Microcontroller based ground resistance monitor
by Senthil N Kumar

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering
Montana State University

Abstract:
During ground fault conditions, the flow of current to earth will produce gradients in and around a substation. The magnitude of gradients caused are dependant upon the ground resistance of the substation. Such gradients, if large enough will cause bodily harm to a person in and around the substation.

To determine dangerous situations if any, monitoring of the ground resistance is necessary. An instrument for such monitoring of the ground resistance, the Microcontroller Based Ground Resistance Monitor, has been developed.

The Monitor measures the value of ground resistance after every 24 hours and stores in memory. The voltage used for the measurement and the time of measurement are also recorded. Readings will be recorded automatically for over a period of three years after which readings will be overwritten. The Monitor also calculates the mean and standard deviation squared of the most recent 64 readings of the ground resistance. An LED is switched 'on' if the present value of ground resistance exceeds the preset maximum value. An error analysis and the results of testing the Monitor in a laboratory setting are also outlined.

MICROCONTROLLER BASED GROUND RESISTANCE MONITOR

by

Senthil N. Kumar

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Electrical Engineering

MONTANA STATE UNIVERSITY
Bozeman, Montana

December 1992

APPROVAL

of a thesis submitted by

Senthil N. Kumar

This thesis has been read by each member of the thesis committee and has been found satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the College of Graduate Studies.

Dec 7 1992
Date

Chairperson, Graduate Committee

Approved for the Major Department

Dec 7 1992
Date

Head, Major Department

Approved for the College of Graduate Studies

12/16/92
Date

R L Brown
Graduate Dean

STATEMENT OF PERMISSION OF USE

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Montana State University, I agree that the Library shall make it available to borrowers under the rules of the Library.

If I have indicated my intention to copyright this thesis by including a copyright notice page, copying is allowable only for scholastic purposes, consistent with "fair use" as prescribed in the U.S Copyright Law. Requests for permission for extended quotation from or reproduction of this thesis in whole or in parts may be granted only by the copyright holder.

Signature [signature]

Date 12/15/92

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

During ground fault conditions, the flow of current to earth will produce gradients in and around a substation. The magnitude of gradients caused are dependant upon the ground resistance of the substation. Such gradients, if large enough will cause bodily harm to a person in and around the substation.

To determine dangerous situations if any, monitoring of the ground resistance is necessary. An instrument for such monitoring of the ground resistance, the Microcontroller Based Ground Resistance Monitor, has been developed.

The Monitor measures the value of ground resistance after every 24 hours and stores in memory. The voltage used for the measurement and the time of measurement are also recorded. Readings will be recorded automatically for over a period of three years after which readings will be overwritten. The Monitor also calculates the mean and standard deviation squared of the most recent 64 readings of the ground resistance. An LED is switched 'on' if the present value of ground resistance exceeds the preset maximum value. An error analysis and the results of testing the Monitor in a laboratory setting are also outlined.

CHAPTER 1

INTRODUCTION

This thesis discusses the design and construction of the microcontroller-based ground resistance monitor. An introduction of ground resistance and the need to monitor it are provided in this chapter.

Grounding in Electric Power Systems

Important aspects of grounding in electric power systems are:

(a) System (neutral) grounding- for proper operation and protection of the system.

(b) Equipment grounding- for safety.

(c) Lightning protection grounding- for protection against lightning.

Grounding electrodes such as wires, rods, pipes, strips, plates are used to achieve adequate grounding. When current is injected in a grounding electrode and collected at a point very far away from the electrode, the soil attains a potential gradient which depends on the current density in the soil and the resistivity of the soil. This produces a potential on the electrode with respect to a faraway point on the soil assumed to be at zero potential. The voltage of the electrode per unit current injected into the soil is defined as the ground resistance of the electrode. It is a function of the size and shape of the electrode and the resistivity of the soil. Most of this ground resistance is offered by the soil in the vicinity of the electrode as the current density in the soil becomes negligible at large distances.

Grounding in Substations

When a line to ground fault occurs, currents of thousands of amperes may be injected into the soil through the grounding electrodes and produce a high potential gradient on the surface of the earth.[1]. In order to keep this potential gradient within safe limits, the grounding system used in the substations is in the form of a horizontal grid buried at a depth of about 2 feet and spread over the entire area of the substation.

A substation grounding system is designed to keep the step and the touch voltage within safe limits. These voltages are defined in the following paragraphs.

Step Voltage: The difference in surface potential experienced by a person bridging a distance of 1m with his feet without contacting any other grounded object.

Touch Voltage: The potential difference between the grounding system potential and the surface potential at the point where a person is standing is known as the touch voltage. The maximum touch voltage to be found within a mesh of a grounding grid is the mesh voltage.

Resistivity and Ground Resistance

The step voltage, the mesh voltage and the ground resistance are all proportional to the resistivity of the soil in the vicinity of the substation. Resistivity of the soil depends on the following factors[1]:

(i) Type of soil (sand, gravel, clay etc.);

(ii) Chemical composition of soil;

(iii) Concentration of salts dissolved in contained water;

(iv) Moisture content of soil;

(v) Temperature;

(vi) Closeness of packing.

As the temperature and precipitation vary, the resistivity of the soil will change. As the resistivity of the soil changes, so does the step voltage, mesh voltage and ground resistance of the grid.

For a particular substation and its grounding system, step voltage and mesh voltage are known functions of the substation ground resistance. With small currents used for measurements, it is easier to measure the ground resistance than the step and mesh voltages. In order to ensure that the station is safe for personnel in all climatic conditions, it is proposed to monitor the ground resistance of the substation grounding system.

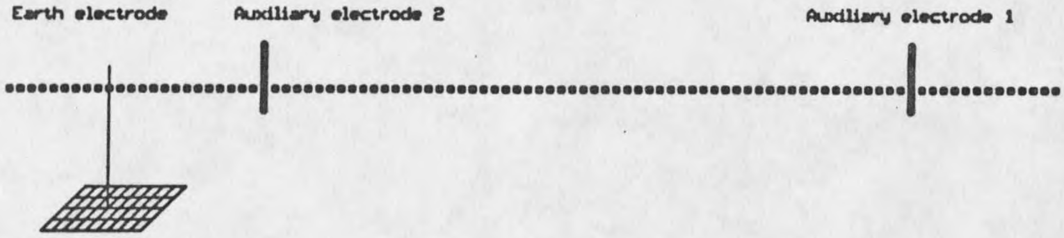
Ground resistance also affects the proper operation of the protective equipment. Monitoring of the ground resistance will help detecting the periods, if any, during which the protective equipment may not provide the required protection against ground faults or lightning.

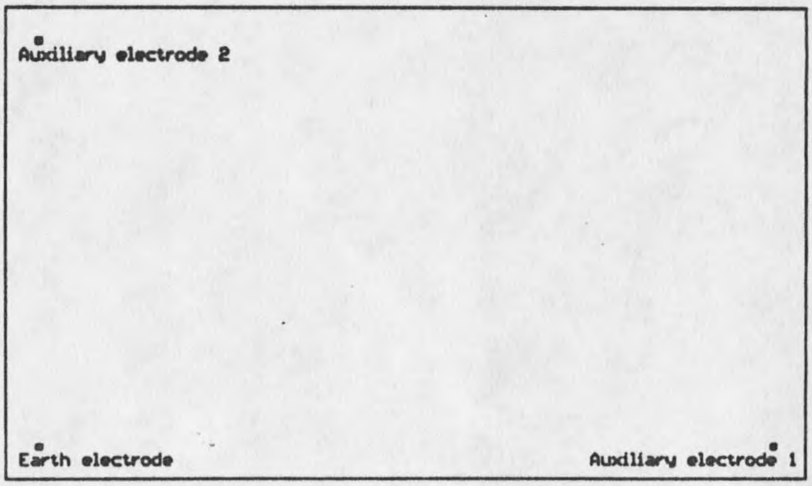
Basis of Measurement

Measurement of ground resistance is made using the three terminal method.[2]. The earth electrode (grid) is used as one terminal into which the test current is injected. The current is collected at a remote point using an auxiliary electrode, which constitutes the second terminal. Another auxiliary electrode (third terminal) is used to sense the voltage at a distant point.

Consider the setup of the earth electrode and the auxiliary electrodes as shown in figure 1. A known current is passed between the earth electrode and auxiliary electrode 1. The potential difference is measured between the earth electrode and auxiliary electrode 2. Both the auxiliary electrodes 1 and 2 are placed at a large distance from the earth electrode and from each other. The measured voltage

Figure 1: Setup for Measurements





divided by the known current will give the ground resistance. Auxiliary electrode 2 is considered to be at zero potential point. This method is used to eliminate the ground resistance of the auxiliary electrode 1 from the measurements.

Thesis Objective

The objective of this thesis is to document a system that has been developed to monitor the ground resistance. This instrument has been primarily necessitated by the fact that no instrument available in the market provides continuous monitoring of ground resistance. The system developed stores in memory the ground resistance measurements made at regular intervals of time. The voltage, current and time of measurement are also recorded with every reading. The features of the instrument developed are summarized below:

(i) Measurement of ground resistance after every 24 hours.

(ii) A range of 0-25.5 Ohms with a resolution of 0.1 Ohms.

(iii) Recording of potential of the grounding system, the ground resistance and the time of measurement.

(iv) Switching of alarm or some visual indication whenever the ground resistance exceeds a preset value.

(v) Error not more than 5%. Ground resistance measurements do not require a high level of accuracy and an error of less than 5% is adequate.

(v) Automated operation.

Thesis Organization

Chapter 1 introduces and defines the problem. Chapter 2 details the experiments leading to the development of the instrument and the setup used at the substation for the initial testing.

Chapter 3 gives details of the analog and digital part of the hardware involved in the design of the instrument. Detailed block schematics of the analog part are provided and discussion is centered on the use of specific blocks in the schematic. A hardware circuit diagram of both the analog and the digital part is also provided.

Chapter 4 covers the overview of the software used in the data acquisition and data processing system of the instrument. Detailed flow charts of the software are provided. The software assembly code used is included in the appendix. Details of the software used to enhance the accuracy of the calculations are also included.

Chapter 5 tabulates the measurements made both at the substation and in the laboratory using the ground resistance monitor. An error analysis is included.

The thesis is concluded with a brief summary of the work done, the results obtained, its benefits, and suggestions for further development in chapter 6.

CHAPTER 2

DEVELOPMENTAL APPROACH

Measurement Setup

The measurements for the development and testing of the microcontroller-based ground resistance monitor were made at the East Gallatin substation of the Montana Power Company in Bozeman, Montana. The experimental setup is shown in figure 2. Eight feet long ground rods were used as auxiliary electrodes buried vertically at locations shown in the figure. Insulated underground leads from the auxiliary electrode were run to a point close to the station fence. The three leads (one from the ground electrode and two from the two auxiliary electrodes) were made available for making measurements with the three terminal instrument placed near the fence of the station.

Because of the limited area of the property of the Montana Power Company around the switchyard, the auxiliary electrodes could not be placed at a very large distance. This would result in a lower measured value of the ground resistance by any instrument used on this setup.

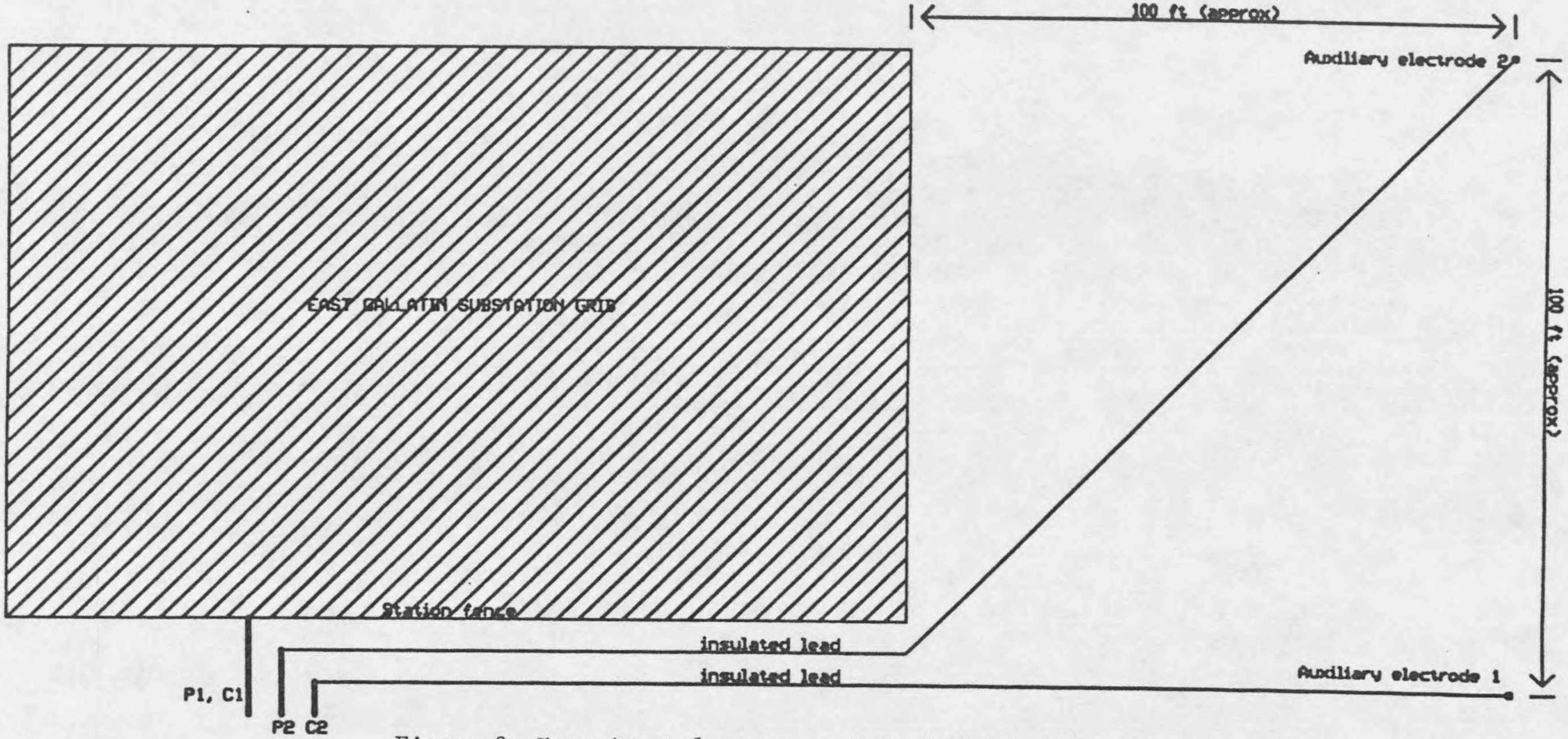


Figure 2: Experimental setup at East Gallatin Substation

Experimental Measurements

Measurements for the development of the instrument were carried out using the setup shown in figure 3. A sine wave generator was connected to the grid through a standard resistor. Auxiliary probe 1 was connected to the sine wave generator ground thus completing the circuit. The voltage drop, V1, across the series combination of standard resistance, Rstd, and ground resistance, Rgnd, was measured using an oscilloscope. The voltage drop, Vgnd, across the ground resistance, Rgnd, was also measured using the oscilloscope. The voltage drop across the standard resistance, Vstd, was calculated as V1-Vgnd and the current injected into the grid (Ignd) was calculated by using Ohm's law for the standard resistance. Ground resistance was then calculated using Ohm's law (Vgnd/Ignd).

This procedure yielded errors due to the noise generated due to the harmonic currents in the ground. The measurements were made using different frequencies. After a series of trials, it was found that 2 KHz was the minimum frequency for which interference from the ambient voltages was less. It has to be recognized that higher frequencies would bring in errors due to the inductance of the leads connecting the electrodes to the instrument.

A high-Q active band pass filter was used to further decrease the interference because of the stray voltages in the

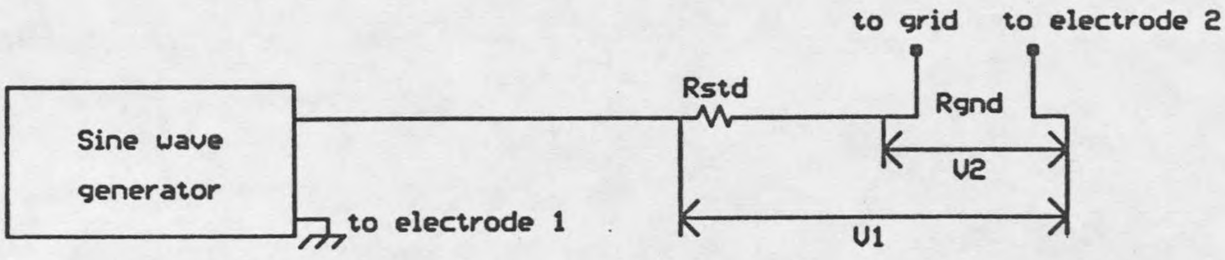


Figure 3: Equipment setup for measurements

ground.

Even with the use of 2 KHz frequency and a high-Q band pass filter, the observations indicated the presence of an offset in the measured voltages. This offset was caused by the leakage currents that flow from the grid into the surrounding earth. These leakage currents may produce potentials of the order of tens of volts. Though the leakage currents change with time, rapid changes in leakage currents are not common. Effect of the leakage currents can be eliminated by the following procedure:

$$Rgnd = Vgnd/Ignd$$

where Rgnd is the ground resistance, Vgnd is the voltage across the ground resistance and Ignd is the current through the ground resistance at 2 KHz. Both the standard resistance and the ground resistance are in series. Hence,

$$Ignd = Istd = Vstd/Rstd$$

where Istd is the 2 KHz current through the standard resistance, Vstd the voltage across the standard resistance and Rstd is the value of the standard resistance. So,

$$Rgnd = Vgnd*Rstd/Vstd$$

If the leakage currents in the ground produce a stray voltage Vlkg, the measured ground voltage is modified and the measured ground resistance is given by:

$$Rgnd = (Vgnd + Vlkg)*Rstd/Vstd$$

This offset caused by the leakage currents was eliminated from the calculations by using the aggregate method. The aggregate method involved the use of two different values of current which produced two different values of voltage across the ground resistance. The difference in voltage and the difference in current was then used to calculate the ground resistance. Assuming that the leakage current and hence Vlkg does not change during the time of the two measurements,

Rgnd = ((Vgnd2 + Vlkg) - (Vgnd1 + Vlkg))*Rstd/
(Vstd2 - Vstd1)

Rgnd = (Vgnd2 - Vgnd1)*Rstd/(Vstd2 - Vstd1)

This eliminates the effect of leakage voltages in the calculation for ground resistance. Following the principle of measurement as given above, the microcontroller-based ground resistance monitor was designed and constructed.

CHAPTER 3

HARDWARE

Analog Hardware

In order to measure the resistance of the ground, the procedure used has a standard resistance in series with the ground resistance. A known current is passed through the series combination and the voltage across the standard resistance and the ground resistance are recorded. The data processing system, which consists of the microcontroller and its peripherals, then uses the voltages recorded and the value of standard resistance to calculate the value of the ground resistance. The feeding point for the current is different from the point where the voltage across the ground resistance is measured . The method used was introduced in chapter 1 (basis of measurement).

The block diagram of the circuit is shown in figure 4. The complete circuit diagram is shown in figure 5. The sine wave used is generated by an oscillator (BB4423). The output frequency of the oscillator is adjusted to 2000 Hertz. The frequency used was determined after intensive testing

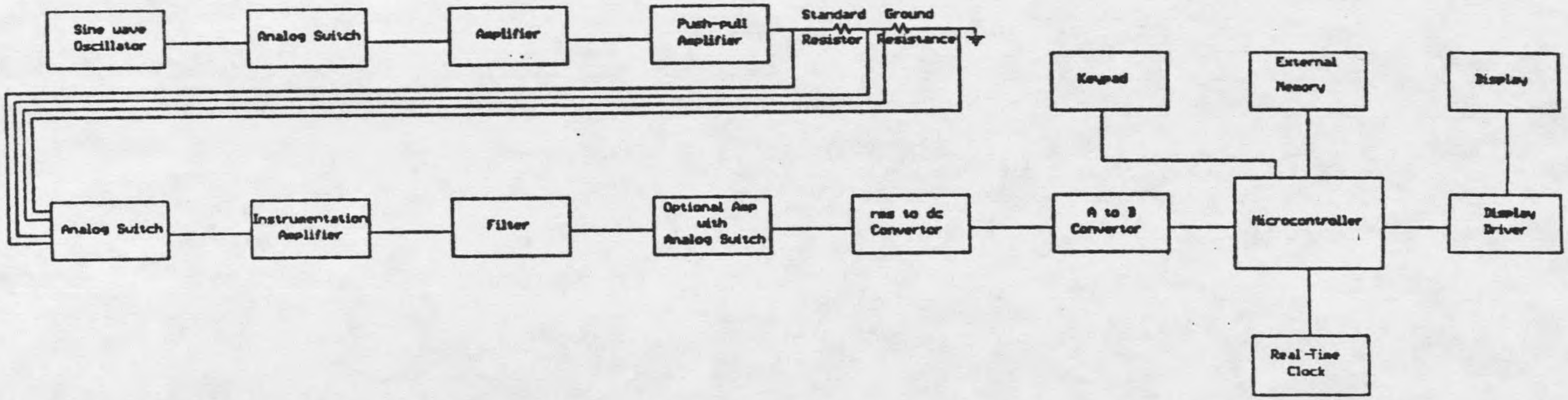


Figure 4: Hardware Block Diagram

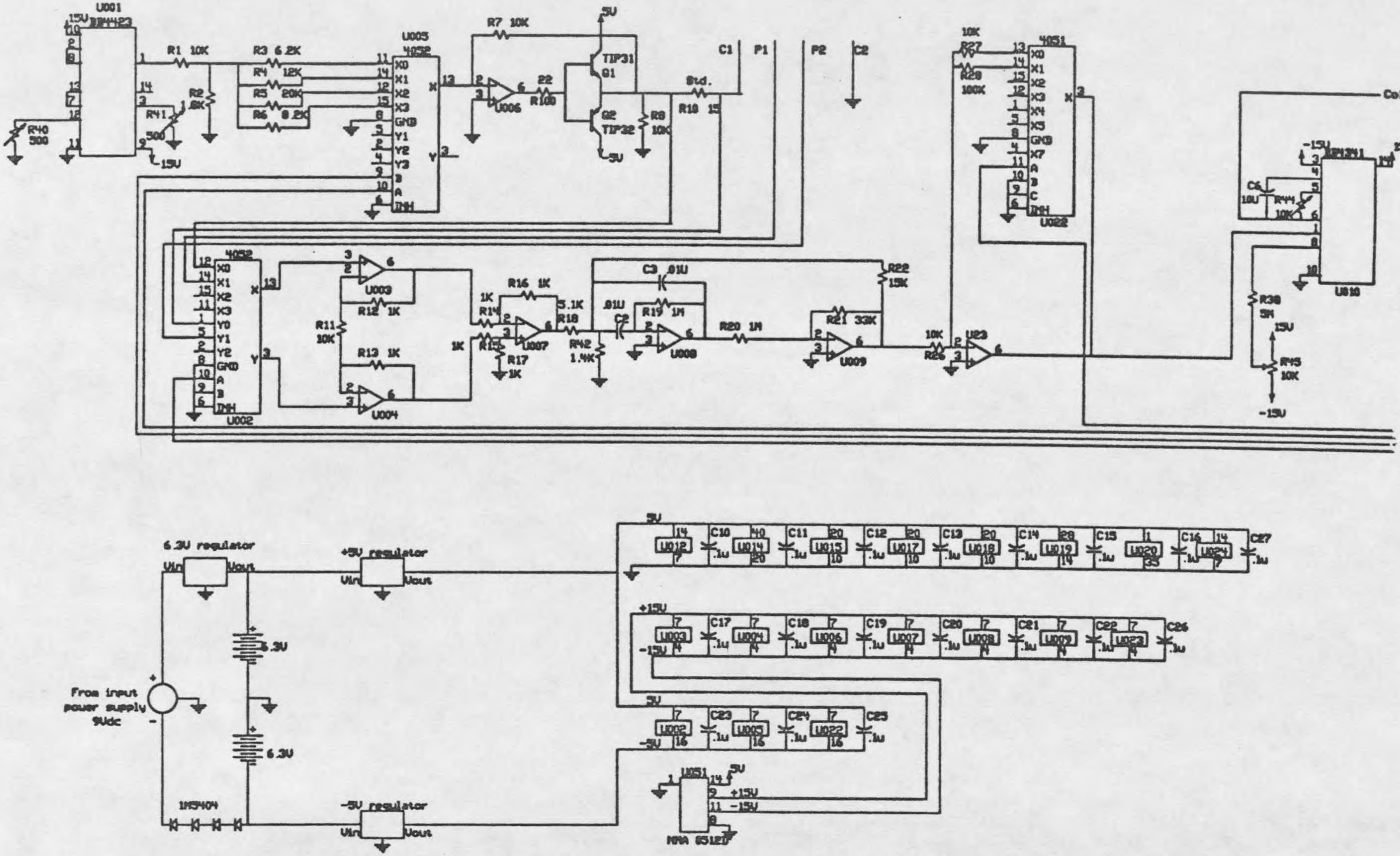


Figure 5: Analog Circuit Diagram

(chapter 2).

The output of the oscillator is a sine wave whose amplitude is 20 Vp-p. This wave form is input to a voltage divider that divides the voltage by a factor of six. This stage could however be disposed of and the necessary voltage reduction can be brought about by adjusting the gain of the amplifier in the next stage.

The output from the voltage divider is input to an amplifier stage. The amplifier stage has adjustable voltage gain which can be controlled by the microcontroller using an analog switch and resistors. The analog switch changes the input resistance to the opamp thus changing the gain of the amplifier stage. The analog switch has four inputs and hence the gain can be adjusted to four different levels. Such an adjustment enables the microcontroller to adjust the output voltage by varying the gain. A change in the output voltage would result in a change in the current flowing through the resistors (the serial combination of standard and ground). The microcontroller thus increases the current through the ground resistance whenever it is below a minimum specified level that is usually specified by a keypad input to the microcontroller.

As the microcontroller-based ground resistance monitor will be used in substations where there are tens of volts due to leakage currents flowing from the grid into the surrounding earth, there is a possibility of error being introduced in the measurements made. To minimize the error, from the leakage currents, two sets of measurements would be made for both the voltage and the current and the difference between the two readings utilized to calculate the ground resistance. This would eliminate the voltage caused by leakage currents flowing from the grid that was common to both the readings. These readings, it may be mentioned, are made for different analog switch settings which in turn means different voltages across each of the resistances. The analog switches are operated by the microcontroller.

The output of this amplifier is fed into a push-pull amplifier. This stage provides up to 100 mA current without distorting the output. The output from the push-pull amplifier is fed into the standard resistor and then into a ground electrode (C1). An auxiliary electrode (C2), buried in the ground at about 100 feet from the ground electrode, is connected to the chassis ground of the ground resistance monitor.

As described in chapter 2, the current probes introduce a high current density area around C1 and C2. To get a fairly accurate value of ground resistance, it is essential that the voltage probe be placed away from this area with high current density. So, the voltage probe (auxiliary electrode P2) is placed at a distance of about 100 feet from either of the other electrodes. This ensures that the voltage probes just pick up the true voltage introduced by the probe C1 and not the apparent voltage because of the high current density caused by the current probes. These voltage probes are connected to the inputs of the analog switch which will be discussed later.

The transistors used for the current source are power transistors and are capable of handling up to 350 mA. Though the minimum current used for the operation of the circuit is lower than that, the current (depending on the analog switch setting) may be as high as 100 mA. These power transistors need heat sinks to handle currents that high. The transistors used in the circuit are provided with heat sinks for the purpose.

An instrumentation amplifier[3] is used to amplify the voltages across the standard resistor and the 'ground resistance' (the resistance formed by the resistance between the two probes (P1 and P2) placed in the ground). The input to the instrumentation amplifier is through a two-channel DPDT analog switch. This analog switch is operated by the microcontroller. The microcontroller connects the instrumentation amplifier to amplify the voltage across the standard resistor and then after measurements are made and the readings recorded, the microcontroller connects the instrumentation amplifier to amplify the voltage across the probes (P1 and P2) in the ground. This is accomplished by using the analog switch.

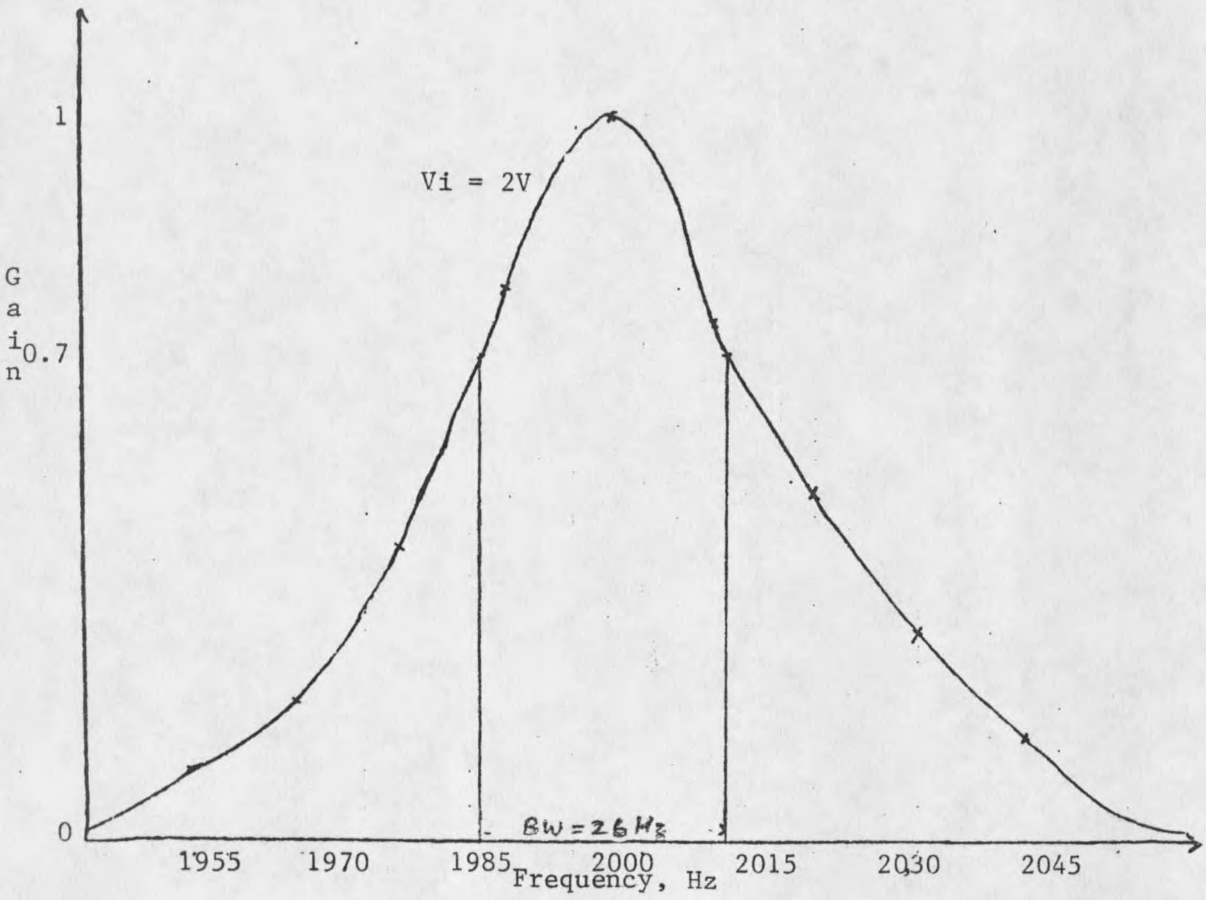
The analog switches used are 74LS4052 ICs and have a maximum of 100 Ohms resistance.[4]. This switch resistance varies with the input voltage. The variation of the resistance with the input voltages used in the monitor is less than 1 Ohm. As the difference in voltage is taken for the calculation of the ground resistance, the effect of this resistance (worst case) in the calculation of ground resistance is less than about 0.3 Ohms. The inaccuracies introduced in the calculation by the analog switch are further quantified in chapter 5 under error analysis.

An interesting case of using a simple inverting or a non-inverting amplifier instead of an instrumentation amplifier could be made at this point. An instrumentation amplifier has a better common mode rejection ratio (CMRR). This higher CMRR would result in the reduction of the common mode noise that is present in the measurements thus making the measurement relatively less susceptible to errors resulting from common mode noise. This justified the use of an instrumentation amplifier instead of a simple inverting or a non-inverting amplifier.

Though the operating frequency for the circuit is not in the near vicinity of any of the strong harmonics of the 60 Hz fields, this still does not make the system interference free. The harmonics of 60 Hz still introduce errors in the measurements. It is essential that the measurements be made only for the signal generated by the instrument. It is necessary to make sure that the harmonics present in the vicinity of the substation do not affect the measurements taken. To ensure this, a filter is used. The filter is an active band pass filter with a high Q.[5]. The pass band of the filter is extremely narrow (about 26 Hz) and is centered around 2 KHz which is the frequency of operation of the ground resistance monitoring system.

When the value of resistance between the ground electrode (P1) and auxiliary electrode 2 (P2) (the ground resistance) goes below 5 Ohms, the voltage difference between the electrodes is bound to be low. In such cases, the readings from the data acquisition system become unreliable. To avoid this, an amplifier is used to amplify the voltages across the resistors whenever the resistance between the two probes goes below the 5 Ohm threshold. The gain of this amplifier is controlled by the microcontroller through the activation of an analog switch. The gain of the amplifier is set to 10 when the resistance between probes is less than 5 Ohms. When the resistance between the two probes is greater than 5 Ohms, then the gain of the amplifier is set to one. The amplifier is used only for the voltage across the ground resistance and not across the standard resistor. The voltage across the standard resistor stays above 0.5V during normal operation and does not introduce any errors. This amplification process is carried out to enable the analog to digital convertor to use a broader range of output instead of just the two or three least significant bits.

Figure 6: Frequency response for an input of 2V



An rms-to-dc convertor (BB4341) is used to convert the rms value of the input voltage to a dc value. This convertor outputs a dc voltage that is equivalent to the rms value of the input voltage. The rms-to-dc convertor should go through at least one full time period for the readings to be fairly accurate. As the operating frequency of the circuit is 2 KHz, the time period is 0.5 mS. A higher waiting period for the rms-to-dc convertor to settle would improve the accuracy of the convertor. In the case of the monitor, the time allowed for settling is about one second. This gives the rms-to-dc convertor about 2000 cycles to sample.

This dc value from the output of the rms-to-dc convertor is converted into a digital signal by the analog to digital convertor. The analog to digital convertor used is a 10-bit successive approximation convertor (CA3310A). The two least significant bits of the analog to digital convertor are left unused owing to port limitations of the microcontroller. The analog to digital convertor takes approximately 10 microseconds to complete the conversion. An external R-C circuit is used for the generation of the clock for the analog to digital convertor. The chip has a built-in track and hold. The digital voltage from the analog to digital convertor is then input to the microcontroller for further processing.

The instrument is also equipped with a auxiliary power supply for continuous functioning in case of a power outage. This 6V acid gel rechargeable battery used for the instrument is provided with a trickle charge circuit. The input voltage is + and - 9 V dc and the input is followed by 4 diodes in series. The diodes used are 1N5404 diodes and have a current capability of 3A. These 4 diodes are used in series to drop the voltage down to about 6.3V so that it is adequate to charge the batteries. The output from the batteries are passed through regulators before it is connected to the main board. The associated circuitry is shown in figure 5.

Digital Hardware

The analog part of the hardware discussed thus far involved the data acquisition part of the microcontroller-based ground resistance monitor. The digital part of the hardware is used for the processing and storing of the data obtained by the analog hardware. The digital hardware, in certain cases, also controls the analog circuitry (gain of amplifiers, inputs to instrumentation amplifier). Besides, the digital part also obtains other pertinent information like the time and date of the measurement.

The digital part is totally controlled by the microcontroller. Port 0 of the microcontroller is connected, in a multiplexed fashion, to the keypad through the buffer, to the external RAM through the address latch, to the real time clock, to the display through the display driver and to the analog to digital convertor. Only one of the above

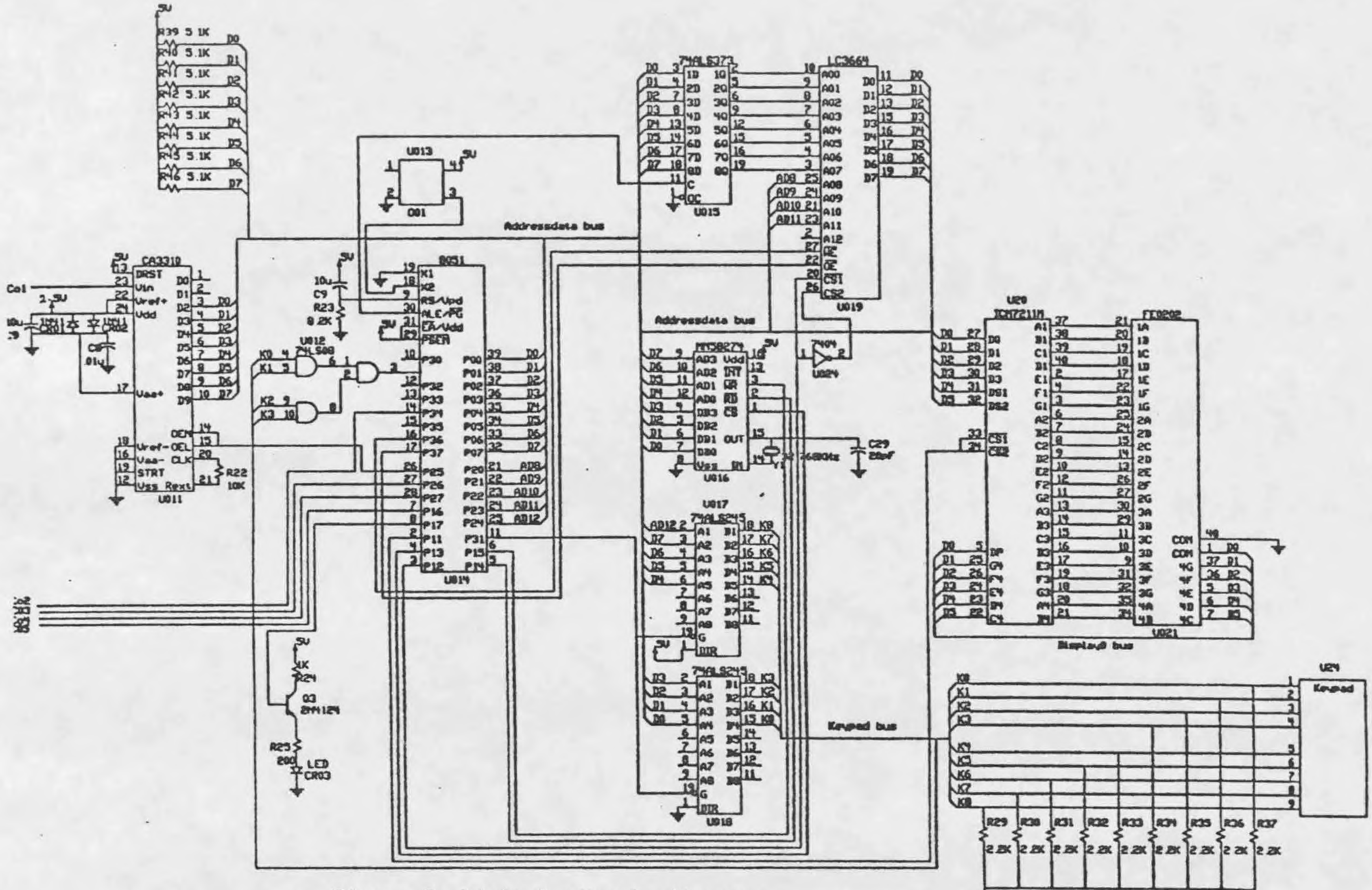


Figure 7: Digital circuit diagram

mentioned peripherals is enabled by the microcontroller at any given point of time.

The microcontroller used in the instrument is an Intel 8751.[6]. This chip belongs to the MCS 51 family of microcontrollers and has an internal program memory of 4K bytes and an internal data memory of 128 bytes. The internal program memory is an EPROM memory and is a read only memory. The internal data memory is a RAM memory and is a read/write memory.

The microcontroller can also use an external data memory of up to 64K bytes. The lower byte of address comes from port 0 and the higher byte of address comes from port 2. During accesses to external data memory, the lower byte of address is multiplexed with the data on port 0. The read and write outputs of the microcontroller are connected to the read and write inputs of the memory chip used. These signals are controlled by the CPU of the microcontroller and will be enabled or disabled depending on the instruction.

In the case of the ground resistance monitor, 4K bytes of external RAM is used to store the readings. This could however be expanded when the need arises. When using the external RAM a transparent latch with port 0 should be used for the least significant byte of the address. The ALE output from the microcontroller should be connected to the address latch input of the latch. This latch holds the lower byte of the address during external reads and writes when port 0 is operating in the multiplexed mode.

Among the peripherals used in the instrument, the keypad is the only source of any external input by the user to the microcontroller. The keypad is used to input the value of the standard resistance, the minimum current required for each measurement, and the preset maximum value of ground resistance. These entries should be made immediately after the power to the instrument is turned 'on'. This is because the microcontroller has all the memory locations in the internal RAM in an indeterminate state after power on reset. Not entering the above mentioned three values immediately after powering the circuit would lead to indeterminate readings (or in no measurement being made).

The keypad is also used to obtain the readings stored in the memory. The readings that can be obtained from the memory are: Voltage, Current, Resistance of all the measurements with the time of recording. The mean and standard deviation of the past 64 readings can also be obtained by using the appropriate keys on the keypad.

The keypad used is a matrix type keypad and has the supporting circuitry to generate an interrupt whenever a key is pressed. The keypad interrupt is disabled when the microcontroller is in the process of measuring the ground resistance.

The microcontroller measures ground resistance once every 24 hours. At the end of every twenty four hours, the software generates an interrupt. This takes the microcontroller off any other process (like outputing readings). The microcontroller then proceeds to measure the readings.

The standard resistance has a value of 15 Ohms. The microcontroller-based ground resistance monitor is capable of measuring ground resistance from 0 to 25.5 Ohms. When the microcontroller determines the ground resistance to be below 5 Ohms, the measurement process is repeated with the voltage across the ground resistance amplified by a factor of 10.

The idea behind the construction of the microcontroller-based ground resistance monitor is to warn the technicians about the ground resistance when it reaches dangerous levels. This dangerous level may be the level when the ground resistance reaches a point where it may prove to be hazardous to a person on or in the vicinity of the substation. Whenever the latest reading of ground resistance goes higher than a preset value (which is input into the microcontroller after power on reset) the microcontroller turns an LED on for a period of about 5 minutes. The output of a port pin is used to control a transistor which supplies the current for the LED. In certain cases where an LED is not strong enough to send a warning signal to the personnel, it may be replaced by an alarm or some other audio indication.

A four digit liquid crystal display (LCD) is used for obtaining readings from the microcontroller. It is connected to the microcontroller through a display driver. The microcontroller needs 6 lines (not including the enable lines) to display these four digits. The microcontroller outputs the number to be displayed (4 bits) along with the digit number (2 bits) for all the four digits and the display driver carries out the decoding involved. The display driver also holds the value of the number last output until a new number is written to it.

As mentioned earlier, a real time clock is essential to generate interrupts to enable the microcontroller to take a reading after every 24 hours. Apart from that, the real time clock is used to provide the time at which the measurement was made. The time information that goes with every reading include the hour, date and month of the measurement. The precise minute at which the measurement was made is not recorded as it is understood that the measurement was made at the beginning of the hour. The real time clock used is Motorola's MM58274 RTC. The importance of the time of measurement is discussed briefly in chapter 1.

While using port 0 of the microcontroller Intel 8751, it is essential to use pull-up resistors for all the pins. All other ports in the 8751 microcontroller have internal pull-ups. The pull-up FET in the port 0 output driver is used only when the port is emitting ones during external memory accesses. Otherwise, the pull-up FET is off. This makes port 0 lines used as outputs to be open drain lines. The pins in port 0 can be used as high impedance inputs when 1s are written to the pins as this leaves both the output FETs off.

CHAPTER 4

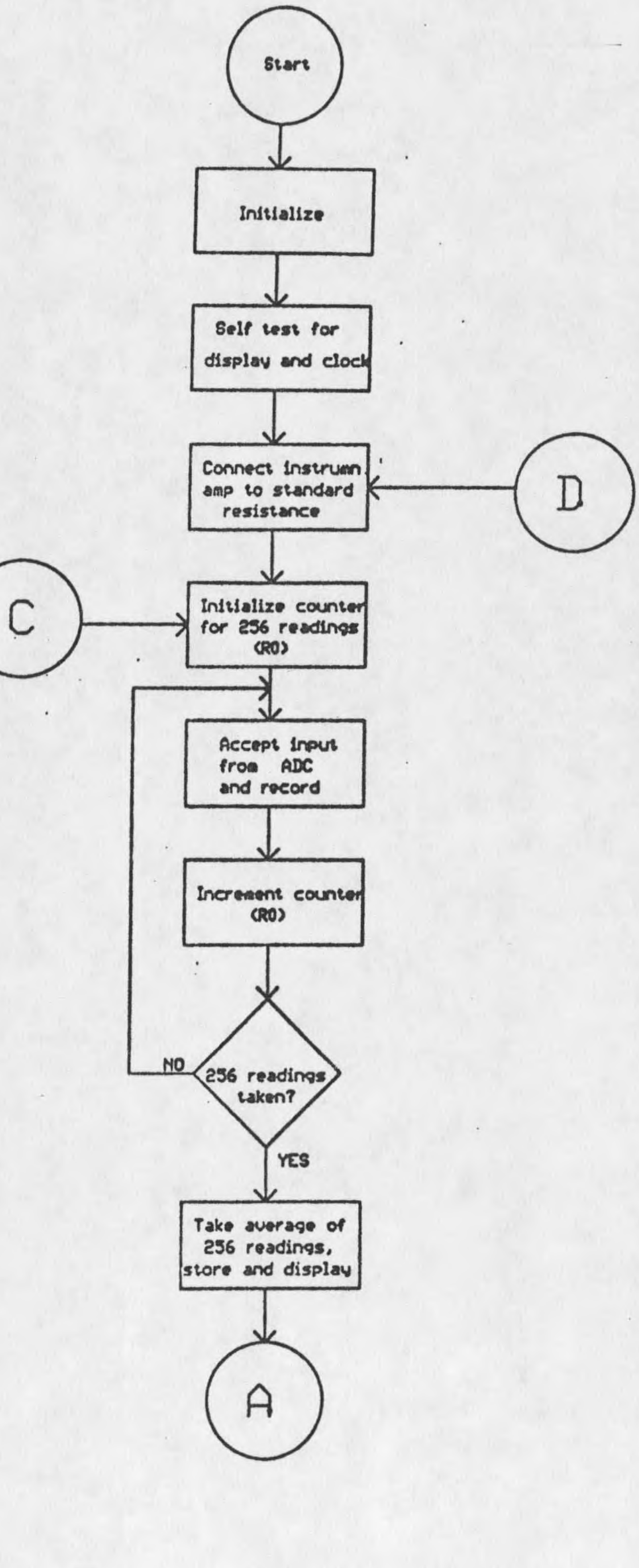
SOFTWARE OVERVIEW

The software used for the project is extensive with well over 3 kilobytes of code. The flow chart used for the software is shown in figure 8 and further discussion will be based on the flow chart.

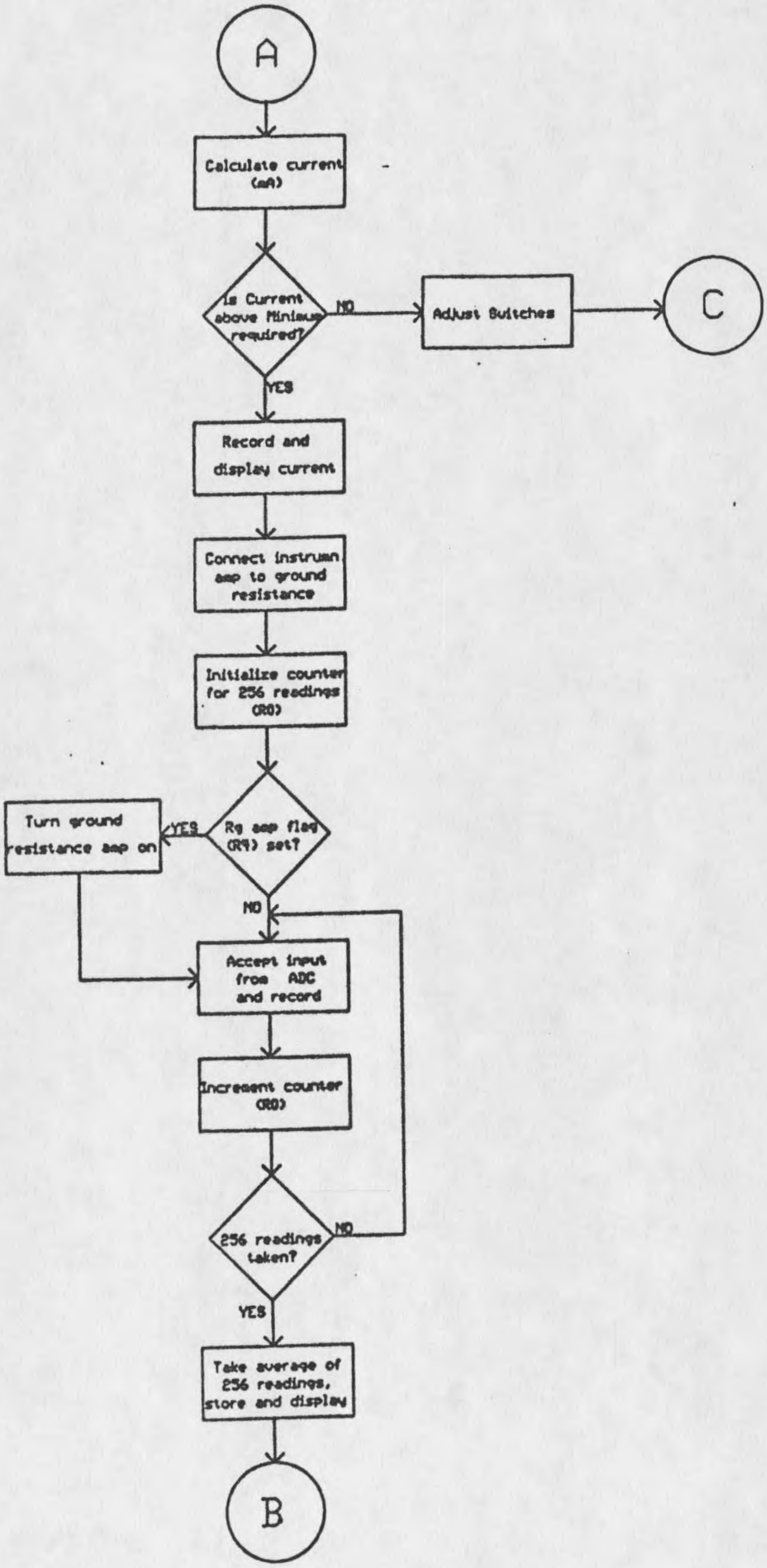
The software main routine starts with the initialization of the microcontroller and its peripherals. The initialization is followed by self test routine for the display and the real time clock (RTC). The display driver and display are tested by outputting a series of numbers starting at 255 and counting down to zero. The RTC is tested by piping the seconds output through the microcontroller to the display until it counts down to zero. After initialization and self test, the microcontroller idles in a loop. Further operation is initiated by interrupts.

The interrupts on the Intel 8051 microcontroller can be of three types: (i) External interrupt, (ii) Timer interrupt, and (iii) Serial port interrupt. Though the microcontroller permits three types of interrupts, the instrument uses only the external interrupt. External interrupts are used from two

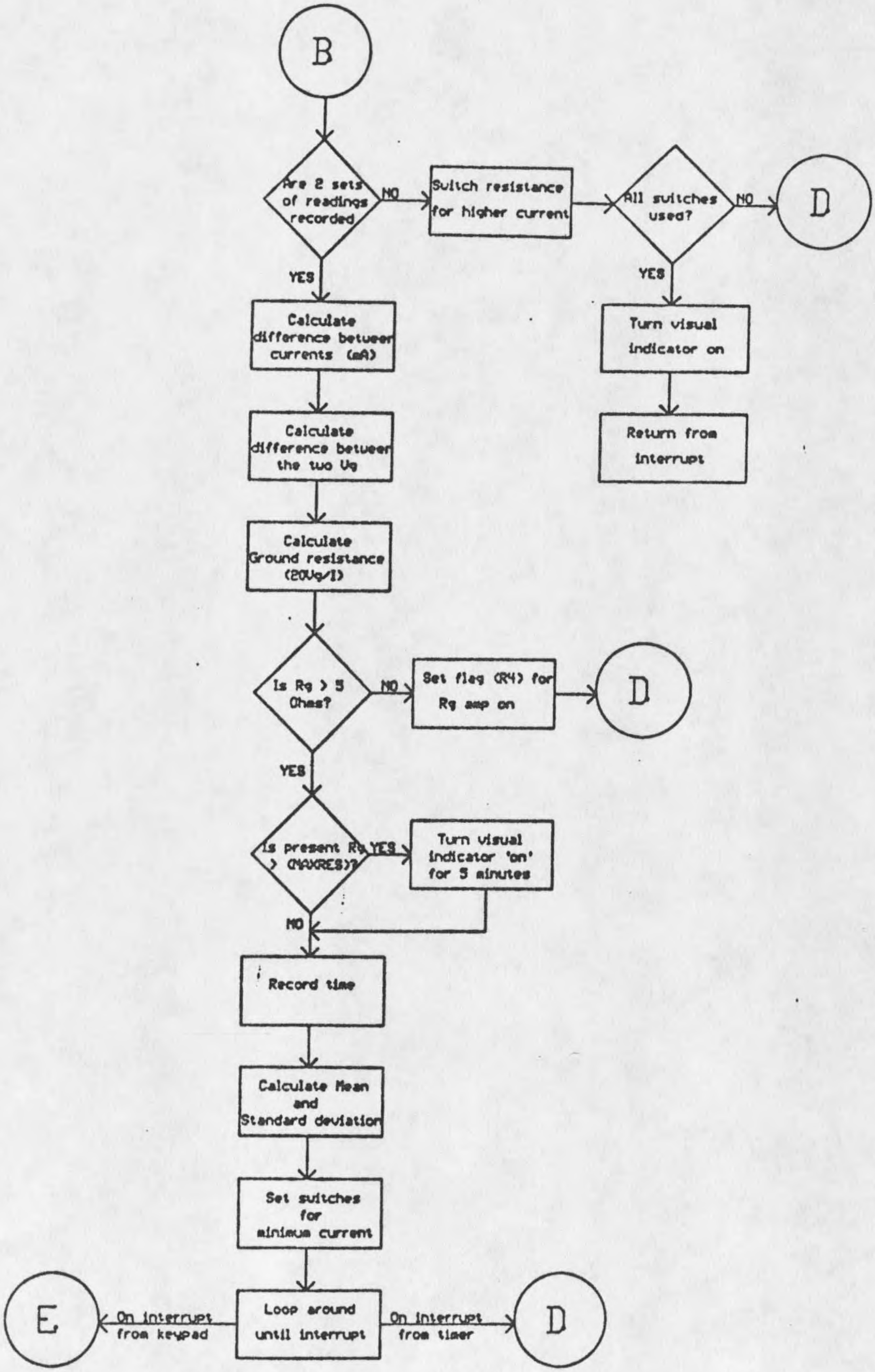
Figure 8: Flowchart of software



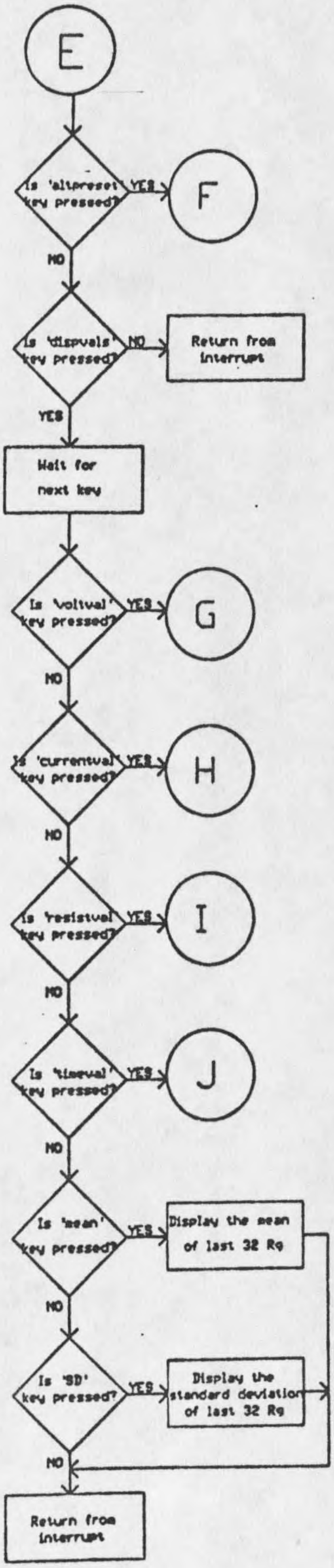
Flowchart of software contd..



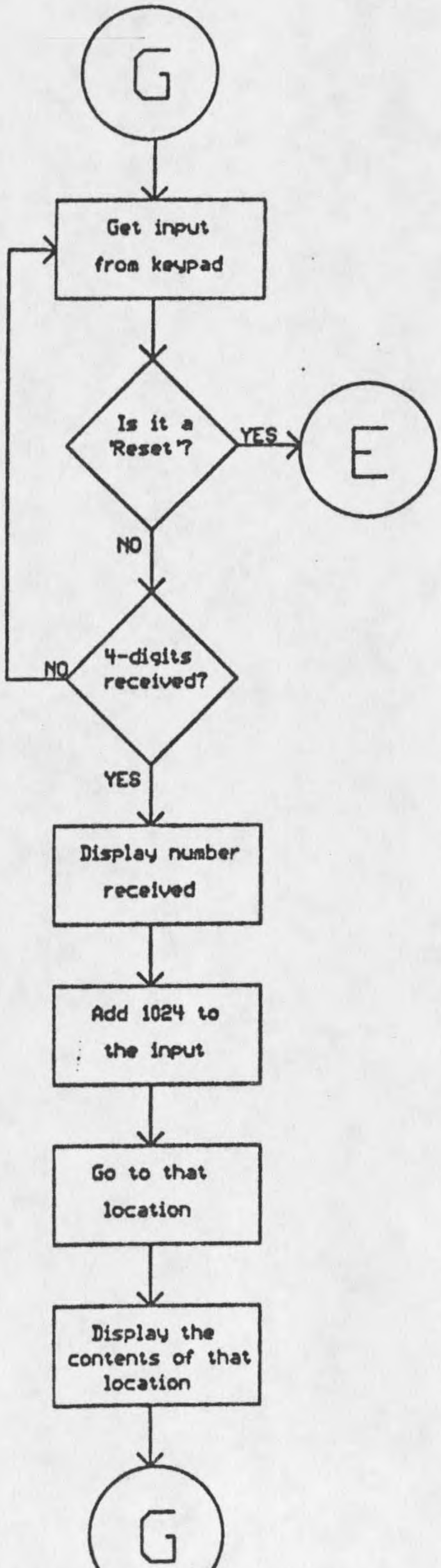
Flowchart of software contd..



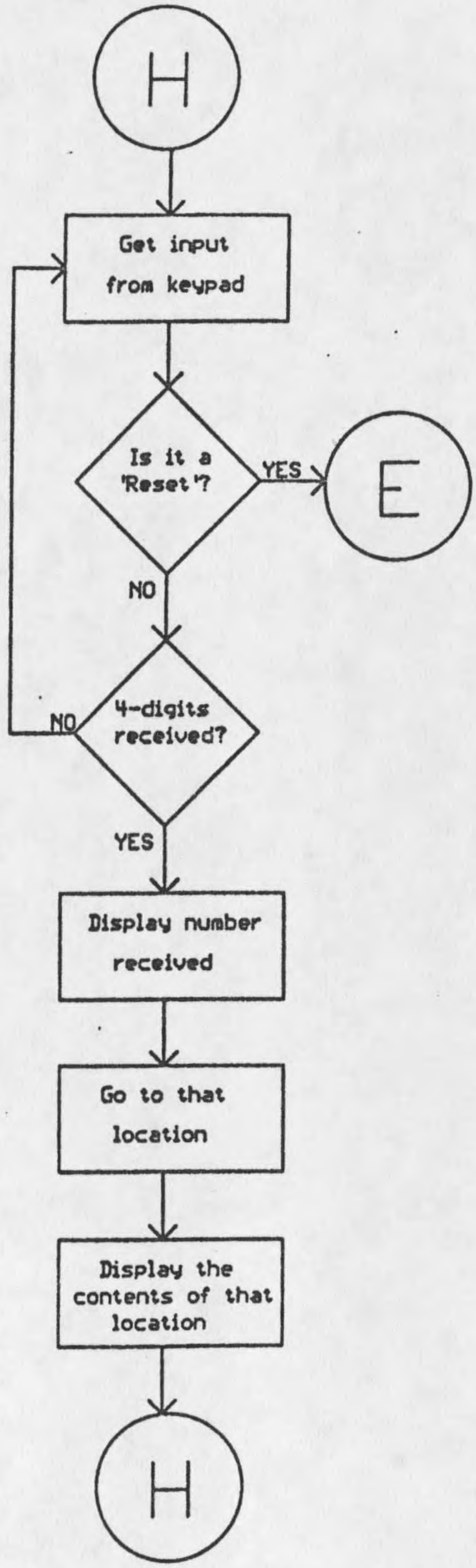
Flowchart of software contd..



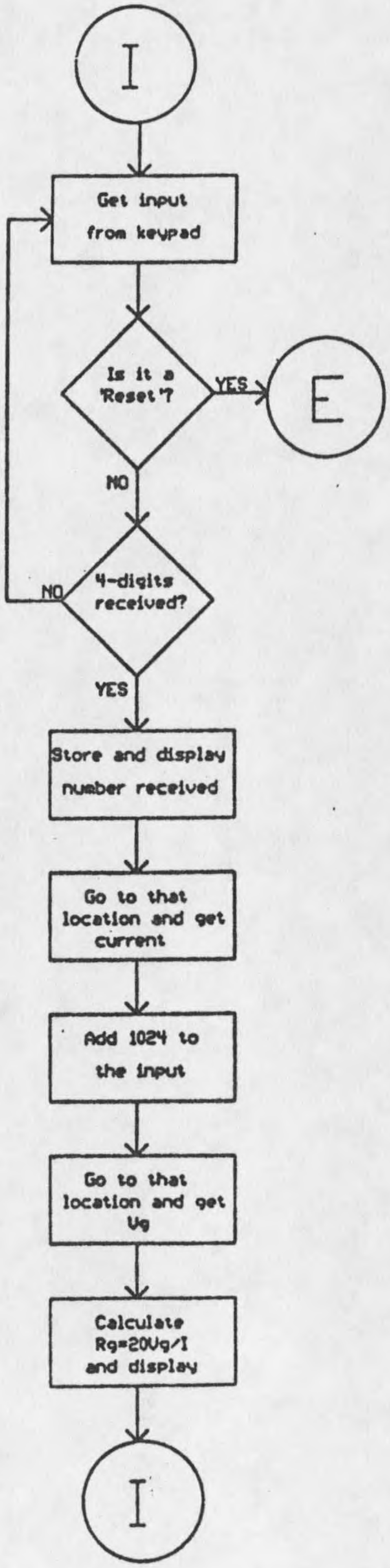
Flowchart of software contd..



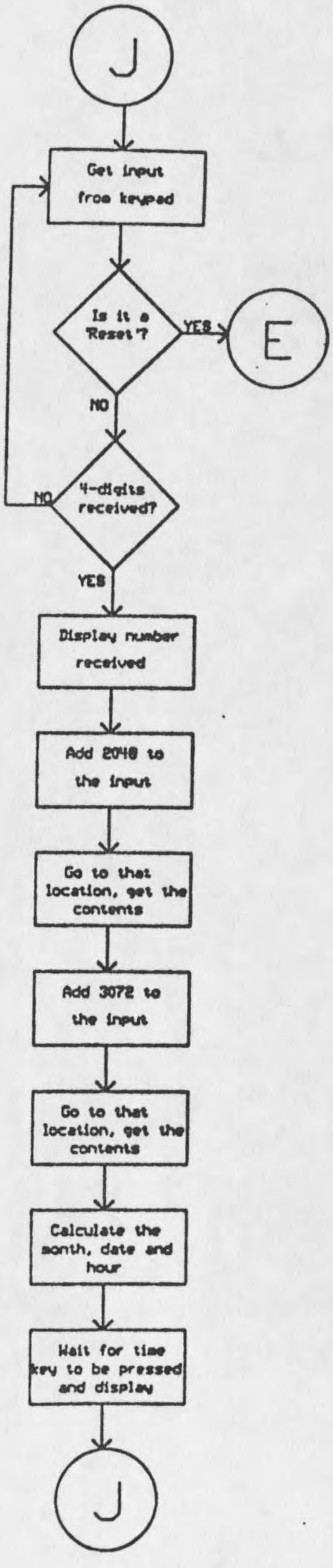
Flowchart of software contd..



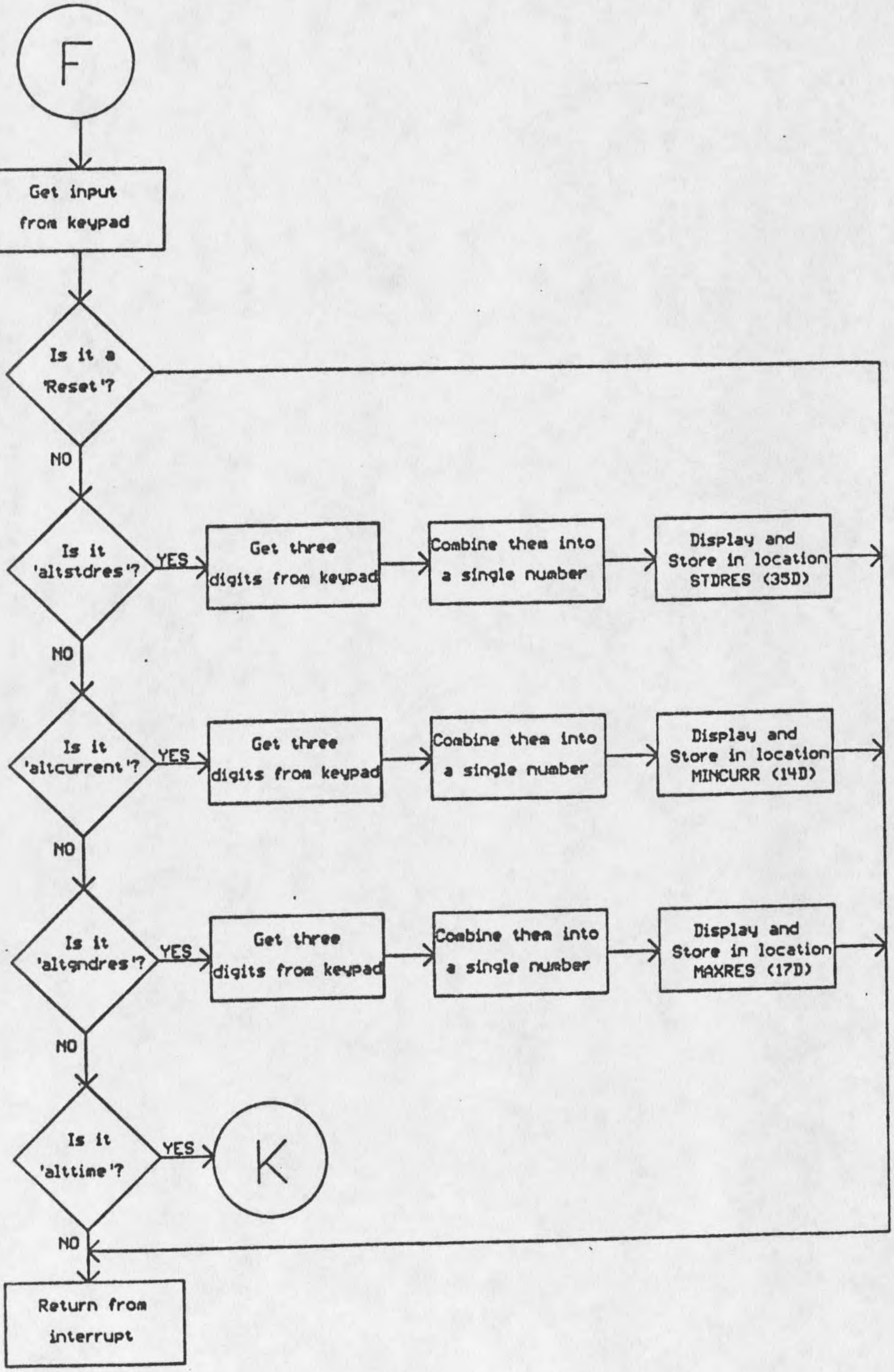
Flowchart of software contd..



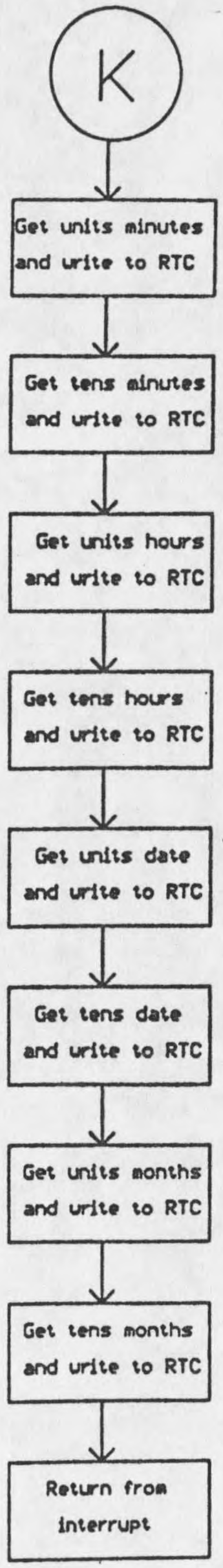
Flowchart of software contd..



Flowchart of software contd..



Flowchart of software contd..



different peripherals: One from the real time clock, and the other one from the keypad. The serial port is not used in this instrument and hence the serial port interrupt will not be used.

Real Time Clock Interrupt

The real time clock chip MM58274 is programmed to interrupt the microcontroller after every 60 seconds at which time the microcontroller reads the time from the RTC. If the time obtained is 0:00 hours, the microcontroller starts the process to make a measurement. If the time is not 0:00 hours, the software goes into the loop again.

Measurement of Resistance

The process of making the measurement includes the operation of the analog switches. The inverting amplifier analog switch is set for minimum amplification and hence minimum current through the series combination of standard and ground resistance. The instrumentation amplifier analog switch is connected across the standard resistor. If the current through the standard resistor is less than the minimum current required (input by the operator of the instrument during initialization), the analog switch is set for a higher amplification level and the procedure to check for minimum current is carried out again. The inverting amplifier analog switch has four different settings, three of which could be used to fulfill the minimum current specification. The fourth setting is saved for the second reading of the measurement (used in aggregate method). If minimum current is not achieved after using all the three available settings, the microcontroller turns an LED on indicating a hardware error in the instrument.

After minimum current is obtained by using the inverting amplifier analog switch, the microcontroller records the voltage across the standard resistor and calculates the current flowing through it. The instrumentation amplifier analog switch is then connected across the ground resistance and the voltage across the ground resistance is recorded. The inverting amplifier analog switch is set to the next amplification level and the voltages across the standard and ground resistances are measured and recorded.

The difference in voltages is used for calculations of ground resistance. This is done to eliminate the effect of leakage currents (discussed in chapter 2) in the calculation of ground resistance. For every voltage that is recorded by the microcontroller, 256 measurements are made and the mean of the 256 measurements used in the calculations.

After recording the readings, the microcontroller uses the ratio of the resistances and the ratio of the voltages across them to calculate the ground resistance. As we know the voltages across both the standard resistor (Vstd) and ground resistance (Vgnd) and the value of the standard resistor (Rstd), the ground resistance (Rgnd) can be calculated using the following equation:

$$Rgnd = (Vgnd/Vstd) x Rstd$$

The standard resistance has a tolerance of 10%. However the error in the measurement of the ground resistance due to the tolerance of the standard resistance is far less than that as the standard resistance was first verified for accuracy using the digital multimeter Tektronix DMM5110. The error caused by the tolerance and the deviation of the standard resistance value due to variation in temperature are discussed in chapter 5 (error analysis).

After the calculation of ground resistance, the microcontroller compares the present value of ground resistance to check if its greater than the preset value of ground resistance. The preset value is input into the microcontroller by the operator during initialization. If the present value of ground resistance is higher than the preset value, then the microcontroller turns an LED on. The LED will stay on until the ground resistance goes below the preset value.

After going through the process of calculating the resistance, the microcontroller then enables the real time clock and records the time of measurement. The time recording is two bytes long and has the hours (5 bits), months (4 bits) and the date (5 bits) of measurement. One byte contains the hours (5 bits) and one bit of the date component (the date component has a total of 5 bits), the other byte has 4 bits of the date component and the months (months component has 4 bits). This time information occupying two bytes for every reading recorded is unpacked when the operator requests the time output from the microcontroller. Minutes are not recorded for the simple reason that readings are taken once every twenty four hours and the minutes reading will be zero.

After recording the time of measurement, the microcontroller now proceeds to calculate the mean and standard deviation of the most recent 64 readings of ground resistance. However, due to the inherent limitations of the microcontroller, the square of the standard deviation is recorded instead of the standard deviation itself.

The mean is calculated using a simple process. The total of the most recent 64 readings of ground resistance are added together in two locations and the division is carried out by shifting the 16 bit quantity right 6 times which is equivalent to dividing the number by 64 in binary arithmetic.

After all the calculations and recording described above are over, the microcontroller then proceeds to store the values of voltage, ground resistance and time in external data memory. The mean and standard deviation are stored in the on chip random access memory (RAM) of the microcontroller. The external data memory chip used is capable of storing about 8 kilobytes but only 4 kilobytes of that memory space is used. The first kilobyte stores the value of current, the second kilobyte stores the voltage and the third and fourth kilobytes store the time.

To store the most recent values the microcontroller has to go through a window shifting routine. This window shifting routine copies all the readings in the external memory to the next location and then copies the latest reading at the top of the pile. The results in the most recent readings being stored at 0000.

After storing the values measured, the microcontroller enables all the appropriate interrupts and goes into a loop until it receives another interrupt.

User Input and Retrieval

The keypad interrupt is used for two purposes: To alter the preset values, and to display the recorded values.

A matrix type keypad with 20 keys is used. The layout of the keypad is shown in the appendix. It is set up so that one of the lines go low as soon as a key is pushed. The accompanying hardware then interrupts the microcontroller which then vectors to a service routine. The interrupt is caused by using logic circuitry on the printed wired board (PWB) of the ground resistance monitor. The real time clock interrupt has a higher priority over the keypad interrupt and hence if the keypad interrupts the microcontroller while it is processing the real time clock interrupt, the keypad interrupt will not be processed. The user has to retry using the keypad after about 2 minutes as the interrupt generated by the keypad is not stored.

As discussed earlier, four different values have to be input into the microcontroller immediately after turning the microcontroller on. Those values are: Standard resistance, minimum current, maximum ground resistance before alarm should be activated, and the time (month, date, hour and minutes). This process can be initiated by pushing the 'altpreset' key.

When the microcontroller receives a keypad interrupt, it scans the keypad and determines which key is pushed and vectors to the appropriate subroutine. The microcontroller software works under the assumption that the first key to be pushed would either be the 'altpreset' or 'displayvals' key. When neither of these keys is the first to be pushed, the microcontroller returns from the interrupt. The software enables the row containing these two keys and it is essential that the first key to be pushed is one of these two keys.

The 'altpreset' key is used to alter the preset values. All numbers entered shall have three digits. The standard resistance used is 15 Ohms. The minimum current and minimum ground resistance is input depending on the nature of the soil in the vicinity of the substation. This flexibility is provided to adapt the instrument for different substations with varied levels of ground resistance. However, the standard resistance used has to be a whole number without any digits after the decimal point.

The 'dispvals' key, as the name indicates, is used to have the microcontroller display the readings taken, the resistance calculated, and the mean and standard deviation of the most recent 64 resistance readings. The microcontroller is capable of storing up to a total of 1024 readings. Each of these 1024 readings have a voltage, resistance and time associated with it. So if the operator needs the 1012th reading of voltage, the 'dispvals' key shall be pushed followed by the 'voltval' key and then the keys 1, 0, 1, 2. The microcontroller then outputs the appropriate voltage reading. To get the corresponding current reading, the 'reset' key is pressed followed by 'displayvals' key. Now, the 'currentval' key is pressed followed by the numerals 1, 0, 1, 2. The microcontroller then outputs the 1012th reading of current. To get the time when the 1012th reading was taken, the same procedure is adopted, this time with the 'timeval' key. To obtain the mean or the standard deviation of the most recent 64 readings, the operator has to hit the mean or the SD key on the keypad.

When reading number 4 is required by the operator, the numbers 0, 0, 0, 4 have to be input at the appropriate time rather than just inputting 4. If this procedure is not followed, the microcontroller goes into a loop waiting for the next number to be input. 1023 is the oldest reading recorded in each category and 0000 is the most recent reading in each category. Any reading requested must be between these two numbers.

If the operator needs to get adjacent readings in the same category, say voltage, it is not necessary to enter 4 digits each time. Two additional keys are provided one for 'increase' and the other for 'decrease'. In effect, after obtaining one reading, the next reading can be obtained by just pushing the increase button.

In the calculation of ground resistance, the voltages used are obtained from the analog to digital convertor (ADC). The ADC outputs digital value of 255 for an input of 2.5 V. This does not affect the calculation of the ground resistance as the ration of voltages are used and the factor involved cancels out. But this factor has to be taken into account in the interpretation of the voltage output by the microcontroller.

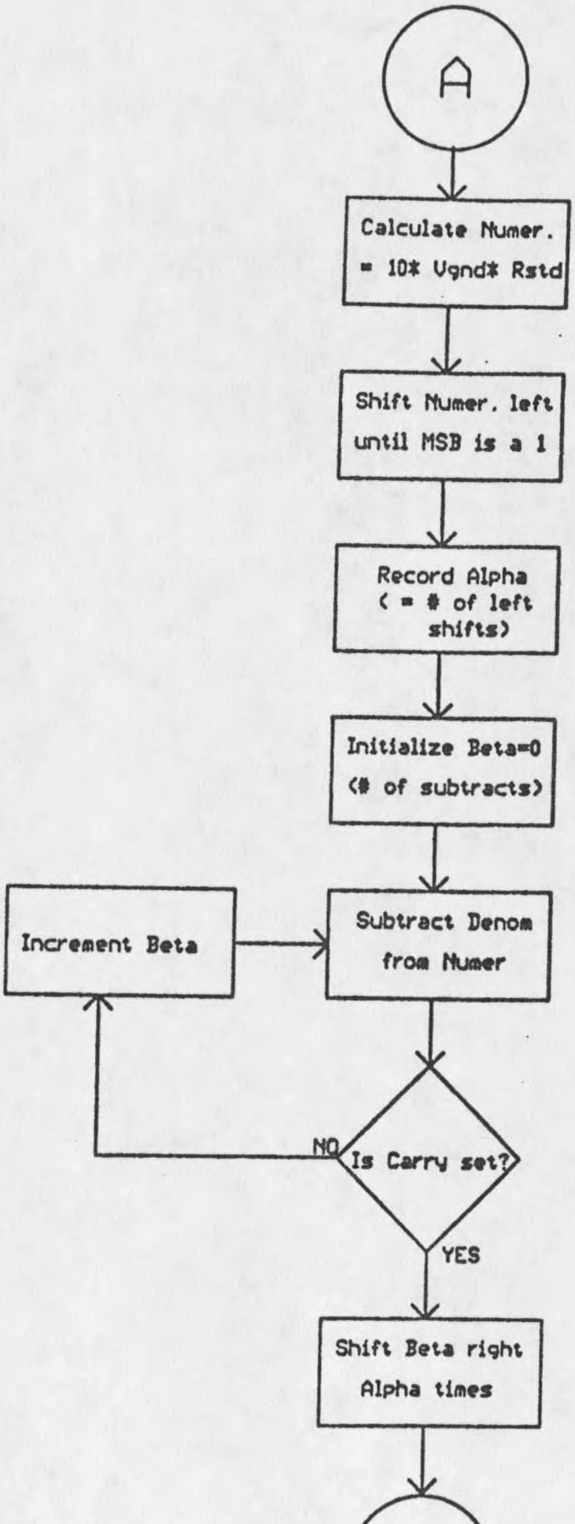
The process for the calculation of the ground resistance involves the use of the ratio of the voltages across the standard resistor and the ground resistance and the known value of the standard resistance. The ground resistance displayed is multiplied by a factor of 10 to improve the resolution of the instrument to 0.1 Ohms. This introduces a multiplication by a factor of 10 in the equation below:

$$Rgnd = 10*(Vgnd/Vstd)xRstd$$

= (10*Vgnd*Rstd)/Vstd

The process used to carry out this calculation is presented in the form of a flow chart in figure 9. The numerator on the right hand side (10*Vgnd*Rstd) of the equation is calculated. The calculated number is then shifted left as many times as possible until the most significant bit (MSB) is a 1 and the number of left shifts (alpha) are recorded. The denominator (Vstd) is subtracted from the numerator as many times as possible until such subtraction results in a carry. This is equivalent to division of the numerator by the denominator. The number of subtracts (quotient of the division) are recorded as beta. Beta is now shifted right alpha times. The resulting number is the ground resistance. The calculations are carried out using multiple precision arithmetic.

Figure 9: Calculation of ground resistance as a ratio of voltages



CHAPTER 5

MEASUREMENTS

Lab Measurements Using Ground Resistance Monitor

The microcontroller-based ground resistance monitor was tested for accuracy in the laboratory environment using known values discrete resistors. The terminals C1 and P1 of the monitor were shorted and connected to one end of the resistor. Terminals C2 and P2 were shorted and connected to the other end of the resistor. The results obtained using the ground resistance monitor were compared with the digital multimeter Tektronix DM51107.[7]. The resistance readings obtained using the monitor, the resistance readings obtained using the multimeter and the difference in resistance readings as a percentage of the full scale reading (25.5 Ohms) of the monitor are shown in table 1.

Noise and Error Analysis

Ground resistance is calculated using the formula in equation(1)

$$R_{gnd}=\frac{V_{gnd2}-V_{gnd1}}{V_{std2}-V_{std1}}*R_{std} \tag{1}$$

Table 1: Measurement of discrete resistance using ground resistance monitor

Monitor reading Ohms	Multimeter reading Ohms	Difference as (% of full scale)
1.9	1.8	0.4
3.8	3.6	0.8
5.9	5.4	2.0
7.1	7.2	0.4
9.5	9.0	2.0
11.1	10.8	1.2
13.2	12.6	2.4
14.1	14.4	1.2
17.0	16.2	3.1
18.2	18.0	0.8

where R_{gnd} is the ground resistance, V_{gnd2} is the voltage across the ground resistance with the second setting of the analog switch, V_{gnd1} is the voltage across the ground resistance with the first setting of the analog switch, R_{std} is the value of standard resistance (and is equal to 15 Ohms), V_{std2} is the voltage across the standard resistance for switch setting 2 and V_{std1} is the voltage across the standard resistance for switch setting 1.

Due to the errors introduced by the different components in the system, the resulting ground resistance calculating using the above formula has an amount of error associated with it. In this section, the focus will be to quantify the errors introduced by the different components in the system and the overall error in the ground resistance calculated.

Error in the System

The total error in the ground resistance measured by the instrument is given by,[7]

$$\sqrt{(A)^2+(B)^2+(C)^2+(D)^2+(E)^2} \qquad (2)$$

where A, B, C, D, E are defined in equations (3) through (7). Partial differentiation of the equation for Rgnd with respect to the different variables on the right hand side of

$$A = \frac{\frac{\delta R_{gnd}}{\delta V_{gnd2}} * \nabla V_{gnd2}}{R_{gnd}} = -\frac{\nabla V_{gnd2}}{(V_{gnd2} - V_{gnd1})} \qquad (3)$$

$$B = \frac{\frac{\delta R_{gnd}}{\delta V_{gnd1}} * \nabla V_{gnd1}}{R_{gnd}} = -\frac{\nabla V_{gnd1}}{(V_{gnd2} - V_{gnd1})} \qquad (4)$$

$$C = \frac{\frac{\delta R_{gnd}}{\delta V_{std2}} * \nabla V_{std2}}{R_{gnd}} = \frac{(V_{std2} - V_{std1}) * \nabla V_{std2}}{V_{std2}^2} \qquad (5)$$

$$D = \frac{\frac{\delta R_{gnd}}{\delta V_{std1}} * \nabla V_{std1}}{R_{gnd}} = \frac{(V_{std2} - V_{std1}) * \nabla V_{std1}}{V_{std1}^2} \qquad (6)$$

$$E = \frac{\frac{\delta R_{std}}{\delta R_{gnd}}}{R_{gnd}} = \frac{\nabla R_{std}}{R_{std}} \qquad (7)$$

$$\frac{\nabla R_{gnd}}{R_{gnd}} = \sqrt{\frac{(\nabla V_{gnd2}^2 + \nabla V_{gnd1}^2)}{(V_{gnd2} - V_{gnd1})^2} + (V_{std2} - V_{std1})^2 * \left(\frac{\nabla V_{std2}^2}{V_{std2}^4} + \frac{\nabla V_{std1}^2}{V_{std1}^4}\right)} \qquad (8)$$

equation(8) gives the percentage error in the calculation of the ground resistance. To evaluate the error in numerical terms, the individual errors in the different components in the right hand side of equation(8) are calculated and substituted.

Error introduced by the deviation in the value of standard resistance: The standard resistance used was tested using a tektronix digital multimeter DM5110 for accuracy. So the

Error introduced by the deviation in the value of standard resistance: The standard resistance used was tested using a tektronix digital multimeter DM5110 for accuracy. So the deviation in the value of the standard resistance is limited to the error in the reading of the multimeter and is given by:

$$R = + (.25\% \text{ of reading } + .04\% \text{ of full scale}).[8].$$
$$= + (.0375 + .08) = + 0.1175 \text{ Ohms}$$

Errors can also be introduced by the variation of Rstd with temperature. For a variation of temperature from -55 C to +155 C, the variation in Rstd is, [9]

$$R = (.25\% \text{ of value } + 0.25)$$
$$= (.0375 + 0.25) = 0.2875 \text{ Ohms}$$

The total deviation in the standard resistor is given by,

$$R_{total} = 0.2875 + 0.1175$$

$$\nabla R_{std} = 0.4050\,Ohms$$

Error introduced by the instrumentation amplifier: Though there are many sources for error in the instrumentation amplifier, we will concentrate our efforts on the two major sources here.

Offset voltage drift due to temperature is a major source of error in the instrumentation amplifier and is given by,[10]

Input offset voltage temperature drift = $(dV_{OS}/dT) * dT$

(dV_{OS}/dT) for the opamp 741 is 15uV/C. Assuming a temperature change of 40 C,

$$(dV_{OS}/dT) * dT = 15\text{uV/C} * 40 \text{ C}$$
$$= 0.6 \text{ mV}$$

The other major source of error in an instrumentation amplifier is due to the common mode voltage and is given by,[11]

$$(2)^{1/2}\ V_{cm}/CMRR_{in\text{-}circuit}$$

The common mode signal is due to the leakage currents at lower frequencies. These common mode signals could be as high as 10 Volts. So,

$$(2)^{1/2}\ V_{cm}/CMRR_{in\text{-}circuit} = (1.414 * 10)/233$$
$$= 60\ \text{mV}$$

Total error from the instrumentation amplifier is 60.6 mV. But this signal has to pass through the filter whose gain at frequencies below 1900 Hz is 1/666. Multiplying the above value by this factor,

$$60.6\ \text{mV}/666 = 90\ \text{uV}$$

Substituting this value in the equation for ground resistance would lead to the cancellation of the error as the same amount of error would be present in the readings for both the switch settings.

Error introduced by the filter: The filter used has a non-linearity of 3% with changes in input voltage.

Error caused by the ground resistance amplifier: This amplifier is an inverting opamp amplifier using the 741 opamp and has an input offset voltage drift caused by temperature variation and is given by,

$$(dV_{os}/dT) * dT = 15\text{uV/C} * 40\ \text{C}$$

= 0.6 mV

Error introduced by the rms-to-dc convertor: The rms-to-dc convertor can have a maximum error of 0.5% of the reading.[12].

Error introduced by the analog to digital convertor (ADC): The ADC has an error of 1/2 LSB. As the resolution of the ADC is 0.0098 V, the error will be 0.0049 V.

Error introduced by the display: The display has a resolution of 0.1 Ohms. This would result in an error which will be as high as 0.05 Ohms. This has to be added or subtracted to the final value of ground resistance.

Error introduced by the analog switches: The first set of analog switches do not play a role in the data acquisition process and hence do not contribute to the errors in the measurement of ground resistance. The second set of analog switches used for the instrumentation amplifier is involved in the data acquisition process and its effect on the measurement process is negligible. This is because the change in resistance of the switches is about 0.3 Ohms in the range used and the switches are connected to the input resistance of the amplifier. The input resistance of the instrumentation amplifier is of the order of kilo Ohms and an addition of 0.3 Ohms to such a high resistance will not introduce any noticeable error.

Substituting the appropriate values in equation(8), and calculating for the total percentage error,

$$\frac{\nabla R_{gnd}}{R_{gnd}} = 8.3\%$$

CHAPTER 6

CONCLUSION AND FUTURE WORK

An instrument for the monitoring of ground resistance has been developed. The instrument measures ground resistance every 24 hours and stores it in memory. Along with the ground resistance, the voltage across the ground resistance and the current flowing through it are also stored. The instrument is also capable of calculating the mean and standard deviation of the most recent 64 readings. The date and time of measurement are stored with each reading. The output on the digital display can read down to 1/10ths of a unit. The instrument can record readings for well over a period of three years after which readings recorded will be over written. The instrument is also provided with batteries and associated circuitry for continuous operation in the event of a power failure.

The instrument has a range of 0-25.5 Ohms. The resolution is at 0.1 Ohms. The maximum error is under 5% of full scale.

Future work in this area may include:

(1) Enhancement of resolution using a 16-bit microcontroller.

(2) Better display techniques for error indications.

(3) Increased data storage.

(4) Collection and analysis of data regarding the ground resistance at substations over a long period of time.

REFERENCES

REFERENCES

1. S. MELIOPOULOS, Power System Grounding and Transients: An Introduction, Marcel Dekker Inc, New York and Basel, 1988.

2. G. F. TAGG, Earth Resistances, George Newnes Limited, London, 1964.

3. J. G. GRAEME, G. E. TOBEY, and L. P. HUELSMAN, Operational Amplifiers- Design and Applications, McGraw Hill Book Company, New York, 1971.

4. Motorola High-speed CMOS Logic Data, Motorola Inc., 1988.

5. W. CHEN, Passive and Active Filters- Theory and Implementations, John Wiley and Sons, New York, 1986.

6. 8-Bit Embedded Controllers, Intel Corporation, Illinois, 1989.

7. G. L. SQUIRES, Practical Physics- 3rd Ed, Cambridge University Press, Cambridge, U.K, 1985.

8. Tektronix DM5110 Programmable DMM Instruction Manual, Tektronix Inc., Oregon, 1989.

9. Digikey Corporation Catalog 101, Digikey Corporation, Minnesota, 1990.

10. Linear IC Databook, National Semiconductor, California, 1982.

11. P. H. GARRETT, Analog I/O Design Acquisition: Conversion: Recovery, Reston Publishing Company, Inc., Virginia, 1981.

12. Integrated Circuits Data Book, Burr Brown, Arizona, 1989.

13. Motorola FACT Data, Motorola Inc., 1990.

14. IEEE Guide for Safety in AC Substation Grounding, The Institute of Electrical and Electronic Engineers Inc., New York, 1986.

15. <u>Ground Resistance Testers</u>, Biddle Instruments, Pennsylvania, 1990.

APPENDICES

APPENDIX A

SOFTWARE

Figure 10: CONTROLLER SOFTWARE

```
;Locations 0-7: registers
;Location 8: used to store A at one place
;Location 9: unused
;Location 10:unused ;locn. 35 is alloted for value of R
;Locations 11-12 are allocated for current measured(Q,R)
;Location 13 is allocated for V1
;Location 14 is for current minimum
;Location 15 is allocated for ground resistance recorded
;Location 16: unused
;Location 17 is for ground resistance preset value
;Location 18 for sum of Resistance lo-byte
;Location 19 for sum of Resistance hi-byte
;Locations 20, 21, 22 used as scratch pads in keypad
     subroutine
;Location 23 contains the 3-digit number from the keypad
;Location 24 for mean of past 64 resistance readings
;Location 25 for SD squared
;Locations 26-29: subr DISPLAYSUB uses for intermediate
     results
;Location 30: RETVAL
;Locations 31-41: unused
;Locations 42-105 allocated for readings of Ground
     Resistance
;Location 106: saved stack pointer after interrupt
;Location 107: saved stack pointer for inc & dec
;Locations 108-127: STACK
;Analog Switches: 00 indicates min. resistance and hence
     min. current
;11 indicates max. resistance and hence max.
     current
;Analog switch for V1, V2: 0 connects Va-Vb= Ground
     resistance
;1 connects Vc-Vd= Standard resistance


RETVAL     EQU 30D
SAVESP     EQU 106D
SAVESP2    EQU 107D
INITSTACK  EQU SAVESP+2

           ORG 0000H
           LJMP INIT              ;Program beginning

           ORG 0003H
           LJMP BEGIN             ;Interrupt from RTC

           ORG 000BH
           RETI                   ;Interrupt from timer

           ORG 0013H
```

```
            LJMP KEYPADSUBR        ;Keypad interrupt

            ORG 0040H
INIT:       CLR P1.7               ;Turn off alarm
            MOV SP,#INITSTACK      ;Set stack to 107D
            MOV IP,#00000000B       ;Interrupts in their natural
                                   ;sequence
            MOV IE,#10000101B      ;Enable interrupts, Kypd & RTC
                                   ;intr.
            MOV R0,#0FFH           ;Decrement  and  display  #s
                                   ;starting
INITLOOP:   LCALL  DISPLAYSUB         ;from  255  to  0  to  check
                                   ;functioning
            LCALL WAIT             ;of display
            DEC R0
            CJNE R0,#0D,INITLOOP
            SETB P1.1              ;Disable display
                                   ;Test routine for RTC
            CLR P1.3               ;Enable RTC
            MOV P0,#00001111B      ;Write 15 to control register of
                                   ;RTC
            CLR P1.5               ;RTC write mode
            LCALL WAIT             ;Wait for things to settle
            SETB P1.5              ;RTC write disable
            MOV P0,#0F0H           ;Clear interrupt register of RTC
            CLR P1.5               ;RTC write mode
            LCALL WAIT             ;Give time to settle
            SETB P1.5              ;RTC write disable
            MOV P0,#05H            ;RTC in out-of-test halt mode
            CLR P1.5
            LCALL WAIT
            SETB P1.5
            MOV R0,#00010000B      ;Move  1  to  all  registers  to
                                   ;ensure
CLKCHK:     MOV A,R0               ;a valid BCD in each of them
            SETB ACC.0
            MOV P0,A
            CLR P1.5
            LCALL WAIT
            SETB P1.5
            MOV A,R0
            ADD A,#00010000B
            MOV R0,A
            CJNE R0,#0F0H,CLKCHK
            MOV P0,#0F1H           ;Select 24 hour mode and leap
                                   ;year
            CLR P1.5
            LCALL WAIT
            SETB P1.5
            MOV P0,#01H            ;Clock to run
            CLR P1.5
            LCALL WAIT
```

```
              SETB P1.5
CLKAGN:       MOV P0,#00101111B     ;Address units seconds
              CLR P1.4              ;RTC read mode
              LCALL WAIT
              MOV A,P0
              SETB P1.4             ;RTC read disable
              CLR ACC.5
              JZ CLKOVR
              MOV R0,A
              LCALL DISPLAYSUB
              CLR P1.3
              LJMP CLKAGN
CLKOVR:       SETB P1.3             ;Disable RTC
              CLR P2.4              ;Keep it ready for interrupt
              MOV P0,#00001111B
              CLR P3.1
LOOP:         MOV R4,#00D           ;Initialize R4 for var gain (gnd
                                    ;res)
              CLR P3.4              ;Initialize for gain of 1
              AJMP LOOP             ;Stay here until an interrupt
                                    ;occurs
BEGIN:        MOV SAVESP,SP
              SETB P3.1             ;Disable keypad buffer
              MOV PSW,#00
              MOV IP,#00000000B
              MOV IE,#00000101B     ;Disables all interrupts
              CLR P2.6              ;Set analog switches for min
                                    ;current
              CLR P2.7
              LCALL WAIT
              MOV R7,#00H           ;Clear R6, R7
              MOV R1,#20D           ;Initialise R1 for gnd res
                                    ;voltage
              MOV R2,#02D           ;Initialize R2 to get 2 sets of
                                    ;readings
              MOV R3,#22D           ;Initialise R0 for Std res
                                    ;voltage
CONTINUE:     CLR P3.5              ;Reenable ADC
              CLR P1.6              ;Connect inst. amp. to std. res.
              LCALL WAITASEC
              LCALL WAITASEC
              LCALL WAITASEC
              SETB PSW.3            ;Change to register bank 1
              CLR A
              MOV R0,A              ;Clear R0,R1,R2
              MOV R1,A
              MOV R2,A
REDERROR:     MOV P0,#0FFH          ;Keep port ready to receive
                                    ;input
              LCALL WAIT
              MOV A,P0              ;Move ADC output i.e 10*V1 to A
              ADD A,R2              ;Add input from ADC to LSB of
```

```
                                        ;sum
           MOV R2,A
           JNC NOREGINC
           CLR C
           INC R1                       ;Increment R1 (MSB) if there is
                                        ;a carry
NOREGINC:  INC R0                       ;Increment counter
           CJNE R0,#00D,REDERROR             ;Jump if 256 readings are
                                             ;not taken
           MOV A,R1                     ;Move average of 256 readings to
                                        ;ACC
           CLR PSW.3                    ;Change reg bank back to 0
           PUSH 0
           MOV 0D,3D
           MOV @R0,A
           POP 0
           SETB P3.5
           PUSH 0
           MOV R0,A                     ;Display the value of voltage
                                        ;from ADC
           MOV 8D,A                     ;Store A
           LCALL DISPLAYSUB
           LCALL WAITASEC
           POP 0
           MOV B,35D                    ;Move the value of std res to B
           MOV A,8D                     ;Restore A
           DIV AB                       ;10*V1/R
           MOV 31D,A
           MOV A,B
           MOV B,#10D
           MUL AB
           MOV B,35D
           DIV AB
           MOV 32D,A
           MOV A,B
           MOV B,#10D
           MUL AB
           MOV B,35D
           DIV AB
           MOV 33D,A
           MOV A,B
           MOV B,#10D
           MUL AB
           MOV B,35D
           DIV AB
           MOV 34D,A
           MOV A,31D
           MOV B,#5D
           DIV AB
           MOV 8D,B
           MOV B,#100D
           MUL AB
```

```
          MOV 31D,A
          MOV A,8D
          MOV B,#10D
          MUL AB
          ADD A,32D
          MOV B,#5D
          DIV AB
          MOV 8D,B
          MOV B,#10D
          MUL AB
          ADD A,31D
          MOV 31D,A
          MOV A,8D
          MOV B,#10D
          MUL AB
          ADD A,33D
          MOV B,#5D
          DIV AB
          ADD A,31D
          MOV 31D,A
          MOV A,B
          MOV B,#10D
          MUL AB
          ADD A,34D
          MOV B,#5D
          DIV AB
          CLR C
          SUBB A,#5D
          JC NOADD
          MOV A,31D
          INC A
          MOV 31D,A
NOADD:    CLR C
          MOV 11D,31D          ;1000I is in A and locn. 11D
          PUSH 0
          MOV R0,31D           ;Display the current calculated
          LCALL DISPLAYSUB
          LCALL WAITASEC
          POP 0
                               ;Adjusting I above bare min.
          MOV A,11D
          CLR C
          SUBB A,14D           ;Sub min. I from current I
          JNC ADJUST           ;Current is enough
          CLR C
          LCALL ADJRES         ;If current is not enough, do it
                               ;again
XCUSE:    LJMP CONTINUE

                               ;Switch adj., Cal. of V1, V2 and
                               ;storing
ADJUST:   SETB P1.6            ;Connect instr amp to gnd res
```

```
            MOV A,R4
            JZ NOAMP
            SETB P3.4
NOAMP:      LCALL WAITASEC          ;Wait for a second for rms2dc to
                                    ;settle
            LCALL WAITASEC
            LCALL WAITASEC
            CLR P3.5                ;Enable ADC
            LCALL WAIT              ;Wait for ADC to settle
            SETB PSW.3              ;Change to reg bank 1
            CLR A
            MOV R0,A                ;Clear R0,R1,R2
            MOV R1,A
            MOV R2,A
ERRORED:    MOV P0,#0FFH            ;Keep port ready for input
            LCALL WAIT
            MOV A,P0                ;Get corresponding voltage value
            ADD A,R2                ;Add input from ADC to LSByte
            MOV R2,A
            JNC REGINCNOT
            CLR C
            INC R1                  ;Inc MSByte if there is a carry
REGINCNOT:      INC R0              ;Increment counter
            CJNE R0,#00D,ERRORED        ;Go thro loop if there are
                                        ;<256 rdgs
            MOV A,R1                ;Move mean of 256 readings to
                                    ;ACC
            CLR PSW.3               ;Switch back to reg bank 0
            MOV @R1,A               ;10*V1 (for gnd res) goes to
                                    ;loc. 20D/21D
            PUSH 0
            MOV R0,A
            LCALL DISPLAYSUB
            LCALL WAITASEC
            MOV A,R0
            POP 0
            INC R1
            INC R3
            MOV A,P2                ;Turn on P2.7, to use another
                                    ;sw setting
            ADD A,#64D
            LCALL WAIT
            MOV P2,A
            CLR P3.4
            INC R6
            DJNZ R2,XCUSE           ;Do it twice (w different analog
                                    ;switch setting)
            MOV A,21D
            CLR C
            SUBB A,20D
            MOV 13D,A               ;13D contains diff. between
                                    ;voltages
```

```
          MOV 39D,11D            ;Copy current to 39D
          MOV 40D,13D            ;Copy voltage to 40D
          MOV 36D,22D
          MOV 37D,23D

          ;***SPECIAL DIVISION PROCESS START****

;Locns used: 31-34,36-41, 36 & 37 have Vstdl & 2, 13D has Vgnd

CARROT:   MOV A,37D              ;Get second voltage of standard
                                 ;res
          SUBB A,36D             ;Subtract first value of
                                 ;standard res
          MOV 36D,A              ;Store the difference at 36D
          MOV 37D,#00D           ;Clear location 37D
          MOV A,35D
          MOV B,#10D
          MUL AB
          MOV B,A                ;10*Rstd in B
          MOV A,13D              ;Get difference in Vg
                                 ;Get standard resistance value
          MUL AB                 ;Multiply diffVg*Rstd
          MOV 38D,A              ;Store A in location 38D
          MOV 37D,B              ;Store MSByte in location 37D
          MOV A,B                ;Move B to A
          JZ  OKAY1
          MOV 39D,#00D           ;Clear location 39D (used for
                                 ;Alpha)
          MOV 40D,#08D
CARROT2:  JB ACC.7,OVER
          RLC A
          CLR C
          INC 39D                ;Rotate left once and increment
                                 ;Alpha
          DJNZ 40D,CARROT2       ;Check all 8 bits
OVER:     MOV 37D,A              ;Store the # after SLs back in
                                 ;locn. 37D
          MOV 40D,39D            ;Copy alpha to 40D
          MOV 32D,38D            ;Copy LSByte into 32D
          MOV 31D,#00D
DUMB:     MOV A,31D
          RL A
          MOV 31D,A
          MOV A,32D
          RLC A
          MOV 32D,A
          JNC NOADD1
          CLR C
          INC 31D
NOADD1:   DJNZ 40D,DUMB
          MOV 38D,32D
          MOV A,31D
```

```
            ADD A,37D               ;37D has MSByte after LSs, 38D
                                    ;has LSByte
                                    ;and 39D has # of LSs
            MOV 37D,A
            LJMP MERGE
OKAY1:      MOV 39D,#08D
            MOV 37D,#00D            ;Clear 37D
            MOV A,38D               ;LSByte into A
            MOV 40D,#08D
AGAIN11:    JB ACC.7,OVER11
            RLC A
            INC 39D
            DJNZ 40D,AGAIN11
OVER11:     MOV 37D,A               ;Alpha is in 39D, # after LSs is
                                    ;in 37D
            MOV 38D,#00D            ;and 38D and difference in Rstd
                                    ;is in 36D.
                                    ;All other locations used in
                                    ;this program are free
MERGE:      LCALL LONGDIV
            MOV A,39D
            JZ CANTSR
            MOV A,32D
            MOV 40D,39D             ;Copy Alpha to 40D
STRTSHFT:   CLR C
            RRC A
            DJNZ 40D,STRTSHFT
            JNC NORNDOFF
            CLR C
            INC A
NORNDOFF:   MOV 32D,A
            MOV A,31D               ;MSByte is in A
            JZ CANTSR               ;If MSByte is zero, no need to
                                    ;SR
            MOV 33D,#00D
            MOV 40D,39D
SHFTRT:     MOV A,33D
            RR A
            MOV 33D,A
            MOV A,31D
            RRC A
            MOV 31D,A
            JNC NOINCREM2
            CLR C
            MOV A,33D
            SETB ACC.7
            MOV 33D,A
NOINCREM2:      DJNZ 40D,SHFTRT
            MOV A,33D
            ADDC A,32D
            MOV 32D,A
            JNC CANTSR
```

```
          CLR C
          MOV A,31D
          INC A
          MOV 31D,A
CANTSR:   LJMP ITISOVER          ;32D has LSByte, 31D has MSByte
LONGDIV:  CLR C
          MOV 31D,#00D           ;MSByte of Quotient in 31D
          MOV 32D,#00D           ;LSByte of quotient in 32D
JUSTCHK:  MOV A,37D
          JNZ STILL
          MOV A,38D
          SUBB A,36D
          JC OVER12
STILL:    MOV A,38D
          SUBB A,36D
          MOV 38D,A
          JNC NODECREM
          CLR C
          DEC 37D
NODECREM: MOV A,32D
          ADDC A,#01D
          MOV 32D,A
          JNC NOINCREM
          CLR C
          INC 31D
NOINCREM: LJMP JUSTCHK           ;Longdiv leaves the quotient in
                                 ;31D (MSByte)
OVER12:   CLR C
          RET                    ;and 32D (LSByte)
ITISOVER: MOV 15D,32D            ;If R4 is not zero, then divide
                                 ;# by 10D
          MOV A,R4
          JZ HOPTWO
          MOV 37D,31D            ;Numerator in locns. 37 & 38
          MOV 38D,32D
          MOV 36D,#10D           ;Denominator in locn. 36
          LCALL LONGDIV
          MOV 15D,32D
          ;***END SPECIAL DIVISION SECTION***

HOPTWO:   MOV A,R4               ;If  R4  is  not  0,  then
                                 ;use amplification
          JNZ NOADTO             ;has already been used
          MOV A,15D
          CLR C
          SUBB A,#50D            ;If amp is not used and if res
          JNC NOADTO             ;is below 5 Ohms, turn amp and
                                 ;set R4 and begin
          CLR C                  ;it all over again
          MOV R4,#10D
          MOV R5,#00D
          LJMP BEGIN
```

```
NOADTO:     MOV A,R4
            JZ NOADTR
            MOV A,13D              ;Set amplification marker on
before
            SETB ACC.7             ;storing
            MOV 13D,A
            MOV R4,#00D
NOADTR:     PUSH 0                 ;Display Rg
            MOV R0,15D
            LCALL DISPLAYSUB
            LCALL WAITASEC
            POP 0
                                   ;Comp. present value with preset
                                   ;and sounding alarm/ visual
                                   ;indication
            CLR P1.7               ;Turn LED off
            MOV A,17D              ;Move preset value to
                                   ;accumulator
            CLR C
            SUBB A,15D             ;Subtract present from preset
            JNC CONT10
            CLR C
            LCALL ALARM            ;jump in alarm if present >
                                   ;preset
                                   ;Recording of time
CONT10:     SETB P1.5              ;Disable RTC write
            CLR P1.3               ;Select RTC
            CLR P1.4               ;Read mode
            MOV P0,#10101111B      ;Units months (higher nibble is
                                   ;address,
            LCALL WAIT             ;lower nibble is input value
            MOV A,P0
            ANL A,#00001111B
            MOV R0,A               ;Lower nibble of R0 is units of
                                   ;month
            MOV P0,#10111111B      ;Read tens months
            LCALL WAIT
            MOV A,P0
            RRC A                  ;Check for 1 in tens
            JNC NEXT               ;No carry, month is between 1..9
            CLR C
            LCALL ADTEN            ;Otherwise it is between 10..12
NEXT:       MOV A,R0
            SWAP A                 ;Higher nibble is months
            ANL A,#11110000B
            MOV R3,A               ;Save month in R3
            CLR P1.3
                                   ;Date
            MOV P0,#10001111B      ;Units date
            LCALL WAIT
            MOV A,P0
            ANL A,#00001111B
```

```
          MOV R0,A
          MOV P0,#10011111B    ;Tens date
          LCALL WAIT
          MOV A,P0
          ANL A,#00001111B
          RRC A
          JNC NEXT1
          CLR C
          LCALL ADTEN
NEXT1:    RRC A
          JNC NEXT2
          CLR C
          LCALL ADTONTY
NEXT2:    MOV A,R0             ;Load date
          ANL A,#00011111B     ;Store last bit of date in R4
          RRC A
          JNC NEXT3
          CLR C
          LCALL BITONESET
NEXT3:    ORL A,R3             ;Lower nibble of R3 is upper 4
                               ;bits of date
          MOV R3,A
                               ;Time
          CLR P1.3
          MOV P0,#01101111B    ;Units hours
          LCALL WAIT
          MOV A,P0
          ANL A,#00001111B
          MOV R0,A
          MOV P0,#01111111B    ;Tens of hours
          LCALL WAIT
          MOV A,P0
          RRC A
          JNC NEXT4
          CLR C
          LCALL ADTEN
NEXT4:    RRC A
          JNC NEXT5
          CLR C
          LCALL ADTONTY
NEXT5:    MOV A,R0             ;Load hours
          ANL A,#00011111B     ;Mask upper 3 bits
          ORL A,R4             ;Lower 5 bits of R4 contain time
          MOV R4,A
          JMP ORIGNL
ADTEN:    PUSH 1               ;Save r1
          MOV 1,A              ;Save acc in R1
          MOV A,R0
          ADD A,#10D
          MOV R0,A
          MOV A,1              ;Restore A
          POP 1                ;Restore R1
```

```
          RET
ADTONTY:  PUSH 1
          MOV 1,A
          MOV A,R0
          ADD A,#20D
          MOV R0,A
          MOV A,1
          POP 1
          RET
BITONESET:     PUSH 1
          MOV 1,A
          CLR A
          ADD A,#128D
          MOV R4,A
          MOV A,1
          POP 1
          RET
ORIGNL:   SETB P1.3            ;RTC disable
          SETB P1.4            ;RTC read disable
          SETB P3.1
          SETB P3.5
          CLR P1.2             ;Memory enable
                               ;Window shifting
          MOV DPTR,#1022D
RESISTWS: CLR P1.2
          MOVX A,@DPTR         ;Read form external memory
          INC DPTR             ;Increment mem. pointer
          MOVX @DPTR,A
          SETB P1.2
          MOV A,DPL            ;DEC DPTR
          CLR C
          SUBB A,#01D
          MOV DPL,A
          JNC CONT1
          CLR C
          DEC DPH              ;If carry decrement DPH
CONT1:    MOV A,DPL            ;DJNZ DPTR,RESISTWS
          CLR C
          SUBB A,#01D
          MOV DPL,A
          JNC CONT2
          CLR C
          DEC DPH
CONT2:    MOV A,DPL
          ORL A,DPH
          JNZ RESISTWS         ;Jump back to shift
          MOV DPTR,#0000D
          CLR P1.2
          MOVX A,@DPTR
          INC DPTR
          MOVX @DPTR,A
          MOV DPTR,#0000D      ;Store loc. 39D at ext. mem 0
```

```
                                    ;(current difference)
          MOV A,15D
          MOVX @DPTR,A
          SETB P1.2
          MOV DPTR, #2046D          ;Do the same window shifting for
                                    ;voltage (starts at 1024, ends
                                    ;at 2047
VOLWS:    CLR P1.2
          MOVX A,@DPTR
          INC DPTR
          MOVX @DPTR,A
          SETB P1.2
          MOV A,DPL                 ;DEC DPTR
          CLR C
          SUBB A,#01D
          MOV DPL,A
          JNC CONT3
          CLR C
          DEC DPH
CONT3:    MOV A,DPL                 ;DEC DPTR
          CLR C
          SUBB A,#01D
          MOV DPL,A
          JNC CONT4
          CLR C
          DEC DPH
CONT4:    MOV A,#255D
          CJNE A,DPL,VOLWS
          MOV A,#03D
          CJNE A,DPH,VOLWS
          MOV DPTR,#1024D
          MOV A,13D                 ;Store 13 at ext mem 1024
                                    ;(difference between voltages)
          CLR P1.2
          MOVX @DPTR,A
          SETB P1.2
          MOV DPTR,#3070D           ;Do same for TIME window
TIMWS:    CLR P1.2
          MOVX A,@DPTR
          INC DPTR
          MOVX @DPTR,A
          SETB P1.2
          MOV A,DPL                 ;DEC DPTR
          CLR C
          SUBB A,#01D
          MOV DPL,A
          JNC CONT5
          CLR C
          DEC DPH
CONT5:    MOV A,DPL                 ;DEC DPTR
          CLR C
          SUBB A,#01D
```

```
            MOV DPL,A
            JNC CONT6
            CLR C
            DEC DPH
CONT6:      MOV A,#255D
            CJNE A,DPL,TIMWS
            MOV A,#07D
            CJNE A,DPH,TIMWS
            MOV DPTR,#2048D     ;Move R4 to 2048 (hours and last
                                ;bit of date)
            MOV A,R4
            CLR P1.2
            MOVX @DPTR,A
            SETB P1.2
            MOV DPTR,#4094D     ;Do same for date
DATWS:      CLR P1.2
            MOVX A,@DPTR
            INC DPTR
            MOVX @DPTR,A
            SETB P1.2
            MOV A,DPL           ;DEC DPTR
            CLR C
            SUBB A,#01D
            MOV DPL,A
            JNC CONT7
            CLR C
            DEC DPH
CONT7:      MOV A,DPL           ;DEC DPTR
            CLR C
            SUBB A,#01D
            MOV DPL,A
            JNC CONT8
            CLR C
            DEC DPH
CONT8:      MOV A,#255D
            CJNE A,DPL,DATWS
            MOV A,#11D
            CJNE A,DPH,DATWS
            MOV DPTR,#3072D     ;Save R3 (date/month) at 3072
            MOV A,R3
            CLR P1.2
            MOVX @DPTR,A
            SETB P1.2
            MOV R0,#104D        ;Do similar for resistance in
                                ;internal memory
RESWS:      MOV A,@R0
            INC R0
            MOV @R0,A
            DEC R0
            DEC R0
            CJNE R0,#41D,RESWS
            MOV 42D,15D         ;Save  15D  at  42D(resistance
```

```
                                    ;after round_off)
                                    ;Calculation of mean
          MOV A,105D
          JZ ENDMEAN
          MOV R0,#106D              ;Location pointer
          MOV 18D,#00D
          MOV 19D,#00D
MEANCAL:  DEC R0
          MOV R1,#18D               ;Location for total -low byte
          MOV A,@R0
          ADD A,@R1
          MOV @R1,A
          INC R1
          RLC A
          ANL A,#00000001B
          ADD A,@R1
          CLR C
          MOV @R1,A
          CJNE R0,#42D,MEANCAL           ;The sum of 64 rdgs is in
                                         ;18, 19D
          MOV A,@R1
          RL A
          RL A
          ANL A,#11111100B
          MOV @R1,A
          DEC R1
          MOV A,@R1
          SWAP A
          RRC A
          CLR C
          RRC A
          CLR C
          ANL A,#00000011B
          INC R1
          ORL A,@R1
          MOV 24D,A
ENDMEAN:  NOP
                                    ;Calculation of SD squared
          MOV R0,#42D
          MOV A,@R0
          JZ ENDSD
          MOV R0,#105D
SDSTART:  MOV R1,#24D
          MOV A,@R0
          CLR C
          SUBB A,@R1
          MOV B,A
          MUL AB
          INC R1
          ADD A,@R1
          MOV @R1,A
          DEC R0
```

```
            CJNE R0,#42D,SDSTART
            MOV A,@R1
            RRC A
            CLR C
            RRC A
            CLR C
            RRC A
            CLR C
            RRC A
            CLR C
            RRC A
            CLR C
            RRC A
            CLR C
            MOV @R1,A
ENDSD:      NOP
            MOV P0,#00001111B    ;Make kb interrupts possible
            MOV IE,#10000101B    ;Enable interrupts
            CLR P2.4
            MOV P0,#00001111B
            CLR P3.1             ;Enable keypad buffer
            MOV SP,SAVESP
            RETI                 ;End of main routine
                                 ;Subroutine for alarm sounding
ALARM:      SETB P1.7            ;Enable alarm
            RET
                                 ;Subroutine for keyboard
KEYPADSUBR:      MOV IE,#10000001B    ;Disable keypad interrupt
            MOV SAVESP,SP        ;Save stack pointer (for safe
                                 ;return)
            CLR P3.1             ;Enable keyboard buffer
            MOV R0,#201D
            LCALL DISPLAYSUB
            LCALL WAITASEC
            CLR P3.1
            LCALL STARTDEB
            MOV R0,#202D
            LCALL DISPLAYSUB
            LCALL WAITASEC
            CLR P3.1
            MOV A,RETVAL
            RRC A
            JNC GOAWAY
            CLR C
            MOV R0,#23D          ;Initialise R0
            MOV P0,#10111111B    ;Send a 0 thro' pin 6 of port 0
            SETB P2.4
            MOV A,P0             ;Move the contents of port 0 to
                                 ;ACC
            RRC A                ;Rotate ACC right thro' carry
            JNC CIRC1            ;If the right extreme button is
                                 ;pressed
```

```
          CLR C
          RRC A                    ;jump to ALTPRESET or if second
                                   ;button
          JNC CIRC2                ;from right is pressed, jump to
                                   ;DISPVALS
          CLR C
GOAWAY:   MOV IE,#10000101B
          MOV SP,SAVESP            ;Restore sp
          CLR P3.1                 ;Enable keypad buffer
          CLR P2.4
          MOV P0,#00001111B
          RETI                     ;Else return from interrupt
CIRC1:    LJMP ALTPRESET
CIRC2:    LJMP DISPVALS
ALTPRESET:     SETB P3.3           ;Keep pin 3.3 ready for input
          MOV C,P3.3               ;Wait until button is released
          JNC ALTPRESET
          CLR C
          PUSH 0
          MOV R0,#203D
          LCALL DISPLAYSUB
          LCALL WAITASEC
          CLR P3.1
          SETB P2.4
          MOV P0,#10101111B        ;Keep 2 rows sensitive for input
          SETB P3.3                ;Keep pin 3.3 ready for input
                                   ;Wait for new key
          JB P3.3,$
          MOV P0,#10111111B
          LCALL STARTDEB
          MOV R0,#204D
          LCALL DISPLAYSUB
          LCALL WAITASEC
          CLR P3.1
          POP 0
          MOV A,P0
          RRC A
          JNC ALTCUR
          CLR C
          RRC A
          JNC ALTSTDRES
          CLR C
          MOV P0,#11101111B
          MOV A,P0
          SWAP A
          RLC A
          JNC ALTGNDRES
          CLR C
          RLC A
          RLC A
          JNC TCIRC
          CLR C
```

```
            MOV IE,#10000101B
            MOV SP,SAVESP
            CLR P3.1
            CLR P2.4
            MOV P0,#00001111B
            RETI
TCIRC:      LJMP ALTTIME
ALTCUR:     PUSH 0
            MOV R0,#205D
            LCALL DISPLAYSUB
            LCALL WAITASEC
            CLR P3.1
            POP 0
            LCALL GETNUM
            MOV 14D,23D
            MOV R0,23D
            LCALL DISPLAYSUB      ;Display the min. current input
            LCALL WAITASEC
            CLR P3.1
            MOV IE,#10000101B
            MOV SP,SAVESP
            CLR P3.1
            CLR P2.4
            MOV P0,#00001111B
            RETI
ALTSTDRES:        PUSH 0
            MOV R0,#206D
            LCALL DISPLAYSUB
            LCALL WAITASEC
            CLR P3.1
            POP 0
            LCALL GETNUM
            MOV 35D,23D
            MOV R0,23D
            LCALL DISPLAYSUB      ;Display the std. res. input
            LCALL WAITASEC
            CLR P3.1
            MOV IE,#10000101B
            MOV SP,SAVESP
            CLR P3.1
            CLR P2.4
            MOV P0,#00001111B
            RETI
ALTGNDRES:        PUSH 0
            MOV R0,#207D
            LCALL DISPLAYSUB
            LCALL WAITASEC
            CLR P3.1
            POP 0
            LCALL GETNUM
            MOV 17D,23D
            MOV R0,23D
```

```
          LCALL DISPLAYSUB     ;Display the max. res. input
          LCALL WAITASEC
          CLR P3.1
          MOV IE,#10000101B
          MOV SP,SAVESP
          CLR P3.1
          CLR P2.4
          MOV P0,#00001111B
          RETI
ALTTIME:  MOV R0,#208D
          LCALL DISPLAYSUB
          LCALL WAITASEC
          CLR P3.1             ;Enable keypad buffer
          MOV R0,#20D
          MOV 21D,#00D
          SETB P3.3            ;Keep pin 3.3 ready for input
TILREL:   MOV C,P3.3
          JNC TILREL
          CLR C
          PUSH 0
          MOV R0,#209D
          LCALL DISPLAYSUB
          LCALL WAITASEC
          CLR P3.1
          MOV P0,#00111111B    ;Keep the numeral rows sensitive
          CLR P2.4
          SETB P3.3            ;Keep pin 3.3 ready for input
          JB P3.3,$            ;Wait here until key is pressed
          MOV R0,#210D
          LCALL DISPLAYSUB
          LCALL WAITASEC
          POP 0
          CLR P3.1
          LCALL KEYREAD
          PUSH 0
          MOV R0,20D
          LCALL DISPLAYSUB
          LCALL WAITASEC
          MOV R0,#01D
          LCALL DISPLAYSUB
          LCALL WAITASEC
          POP 0
          SETB P1.4
          CLR P1.5
          MOV A,#05H           ;Set clock in out-of-test halt
                               ;mode
          CLR P1.3
          LCALL WAIT
          SETB P1.3
          MOV A,20D
          ORL A,#01000000B     ;Units min
          MOV P0,A
```

```
        CLR P1.3
        LCALL WAIT
        SETB P1.3
        CLR P3.1
        CLR P2.4
        MOV P0,#00111111B
UNMIN:  SETB P3.3           ;Keep pin 3.3 ready for input
        JB P3.3,$
        LCALL KEYREAD
        PUSH 0
        MOV R0,20D
        LCALL DISPLAYSUB
        LCALL WAITASEC
        MOV R0,#02D
        LCALL DISPLAYSUB
        LCALL WAITASEC
        POP 0
        MOV A,20D
        ORL A,#01010000B    ;Tens min
        MOV P0,A
        CLR P1.3
        LCALL WAIT
        SETB P1.3
        CLR P3.1
        CLR P2.4
        MOV P0,#00111111B
TENMIN: SETB P3.3           ;Keep pin 3.3 ready for input
        JB P3.3,$
        LCALL KEYREAD
        PUSH 0
        MOV R0,20D
        LCALL DISPLAYSUB
        LCALL WAITASEC
        MOV R0,#03D
        LCALL DISPLAYSUB
        LCALL WAITASEC
        POP 0
        MOV A,20D
        ORL A,#01100000B    ;Units hours
        MOV P0,A
        CLR P1.3
        LCALL WAIT
        SETB P1.3
        CLR P3.1
        CLR P2.4
        MOV P0,#00111111B
UNHRS:  SETB P3.3           ;Keep pin 3.3 ready for input
        JB P3.3,$
        LCALL KEYREAD
        PUSH 0
        MOV R0,20D
        LCALL DISPLAYSUB
```

```
        LCALL WAITASEC
        MOV R0,#04D
        LCALL DISPLAYSUB
        LCALL WAITASEC
        POP 0
        MOV A,20D
        ORL A,#01110000B     ;Tens hours
        MOV P0,A
        CLR P1.3
        LCALL WAIT
        SETB P1.3
        CLR P3.1
        MOV P0,#00111111B
        CLR P2.4
TENHRS: SETB P3.3            ;Keep pin 3.3 ready for input
        JB P3.3,$
        LCALL KEYREAD
        PUSH 0
        MOV R0,20D
        LCALL DISPLAYSUB
        LCALL WAITASEC
        MOV R0,#05D
        LCALL DISPLAYSUB
        LCALL WAITASEC
        POP 0
        MOV A,20D
        ORL A,#10000000B     ;Units days
        MOV P0,A
        CLR P1.3
        LCALL WAIT
        SETB P1.3
        CLR P3.1
        CLR P2.4
        MOV P0,#00111111B
UNDAYS: SETB P3.3            ;Keep pin 3.3 ready for input
        JB P3.3,$
        LCALL KEYREAD
        PUSH 0
        MOV R0,20D
        LCALL DISPLAYSUB
        LCALL WAITASEC
        MOV R0,#06D
        LCALL DISPLAYSUB
        LCALL WAITASEC
        POP 0
        MOV A,20D
        ORL A,#10010000B     ;Tens days
        MOV P0,A
        CLR P1.3
        LCALL WAIT
        SETB P1.3
        CLR P3.1
```

```
           CLR P2.4
           MOV P0,#00111111B
TENSDAYS:  SETB P3.3              ;Keep pin 3.3 ready for input
           JB P3.3,$
           LCALL KEYREAD
           PUSH 0
           MOV R0,20D
           LCALL DISPLAYSUB
           LCALL WAITASEC
           MOV R0,#07D
           LCALL DISPLAYSUB
           LCALL WAITASEC
           POP 0
           MOV A,20D
           ORL A,#10100000B       ;Units months
           MOV P0,A
           CLR P1.3
           LCALL WAIT
           SETB P1.3
           CLR P3.1
           CLR P2.4
           MOV P0,#00111111B
UNMONTS:   SETB P3.3              ;Keep pin 3.3 ready for input
           JB P3.3,$
           LCALL KEYREAD
           PUSH 0
           MOV R0,20D
           LCALL DISPLAYSUB
           LCALL WAITASEC
           MOV R0,#08D
           LCALL DISPLAYSUB
           LCALL WAITASEC
           POP 0
           MOV A,20D
           ORL A,#10110000B       ;Tens months
           MOV P0,A
           CLR P1.3
           LCALL WAIT
           SETB P1.3
           MOV P0,#01H
           CLR P1.3
           LCALL WAIT
           SETB P1.3
           SETB P1.4
           SETB P1.5
           PUSH 0
           MOV R0,#99D
           LCALL DISPLAYSUB
           LCALL WAITASEC
           POP 0
           MOV SP,SAVESP
           CLR P3.1
```

```
          CLR P2.4
          MOV P0,#00001111B
          MOV IE,#85H
          RETI
GETNUM:   DEC R0                  ;R0 contains the value of sc.
                                  ;pad slot
GETWAIT:  SETB P3.3               ;Keep pin 3.3 ready for input
          MOV C,P3.3              ;Wait until released
          JNC GETWAIT
          CLR C
          PUSH 0
          MOV R0,#220D
          LCALL DISPLAYSUB
          LCALL WAITASEC
          CLR P3.1
          POP 0
          CLR P2.4
          MOV P0,#00001111B       ;Keep all rows sensitive
ALTRDY:   SETB P3.3               ;Keep pin 3.3 ready for input
          JB P3.3,$               ;Wait here in loop till a new
                                  ;key
                                  ;is depressed
                                  ;After a new key is depressed
                                  ;jump to
          PUSH 0
          MOV R0,#209D
          LCALL DISPLAYSUB
          LCALL WAITASEC
          CLR P3.1
          POP 0
          LCALL KEYREAD           ;keyread subroutine to read the
                                  ;key
          PUSH 0
          MOV R0,#210D
          LCALL DISPLAYSUB
          LCALL WAITASEC
          CLR P3.1
          POP 0
          CJNE R0,#20D,GETNUM ;Keep doing it until you get 3
                                  ;digits
          MOV R0,#22D
          MOV A,@R0               ;We have three digits in
                                  ;locations 20-22
          MOV B,#100D             ;the three digits are converted
          MUL AB                  ;into a single number and stored
                                  ;in locn. #23
          MOV @R0,A
          DEC R0
          MOV A,@R0
          MOV B,#10D
          MUL AB
          INC R0
```

```
          ADD A,@R0
          MOV @R0,A
          DEC R0
          DEC R0
          MOV A,@R0
          INC R0
          INC R0
          ADD A,@R0
          MOV 23D,A
          RET
KEYREAD:  SETB P3.3
          MOV P0,#11111111B    ;Reads the key and assigns the
          CLR P3.1
          CLR P2.4
          LCALL STARTDEB
          MOV A,P0             ;appropriate numeric value to
          RRC A                ;location pointed by register r0
          JNC CIRC4
          CLR C
          RRC A
          JNC CIRC3
          CLR C
          RRC A
          JNC CIRC12
          CLR C
          RRC A
          JNC CIRC11
          CLR C
          SETB P2.4
          MOV P0,#01111111B
          LCALL WAIT
          MOV A,P0
          RRC A
          JNC CIRC8
          CLR C
          RRC A
          JNC CIRC7
          CLR C
          RRC A
          JNC CIRC6
          CLR C
          RRC A
          JNC CIRC5
          CLR C
          MOV P0,#10111111B
          LCALL WAIT
          MOV A,P0
          RRC A
          RRC A
          RRC A
          JNC CIRC0
          CLR C
```

```
            RRC A
            JNC CIRC9
            CLR C
            MOV P0,#11011111B
            LCALL WAIT
            MOV A,P0
            RRC A
            JNC CIRCD
            CLR C
            RRC A
            JNC CIRCI
            CLR C
            RRC A
            JNC CIRCSD
            CLR C
            RRC A
            JNC CIRCMN
            CLR C
            MOV P0,#11101111B
            LCALL WAIT
            MOV A, P0
            RRC A
            JNC CIRCE
            CLR C
            MOV IE,#10000101B
            CLR P3.1
            CLR P2.4
            MOV P0,#00001111B
            RETI
CIRC11:     LJMP ONE
CIRC12:     LJMP TWO
CIRC3:      LJMP THREE
CIRC4:      LJMP FOUR
CIRC5:      LJMP FIVE
CIRC6:      LJMP SIX
CIRC7:      LJMP SEVEN
CIRC8:      LJMP EIGHT
CIRC9:      LJMP NINE
CIRC0:      LJMP ZERO
CIRCD:      LJMP DECREASE
CIRCI:      LJMP INCREASE
CIRCSD:     LJMP SD
CIRCMN:     LJMP MEAN
CIRCE:      LJMP EXIT
ONE:        MOV @R0,#1D
WONE:       SETB P3.3             ;Keep pin 3.3 ready for input
            MOV C,P3.3
            JNC WONE
            CLR C
            RET
TWO:        MOV @R0,#2D
WTWO:       SETB P3.3             ;Keep pin 3.3 ready for input
```

```
        MOV C,P3.3
        JNC WTWO
        CLR C
        RET
THREE:  MOV @R0,#3D
WTHREE: SETB P3.3             ;Keep pin 3.3 ready for input
        MOV C,P3.3
        JNC WTHREE
        CLR C
        RET
FOUR:   MOV @R0,#4D
WFOUR:  SETB P3.3             ;Keep pin 3.3 ready for input
        MOV C,P3.3
        JNC WFOUR
        CLR C
        RET
FIVE:   MOV @R0,#5D
WFIVE:  SETB P3.3             ;Keep pin 3.3 ready for input
        MOV C,P3.3
        JNC WFIVE
        CLR C
        RET
SIX:    MOV @R0,#6D
WSIX:   SETB P3.3             ;Keep pin 3.3 ready for input
        MOV C,P3.3
        JNC WSIX
        CLR C
        RET
SEVEN:  MOV @R0,#7D
WSEVEN: SETB P3.3             ;Keep pin 3.3 ready for input
        MOV C,P3.3
        JNC WSEVEN
        CLR C
        RET
EIGHT:  MOV @R0,#8D
WEIGHT: SETB P3.3             ;Keep pin 3.3 ready for input
        MOV C,P3.3
        JNC WEIGHT
        CLR C
        RET
NINE:   MOV @R0,#9D
WNINE:  SETB P3.3             ;Keep pin 3.3 ready for input
        MOV C,P3.3
        JNC WNINE
        CLR C
        RET
ZERO:   MOV @R0,#0D
WZERO:  SETB P3.3             ;Keep pin 3.3 ready for input
        MOV C,P3.3
        JNC WZERO
        CLR C
        RET
```

```
DECREASE: MOV A,20D
          CLR C
          SUBB A,#01D
          JC DECON
          MOV 20D,A
          LJMP DECOVR
DECON:    MOV A,21D
          CLR C
          SUBB A,#01D
          JC DECTO
          MOV 21D,A
          LJMP DECOVR
DECTO:    MOV A,22D
          CLR C
          SUBB A,#01D
          JC DECTRE
          MOV 22D,A
          LJMP DECOVR
DECTRE:   MOV A,23D
          CLR C
          SUBB A,#01D
          JC DECERR
          MOV 23D,A
DECOVR:   MOV SP,SAVESP2
          LJMP WDEC
DECERR:   CLR C
          MOV SP,SAVESP
          RET
WDEC:     SETB P3.3
          MOV C,P3.3
          JNC WDEC
          CLR C
          RET
INCREASE: MOV A,20D
          ADD A,#01D
          MOV 20D,A
          MOV SP,SAVESP2
WINC:     SETB P3.3             ;Keep pin 3.3 ready for input
          MOV C,P3.3
          JNC WINC
          CLR C
          RET
MEAN:     MOV R0,#24D
WMEAN:    SETB P3.3             ;Keep pin 3.3 ready for input
          MOV C,P3.3
          JNC WMEAN
          CLR C
          RET
SD:       MOV R0,#25D
WSD:      SETB P3.3             ;Keep pin 3.3 ready for input
          MOV C,P3.3
          JNC WSD
```

```
          CLR C
          RET
EXIT:     MOV IE,#85H
          MOV SP,SAVESP
          CLR P3.1
          CLR P2.4
          MOV P0,#00001111B
          RETI
DISPVALS: SETB P3.3                  ;Keep pin 3.3 ready for input
          MOV C,P3.3                 ;Stays here as long as key is
                                     ;depressed
          JNC DISPVALS
          CLR C
          PUSH 0
          MOV R0,#02D
          LCALL DISPLAYSUB
          LCALL WAITASEC
          CLR P3.1
          POP 0
          SETB P2.4
          MOV P0,#10101111B
DISPRDY:  SETB P3.3                  ;Stays here till next key is
                                     ;depressed
          JB P3.3,$
          LCALL STARTDEB
          MOV A,RETVAL
          RRC A
          SETB P2.4
          MOV P0,#10111111B          ;Jumps on currentval if that key
          LCALL WAIT
          JNC DISPVALS
          CLR C
          LCALL WAIT
          MOV A,P0                   ;is depressed
          RRC A
          JNC DEAD
          CLR C
          RRC A                      ;Jumps on resistval if that key
          JNC CIRCUMVENT             ;is depressed
          CLR C
          MOV P0,#11101111B
          LCALL WAIT                 ;Wait for things to settle
          MOV A,P0
          RRC A
          RRC A
          JNC TIMECIRC
          CLR C
          RRC A
          RRC A
          JNC HELP
          CLR C
          MOV IE,#10000101B
```

```
           CLR P3.1
           CLR P2.4
           MOV P0,#00001111B
DEAD:      RETI                      ;If neither is pressed go to
                                     ;main prog
TIMECIRC:  LJMP TIMEVAL
HELP:      LJMP RESISTVAL
CIRCUMVENT:     LJMP VOLTVAL
RESISTVAL:      SETB P3.3            ;Keep pin 3.3 ready for input
           MOV C,P3.3
           JNC RESISTVAL             ;Wait here until key is released
           CLR C
           PUSH 0
           MOV R0,#03D
           LCALL DISPLAYSUB
           LCALL WAITASEC
           CLR P3.1
           POP 0
           MOV DPTR,#00H
           LCALL GETFOURNUM
           LCALL MAGIC
           LCALL DISPLAYSUB
           LCALL WAITASEC
           MOV R0,DPH
           LCALL DISPLAYSUB
           LCALL WAITASEC
           MOV R0,DPL
           LCALL DISPLAYSUB
           LCALL WAITASEC
           CLR P3.1                  ;Reenable KP
           LJMP RESISTVAL
VOLTVAL:   SETB P3.3                 ;Keep pin 3.3 ready for input
           MOV C,P3.3
           JNC VOLTVAL               ;Wait here until key is released
           CLR C
           PUSH 0
           MOV R0,#04D
           LCALL DISPLAYSUB
           LCALL WAITASEC
           CLR P3.1
           POP 0
           MOV DPTR,#1024D
           LCALL GETFOURNUM
           LCALL MAGIC
           MOV A,R0
           CLR C
           RLC A
           JNC NOADTRE
           CLR C
           MOV A,R0
           ADD A,#100D
           MOV R0,A
```

```
          LJMP NOADTRE
NOADESC:  MOV A,R0
NOADTRE:  LCALL DISPLAYSUB
          LCALL WAITASEC
          CLR P3.1                ;Reenable KP
          LJMP VOLTVAL
TIMEVAL:  SETB P3.3               ;Keep pin 3.3 ready for input
          MOV C,P3.3
          JNC TIMEVAL            ;Wait here until key is released
          CLR C
          PUSH 0
          MOV R0,#05D
          LCALL DISPLAYSUB
          LCALL WAITASEC
          CLR P3.1
          POP 0
          MOV DPTR,#2048D
          LCALL GETFOURNUM
          LCALL MAGIC
          MOV A,DPH
          ADD A,#4D
          MOV DPH,A
          CLR P1.2
          MOVX A,@DPTR
          SETB P1.2
          MOV R3,A
          MOV A,R0
          MOV R4,A
          LCALL TIMEDISPLAY
          LJMP TIMEVAL
ADJRES:   CLR C                   ;Adjustment of resistance value
          MOV A,P2
          MOV R0,A
          ANL A,#11000000B
          CJNE A,#11000000B,NOTYET
          LJMP NOCURRENT
NOTYET:   MOV A,R0
          ADD A,#64D
          MOV P2,A
          MOV R0,#106D
          LCALL DISPLAYSUB
          LCALL WAITASEC
          CLR P3.5
          RET
NOCURRENT:     LJMP ALARM
          RET
WAITASEC: PUSH 3D
          PUSH 4D
          MOV R3,#100D            ;One second wait loop
THERE:    MOV R4,#255D
HERE:     LCALL WAIT
          DJNZ R4,HERE
```

```
          DJNZ R3,THERE
          POP 4D
          POP 3D
          RET
WAIT:     NOP
          NOP
          NOP
          NOP
          NOP
          RET
TIMEDISPLAY:   MOV A,R3
          SWAP A
          ANL A,#00001111B      ;Retreive months
          MOV R0,A
          LCALL DISPLAYSUB
          LCALL WAITASEC
          SETB P2.4
          MOV P0,#11101111B
          CLR P3.1              ;Reenable KP
          MOV A,R3
          RLC A
          CLR C
          ANL A,#00011111B
          MOV R3,A
          MOV A,R4
          RLC A
          JNC ESCLBIT
          CLR C
          LCALL LASTBIT
ESCLBIT:  MOV A,R3
          MOV R0,A
LOOK:     SETB P3.3             ;Keep pin 3.3 ready for input
          JB P3.3,$
          LCALL DISPLAYSUB      ;Display  date  after  key  is
pressed
          LCALL WAITASEC
          SETB P2.4
          MOV P0,#11101111B
          CLR P3.1              ;Reenable kp
          MOV A,R4
          ANL A,#00011111B
          MOV R0,A
LOOK1:    SETB P3.3             ;Keep pin 3.3 ready for input
          JB P3.3,$
          LCALL DISPLAYSUB      ;Display  time  after  key  is
pressed
          LCALL WAITASEC
          CLR P3.1              ;Reenable kp
          RET
LASTBIT:  MOV A,R3
          RRC A
          SETB C
```

```
        RLC A
        MOV R3,A
        RET
MAGIC:                          ;Compact 4 digits into a number
        MOV R0,#11D             ;and gets the value at that
locn.
        LCALL DISPLAYSUB        ;to R0
        LCALL WAITASEC
        CLR P3.1                ;Reenable keypad
        PUSH 1
        PUSH 2
        MOV A,23D
        MOV R1,A
        MOV R0,#00D
        LCALL MUL10
        MOV A,22D
        ADD A,R1
        MOV R1,A
        JNC MAGNEXT1
        CLR C
        MOV A,R0
        INC A
        MOV R0,A
MAGNEXT1:
        LCALL MUL10
        MOV A,21D
        ADD A,R1
        MOV R1,A
        JNC MAGNEXT2
        CLR C
        MOV A,R0
        INC A
        MOV R0,A
MAGNEXT2:
        LCALL MUL10
        MOV A,20D
        ADD A,R1
        MOV R1,A
        JNC MAGNEXT3
        CLR C
        MOV A,R0
        INC A
        MOV R0,A
MAGNEXT3:
        MOV A,R0
        ANL A,#4D
        JNZ MAGERR
        MOV A,R0
        ADD A,DPH
        MOV DPH,A
        MOV A,R1
        ADD A,DPL
```

```
           MOV DPL,A
           JNC MAGEND
           CLR C
           MOV A,DPH
           INC DPH
MAGEND:    MOV R0,#12D
           LCALL DISPLAYSUB
           LCALL WAITASEC
           CLR P3.1
           POP 2
           POP 1
           SETB P3.1
           SETB P3.5
           CLR P1.2
           MOVX A,@DPTR
           SETB P1.2
           MOV R0,A
           RET
MAGERR:    MOV R0,#13D
           LCALL DISPLAYSUB
           LCALL WAITASEC
           MOV SP,SAVESP
           MOV IE,#10000101B
           CLR P3.1
           CLR P2.4
           MOV P0,#00001111B
           RETI

MUL10:
           MOV B,#10D
           MOV A,R0
           MUL AB
           MOV R0,A
           MOV A,R1
           MOV B,#10D
           MUL AB
           MOV R1,A
           MOV A,B
           ADD A,R0
           MOV R0,A
           RET

GETFOURNUM:      MOV R0,#10D
           LCALL DISPLAYSUB
           LCALL WAITASEC
           CLR P3.1
           MOV R0,#23D
NEXTGET:   CLR P2.4
           MOV P0,#00001111B
           SETB P3.3
           JB P3.3,$                 ;Wait until key is pressed
           MOV SAVESP2,SP
```

```
            LCALL KEYREAD
            PUSH 0
            MOV A,@R0
            MOV R0,A
            LCALL DISPLAYSUB
            LCALL WAITASEC
            CLR P3.1
            POP 0
            DEC R0
            CJNE R0,#19D,NEXTGET
            SETB P3.1
            RET
STARTDEB:   MOV R6,#4D
STDEBTO:    MOV R7,#255D
STDEBON:    NOP
            NOP
            NOP
            NOP
            NOP
            NOP
            NOP
            NOP
            NOP
            NOP
            SETB P3.3              ;Keep pin 3.3 ready for input
            MOV C,P3.3
            JC  WRONG
            DJNZ R7,STDEBON
            DJNZ R6,STDEBTO
            MOV RETVAL,#1D
            RET
WRONG:      CLR C
            MOV RETVAL,#0D
            RET
DISPLAYSUB:
            PUSH P0
            SETB P3.5              ;Disable ADC
            SETB P1.3              ;Disable RTC
            SETB P3.1              ;Disable keypad
            MOV  A,R0
            MOV  B,#10D
            DIV  AB
            MOV  26D,B
            MOV  B,#10D
            DIV  AB
            MOV  27D,B
            MOV  B,#10D
            DIV  AB
            MOV  28D,B
            MOV  B,#10D
            DIV  AB
            MOV  29D,B
```

```
MOV   A,26D
SETB  ACC.4
SETB  ACC.5
MOV   P0,A
CLR   P1.1
SETB  P1.1
LCALL WAIT
MOV   A,27D
SETB  ACC.5
CLR   ACC.4
MOV   P0,A
CLR   P1.1
SETB  P1.1
LCALL WAIT
MOV   A,28D
SETB  ACC.4
CLR   ACC.5
MOV   P0,A
CLR   P1.1
SETB  P1.1
LCALL WAIT
MOV   A,29D
CLR   ACC.4
CLR   ACC.5
MOV   P0,A
CLR   P1.1
SETB  P1.1
LCALL WAIT
POP P0
RET
END
```

APPENDIX B

OPERATION MANUAL

General Information

The microcontroller based ground resistance monitor is used to measure the ground resistance of the substation grid. The instrument has a 4 K byte memory that will hold 1024 readings (for a period of over 2 years[***]) without any manual intervention. After the two year period, the older readings will be over-written. The instrument also records the hour, date and month of measurement. The voltage used for the measurement is also recorded and can be retrieved by the user. The mean and (the square of) the standard deviation of the past 64 readings of ground resistance can be obtained by the push of a button.

The instrument makes a measurement every 24 hours (at 0:00 hours). Measurements can however be made manually by the push of a button as and when it is necessary. The instrument turns on a visual indicator when the present value of ground resistance exceeds the preset value of ground resistance input during initialization.

The instrument has 3 power supply sockets. One marked for 5 Volts, one for -5 Volts and the third one is common. The instrument is provided with a power pack that has batteries to supply power in the event of a power outage. The power pack takes an input of 9 Volts and has regulators provided. Charging circuitry for the battery is also incorporated in the power pack.

4 sockets are provided by the instrument for the connection of the 4 probes. C1 is the current output pin and C2 is the chasis ground. The lines P1 and P2 are for connections to the auxiliary probes.

The instrument provides reaings with a margin of error of less than 5% full scale. The range of the instrument is 0.1 Ohms to 25.5 Ohms. The resolution is 0.1 Ohms. Interference effects are reduced using special measurement techniques and filters. The instrument outputs through a 4-digit LCD display.

[***] Calculation based on one reading per day.

Initialization of the Monitor

It is essential for the functioning of the monitor that the values of the standard resistance, minimum input current, maximum value of ground resistance before alarm is turned on are entered. This prevents the monitor from entering an indeterminate state. The mode of entry and the response from the monitor are described below. It is also advisable that the monitor be reset and the time be set immediately after the monitor is turned on.

Altering current

Action	Response
Press 'altpreset'	Display: 201, 202 203 (on release)
Press 'altcurrent'	Display: 204, 205 220 (on release)
Press digit one	Display: 209, 210, 220
Press digit two	Display: 209, 210, 220
Press digit three	Display: 209, 210, number entered.

Altering standard resistance

Action	Response
Press 'altpreset'	Display: 201, 202 203 (on release)
Press 'altstdres'	Display: 204, 206, 220

Then go ahead and enter the 3-digit number. The response will be the same as in the previous case of altering current.

Altering preset maximum ground resistance

Action	Response
Press 'altpreset'	Display: 201, 202 203 (on release)
Press 'ground resistance'	Display: 204, 207 220 (on release)

Then go ahead and enter the 3-digit number. The response will be the same as in the previous cases.

Altering time

Action	Response
Press 'altpreset'	Display: 201, 202 220 (on release)
Press 'time'	Display: 204, 208 209 (on release)
Enter units minutes	Display: 210, number entered, 1
Enter tens minutes	Display: number entered, 2

Enter units hours	Display:	number entered, 3
Enter tens hours	Display:	number entered, 4
Enter units date	Display:	number entered, 5
Enter tens date	Display:	number entered, 6
Enter units month	Display:	number entered, 7
Enter tens months	Display:	number entered, 8, 99

It has to be noted that in the above process, the number entered (except in the case of 'alttime') shall always contain 3 digits. For instance, a standard resistance value of 15 should be entered as 015.

Retreiving readings from the Monitor

Retreiving Voltage and Ground Resistance

Action		Response
Press 'dispvals'	Display:	201, 202 2 (on release)
Press 'voltage'	Display:	4, 10
Input 4 digits	Display:	Echo after each entry 11, 12, voltage, 4, 10

Further readings can be obtained by pressing 4 digits or by using the increase/decrease key. This process can be terminated using the key 'reset'.

The procedure is the same for ground resistance retreivals except the display of 4 being replaced by the number 6.

Retreiving time

Action		Response
Press 'dispvals'	Display:	201, 202 2 (on release)
Press 'time'	Display:	5, 10
Input 4 digits	Display:	echo after each digit 11, 12, months
Press 'time'	Display:	date
Press 'time'	Display:	hours, 5, 10.