DESIGN AND CONSTRUCTION OF A COMPUTER CONTROLED POWER MONITORING DEVICE

BY

KAMAL AHMAD FOLORUNSHO

REG NO.: 95/4522

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

FEDERAL UNIVERSITY OF TECHNOLOGY MINNA

A PROJECT REPORT SUBMMITED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR IN ENGINEERING (B. Eng.) DEGREE IN THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING, SCHOOL OF ENGINEERING AND ENGINEERING TECHNOLOGY, FEDERAL UNIVERSITY OF TECHNOLOGY MINNA, NIGER STATE, NIGERIA

DECEMBER 2000

DECLARATION

I **Kamal Ahmad F**. hereby declare that this project is an original concept of mine. It was designed, constructed and tested under the supervision of **Mr. Kenneth Pinne** my project supervisor.

The project has not been presented in any form for the award of diploma or degree certificate. All information extracted and derived from published and unpublished work has been acknowledged.

....

KAMAL, AHMAD FOLORUNSHO DATE

CERTIFICATION

This is to certify the project titled design and construction of a computer controlled power monitoring device was carried out by **Kamal Ahmad F**. under the supervision of **Mr. Kenneth Pinne** and submitted to the electrical and computer engineering department of the federal university of technology Minna, in partial fulfillment of the requirement for the award of a bachelor (B ENG) degree in electrical and computer engineering

Mr. Kenneth Pinne

Project supervisor

Signature & date

Dr. Y. A. Adediran

H. O. D. Elect. / Compt. Eng.

------------------------------------ 16/1/01

Signature & date

External supervisor

------------------------------------ 17/1/01

Signature & date

DEDICATION

To Allah the most gracious and the most merciful

ACKNOWLEDGEMENT

My appreciation goes to God for his infinite mercy, blessing and guidance. I also thank my family Alhaji Ahmad O. Kamal, Alhaja Aishat A. Kamal, Niyi Kamal, Sherifat Kamal and Rahmat & Abdul Rasheed Na'allah for their love, perseverance and support throughout my stay at the university. My sincere appreciation goes to my guardian who has also been a pillar of support for me till the very end. And my supervisor, Mr. Kenneth Pinne for being so helpful and supportive.

I thank my friends and well wishers who have been there for me throughout my days in school. People like Abdullahi Abdul Azeez, Bode Akinyemi, Yerima Suleiman, Hassan Kibiya, Shola Sanni, Elizabeth Ejele, Amina Ndatsu, Taranum Reyaz, Kamaldeen Sanni others are Patience, Gbolabo, Sherrif, Nurudeen, Niran, Olamide and Toyin. My lecturers, Dr. Y. Adediran, Mr. Danjuma, Mr. Paul Atah and others.

This project is here today despite the stress and strain of university education in Nigeria because there is God by my side and the collaborative effort of all the people mentioned above and those that were not mentioned at all, once again I thank you all.

ABSTRACT

The final year project is supposed to give students the opportunity to show their ingenuity and understanding of their individual fields of study. This project is done with the hope of improving the present state of industrial measurements and control in the power generation and other sectors of the Nigerian economy. By taking the first step, which is using the computer to measure and monitor power, I believe that the dream of using the computer to control a power generating or transmitting station with little or no human interference will one day be achieved.

TABLE OF CONTENTS

CHAPTER ONE

INTRODUCTION

When the word computer is mentioned we think about desktops and laptop personal computers. But actually microcomputers vary from single chip systems such as calculators to industrial controllers. In this present age computers are used in science and technology space military and industrial applications and this is made possible by computers being able to take input and give out output to mechanical or electronic devices other than the keyboard, monitor or hard disk drives this particular project will provide an insight on a new way of using the computer for interfaced through the parallel port.

The computer-controlled power-monitoring device, brings about the definition of the word power, power is the rate at which energy is expended. This definition shows the close relationship between energy and power, in the present world of resource and energy management, power monitoring becomes important especially in the case of electrical power in which a fixed range of power is always required for a particular device. And anything outside this range could damage the device. Power monitoring is an important feature of the generating and the consuming ends of the international power sector. Using the computer to control a power-monitoring device gives a good degree of sensitivity, accuracy speed and also gives an insight into industrial control, because the control to be effected on a system is a function of the output of that system. If the computer can be used to measure and monitor the power output of a system then it could be made to totally control that system

1.1 AIMS AND OBJECTIVES

The aim of this project is to produce a computer-controlled power monitoring device which can be used as:

i. A measuring to measure voltage and power

ii. To serve as a step towards using the computer as a control tool because the control tool because controlling the input of a system could be a function of the output power.

iii. To give deep insights into interfacing through the parallel port of an IBM compatible PC which is originally designed to be a printer port

1.2 LITERATURE REVIEW

The computer controlled power-monitoring device is an intelligent instrument developed and used for monitoring power. Although the sensory organ of the human body can be extremely sensitive and responsive modern science and technology rely in the development of much more precise measuring and analytical tools for the studying, monitoring and controlling of all kinds of phenomenon.

Highly developed instrumentation was born during the industrial revolution of the 18^{th} and 19^{th} century, particularly in the areas of dimensional measurement, electrical measurement and physical analysis. Manufacturing processes of the time required instruments capable of achieving new standards of linear precision. The industrial application of electricity required instruments to measure current, voltage, resistance and power. Analytical methods using such instruments, as the microscope and spectroscope, became increasingly important, light radiation given off by incandescent substances, began to be used to identify the composition of chemical substances and stars.

In the 20th century, the growth of modern industry, the introduction of computerization, and the advent of space exploration have spurred still greater development of instrumentation, particularly of electronic devices. Often a transducer, an instrument that changes energy from one form to the other (such as thermistor, photocell, or microphone) is said to transform a sample of the energy to be measured into electrical impulses that are more easily processed and stored. The introduction of the electronic computer in the 1950s, brought with it great capacity for information process and storage, virtually revolutionalized methods of instrumentation, for it allowed the simultaneous comparisons and analysis of large amounts of information.

Today there are lots of computerized measuring instruments used in industries. Instrument computerization has been via the internal expansion slot inside the microcomputer, which has the possibility of damaging the entire computer when the instrument is wrongly operated. Computerization is also done via the printer port, which is safer. The low level and DOS commands in C++ programming language used in this project opens the way to instrument computerization through the IBM compatible PC parallel port is presented in this project.

All the information on how the Computer Controlled power measuring and monitoring device was developed in four chapters.

1.3 PROJECT OUTLINE

- CHAPTER ONE: A general introduction to the project, aims and objectives of the project, literature review and an insight to the necessity of the invention.
- CHAPTER TWO: Deals with the system design discussed here, the procedure taken in designing the various stages and modules of the project
- CHAPTER THREE: Discussed the construction, testing and results obtained
- CHAPTER FOUR: contains the conclusion recommendations

CHAPTER TWO

SYSTEM DESIGN

The computer controlled power measuring and monitoring device (CCPMMD) is a peripheral hardware connected to the parallel port of an IBM compatible PC and controlled by software, device driver. Thus it can be said to have two principal aspects hardware and software.

The quantity to be measured initially is AC-voltage; the voltage conditioning circuit first conditions it. The function of the voltage conditioning circuit is scaling with the use of voltage divider and then a series Ik* resistor that acts as a current limiter and also the conversion factor when the voltage is being converted to power. The analogue to digital converter {ADC) is software controlled. The function of the ADC is to convert an analogue signal to an 8-bit digital signal that will be fed into the computer parallel port. The software then processes the digital data. This process includes conversion from reference value to actual value, voltage scaling, conversion to power and then status determination. Based on the status of the power under determination a message is prompted on the monitor screen.

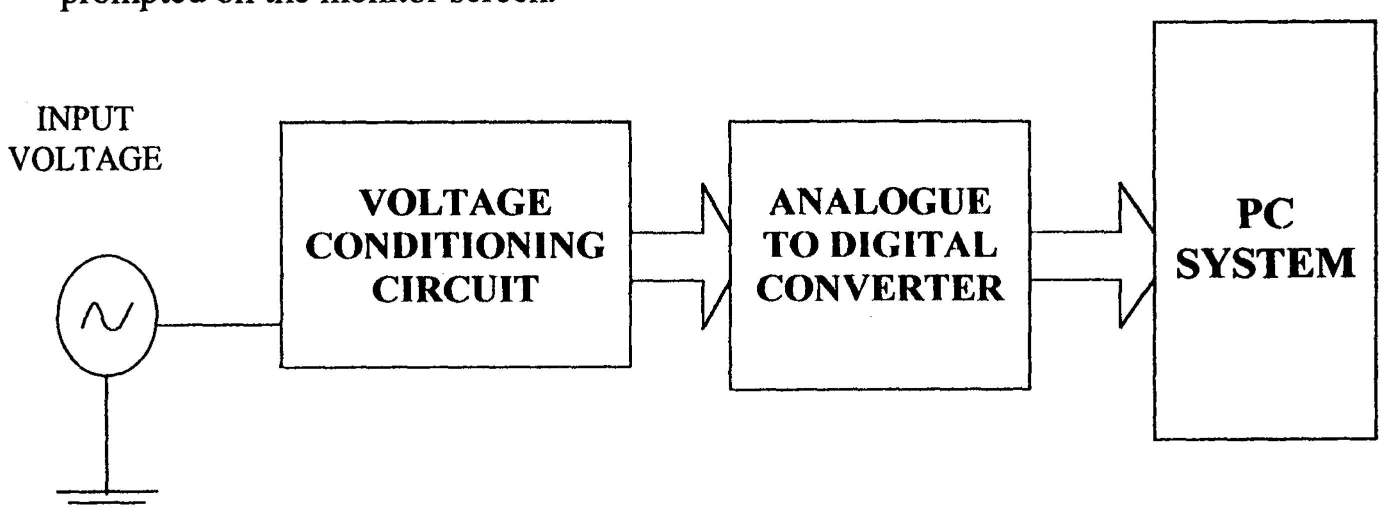


FIG 2.1 BLOCK DIAGRAM OF A HARDWARE ASPECT OF THE CCPMD

2.1 OPERATION OF THE PARALLEL PORT

The parallel port (printer port) is found in any of these two ways

- On a multi-input/output (1/0) card

- On the motherboard

The PC addresses or accesses the various 1/0 PORTS by using a unique address code. Each device on board in the computer has an address that no other device in the PC shares with it.

The parallel port on most IBM compatible start with an ID of 0378 base 16 (Hexadecimal). There are up to three ports in the parallel port, the data port, the status port and the control port. The starting address is used for the data port and the next address is used for the status port and then followed by the control port. The FIG 2.0 below shows the port map of the printer interface port.

ADDRESS (HEX)	PORT	PORT NAME
0378	7 6 5 4 3 2 1 0 Data bits 0 - 7	DATA PORT
0379	7 6 5 4 3 2 1 0 Busy (7), Acknowledge (6), out of paper (5), selected (4), error (3), not used (2, 1, 0)	STATUS PORT
037A	7 6 5 4 3 2 1 0 Not used (7, 6, 5, 4), Not select (3), initialise (2), auto-linefeed (1), strobe (0)	CONTROL PORT

FIG 2.2 PARALLEL PORT MAP

The parallel port on the IBM compatible PC is a 25-pin connector, often referred to as a DB-25 connector. Fig 2.3 shows the pin out for the connector. Note that only a little more than half of the pins are used. The others are grounded.

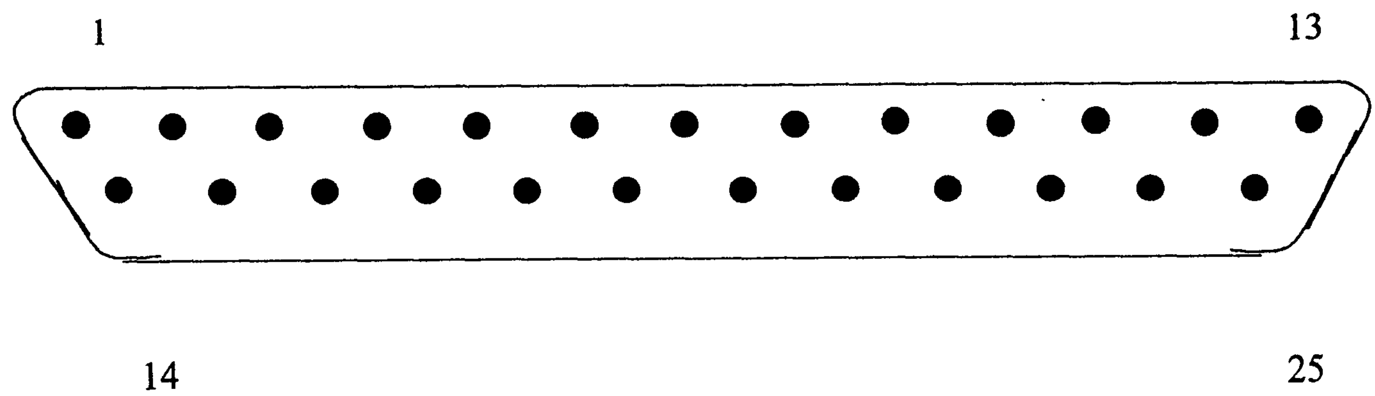


FIG 2.3 PARALLEL PORT PIN OUT

TABLE 2.1 PINS AND FUNCTIONS ON A PARALLEL PORT

PIN	FUNCTION
1	STROBE
2	DATA 0 (LSB)
3	DATA 1
4	DATA 2
5	DATA 3
6	DATA 4
7	DATA 5
8	DATA 6
9	DATA 7 (MSB)
10	ACKNOWLEDGE (STATUS)
11	BUSY (STATUS)
12	OUT OF PAPER (STATUS)
13	SELECTED (STATUS)
14	AUTO LINE FEED (COUNTROL)
15	ERROR (STATUS)
16	INITIALISE (COUNTROL)
17	SELECT (COUNTROL)
18 - 25	GROUND

From Fig 2.3 and Table 2.1 it can be determined that the data port has 8 lines, the status port has 5 and the control port has 4 lines.

2.2 VOLTAGE CONDITIONING CIRCUIT

The measured voltage is passed through a voltage divider circuit consisting of resistors R1 and R2. This acts as a scaling factor of the incoming voltage. And also to rectify incoming voltage.

Because

$$\frac{V_X}{R1 + R2} = \frac{V_{IN}}{R2}$$

Therefore $$V_{IN} = V_X \left[\frac{R2}{R1 + R2}\right]$$

So $\left[\frac{R2}{R1 + R2}\right]$ acts as a scaling factor which is used in the

Computer software aspect of the project. The resistor R3 acts as a current limiter

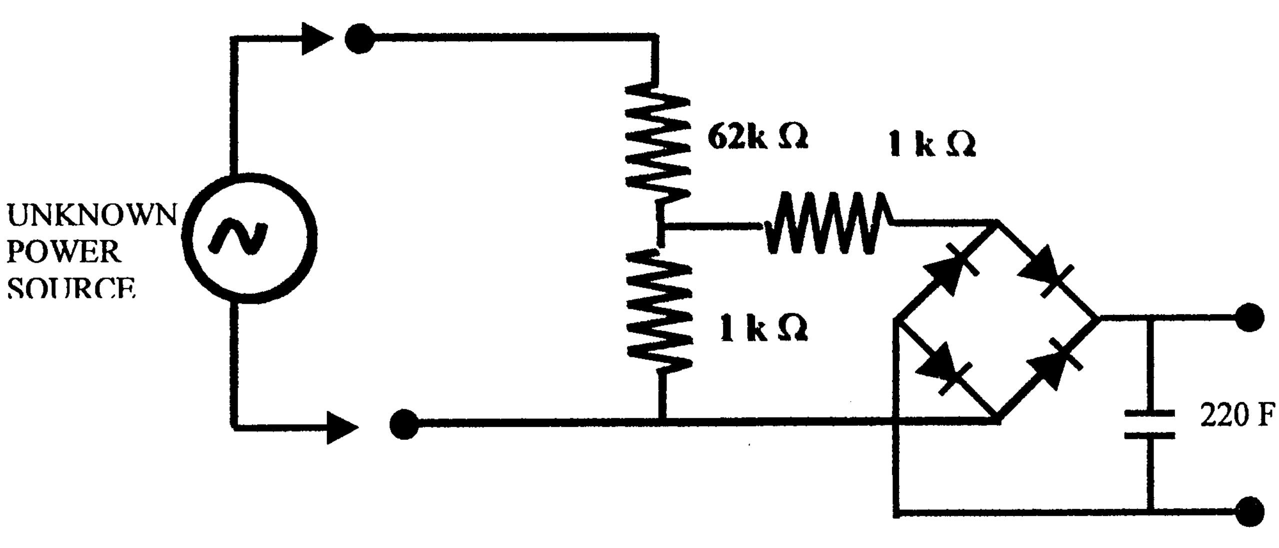


FIG 2.4 BLOCK DIAGRAM OF VOLTAGE CONDITIONING CIRCUIT

2.3 ANALOGUE TO DIGITAL CONVERTER

An ADC takes the instantaneous value of an analogue input signal and then produces as its output a coded digital word with a weight that corresponds to the level of the analogue signal. The ADC has some degree of uncertainty over the conversion signal and can take any value within a defined range. Whereas the digital output can only exist as a fixed number of discrete codes. The uncertainty of an ADC is called quantising error and ±0.5 of the least signal bit consider as 8-bit ADC with an input span of 5v used in the project. The digital output, an 8-bit code takes values from 00000000, 00000001, 00000010 and so on output 11111111, which means that it has 256 quantisation levels. Suppose that the analogue input is 1.25V, then the digital code is 0011 1111. When the input is 1.27V, then the digital code changes to 0100 0000, that is on LSB step. However an analogue value of 1.26v, just halfway between 1.25v and 1.27v could give a digital code of 00111111 or 010000000 this is what we refer to as the uncertainty of an ADC.

Another important parameter of an ADC is its conversion time. The time interval between the command, being given to the ADC to begin the conversion, and the appearance at the output, of the complete digital equivalent of the analogue value. The speed of conversion varies with the type of ADC and can be as short as a few nano seconds for the ultra fast types, or as slow as several milliseconds for the others.

2.3.1 PRACTICAL ADC CIRCUIT

A wide variety of methods are used in analogue to digital conversion. These range from the slow and inexpensive to the very fast types which are therefore relatively costly. The common methods are

- Voltage to frequency
- Parallel or flash conversion
- Single ramp and counter
- Dual / triple ramp
- Successive approximation

The method used in the project is the successive approximation method. This is a popular method for use in microprocessor systems since it is relatively fast, has good accuracy and can be software controlled. A typical successive approximation converter is shown in Fig 2.4.

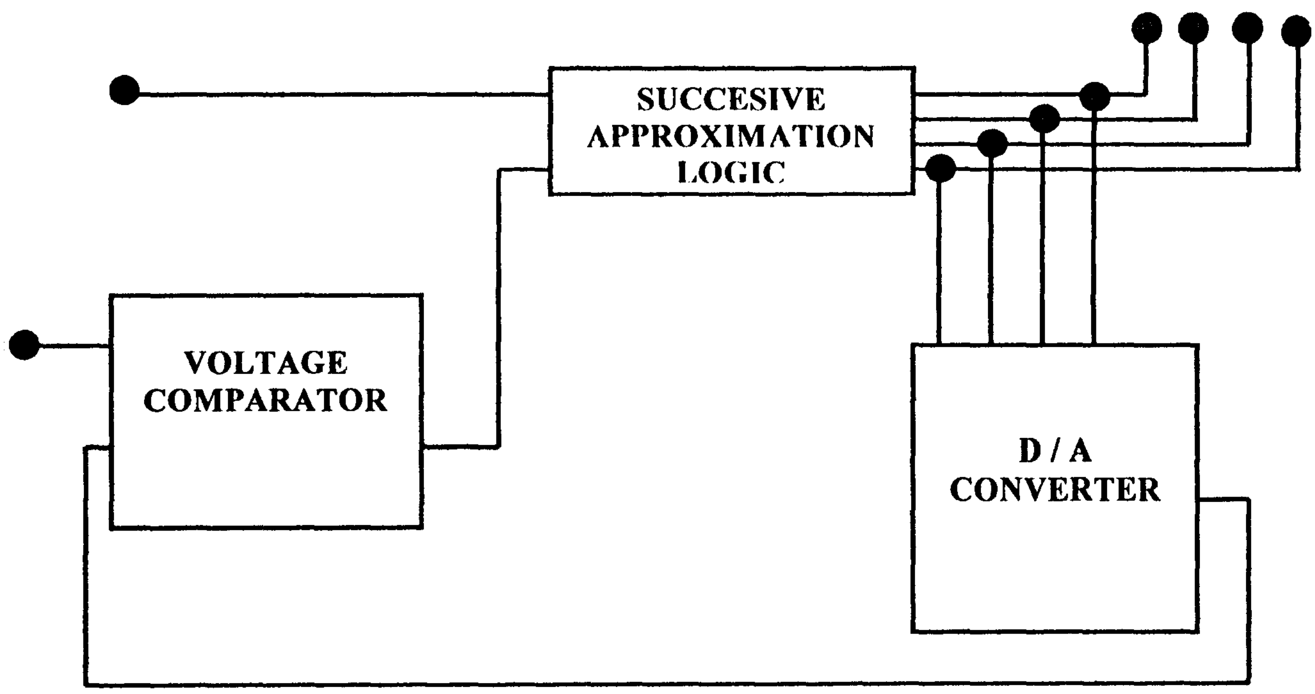


Fig 2.4 DIAGRAM OF A SUCCESSIVE APPROXIMATION-TYPE A/D CONVERSION

At the start of the conversion, the MSB of a register in the successive approximation logic is set to 1 by the programming logic so that the 8-bits the register reads 1000000, this in turn is converted by the DAC and the value compared with the analogue input. Suppose it is less than V_{in}, the logic 1 in that position is retained and next most sufficient bit is also set to I (register holds 0110 0000) and used for comparism. This process continues until all bits have been tried and a point of balance is reached; that is when V_{in} is just greater than DAC output voltage. Then the resultant digital code on the register is related to the output of the A/D converter.

2.3.2 AN ANALOGUE TO DIGITAL CONVERTER INTEGRATED CIRCUIT (ADC 0804)

In this project ADC 0804, an 8-bit A/D converter IC is used and is shown in fig 2.5 and a list of the name and function of each pin on the IC is shown in TABLE 2.2. The ADC 0804 was designed to interface directly with the 8086 or Z80 microprocessors. Some pin labels on the ADC 0804 correspond to pin on popular microprocessors. For instance, the ADC 0804 uses RD. WR and NTR as pin labels, which corresponds to the RD, WR and INTR, pins on the 8085 microprocessor. The ADC 0804 can also be interfaced with other popular 8-bits microprocessors such as the Motorola 6800 and 6502.

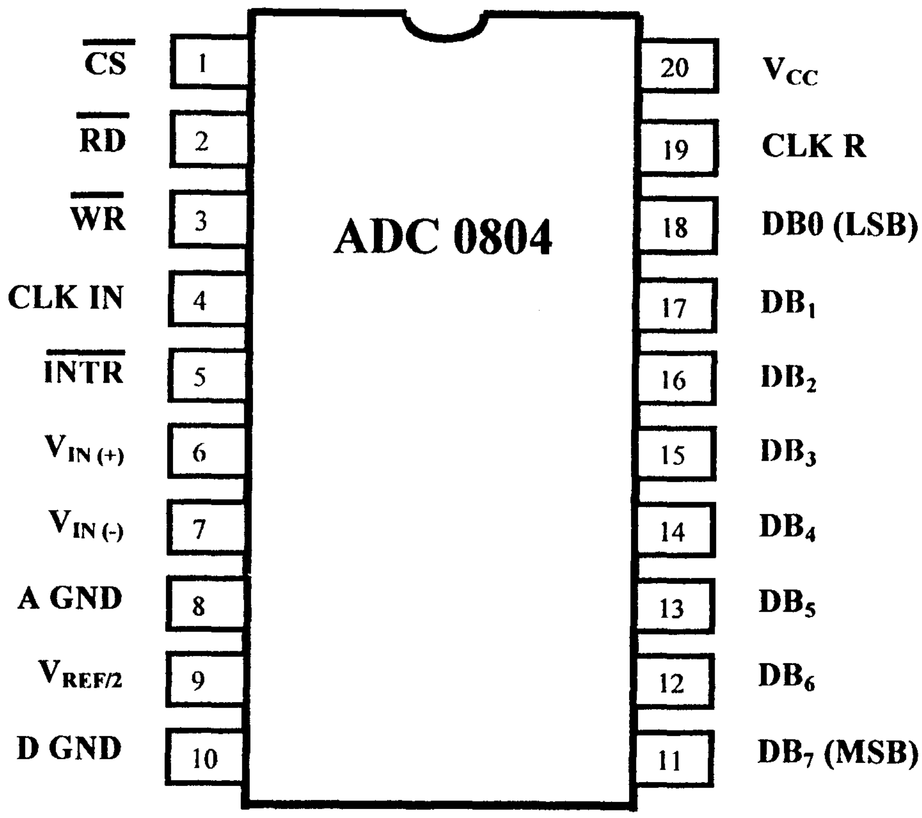


FIG 2.4.1 ADC 0804 ANALOGUE TO DIGITAL CONVERTER INTERGRATED CIRCUIT

The ADC 0804 is a CMOS 8-bit successive approximation A/D converter. It has three state output so that it can interface directly with a microprocessor based system data bus. The ADC 0804 had binary output and features a short conversion time of only 100microseconds. It input and outputs are both MOS and TTL compatible. It has an on-chip clock generation. The on-chip generator needs two external components (resistor or capacitor) to operate. The ADC 0804 operator on a standard +5v dc power supply and can encode input analogue voltages ranging from 0-5v the wiring of ADC 0804 used in this project is shown in fig. 2.5

TABLE 2.2 PIN LABELS AND FUNCTIONS OF THE ADC 0804

PIN OUT	SYMBOL	INPUT /OUTPUT OR POWER	DESCRIPTION
1	CS	Input	Chip select line from microprocessor
2	RD	Input	Read line from microprocessor
3	WR	Input	Write line from microprocessor
4	CLK IN	Input	Clock
5	INTR	output	Interrupt line (goes to microprocessor interrupt input)
6	VIN (+)	Input	Analogue voltage (positive input)
7	V_{IN} (-)	Input	Analogue voltage (negative input)
8	A GND	power	Analogue ground
9	V_{REF}	input	Alternate voltage reference
10	D GND	power	Digital ground
11	DB7	output	Data output (MSB)
12	DB6	output	Data output
13	DB5	output	Data output
14	DB4	output	Data output
15	DB3	output	Data output
16	DB2	output	Data output
17	DB1	output	Data output
18	DB0	output	Data output (LSB)
19	CLKR	input	Connect external resistor for clock
20	VCC(or ref)	POWER	Positive of 5v power supply and primary reference voltage

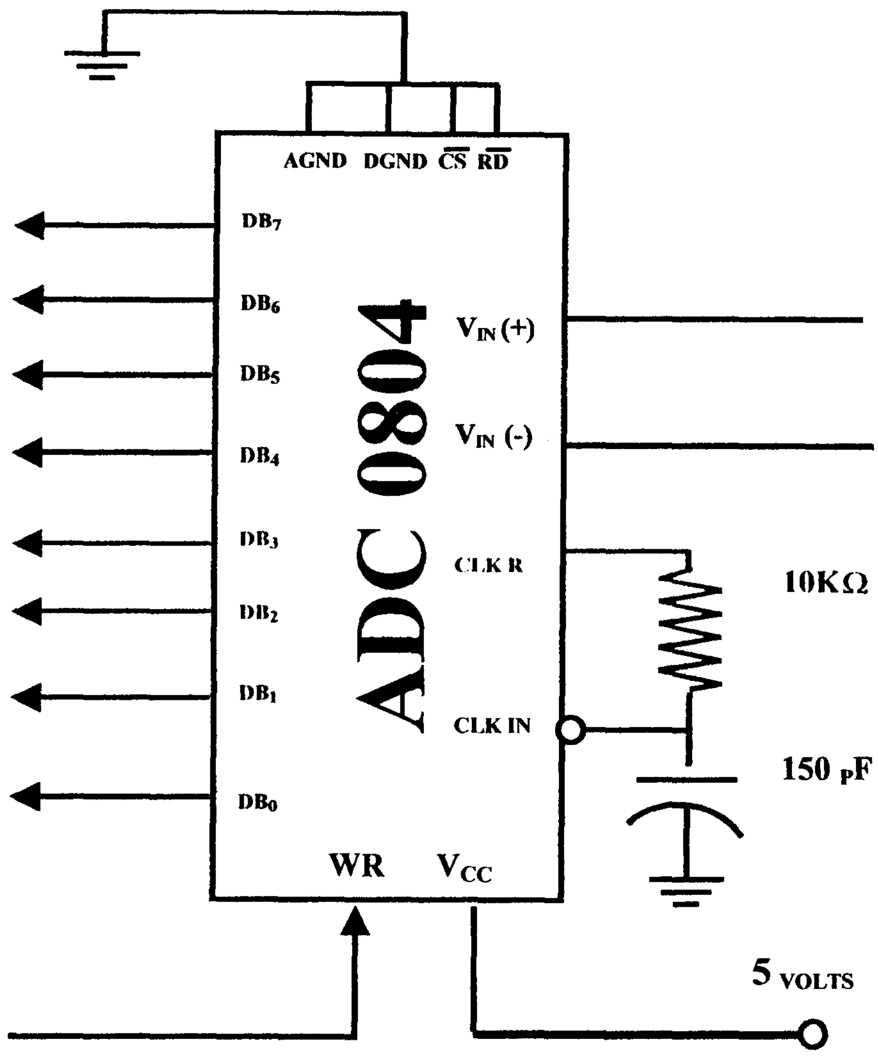


FIG 2.6 WIRING DIAGRAM OF THE ADC 0804

The function of the ADC in this project is to encode the difference in voltage between V_{in} (+) and V_{in} (-) compared to the reference voltage, 5v to a corresponding binary value. For instance, the resolution of the ADC 0804 is 8 bits or 0.39 percent. This means that for each 0.02v (0.39% of 5v = 0.02) increase in voltage at the analogue inputs the binary count increase by 1. To start conversion, a signal is applied to the WR input, this starts the A / D conversion process. When the conversion is finished. The binary output is updated. Though it is possible to use a timer (e.g. 555 timer) in this project it will be timed by the PC through the parallel ports, control port. Due to the fact that the ADC 0804 uses the successive approximation technique in its conversion process, it has a high conversion rate. The resistor R1 and capacitor C1 connected to the CLKR and CLKIN inputs to the ADC0804 course the internal clock to operate. The valves of R1 and C1 used in this project are the manufacturer recommendations.

2.4 THE SOFTWARE

In this project the PC controls the peripheral device. The function of the software that handles this processes are: -

Step 1: To tell the ADC 0804 to start conversion

Step 2: To read the incoming digital word from the ADC

Step 3: To convert the voltage from the reference valve to actual value taking the scaling factor into consideration

Step 4: To convert power voltage to power using the fixed resistor value (R3)

Step 5: To check the status of the power line according to desired values.

Step 6: To report status or if necessary provide an audio visual Alarm.

The software is developed using the C++ programming language. C++ is a high level language with some low level capability. It is a structured programming language with object oriented programming features. It was referred to as A then B and then C. With recent developments it has been upgraded to what we now know C++

C++ was invented in 1980 by Bjarne Straustrop at bell laboratories in Murray Hall, New Jersey; it was changed to C++ in 1983. Since then it has undergone two major revisions one in 1985 and the other in 1990 C++ is noted for its object orientation and it is most commonly used in the development of operating systems, system software and Hardware device drivers. It is very good and portable due to its machine language compatibility. The software in this project has been made to be user friendly and mouse enabled.

The flowchart for this program is show in Fig. 2.7.

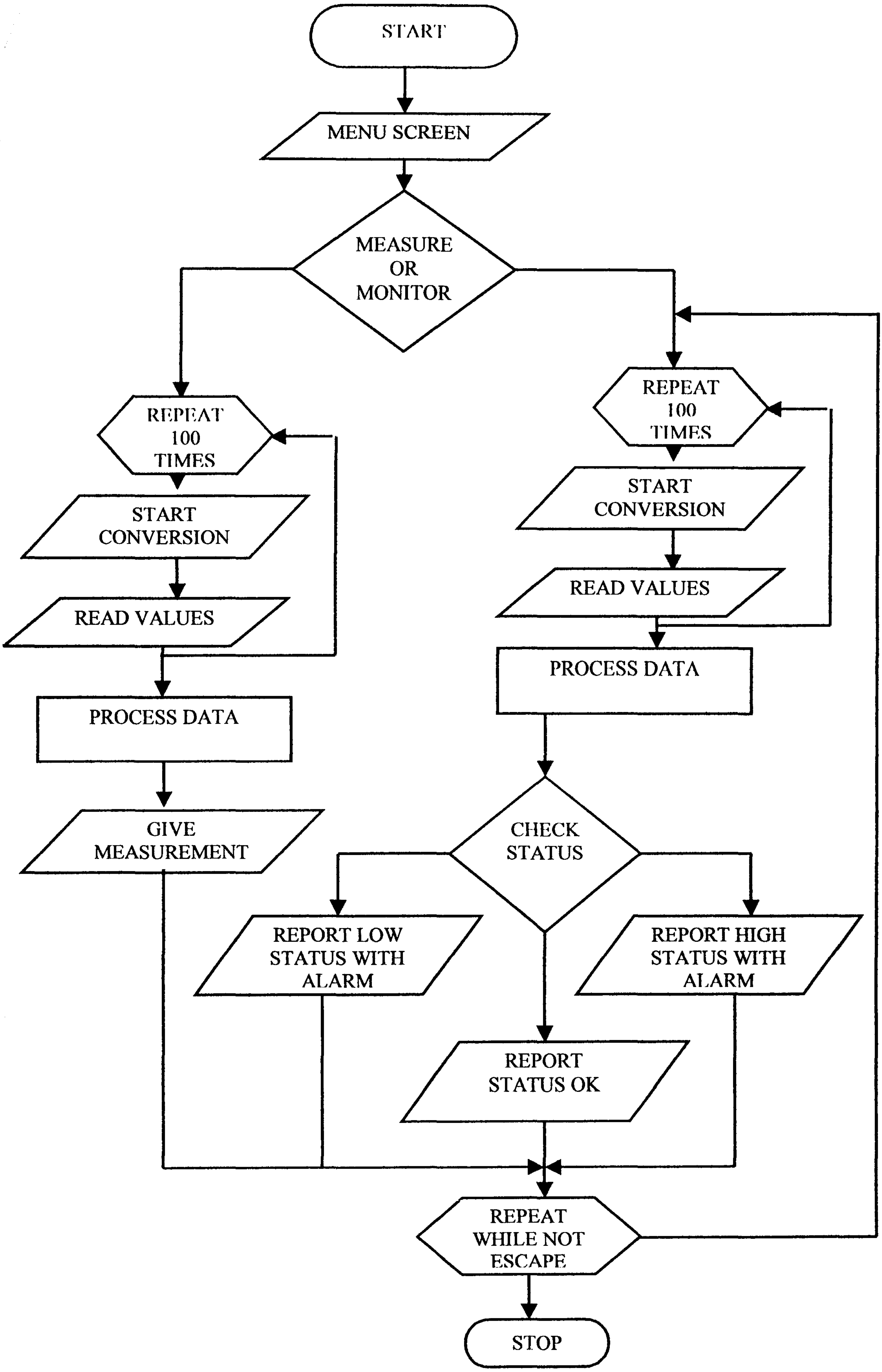


FIG 2.7 SOFTWARE FLOWCHART

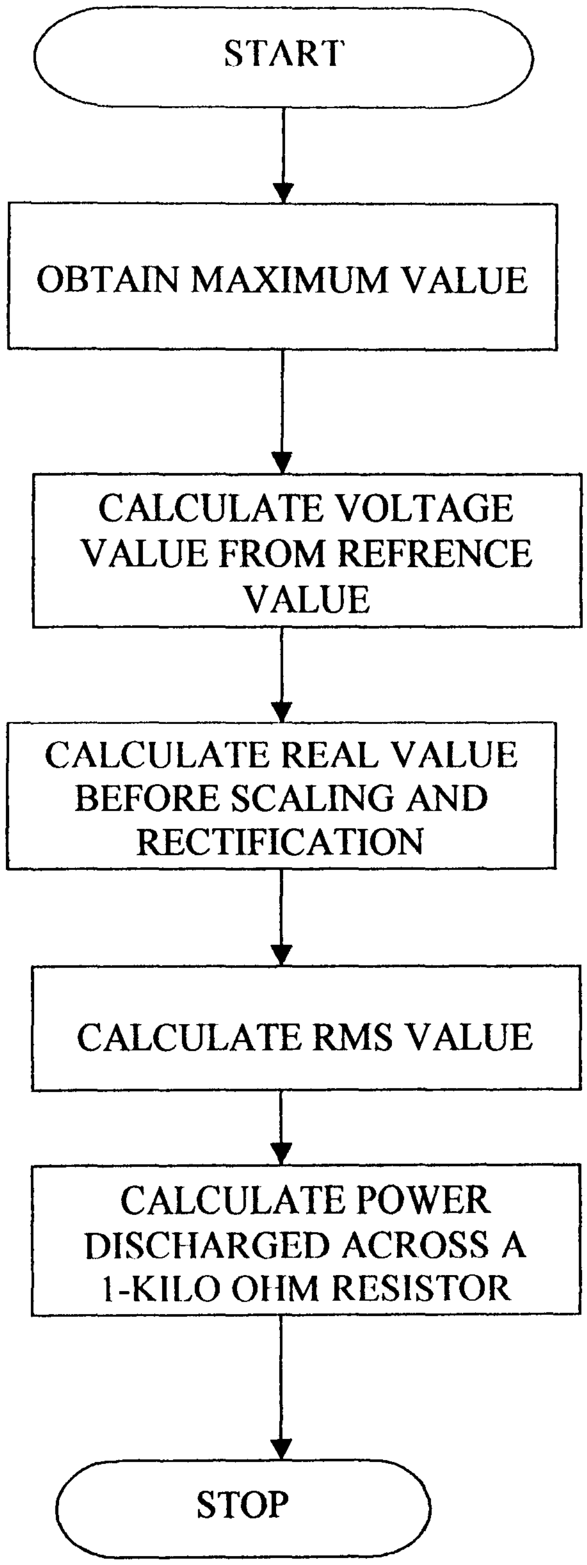


FIG 2.8 EXPANDED FLOWCHART ON PROCESS DATA

2.4.1 SOURCE CODE LISTING

```
#include <dos.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <graphics.h>

void intro(void);
void outro(void);
void choice();
void measure(void);
void monitor(int we);
void alarm(char status);
double reader(void);
float req=48.4;

void show_mouse()
{
      union REGS registers;
      registers.x.ax=1;
      int86(0x33,&registers, &registers);
}

inline void interupt(int &c,int &d, int &b)
{
      union REGS registers;
      registers.x.ax=3;
      int86(0x33,&registers, &registers);
      c=registers.x.cx;
      d=registers.x.dx;
      b=registers.x.bx;
}

main()
{
int o;
intro();choice();return 0;
}


void intro(void){
 int i,y=20,maxx,maxy;char *intros[15];int gdriver = DETECT,gmode,errorcode;
 initgraph(&gdriver,&gmode,"C:\\borlandc\\bgi");
 errorcode = graphresult();
 if(errorcode != grOk){
      printf("graph error : %\n",grapherrormsg(errorcode));
      printf("press any key to halt:");
      getch();exit(1);}
 maxx = getmaxx();
 maxy = getmaxy();
 cleardevice();gotoxy(1,1);
 intros[0] = "DESIGN AND CONSTRUCTION OF A COMPUTER CONTROLED";
 intros[1] = "POWER MEASURING AND MONITORING DEVICE";
 intros[2] = "BY";
 intros[3] = "KAMAL AHMAD F.";
 intros[4] = "95/4522EE ";
 intros[5] = "UNDER THE SUPERVISION OF MR. KENNETH PINNE";
 intros[6] = "DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING";
 intros[7] = "SCHOOL OF ENGINEERING AND ENGINEERING TECHNLOGY" ;
 intros[8] = "FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA";
 intros[9] = "SUBMMITTED IN PARTIAL FULFILMENT FOR ";
 intros[10] = "THE AWARD OF BACHELOR (B. ENG) DEGREE";
 intros[11] = "IN ELECTRICAL AND COMPUTER ENGINEERING";
 intros[12] = "";
 intros[13] = "DECEMBER 2000";
 intros[14] = "Press any key to continue";
 bar(0,0,maxx,maxy); setfillstyle(SOLID_FILL,BLUE);
 bar(5,5,maxx-5,maxy-5); settextjustify(CENTER_TEXT,CENTER_TEXT);
 settextstyle(SANS_SERIF_FONT, HORIZ_DIR,1);
 for(i = 0 ; i <=14 ; i++){
```

```
   if(i==3){ settextstyle(GOTHIC_FONT, HORIZ_DIR,4);y+=10;}
   if(i==4){ settextstyle(SANS_SERIF_FONT, HORIZ_DIR,2);y-=3;}
   if(i==0)  settextstyle(TRIPLEX_FONT, HORIZ_DIR,2);
   if(i==1)  settextstyle(TRIPLEX_FONT, HORIZ_DIR,2);
   if(i==10){ settextstyle(SANS_SERIF_FONT, HORIZ_DIR,3);y+=10;}
   if(i==11){ settextstyle(SANS_SERIF_FONT, HORIZ_DIR,3);y-=5;}
   outtextxy(maxx/2,y,intros[i]);
   settextstyle(SANS_SERIF_FONT, HORIZ_DIR,1);
   y = y + 30;}
getch();cleardevice();gotoxy(1,1);
      union REGS registers;
      registers.x.ax=0;
      int86(0x33,&registers, &registers);
}

void choice()
{ upe:
int x = getmaxx();int y = getmaxy();
      union REGS registers;
      registers.x.ax=2;
      int86(0x33,&registers, &registers);
cleardevice();
setcolor(WHITE);rectangle(0,0,x,y);
setfillstyle(SOLID_FILL,LIGHTBLUE);
floodfill(2,2,WHITE);

setcolor(WHITE);
rectangle(0,0,640,30);
setfillstyle(SOLID_FILL,DARKGRAY);
floodfill(1,1,WHITE);setcolor(WHITE);
settextstyle(TRIPLEX_FONT, HORIZ_DIR,2);
settextjustify(CENTER_TEXT,CENTER_TEXT);
outtextxy(x/2,12,"MAIN MENU");

setcolor(BLACK);
setlinestyle(SOLID_LINE,0,2);
rectangle(100,170,200,200);
setfillstyle(SOLID_FILL,BROWN);
floodfill(101,171,BLACK);setcolor(WHITE);
line(100,170,199,170);line(100,170,100,199);
setcolor(BLACK);outtextxy(150,180,"MEASURE");

setcolor(BLACK);
setlinestyle(SOLID_LINE,0,2);
rectangle(400,170,500,200);
setfillstyle(SOLID_FILL,BROWN);
floodfill(401,171,BLACK);setcolor(WHITE);
line(400,170,499,170);line(400,170,400,199);
setcolor(BLACK);outtextxy(450,180,"MONITOR");

setcolor(BLACK);
setlinestyle(SOLID_LINE,0,2);
rectangle(250,320,350,350);
setfillstyle(SOLID_FILL,DARKGRAY);
floodfill(251,321,BLACK);
setcolor(WHITE);line(250,320,349,320);
line(250,320,250,349);setcolor(BLACK);
outtextxy(300,330,"QUIT");

int c=0,b=0,d=0,q=0;
up:
show_mouse();interupt(c,d,b);
if (c <= 200 && c >=100 && d<=200 && d>=170 && b==1)measure();
if (c <= 500 && c >=400 && d<=200 && d>=170 && b==1){monitor(q);goto upe;}
if (c <= 350 && c >=250 && d<=350 && d>=320 && b==1)outro();
if (!kbhit()) goto up;}

void outro(void)
{       union REGS registers;
      registers.x.ax=2;
      int86(0x33,&registers, &registers);
```

```
        int i,y=200,maxx,maxy; char *goo[3];
        maxx = getmaxx();
        maxy = getmaxy();cleardevice();gotoxy(1,1);
        goo[0] = "THIS IS THE END OF THE PROGRAM ";
        goo[1] = "PIRACY IS A CRIMINAL OFFENCE";
        goo[2] = "Press any key to Exit";
        setfillstyle(SOLID_FILL,WHITE);
        bar(0,0,maxx,maxy); setfillstyle(SOLID_FILL,BROWN);
        bar(5,5,maxx-5,maxy-5); settextjustify(CENTER_TEXT,CENTER_TEXT);
        settextstyle(TRIPLEX_FONT, HORIZ_DIR,4);
        for(i = 0 ; i <=2 ; i++){
        if(i==2){y+=80;settextstyle(SANS_SERIF_FONT, HORIZ_DIR,2);}
        outtextxy(maxx/2,y,goo[i]);settextstyle(TRIPLEX_FONT,
HORIZ_DIR,2);y+=80;}
        getch();cleardevice();gotoxy(1,1);closegraph();exit(1);}

void measure(void)
 {          union REGS registers;
        registers.x.ax=2;
        int86(0x33,&registers, &registers);
        cleardevice();
        double power;
        int i;char str[25],buffer[85];int x=getmaxx();int y=getmaxy();
        setcolor(WHITE);rectangle(0,0,x,y);
        setfillstyle(SOLID_FILL,LIGHTBLUE);
        floodfill(1,1,WHITE);

        setlinestyle(SOLID_LINE,0,3);
        setcolor(WHITE);rectangle(0,0,640,30);
        setfillstyle(SOLID_FILL,DARKGRAY);
        floodfill(2,2,WHITE);setcolor(WHITE);
        settextstyle(TRIPLEX_FONT, HORIZ_DIR,2);
        settextjustify(CENTER_TEXT,CENTER_TEXT);

        outtextxy(x/2,12,"MEASURING SCREEN");
        setcolor(BLACK);rectangle(120,80,520,300);
        setfillstyle(SOLID_FILL,LIGHTGRAY);
        floodfill(130,90,BLACK);setcolor(WHITE);
        line(120,300,120,80);line(120,80,518,80);

        setlinestyle(SOLID_LINE,0,3);
        rectangle(140,140,500,220);
        setfillstyle(SOLID_FILL,YELLOW);
        floodfill(142,150,WHITE);setcolor(BLACK);
        line(140,140,140,220);line(140,140,500,140);

        setcolor(BLACK);setlinestyle(SOLID_LINE,0,2);
        rectangle(180,420,399,450);
        setfillstyle(SOLID_FILL,DARKGRAY);
        floodfill(181,421,BLACK);
        setcolor(WHITE);line(180,420,399,420);
        line(180,420,180,449);setcolor(BLACK);
        outtextxy(300,430,"QUIT TO MAIN MENU");
        power=reader();
        gcvt(power,8, str);sprintf(buffer, "%s WATTS", str);
        int c,b,d;
        outtextxy(320,180,buffer);
        upme:show_mouse();interupt(c,d,b);
        if(c<=350 && c>=180 && d<=450 && d>=420 && b==1)choice();
        if(!kbhit())goto upme;
        }


void monitor(int we)
 {          int i,x=getmaxx(),y=getmaxy();char sta;
        double power;
        union REGS registers;
        registers.x.ax=2;
        int86(0x33,&registers, &registers);
        cleardevice();
        setlinestyle(SOLID_LINE,0,2);
        setcolor(WHITE);rectangle(0,0,x,y);
```

```
        setfillstyle(SOLID_FILL,LIGHTBLUE);
        floodfill(1,1,WHITE);
        setlinestyle(SOLID_LINE,0,2);
        setcolor(WHITE);rectangle(0,0,640,30);
        setfillstyle(SOLID_FILL,DARKGRAY);
        floodfill(2,2,WHITE);setcolor(WHITE);
        settextstyle(TRIPLEX_FONT, HORIZ_DIR,2);
        settextjustify(CENTER_TEXT,CENTER_TEXT);
        outtextxy(x/2,12,"MONITORING SCREEN");

        settextstyle(TRIPLEX_FONT, HORIZ_DIR,2);
        settextjustify(CENTER_TEXT,CENTER_TEXT);
        setcolor(BLACK);setlinestyle(SOLID_LINE,0,2);
        rectangle(50,250,200,300);
        setfillstyle(SOLID_FILL,LIGHTGRAY);
        floodfill(54,254,BLACK);
        setcolor(WHITE);line(50,250,199,250);
        line(50,250,50,299);setcolor(BLACK);
        outtextxy(125,275,"CONTINUE");

        rectangle(350,250,600,300);
        setfillstyle(SOLID_FILL,LIGHTGRAY);
        floodfill(354,254,BLACK);
        setcolor(WHITE);line(350,250,599,250);
        line(350,250,350,299);setcolor(BLACK);
        outtextxy(475,275,"QUIT TO MAIN MENU");
        setcolor(BLACK);setlinestyle(SOLID_LINE,0,2);
         if (we==1){getch();}
         int m=0,n=0,o=0;
        upmo: show_mouse();interupt(m,n,o);
        if(m<=200 && m>=50 && n<=300 && n>=250 && o==1)
        {power=reader();
          if(power+8.4<req)sta='L' ; else
          {if(power-9.2>req)sta='H' ;else sta='O';}alarm(sta);}
        if(!(m<=600 && m>=350 && n<=300 && n>=250 && o==1))goto upmo;
        return;}


void alarm(char status)
{       union REGS registers;
        registers.x.ax=2;
        int86(0x33,&registers, &registers);
cleardevice();int i;

switch(status){

case 'H':    setcolor(WHITE);rectangle(125,85,515,295);
             rectangle(127,87,513,293);
             settextstyle(TRIPLEX_FONT, HORIZ_DIR,5);
             setfillstyle(SOLID_FILL,RED);
             floodfill(129,89,WHITE);setcolor(BLUE);
             outtextxy(320,190,"TOO HIGH");
             setcolor(WHITE);rectangle(170,420,420,450);
             setfillstyle(SOLID_FILL,RED);
             floodfill(172,422,WHITE);setcolor(BLACK);
             line(171,421,419,421);line(171,421,171,449);
             line(171,422,419,422);line(172,421,172,449);
             line(171,423,419,423);line(173,421,173,449);
             settextstyle(TRIPLEX_FONT, HORIZ_DIR,1);setcolor(BLUE);
             outtextxy(300,430,"Press any key to continue");

             do{for(i=0;i<200;i+=5){
             sound(1400+i);delay(50);}
             }while(!kbhit());
             nosound();break;

case 'L':    setcolor(WHITE);rectangle(125,85,515,295);
             rectangle(127,87,513,293);
             settextstyle(TRIPLEX_FONT, HORIZ_DIR,5);
             setfillstyle(SOLID_FILL,BLUE);
             floodfill(130,90,WHITE);setcolor(RED);
             outtextxy(320,190,"TOO LOW");
```

```
            setcolor(WHITE);rectangle(170,420,420,450);
            setfillstyle(SOLID_FILL,BLUE);
            floodfill(172,422,WHITE);
            setcolor(BLACK);
            line(171,421,419,421);line(171,421,171,449);
            line(171,422,419,422);line(172,421,172,449);
            line(171,423,419,423);line(173,421,173,449);
            settextstyle(TRIPLEX_FONT, HORIZ_DIR,1);setcolor(RED);
            outtextxy(300,430,"Press any key to continue");

            do{for(i=0;i<200;i+=5){
            sound(1400+i);delay(50);}
            }while(!kbhit());
            nosound();break;

default :   setcolor(WHITE);rectangle(125,85,515,295);
            rectangle(127,87,513,293);
            settextstyle(TRIPLEX_FONT, HORIZ_DIR,5);
            setfillstyle(SOLID_FILL,LIGHTBLUE);
            setcolor(WHITE);floodfill(129,89,WHITE);
            outtextxy(320,190,"STATUS O K");
            setcolor(WHITE);rectangle(170,420,420,450);
            setfillstyle(SOLID_FILL,LIGHTBLUE);
            floodfill(172,422,WHITE);
            setcolor(BLACK);
            line(171,421,419,421);line(171,421,171,449);
            line(171,422,419,422);line(172,421,172,449);
            line(171,423,419,423);line(173,421,173,449);
            settextstyle(TRIPLEX_FONT, HORIZ_DIR,1);setcolor(WHITE);
            outtextxy(300,430,"Press any key to continue");
            do{fflush(stdout);}while(!kbhit());break;}int q=1;monitor(q) ;}


double reader(void)
{       int i;
      float v[100];double vmax,vrms,p;
      for(i=0;i<100;i++){
      outport(0x037a,0xff);delay(5);
      outport(0x037a,0xf0);//delay(10);
      v[i] = inport(0x0378) & 255;}
      for(i=0;i<100;i++){if (vmax<v[i]) vmax=v[i];}
        vmax=vmax *5.0/255;vmax*=57;
        vrms=vmax/1.4142;
        p=vrms*vrms/1000;  return p;}
```

2.5 THE POWER SUPPLY

The circuit of the project requires 5-volt dc voltage supply from the mains electricity supply of 240 volts AC. This is achieved by going through the following stages.

- Stepping down with a transformer
- Full wave rectification
- Smoothing or ripple removal circuit
- Voltage regulation
- Power display led

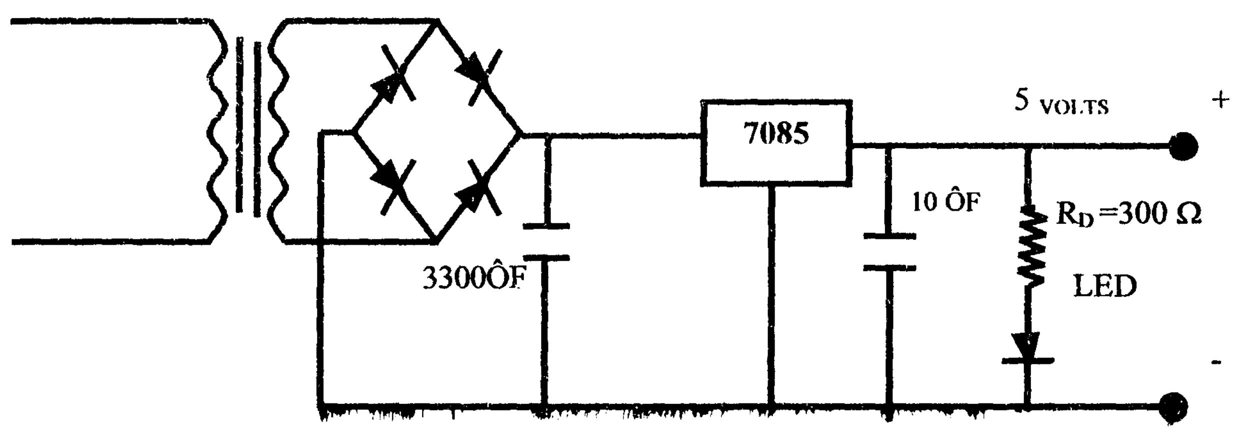


FIG 2.7 THE POWER SUPPLY CIRCUIT

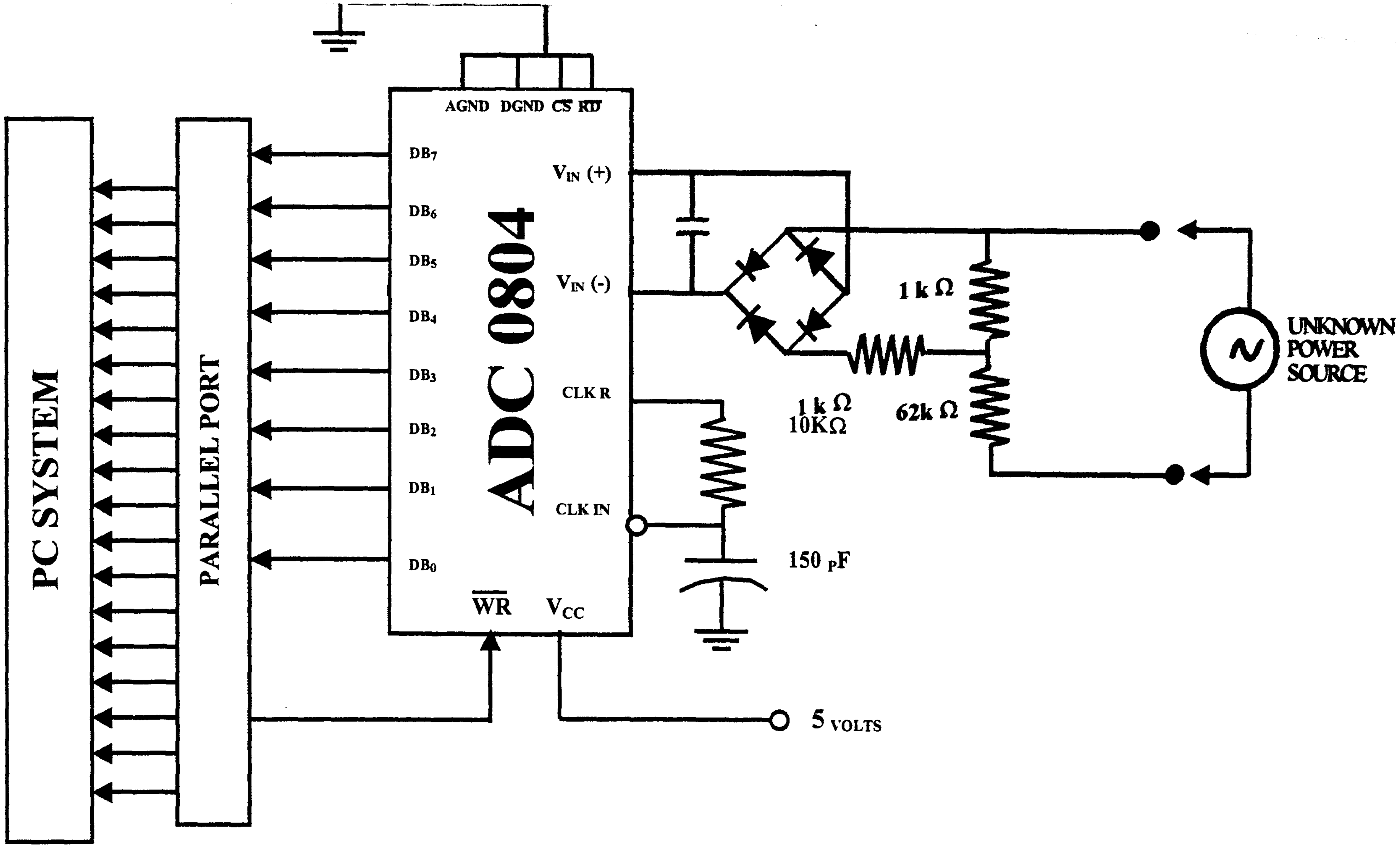


FIG 2.7 CIRCUIT DIAGRAM OF

THE COMPUTER CONTROLLED POWER MONITORING DEVICE (CCPMD)

CHAPTER THREE

CONSTRUCTION PROCEDURE

On completion of the design of this project, component, needed for its construction were bought and assembled together.

The construction started with the implementation of the design on a breadboard. This was done in modules proceeding from one module to another after due testing. The modules are as follows:

The power supply circuit

The Analogue to digital conversion circuit

The voltage conditioning circuit

Various aspects of the design were then modified to obtain a desirable working circuit before it was constructed permanently on a Vero board.

i. The Vero board layout was planned taking into consideration of unnecessary distance to avoid use of excess wire an adequate distance to avoid short circuits.

ii. IC socket was place in the place where the ADC was planned to be

iii. Components like resisters, capacitors and diodes were fixed and soldered directly onto the board

iv. A digital multimeter was then used to check the contact and continuity where necessary.

v. The ADC was inserted into its socket and then tested.

On the other hand, the software was written, edited, and debugged using a Borland C++ compiler.

When both software and hardware were ready, testing, debugging was done separately, and finally they were interfaced and tested.

3.1 TOOLS USED

With the use of instrument such as soldering iron, soldering lead, solder sucker, Vero board, connecting wire, Breadboard, wire, cutter and screw driver, digital multimeter and long nose pliers, the steps already mentioned above were taken to construct the ccpmmd.

3.2 MODE OF OPERATION

STEP 1 Connect the CCPMMD to the computer via the parallel port

STEP 2 Connect the CCPMMD to its power source and switch on

STEP 3 Load the CCPMMD software on the screen, which will present a welcome screen. Shown in Fig 3.1

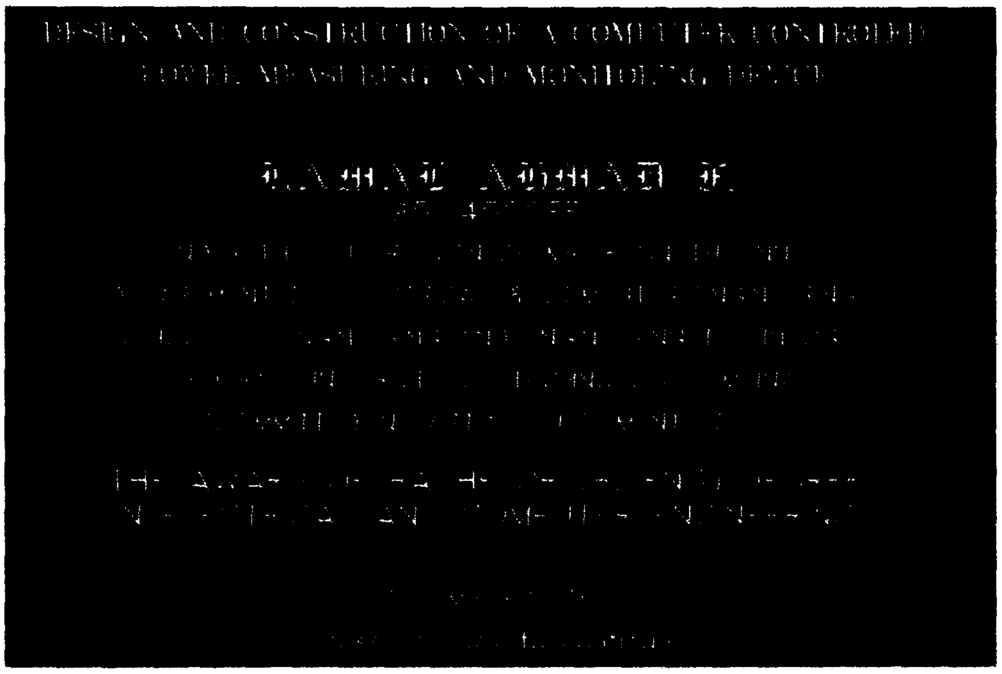
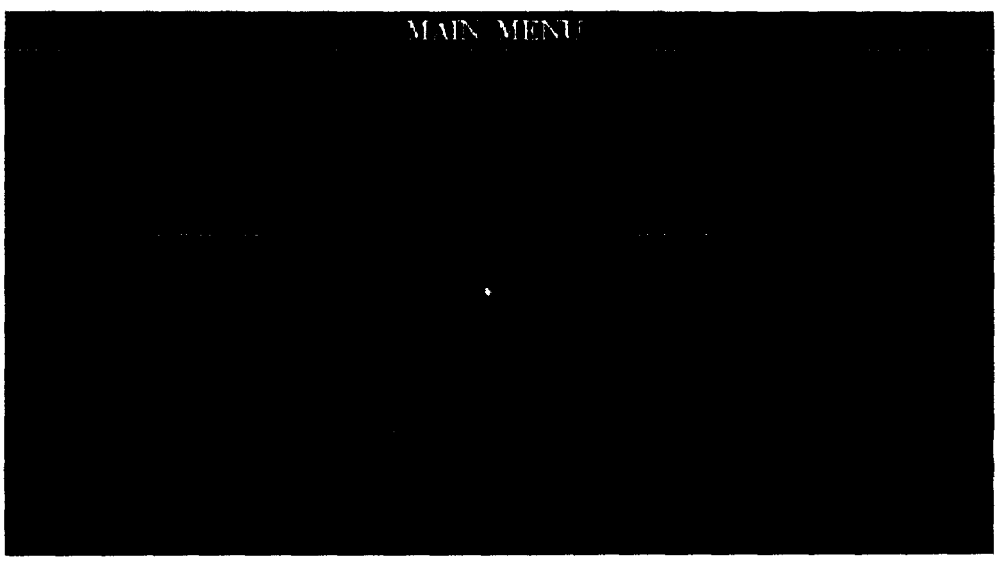


FIG 3.1 WELCOME SCREEN

And then the main menu shown in Fig 3.2



STEP 4.1 To measure, click on the measure button, which will then yield the measuring screen Shown in Fig 3.3

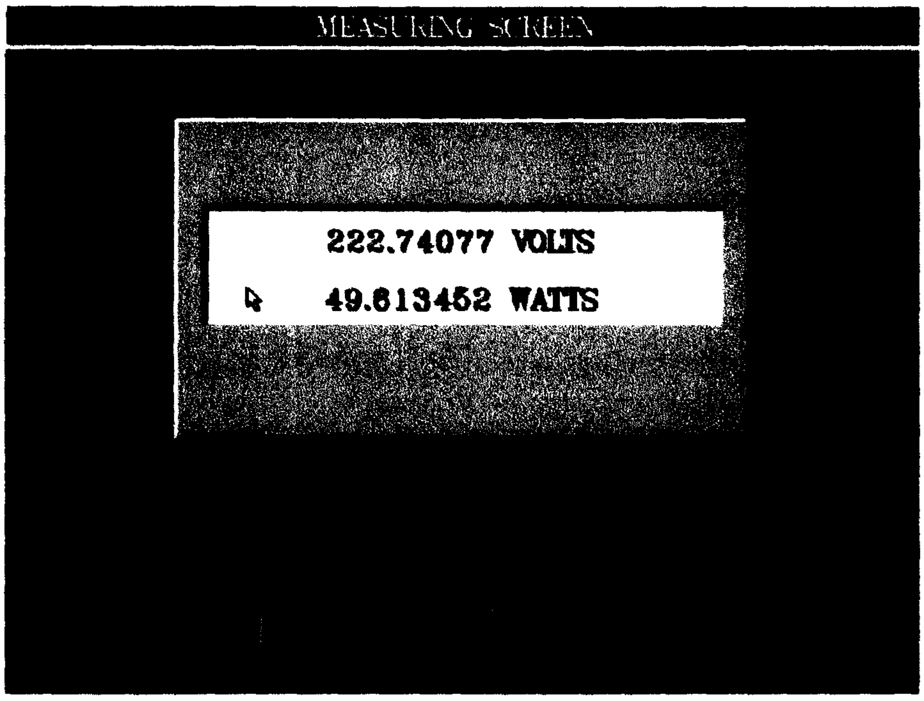
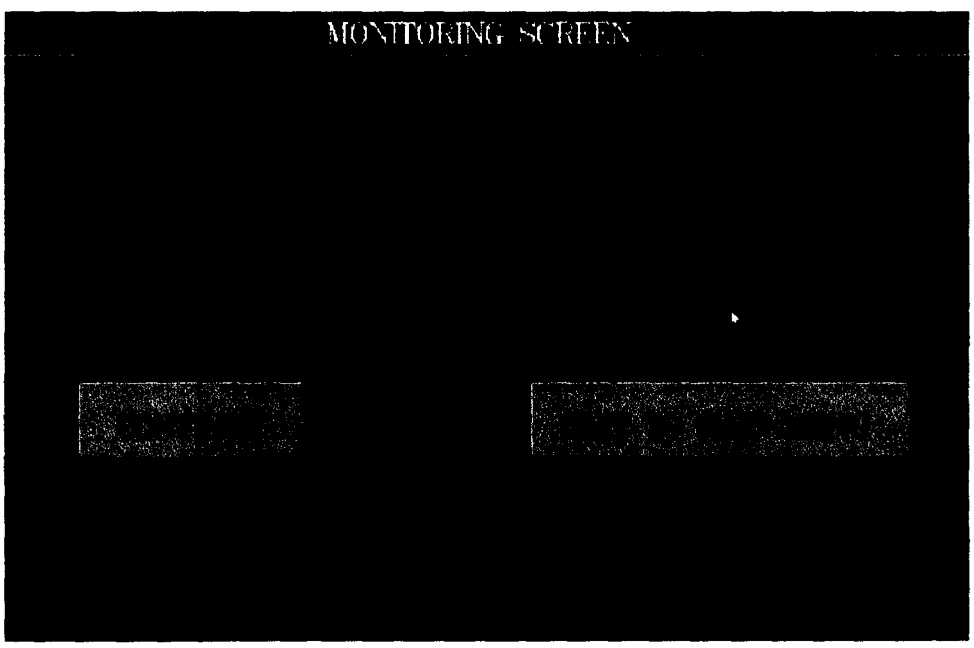


FIG 3.2 MEASURING SCREEN

STEP 4.2 To monitor, click on the monitor button, which will then yield the monitoring screen Shown in Fig 3.4



STEP 4.2.1 Click continue

If voltage falls in the range 200 - 240 the program will show Fig 3.5

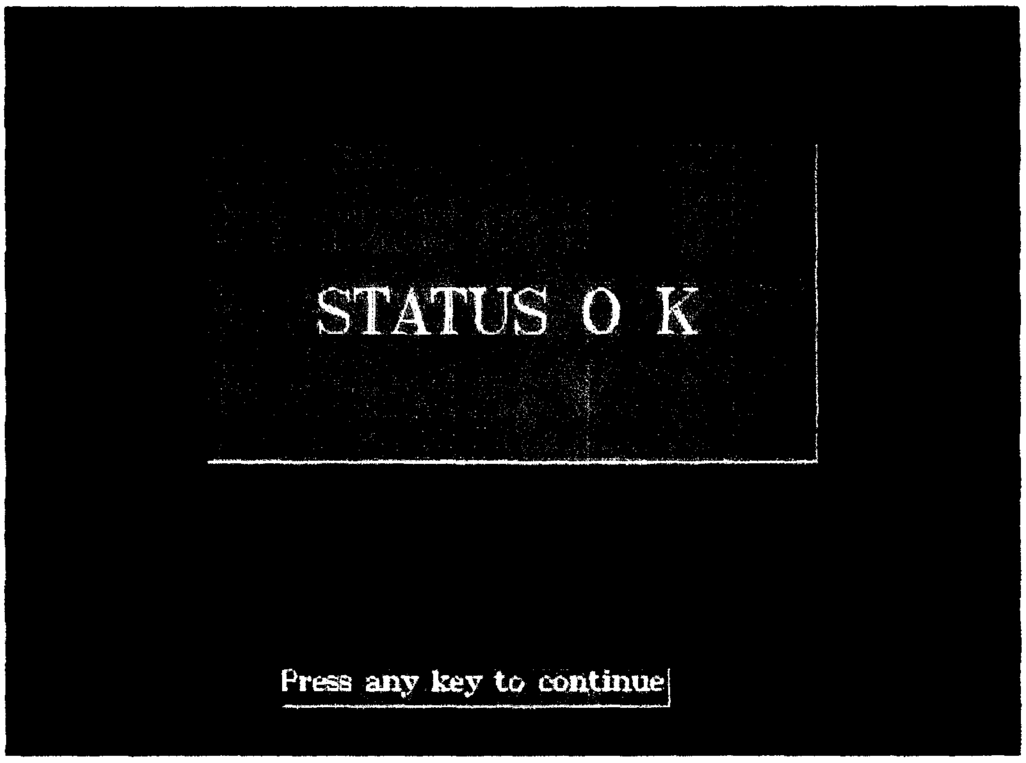
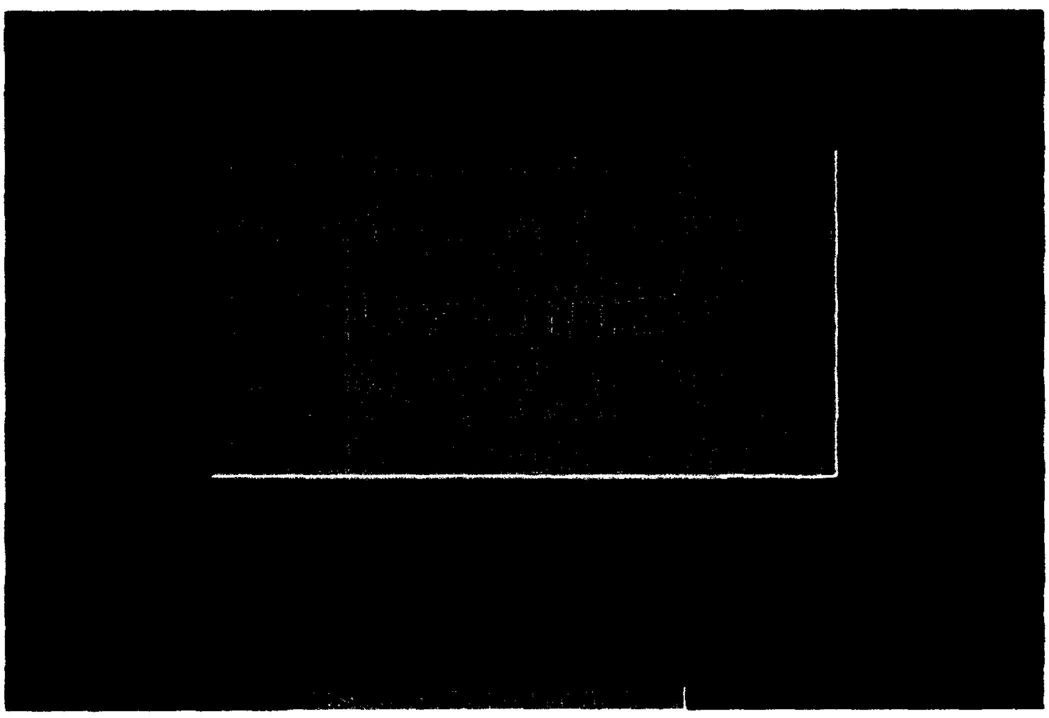


FIG 3.5 STATUS OK

If the voltage is higher than the above range the program will show Fig 3.6 with an alarm



If the voltage is lower than the specified range the program will show Fig 3.7 with an alarm

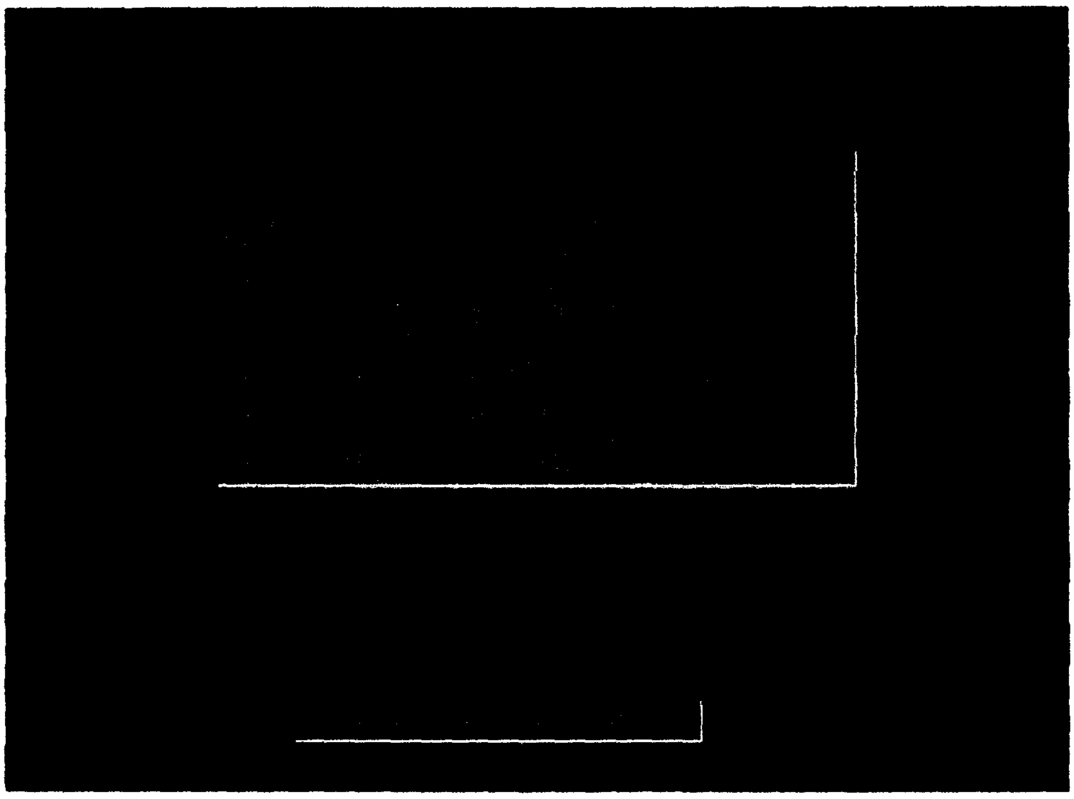
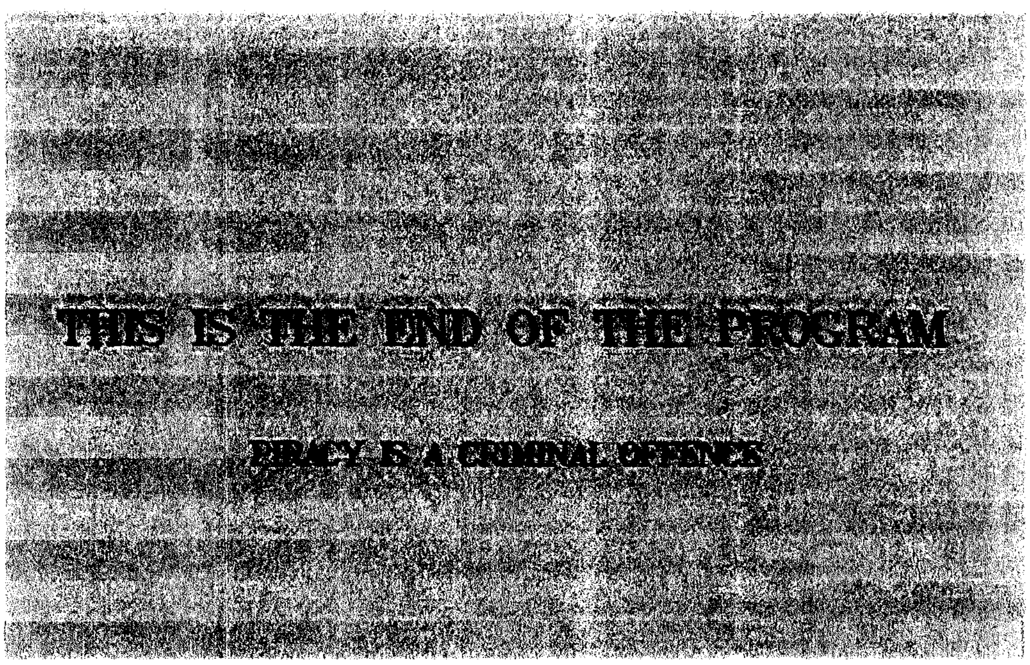


FIG 3.7 STATUS LOW

STEP 5 Click quit to exit program on return to the main menu shown in Fig 3.2 and then click quit to show Fig 3.8



3.3 TESTING

The testing of the hardware started from the construction stage as each of the modules on completion was tested before moving on to the next one. The circuit was in each case checked for continuity using a digital multimeter also the required characteristics of each component was tested before and after soldering.

Finally, after interfacing on the PC. The resultant measuring device was then tested. Measuring the voltage output of a variable power supply unit tested the voltage measurement.

3.4 RESULT

The measure part of the program gave the following results

ACTUAL VOLTAGE (VOLTS)	ACTUAL POWER DISCHARGED ACROSS 1K RESISTOR (WATTS)	MEASURED VOLTAGE (VOLTS)	MEASURED POWER DISCHARGED ACROSS 1K RESISTOR (WATTS)
210	44.1	208	43.27
220	48.8	223	50.watts

The monitor part of the program gave the following results

VOLTAGE (VOLTS)	POWER (WATTS)	STATUS
200	40.0	LOW
210	44.1	OK
220	48.8	OK
230	52.9	OK
240	57.6	HIGH

3.5 DISSCUSSION OF RESULTS

The actual values (expected readings) of voltage were compared with the actual reading on the multimeter after being tested, and were found to have an accuracy of ± 5volts

The monitoring section gave a high alarm when the voltage was greater than or equal to 240 volts. And a low alarm when was less than or equal to 200 volts.

3.6 LIMITATIONS

The CCPMMD can only measure or monitor voltage values between 100 volts to 300 volts AC voltage.

CHAPTER FOUR

CONCLUSION AND RECOMMENDATIONS

The development of a computer controlled power-monitoring device via the parallel port interface was presented in the project.

The aim of the project was achieved after many technical difficulties like;

- Some components were damaged and were replaced several times.
- No available device to test digital components

4.1 RECOMMENDATION

(i) This project could be improved by developing a control component to this device such that the device could control the power it has measured or monitored.

(ii) The computer controlled power monitoring devices could be improved to be a monitoring and controlling device as this could help in the power generating and transmitting sectors of the Nigerian Economy

(iii) Design and construction of electronics projects should be a regular phenomena to all student of the electrical and computer engineering department right from the intermediate levels so as to provide practical experience and to familiarize themselves with electronic components.

(iv) The concerned student under strict supervision of their supervisors should execute the final year project in the departmental laboratory.

4.2 REFERENCE

1 Bramer, B. and Bramer, S. (1995) "C++ for engineers",

Great Britain: Arnold publishers.

Pages 16, 20, 21, 22, 23, 24

2 Tokheim, Roger. (1998) "Digital Electronics",

Singapore: McGraw – Hill Book Company.

Pages 10, 11, 14, 15.

3 Holland, Robin (1995), "Microcomputer fault finding and design",

Greeat Britain: Macmillian press LTD.

Pages 5, 6.

4 Kaulner, Barry (1992), "PC architecture and assembly Language"

United Kingdom: Sigma press,

Pages 6, 7, 8.