CHAPTER - IV

SOFTWARE DETAILS

CHAPTER - FOUR

SOFTWARE DETAILS

4.Ø INTRODUCTION

The necessary software in the present work is developed using higher level language, Turbo C. The developed program is used to implement the hardware as a multichannel DAS. The program have two main functions which are , i) Get value (get_an_int()) and ii) Device control (card_flot (int a)).

i) Get value function : In this function the program takes the values of various informations from the keyboard. The informations are : number of readings, sampling rate per hour, required parameter (application) number and it display the required time for the acquisition.

ii) Device control function : This function interacts the various devices such as 8255, ADC 71Ø9 and multiplexer 4Ø51 of a hardware. The device control functions are initialization of 8255, channel selection using multiplexer (MUX 4Ø51) and control of ADC for checking the status in a loop.

4.1 FEATURES OF THE SOFTWARE

The software developed have many features which are discussed below.

a) IBM PC compatible, userfriendly software for data acquisition and on line display in graphical form.

b) The user can select up to 12 analog input channels in sequence or random.

c) Real time display of data in graphical form.

d) Data acquisition system starts after key stroking.

e) Real time streaming of data to data file.

f) The user can select the gain of the analog input channel.

g) The software can check the over range and the polarity of the input analog signal.

h) The run time of the data acquisition system is variable, the user can set variable run time duration through the options.

i) The software is set-up with self explanatory menu. The user can enter the required parameter (physical function) number, the number of readings to be taken. The software displays the time required for the acquisition, sampling rate.

j) Program is made available in the form of executable file.

This program collects the data from the user selected analog channel, store the data in the data file . All channels start and stop at the same time and sampled at one rate.

4.2 SYSTEM REQUIREMENT

I) IBM PC/XT/AT compatible computer with minimum 360 KB memory.

II) Minimum one floppy drive.

III) MS-DOS / PC-DOS version 2.0 or more.

4.3 SYSTEM OPERATION SPECIFICATIONS

The software is used for the following four cases :

case 1 - Temperature measurement :

case 2 - Find out the electrical field for conductivity measurement:

case 3 - Thermoelectric power measurement:

case 4 - Diode characteristics measurement:

The details of operations are as follows :

case 1 - Temperature sensor LM 335 along with its chord is connected to channel no. zero (ch-Ø) and hit the key for data acquisition. Pin configurations for channel Ø and temperature sensors as explained earlier.

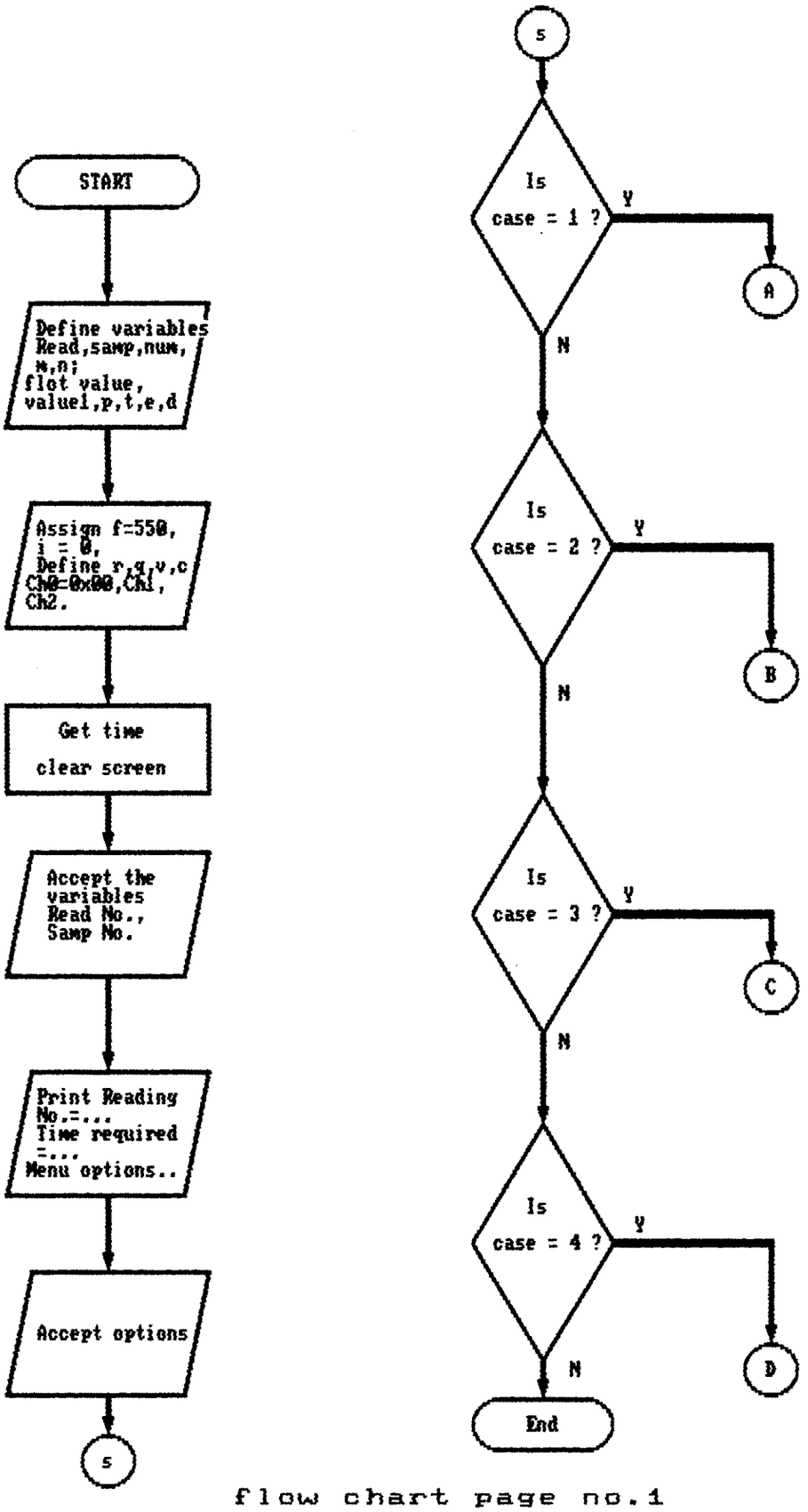
case 2 - Finding the electrical field for conductivity measurement, channel zero (ch-Ø) is used for the applied electrical field measurement and channel one (ch-1) is used for current density measurement. Connect as per mentioned above and hit the key at option.

case 3 - For the application of thermoelectric power measurement channel zero (ch-Ø) is used for furnace temperature measurement, channel one (ch-1) is used for the thermo emf measurement and channel two (ch-2) is used for the measurement of difference temperature.

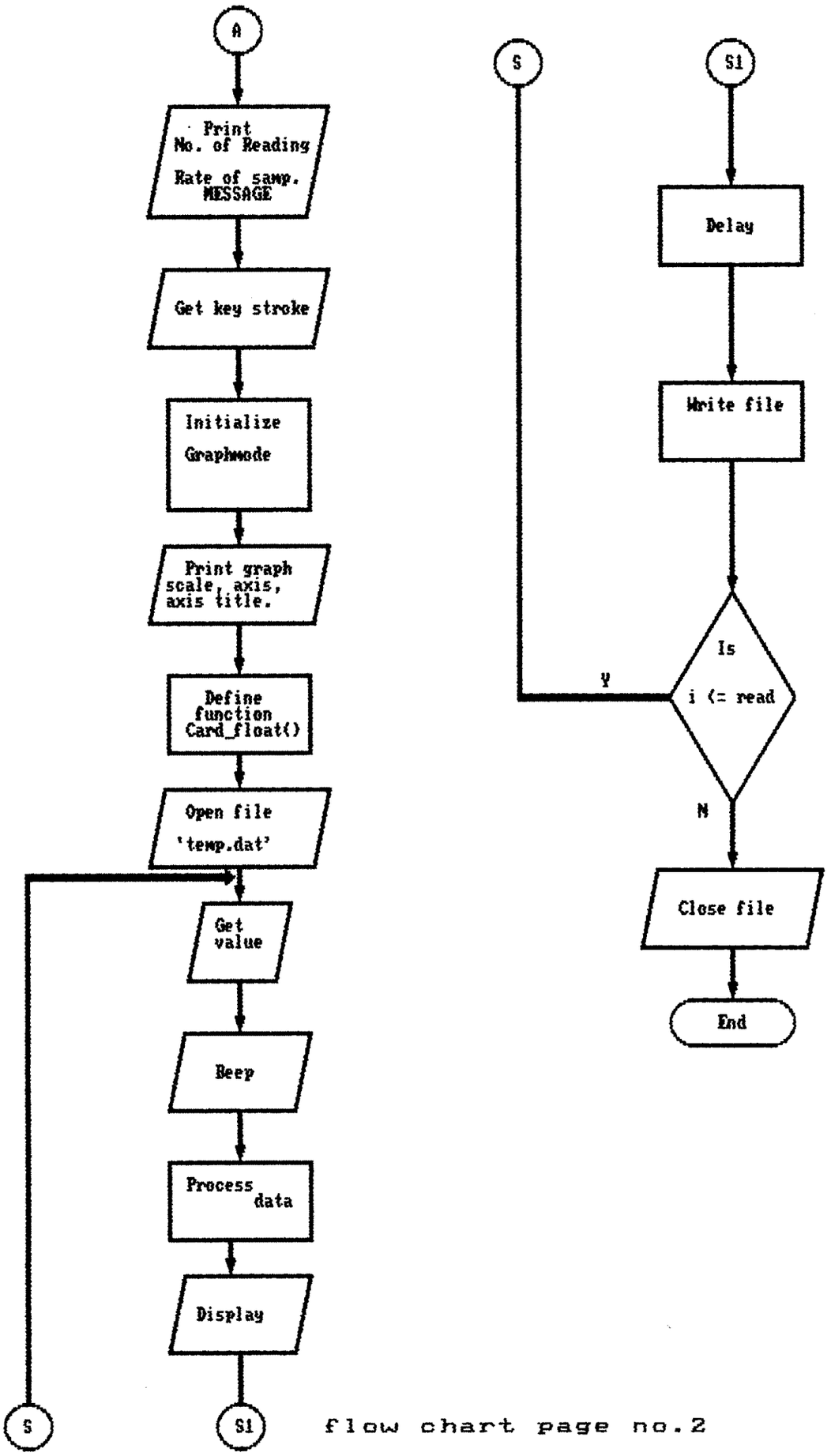
case 4 - In diode characteristics measurement application channel zero (ch-Ø) is used for the voltage measurement and channel two (ch-1) is used for the current measurement.

4.4 FLOWCHARTING AND LISTING OF THE PROGRAM

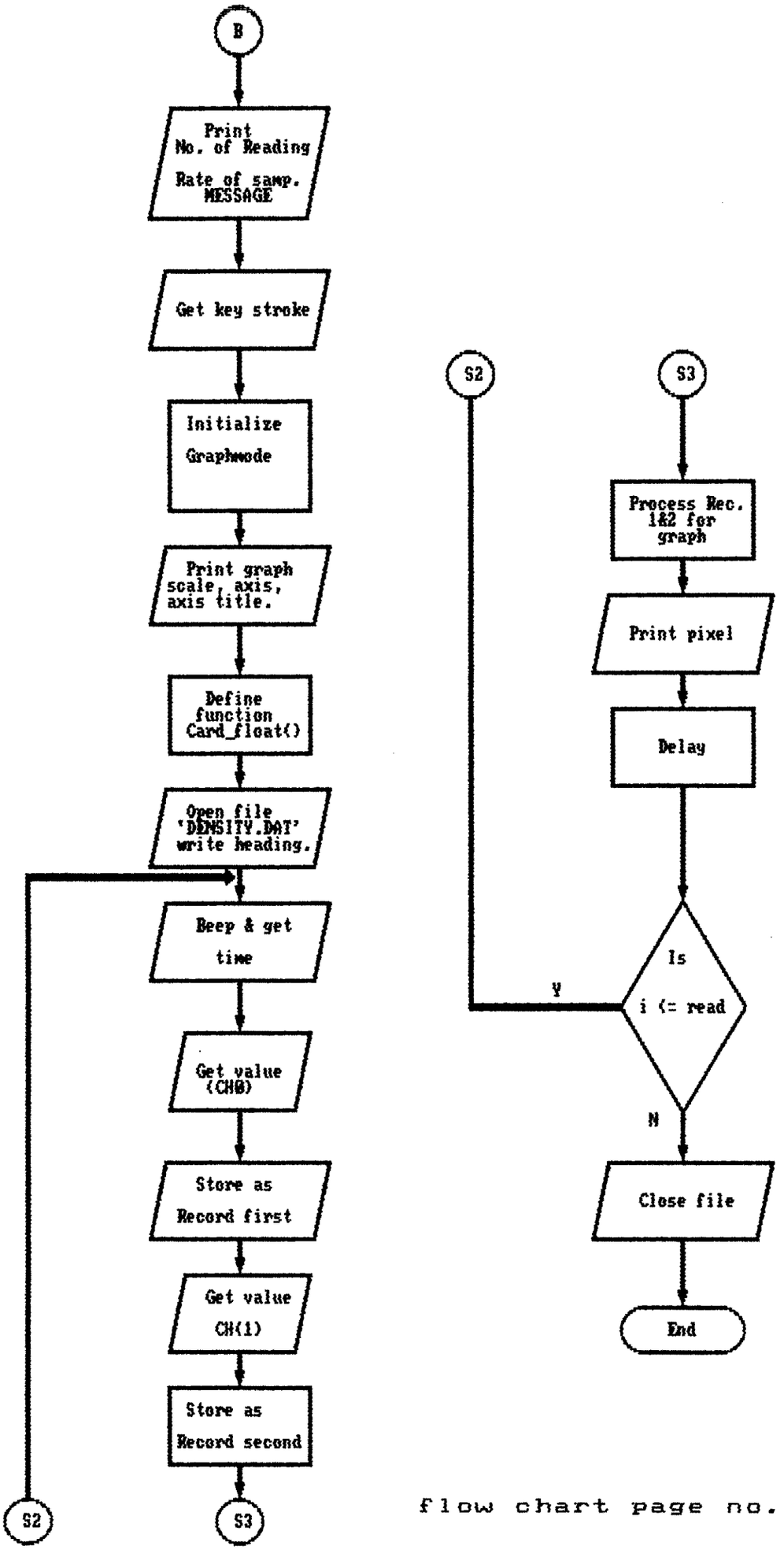
The flowchart and software listing of the program for this system is given below.



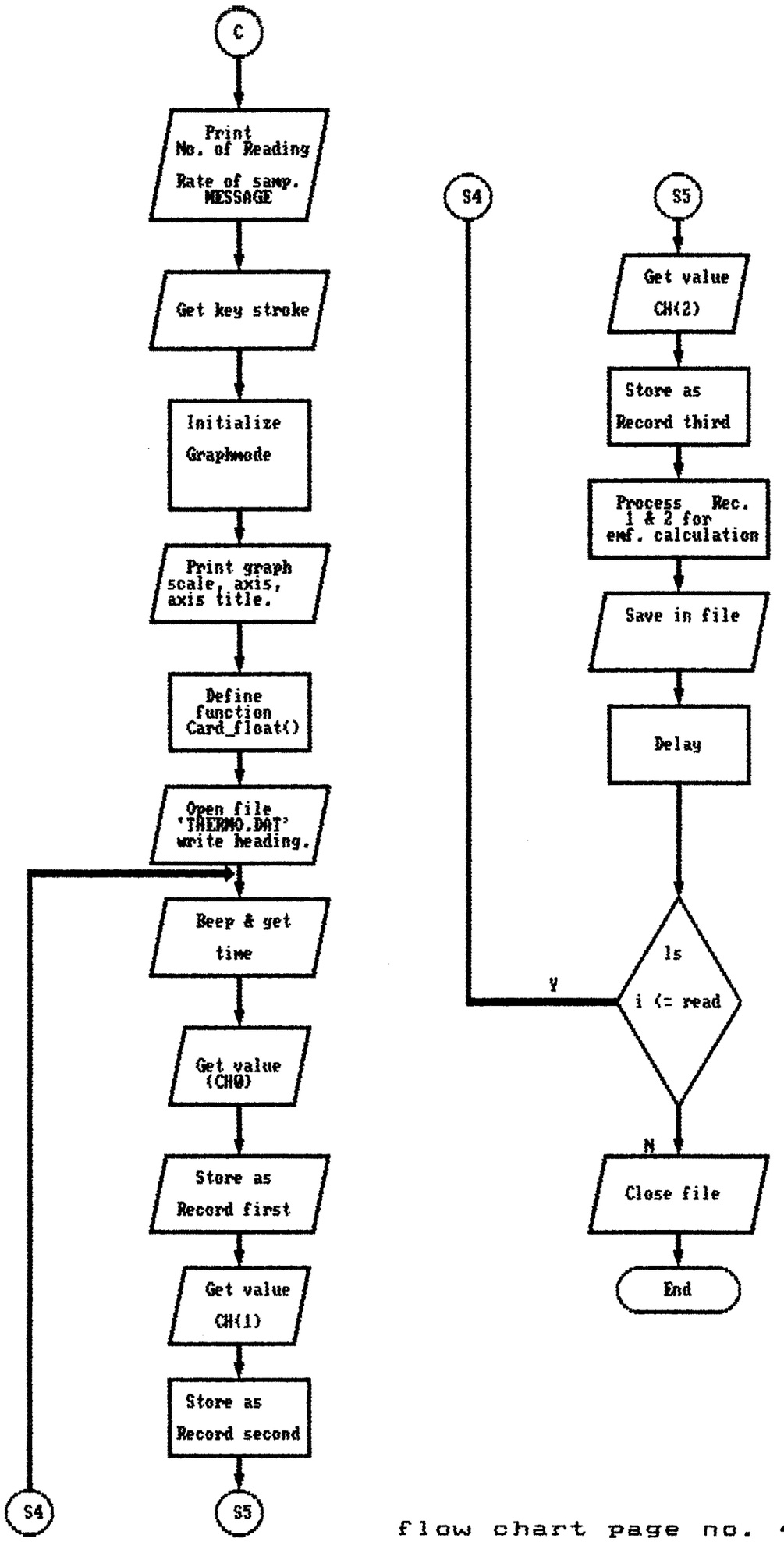
flow chart page no.1



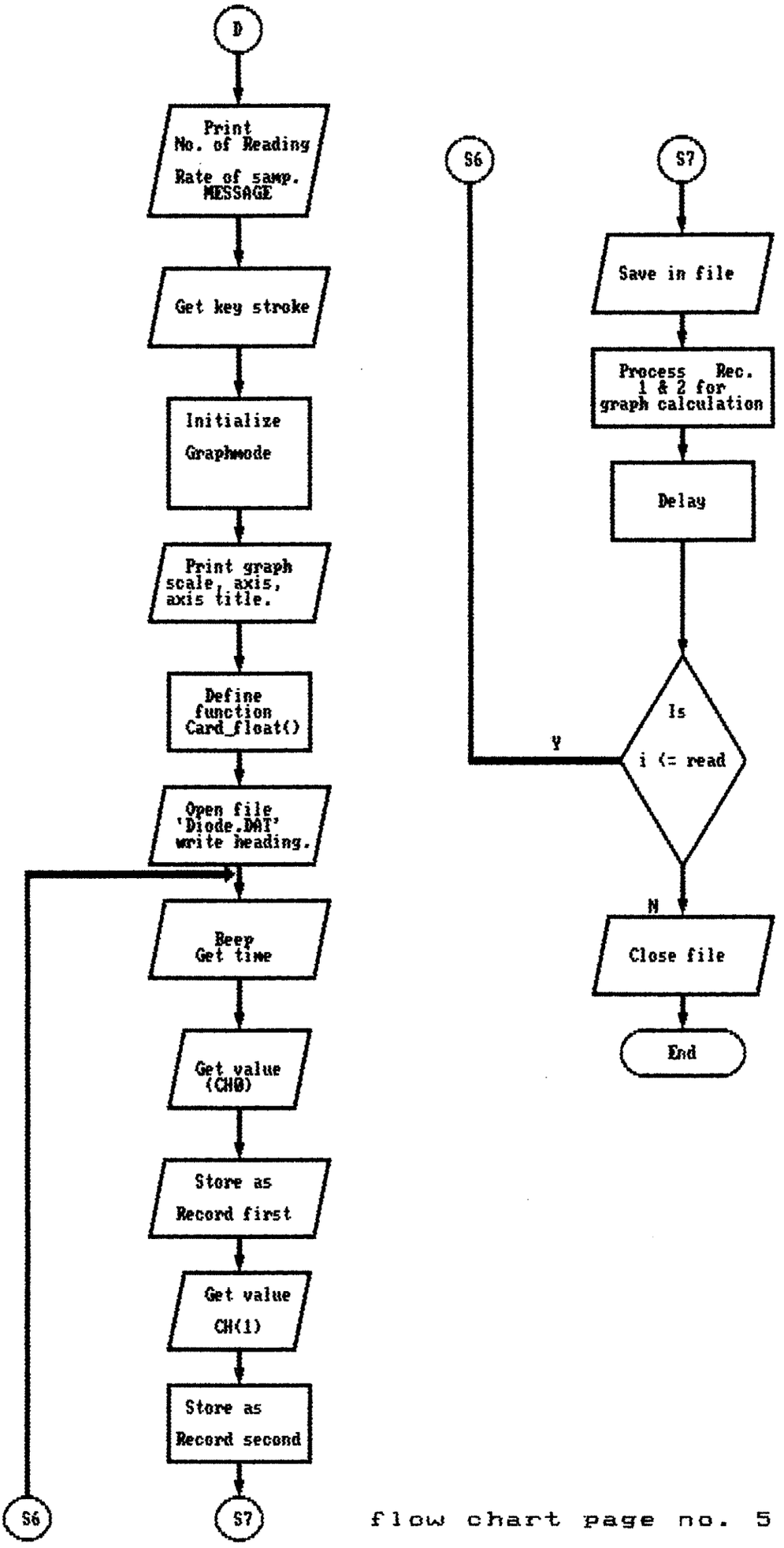
flow chart page no.2



flow chart page no. 3

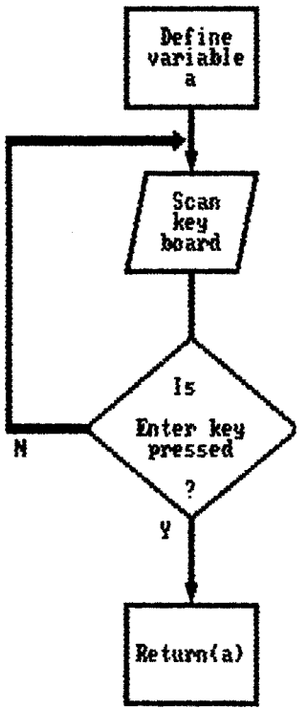


flow chart page no. 4

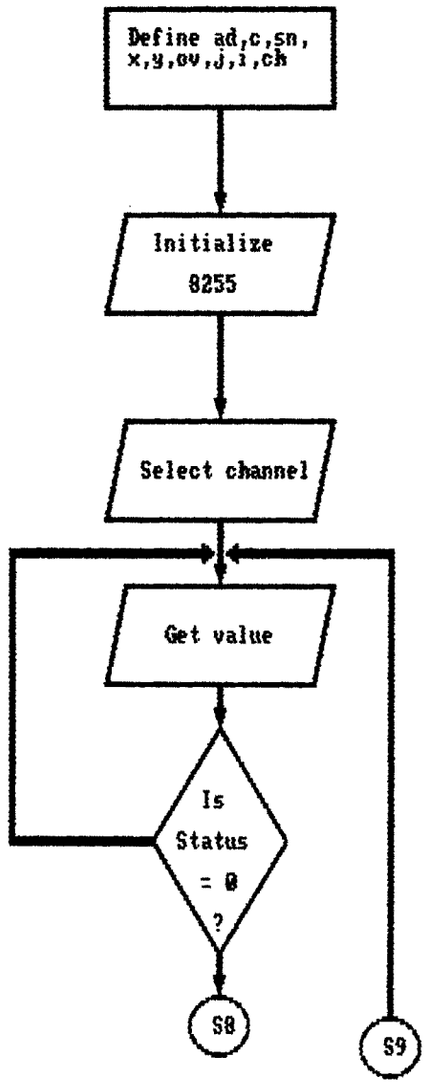


flow chart page no. 5

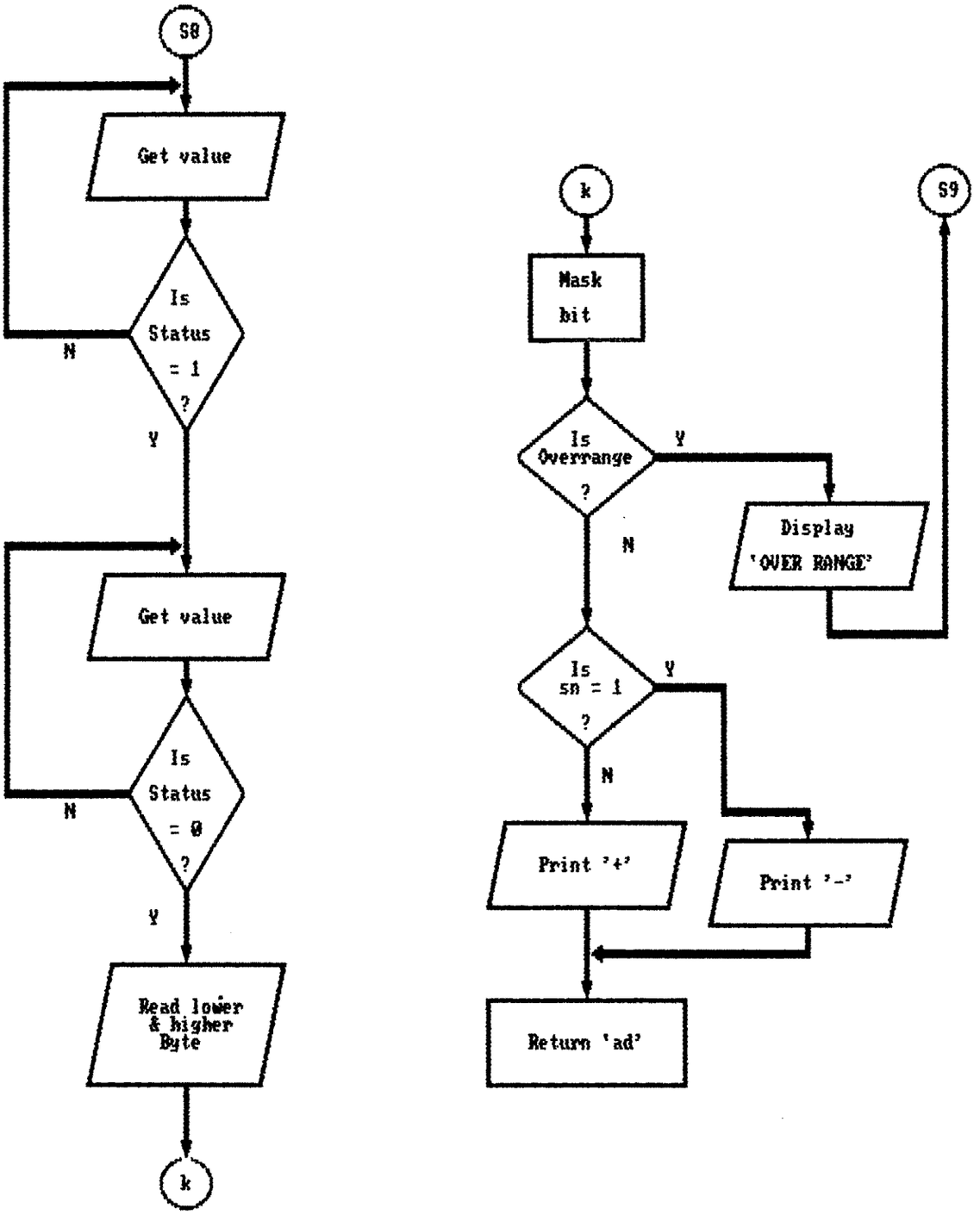
get_an_int()



card_float()



flow chart page no. 6



Flow chart page no.7

```
/* SOFTWARE FOR DATA ACQUISITION SYSTEM USING COMPUTER */

/* PROGRAMER : MR. BHARAT T. JADHAV */

/* DEPARTMENT OF PHYSICS/ELECTRONICS,*/

/* SHIVAJI UNIVERSITY, KOLHAPUR. */

/* GUIDE : Dr. S. H. CHAVAN */



#include <io.h>
#include <stdlib.h>
#include <process.h>
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
#include <dos.h>
#include <time.h>
main()
{
FILE *fp;

int read=0,samp=0,num,m,n;
float value,value1,p,t,e,d;
int f=550;
int i=0;
int r,q,v,c;
int ch0 = 0x00;
int ch1 = (ch0+1)*16;
int ch2 = (ch1+1)*16;

void  gotoxy (int x, int y);
int graphdriver = DETECT, graphmode;
time_t start,end;
clrscr();

printf("Enter the No. of Reading :");
read = get_an_int();
printf("Enter the Sampling Rate per HOUR:");
samp = get_an_int();
printf("\nThe  total  time  required :  %.2d" , read  *  (3600  /
samp));
printf("  Seconds.\n\n\n");
```

```
printf("Enter the required parameter no.:\n\n\n");


printf("1: TEMPERATURE MEARSUREMENT:\n");
printf("2: THE ELECTRICAL FIELD FOR CODUCTIVITY MEASUREMENT:\n");
printf("3: THERMOELECTRIC POWER MEARSUREMENT:\n");
printf("4: DIODE CHARACTRISTICS MEARSUREMENT:\n");
printf("5: EXIT:\n");

num = get_an_int();
printf("Entered value is: %d\n",num);

switch(num)

{
        case 1: {
                printf("No.:%d.\n",read);
                printf("Rate :%d.\n",samp);
           printf("Press any key.\n");
           printf("To start the sampling.\n");
           printf("Also see the progress.\n");
           getch();

           initgraph(&graphdriver , &graphmode, " " );
           rectangle(100,50,600,300);
           outtextxy(250,330,"T I M E ");
           outtextxy(20,100,"T");
           outtextxy(20,100,"T");
           outtextxy(20,110,"E");
           outtextxy(20,120,"M");
           outtextxy(20,130,"P");
           outtextxy(20,160,"Oc");
           outtextxy(70,300,"0");
           outtextxy(70,240,"20");
           outtextxy(70,180,"40");
           outtextxy(70,120,"60");
           outtextxy(70,60,"80");
           outtextxy(100,305,"0");
           outtextxy(200,305,"700");
           outtextxy(300,305,"1400");
           outtextxy(400,305,"2100");
           outtextxy(500,305,"2800");
           outtextxy(600,305,"3500");

           value = card_float(ch0);
           fp=fopen("temp.dat","w");

           if(fp==NULL)
             {
                    printf("error in opening file");
             }
            q = 0;
```

```
        for (i=0; i <= read; i++)

          {
          start = time(NULL);

          outtextxy(100,100,"NORMAL");
          value = card_float();
         sound(f);
         delay(1000);
         nosound();
               p = value * (32.5/450);/ * multiply by  the slope


               fprintf(fp," %4d  %4d   %4f\n ",q,value,p);
          m = (q/7+100);
          n = (300-p/3);
          putpixel(m,n,15);
          q = (q + (3600/samp));
                do
                {
                end = time(NULL);
                }
                while(difftime(end,start) != 3600/samp);
         }

}
        fclose(fp);
        if(fp==NULL)
          {
                 printf("error in opening file");
          }

        break;
case 2:  {
        printf("No.:%d.\n",read);
        printf("Rate :%d.\n",samp);
        printf("Press any key.\n");
        printf("To start the sampling.\n");
        printf("Also see the progress.\n");
        getch();


        initgraph(&graphdriver , &graphmode, " " );
        rectangle(100,50,600,300);
        outtextxy(250,330,"APPLIED FIELD IN V/CM");
        outtextxy(20,100,"C");
        outtextxy(20,110,"U");
        outtextxy(20,120,"R");
        outtextxy(20,130,"R");
        outtextxy(20,140,"E");
```

```
outtextxy(20,150,"N");

outtextxy(20,160,"T");

outtextxy(20,170,"  ");
outtextxy(20,180,"D");
outtextxy(20,190,"E");
outtextxy(20,200,"N");
outtextxy(20,210,"S");
outtextxy(20,220,"I");
outtextxy(20,230,"T");
outtextxy(20,240,"Y");
outtextxy(20,270,"A/cm.sqr.");

outtextxy(70,300,"00");
outtextxy(70,250,"10");
outtextxy(70,200,"20");
outtextxy(70,150,"30");
outtextxy(70,100,"40");
outtextxy(70,50,"50");

outtextxy(100,305,"00");
outtextxy(200,305,"40");
outtextxy(300,305,"80");
outtextxy(400,305,"120");
outtextxy(500,305,"160");
outtextxy(600,305,"200");

value = card_float();
fp=fopen("DENSITY.DAT","w");

if(fp==NULL)
  {
        printf("error in opening file");
  }
 fprintf(fp,"Sr.No.\t  E.FIELD\t CURRENT DENSITY\t");
 fprintf(fp," \t  V/cm\t\t A/cm sqr\t\n");
      fprintf(fp,"-----------------------------

 --------------\n");


 r = q = 0;
for (i=1; i <= read; i++)

  {
  sound(f);
 delay(1000);
 nosound();
 start = time(NULL);
 value = card_float(ch0);

       r = value * 7.5;  /* multiply by a factor 7.5 REA
```

```
IN V */

                value = 0;
                value = card_float(ch1);
                     q = value / 10;  /* devide by a factor 10 READING

microAmp*/
                q = q / .785;
                fprintf(fp," %d\t %4d\t\t  %5d\t  \n ",i,r,q);
                 v = r * 25 + 100;
                 c = 300 - q * 5;
                 putpixel(v,c,15);
                       do
                       {
                       end = time(NULL);
                       }
                       while(difftime(end,start) != 3600/samp);
                }

   }
            fclose(fp);
            if(fp==NULL)
              {
                     printf("error in closing file");
              }
    break;
   case 3: {
            printf("No.:%d.\n",read);
            printf("Rate :%d.\n",samp);
            printf("Press any key.\n");
            printf("To start the sampling.\n");
                 printf("And it records the

            data in file : THERMO.DAT.\n");


            initgraph(&graphdriver , &graphmode, " " );
            rectangle(100,25,650,325);
            line(100,175,650,175);
            outtextxy(350,330,"T E M P.   ");
            outtextxy(20,100,"T");
            outtextxy(20,110,"H");
            outtextxy(20,120,"E");
            outtextxy(20,130,"R");
            outtextxy(20,140,"M");
            outtextxy(20,150,"O");
            outtextxy(20,160,"E ");
            outtextxy(20,170,"L");
```

```
                outtextxy(20,180,"E");
                outtextxy(20,190,"C");
                outtextxy(20,200,"T");
                outtextxy(20,210," ");
                outtextxy(20,220,"P");
                outtextxy(20,230,"O");




                outtextxy(20,240,"W");
                outtextxy(20,250,"E");
                outtextxy(20,260,"R");
                outtextxy(20,270," ");
                outtextxy(20,280,"mW");

                value = card_float();
                fp=fopen("THERMO.DAT","w");

                if(fp==NULL)
                  {
                        printf("error in opening file");
                  }
                     fprintf(fp,"FN.TEMP\t,E.M.F.\t,DIFF.T.\t,T.E.P.\t\

                 sound(f);
                 delay(1000);
                 nosound();

                 for (i=0; i <= read; i++)

                  {
                  start = time(NULL);

                  value = card_float(ch0);
                       t  = value * 25 + 25;  /* multiply by a factor 25
add 25 */

                  value = card_float(ch1);
                  e = value;              /* Read emf */

                  value = card_float(ch2);
                       d  = value * 25;       /* The diffence temp of  s
face */

                  p = e / d;             /* Ratio of emf & diff. temp */
                  fprintf(fp,"%4.2f, %3.4f, %3.4f, %2.6f\n", t,e,d,p);



                do
                        {
```

```
                end = time(NULL);
                }
                while(difftime(end,start) != 60);
        }

        fclose(fp);
        if(fp==NULL)
          {
                printf("error in closing file");
          }

}
getch();
break;
case 4: {
        printf("No.:%d.\n",read);
        printf("AT EACH BEEP INCREASE THE VOLTAGE\n");
        printf("PRESS ANY KEY TO PROCEED\n");
        initgraph(&graphdriver , &graphmode, " " );
        rectangle(100,50,600,300);
        outtextxy(250,330,"V O L T A G E  IN mV ");
        outtextxy(20,90,"C");
        outtextxy(20,100,"U");
        outtextxy(20,110,"R");
        outtextxy(20,120,"R");
        outtextxy(20,130,"E");
        outtextxy(20,140,"N");
        outtextxy(20,150,"T");
        outtextxy(70,300,"100");
        outtextxy(70,250,"200");
        outtextxy(70,200,"300");
        outtextxy(70,150,"400");
        outtextxy(70,100,"500");
        outtextxy(70,50,"600");

        outtextxy(100,305,"000");
        outtextxy(200,305,"200");
        outtextxy(300,305,"400");
        outtextxy(400,305,"600");
        outtextxy(500,305,"800");
        outtextxy(600,305,"1000");

        value = card_float();
        fp=fopen("DIODE.DAT","w");
```

```
            if(fp==NULL)
              {
                    printf("error in opening file");
                  }
                 fprintf(fp,"Sr.No.\t                          Voltage(mv

Current(micr.A)\t\n");
                 fprintf(fp,"--------------------------------
------\n");




             r = q = 0;
            for (i=1; i <= read; i++)

              {
              sound(f);
             delay(1000);
             nosound();
             start = time(NULL);
             value = card_float(ch0);
             r = value * 4;  /* multiply by a factor 4 */
             value = 0;
             value = card_float(ch1);
             q = value * 10;  /* multiply by a factor 10 */
             fprintf(fp," %d\t %4d\t\t  %5d\t  \n ",i,r,q);
              v = r + 100;
              c = 300 - q;
              putpixel(v,c,15);
                    do
                    {
                    end = time(NULL);
                    }
                    while(difftime(end,start) != 3600/samp);
             }

   }
                fclose(fp);
            if(fp==NULL)
              {
                    printf("error in closing file");
                  }
    break;

        case 5: break;
}

gotoxy(10,10);
clrscr();
closegraph();
}
```

```
/*------*/
get_an_int()
{
int a;
scanf("%4d",&a);
return(a);
}
/*------------*/
card_float(int a)



{
  float ad;
  int x,y,ov,j,i,ch;
  float c,sn;
  outp(Øx243,Øx92);       /* initialize 8255 */
  ch = a;
  outp(Øx242,ch);       /* Channel slection -Ø */

   do
   {
        x = inp(Øx241);
        y = (x & Øx8Ø);                /* status check Ø */
        }
   while(y == 128);

        do
           {
           x = inp(Øx241);           /* Status check 1 */
           y = x & Øx8Ø;
           }
   while(y == Ø);

           do
                   {
                   x = inp(Øx241);      /* status check Ø */
                   y = x & Øx8Ø;
                   }
           while(y == 128);
  x = inp(Øx24Ø);                        /* read lower byte */

  y = inp(Øx241);                       /* read higher byte */

    ov = (y & Øx1Ø);                   /* check over range */
    if (ov == 16)
   outtextxy(35Ø,25,"OVER RANGE\n");
    sn = (y & Øx2Ø);                   /* check for polarity */
    c = (y & Øxf);
    ad = ((x+256*c)/1Ø);
    if (sn == Øx2Ø)
   /* printf("+")*/
```

```
        ad = ad ;
        else
 /*       printf("-");*/
        ad = ad * (-1);
        return(ad);
}
```

REFERENCES :

1) Programming In Turbo C,

-Tim Grady.

2) Turbo C The Complete Reference,

-Herbert Schield.

3) Master In Turbo C,

-Stain Kennly-Bottle,

B.P.B. Publications.

4) Basic C For Basic Programmers,

-T. D. Brown Jr.