```
unit GannU;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ComCtrls, ExtCtrls;

type
  TForm1 = class(TForm)
    Label1: TLabel;
    LengthEdit: TEdit;
    Label2: TLabel;
    mEdit: TEdit;
    m1Edit: TEdit;
    m2Edit: TEdit;
    fEdit: TEdit;
    U_1Edit: TEdit;
    U_0Edit: TEdit;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Button1: TButton;
    ProgressBar1: TProgressBar;
    EndTimeEdit: TEdit;
    Label9: TLabel;
    PaintBox1: TPaintBox;
    Cxn0: TCheckBox;
    Cxmu1: TCheckBox;
    Cxmu2: TCheckBox;
    CE: TCheckBox;
    CV1: TCheckBox;
    CV2: TCheckBox;
    Ct12: TCheckBox;
    CC: TCheckBox;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
function ExternalU(t:Extended):Extended;
procedure Puasson(Sender: Tobject; t: Extended);
procedure InitAll(Sender: Tobject);
procedure UpdateSettings(Sender: Tobject);
procedure UpdateProgressbar(Sender: TObject;a,b: Integer);
procedure MyDrawArray(Sender: Tobject);
procedure FreeAllSamples(Sender: Tobject);

var
  Form1: TForm1;
  t, sig   : Extended;
  t_end, nt: Extended;
  Length   : Extended;           //Legth of crystal
  m, m1, m2: Integer;            //Scatter parameters
  U_0, U_1 : Extended;           //External voltage amplitude
  R        : Extended;           //R
  t21, te1 : Extended;           //Model parameters
  vs1, vs2 : Extended;           //Speeds
  Delta    : Extended;           //The difference between lower and
                                 //higher glades
  xmu10, xmu20: Extended;        //A moving abilities
  t0         : Extended;         //Grid temperature
  D1, D2, xno1, xno2: Extended;  //Ds and admixtures
  Eps, f, ht  : Extended;        //Epsylon, frequency, time step
  dz         : Extended;
  sigma      : extended;
  e_t, e, t12: array of Extended;
  xn0,xn1,xn2: array of Extended;
  T1, V1, V2 : array of Extended;
  xmu1, xmu2 : array of Extended;
  Cur        : array of Extended;
```

```
const
  Eps0 = 8.85e-14;              //Epsylon zero and electron charge
  El   = 1.6e-19;               //I think they're won't change...

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin
  Application.Title := 'Gann Diode';
  Length:= 3e-4;            //Appending the first-time settings
  m := 61;
  m1:= 4;
  m2:= 7;
  U_0 := 4;
  U_1 := 2;
  R   := 60;
  t21 := 2.0e-12;
  te1 := 0.8e-12;
  vs1 := 2e7;
  vs2 := 5e6;
  xmu10 := 7500;
  xmu20 := 150;
  Delta := 0.36;
  t0 := 1.2e-2;
  D1 := 20;
  D2 := 10;
  xno1  := 2e16;
  xno2  := 7.5e15;
  Eps   := 12.5;
  f  := 3.5e10;
  ht := 1e-15;
  t_end:= 5e-12;
  //Initialize Edits
  LengthEdit.Text:= FloatToStr(Length);
  m1Edit.Text := IntToStr(m1);
  m2Edit.Text := IntToStr(m2);
  mEdit.Text  := IntToStr(m);
  fEdit.Text  := FloatToStr(f/1e6);
  U_0Edit.Text:= FloatToStr(U_0);
  U_1Edit.Text:= FloatToStr(U_1);
  EndTimeEdit.Text:= FloatToStr(t_end*1e9);
  //Initialize CheckBoxes
  Cxn0.Checked:= true;
  CV1.Checked:= true;
  CV2.Checked:= not true;
  CE.Checked := true;
  Cxmu1.Checked:= not true;
  Cxmu2.Checked:= not true;
  CC.Checked:= true;


end;

//Define an external voltage as function.
function ExternalU(t:Extended):Extended;
begin
  Result := U_0 + U_1 * sin(2*Pi*f*t);
end;

//Make User to see a progress
procedure UpdateProgressbar(Sender: TObject; a,b: Integer);
begin
  with Form1.Progressbar1 do
    begin
      Position:= Round(a/b*Max);
      Application.ProcessMessages;
    end;

end;

//Emptying all arrays before using 'em again
procedure FreeAllSamples(Sender: Tobject);
var i: Longint;
begin
```

```
    For i:= 1 to m do
      begin
        xn0[i]:=0;
        xn1[i]:=0;
        xn2[i]:=0;
        e_t[i]:=0;
        e[i]  :=0;
        T1[i] :=0;
        V1[i] :=0;
        V2[i] :=0;
        t12[i]:=0;
        xmu1[i]:=0;
        xmu2[i]:=0;
        Cur[i]:=0;
      end;
  end;

  //Procedure to relocate memory
  procedure MRelock(Sender: Tobject);
  begin
      //Must Preserve a Memory because arrays are dynamic.
    try
      SetLength(xn0,m+2);
      SetLength(xn1,m+2);
      SetLength(xn2,m+2);
      SetLength(e_t,m+2);
      SetLength(e,  m+2);
      SetLength(T1, m+2);
      SetLength(V1, m+2);
      SetLength(V2, m+2);
      SetLength(t12,m+2);
      SetLength(xmu1,m+2);
      SetLength(xmu2,m+2);
      SetLength(Cur, m+2);
    except
      begin  //Except, if EOnOutOfMemory error. Needed!
        ShowMessage('Недостаточно памяти. Уменьшите число шагов.');
        Exit;  //Must terminate, or...
      end;
    end;
  end;

  //Procedure to apply User-changed settings
  procedure UpdateSettings(Sender: Tobject);
  begin
    try
    Length:= StrToFloat(Form1.LengthEdit.Text);
    m1    := StrToInt(Form1.m1Edit.Text);
    m2    := StrToInt(Form1.m2Edit.Text);
    m     := StrToInt(Form1.mEdit.Text);
    f     := StrToFloat(Form1.fEdit.Text)*1e6;
    U_0   := StrToFloat(Form1.U_0Edit.Text);
    U_1   := StrToFloat(Form1.U_1Edit.Text);
    t_end := StrToFloat(Form1.EndTimeEdit.Text)/1e9;
    except
      Begin
        ShowMessage('Постарайтесь корректно заполнить форматы!');
        Exit;
      end;
    end;
  end;

  //Initialization procedure.
  procedure InitAll(Sender: Tobject);
  var i: Longint;
  begin
    MRelock(Form1);
    for i := 1 to m do
      begin
      if i<=m1 then xn0[i]:=xno1;
      if i>=m2 then xn0[i]:=xno2;
      if (i>m1) and (i<m2) then xn0[i]:= xn0[i-1]-(xno1-xno2)/(m2-m1);
      end;
    for i := 1 to m do
      begin
        xn1[i]:=0.99*xn0[1];
```

```
        xn2[i]:=0.01*xn0[i];
        T1[i] := t0;
      end;
    xn1[1]:=xn1[2];
    xn2[1]:=xn2[2];
    t1[1] :=t1[2];
    t1[m] :=t1[m-1];

  end;

  procedure Puasson(Sender:Tobject; t: Extended);
  var i : LongInt;
  begin
    dz := Length/(m-1);
    e_t[1]:=0;
    for i := 2 to m do
      e_t[i]:= e_t[i-1] + el/(eps*eps0)*(xn1[i]+xn2[i]-xn0[i])*dz;
    sigma := 0;
    for i := 2 to m do
      sigma := sigma + dz*(e_t[i-1]+e_t[i])/2;
    for i := 1 to m do
      e[i] := e_t[i]+(ExternalU(t)-sigma)/Length;
  end;

  procedure TForm1.Button1Click(Sender: TObject);
  var i,j : LongInt;
      xm_e: Extended;
  begin
    Form1.Cursor:= crHourglass;
    InitAll(Self);
    UpdateSettings(Self);
    MRelock(Self);
    FreeAllSamples(Self);
    t:= 0;
    dz := Length/(m-1);
    InitAll(Self);
    //Need bound?
    nt:= t_end/ht;
    If nt >= 40000 then nt := 39990; //Why this?
    for j:= 1 to Round(nt) do
      begin
        UpdateProgressbar(Self,j,Round(nt));
        t := ht*(j-1);
        Puasson(Self, t);
        for i := 1 to m do
          t12[1]:= (t21/R)*exp(Delta/T1[i]);
        for i := 1 to m do
          begin
          xmu1[i]:=xmu10/(1+(xmu10*E[i]/vs1)*(xmu10*E[i]/vs1));
          xmu2[i]:=xmu20/(1+(xmu20*E[i]/vs2)*(xmu20*E[i]/vs2))
          end;
        for i := 1 to (m-1) do
          begin
          V1[i]:= -E[i]*xmu1[i]/(1-exp(-E[i]*xmu1[i]*dz/d1))*
            (xn1[i+1]*exp(-E[i]*xmu1[i]*dz/d1)-xn1[i]);
          //Possible source of error down here.
          V2[i]:= -E[i]*xmu2[i]/(1-exp(-E[i]*xmu2[i]*dz/d2))*
            (xn2[i+1]*exp(-E[i]*xmu2[i]*dz/d1)-xn2[i]);
          end;
          //End of possible source of error.
        for i := 2 to (m-1) do
          begin
          xm_e := V1[i]*2/(xn1[i+1]+xn1[i]);
          t1[i]:= t1[i]+(-xm_e*(t1[i+1]-t1[i-1])/(2*dz)+
            1.5*E[i]*E[i]*xmu1[i]-(t1[i]-t0)/te1)*ht;
          end;
        for i := 2 to m do
          begin
            xn1[i]:= xn1[i]+(-(V1[i]-V1[i-1])/dz-xn1[i]/(t12[i]+te1)+  //changed
              xn2[i]/t21)*ht;
            xn2[i]:= xn2[i]+(-(V2[i]-V2[i-1])/dz+xn1[i]/(t12[i]+te1)-
              xn2[i]/t21)*ht;
          end;

        for i:= 1 to m do sig := sig +(V1[i]+V2[i])*dz;
      end;
```

```
      For i :=1 to (m-1) do
        Cur[i]:= 1/Length*el*(xn1[i]+xn2[i])*(V1[i]+V2[i])*dz;
      Form1.Cursor := crDefault;
    ProgressBar1.Position:=0;
    MyDrawArray(Self);
  end;

  //Draw
  procedure MyDrawArray(Sender: Tobject);
  var i: Integer ;
      h1,h2,h3,h4,h5,h6,h7,h8: Extended;
  begin
    //Drawing everything onto PaintBox.
    h2:=0;
    For i := 1 to (m-1) do
    begin
      If xn0[i]>xn0[i+1] then h1:= xn0[i];
      h2 := h2 + E[i];
      If t12[i]>t12[i+1] then h3:= t12[i];
      If V1[i] >V1[i+1]  then h4:= V1[i];
      If V2[i] >V2[i+1]  then h5:= V2[i];
      If xmu1[i]>xmu1[i+1] then h6:= xmu1[i];
      If xmu2[i]>xmu2[i+1] then h7:= xmu2[i];
      If Cur[i] > Cur[i+1] then h8:= Cur[i];
    end;
    h2 := h2/m;
    With Form1.PaintBox1.Canvas do
      begin
        Pen.Color:=clBlue;
        Brush.Color:=clBlack;
        Rectangle(1,1,320,241);
        //Drawing xn0
        If Form1.Cxn0.checked = true then
        begin
        Pen.Color:= clRed;
        MoveTo(0,160);
        For I:=1 to m do LineTo(Round(320*i/m), 100-Round(7*xn0[i]/h1));
        end;
        //Drawing xmu1
        If Form1.Cxmu1.checked = true then
        begin
        Pen.Color:=clLime;
        Moveto(0,160);
        For I:=1 to m do LineTo(Round(320*i/m), 100-Round(7*xmu1[i]/h6));
        end;
        //Drawing xmu2
        If Form1.Cxmu2.checked = true then
        begin
        Pen.Color:=clGreen;
        Moveto(0,160);
        For I:=1 to m do LineTo(Round(320*i/m), 100-Round(7*xmu2[i]/h7));
        end;
        //Drawing V1
        If Form1.CV1.checked = true then
        begin
        Pen.Color:=clYellow;
        Moveto(0,160);
        For I:=1 to m do LineTo(Round(320*i/m), 100-Round(0.1*V1[i]/h4));
        end;
        //Drawing V2
        If Form1.CV2.checked = true then
        begin
        Pen.Color:=clFuchsia;
        Moveto(0,160);
        For I:=1 to m do LineTo(Round(320*i/m), 100-Round(0.2*V2[i]/h5));
        end;
        //Drawing E
        If Form1.CE.checked = true then
        begin
        Pen.Color:=clBlue;
        MoveTo(0,160);
        For I:=1 to m do LineTo(Round(320*i/m), 100-Round(3*E[i]/h2));
        end;
        //Drawing Current
        If Form1.CC.checked = true then
        begin
```

```
        Pen.Color:=clAqua;
        MoveTo(0,160);
        For I:=1 to m do LineTo(Round(320*i/m), 100-Round(3*Cur[i]/h8));
        end;
        //Drawing t12
        If Form1.Ct12.checked = true then
        begin
        Pen.Color:=clWhite;
        MoveTo(0,160);
        For I:=1 to m do LineTo(Round(320*i/m), 100-Round(3*t12[i]/h3));
        end;
        //Drawing zero Line
        Pen.Color:= clGray;
        Pen.Style:= psDash;
        MoveTo(0,160);
        LineTo(320,160);
        Brush.Color:= clLime;
        Pen.Style:= psSolid;
        TextOut(4,162,'0');
        TextOut(250,161,'Z=>');
    end;
end;

end.
```