CHAPTER 3
SOFTWARE FOR DISPLAYING THE CHARACTERISTICS OF A DEVICE, CALCULATION OF PARAMETERS AND CIRCUIT DESIGN USING THE DEVICE
3.1 Software package development

The author has developed a software for the display of device characteristics, calculation of its parameters and circuit design using these devices. The software package in the form of a "MENU" is developed as a general purpose package to cover the devices viz. Transistor, FET, Diode, Zener diode, UJT, SCR, Diac and Triac as shown in Fig.3.1. However, the present project is confined to the study of the devices devices viz. Transistor, FET, MOSFET, Diode, Zener diode and UJT. The author is proposes to extend this work to cover the other devices viz. SCR, Diac and Triac in future.

This package is a menu similar to any standard menu, and it comprises of a large number of files. The main menu consists of six columns, the first column is devoted for drawing the characteristics of a transistor. The second, third, fourth and fifth columns are devoted for drawing the characteristics of FET, UJT, Zener Diode and ordinary Diode respectively. The sixth column is devoted for quitting from the program.

The options presented in the main menu are selected by positioning the cursor with the help of the horizontal arrow keys. Suppose if the characteristics of a transistor are to be displayed, the cursor is placed on the first column.
<table>
<thead>
<tr>
<th></th>
<th>Trans/UJT</th>
<th>FET</th>
<th>SCR/Triac</th>
<th>Diac</th>
<th>Diodes</th>
<th>Quit</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPN</td>
<td>IBmax. 10uA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PNP</td>
<td>IBmax. 100uA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UJT</td>
<td>IBmax. 1mA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IBmax. 10mA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IBmax. 100mA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Draw char.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig 3.1. DEVICE CHARACTERISTICS MENU
(i.e., Transistor) and then pressing the enter key will display another sub-menu. This sub-menu, has seven rows, the first three rows are meant for selecting the device type i.e., the Transistor(PNP or NPN). This is selected by placing the cursor on the appropriate row and pressing the enter key. After selecting the device, the cursor is next moved by manipulating the arrow keys and positioned on the required base current max. in case of a transistor and entered. Finally, after selecting the device and the base current max. the cursor is positioned on the "draw char" row and on entering -the characteristics of the device are displayed on the monitor.

The procedure is repeated for drawing the characteristics of the other devices also.
3.2. SOFTWARE

i. ALGORITHM:

The software for the computer based system for the proposed study is mainly divided into three parts viz. a. Displaying the characteristics of a device b. Calculation of the device parameters and c. Circuit design using the device parameters.

ii. Procedure adopted for displaying the characteristics of the device:

The first step of the program will initialize all the programmable peripheral interfaces (PPIs) for data transfer to-and-from the ports. This step also contains data conversion program for D/A and A/D conversions for applying (i) appropriate voltages/currents to the device under test; and (ii) for measuring the voltage across the device \( (V_1) \) and the current \( (I_1) \) through it respectively. After measuring the voltage across the device and the current flowing through it, the program draws a line from the origin to the point defined by the current values of \( (V_1, I_1) \). This point becomes the starting point for the next voltage and current point. This process is repeated for different supply voltages starting from zero to the maximum value fixed at 15V to get the characteristic curves of the device under test. This characteristic corresponds to a specific base current \( I_{B1} \).
The base current is incremented in suitable steps and this process is repeated for different base currents up to the $I_B$ max. allowed.

iii. Procedure adopted for calculation of the parameters of a device:

After drawing the device characteristics, the computer enters into the second step where different parameters of the device are calculated from the slope of the selected characteristic curves.

iv. Procedure adopted for the circuit design:

Finally, the computer enters into the third step viz. the circuit design program where it asks the user to provide the required specifications for the circuit design such as $V_{BE}$, the gain of the amplifier ($G$), lower cut-off frequency ($f_L$) and higher cut-off frequency ($f_H$) etc. in case of transistor amplifier. After entering these specifications, the component values of the desired circuit are evaluated and are displayed on the monitor.
3.3. Flow chart for display of characteristics, calculation of parameters and circuit design for different devices:

a. Transistor:

The Fig. 3.3 shows the flow chart of the P.C. Based system for displaying the transistor characteristics, calculation of parameters. For drawing the characteristics of a three terminal device like a Transistor, the P.C. based system first sets the base current \( I_B \) through one digital to analog (D/A) converter and the supply voltage \( V_{CC} \) is set by using another digital to analog (D/A) converter. The current flowing through the transistor and voltage across the device are measured through the analog to digital (A/D) converter. After measuring the voltage across the device and the current flowing through it, the program draws a line from the origin to the point defined by the present values of \( V_{CE1}, I_C1 \). This point becomes the starting point for the next voltage and current point \( (V_{CE2}, I_C2) \). This process is repeated for different supply voltages starting from zero to the maximum value fixed at 15V to get a characteristic curve of the device under test. This characteristic curve corresponds to a specific value of the base current \( I_{B1} \). The base current is incremented in suitable steps and this process is repeated for different base currents. This process provides a set of output characteristic curves of the device. Similar procedure is adopted for drawing the input characteristics.
Fig. 3.2 Flow chart for displaying the characteristics of a transistor calculation of parameters and circuit design.
(\(I_B\) versus \(V_{BE}\) keeping \(V_{CE}\) constant); the Forward transfer characteristics (\(I_B\) versus \(I_C\) keeping \(V_{CE}\) constant); the reverse transfer characteristics (\(V_{BE}\) Versus \(V_{CE}\) keeping \(I_B\) constant).

After drawing the characteristics of a transistor, the computer calculates the parameters of the device under test by measuring the slopes of the characteristic curves by employing the "arrays technique".

Finally, the computer evaluates the following equations 3.1 to 3.14 for designing a transistor amplifier circuit as per the specifications entered by the designer using these values of the parameters calculated from the characteristics. The results are then displayed on the computer monitor. Specific example of a transistor amplifier design is presented in chapter 4.

\[
I_C = \frac{V_{RC}}{R_C} \tag{3.1}
\]

\[
R_E = \frac{((h_{fe}^*(R1*R2)/(R1+R2)/A_v)-h_{ie})}{(1+h_{fe})} \tag{3.2}
\]

\[
V_C = V_{CC} - V_C \tag{3.3}
\]

\[
I_2 = \frac{I_C}{10} \tag{3.4}
\]

\[
I_B = \frac{I_C}{h_{fe}} \tag{3.5}
\]

\[
I_E = I_B + I_C \tag{3.6}
\]

\[
V_B = V_E + V_{BE} \tag{3.7}
\]

\[
R_2 = \frac{V_B}{I_2} \tag{3.8}
\]

\[
R_1 = \frac{(V_{CC} - V_B)/(I_B + I_2)} \tag{3.9}
\]
\[
Z_I = \frac{(R_1*R_2)}{(R_1+R_2)}*(h_{ie} + R_E(1+h_{fe})) \quad \text{(3.10)}
\]

\[
C_1 = \frac{1000}{(2nfL*0.1*Z_I)} \quad \text{(3.11)}
\]

\[
h_{ib} = \frac{h_{ie}}{(1+h_{fe})} \quad \text{(3.12)}
\]

\[
C_2 = \frac{1000}{(2nfL*(h_{ib}+R_L)} \quad \text{(3.13)}
\]

\[
C_3 = \frac{1000}{(2nfL*0.1*R_L)} \quad \text{(3.14)}
\]

b. Field Effect Transistor (FET):

The Fig.3.4 shows the flow chart of the P.C. Based system for displaying the transistor characteristics, Calculation of parameters. For drawing the characteristics of a FET the P.C. based system first sets the gate voltage \( V_G \) through one digital to analog (D/A) converter and the supply voltage is set by using another digital to analog (D/A) converter. The current flowing through the device \( I_D \) and voltage across the device \( V_D \) are measured through the analog to digital (A/D) converter. After measuring the voltage across the device and the current flowing through it, the program draws a line from the origin to the point defined by the current values of \((V_D1,I_D1)\). This point becomes the starting point for the next voltage and current point \((V_D2,I_D2)\). This process is repeated for different supply voltages starting from zero to the maximum value fixed at 15V to get a characteristic curve of the device under test. This
Fig. 3.3 Flow chart for displaying the characteristics of a FET suit design
Calculation of parameters and circuit design
characteristic corresponds to a specific gate voltage \( V_G \). The gate voltage is incremented in suitable steps and this process is repeated for different gate voltages. This process provides a set of output characteristic curves of the device. Similar procedure is adopted for drawing the transfer characteristics curves (\( V_{GS} \) versus \( I_D \)) for constant \( V_D \).

After drawing the characteristics of the device the computer calculates the parameters of the device under test by measuring the slope of the characteristics by employing the "arrays technique".

Finally, the computer evaluates the following equations 3.15 to 3.21 for designing an amplifier circuit as per the specifications entered by the designer using the values of the parameters calculated from their characteristics. The results are then displayed on the computer monitor. The specific example of a FET amplifier design is presented in chapter 4.

\[
\begin{align*}
Z_I &= R_g \\
Z_o &= R_D \\
R_S &= \frac{(R_D R_L)/(R_D + R_L)}{A_v} \\
A_v &= r_d g_m \\
C_c &= \frac{1}{2nf_1 R_g}
\end{align*}
\]
\[ C_s = \frac{1}{2nf_1^*R_L} \] (3.20)

c. Unijunction Transistor (UJT):

The Fig. 3.5 shows the flow chart of the P.C. Based system for Displaying the UJT characteristics, Calculation of Parameters. For drawing the characteristics of a UJT, the P.C. based system first sets the supply voltage \( V_{BB} \) through first digital to analog (D/A) converter and the emitter supply voltage \( V_E \) is set by using another digital to analog (D/A) converter. The current flowing through the emitter \( I_E \) and voltage across the device \( V_E \) are measured through the analog to digital (A/D) converter. After measuring the voltage across the device and the current flowing through it, the program draws a line from the origin to the point defined by the current values of \( V_{E1}, I_{E1} \) for a given \( V_{BB} \). This point becomes the starting point for the next voltage and current point \( (V_{E2}, I_{E2}) \). This process is repeated for different emitter voltages starting from zero to the maximum value fixed at 10V to get a characteristic curve of the device under test. This characteristic corresponds to a specific supply voltage \( V_{BB} \). The supply voltage is incremented in suitable steps and this process is repeated for different \( V_{BB} \)'s. This process provides a set of emitter characteristics of the UJT.

After the drawing characteristics of the device the computer calculates the parameter of the device under test by
Fig. 3.4 Flow chart for displaying the Characteristics of a UJT Calculation of Parameters and Circuit design
measuring the slope of the characteristics by employing the "arrays technique".

Finally, the computer evaluates the following equations 3.21 to 3.25 for designing a relaxation oscillator circuit as per specifications entered by the designer using the values of the parameters calculated from their characteristics. The results are then displayed on the computer monitor. A specific example of a relaxation oscillator design is presented in chapter 4.

$$R_{\text{max}} = \frac{(V_{BB} - V_p)}{I_p} \quad \text{(3.22)}$$

$$R_{\text{min}} = \frac{(V_{BB} - V_v)}{I_v} \quad \text{(3.23)}$$

$$R = (R_{\text{max}} \times R_{\text{min}})^{1/2} \quad \text{(3.24)}$$

$$C = \frac{3.2030*t}{\log((1/(1-\eta)))} \quad \text{(3.25)}$$

\text{d. Diode:}

The Fig. 3.6 shows the flow chart of the P.C. Based system for Displaying the diode characteristics, Calculation of Parameters. For drawing the characteristics of a two terminal device like an ordinary Diode the P.C. based system first sets the supply voltage by using digital to analog (D/A) converter. The current flowing through the device $I_A$
Fig. 3.5 Flow chart for displaying the Characteristics of a Diode Calculation of Parameters and Circuit design.
and voltage across the device $V_{AK}$ are measured through the analog to digital (A/D) converter. After measuring the voltage across the device and the current flowing through it, the program draws a line from the origin to the point defined by the present values of $(V_{AK1}, I_{A1})$. This point becomes the starting point for the next voltage and current point $(V_{AK2}, I_{A2})$. This process is repeated for different supply voltages starting from zero to the maximum value fixed at 15V to get a characteristic curve of the device under test.

After drawing the characteristic of the device the computer then calculates the forward and reverse resistances $R_F$ and $R_P$ of the device under test by measuring the slope of the characteristic by employing the "arrays technique".

Finally, the computer evaluates the equation (3.26) for designing rectifier using an ordinary diode. The results are then displayed on the computer monitor. Specific example of the design of a specific Half-Wave rectifier is presented in chapter 4.

$$E_p = V_p - (I_f R_f) \quad \quad (3.26)$$

e. Zener Diode:

The Fig.3.7 shows the flow chart of the P.C. Based system for Displaying the characteristics of Zener diode and, Calculation of Parameters. For drawing the characteristics of its a two terminal device like a Zener Diode the P.C. based
Fig. 3.6 Flow chart for displaying the Characteristics of a Zener Diode Calculation of Parameters and Circuit design
system first sets the supply voltage by using digital to analog (D/A) converter. The current flowing through the device $I_Z$ and voltage across the device $V_Z$ are measured through the analog to digital (A/D) converter. After measuring the voltage across the device and the current flowing through it, the program draws a line from the origin to the point defined by the present values of $(V_{Z1}, I_{Z1})$. This point becomes the starting point for the next voltage and current point $(V_{Z2}, I_{Z2})$. This process is repeated for different supply voltages starting from zero to the maximum value fixed at 15V to get a characteristic curve of the device under test.

After drawing the characteristic of the device the computer then calculates the forward and resistance $R_F$ and $R_R$ of the device under test by measuring the slope of the characteristic by employing the "arrays technique".

Finally, the computer evaluates the equations for designing a shunt voltage regulator using a zener diode. The results are then displayed on the computer monitor. Specific example of the design of a shunt regulator is presented in chapter 4.

\[ I_L = \frac{R_L}{V_Z} \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quan
3.4. PROGRAM

```c
#include "boptions.h"
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <graphics.h>

float V,T,V1,V2,V3,V4,P1,d5,sp,hoe;
int A,B,C,CR, P, R, CR1, f1, f, x, y, x2, y2, a, b, t, b1, b2,
c1, n, m, g, gm, ss, ss1, df, d2, d3, d4, o, pol, j;
long int i;
char letter[173];
float V5[153];
float It[153];
extern int tibeax, ttype;
void npndraw(void);
void npndraw(void);
void ujtdraw(void);

A = 0x124;
B = 0x125;
C = 0x126;
CR = 0x127;
outp (CR,0x92);
P = 0x130;
Q = 0x131;
R = 0x132;
CR1 = 0x133;
outp (CR1,0x80);
void npndraw()
{
    switch (tibmax) {
    case MICRO10 :
        outp(C,0x00) /* send code to port */
        break;
    case MICRO100 :
        outp(C,0x10)
        break;
    case MILLI1 :
        outp(C,0x20)
        break;
    case MILLI10 :
        outp(C,0x30)
        break;
    case MILLI100 :
        outp(C,0x40)
        break;
    }
    gd = CGA;
gm = CGAH!
P1 = 80.0;
Q1 = 160.0;
initgraph(&gd,&gm,"" );
line (80,160,600,160);
line (80,160,80,10);
```
_scale();

f1 = 0x60; 
for (i = 1; i <=14000; ++i) {
   dotoutp (Q,f1);
   P1 = 80.0;
   Q1 = 160.0;
   j=0;
   f = 0x0;
   dotoutp (P,f);
   for (i=1; i<=14000; ++i){
   } 
   outp (R,0x40);
   for (i=1;i<=80000;++i){
   }
   _adc();
   V = d5;
   V1 = V4+4.0;
   outp (R,0x00);
   for (i=1;i<=80000;++i){
   }
   _adc();
   _pbr();
   V2 = d5;
   V3 = V2+4.0;
   V4 = V1-V3;
   V5(j)= abs(V3*4.0);
   ICj3 =abs(V4/5.0)+1500.0;

   line (P1,Q1,(V5(j)+80.0),(160.0-ICj3));

   P1 = 80.0+V5(j);
   Q1 = (160.0-ICj3);
   f = f+0x1e;
   j=j+1;
   while(f<=0x1e3);
   outp(P,0x000);
   f1 = f+0x0f;
   if(f1 == 0x1e3)
      _slope();
   )while(f<=0x1e3);
   slope();
   outp(0,0x00);
   closegraph();
}

tranf();
tranr();
stem();
getch();
restorecrtmode();
)
```c
_slope()
{
    char s[203];
    ho=(IC53-IC33)/(V5C53-V5C33)*40.0/1500.0;
}
_adc()
{
    A = 0x124;
    B = 0x125;
    C = 0x126;
    CR = 0x127;
    for (i=1;i<=3000;++i)
    {
        do{
            ss=inp(B);
            ss1 = 0x40 & ss;
        }while(ss1!=0x40);
        d1=inp(A);
        d2=inp(B);
        a=0x10 & d2;
        pol=0x20 & d2;
        if (pol==16)
        {
            printf("over range");
        }
        else
        {
            d3=(15 & d2);
            d4=(d3*256+d1);
            d5=(2.0*2.0*d4)/4096.0;
            if (pol==32)
            {
                d5=d5;
            }
            else
            {
                d5=-d5;
            }
        }
    }
}
_scale()
{
    b1=15;
    for (i=0;i<=10;++i)
    {
        line(80,160-b1,70,160-b1);
        b1=b1+15;
    }
    b2=40;
    for(i=0;i<10;++i)
    {
        line(80+b2,160,80+b2,165);
        b2=b2+40;
    }
    a=20;b=10;
    for(i=1;i<10;++i)
    {
        gotoxy(7,a);
        printf("%d",b);
        a=a-2;
        b=b+10;
    }
    x2=15;
y2=1;
```
for(i=1;i<=10;++i){
gotoxy_Ex2,22);
printf("H2d",y2);
x2=x2+5;
y2=y2+1;
}
gotoxy(19,24);
printf("COLLECTOR-EMITTER VOLTAGE VCE (V)\n");

_pbr()
{
    for(t=0;t<=16;++t)
    {
        gotoxy(4,4+t!);
        printf("Xc", letter[t]);
    }
}

void pnpdraw(void)
{
    void npndraw();
}

void ujtdraw(void)
{
    void npndraw();
}

 tranf()
{
    float V,T,V1,V2,V3,V4,V5,V6,V7,V8,V9,V10,V11,P1,01,d5,hfe,y2;
    long int i;
    char letter[17];
    float ICC[15];
    float IB[15];

    intial();

    _scale1();

    f1 = 0x2d;
    do
    {
        outp (P,f1);
        for (i = 1; i <=14000; ++i) {
            P1 = 80.0;
            Q1 = 160.0;
            j=0;
            f = 0x2d;
            do
            {
                outp (Q,f);
                for (i=1; i<=14000; ++i) {
                    outp (R,0x00);
                    for (i=1;i<80000;++i){
                }
\_adc();

V = d5;
V1 = V;
outp (R,0x80);
for (i=t;i<=80000;i++)
{ }
\_adc();
\_pbr();
V2 = d5;
V3 = V2;

outp(R,0x40);
for (i=t;i<=80000;i++)
{ }
\_adc();

V7 = d5;
V8 = V7*4.0;

outp(R,0xc0);
for (i=t;i<=80000;i++)
{ }
\_adc();

V9 = d5;
V10 = V9*4.0;

V4 = V1-V3;
V11 = V8-V10;

IBCj3 =abs(V4/1500.0)*225000.0;
ICLj3 =abs(V11/5)*1500.0;

line (P1,G1,(IBCj3+80.0),(160.0-ICLj3));
P1 = 80.0+IBCj3;
G1 = (160.0-ICLj3);

f = f+0x1e;
j=j+1;

)while(f<0x0f0);
outp(G,0x00);
f1 = f+0xf;

)while(f1<0x40);
outp(P,0x00);
slope();
gokey(52,5);
printf("VCE=1V");
gokey(50,4);
printf("VCE=2V");
closegraph();
)
slope()
{
    hfe=(ICC53-ICC43)/(IBC53-IBC43)*(225000.0/1500.0);
}

scale()
{
    b1=15;
    for (i=0;i<=10;++i)
    {
        line(80,160-b1,70,160-b1);
        b1=b1+15;
    }

    b2=40;
    for(i=0;i<10;++i)
    {
        line(80+b2,160,80+b2,165);
        b2=b2+40;
    }

    a=20;b=1;
    for(i=1;i<10;++i)
    {
        gotoxy(7,a);
        printf("%d",b);
        a=a-2;
        b=b+1;
    }

    x2=15;
    y2=0.2;
    for(i=1;i<10;++i)
    {
        gotoxy(x2,22);
        printf("X1.1f",y2);
        x2=x2+5;
        y2=y2+0.2;
    }

    gotoxy(19,24);
    printf("BASE CURRENT IB (mA)\n");
}

_pbrf()
{
    char letter[20] = { "Ic","c","D","L","E","C","T","R","E","U","R","E","N","T","mA"};
    for(t=0;t<19;++t)
    {
        gotoxy(4,2+t);
        printf("%c",letter[t]);
    }
}

initial()
{
    A = 0x124;
    B = 0x125;
    C = 0x126;
    CR = 0x127;
    gd = CGA;
    gm = CGA1;
    P1 = 80.0;
    Q1 = 160.0;
initgraph(&d,&gm,"";
line (80,160,600,160);
line (80,160,80,10);

outp (CR,0x92);

P = 0x130;
Q = 0x131;
R = 0x132;
CR1 = 0x133;
outp (CR1,0x80);
}
trani()
{
    float V,T,V1,V2,V3,V4,P1,Q1,d5,hi,e,y2;
    long int i;
    char letter[17];
    float vs[153];
    float IC[153];
    intial ()
    {
        _scale2();
        f1 = 0x1e;
        do{
            outp (P,f1);
            for (i = 1; i <=14000; ++i) {
                P1 = 80.0;
                Q1 = 160.0;
                j=0;
                f = 0x1e;
                do{
                    outp (Q,f);
                    for (i=1; i<=80000;++i){
                        adc(>;
pbr2();
                    }
                    outp (R,0x00);
                    for (i=t;i<=80000;++i)}
                _adc();
                V = d5;
                V4 = V=4.0;
                outp (R,0x80);
                for (i=1;i<=80000;++i){
                _adc();
                _pbr2();
                V2 = d5;
                V3 = V2*4.0;
                V4 = V1-V3;
                V5[i] = V3*40.0;
                IC[i]=(V4/1500)*19500.0;
line (P1,01, (V5CJ3,80.0), (160.0-I3J3));

P1 = 80.0+V5CJ3;
01 = (160.0-I3J3);

f = f+0x10;
j = j+1;

while(f<0xfo);
outp(0,0x00);
f1 = f+0x1f;
while(f1<0x2d);
outp(P,0x00);
slope21;
gotoxy(20,b);
printf("VCE=1V");
gotoxy(30,4);
printf("VCE=2V");
closegraph();}

_slope2();
{
    hie=(IC53-I3C3)/(V5CJ3-V5CJ3)*(40.0/19500.0);
}

_scale2();
{
    b1=15;
    for (i=0;i<=10;++i){
        line(80,160-b1,70,160-b1);
        b1=b1+15;
    }
    b2=40;
    for (i=0;i<10;++i){
        line(80+b2,160,80+b2,165);
        b2=b2+40;
    }
    a=20;b=1;
    for (i=0;i<10;++i){
        gotoxy(7,a);
        printf("d",b);
        a=a-2;
        b=b+1;
    }
    x2=15;
    y2=0.2;
    for (i=0;i<10;++i){
        gotoxy(x2,22);
        printf("%1.1f",y2);
        x2=x2+5;
        y2=y2+0.20;
    }
    gotoxy(19,24);
    printf("fifth EMITTER VCLTA6E VBE (V)");
}
_pbr2()
{
    for(t=0; t<=12; ++t){
        gotoxy(4, 4 + t);
        printf("Zc", letter[t]);
    }
}

float V, T, V1, V2, V3, P1, Q1, d5, hre;
long int i;
char letter[173];
float VCC[153];
float VBEC[153];

initial()
{
    _scale3();
    f1 = 0x1e;
    do{
        outp(0, f1);
        for(i = 1; i <= 14000; ++i) {
            P1 = 80.0;
            Q1 = 160.0;
            j=0;
            f = 0x1e;
            do{
                outp(P, f);
                for(i=1; i<=8000; ++i){
                    _adc();
                    V = d5;
                    V1 = V;
                    outp(R, 0x40);
                    for(i=1; i<=8000; ++i){
                        _adc();
                        _pbr3();
                        V2 = d5;
                        V3 = V2*4.0;
                        VCC[j] = abs(V3*40.0);
                        VBEC[j] = abs(V1*100.0);
                        line(P1, Q1, (VCC[j]+80.0), (160.0-VBEC[j]));
                    }
                }
            }
        }
    }
}

P1 = 80.0 + VCC[j];
Q1 = 160.0 - VBEC[j];
f = f+0x1e;
j=j+1;
while(f<=0xe1);
outp(P,0x00);
f1 = f1+0x3c;
}
while(f1<=0x60);
outp(0,0x00);
gotxy(60,14);
printf("Ib=0.1mA");
gotxy(60,12);
printf("Ib=0.2mA");
\_slope3();
gotxy(60,12);
\_slope3();
{ 
 hrer=(VBEC53-VBEC33)/(VCE53-VCE33)*(40.0/100.0);
}
\_scale3()
{
 b1=15;
 for (i=0;i<=10;++i){
 line(80,160-b1,70,160-b1);
 b1=b1+15;
 }
 b2=40;
 for(i=0;i<=10;++i){
 line(80+b2,160,80+b2,165);
 b2=b2+40;
 }
a=20;b=2;
 for(i=0;i<=10;++i){
 gotxy(7,a);
 printf("Xd",b);
 a=a-2;
 b=b+2;
 }
x2=15;
y2=1;
 for(i=0;i<=10;++i){
 gotxy(x2,22);
 printf("%d",y2);
 x2=x2+5;
y2=y2+1;
 }
gotxy(19,24); 
 printf("COLLECTOR-EMITTER VOLTAGE VCE (V)\n");
}
\_pbr3()
{
 for(t=0;t<=19;++t){
 gotxy(4,t+1);
 printf("%c",letter[t]);
 }
}
```c
#include <stdio.h>

#define pi 3.1415926536

int main(void)
{
    float fL, fH, rd, rdH, gM, Rs, U, CC, CS;
    float Rg = 2.2;
    float RD = 2.2;
    Rs = (1/(1/RD+1/RL))/AV;
    U = rd * gH;
    CC = 1/(2*PI*fL*Rg);
    CS = 1/(2*PI*fL*RL);

    // Processing...

    return 0;
}
```
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <graphics.h>

extern int ftype, fvmax;

void nfdraw(void)
{
    feto();
    feti();
    famp();
}

void pfedraw(void)
{
    feto();
    feti();
    famp();
}

feto()
{
    float V, V1, V2, V3, V4, P1, B1, d5, rd;
    int A, B, C, CR, P, Q, R, CR1, f1, f, x, y, x2, y2, a, b, l, b1, b2, c1,
        n, m, g, gd, gm, ss, s, s1, d1, d2, d3, d4, o, pol, j;
    long int i;

    char letter[173];
    float V5[153];
    float IC153;
    A = 0x124;
    B = 0x125;
    C = 0x126;
    CR = 0x127;

    gd = CGA;
    gm = CGAH1;

    P1 = 80.0;
    O1 = 160.0;

    initgraph(&gd, &gm, "");
    line (80, 160, 600, 160);
    line (80, 160, 80, 10);
    outp (CR, 0x92);

    P = 0x130;
    Q = 0x131;
    R = 0x132;
    CR1 = 0x133;
    outp (CR1, 0x80);
    _scale();
    f1 = 0x00;
do{
  outp (0, f1);

  for (i = 1; i <= 14000; ++i) {
  }
P1 = 80.0;
Q1 = 160.0;
j = 0;
f = 0x2d;
do{
  outp (P, f);
  for (i = 1; i <= 14000; ++i) {
  }
  outp (R, 0x40);
  for (i = 1; i <= 14000; ++i) {
  }
  _adc();
  V = d5;
  V1 = V4*4.0;
  outp (R, 0x40);
  for (i = 1; i <= 14000; ++i) {
  }
  _adc();
  _br();
  V2 = d5;
  V3 = V2*4.0;
  V4 = V1 - V3;
  V5 = V3*4.0;
  ICj3 = (V4/5.0)*2400.0;
  line (P1, Q1, (V5E53 + 80.0), (160.0 - ICj3));
P1 = 80.0 + V5E53;
Q1 = (160.0 - ICj3);
f = f + 0x1e;
j = j + 1;
}
}while (f <= 0x2f);
outp (P, 0x00);
f1 = f + 0x1e;
}
while (f1 <= 0xff);
slope();
outp (P, 0x00);
closegraph();
}
slope()
{
  rd = (ICj3 - ICj3)/(V5E33 - V5E33) * (40.0/2400.0);
}
_adc() {
    A = 0x124;
    B = 0x125;
    C = 0x126;
    CR = 0x127;

    for (i=1;i<=30;++i) {
        do {
            ss = inp(B);
            ssf = 0x40 & ss;
        } while (ssf=0x40);
        d1 = inp(A);
        d2 = inp(B);
        o = 0x10 & d2;
        pol = 0x20 & d2;
        if (o==16) {
            printf("over range");
        } else {
            d3 = (15 & d2);
            d4 = (d3*256+d1);
            d5 = (2.0*d2.0*d4)/4096.0;
            if (pol==32) {
                d5 = d5;
            } else {
                d5 = -d5;
            }
        }
        d5 = scale();
    }
}

_scale() {
    b1=15;
    for (i=0;i<=10;++i) {
        line(80,160-b1,70,160+b1);
        b1 = b1 + 15;
    }

    b2=40;
    for (i=0;i<10;++i) {
        line(80+b2,160,80+b2,165);
        b2 = b2 + 40;
    }

    a=20;b=10;
    for (i=1;i<=10;++i) {
        gotoxy(7,a);
        printf("M",b);
        a=a-2;
        b=b+10;
    }

    x2=15;
    y2=1;
    for (i=1;i<=10;++i) {
        gotoxy(x2,y2);
        printf("X",y2);
        x2=x2+15;
        y2=y2+1;
    }
}

x2=x2+5;
y2=y2+1;
}
gotoxy(19,24);
printf("DRAIN-SOURCE VOLTAGE VDS (V)\n");

_pbr()
{
  char letterC133 = {'D', 'R', 'I', 'N', 'E', 'T'},
                    'C', 'U', 'R', 'E', 'N', 'T'};
  for(t=0; t<=12; ++t){
      gotoxy(4, 4+t);
      printf("Xcn,letterCt3); 
  }
  feti()
  {
    float V, T, V1, V2, V3, V4, V6, P1, Q1, d5, gM;
    int A, B, C, CR, P, Q, R, CR1, f1, x, y, t, b1, b2, 
        c1, n, m, g, gd, gm, ss, ss1, d1, d2, d3, d4, o, pol, j;
    long int i;
    char letterC173;
    float V5E153;
    float 1C153;
    P1 = 540.0;
    Q1 = 160.0;
    initgraph(&g«, "l);
    line (80, 160, 540, 160);
    line (540, 160, 540, 10);
    outp (CR, 0x92);
    P = 0x130;
    Q = 0x131;
    R = 0x132;
    CR1 = 0x133;
    outp (CR1, 0x80);
    _scale();
    f1 = 0x3c;
    do{
      outp (P, f1);
      for (i = t; i <=14000; ++i) {
      }
      P1 = 80.0;
      Q1 = 160.0;
      j =0;
      f = 0x0;
      do{
        outp (Q, f);
        for (i=t1; i<=14000; ++i){
      }
      outp (R, 0x01);
      for (i=t1; i<=60000; ++i){
      }
\_\text{add}();
V6 = d5;

\text{outp}(R,0x40);
\text{for}(i=1;i<80000;++i)\{
\}
\_\text{add}();

V = d5;
V1 = V4.0;
\text{outp}(R,0xCO);
\text{for}(i=1;i<80000;++i)\{
\}
\_\text{phr}();
V2 = d5;
V3 = V2*4.0;
V4 = V1-V3;
V5j3 =\text{abs}(V6+40.0);
ICj3 =\text{abs}(V4/5.0)*1500.0;
\text{line}(P1,01,(V5Cj3+540.0),(160.0-ICj3));
P1 = 540.0+V5[Cj3];
Q1 = (160.0-ICj3);
f = f+0x1e;
j=j+1;
\text{while}(f<0x90);
\text{outp}(0,0x00);
f1 = f+0xf;
\text{while}(f1<0x40);
slope2();
\text{outp}(P,0x00);
closegraph();
\}
\_\text{slope2}();
\{
\text{gm}=(IC5j-IC3j)/(V5[C5j-V5[C3j]140.0/2000.0);\}
\_\text{scale}();
\{
b1=15;
\text{for}(i=0;i<10;++i)\{
\text{line}(540,160-b1,550,160-b1);
b1=b1+15;
\}
b2=40;
\text{for}(i=0;i<10;++i)\{
\text{line}(80+b2,160,80+b2,165);
b2=b2+40;
\}

a=20;b=5;
for(i=1;i<=10;++i){
gotoxy(70,a);
printf("X%d",b);
a=a-2;
b=b+5;
}
x2=x2+5;
y2=y2+1;
for(i=1;i<=10;++i){
gotoxy(x2,y2);
printf("%d",y2);
}
gotoxy(19,24);
printf("COLLECTOR-EMITTER VOLTAGE (Vce)\n");
}
_pbr()
{
    for(t=0;t<=16;++t){
gotoxy(74,4+t);
printf("%c",letter[t]);
}
}

/* Program for the Design of a Single Stage Common-Source Amplifier */

float R1,R2,RD,Rg,Rs,U,CS,RL,rd,gH,CC,AV;
float fL,fH,VDS,VD;PI,Zi,Zo,at,bt,ct,dt,et;
PI = 3.141;
initgraph(&gd,&gm,""y); rectangle(0,50,500,190);
line(0,68,500,68); gotoxy(4,4);
printf("Amplifier Circuit Design using FET BM 10\n");
printf("\n\n\n");
goxy(4,8); printf("Requirements Design Values Experimental Values\n");
goxy(6,13); printf("VDS in V=");
scanf("%f",&VDS);
goxy(6,14); printf("RL in K =");
scanf("%f",&RL); gotoxy(6,15); printf("AV =");
scanf("%f",&AV); gotoxy(6,16); printf("fL in Hz =");

scanf("f",&FL);
gotoxy(6,17);
printf("fH in KHz = ");
scanf("f",&FH);
gotoxy(6,18);
printf("rd in K= ");
scanf("f",&rd);
gotoxy(6,19);
printf("gM = ");
scanf("f",&GM);
gotoxy(6,20);
printf("Zi in K = ");
scanf("f",&Zi);
gotoxy(6,21);
printf("Zo in K= ");
scanf("f",&Zo);

Rg = Zi;
RD = Zo;
Rs = (1/(1/RD+1/RU))/AV;
U = rd * gM;
CC = 1/(2*PI*fL*Rg);
CS = 1/(2*PI*fL*RL);

goxy(24,13);
printf("RD=2.22f\n",RD);
goxy(24,14);
printf("Rg=2.22f\n",Rg);
goxy(24,15);
printf("rd=2.22f\n",rd);
goxy(24,16);
printf("gM=2.22f\n",gM);
goxy(24,17);
printf("CC=2.22f\n",CC);
goxy(24,18);
printf("CS=2.22f\n",CS);
goxy(24,19);
printf("U=2.22f\n",U);
goxy(48,13);
scanf("f",&at);
goxy(48,14);
scanf("f",&bt);
goxy(48,15);
scanf("f",&ct);
goxy(48,16);
scanf("f",&dt);
goxy(48,17);
scanf("f",&et);
getch();
}
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <graphics.h>

float V,T,V1,Vp,Vv,Vp,M,
int A,B,C,Ry2,PTQ,R,CR1,f1,f1x,y,x2,x,a>b,t,b1,b2,c1,nr»f9,gd,ga,ss,ss1,d1,d2,d3,d4,o,pol,j;
long int i;
char letterC173;
float V5C153;
float IC153;

main ()
{
    A = 0x124;
    B = 0x125;
    C = 0x126;
    CR = 0x127;
    gd = CGA;
    gm = CGAH;
    P1 = 80.0;
    Q1 = 160.0;
    inilgraph(&gd,&gm,""); 
    line (80,160,600,160);
    line (80,160,80,10); 
    outp (CR,0x92);
    P = 0x130;
    Q = 0x131;
    R = 0x132;
    CR1 = 0x133;

    outp (CR1,0x80);
    _scale();

    f1 = 0x2d;
    dot
    outp (P,f1);
    for (i = 1; i <=14000; ++i) C
    outp (R,0x00); 
    for (i=1;i<=8000;++iK
    adc();
    V = d5;
    VI = V*4.0; 
    outp (R,0x80); 
    for (i=1;i<=80000;++iK
}
ADC();
_pbr();
V2 = d5;
V3 = V2*4.0;
V4 = V1-V3;
V5Cj = (V3*15.0);
ICj = (V4/33)*800.0;
line (P1,Q1,(ICj+80.0),(160.0-V5Cj));
P1 = 80.0+ICj;
Q1 = (160.0-V5Cj);
f = f+0xf;
j=j+1;
while(f<0xff);
outp(G,0x00);
f1 = f1+0xf;
while(f1<0x30);
minim();
outp(P,0x00);
_ump();
closegraph();

minim()
{
  for(i = j; j<10; ++j){
    if(Vv <= V5Cj)
      V5Cj = Vv;
    if(lv <= ICj)
      ICj = lv;
  }
  for(i = j; j<10; ++j){
    if(Vp >= V5Cj)
      V5Cj = Vp;
    if(lp <= ICj)
      ICj = lp;
  }
}
_adc()
{
  A = 0x124;
  B = 0x125;
  C = 0x126;
  CR = 0x127;
  for (i=1;i<=3000;++i){
    }
    do{
      ss=inp(B);
      ssf= 0x40 & ss;
    }while(ssf!=0x40);
d2=inp(B);
d2=inp(B);
o=0x10 & d2;
pol=0x20 & d2;
if (o==16K
printf("over range");
}else

d3=(15 & d2);
d4=(d3*256+d1);
d5=(2.0*2.0*d4)/4096.0;
if (pol=32){
d5=d5;
}
else

d5=-d5;
}

_scale()
{
  b1=15;
  for (i=0;i<10;++i)
    line(80,160-b1,70,160-b1);  
b1=b1+15;
}

  b2=40;
  for(i=0;i<10;++i){
    line(80+b2,160,80+b2,165);
b2=b2+40;
}
  a=20; b=1;
  for(i=1;i<10;++i){
    gotoxy(7,a);
    printf("%d", b);
    a=a-2;
b=b+1;
  }
x2=15;
y2=5;
  for(i=1;i<10;++i){
    gotoxy(x2,22);
    printf("%d", y2);
x2=x2+5;
y2=y2+5;
  }
  gotoxy(19,24);
  printf("EMITTER CURRENT (IE)\n");
}

_pbr()
{
  for(t=0;t<15;++t){
    gotoxy(4,t+4);
    printf("Zc",letter[t]);
  }
uamp()
{
/* Program for the Design of a Relaxation Oscillator */
float R,Rmax,Rmin,n,C,Vv;
float fl,fH, VBB,Vp,Ip,lv,1,1;

initgraph(&gd,&gm,"");
rectangle(0,50,500,190);
line(0,68,500,68);
gotoxy(4,4);
printf("Relaxation Circuit Design using UJT 2N2646 \n");
printf(" \n")f
gotoxy(4,8);
printf("Requirements Design Values Experimental Values\n");
gotoxy(6,12);
printf("VBB in V=" );
scanf("%f", &VBB);
gotoxy(6,13);
printf("Rmax in K = " );
scanf("%f", &Rmax);
gotoxy(6,14);
printf("Rmin in K = ");
scanf("%f", &Rmin);
gotoxy(6,15);
printf("Vp in V = ");
scanf("%f", &Vp);
gotoxy(6,16);
printf("Vv in V= ");
scanf("%f", &Vv);
gotoxy(6,17);
printf("Ip = ");
scanf("%f", &Ip);
gotoxy(6,18);
printf("Iv = ");
scanf("%f", &Iv);
gotoxy(6,19);
printf("t in sec.=");
scanf("%f", &t);

n = Vp/VBB;
Rmax = (VBB-Vp)/Ip;
Rmin = (VBB-Vv)/Iv;
R = sqrt(Rmax*Rmin);
l = log(1/(1-n));
C = 3.2030*l/1;

gotoxy(24,13);
printf("R=%f\n",R);
gotoxy(24,14);
printf("C=%f\n",C);
gotoxy(24,15);
printf("n=%f\n",n);
getch();
}
```c
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <graphics.h>

extern int ibias, ivnax;

void diodeforward(void)
{
    diodefO;
}

void diodereverse(void)
{
    dioderO;
    diodefulU;
}

diodef()
{
    float V, T, V1, V2, N3, N4, P1, Q1, d1, d3, sp;
    int A, B, C, CR, P, Q, R, CR1, f1, f, x, y, x2, y2, a, b, t, b1,
        b2, c1, n, m, gd, gm, ss, ss1, d1, d2, d3, d4, o, pol, j;
    long int i;
    char letter173;
    float V5C153;
    float IC153;

    A = 0x124;
    B = 0x125;
    C = 0x126;
    CR = 0x127;

    gd = CGA;
    gm = CGW1;

    P1 = 80.0;
    Q1 = 160.0;
    initgraph(&gd, &gm, "")
    line (80, 160, 600, 160);
    line (80, 160, 80, 10);
    outp (CR, 0x92);
    P = 0x130;
    Q = 0x131;
    R = 0x132;
    CR1 = 0x133;
    outp (CR1, 0x80);
    _scale();
    P1 = 80.0;
    Q1 = 160.0;
    j = 0;
    f = 0x1E;
    do
    {
        outp (P, f);
        for (i = 1; i <= 14000; ++i)
        {
        }
        outp (R, 0x40);
        for (i = 1; i <= 80000; ++i)
        {
        }
    } while (1);
```
adc();
V = d5;
V4 = V4.0;
outp (R,0xc0);
for (i=1;i<40000;i++)
adc();
pbr();
V2 = d5;
V3 = V2*4.0;
V4 = V1-V3;
V5[j] = abs(V3*200.0);
if[j] = abs(V4/33.0)*1200.0;
line (P1,Q1,(V5[j]+80.0),(160.0-I[j]));

P1 = 80.0+V5[j];
Q1 = (160.0-I[j]);
f = f+0x1e;
j=j+1;
while(f<0xc3);
outp(P,0x00);
slope();
getch();
}
slope()
{
Rf=(I[j]-I[j])/(0xc5j-V[j]/1200.0);
}
adc()
{
A = 0x124;
B = 0x125;
C = 0x126;
CR = 0x127;
for (i=1;i<3000;++i)
{
do{
ss=inp(B);
ssf= 0x40 & ss;
while(ssf!=0x40);
d1=inp(A);
d2=inp(B);
o=0x10 & d2;
pol=0x20 & d2;
if (o==16)(
printf("over range");
}
else
d3=(15 & d2);
d4=(d3*256+d1);
d5=(2.0*2.0*d4!/4096.0;
if (pol==32)(
	d5=+d5;
}
else
d5=-d5;
.scale()
{
    b1=15;
    for (i=0;i<10;++i)
    line(80,160-b1,70,160-b1);
    b1=b1+15;
}

b2=40;
for(i=0;i<10;++i)
line(80+b2,160,80+b2,165);
b2=b2+40;

a=20; b=5;
for(i=1;i<=10;++i)
    gotoxy(7,a);
    printf("%d",b);
    a=a-2;
    b=b+5;
}
x2=15;
y2=2;
for(i=1;i<=10;++i)
    gotoxy(x2,22);
    printf("Xc",letterE153);
}

stdout()
{
    for(i=0;i<14;++i)
        gotoxy(4,4+i);
        printf("Xc",letterE153);
}

dioder()
{
    float V,T,V1,V2,V3,V4,P1,Q1,d5,sp;
    int A,B,C,R,P,q,R,C,R1,f1,f,x,y,x2,y2,a,b,t,b1,b2,c1,n,m,g,gd,gm,ss,ss1,d1,d2,d3,d4,o,pol,j;
    long int i;
    char letterE173;
    float V5E153;
    float IC153;

    P1 = 540.0;
    Q1 = 30.0;
    initgraph(&gd,&gm,"*");
    line(80,30,540,30);
    line(540,180,540,30);
    outp(CR,0x921);
P = 0x130;
Q = 0x131;
R = 0x132;
CR1 = 0x133;

outp (CR1,0x80);
_scale();
P1 = 540.0;
Q1 = 30.0;
j=0;
f = 0x00;
do{
  outp (P,f);
  for (i=1; i<=14000; ++i){
    
  }
  outp (R,0x40);
  for (i=1;i<=80000;++i){
    
  }
  _adc();
V = d5;
V1 = V4.0;
outp (R,0x00);
for (i=1;i<=80000;++i){
  
}
_pbr();
V2 = d5;
V3 = V2*4.0;
V4 = V1-V3;
V5[j] = V3*40.0;
IC[j] = (V4/33.0)*1500.0;
line (P1, Q1, (V5[j]+540.0), (30.0-I[j]));
P1 = 540.0+V5[j];
Q1 = (30.0-I[j]);
f = f+0x1e;
j=j+1;
}while(f<=0x33);
slope();
outp(P,0x00);
getch();
}
slope()
{
  sp=(IIC[j]+IIC3)]/(V5[j]+V53)*40.0/200.0;
gotoxy(25,45);
printf("%f",sp);
scale()
{
    b1=15;
    for (i=0;i<9;++i){
        line(540,30+b1,550,30+b1);
        b1=b1+15;
    }
    b2=40;
    for(i=0;i<10;++i){
        line(540-b2,30,540-b2,25);
        b2=b2+40;
    }

    a=24;b=50;
    for(i=1;i<10;++i){
        gotoxy(70,a);
        printf("Xa",b);
        a=a-2;
        b=b-5;
    }
    x2=17;
    y2=10;
    for(i=1;i<10;++i){
        gotoxy(x2,y2);
        printf("Xd",b);
        x2=x2+5;
        y2=y2-1;
    }
    gotoxy(30,2);
    printf("REVERSE VOLTAGE Vr (V)"); 
}

_pbr()
{
                      'C','U','R','E','N','T'};
    for(t=0;t<14;++t){
        gotoxy(74,5+t);
        printf("Xe",letter[t]);
    }
}
diodeful()
{
    /* Program for the Design of a Half wave Rectifier */

    float Ef,Vp,If,Vf,Rf;
    int gd,gm;

    gd = CGA;
    gm = CGA11;

    initgraph(&gd,&gm,"");
    rectangle(0,50,500,190);
    line(0,68,500,68);
gotoxy(4,4);
printf("Half-wave rectifier Circuit Design using VPR 25\n");
printf("\n\n");
gotoxy(4,8);
printf("Requirements	Design Values	Experimental Values\n");
gotoxy(6,12);
printf("Ef in V="");
scanf("%f",&Ef);
gotoxy(6,13);
printf("Vf in V="");
scanf("%f",&Vf);
gotoxy(6,14);
printf("Rf in K=");
scanf("%f",&Rf);

Vp = (Ef-Vf);

gotoxy(24,13);
printf("Vp =%f V\n",Vp);
getch();
}
}