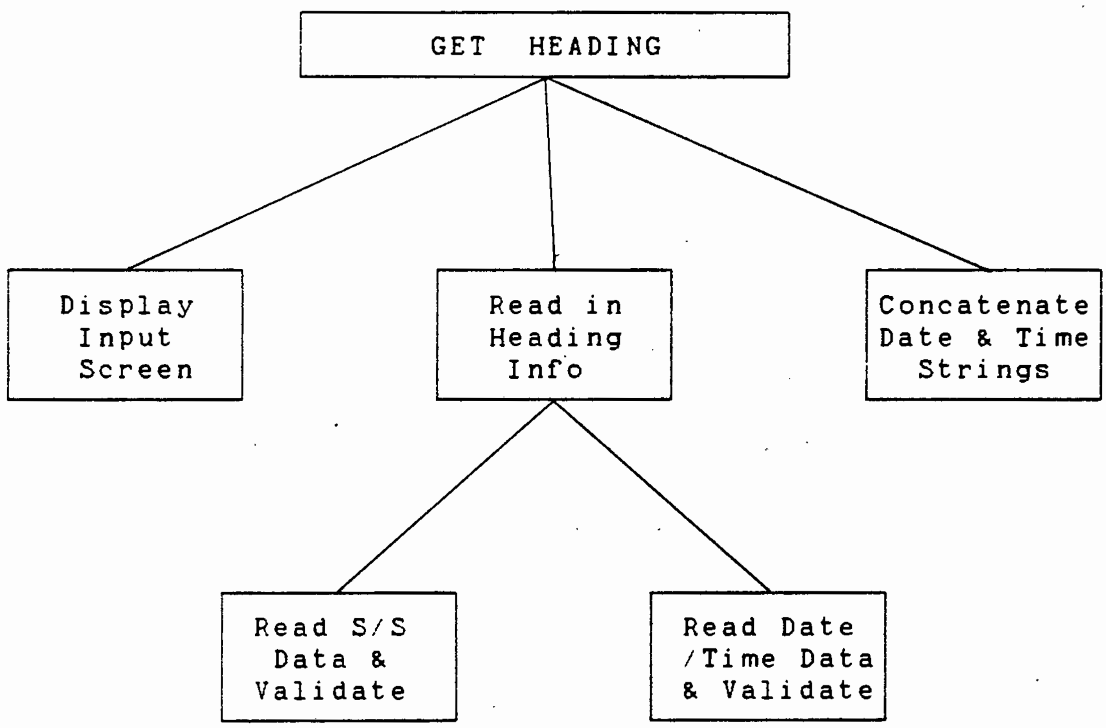
The Initialize Procedure stores the previous configuration of the serial port and sets the port to the following configuration:

- 300 Bd
- 1 Stop bit
- Even parity
- no handshaking
- 8 bit word length
- 1 start bit (set automatically)

The procedure also performs an initial read of the serial port to clear the receive buffer. This is done by using a procedure Get Serial Character which is discussed under the Get Data Procedure (8.5.5).

The Initialize Procedure clearly falls in the area of the 'Terminal Controller' part of the program as it sets up the serial port for receiving data from an external device.

8.5.3 Get Heading Procedure

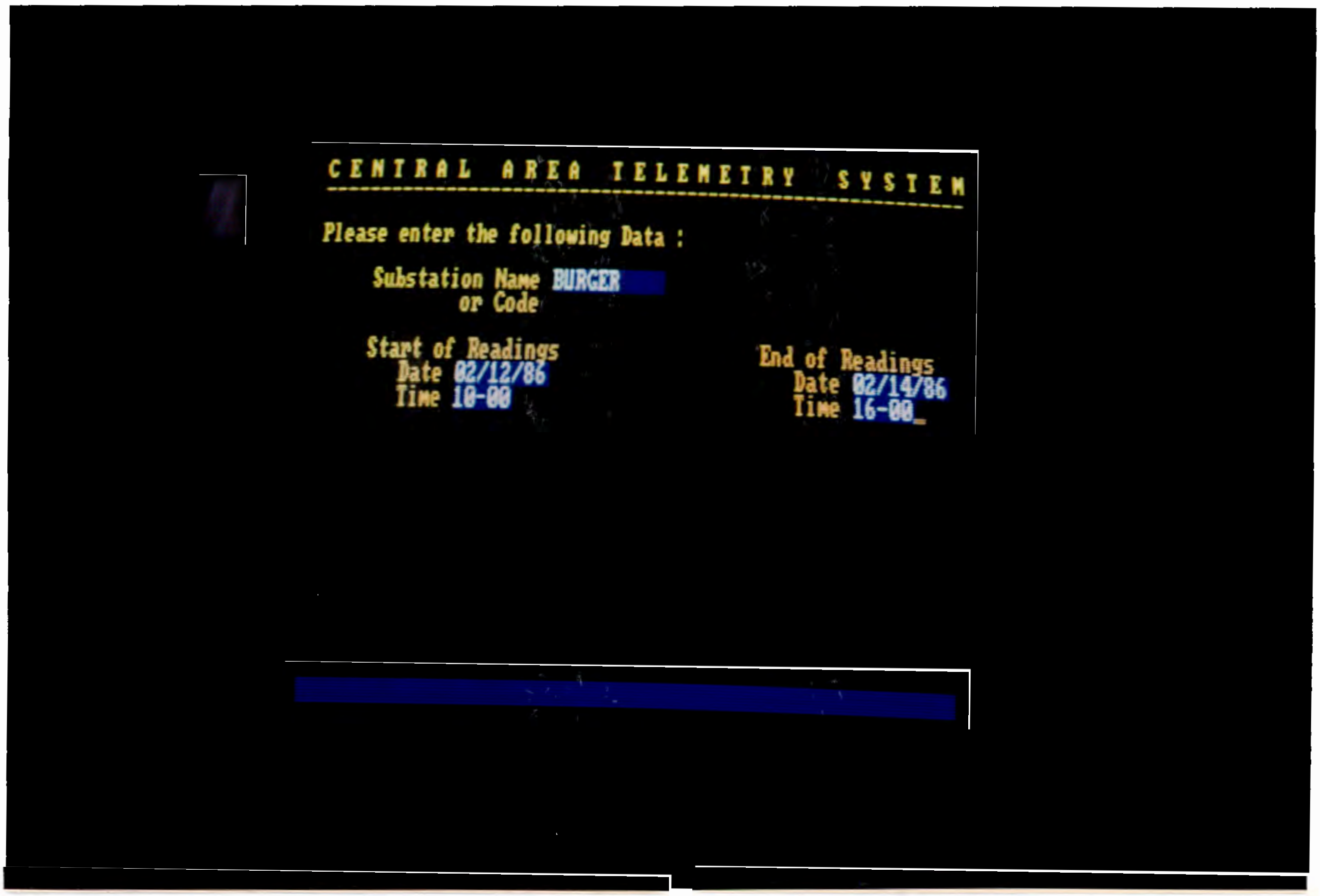


JACKSON CHART FOR GET HEADING PROCEDURE

Fig. 32

Display Input Screen

This part of the procedure provides for the display of a screen indicating the information the user is required to input. A photograph of the data input screen is shown in fig. 33.



INFORMATION INPUT SCREEN

Fig. 33

In order to make the system as user friendly as possible due attention was given to screen layout and effective use of colours. The procedure is written so that the cursor points to the field where the information is required. It is not possible to enter information elsewhere on the screen.

<u>Read in Heading Information</u>

The substation name is read in using the procedure Input String. Validation of the substation name is done using a subrange of substation names.

Date and Time information is entered using the Read 2 Digits Procedure. This procedure reads a two digit number at position (x, y). Validation of date and time information is done.

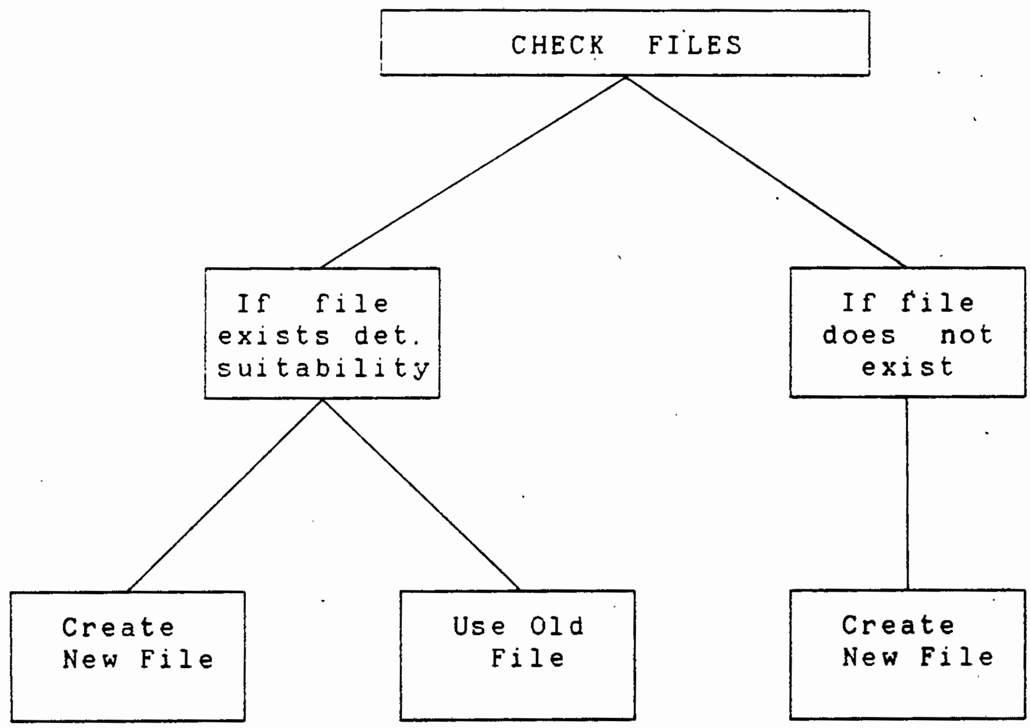
The user has limited cursor control. By means of the <u>cntrl E</u> button the data in the field just entered is erased and the cursor moves automatically to the beginning of that field.

<u>Concatenation of date and time information</u>

Substation name and date/time information is passed back to the main procedure Get Heading which concatenates the date and time strings as shown by the following example: 08h00 23/11/86. A further check is done at this stage. By consulting the Number of Readings function the procedure checks to see that no more than 200 hrs elapse between start and finish times entered by the user. 8.2.1 contains details on the error messages given.

This procedure falls into the functional area of <u>User Interface.</u>

8.5.4 Check Files Procedure



JACKSON CHART OF CHECK FILES PROCEDURE

Fig. 34

The procedure firstly checks whether a particular file exists on the disc. If it does exist the following checks are done:

(1) Determines whether there is sufficient space to add new data. There must be less than 200 records in total.

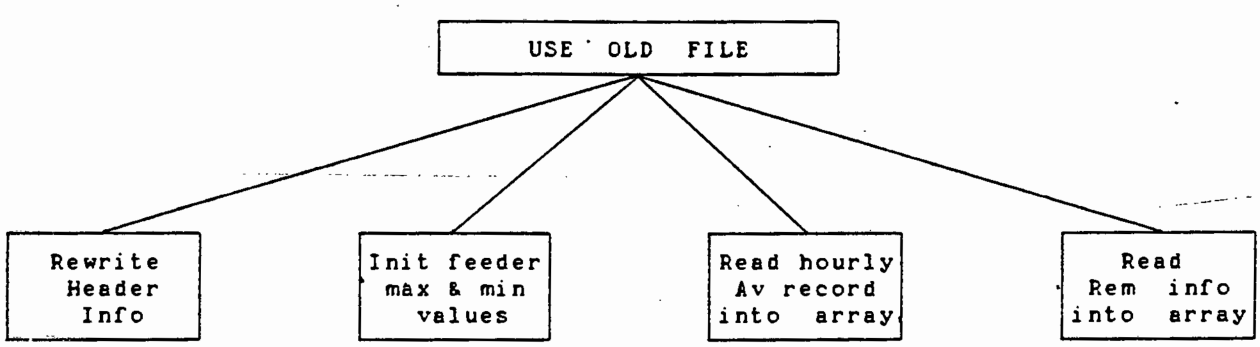
(2) Determines whether the new set of readings start within 29 minutes of the end of the previous readings. An error message is given if this limit is exceeded.

If the file is suitable for further use the variable Create Code is assigned the value 0. If the file is unsuitable or does not exist Create Code is assigned one of the following values:

Create Code = 1 if (Old Hours + Hours) > 200
= 2 if too may intervals missed
= 3 if file does not exist

Messages to this effect are given to the user.

If Create Code is set to '0' Use Old File procedure is used.



JACKSON CHART FOR USE OLD FILE PROCEDURE

Fig. 35

If the existing file is still suitable this procedure does the following:

(1) Rewrites the header information to reflect change in date and time.

(2) Initializes feeder maximum and minimum arrays at values found in existing data.

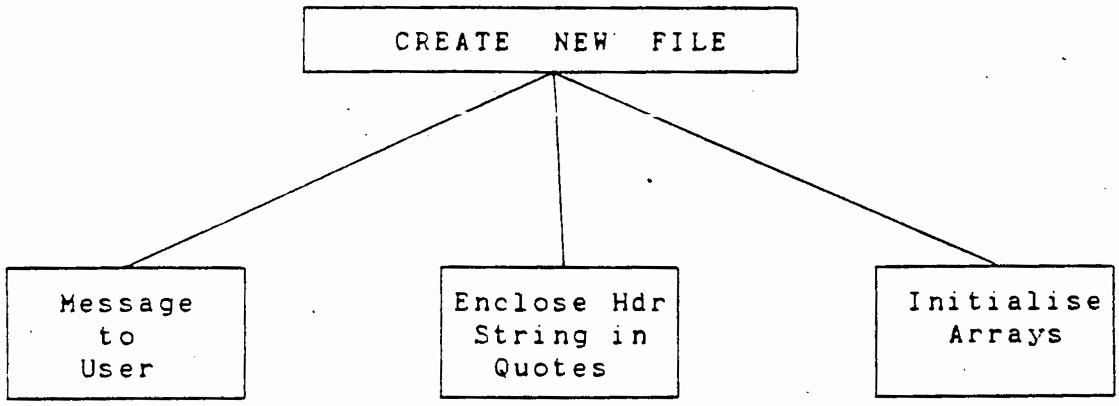
(3) Reads hourly averages of existing data into relevant arrays.

(4) Reads the remainder information into relevant arrays.

Two Procedures Values from String and Number from String are used to extract the information from the existing file. Values from String extracts the eight numbers from a text line string. Number from String extracts one number from a text string.

At this stage the system is ready to accept new data and a message appears on the screen telling the user to input the data.

If Create Code has been set to 1, 2 or 3 a new file is created using the Create New File Procedure.



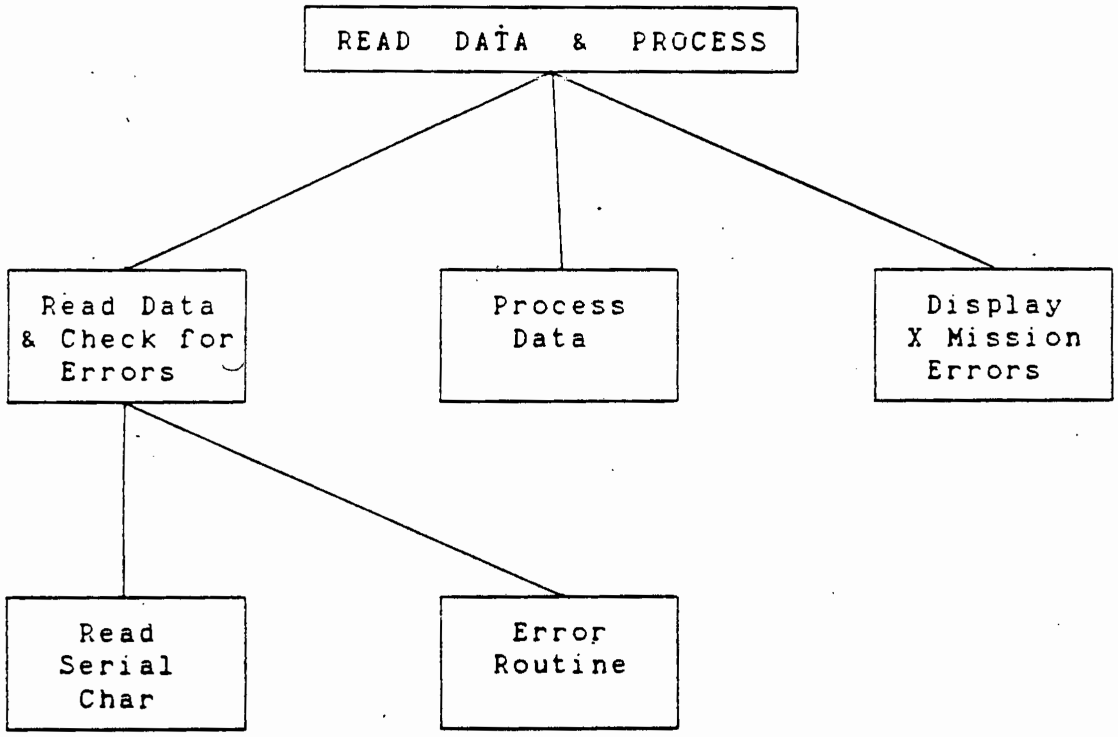
JACKSON CHART FOR CREATE NEW FILE PROCEDURE

Fig. 36

The procedure creates a new file identified by the substation name. Depending on the value of Create Code a message is given to the user (8.2.1).

The procedure also encloses the heading information in quotation marks. This is to enable Lotus 1-2-3 to recognize the heading information as text as opposed to numerics in the remainder of the file.

8.5.5 Get Data Procedure



JACKSON CHART OF GET DATA PROCEDURE

Fig. 37

This procedure reads the incoming data stream and assigns each byte to the correct array. The data is read by using the procedure Get Serial Character. This procedure reads in a character from the Zenith COM1 serial port. A timeout will occur if the port is polled 1000 times before a character is received.

Data is read into the port as follows:

The Receive Data line of the USART is normally high. A falling edge on this line triggers the beginning of a start bit. The validity of this start bit is checked by again strobing this bit at its nominal centre. If a low is detected again, it is a valid start bit, and the bit counter will start counting. The bit counter thus locates the centre of the data bits, the parity bit and the stop bit. If a parity error occurs, the parity error flag is set. If a low level is detected as the stop bit, the framing error flag is set. The stop bit signals the end of a character. This character is then loaded into the I/O buffer of the USART. The Get Serial Character Procedure scans the port and the character is read into memory.

If a previous character has not been fetched by the CPU, the present character replaces it in the I/O buffer and the Overrun Error flag is raised. The occurrence of any of these errors does not affect the operation of the USART.

The data speed of 300 Bd means that one character is received approximately every 33 ms. This means that the polling method is quite adequate as there is more than sufficient time for housekeeping and it is 100% certain that an Overrun Error will not occur.

If a character has been received bit 0 of the status register of the USART is set to 1. If receive characters occur the corresponding bit in the status register is set to 1.

If an error has occurred the main procedure passes the information to the <u>Error Handler</u> Procedure which sums the various classes of error.

The main procedure processes the data by calculating hourly averages of loadings for each feeder. The maximum and minimum loadings are also determined. Ideally it would have been preferable to store and display each reading graphically and in a table. Lotus 1-2-3 imposes a restriction, however, since only 200 points can be displayed on a graph. It is clearly impossible to display one week's readings graphically if no averaging is done. Willis and Vismor (12) have stated that a valid sampling rate for a large number of consumers can extend to 1 hour. For this reason it was decided that an hourly average of 15 min readings would not distort the trend significantly while allowing for the flexibility of displaying up to 1 week's data. It is also important to note that this averaging does not affect the maximum and minimum values which are stored in the output file. The scan of these values is done before any averaging takes place. In many cases the critical value in a set of readings is the maximum value as this determines the degree of overload on the system. The resultant data is stored in arrays.

If errors occurred during transmission the Error Message Procedure displays the total number of errors in the top right hand section of the screen. The user is given the option of retransmitting the data.

The Get Data Procedure contains aspects of terminal control and file handling.

8.5.6 Data to Disc Procedure

At the instruction of the user this procedure writes the data to a file which is stored on a floppy disc. Each set of readings is stored as a single file which has the following structure:

- A heading record containing the substation name, and start/end time and date of the readings.

- One record for the maximum and another for the minimum loading for each feeder over the period described in the heading record.

- The records containing the hourly averages of the data recordings, headed by the number of hourly averages.

- The last two records, representing the last part of an hour just prior to the termination of the sum of readings viz. the number of 15 minute intervals and the sum of loadings for the remaining period.

The file is of type Text because more than one different type of record is stored. In order to make the file compatible with Lotus 1-2-3 the actual text (as opposed to the numerics) is enclosed in double quotation marks by this procedure.

This procedure falls in the area of <u>File Management</u>.

8.5.7 <u>Restore Serial Procedure</u>

The procedure restores the serial port COM1 of the computer to the way it was configured prior to running the Initialize procedure.

8.6 <u>Program Validation</u>

The program was validated as a whole by using known input data and checking the correctness of the output. The following types of load data were used:

(1) Constant and equal loadings for each of 8 feeders.

(2) Ramp data for each of 8 feeders.

(3) Constant and equal loadings for certain of the feeders, the remaining having a zero load.

The resultant output files were used as input to the macro driven Lotus 1-2-3 system to produces graphs and tables of data. In each case the final output matched the known input within $\pm$ 2%. This inaccuracy is due to the hardware of the data gathering system.

Incorrect substation name and date/time information were used as input to check the validation procedures. Correct error messages were displayed each time.

Three sets of load data were used as input to simulate a week's readings. The Check files procedure was checked by entering different date/time information. In each case a new file was created if more than 29 minutes elapsed between sets of data.

A time gap was introduced in a set of data and the required "timeout" error was displayed.

A particular problem was the generation of errors by the occurrence of noise before the transmission of data commenced. A possible solution would be to include hardware filtering of the FSK signal before it is demodulated.

9.0 DEVELOPMENT OF A MACRO DRIVEN LOTUS MODEL

Overview

A brief background of the Lotus package is given. The main emphasis is on the development of a macro driven Lotus model to produce graphs and tables of load data.

9.1 Background to Lotus 1-2-3

Lotus 1-2-3 is a spreadsheet package which is particularly useful for producing graphs and tables of information. It is only possible to use this package on IBM compatible microcomputers. This was one reason for deciding to use a Zenith microcomputer for this project.

Menus & Macros

Extensive use is made of menus defined by the Lotus 1-2-3 system to manipulate data. In order to display data in the form of graphs and tables the user must have a knowledge of the Lotus system as several steps are necessary before reaching the required format.

Lotus 1-2-3 has a facility that allows the programmer to store sequences of keystrokes for future uses. A stored sequence of keystrokes is often called a keyboard macro. Since the users of the Load Monitoring system have little knowledge of the Lotus system, keyboard macros have been developed to automate the process as far as possible. The development of these macros is discussed in detail in the remainder of this section. A master worksheet file has also been developed as part of the model to ensure that the final system is easy to use.

9.2 Data files used as input to the Lotus Model

The "Data Input Software" package discussed in section 8.0 produces an output file which is usable by Lotus 1-2-3. An example of such a data file is given below:

```
 10 "BURGER" "08/07/85" "08H00" "08/07/85" "22H00"   Header
146 135 165  51  66 120   0   0                      Max Reading
 79  73  72  20  57  51   0   0                      Min Reading
 15                                                  No of hours
 94  87  94  29  65  68   0   0
 91  88  91  31  68  68   0   0
 95  88  92  29  70  68   0   0
 91  86  91  32  68  65   0   0
 94  86  92  32  65  65   0   0
 86  77  79  27  67  59   0   0
 80  74  73  22  67  53   0   0
 79  74  73  22  72  52   0   0                      Hourly av readings on each feeder
 84  78  81  24  67  60   0   0
106  97 109  35  65  81   0   0
136 125 150  47  65 109   0   0
138 127 156  44  63 113   0   0
129 116 136  40  61 104   0   0
130 120 145  42  60 106   0   0
120 111 124  36  63  94   0   0
  0                                                  No of intervals after completion of last full hour
  0   0   0   0   0   0   0   0                      Sum of readings taken during remaining intervals
```

EXAMPLE OF A DATA FILE

Fig. 38

Line 1 - contains file header information ie. name of substation and start and finish times of data reading.

Line 2 - contains maximum load readings on the particular feeder (each column represents a feeder (1-8)).

Line 3 - contains minimum load reading on the particular feeder.

Line 4 - number of hours between start and finish times.

Lines 5 - 26 hourly averages for feeder loadings.

Line 27 - number of 15 minute intervals remaining after the last full hour.

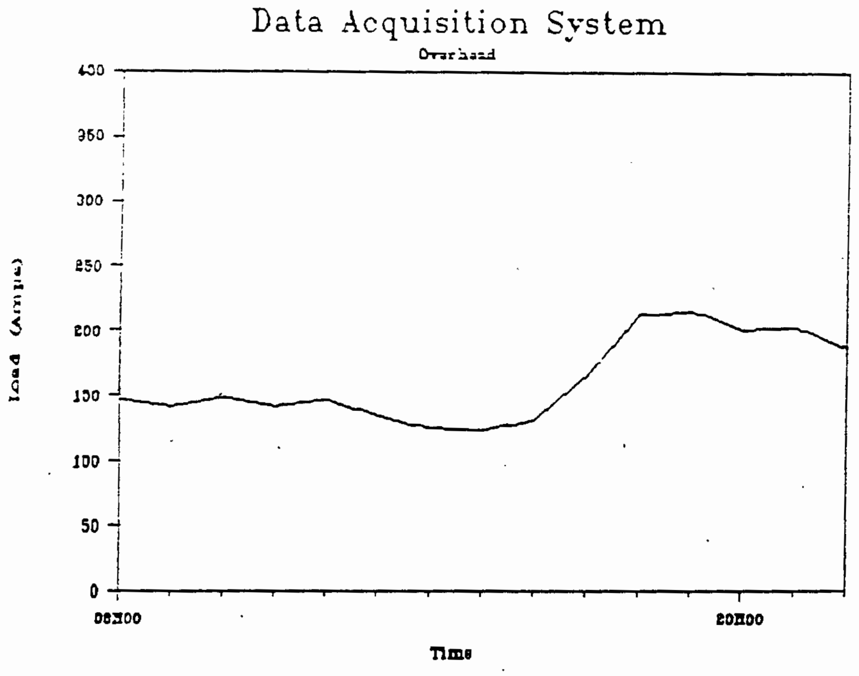
Line 28 - the sum of 15 minute readings over the remaining 15 minute intervals for each feeder.

This type of data file is referred to as a "Print" file (abbreviated to file name. PRN)

Note that this is a data file used as input to Lotus 1-2-3 and is not a worksheet file. The print file is "imported" into Lotus and the data is then manipulated by the Lotus system to produce tables and graphs.

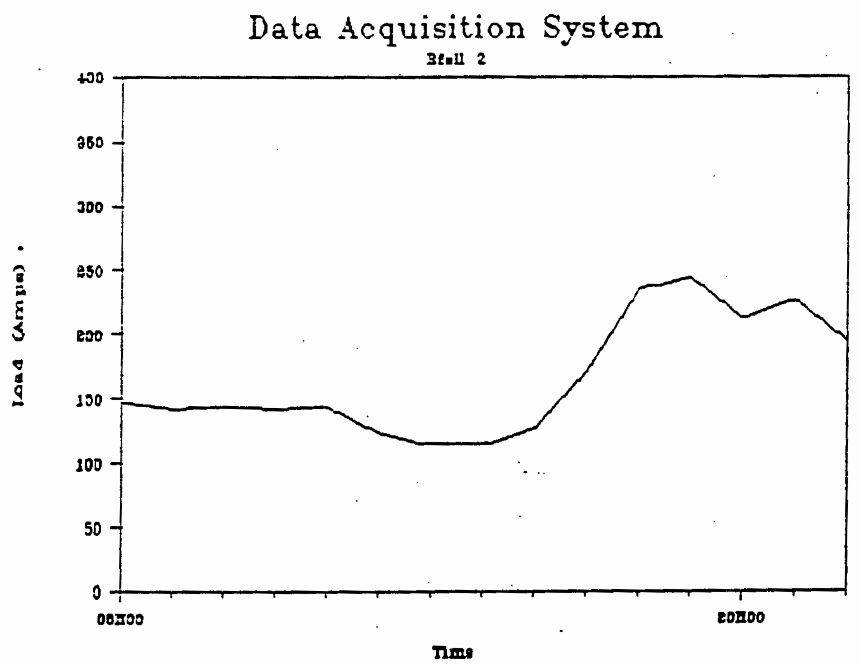
9.3 Production of Graphs and Tables

To illustrate to the reader how the data is processed and displayed by the Lotus Model a representative set of tables and graphs is shown in figures 39 to 41. Field test data was used for the production of these particular graphs and tables.



GRAPH OF LOAD VS TIME

Fig. 39



GRAPH OF LOAD VS TIME

Fig. 40

Graphs

As can be seen in figure 39 the graph illustrates load (y-axis) versus time (x-axis). The load is indicated in amperes up to a maximum of 400 A in this case. The maximum value of load is determined from the CT ratio - 400:5 in this case. It is not possible for the load to exceed 400 A on the feeder.

The feeder name is printed at the top of the graph. Due to constraints of the system it would involve unnecessary complications to print the substation name on the graph. This will have to be added by the user.

Actual times are printed on the x axis at every 12 intervals. This has been done to avoid crowding. The addition of the dates on the x axis would have involved unnecessary complication to the macros, hence the user must write in this information once the graph is printed. The system will adjust the time axis to allow for differing quantities of data (eg for 1 day, 1 week etc).

The production of load versus time graphs is the main objective of the system. This information will assist the planning and efficient operation of the reticulation network.

Tables

It is also possible to produce tables of numerical data as shown in figure 41. This table not only shows the numerical version of the data produced in the graphs but also indicates the maximum and minimum loadings on the feeders during the time period under consideration.

Time		BURGER 08/07/85		08H00 08/07/85		22H00		
	229	212	259	100	107	235	0	0
	124	115	113	39	89	100	0	0
08H00	147	136	147	57	102	133	0	0
09H00	143	138	143	61	107	133	0	0
10H00	149	138	144	57	110	133	0	0
11H00	143	135	143	63	107	127	0	0
12H00	147	135	144	63	102	127	0	0
13H00	135	121	124	53	105	116	0	0
14H00	125	116	115	43	105	104	0	0
15H00	124	116	115	43	113	102	0	0
16H00	132	122	127	47	105	118	0	0
17H00	166	152	171	69	102	159	0	0
18H00	213	196	235	92	102	214	0	0
19H00	216	199	245	86	99	222	0	0
20H00	202	182	213	78	96	204	0	0
21H00	204	188	227	82	94	208	0	0
22H00	188	174	195	71	99	184	0	0

<u>TABLE OF LOAD DATA</u>

Fig. 41

The hourly intervals are shown on the left hand side of the table while the substation name and start and finish dates and times of gathering data are shown in the heading.

The maximum feeder loadings are shown in line 2 while the minimum feeder loadings are shown in line 3. To label these quantities would have caused unnecessary complications to the worksheet. The rest of the data in the table represent the hourly averages of feeder loadings during the time period.

The user was happy to accept the table in its present form without further labelling. The labelling could possibly be introduced at a later stage if this proves desirable.

9.4 Development of the Model

9.4.1 Worksheet Details

All the formulae and macros required to process the data are contained in the "Master Worksheet". The Master Worksheet file (known as Master) is stored on the hard disc of the Zenith microcomputer. The Master Worksheet contains no specific information about any particular substation.

The Master Worksheet is used to create specific worksheets for each substation on the system. These worksheets contain the following specific information about a particular substation:

1. Substation name
2. Names of feeders
3. CT ratios corresponding to each feeder

The substation worksheet file is identified by the name of the substation (eg Burger).

The substation worksheet will contain the current load data which was last imported from a data file. When a new data file is read into the worksheet file, the old data is erased. Substation worksheet files are also stored on the hard disc of the microcomputer as there will be a maximum of 10 files in the system. Data files on the other hand are stored on floppy discs.

An explanation of the way in which data is imported into the worksheet file and then processed is given in the next paragraph.

9.4.2 Macro Instruction Details

As explained earlier in this section macros are a series of keystrokes which are executed automatically by the microcomputer when instructed to do so. This system makes extensive use of macros to ensure maximum user friendliness.

The macros used in this system are as follows:

(1) \O : Auto-execute Macro - creates substation worksheets from master worksheet.

(2) \I : Imports data and performs calculations.

(3) \H : Calculates times associated with each set of readings.

(4) \P : Prints tables of load data.

(5) \S : Saves file on disc.

(6) \G : Produces graphs from load data.

The macros are contained on both the master worksheet and the various substation worksheets, although in slightly differing forms.

Appendix 8 contains a listing of each of the macros described above. While the exact details of each macro will not be described, a flow chart and explanation of the logic of each macro will be given.

9.4.3 Auto Execute Macro \0

Please refer to figure 42 for a flowchart of this macro. Whenever a worksheet file is retrieved this macro executes automatically. There are two situations where this macro operates:

(1) Creating a new substation worksheet file.

Once a substation worksheet file has been created, it is used repeatedly with different data files. Only when an additional substation is added to the system, does it become necessary to create a new substation worksheet.

In this case the 'Master' worksheet file is retrieved.

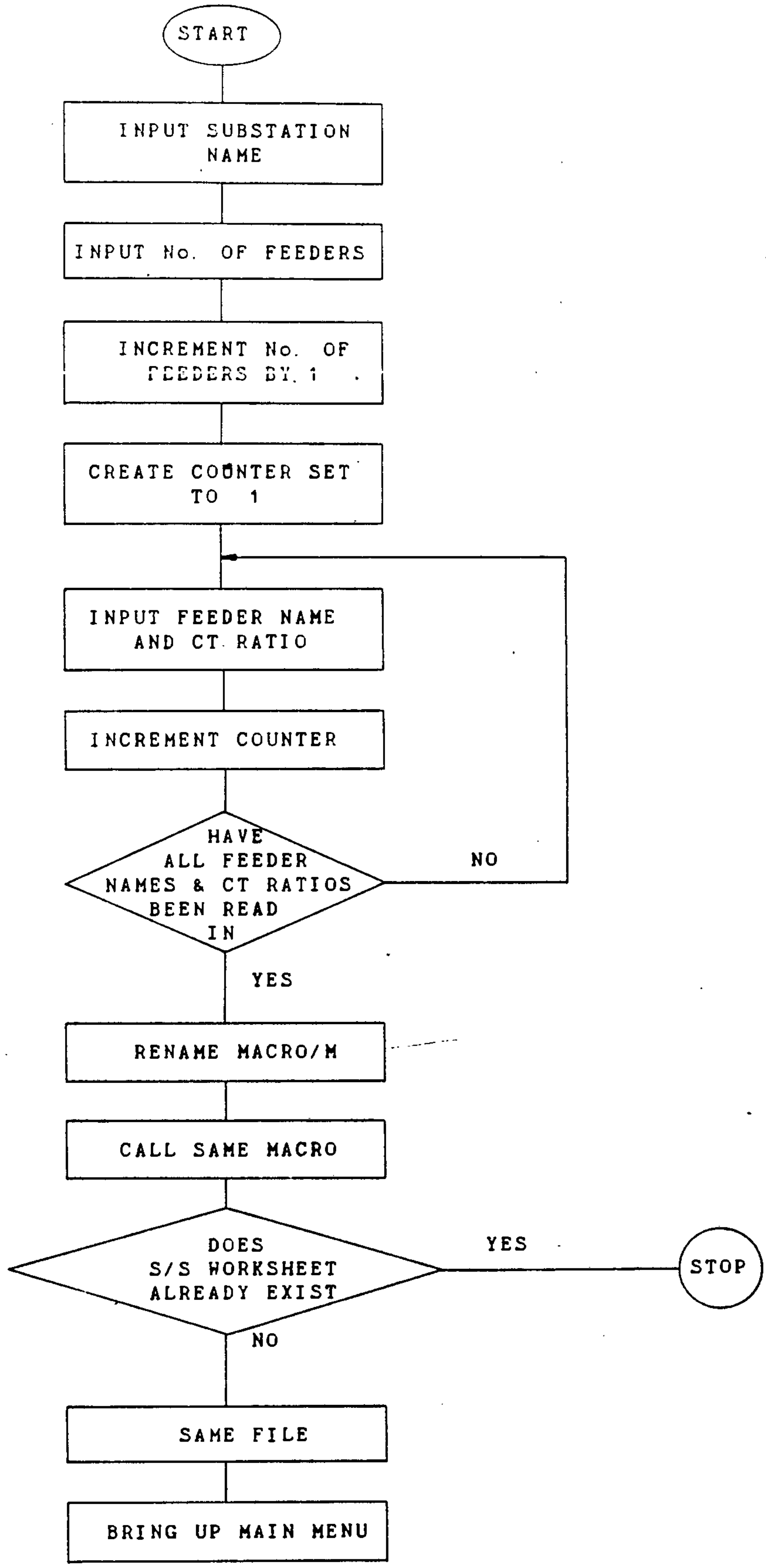
The user is prompted to input the following information:

(a) Substation name
(b) Number of feeders
(c) Feeder names
(d) Corresponding CT ratios

Once this information has been entered the macro renames itself \m which when called to execute will show the main menu of the load monitoring system

ie. Input Graph Save Quit.

This menu has been defined by the model designer and is particular to the Load Monitoring Lotus model. Once the macro has been renamed \m, a particular substation file has been defined, main menu will be displayed when the substation file is retrieved.



FLOWCHART OF AUTO EXECUTE MACRO \0

Fig. 42

The system also does a check to ensure that the substation worksheet does not already exist. If it does exist the macro ceases to operate. If the file does not exist, it is saved in the normal way. Please refer to Appendix 8 for a listing of the \0 macro.

(2) <u>Using an existing substation worksheet file to process new data</u>

In this case the \0 macro has been modified to immediately display the main menu. The user can then import a new data file and obtain graphs and tables of the data.

9.4.4 <u>Import Option on Main Menu</u>

This option involves two macros:

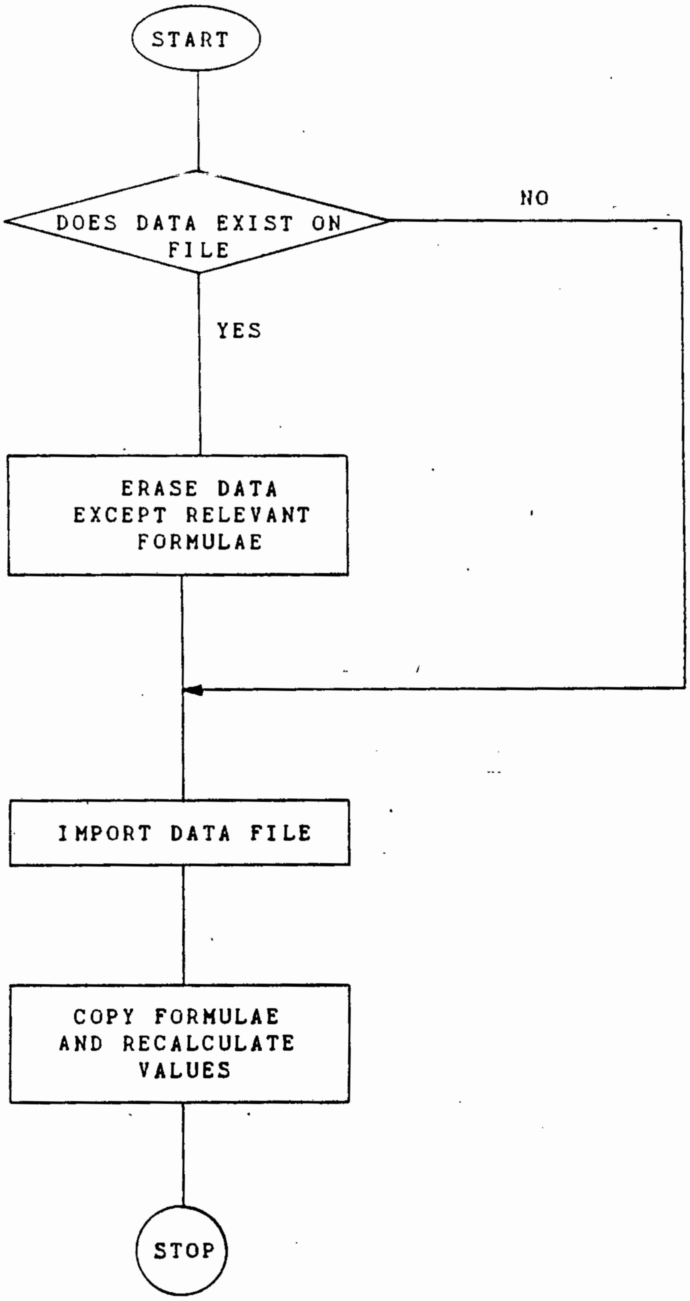
(1) <u>\I - Import Macro</u>

Please refer to figure 43 for a flowchart of this macro. Once a worksheet file has been retrieved the macro checks to see whether data is contained in the file. The macro erases all existing data if it exists.

The new data file is then imported and the worksheet is recalculated.

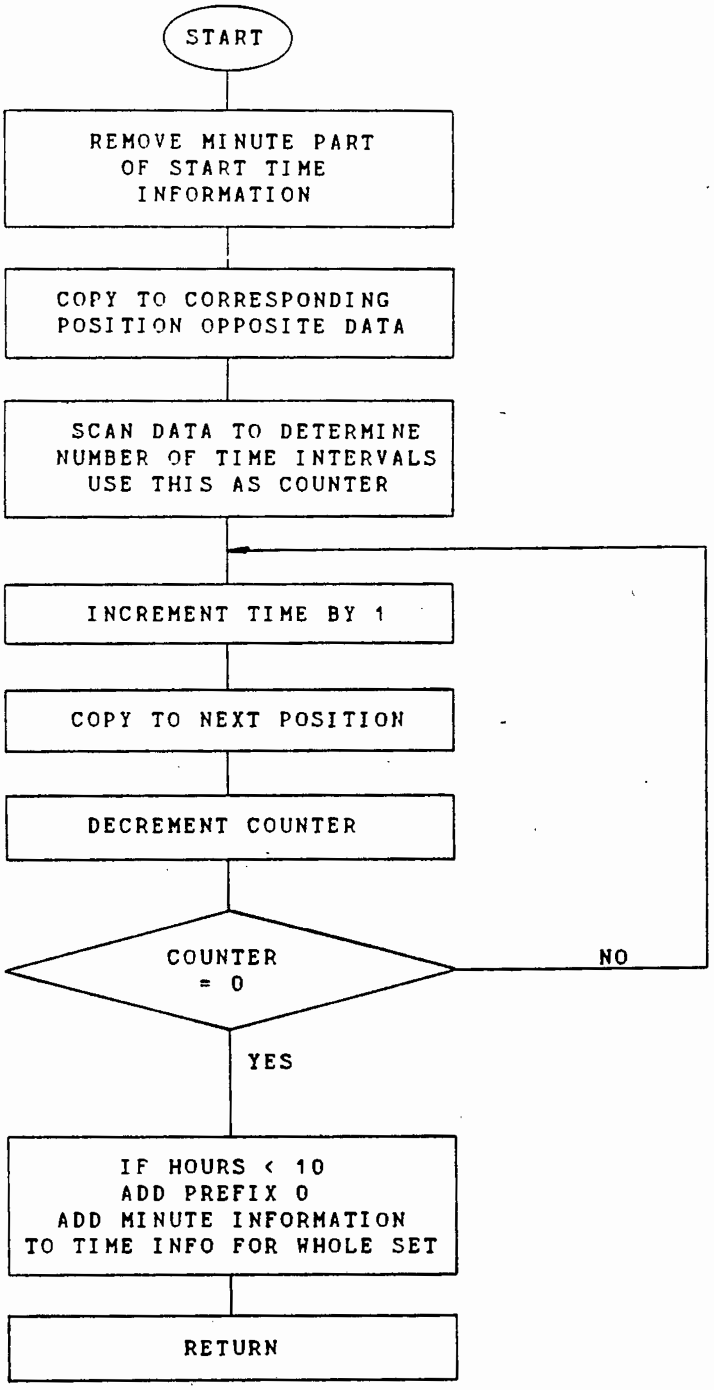
(2) <u>\H - Time Allocation Macro</u>

At this stage the worksheet does not provide information of times when load data was obtained. This information is essential to the correct interpretation of the data. \H allocates time information to load figures.



FLOWCHART OF DATA INPUT SUBROUTINE \I

Fig. 43



FLOWCHART OF TIME MACRO \H

Fig. 44

Please refer to figure 44 for a flowchart of \H which executes after \I. The macro uses the start time which was read in with the data file as the starting point for allocating times. The time opposite the first set of readings is obtained from the starting time in the "Header". The macro then scans the data to determine the number of times that must be allocated. Times are then allocated in hourly intervals next to successive sets of data. This is done by incrementing the starting time by 1 at each step. If the 'hour' information is greater than 24 the next item of information is a '1'.

At this stage the macro completes the time information by returning to the first set of readings and copying the minute information as text from the heading to the hour information. In addition a '0' is added as a prefix if the hour information is less than 10.

eg. If starting time is "06h15" the third set of time information would be allocated as "8" by the first stage of the macro and then modified to "08h15" by the second stage of the macro.

Please refer to Appendix 8 for a complete listing of these macros..

9.4.5 Graph Option on Main Menu

This option involves the \G macro. Figure 45 shows the flowchart.

The system stores graph information in graph files as opposed to worksheet files. The graph file is stored as the substation name followed by the feeder number eg. Burger 2.

Firstly the macro displays the graph menu on the screen. The user then selects the feeder number. At this stage the macro clears all existing data from the graph file if it exists. The title information and x and y axis information is then provided as input to the system. The system then accesses the required data and draws the graph of load data on the screen. Examples of these graphs are shown in figures 39 and 40.

The graph data file is then saved and the macro causes the graph menu to be displayed again. The user can proceed to look at other load data graphs or press 'ESC' to return to the main menu.

The graphical information can be printed using the usual Lotus 1-2-3 printgraph facility. The user can refer to the Lotus 1-2-3 manual for details on how to accomplish this task. Note that the 'Print' option on the main menu only prints out tables of data.

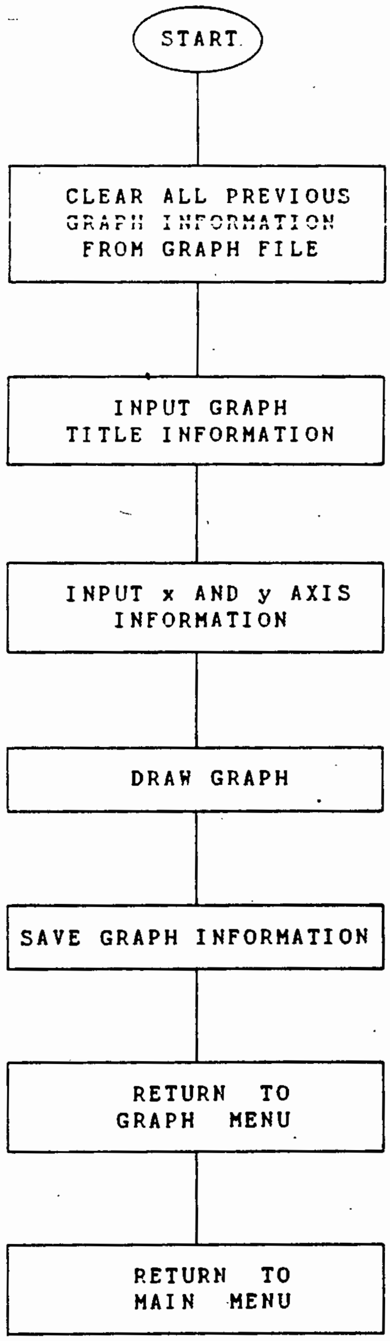
Please refer to Appendix 8 for a complete listing of this macro.

9.4.6 Print Option on Main Menu

This option involves the macro \P and provides for tables of data to be printed. Appendix 8 contains a listing of this macro. No flowchart is provided owing to simplicity.

9.4.7 Quit Option on Main Menu

This option involves the macro \Q and simply returns the user to the normal Lotus 1-2-3 worksheet.



FLOW CHART OF GRAPH MACRO

Fig. 45

9.4.8 Calculations Performed on the Worksheet

The data which is imported into the Lotus worksheet does not represent the true load readings taken. The maximum reading possible on a feeder will be represented by the number "255" which corresponds to an output of 11111111 on the A/D converter. This reading must then be converted to a "real" value by a formula on the worksheet. The formula is as follows.

$$\text{Load reading} = \frac{\text{digital equivalent}}{255} \times \text{CT ratio}$$

This formula is copied into all relevant positions on the worksheet and calculates the real value of load readings when a data file is imported.

Care has been taken in the design of the model to ensure that values are correct, hence the large number of "calc" statements in the macro's.

10.0 IMPLEMENTATION OF SYSTEM AND FIELD TESTS

Overview

This section explains how the system was assembled and mounted in a suitable housing. Details of positioning of radio receiver, outstation board and tag blocks are given. Finally Field Testing procedures are described.

10.1 Test Site

In order to do field tests on the system, a prototype outstation was installed at Burger Substation in Brackenfell. This substation was chosen for its convenient location close to Escom Test Department. There are six feeders connected to this substation namely:

1. Everite
2. Brackenfell 1
3. Overhead
4. Brackenfell 2
5. Zeeland
6. Bus Section 1

Fig. 46 shows the switchgear and ammeters for the above feeders.

SWITCHGEAR OF BURGER SUBSTATION

Fig. 46

The transducers for the load monitoring system are mounted in the rear section of the equipment shown in the photograph.

The radio set which triggers the system has been installed at the Central Control Centre, Bellville.

10.2 Outstation Details

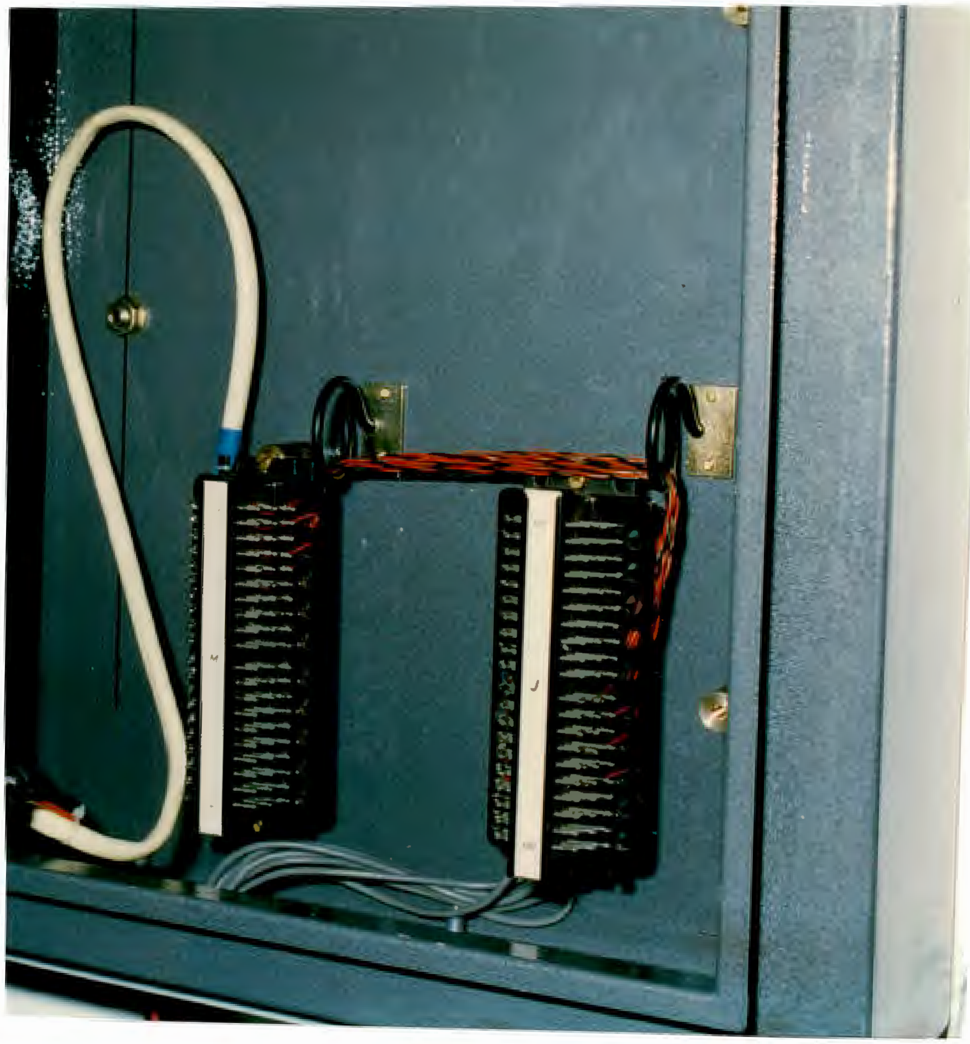
A Class B meter box was selected to house the outstation hardware. This particular box was selected for its robustness, dustproofness and convenient size.

In order to maximize space utilisation, the area inside the box has been divided by means of a hinged metal petition.

Division of the interior space was made as follows:

10.2.1 Cable Termination Tag Blocks

In order to provide maximum flexibility the tag block system of terminating cables was chosen. By changing the cross jumper connections it is possible to change the configuration of the system without difficulty. Additional feeders can also be easily added to the load monitoring system. The photograph in figure 47 shows the position of the tag blocks.



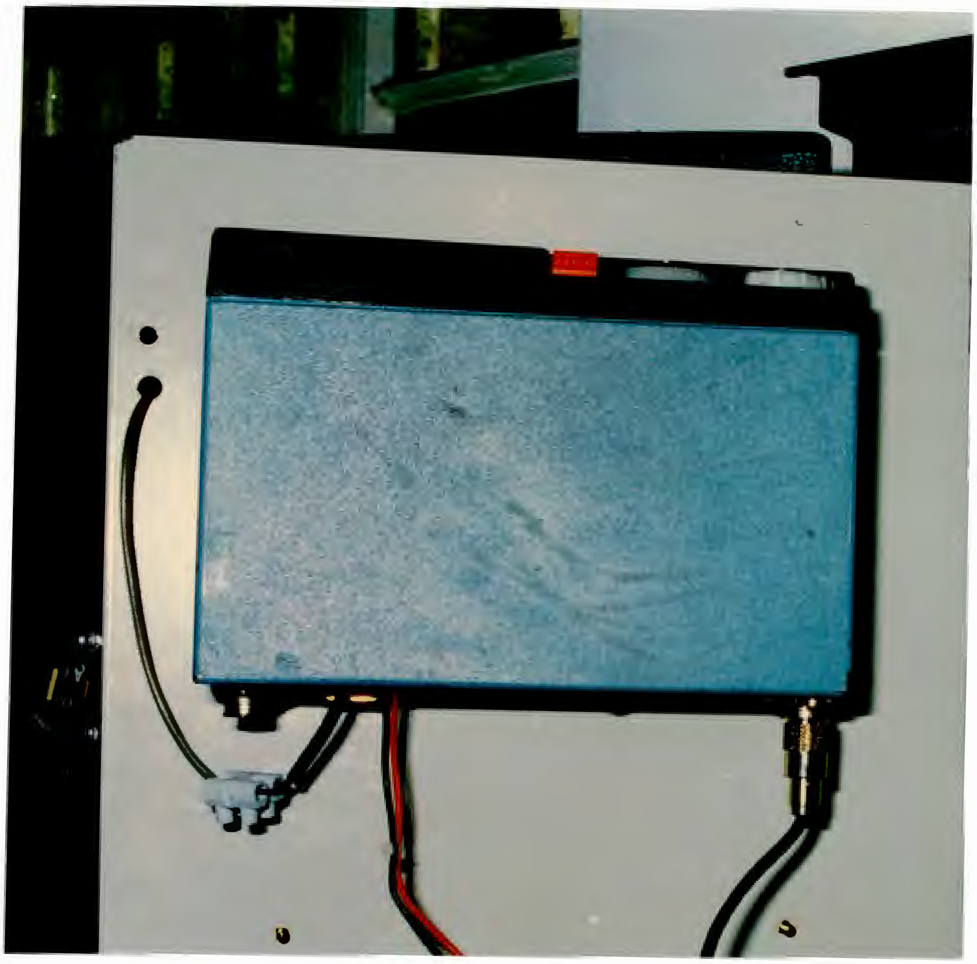
POSITIONING OF TAG BLOCKS

Fig. 47

The cable from the outstation board is terminated on the left hand side of the left hand tag block while the cables from the transducers are terminated on the left hand side of the right hand tag block. Jumper wires provide the connections between the cables. As can be seen in figure 47 two rings have been provided to help position the jumper wires. Please refer to Appendix 5 for details on tagblock connections.

10.2.2 Radio Receiver

The radio receiver which provides the triggering pulse for the system is mounted on the swing-petition by means of a bracket as shown in fig. 48.



MOUNTING OF RADIO RECEIVER

Fig. 48

The radio can easily be removed from the bracket for maintenance purposes. The power supply cable, antenna cable and the "trigger pulse" cable can be clearly seen in the photograph. 12 V supply for the Pye Olympic radio is provided by an external Motorola power supply not shown in the photograph.

10.2.3 Outstation Board

The outstation board and power supply were mounted in an aluminium housing to provide protection from dampness and dust. Figure 49 shows the mounting of the equipment on the front side of the swinging - petition.

MOUNTING OF OUTSTATION BOARD

Fig. 49

The changeover switch (for changing from 'read' to 'record' mode) and the tape recorder socket is positioned on the underside of the box. A connector plug has also been provided on the underside of the housing. This plug provides for the connection of the transducer cables onto the inputs of the outstation board. The cable and plug are shown in the photograph. The plug allows for the easy removal of the entire outstation board and power supply should this prove necessary.

10.2.4 General

A plate incorporating cable glands has been built into the bottom of the meter box. All cables enter the meter box though these glands which give a neat appearance and improve dustproofness of the system.

A test radio antenna has been positioned temporarily on the side of the substation building. When the system is finally implemented, a permanent fixture will be made.

10.3 Field Tests

The prototype unit has been field tested for approximately 2 weeks. Validation of data was done by taking meter readings at specific times while the unit was in operation.

A problem was experienced with the CD 4040 counter which addresses the E^2PROM. This was due to loss of pin contact with the socket resulting in data being allocated to the wrong memory location. Problems were also experienced with the radio receiver power supply but this was also rectified. Thereafter recorded data and measured data agreed to within 5%.

11.0 SUGGESTED IMPROVEMENTS TO THE SYSTEM

Overview

This section outlines three possible improvements which could be made to the system.

11.1 Implementation of a Selective Calling System

The Load Monitoring System will be installed at 10 strategic substations as discussed in Section 6.0. At present there is no means of selectively triggering particular outstations.

Under normal circumstances a particular outstation will only be operational for approximately 2 days at a stretch. The reasons for this are given in Section 6.0. An operator has to physically visit the outstation switch the power on and ensure the changeover switch is in the "Read Data" position to start the data scan. The reason for this is that the radio transmitter keys all units without any selection.

Implementation of a selective calling system would enable a particular outstation to be selected remotely for data scanning for a particular period. Power would be left on and the changeover switch in the "Read data" position. The system would be implemented as follows:

(i) The Zenith microcomputer used for processing data would be used for controlling the selective triggering system.

(ii) Each outstation would be equipped with a ZVEI (5 tone) decoder.

(iii) A programmable ZVEI encoder would be installed at Central Area Control Centre.

(iv) Software would be implemented on the Zenith microcomputer to do the following:

- Accept input from the user regarding which substations must be scanned and for how long.

- Access the required ZVEI codes in a look-up table.

- Send out the required ZVEI tones sequentially every 15 minutes.

- Cease sending out a particular tone when its prescribed time period has elapsed.

- Produce a printout of start and stop times for each outstation scan period.

This sytem would replace the hardware timer currently used in the radio transmitter.

Even though each outstation was powered up and ready to scan data, this would not occur until the required ZVEI code was received.

The implementation of a selective calling system would increase the flexibility of the overall system while cutting down on manpower requirements.

11.2 Installation of Battery Standby

At present the D.C. power supply to the system depends entirely on the A.C. mains supply to the substation. Should this supply fail data scanning will cease. In addition when the power is restored the counter which addresses the E^2PROM will reset causing new data to overwrite existing data.

Although the risk of supply failure is not high, this is clearly an undesirable situation which will give rise to incorrect data should a supply failure occur.

A solution would be to install a 52 V DC converter which is normally fed by the 240 V mains. In the event of a supply failure a 48 V battery bank would provide standby power to the DC/DC converter. The regulating circuit within the DC/DC converter would take up the differences in voltages and regulate the required outputs and the load monitoring system would operate as normal.

11.3 Implementation of Data Transmission using a Radio Channel

Data acquisition at the outstation is carried out on a digital basis. It would thus be possible to modify the system to enable data transmission to take place between the oustations and master station using a radio channel.

If this modification were implemented personnel would no longer have to visit outstations to collect data. This would result in a manpower saving. Considerable cost would be involved however to implement the modification. Hence it was decided to only consider implementation of data transmission over a radio channel in the long term.

12.0 CONCLUSIONS

Overview

This section summarizes the conclusions reached and assesses the extent to which research objectives have been met.

12.1 Conclusions

A literature search indicated that no previous study appears to have been done on the benefits of implementing load monitoring on 11 kV ring networks. Following an analysis of overload capability of underground cables and a critical examination of network planning and operating techniques conclusions reached were as follows:

(i) The exact effect of cyclic loading on the overload performance of underground cables is unknown at present.

(ii) Availability of load data from protected substations in areas of high load would facilitate optimization of network switching procedures. This would minimize the possibility of overloading and resultant cable damage.

(iii) The same load data would facilitate more accurate calculation of diversity factors and subsequent specification of cable sizes. This would result in cost saving.

12.2 Extent of fulfilment of research objectives

The first objective of this project was to determine whether definite benefits would be derived from implementing additional load monitoring on the network. This objective has been met in that definite benefits have been established as outlined in 12.1 above.

The second objective was to implement a suitable load monitoring system on the network. It is felt that this objective has also been adequately met with one exception. One of the secondary objectives required a data transmission via a radio link to be implemented between outstations and the control centre. At present data is transferred by means of cassette tape. This is not felt to be an optimum means of data transfer because of the inconvenience and time wastage involved in visiting the outstations in order to collect the data.

REFERENCES

1. Cotton H, The Transmission and Distribution of Electric Energy. The English Unversities Press, First Edition, 1937.

2. Weedy BM, Electric Power Systems, John Wiley, First Edition, 1967.

3. Simmons D.M, Calculation of the Electrical Problems of Underground Cables, The Electric Journal, East Pittsburgh, May - Nov 1932.

4. Neher J. and McGrath M, The Calculation of Temperature Rise and Load Capability of Cable Systems, Trans AIEE, Oct. 1957.

5. Katz C et al, Emergency Overload Characteristics of Extruded Cables, IEEE Trans PAS, Vol PAS-103, No 12, Dec 1984.

6. Goldenberg H, Calculation of Cyclic Rating Factors for Cables Laid Direct or in Ducts, Proc IEE, Vol 104 Part C, 1957.

7. Van Hoewe C et al, Overloadability of Cable Systems CIRED, 1983.

8. Sarikas R and Thacker H, Distribution System Load Characteristics and their Use in Planning and Design, Trans AIEE, Vol 76, August 1957.

9. Schroder F.S and Kwong F.C, On Line Production of Planning Load Data in Substations, CIRED 1981.

10. Realp E et al, Data Acquisition in a Distribution System using the MV Underground Network as the Transmission Path, CIRED 1981.

11. Brown J, Choosing between a Data Logger and Data Acquisition System, Inst & Control Systems, Vol 49, No 9, Sept 1976.

12. Willis H.L. et al, Some Aspects of Sampling Load Curve Data on Distribution Systems, IEEE Trans, Vol. PAS. - 104, No 11, Nov. 1985.

BIBLIOGRAPHY

1. Taylor E and Boal G, Electric Power Distribution, Edward Arnold, First Edition, 1966.

2. Sanford F, Electric Distribution Fundamentals, McGraw-Hill, First Edition, 1940.

3. Barnes C.C, Power Cables : Their Design and Installation, Chapman and Hall, First Edition, 1953.

4. Millman J, Microelectronics : Digital and Analog Systems, McGraw-Hill, First Edition, 1983.

5. Gane C and Sarson T, Structured Systems Analysis, Prentice-Hall, First Edition, 1979.

6. Jackon M.A, Principles of Program Design, Academic Press, First Edition, 1978.

7. Rabiner L.R. and Gold B, Theory and Application of Digital Signal Processing, Englewood Cliffs, First Edition, 1975.

8. Black W.Z, Short-Circuit Temperature Ratings of Buried Extruded Dielectric Cables, IEEE Trans PAS, Vol PAS-97, No 2, Apr 1978.

9. Kirn W.G and Adler R.B, A Distribution - System - Cost Model and its Application to Optimal Conductor Sizing, Trans IEEE, Vol PAS-101, No 2 Feb 1982.

10. Maletinsky V, and Schaffer G, Power System Control in Electrical Distribution Networks, Brown Boveri Review, Feb 1984.

11. Robb, A.W and Nazaroff W.W, Field Data Logger with EPROM storage, Rev Scc Inst, Vol 54, Sept 1983.

12. De Almeida A and Martins A, A Microcomputer Based System for Power Demand Monitoring and Control, CIRED 1981.

13. Bollman H and Muller K, Supervisory and Control Systems for Electrical Transmission and Distribution Networks, CIRED 1981.

14. Churuca J, Power System Load Data Acquisition and Recording, Project Report C.I.A.E., 1978.

15. CMOS Data Book.

16. Linear Data Book.

17. IBM P.C. Technical Reference Manual.

18. Zenith PC-150 Technical Manual.

19. IEC Publication 287, 1969, Calculation of Continuous Rating of Cables.

NOTE ON THE INCLUSION OF APPENDICES

Appendices 1 and 2 contain detailed information on the theory of ring distributors and performance of cables respectively. This information was located during the literature search and is related directly to the material in the main text but is not critical to the argument followed. The information has been included to provide easy reference for the reader should this information be required.

The remaining appendices contain information relating to the design and implementation of the system. Software listings have been located in the appendices as it is felt that these are not central to explaining the development of the system.

APPENDIX 1

THEORY OF RING DISTRIBUTORS

As is indicated by the name, a ring distributor is a distributor which is arranged to form a closed circuit, and it can have one or several feeding points. It is quite clear that for the purpose of calculating the voltage distribution the distributor can be considered as consisting of a series of open distributors each fed at both ends. The chief interest in the ring distributor therefore lies in the economy in copper which can be effected by its use and by the proper choice in the number of feeding points. The following treatment of the problem is due to Russell,* and in order that it may be amenable to mathematical treatment it is assumed that the distributor is in the form of a circle, that the generating station, G is in the centre, and that the feeding points, F, are arranged uniformly round the distributor as shown in Fig. A1.

Let I be the total current output of the station, and n the number of feeding points. Then the current in each feeder will be I/n and the current adjacent to

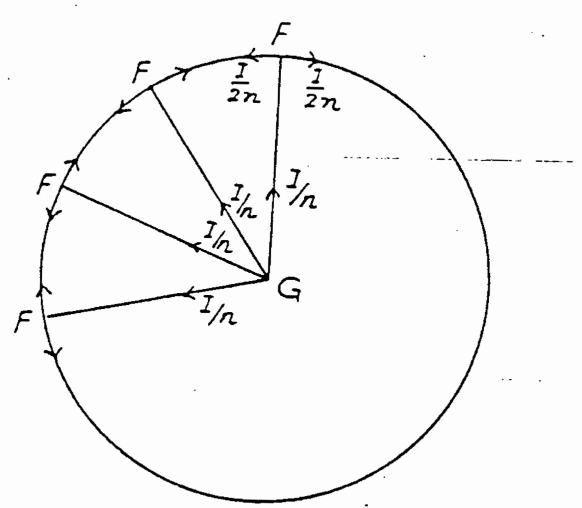


Fig. A1 - RING MAIN

each feeding point in each distributor section I/2 .

Let V_1 = volt drop along each feeder

a_f = cross-section of feeder

r = radius of circle

$$\therefore V_1 = \frac{I}{n} \times \frac{2\rho r}{a_f} \qquad \text{(reckoning both sides)}$$

$$\therefore a_f = \frac{I}{n} \times \frac{2\rho r}{V_1}$$

$$= \frac{2\rho r}{V_1 n}$$

If V is the total allowable drop for the whole system, then $(V - V_1)$ will be the drop in each distributor. Hence $(V - V_1)$

$$= \frac{1}{8} \text{ current x resistance}$$

where current = Sum of currents tapped off over one section

$$= \frac{I}{n}$$

and resistance = $\dfrac{\rho}{a_d} \times \dfrac{2\pi r}{n}$

where a_d = cross-section of a distributor

$$\therefore V - V_1 = \frac{1}{8} \times \frac{I}{n} \times \frac{2 \times 2\pi\rho r}{a_d n} \qquad \text{(reckoning both sides)}$$

$$= \frac{2\pi \rho r I}{4n^2 a_d}$$

$$\therefore a_d = \frac{2\pi \rho r I}{V_1 n}$$

Volume of copper in radial feeders

$$= 2nra_f$$

$$= 2nr \times \frac{2\rho r I}{V_1 n}$$

$$= \frac{4 I r^2 \rho}{V_1}$$

Volume of copper in ring main

$$= \quad 2 \times 2\pi r \times ad$$

$$= \quad 2 \times 2\pi r \times \frac{2\pi \rho r I}{4 n^2 (v - v_1)}$$

$$= \quad \frac{4\pi^2 r^2 \rho I}{2 n^2 (v - v_1)}$$

Total volume of copper:

$$\text{Vol.} = \frac{4 I r^2 \rho}{v_1} + \frac{4\pi^2 r^2 \rho I}{2 n^2 (v - v_1)}$$

$$\therefore \frac{d\,\text{Vol}}{d v_1} = -\frac{4 I r^2 \rho}{v_1^2} + \frac{4\pi^2 r^2 I \rho}{2 n^2 (v - v_1)}$$

This is zero and corresponds to a minimum volume

when $$\frac{4 I r^2 \rho}{v_1^2} = \frac{4\pi^2 r^2 \rho I}{2 n^2 (v - v_1)^2}$$

which reduces to $$v_1 = \frac{v \times n\sqrt{2}}{\pi + n\sqrt{2}}$$

Substituting this in the equation for the volume we have for the minimum volume corresponding to the given total drop :

$$\frac{4 I r^2 \rho (\pi + n\sqrt{2})}{v n \sqrt{2}} + \frac{4\pi^2 r^2 I \rho}{2 n^2 \left(v - \frac{v n \sqrt{2}}{\pi + n\sqrt{2}}\right)}$$

which reduces to $$\frac{4 I r^2 \rho}{v} \left(1 + \frac{\pi}{n\sqrt{2}}\right)^2$$

If n were infinite this would reduce to

$$\text{Vol}_\infty = \frac{4 I r^2 \rho}{v}$$

$$\therefore \frac{\text{Vol}}{\text{Vol}_\infty} = \left(1 + \frac{\pi}{n\sqrt{2}}\right)^2$$

The dependence of this ratio on the number n of feeding points is indicated by the following table:

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	100.
$\frac{Vol}{Vol_\infty}$	10.4	4.46	3.03	2.43	2.09	1.88	1.73	1.63	1.56	1.49	1.014

We see at once how a saving in copper in such a system can be effected by using a large number of radial feeders.

(Any page or figure references in this Appendix are to IEC Standard 287).

APPENDIX 2

CALCULATION OF CABLE LOSSES

1.0 A.C. RESISTANCE OF CONDUCTOR R

The a.c. resistance per unit length of the conductor at its maximum temperature is given by the following formula, except in the case of pipe-type cables (see Sub-clause 4.4):

$$R = R' \left(1 + y_s + y_r \right)$$

where:

R = alternating current resistance of conductor at maximum operating temperature, ohm/cm

R^1 = d.c. resistance of conductor at maximum operating temperature, ohm.cm

y_s = the skin effect factor

y_p = the proximity effect factor

1.1 D.C. Resistance of conductor R'

The d.c. resistance per unit length of the conductor at its maximum operating temperature 0 is given by:

$$R' = R_o \left[1 + a_{20} \left(\theta - 20 \right) \right]$$

where:

R_o = d.c. resistance of the conductor at 20°C, ohm/cm
The value of R shall be derived directly from IEC Publication 287, Nominal Cross-sectional Areas and Composition of Conductors of Insulated Cables.

a_{20} = constant mass temperature coefficient at 20°C per deg C and is:

0.003 93 for copper (IEC Publication 287, International Standard of Resistance for Copper)

0.004 04 for aluminium (IEC Publication 111, Recommendation for the Resistivity of Commercial Hard-drawn Aluminium Electrical Conductor Wire)

θ = maximum operating temperature, deg C.

1.2 Skin effect factor ys

The skin effect factor is given by:

$$y_s = \frac{x_s^4}{192 + 0{,}8\,x_s}$$

where:

$$x_s^2 = \frac{8\pi f}{R'} \times 10^{-9} k_s$$

f = supply frequency, Hz

Values for k_s are given in Table I.

The above formula is accurate, providing x_s does not exceed 2.8 and therefore applies to the majority of practical cases.

1.3 Proximity effect factor y_p

The proximity effect factor is given by:

$$y_p = \frac{x_p^4}{192 + 0{,}8\,x_p^4}\left(\frac{d_c}{s}\right)^2 \left[0{,}312\left(\frac{d_c}{s}\right)^2 + \frac{1{,}18}{\dfrac{x_p^4}{192 + 0{,}8\,x_p^4} + 0{,}27}\right]$$

where:

$$x_p^4 = \frac{8\pi f}{R'}\, 10^{-9} k_p$$

d_c = diameter of conductor, cm

s = distance between conductor axes, cm

In the case of multicore cables with sector-shaped conductors, the value of y_p should be multiplied by 2/3 to obtain the proximity effect factor.

$d_c = d_x$ = diameter of a circular stranded conductor of the same cross-sectional area, cm

$s = (d_x + 1)$, cm

where:

t = thickness of insulation between conductors, cm

Values for k_p are given in Table I.
The above formula is accurate, provided x_p does not exceed 2.8 and, therefore, applies to the majority of practical cases.

2.0 DIELECTRIC LOSSES W_d

These are only significant when the voltage exceeds 30 kV for paper insulated cables which is not the case in this application.

3.0 SHEATH LOSS FACTOR λ_1

The power loss in the sheath (λ_1) consists of losses caused by circulating currents (λ_1') and eddy currents (λ_1''),
thus:

$$\lambda_1 = \lambda_1' + \lambda_1''$$

The formulae given in this section express the sheath loss in terms of the total power loss in the conductors.

Three-core cables with common sheath

For a three-core cable where the cores are contained in a common metallic sheath, λ_1' is negligible and the loss factor is, therefore, given by one of the following formulae:

For round or oval conductors, and where the sheath resistance R_s is less than or equal to $1\mu\Omega$/cm:

$$\lambda_1' = \frac{3R_s}{R}\left[\left(\frac{2c}{d}\right)^2 \frac{1}{1+\left(\frac{159 R_s 10^6}{f}\right)^2} + \left(\frac{2c}{d}\right)^4 \frac{1}{1+4\left(\frac{159 R_s 10^6}{f}\right)^2}\right]$$

where:

c = distance between the axis of one conductor and the axis of the cable, cm

d = mean diameter of the sheath, cm

f = frequency, Hz

For round or oval conductors, and where the sheath resistance R_s is greater than $1 \mu\Omega$/cm:

$$\lambda_1'' = \frac{3{,}2\,\omega^2}{R R_s}\left(\frac{2c}{d}\right)^2 10^{-18}$$

For sector-shaped conductors, and R_s any value:

$$\lambda_1'' = 0{,}94\,\frac{R_s}{R}\left(\frac{2r_1+t}{d}\right)^2 \frac{1}{1+\left(\frac{159 R_s 10^6}{f}\right)^2}$$

where:

r_1 = radius of the circle circumscribing the three shaped conductors, cm

t = thickness of insulation between conductors, cm

Three-core cables with steel tape armour

The addition of steel tape armour increases the eddy-current loss in the sheath. The values for λ_1'', given in Sub-clause 6.7 should be multiplied by the following factor if the cable has steel-tape armour:

$$\left[1+\left(\frac{d}{d_A}\right)^2 \frac{1}{1+\frac{d_A}{\mu\delta}}\right]$$

where:

d = mean diameter of armour, cm

μ = relative permeability of the steel tape (usually taken as 300)

δ = equivalent thickness of armour $= \dfrac{A}{\pi d_A}$, cm

where A = cross-sectional area of the armour, cm^2

This correction is only known to be applicable to tapes 0.3 mm to 1.0 mm thick. The subject is still being considered for thinner tapes. (See Sub-clause 7.3)

4.0 ARMOUR AND REINFORCEMENT LOSSES λ_2

The formulae given in this section express the power loss occurring in metallic armour in terms of an increment, λ_2, of the power loss of all conductors.

Three-core cables - Steel tape armour or reinforcement

The following formulae apply to tapes 0.3 to 1.0 mm thick. Formulae for tapes less than 0.3 mm in thickness are under consideration.
The hysteresis loss is given for a frequency of 50 Hz by:

$$\lambda_2' = \frac{s^2 k^2 10^{-9}}{R d_A \delta}$$

where:

s = distance between conductor axes, cm

δ = equival thickness of armour, cm

i.e. $\dfrac{A}{\pi d_A}$

and:

A = armour cross-sectional area, cm^2

d_A = mean diameter of armour, cm

The factor k is given by:

$$k = \frac{1}{1 + \frac{d_A}{\mu \delta}}$$

where:

μ = relative permeability of the steel tape, usually taken as 300

For frequencies f other than 50 Hz, multiply the value given by the above formula by the factor $\frac{f}{50}$

The eddy-current loss is given for a frequency of 50 Hz by:

$$\lambda_2' = \frac{2{,}25\, s^2 k^2 \delta\, 10^{-8}}{R\, d_A}$$

and for any other frequency the value calculated from this formula must be multiplied by the factor $\left(\frac{f}{50}\right)^2$

The total armour loss factor is given by the sum of both hysteresis and eddy-current losses, thus:

$$\lambda_2 = \lambda_2' + \lambda_2''$$

NOTE - Magnetic armour or reinforcement, if any, increase eddy-current losses in the sheaths. Reference should be made to Sub-clause 6.8.

5.0 THERMAL RESISTANCE OF THE CONSTITUENT PARTS OF A CABLE T_1, T_2 AND T_3

This section gives the formulae for the calculation of thermal resistances per unit length of the different parts of the cable, T_1, T_2 and T_3.

5.1 Thermal resistance between core conductor and sheath T_1

Three-core cables, screened type.

Cables of this type may be first considered as belted cables for which is 0,5. Then, in order to take account of the thermal conductivity of the metallic screens, the result must be multiplied by a factor K, called the screening factor, which is given in figure 3, page 64, IEC (287) for different values of $\frac{t}{d_c}$ and different cable specifications.

Thus: $$T_1 = K \frac{\rho T}{2\pi} G$$

5.2 Calculation of Thermal Resistance T_2

Single-core cables and three-core cables having a common metallic sheath

$$T_2 = 0{,}366\, \rho_T \log_{10}\left(\frac{D_a}{D_s}\right)$$

where:

D_a = external diameter of the bedding, cm

D_s = internal diameter of the bedding, cm

5.3 Calculation of Thermal Resistance T_3

The external servings are generally in the form of concentric layers and:

$$T_3 = 0{,}366\ \rho_T \log_{10}\left(\frac{D_e}{D_a'}\right)$$

where:

D_e = external diameter of the outer serving, cm

D_a' = internal diameter of the outer serving, cm

APPENDIX 3

PROJECT COST ESTIMATE

The estimated cost of implementing the project with 10 outstations is as follows:

ITEM		COST
1.	Development Cost	R12 000
2.	Electronic Components	R 1 500
3.	P.C. Board Manufacture	R 300
4.	Assembly & Testing Boards	R 2 000
5.	Outstation Housing	R 700
6.	Radio Receiver & Transmitter	R 4 400
7.	Radio Antennae	R 1 100
8.	Labour - Installation & Commissioning	R 2 000
9.	Transport	R 500
	TOTAL COST	R24 500

APPENDIX 4

SPECIFICATIONS OF TRANSDUCER

MODEL	M125TRA/A
TYPE	RECTIFIER
INPUT	5A
OUTPUT	5 mA
LOAD	0 ... 2 koHM
CALIBRATION	0 ... 10 ... 40%
ACCURACY	CLASS 1

APPENDIX 5

OUTSTATION TAGBLOCK CONNECTIONS

BLOCK J - 6 X 2 pair cables to transducers

BLOCK M - 10 pair cable to outstation board plug

CROSS CONNECTIONS

M 1ab ———— J 11ab
M 2ab ———— J 12ab
M 3ab ———— J 13ab
M 4ab ———— J 14ab
M 5ab ———— J 15ab
M 6ab ———— J 16ab

APPENDIX 6

RS 232 PLUG PIN ALLOCATIONS

EIA RS 232C AND CCITT V.24 INTERFACE

PIN NO	EIA	CCITT	ABRIV	FUNCTION	TO DTE	TO DCE
1	AA	101	FG/GND	CHASSIS GROUND		
2	BA	103	TXD	TRANSMITTED DATA		→
3	BB	104	RXD	RECEIVED DATA	←	
4	CA	105	RTS	REQUEST TO SEND		→
5	CB	106	CTS	CLEAR TO SEND OR READY FOR SENDING	←	
6	CC	107	DSR	DATA SET READY	←	
7	AB	102	SG	SIGNAL GROUND OR COMMON RETURN		
8	CF	109	DCD/ RLSD	DATA CARRIER DETECT OR RECEIVE LINE SIGNAL DETECTOR	←	
9	-	-	+V	+VDC TEST VOLTAGE	←	
10	-	-	-V	-VDC TEST VOLTAGE	←	
11	-	-	BO	BUSY OUT		→
12	SCF	122	SDCD SRLSD	SECONDARY CARRIER DETECT OR SECONDARY RECEIVE LINE SIGNAL DETECTOR	←	
13	SCB	121	SCTS	SECONDARY CLEAR TO SEND OR SECONDARY READY FOR SENDING	←	
14	SBA	118	STD/ STXD	SECONDARY TRANSMITTED DATA		→
15	DB	114	SCT/ TXC	TRANSMITTED CLOCK OR TRANSMITTER SIGNAL ELEMENT TIMIN	←	
16	SBB	119	SRD/ SRXD	SECONDARY RECEIVED DATA	←	
17	DD	115	SCR/ RXC	RECEIVED CLOCK OR RECEIVER SIGNAL ELEMENT TIMING	←	
18	LL	141		LOCAL LOOPBACK		→
19	SCA	120	SRTS	SECONDARY REQUEST TO SEND		→
20	CD	108/2 108/1	DTR	DATA TERMINAL READY CONNECT DATA SET TO LINE		→

20	-	-	BO	BUSY OUT		→
21		140		REMOTE LOOPBACK		→
	CG	110	SQ	SIGNAL QUALITY DETECTOR		
22	CE	125	RI	RING INDICATOR	←	
23	CH	111	SS	DATA RATE SELECT		→
	CI	112		DATA RATE INDICATOR	←	
24	DA	113	SCIE/ EXC	EXTERNAL TRANSMITTED CLOCK OR TRANSMIT SIGNAL ELEMENT TIMING		→
25	-	142	TI	TEST INDICATOR OR	←	
	-	-	BO	BUSY OUT		→

APPENDIX 7

DATA INPUT PROGRAM LISTING

A brief discussion of all the procedures used in the main program is given below :

Procedure Initialise: The serial port, COM1, through which the cassette data is fed, is initialised according to the requirements of the system.

Procedure GetHeading: Before the data is read in, the substation name, and the start and end date and time of the readings must be entered. Fairly strict checks are performed to ensure that this data is accurate (procedures 'InputStr', 'Read2Digits', and 'NumberOfIntervals').

Procedure GetData: The data is stored in the Buffer array as it comes in. This array stores the sum of up to 4 readings or one hour's worth for each feeder. The average for each hour is then stored in the ReadingArray. GetData reads in 'NumReads' sets of readings, where NumReads is determined during the procedure 'GetHeading', by the procedure 'NumberOfIntervals'.

Procedure DataToDisk: When the serial data readings are finished, they are stored on disk, in a file identified by the name of the substation. The form of the file is described above.

Procedure CheckFiles: More than one cassette reading is used to fill one file. 'CheckFiles' examines the data disk to see whether a new file must be created or the old one may be used.

Procedure UseOldFile: The existing data is read into ReadingArray, and the new data is appended to ReadingArray. (The files are small, so loading the data into memory does not take long.)

Procedure CreateNewFile: The old file will be erased, but before doing so the operator is given the option of exiting so that he may use Lotus to produce a hard copy.

Procedure RestoreSerial: Restores the serial port to its initial state before this program was invoked.

References:
----

1. IBM PC Technical Reference Manual
   pp 2.141 - 2.164
   pp A.19 - A.22

2. Zenith PC-150 Technical Manual
   pg 3.18

```
  }
type
   DateStr     = string[8];
   TimeStr     = string[5];
   FileStr     = string[14];
   TwoCharStr  = string[2];
   FeederArray= array[1..8] of integer;
   Reading     = record
                     Feeder:      FeederArray;
                  end;
   Header      = record
                     SubName:     string[10];
                     StartDate,
                     StopDate:    DateStr;
```

```
                    StartTime,
                    StopTime:    TimeStr;
                 end;

    Status       = (parity,overrun,framing,break,timeout);
    StatusArray  = array[Status] of boolean;
    CharSet      = set of Char;                {pass set as param for input checks}
    AnyStr       = string[255];
    NumSet       = set of 0..99;

const
    maxReadings = 800;      { max readings per feeder, determined by Lotus graph
                              limitation of 200 points (800/4) }
    NumSubs     =      12;
    SubNameArray:     array[1..NumSubs] of string[10] =
                      ('BURGER','ORANGE','OAKGLEN','VREDELUST','THERMO','BEROL',
                       'ALBATROS','BALVENIE','SQUIRE','BELLVILLE','GREATMANS',
                       'COMPOST');

    FeederMin:        FeederArray =                {initialise array of Mins
                      (255,255,255,255,255,255,255,255);
    FeederMax:        FeederArray =                {initialise array of Maxs
                      (0,0,0,0,0,0,0,0);
    CreateCode:       char = '0';                  {code to use old file}

var
    TimeOutCount:        integer;            {no. of times GetSerialChar called}
    DataReady:             boolean;
    Error:                 StatusArray;
    SaveControl,                               {save start condition of COM1,}
    SaveDivisor:           integer;            {restore on exit}

    HeaderRec:             Header;
    ReadingArray:          array[1..512] of reading; {hourly avs of readings}
    ReadingFile:           file of Reading;
    ReadStrFile:           Text;

    NumReads:              integer;         {number of readings at 15 min intervals
    Hours:                 integer;         {NumReads div 4}

    FileName:              FileStr;         {name of disk file}
    HeaderLine:            anystr;          {header line in text file}
    TextLine:              anystr;          {any other line}
    SubNamePstn:           integer;         {position of string in SubNameArray}

    remainder:             integer;         {(0..3) - 15 min readings left over}
    BufferSum:             FeederArray;     {for averages}

    OldHours:              integer;         {hours already in the file}
    OldStartDate,                           {information extracted from header..}
    OldStopDate:           DateStr;         {line of existing file}
    OldStartTime,
    OldStopTime:           TimeStr;
```

```
NumErrors,                          {total errors during transmission}
NumParity,                          {subdivision of errors}
NumOverrun,
NumBreak,
NumFraming:         integer;
```

```
{*********************************************************************
          P r o c e d u r e     G e t S e r i a l C h a r
*********************************************************************}
procedure GetSerialChar(var val:                    byte;
                        var DataReady:              boolean;
                        var Error:                  StatusArray);
  { Reads in a character from the Zenith COM1 serial port.
    The serial port is preset at the head of the program by 'Initialise',
    to the following configuration :
            300 baud
            1 stop bit
            even parity
            no handshaking
            8 bit words
            1 start bit (set automatically)
    Error conditions are notified by the boolean variables.  }

var
   status:      byte;                   { Line Status register, determine errors
const
   ReceiveReg: integer = $3f8;          { Recive Buffer Register for serial #1 }
   ControlReg: integer = $3fb;          { Line Control Register for serial #1 }
   StatusReg:  integer = $3fd;          { Line Status Register }
   ModemReg:   integer = $3fc;          { Modem Control Register }
   ModemStat:  integer = $3fe;          { Modem Status Register }

begin
   status := port[StatusReg];
   if (status and $01) > 0 then
   begin
      DataReady := true;
      val := port[ReceiveReg];
   end
   else DataReady := false;
   if (status and $02) > 0 then Error[Overrun] := true
                        else Error[Overrun] := false;
   if (status and $04) > 0 then Error[parity] := true
                        else Error[parity] := false;
   if (status and $08) > 0 then Error[framing] := true
                        else Error[framing] := false;
   if (status and $10) > 0 then Error[break] := true
                        else Error[break] := false;
end;
```

```
{***********************************************************************}
procedure Initialise;
{-------------------}
{Initialises COM1 as described in GetSerialChar, followed by
 Initial read of serial port, to clear the receive buffer.
 }
const
   ControlReg: integer = $3fb;          { Line Control Register for serial #1 }
   DivisorReg: integer = $3f8;          { Divisor Latch, word }
var
   buffer:   byte;
begin
   {set serial port}
   SaveControl := port[ControlReg];
   port[ControlReg] := $1b;                           {set word lengtth, etc}
   port[ControlReg] := port[ControlReg] or $80;       {DLAB = 1}
   SaveDivisor := port[DivisorReg];
   portW[DivisorReg] := 384;                          {300 baud}
   port[ControlReg] := port[ControlReg] and $7f;      {DLAB = 0, receive Reg ready

   {clear receive buffer}
   GetSerialChar(buffer, DataReady, Error);
end;

procedure RestoreSerial;
{----------------------}
{Reset Serial port to the condition it was on entry}
const
   ControlReg: integer = $3fb;          { Line Control Register for serial #1 }
   DivisorReg: integer = $3f8;          { Divisor Latch, word }
begin
   port[ControlReg] := SaveControl;
   port[ControlReg] := port[ControlReg] or $80;       {DLAB = 1}
   port[DivisorReg] := SaveDivisor;
   port[ControlReg] := port[ControlReg] and $7f;      {DLAB = 0, receive Reg ready
end;
```

```
{*****************************************************************
        F u n c t i o n     N u m b e r O f R e a d i n g s
*****************************************************************}
procedure NumberOfIntervals(var  BeginDate, EndDate:    DateStr;
                            var  BeginTime, EndTime:    TimeStr;
                            var  NumInts           :    integer);
   { Determines the number of reading intervals between two date/time's
     by calculating the number of fifteen minute intervals between Begin
     and End.
     The GetDateTime procedure has already checked for errors in the
     inputting of the dates and times.
   }
var
   BeginMin, EndMin:         integer;       {0..59}
   BeginHour, EndHour:       integer;       {0..23}
   BeginDay, EndDay:         integer;       {0..31}
   BeginMonth, EndMonth:     integer;       {1..12}
   BeginYear, EndYear:       integer;       {85..99}
   result:                   integer;       {special var reqd for 'val' procedure

   Mins, Hours,
   Days, Month:              integer;

begin
   Mins := 0;   Hours := 0;     {initialise variables}
   Days := 0;   Month := 0;
   {get variables from Date and Time strings}
   Val(Copy(BeginTime,4,2), BeginMin,   result);
   Val(Copy(EndTime,4,2),   EndMin,     result);
   Val(Copy(BeginTime,1,2), BeginHour,  result);
   Val(Copy(EndTime,1,2),   EndHour,    result);
   Val(Copy(BeginDate,4,2), BeginDay,   result);
   Val(Copy(EndDate,4,2),   EndDay,     result);
   Val(Copy(BeginDate,1,2), BeginMonth, result);
   Val(Copy(EndDate,1,2),   EndMonth,   result);
   Val(Copy(BeginDate,7,2), BeginYear,  result);
   Val(Copy(EndDate,7,2),   EndYear,    result);

   {determine time difference between Begin and End}
   Mins := EndMin - BeginMin;
   if Mins < 0 then     {carry has taken place}
   begin
      Mins := Mins + 60;
      Hours := -1;
   end;
   hours := EndHour - BeginHour + Hours;
   if Hours < 0 then    {carry condition}
   begin
      Hours := Hours + 24;
      Days := -1;
   end;
   Days := EndDay - BeginDay + Days;
   if Days < 0 then     {carry condition}
```

```
   case BeginMonth of
     1,3,5,7,8,10,12: Days := Days + 31;
     4,6,9,11:        Days := Days + 30;
   else {February}
      if (BeginYear mod 4) > 0 then    {not leapyear}
         Days := Days + 28
      else
         Days := Days + 29;
   end;
   {not necessary to continue because one tape will never contain more
    than a few days information}

   Hours := Days *24 + Hours;
   NumInts := (Hours * 4) + (Mins div 15);
end;
```

```
{***************************************************************
          D a t a   I n p u t .   U t i l i t i e s
***************************************************************}

procedure Beep;
(*  Beep sounds the terminal bell or beeper *)
begin
  Write(^G);
end;

procedure ClearMessageLine;
(* Clears the line that has been reserved for messages *)
var
  x, y:  integer;
begin
   x := whereX;  y := whereY;
   gotoXY(10,24);
   write('
   gotoXY(x,y);
end;

procedure InputStr(var Str          : AnyStr;
{-----------------}    L,                              {Str :  string[L]}
                       X,Y           : Integer;       {X,Y = screen coords}
                       Term          : CharSet;       {set of terminator characters}
                   var TermCh        : Char      );
   {reads a charcter string from the Keyboard. Standard wordstar control
    characters may be used to to edit the text.
    The Substaion name is the only string required. Hence error checking
    is performed within this routine.
    Only as many letters as are required to uniquely identify the string
    as an element of the set to which it belongs need be typed in. The
    remaining characters are typed automatically.
    }
var
  Pos  : Integer;
  Ch : Char;
  i  : integer;
  StrOccurs:   integer;
begin
  Pos := 0;   Str := ''; {start with empty string}
  repeat
    GotoXY(X + Pos +1,Y); Read(Kbd,Ch);
    case Ch of
      #32..#126 : if Pos < L then
                  begin
                    if Length(Str) = L then
                      Delete(Str,L,1);
                    Pos := Pos + 1;
                    Insert(Ch,Str,Pos);
                    ClearMessageLine;
                    StrOccurs := 0;
                    for i := 1 to NumSubs do
```

```
                        if Str = copy(SubNameArray[i],1,Pos) then
                        begin
                           StrOccurs := StrOccurs + 1;
                           SubNamePstn := i;
                        end;
                     case StrOccurs of
                       0      : begin  {no SubName match}
                                  beep;
                                  gotoXY(X +1,Y); for i := 1 to L do write(' ');
                                  Delete(Str,1,L);
                                  Pos := 0;
                                  gotoXY(10,24);
                                  write('TYPING ERROR OR SUBSTATION NOT ON LIST
');

                                end;
                       1      : begin   {one identical match found}
                                  Str := SubNameArray[SubNamePstn];
                                  Pos := length(Str);
                                  GotoXY(X +1,Y); write(Str);
                                end;
                       2..20 : write(copy(Str,Pos,L));
                     end; {case}
                   end
                   else Beep;
      ^S         : if Pos > 0 then
                     Pos := Pos - 1
                   else Beep;
      ^D         : if Pos < Length(Str) then
                     Pos := Pos + 1
                   else Beep;
      ^A         : Pos := 0;
      ^F         : Pos := Length(Str);
      ^G         : if Pos < Length(Str) then
                   begin
                     Delete(Str,Pos + 1,1);
                     write(copy(Str,Pos +1,L),' ');
                   end;
      ^H,#127    : if Pos > 0 then
                   begin
                     Delete(Str,Pos,1);
                     write(^H,copy(Str,Pos,L),' ');
                     Pos := Pos - 1;
                   end
                   else Beep;
      ^Y         : begin
                     gotoXY(X +1,Y); for i := 1 to L do write(' ');
                     Delete(Str,1,L);
                     Pos := 0;
                   end;
  else
    if not (Ch in Term) then Beep;
  end;  {of case}
until Ch in Term;
Pos := Length(Str);
```

```
  GotoXY(X + Pos +1 ,Y);
  Write('' :L - Pos);
  TermCh := Ch;
end;


Procedure Read2Digits(x,y: integer;   var TwoDigitNum: integer;
{-------------------}                 AllowSet:        Numset;
                                      Term:            CharSet;
                                      var TermCh:      Char    );
  {Reads a two digit integer at position (x,y).
   All the date and time variables are two digits, so this routine
   is used for inputting data neatly. Also, because 2 digits are required
   every time, this simplifies error checking.
   Characters in the Term-inatition set are allowed, so the operator
   is given cursor control.
   }
var
   ch:          char;
   digitstr:    string[2];
   Pos:         integer;
   result:      integer;
begin
   Pos := 0;
   TermCh := chr(0);
   DigitStr := '';
   TwoDigitNum := -1;         {initialise to always be out of range}
   repeat
      gotoXY(x + Pos, y);
      repeat until keypressed; read(kbd,ch);
      case ch of
        #48..#57: begin
                     Pos := Pos + 1;
                     write(ch);
                     DigitStr := DigitStr + ch;
                     if Pos = 2 then TermCh := ^M;
                     ClearMessageLine;
                  end;
             ^E : TermCh := ^E;          {^E is only cntl char allowed}
         ^H,#127: if Pos = 1 then
                  begin
                     DigitStr := '';
                     gotoXY(x,y); write('  ');
                     Pos := 0;
                  end
                  else Beep;
        #32..#47,
       #58..#126: begin
                     beep;
                     ClearMessageLine;
                     gotoXY(10,24);
                     write('Numerics only');
                  end;
      else
```

```
        beep;
    end;   {case}
    if Pos = 2 then
    begin
        Val(DigitStr, TwoDigitNum, Result);
        if not (TwoDigitNum in Allowset) then
        begin
            beep;
            gotoXY(10,24);
            write('Numeric expression out of bounds');
        end;
        Pos := 0;
        DigitStr := '';
    end;
  until (TwoDigitNum in AllowSet) or (TermCh = ^E);
end;
```

```
{***********************************************************************
         P r o c e d u r e    G e t H e a d i n g
***********************************************************************}
Procedure GetHeading(var HeaderRec: Header);
   {The Substation Name, and the Start and Stop Dates and Times of each
    set of readings are input from the Keyboard.
    To begin with, Date and Time variables are integers. This allows
    easy input-error checking. The variables are grouped together and
    passed back to the calling program as character strings, which is
    more in keeping with general practice (eg Dbase).
    As far as cursor control is concerned, only fields before the current
    one may be edited (ie. only ^E is offerred). A field MUST be retyped if
    a previous one was edited. This ensures that (correct) values are entered
    in all the fields.
    Error checking takes place in the respective input routines (string or
    numeric), and in addition, this procedure checks that the start and stop
    dates are within 200 hours (800 intervals) of each other.
    }
const
  Term : CharSet  =  [^E,^I,^M];    {terminator chars for inputs}

var
   StartMin, StopMin:          integer;      {0..59}
   StartHour, StopHour:        integer;      {0..24}
   StartDay, StopDay:          integer;      {0..31}
   StartMonth, StopMonth:      integer;      {1..12}
   StartYear, StopYear:        integer;      {85..99}
   result:                     integer;
   str2:                       string[2];
   L : Integer;
   TermCh : Char;

procedure StrZero(TwoDigitInt: integer;  var str2: TwoCharStr);
 {converts the 2 digit integer, int, to a two character string, with the
  str2[1] = '0' if int <= 9 }
begin
str(TwoDigitInt:2,str2);
   if TwoDigitInt <= 9 then str2[1] := '0';
end;

begin {GetHeading}
 with HeaderRec do
 begin
   ClrScr;
   gotoXY(10,2);  write('C E N T R A L   A R E A   T E L E M E T R Y   S Y S
M');
   gotoXY(10,3);  write('-----------------------------------------------------
--');
   gotoXY(10,5);  write('Please enter the following Data :');
   gotoXY(15,7);  write('Substation Name ');
   gotoXY(15,8);  write('         or Code ');
   gotoXY(15,10); write('Start of Readings');
   gotoXY(18,11); write('Date ');
```

```
  gotoXY(18,12); write('Time ');
  gotoXY(50,10); write('End of Readings');
  gotoXY(53,11); write('Date ');
  gotoXY(53,12); write('Time ');


  {read inputs}
  TextBackGround(Blue);
  TextColor(White);
  with HeaderRec do
  repeat    {until the StopDate is after the StartDate}
    gotoXY(31,7);  write('              ');
    gotoXY(23,11); write('MM/DD/YY');
    gotoXY(23,12); write('HH-MM');
    gotoXY(58,11); write('MM/DD/YY');
    gotoXY(58,12); write('HH-MM');

    L := 1;
    repeat
      case L of
         1 : InputStr(SubName,10,30,7,Term,TermCh);
         2 : begin end;
         3 : read2digits(23,11, StartMonth,[1..12], Term,TermCh );
         4 : read2digits(26,11, StartDay,  [1..31], Term,TermCh );
         5 : read2digits(29,11, StartYear, [85..99],Term,TermCh );
         6 : read2digits(23,12, StartHour, [0..23], Term,TermCh );
         7 : read2digits(26,12, StartMin,  [0..59], Term,TermCh );
         8 : read2digits(58,11, StopMonth,
[1,StartMonth,StartMonth+1],Term,TermCh );
             {Allows Month = 1 always}
         9 : read2digits(61,11, StopDay,   [1..31], Term,TermCh );
        10 : read2digits(64,11, StopYear,  [StartYear,StartYear+1],
Term,TermCh);
        11 : read2digits(58,12, StopHour,  [0..23], Term,TermCh );
        12 : read2digits(61,12, StopMin,   [0..59], Term,TermCh);
      end;
      if (TermCh = ^I) or (TermCh = ^M) then
         if L = 12 then
           L := 1
         else L := L + 1
      else
         if TermCh = ^E then
           if L = 1 then
             L := 1            {can't go outside record}
          else L := L - 1;
    until (TermCh = ^M) and (L = 1);
```

```
      {concatenate Date and time strings}
      StartDate := '';  StopDate := '';
      StartTime := '';  Stoptime := '';
      StrZero(StartMonth,Str2);  StartDate := StartDate + Str2 + '/';
      StrZero(StartDay,Str2);    StartDate := StartDate + Str2 + '/';
      StrZero(StartYear,Str2);   StartDate := StartDate + Str2;
      StrZero(StopMonth,str2);   StopDate := StopDate + str2 + '/';
      StrZero(StopDay,str2);     StopDate := StopDate + str2 + '/';
      StrZero(StopYear,str2);    StopDate := StopDate + str2;
      StrZero(StartHour,str2);   StartTime := StartTime + str2 + 'H';
      StrZero(StartMin,str2);    StartTime := StartTime + str2;
      strZero(StopHour,str2);    StopTime := StopTime + str2 + 'H';
      strZero(StopMin,str2);     StopTime := StopTime + str2;

      NumberOfIntervals(Startdate, StopDate, StartTime, StopTime, Numreads);
      Hours := NumReads div 4;
      if (NumReads <= 0) or (NumReads > 800) then
      begin
         gotoXY(10,24);
         beep;
         write('Start and End dates not within limits   ');
      end;
   until (NumReads >= 1) and (NumReads <= 800);

   TextBackGround(Black);
   textcolor(yellow);
  end;  {with HeaderRec}
end;    {GetHeadings}

{***************************************************************
           F i l e   U t i l i t i e s
***************************************************************}
function FileExist(var  FileName: FileStr): boolean;
{-----------------}
{checks for the existence of a file}
var
   Fil:  Text;
begin
   assign(Fil, FileName);
   {$I-}     {I/O checking directive off}
   Reset(Fil);
   {$I+}
   FileExist := (IOresult = 0);
   close(Fil);
end;

procedure ValsFromStr(var ValArray:  FeederArray;  var Str: AnyStr);
{--------------------}
{extracts the eight numbers from a text line string, where numbers are
 allocated four chars in the string.
 }
var
  i,j:    integer;
```

```
  result: integer;
begin
   for i := 1 to 8 do
   begin
      j := 0;                         {j counts blanks that separate 'numbers'
      while Pos(' ', Str) > 0 do      {in text line}
      begin
         j := j +1;
         delete(Str,1,1);             {delete the blanks}
      end;
      val(copy(Str, 1, 4-j), ValArray[i], result);
      if result > 0 then beep;
      delete(Str, 1, 4-j);
   end;
end;   {ValsFromStr)}

procedure NumFromStr(var  Num: integer;  var Str: AnyStr);
{-------------------}
{extracts a number from a text string where the text number is 4 chars long}
var
  i, result: integer;
begin
   i := 0;                         {i counts blanks that separate 'numbers'}
   while Pos(' ', Str) > 0 do      {in text line}
   begin
      i := i +1;
      delete(Str,1,1);             {delete the blanks}
   end;
   val(copy(Str, 1, 4-i), Num, result);
   if result > 0 then beep;
   delete(Str, 1, 4-i);
end;  {NumFromStr}

procedure ConnectMessage;
{----------------------}
{writes message telling to connect up cassette to serial port}
var
  ch: char;
const
  space:  char = ' ';
begin
   textcolor(green);
   repeat
      gotoXY(1,16);
      writeln('                SO FAR SO GOOD - '); beep;
      writeln('             NOW CONNECT UP THE CASETTE TO THE SERIAL PORT');
      writeln('             AND LETS START THINGS ROLLING');
      writeln;
      write('             PRESS SPACE BAR TO CONTINUE...');
      repeat until keypressed;
      read(kbd,ch);
      if ch <> space then beep;
   until ch = space;
   textcolor(yellow);
```

```
end;   {ConnectMessage}


{*********************************************************************
                File        I n i t i a l i z a t i o n
*********************************************************************}
procedure UseOldFile(var FileName: FileStr);
{-------------------}
{If the existing file is still usable, this procedure rewrites the header
 line to reflect the change in stop date and time, and copies the other
 data into the relevant arrays.
 }
var
  i,j:       integer;
  Buffer:    FeederArray;
begin
   assign(ReadStrFile, FileName);
   reset(ReadStrFile);

   ClrScr; gotoXY(10,10);
   writeln('LOADING EXISTING FILE INTO MEMORY');

   readln(ReadStrFile, HeaderLine);
   {delete OldStop- Date & time and replace with new}
   delete(HeaderLine, Length(HeaderLine) -16, 17);
   With HeaderRec do
   HeaderLine := concat(HeaderLine, StopDate,'" "',StopTime,'"');

   readln(ReadStrFile,TextLine);
   ValsFromStr(FeederMax, TextLine);         {initialise feeder max }
   readln(ReadStrFile, TextLine);            {and min at previous vals}
   ValsFromStr(FeederMin, TextLine);
   readln(ReadStrFile, TextLine);            {same as OldHours}

   for i := 1 to OldHours do         {put hourly avs into ReadingArray}
   begin
      readln(ReadStrFile, TextLine);
      with ReadingArray[i] do ValsFromStr(Feeder, TextLine);
   end;

   {read remainder and BufferSum}
   readln(ReadStrFile, TextLine);
   NumFromStr(remainder, TextLine);
   readln(ReadStrFile, TextLine);
   ValsFromStr(BufferSum,  TextLine);

   close(ReadStrFile);
   ClrScr;
   ConnectMessage;
end;    {UseOldFile}
```

```
procedure CreateNewFile(var FileName : AnyStr;  CodeChar: char);
{---------------------}
{Creates a new file with name given by the variable, FileName, if necessary.
The reason for creating the file is given by the parameter CodeChar, and a
message to this effect is flashed onto the screen.
The HeaderLine is assembled, and other initial conditions set.
}
var
  i,j:                 integer;
  ch:                  char;
  SubNamePstnStr:      string[4];
begin
   ClrScr;
   gotoXY(1,7);
   case CodeChar of
     '1' : begin
              writeln('      The existing file, ',FileName,',is too large to')
              writeln('      allow new information to be added to it.');
              writeln('      Pease ensure you have a hard copy of the data');
              writeln('      before you continue.');
           end;
     '2' : begin
              writeln('      More than one fifteen minute time interval has');
              writeln('      elapsed since the end of the last set of data.');
              writeln('      Pease ensure you have a hard copy of the data');
              writeln('      before you continue.');
           end;
     '3' :    writeln('     ',Filename,' does not exist.');
   end; {case}
   repeat
      gotoXY(1,12);
      writeln('      Choose one of the followinng :-');
      writeln;
      writeln('           0 - Exit to Dos');
      writeln;
      writeln('           1 - Continue');
      writeln;
      write('     Enter your Choice..');
      repeat until keypressed;
      read(kbd,ch); writeln(ch);
      if not (ch in ['0','1']) then beep;
   until ch in ['0','1'];
   if ch = '0' then
   begin
      ClrScr;
      halt;     {return to Dos} .
   end
   else begin
      ClrScr;  gotoXY(10,10);
      writeln('New file ',fileName,' will be created');
      ConnectMessage;
   end;
```

```
   with HeaderRec do            {set HeaderLine for text file}
   begin
      {for writing files to Lotus, strings must be enclosed in quotes}
      Str(SubNamePstn:4, SubNamePstnStr);
      HeaderLine := concat(SubNamePstnStr,' "',SubName,'" "',StartDate);
      HeaderLine := concat(HeaderLine,  '" "',StartTime,'"');
      HeaderLine := concat(HeaderLine, ' "',StopDate,'" "',StopTime,'"');
   end;
   for j := 1 to 8 do          {initialise arrays}
   begin
      BufferSum[j] := 0;     {Buffer of one hour's worth of data}
      FeederMax[j] := 0;
      FeederMin[j] := 255;
   end;
   OldHours := 0;              {since new file, no previous readings in file}
   remainder := 0;
end;   {CreateNewFile}
```

```
{*****************************************************************
         P r o c e d u r e     C h e c k F i l e s
*****************************************************************}
procedure CheckFiles(var HeaderRec: Header; var CreateCode: char);
{-------------------}
   {Checks to see whether the file of a particular substation exists,
    and if it does, whether it has enough space to fit the new set of
    readings (in all, there must be less than 200 records in order to
    fit on the Lotus graphs).  Also, the new set of readings must start
    within 29 minutes of the end of the previous set (ie. the loss of only
    one set of readings is allowed) or the program will not allow the
    new set to be appended to the old.
    The file is assumed to be re-usable (CreateCode = 0), but if it is found
    not to be, CreateCode is assigned the appropriate value.
    }
var
   i            : integer;
   MissedInts   : integer;     {no. of 15 min intervals where no readings taken
   OldInts      : integer;     {no. of existing 15 min intervals}

begin
   CreateCode := '0';
   with HeaderRec do
   FileName := concat('A:', copy(SubName,1,8), '.PRN');
   {PRN is the extension required by Lotus for reading external text files}
   assign(ReadStrFile, FileName);
   if FileExist(FileName) then
   begin
      Reset(ReadStrFile);
      Readln(ReadStrFile, HeaderLine);
      {file will always have header line}
      {Extract Old Dates and Times (pstns fixed relative to end of string)}
      OldStopDate  := copy(HeaderLine, length(HeaderLine) -16, 8);
      OldStopTime  := copy(HeaderLine, length(HeaderLine) -5,  5);
      OldStartDate := copy(HeaderLine, length(HeaderLine) -35, 8);
      OldStartTime := copy(HeaderLine, length(HeaderLine) -24, 5);
      NumberOfIntervals(OldStartDate, OldStopDate, OldStartTime,OldStopTime,
                        OldInts);
      OldHours := OldInts div 4;          {no. records already in file}
      with HeaderRec do NumberOfIntervals(OldStopDate, StartDate,
                                          OldStopTime, StartTime, MissedInts);

      if (MissedInts = 0) or (MissedInts = 1) then
      begin
         if (OldHours + Hours) > 200 then CreateCode := '1';
         {else UseOldFile}
      end
      else CreateCode := '2';      {too many missed intervals}
   end  {if FileExist}
   else CreateCode := '3';
   close(ReadStrFile);
end;
```

```
{*********************************************************************
              S e r i a l     E r r o r     U t i l i t i e s
**********************************************************************}
procedure ErrorMessage(NumErrors: integer);
{----------------------}
{If one or more errors occurred during the transmission, the operator
 is asked if he would like to retransmit the data.
 }
var
  ch:     char;
begin
   if NumErrors > 0 then
   begin
      ClrScr;
      gotoXY(10,7);
      writeln('There were ',NumErrors,' errors during the transmission,');
      writeln('                broken up as follows :');
      gotoXY(15,10); write('Parity Errors   -   ',NumParity:4);
      gotoXY(15,11); write('Framing Errors -   ',NumFraming:4);
      gotoXY(15,12); write('Break Errors    -   ',NumBreak:4);
      gotoXY(15,13); write('Overrun Errors -   ',NumOverrun:4);
      gotoXY(10,16);
      write('Do you want to abort (Y/N) ? ');
      repeat
         gotoXY(39,16);
         read(ch);
      until ch in ['y','Y','n','N'];
      if UpCase(ch) = 'Y' then halt;
   end;
end;   {ErrorMessage}


procedure ErrorHandler(var ErrorType: Status;  var NumErrors: integer);
{----------------------}
{Errors that occur during normal serial input, ie. parity, framing, overrun
 and break, are summed as they occur, and printed after completion of the
 data input by "ErrorMessage".
 TimeOut errors (where a serial character is not received within the normal
 period) are different. Generally, a timeout error is fatal, and the program
 is abandoned. One obvious cause of a timeout error is if the start-end
 interval, as entered, exceeds the amount of data on the cassette tape.
 A second is if an erroneous start bit was detected at the start of the tape,
 long before the actual data.
 }
var
  x, y:   integer;        {remember screen coords}
begin
   NumErrors := NumErrors +1;
   case ErrorType of
     parity   : NumParity := NumParity +1;
     overrun  : NumOverrun := NumOverrun +1;
     framing  : NumFraming := NumFraming +1;
     break    : NumBreak := NumBreak +1;
```

```
    timeout   : begin
                   window(1,1,80,25);
                   TextColor(Yellow);
                   ClrScr;
                   gotoXY(30,10); write('      Timeout Error');
                   gotoXY(30,12); write('Abandoning Data Gathering');
                   halt;
                end;
   end;
end;
```

```
{**************************************************************
        P r o c e d u r e   G e t D a t a
**************************************************************}
Procedure GetData;
{----------------}
  {Reads the incoming data stream and assigns each byte to the correct
   array. If the Old file is being used, then the ReadingArray will already
   be filled with OldHours elements, and new elements will be added from
   that point.
   A timeout will occur if the GetSerialChar procedure is polled 1000 times
   before a character is received, ie. no character is being sent. (At
   300 baud, it was found that DataReady is true after about 200 calls
   to the procedure.)
   If errors are detected, appropriate action is taken.
   The ASCII characters representing the incoming bytes are scrolled accross
   the screen, windowed in the bottom half of the screen. Error messages
   are allocated the right top corner of the screen.
   }
var
   i,j:              integer;                          {counters}
   start:            integer;                          {start count}
   ErrorType:        Status;
   Buffer:           byte;         {stores readings in memory as they come in.
   NumErrors:        integer;

begin
   ClrScr;
   gotoXY(10,4);  write('C E N T R A L   A R E A   T E L E M E T R Y   S Y S
M');
   gotoXY(10,5);  write('-------------------------------------------------------
--');
   gotoXY(1,12);
   writeln('==========RECEIVING DATA
=========================================================');
   gotoXY(1,24);

write('__________________________________________________________________________
____');

   window(1,13,79,23);  ClrScr;
   TextColor(LightRed);
   NumErrors := 0;    NumParity :=0;
   NumOverrun :=0;    NumBreak  :=0;   NumFraming :=0;
   start := (OldHours * 4) + remainder;

   for i := (1 +start) to (NumReads +start) do
   begin
      {initial fetch data}
      for j := 1 to 8 do
      begin
         TimeOutCount := 0;
         Error[TimeOut] := false;
         if (i = start +1) then     {first character, no timeout}
```

```
            repeat
               GetSerialChar(Buffer, DataReady, Error);
            until DataReady
         else
            repeat
               GetSerialChar(Buffer, DataReady, Error);
               TimeOutCount := TimeOutCount + 1;
               if TimeOutCount > 1000 then Error[TimeOut] := true;
            until DataReady or Error[TimeOut];

         {check if an error in the transmission occurred}
         for ErrorType := parity to timeout do
            if Error[ErrorType] then ErrorHandler(ErrorType,NumErrors);
         write(chr(buffer));
         if Buffer < FeederMin[j] then FeederMin[j] := Buffer;
         if Buffer > FeederMax[j] then FeederMax[j] := Buffer;
         BufferSum[j] := BufferSum[j] + Buffer;

         if (i mod 4) = 0 then  {one hour's worth of data}
         begin
            with ReadingArray[i div 4] do Feeder[j] := BufferSum[j] div 4;
            BufferSum[j] := 0;
         end;  {buffer}
      end;  {for j loop}
   end; {for NumReads loop}
   window(1,1,80,25);
   TextColor(Yellow);
   ErrorMessage(NumErrors);
   remainder := (NumReads + remainder) mod 4;
end;
```

```
{*******************************************************************
        P r o c e d u r e    D a t a T o D i s k
*******************************************************************}
procedure DataToDisk(var FileName:  FileStr);
{-------------------}
{Writes data to the file given by FileName. The data must be written in
 the strict order described at the top of the program.
 }
var
  i,j:          integer;
  FeederStr:    string[5];        {string[5] ensures always a space between
                                   numbers on disk}
begin
   assign(ReadStrFile, FileName);
   rewrite(ReadStrFile);

   ClrScr; gotoXY(15,12);
   writeln('WRITING TO DISK...');

   {first HeaderLine, then max and mins, then data}
   writeln(ReadStrFile, HeaderLine);
   for j := 1 to 8 do
   begin
      str(FeederMax[j]:4, FeederStr);
      write(ReadStrFile, FeederStr);
   end;
   writeln(ReadStrFile);
   for j := 1 to 8 do
   begin
      str(FeederMin[j]:4, FeederStr);
      write(ReadStrFile, FeederStr);
   end;
   writeln(ReadStrFile);

   {hourly data readings}
   Str((OldHours + Hours):4, FeederStr);       {no. of readings at top}
   writeln(ReadStrFile, FeederStr);
   for i := 1 to (OldHours + Hours) do         {OldHours =0 if New file created}
   with ReadingArray[i] do
   begin
      for j := 1 to 8 do
      begin
         Str(Feeder[j]:4, FeederStr);
         write(ReadStrFile, FeederStr);
      end;
      writeln(ReadStrFile);
   end;

   Str(remainder:4, FeederStr);                {append remainder + bufferSum}
   writeln(ReadStrFile, FeederStr);
   for j := 1 to 8 do
   begin
      Str(BufferSum[j]:4, FeederStr);
```

```
      write(ReadStrFile, FeederStr);
   end;
   writeln(ReadStrFile);

   close(ReadStrFile);
end;
```

```
{****************************************************************}
begin {main}
   Initialise;
   GetHeading(HeaderRec);
   CheckFiles(HeaderRec,CreateCode);
   if CreateCode = '0' then UseOldFile(FileName)
                       else CreateNewFile(FileName, CreateCode);
   GetData;
   DataToDisk(FileName);
   RestoreSerial;
end.
```

APPENDIX 8

MACRO LISTINGS

AUTO-EXECUTE MACRO \0

```
/x1Name of Sub-station?~ae2~
/cae2~v20.ac20~/cae2~ag38~
/rncfeedname~v2~
/rncfeedname2~v12~
/rncfeedct~k4~
/xnHow Many Feeders? ~ah8~
/rncnofeed~ah8~
{goto}nofeed~{edit}+1{calc}~
/rnccount~ah9~
{goto}count~1~
/x1Enter Feeder Name~feedname~
/xnEnter C.T. Ration~Feedct~
/cfeedname~feedname2~
{goto}count~
{edit}+1{calc}~
/rncfeedname~{right}~
/rncfeedname2~{right}~
{goto}feedct~/rndfeedct~{right}/rncfeedct~~
/xinofeed-count~/xgag23~
/rnc\0~\m~
/xc\s~
{goto}ae3~{edit}{bs}r~{down}
{edit}{home}{right}{del}{del}{del}{del}{del}~
/xc\s~
/fr
~
```

IMPORT MACRO

```
Import
Import data and compute values
{calc}
/xc\i~
/xc\h~
/xmmainmenu~
```

MACRO \I

```
/~~pd{home}xib4~{left}re.{end}{down}{end}{down}{end}{right}~{goto}k6~
   /re{end}{down}{end}{right}~{home}{right}
/fin
{?}
~
{goto}k5
/ck5.r5 {down}.{end}{left}{end}{down}{right}{right}{right}~
/reb1~{home}Time~{goto}c1~/rlr{end}{right}~
/cb1.g1~k1~
{calc}/~p~
/xr
```

MACRO\H

```
/~~pd/ce1 u18 {goto}u18~{edit}{home}{right}{del}{del}~
/cel~a5~
/rnctime~a5~
/{goto}a3~@count({end}{down}{right}.{end}{down}{calc}~/rnccount~a3~
/{goto}a5~{edit}{bs}{bs}{bs}{home}{del}~
/rnctime~{down}~{goto}time~
+{up}+1{calc}~
/xitime>24~1~
{goto}count~
{edit}-1{calc}/xicount~/xgu7~
/rnctime~a5~
{goto}time~
/xitime<10~{edit}{home} 0~xgu17~
{edit}{home}~
{edit}
H00
~/rnctime~{down}~
/xitime~/xgu14~
/rea3~/~p~
/xr
```

GRAPH MACRO \G

```
1
U Unused
/~~pd{goto}
k
5~
/grg
osymu
k
4~
gtfData Acquisition System~
ts
U
~
tyLoad {Amps}
txTime~
qx{home}{end}{down}.{end}{down}~
oss12~
qa.{end}{down}~
vs
1
~
U q
U {goto}
U v
U 37
U /m{down}~
U v
U 23~/~~p~/re
U v
U 25.
U v
U 35~
rq /xmmenu~
/xmmainmenu~
```

MACRO \P

```
{goto}k5~
/ppr{bs}.{end}{right}{end}{down}~
agq
/xr
```

MACRO \S

```
/fs
~c
{esc}/xr
```

APPENDIX 9

INTRAC TELEMETRY SYSTEM

A Motorola Intrac system has been implemented for providing telemetry at the large 132/11 kV stepdown substations. This system has provided extremely useful information but only as far the 11 kV busbars of the stepdown substation.

A brief description of the features of this system is provided for background purposes and to provided a comparison with the Load Monitoring System discussed later in this report.

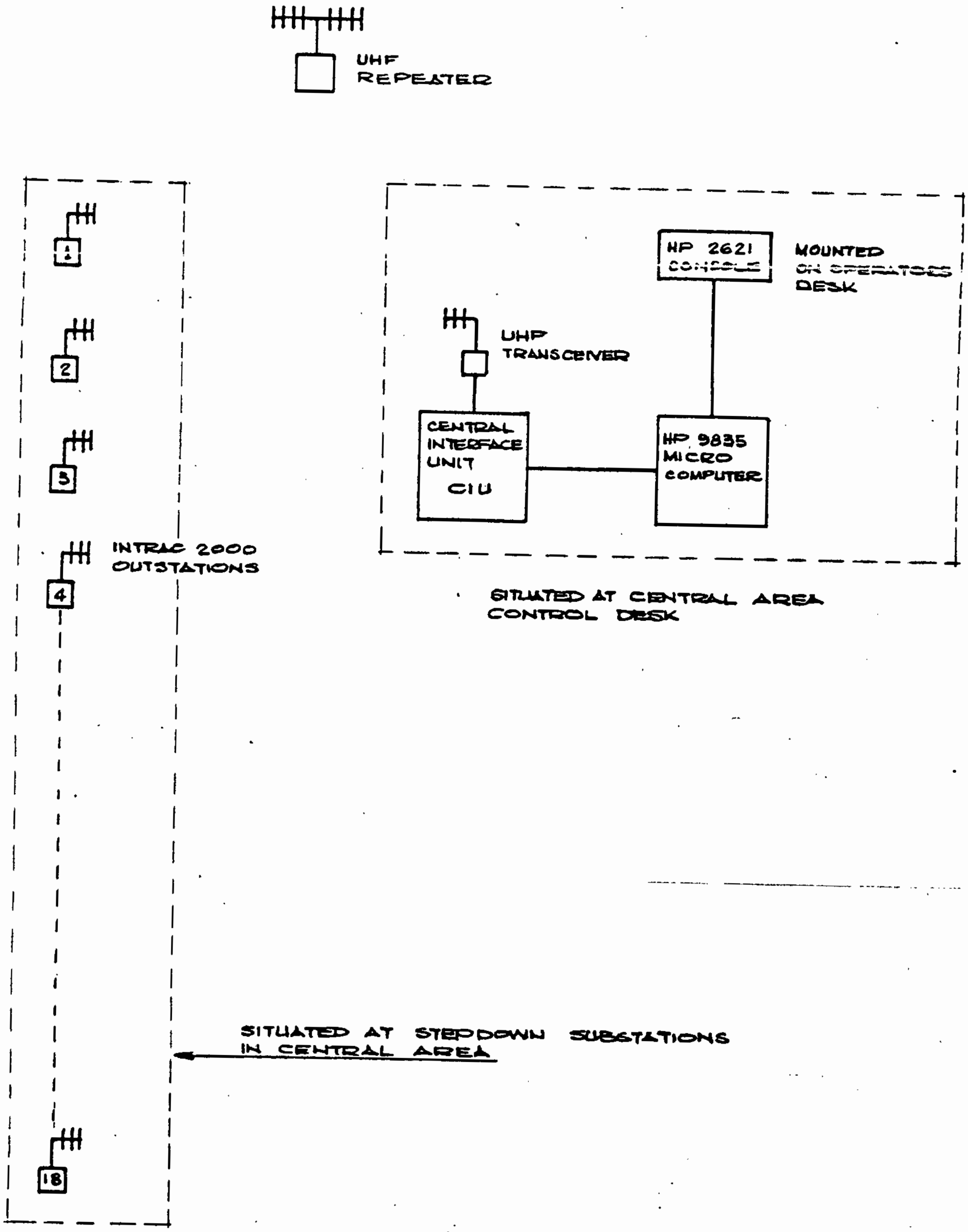
At each stepdown substation a Motorola Intrac 2000 outstation has been installed. These outstations are labelled 1 - 18 in the diagram (Fig. 7). This outstation is not microprocessor based and comprises mainly CMOS integrated circuitry. Each outstation reports information on the following quantities.

Analogues	(voltage , current) - 18 (max.)
Status	(breaker indications on or off - 24 (max.)
Alarms	(analogues out of limits) - 24 (max.)

The outstations are not equipped for controlling breakers remotely from the Urban Control Centre. The system is intended purely as a Telemetry system.

Communication between the Urban Control Centre and the 18 outstations is by means of UHF Radio. A repeater situated on a nearby hill ensures an effective radio path between all the outstations and the Control Centre. This is indicated in Fig. A2.

A Central Interface Unit (CIU - Motorola 6800 board) scans each outstation by address in turn. The information is transferred from the CIU to an HP 9835 computer. The system software is resident in this computer. Unfortunately this system provides no means of storing analogue data for use at a later stage. The information has to be manually recorded from the operator's VDU. This causes unnecessary work for the operators. The Operator's VDU (HP 2621) communicates in a serial mode at 9600 Bd with the 9835 computer.



SCHEMATIC DIAGRAM OF MOTOROLA INTRAC TELEMETRY SYSTEM FOR URBAN SUBSTATIONS

Fig. A2

While the system performs well, it has some major disadvantages namely:

1. Inability to store information and produce numerical and graphical reports.

2. High cost of Intrac outstation ± R30 000 per station.

3. Incompatibility of later generation Motorola outstation (μ Processor based) with present CIU and system software. A major software rewrite at considerable cost would be necessary to rectify this shortcoming.

GLOSSARY

GLOSSARY OF TERMS

ANALOGUE	-	refers to a quantity that can be monitored eg. current, voltage (in the data acquisition context)
A/D	-	Analogue to Digital conversion
BASIC	-	High level computer language; stands for: Beginners All Purpose Symbolic Instruction Code
BAUD	-	Bit per second
CCITT	-	Committee Cominique International Telegrafique et Telefonique
CIU	-	Central Interface Unit
CMOS	-	Complimentary Metal Oxide Silicon; low power consumption logic
CT	-	Current Transformer
DOS	-	Disc Operating System
E^2PROM	-	Electrically Erasable Programmable Read Only Memory
FEEDER	-	Overhead line or underground cable transferring power to or from a substation
FSK	-	Frequency Shift Key Modulation
IC	-	Integrated Circuit Package
INTRAC	-	Trade name for a particular telemetry system
JFET	-	Junction Field Effect Transistor
LOTUS 1-2-3	-	Spreadsheet package used to produce graphs and tables of data
LSB	-	Least Significant Bit
MSB	-	Most Significant Bit
OUTSTATION	-	Hardware situated at substation to gather data
PASCAL	-	High level computer language
RETICULATION	-	Distribution of electrical power at voltages of 11 kV and lower
RS 232	-	Standard for serial data comminication between data set and data terminal
SOLKOR	-	Type of protection used on cable networks; works on principle of checking a current entering a link and leaving a link
TTL	-	Transistor Transistor Logic
UART	-	Universal Asynchronous Receiver Transmitter; converts parallel data to serial and vice versa
UHF	-	Ultra High Frequency; used in regard to radio transmission and falls in band 300 MHz - 1 GHz
VDU	-	Visual Display Unit
ZVEI	-	Tone calling system used to identify mobile radios; stands for: Zentral - Verband Electrotechnische Industries