

REZISTORLI BO‘LUVCHI UCHUN EHM DASTURINI YARATISH

Jurayev Saydulla Abdurahmonovich

TKTI katta o‘qituvchisi

Hammuallif talabalar:

Xudaynazarova Sabina Sherzod qizi;

Qo‘ziyev Shahboz Karim o‘g‘li;

Xuvaydullayeva Fotima Xabibulla qizi;

Abdurashidov Bekjon Shuxrat o‘g‘li.

ANNOTATSIYA

Ushbu maqolada elektron qurilmalarni loyihalashda keng qo‘llaniladigan rezistorli kuchlanish bo‘luvchilarni hisoblash jarayonini avtomatlashtirish masalasi ko‘rib chiqilgan. An‘anaviy hisob-kitoblardagi vaqt sarfi va sermehnatlikni kamaytirish maqsadida Delphi dasturlash muhitida maxsus EHM dasturi ishlab chiqilgan. Dasturiy ta‘minot tarkibidagi "case" tanlash kaliti yordamida Om va quvvat qonunlariga asoslangan algoritmlar, shuningdek, o‘quv jarayonida ushbu dasturdan foydalanishning dolzarbligi va afzalliklari bayon etilgan.

Kalit so‘zlar: rezistorli bo‘luvchi, kuchlanish, EHM dasturi, Delphi, algoritmi, Om qonuni, quvvat, avtomatlashtirish, ta‘lim jarayoni.

АННОТАЦИЯ

В данной статье рассматривается вопрос автоматизации процесса расчета резистивных делителей напряжения, широко используемых при проектировании электронных устройств. В целях сокращения трудозатрат и времени, затрачиваемого на традиционные ручные расчеты, в среде программирования Delphi разработана специальная программа для ЭВМ. Описаны алгоритмы, основанные на законе Ома и формуле мощности, реализованные с помощью оператора выбора "case", а также обоснована актуальность и преимущества внедрения данной программы в образовательный процесс.

Ключевые слова: резисторный делитель, напряжение, компьютерная программа, Дельфи, алгоритм, закон Ома, мощность, автоматизация, образовательный процесс.

ABSTRACT

This article addresses the issue of automating the calculation process for resistive voltage dividers, which are widely used in electronic device design. To reduce the time consumption and labor intensiveness of traditional manual calculations, a specialized computer program has been developed using the Delphi

programming environment. The algorithms based on Ohm's law and power formulas, implemented via the "case" selection statement, are described, and the relevance and advantages of utilizing this software in the educational process are highlighted.

Keywords: resistor divider, voltage, computer program, Delphi, algorithm, Ohm's law, power, automation, educational process.

KIRISH

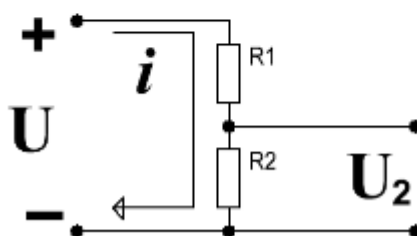
Elektron qurilmalar yaratishda kuchlanishni kamaytirish maqsadida rezistorli bo'luvchilardan foydalanish zarurati paydo bo'ladi. Har safar rezistorlarning qiymatini qog'ozga yozib hisoblash sermehnat ishidir. Bunday ishlar tadqiqotchilardan ancha vaqt sarflashni talab qiladi. Zamonaviy electron vositalardan xususan EHM dasturlaridan foydalanishning imkoni yaratilsa yuqoridagi sermehnatlik va vaqt sarflash tejaladi.

Shu yo'nalishda bir necha talabalar bilan suhbat o'tkazib ularning fikrlarini o'rganildi. Ta'lim jarayonida ushbu masalada tadqiqot o'tkazish uchun EHM uchun dastur yaratilsa va undan o'quv jarayonlarida foydalalanilsa qanday o'zgarishlar bo'lishi mumkinligi o'rganildi. Talabalar ushbu mavzuda dastur yaratilishi maqsadli ish bo'lishi ta'kidlab yuqorida aytilgan EHM dasturini yaratish dolzarb ekani suhbat yakunida aniqlandi. Ushbu xolatdan kelib chiqib R1 va R2 rezistorlari yordamida kuchlanishni bo'luvchi ekvivalent chizma ishlab chiqildi.(1-rasm.)

$$i = \frac{U}{R1+R2} \quad i = \frac{U_2}{R2}$$

$$R1 = R2 \left(\frac{U}{U_2} - 1 \right)$$

$$P_{rez} = i \cdot U$$



1 – rasm. Kuchlanishni bo'luvchi rezistorlarning ekvivalent chizmasi.

Ushbu chizmaga quvvat ($P=I \cdot U$) va Om ($I=U/R$) qonini formulalari ham qo'shib yozildi. Delphi dasturlash muhiti yordamida dasturi yaratildi. Dasturlashda "case"

tanlash kaliti yordamida protseduralar ishlab chiqildi. Bunda tegishli algoritmlar bajarilganda kalit tegishli shartga o'tadi va hisoblashni bajaradi:

```
r2:=R2rezistor.Position+R2rezistor01.Position/10;
Rezistor2.Caption:='R2 = '+FloatToStr(r2)+' Om ';
r1:= r2*( (u/u2)-1 );
Rezistor1.Caption:='R1 = '+Format('%.02f',[r1])+' Om ';
i:=u2/r2;
i:=i*1000;//---milli amper-----
if i>=1000 then Iumumiy.Caption:='i = '+Format('%.02f',[i/1000])+' A '
                else Iumumiy.Caption:='i = '+Format('%.02f',[i])+' mA ';
p:=i*u/1000;
Prezistor.Caption:='P = '+Format('%.04f',[p])+' W ';
p1:=i*u1/1000;
Prezistor1.Caption:='P1>= '+Format('%.04f',[p1])+' W ';
p2:=i*u2/1000;
Prezistor2.Caption:='P2>= '+Format('%.04f',[p2])+' W ';
```

Bu birinchi tanlov asosidagi proceduralarning bajarilishi. Endi ikkinchi tanlovni bayon qilamiz:

```
r2:=R2rezistor.Position+R2rezistor01.Position/10;
r2:=r2*1000;
Rezistor2.Caption:='R2 = '+FloatToStr(r2/1000)+' kOm ';
r1:= r2*( (u/u2)-1 );
Rezistor1.Caption:='R1 = '+Format('%.02f',[r1/1000])+' kOm ';
i:=u2/r2;
i:=i*1000000;//----mikro amper-----
if i>=1000 then Iumumiy.Caption:='i = '+Format('%.02f',[i/1000])+' mA '
                else Iumumiy.Caption:='i = '+Format('%.02f',[i])+' mkA ';
p:=i*u/1000000;
Prezistor.Caption:='P = '+Format('%.06f',[p])+' W ';
p1:=i*u1/1000000;
Prezistor1.Caption:='P1>= '+Format('%.06f',[p1])+' W ';
p2:=i*u2/1000000;
Prezistor2.Caption:='P2>= '+Format('%.06f',[p2])+' W ';
```

Bu ikkinchi tanlov asosidagi proceduralarning bajarilishi. Endi uchinchi tanlovni bayon qilamiz:

```

r2:=R2rezistor.Position+R2rezistor01.Position/10;
r2:=r2*1000000;
Rezistor2.Caption:='R2= '+FloatToStr(r2/1000000)+' MΩ';
r1:= r2*( (u/u2)-1 );
Rezistor1.Caption:='R1= '+Format('%.02f',[r1/1000000])+' MΩ';
i:=u2/r2;
i:=i*1000000000;//----nano amper-----
if i>=1000 then Iumumiy.Caption:='i= '+Format('%.02f',[i/1000])+' mA'
      else Iumumiy.Caption:='i= '+Format('%.02f',[i])+' nA';
p:=i*u/1000000000;
Prezistor.Caption:='P = '+Format('%.08f',[p])+' W ';
p1:=i*u1/1000000000;
Prezistor1.Caption:='P1>= '+Format('%.08f',[p1])+' W ';
p2:=i*u2/1000000000;
Prezistor2.Caption:='P2>= '+Format('%.08f',[p2])+' W ';

```

Ushbu dastur kodlari yuqoridagi chizma bo'yicha hisoblashlar amalga oshirilishiga xizmat qiladi. O'ziga hos algoritim asosida ishlaydigan dasturning to'liq kodini yoritamiz:

```
unit Rezistorni_xisoblash;
```

```
interface
```

```
uses
```

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

```
type
```

```
TForm1 = class(TForm)
  R2rezistor: TTrackBar;
  Rezistor1: TLabel;
  Image1: TImage;
  Rezistor2: TLabel;
  RadioButton1: TRadioButton;
  RadioButton2: TRadioButton;

```

```
RadioButton3: TRadioButton;
Ukirish1: TTrackBar;
U2chiqish1: TTrackBar;
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
Umanbaa: TEdit;
U2chiq: TEdit;
Ukirish01: TTrackBar;
U2chiqish01: TTrackBar;
Label6: TLabel;
Timer1: TTimer;
Prezistor: TLabel;
Iumumiy: TLabel;
R2rezistor01: TTrackBar;
Label1: TLabel;
Prezistor1: TLabel;
Prezistor2: TLabel;
Chiqish: TButton;
procedure Timer1Timer(Sender: TObject);
procedure U2chiqish1Change(Sender: TObject);
procedure Ukirish1Change(Sender: TObject);
procedure ChiqishClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
  procedure Xisoblash(b:byte);
end;

var
  Form1: TForm1;

implementation

{$R *.dfm}
```

```

procedure TForm1.Xisoblash(b:byte);
var
u,u1,u2:real;
r1,r2:real;
i:real;
p,p1,p2:real;

begin
u:=Ukirish1.Position+Ukirish01.Position/10;
if u=0 then Ukirish01.Position:=1;
u2:=U2chiqish1.Position+U2chiqish01.Position/100;
if u2=0 then U2chiqish01.Position:=1;
Umanbaa.Text:=FloatToStr(u)+' V';
U2chiq.Text:=FloatToStr(u2)+' V';
u1:=u-u2;
case b of
1: begin//-----
r2:=R2rezistor.Position+R2rezistor01.Position/10;
Rezistor2.Caption:='R2 = '+FloatToStr(r2)+' Om ';
r1:= r2*( (u/u2)-1 );
Rezistor1.Caption:='R1 = '+Format('%.02f',[r1])+' Om ';
i:=u2/r2;
i:=i*1000;//---milli amper-----
if i>=1000 then Iumumiy.Caption:='i = '+Format('%.02f',[i/1000])+' A '
else Iumumiy.Caption:='i = '+Format('%.02f',[i])+' mA ';
p:=i*u/1000;
Prezistor.Caption:='P = '+Format('%.04f',[p])+' W ';
p1:=i*u1/1000;
Prezistor1.Caption:='P1>= '+Format('%.04f',[p1])+' W ';
p2:=i*u2/1000;
Prezistor2.Caption:='P2>= '+Format('%.04f',[p2])+' W ';
end;//-----
2: begin//-----
r2:=R2rezistor.Position+R2rezistor01.Position/10;
r2:=r2*1000;
Rezistor2.Caption:='R2 = '+FloatToStr(r2/1000)+' kOm ';

```

```

r1:= r2*( (u/u2)-1 );
Rezistor1.Caption:='R1 = '+Format('%.02f',[r1/1000])+ ' kOm ';
i:=u2/r2;
i:=i*1000000;//----mikro amper-----
if i>=1000 then Iumumiy.Caption:='i = '+Format('%.02f',[i/1000])+ ' mA '
      else Iumumiy.Caption:='i = '+Format('%.02f',[i])+ ' mkA ';
p:=i*u/1000000;
Prezistor.Caption:='P = '+Format('%.06f',[p])+ ' W ';
p1:=i*u1/1000000;
Prezistor1.Caption:='P1>= '+Format('%.06f',[p1])+ ' W ';
p2:=i*u2/1000000;
Prezistor2.Caption:='P2>= '+Format('%.06f',[p2])+ ' W ';
end;//-----

```

3: begin//-----

```

r2:=R2rezistor.Position+R2rezistor01.Position/10;
r2:=r2*1000000;
Rezistor2.Caption:='R2= '+FloatToStr(r2/1000000)+' MOm';
r1:= r2*( (u/u2)-1 );
Rezistor1.Caption:='R1= '+Format('%.02f',[r1/1000000])+ ' MOm';
i:=u2/r2;
i:=i*1000000000;//----nano amper-----
if i>=1000 then Iumumiy.Caption:='i = '+Format('%.02f',[i/1000])+ ' mkA '
      else Iumumiy.Caption:='i = '+Format('%.02f',[i])+ ' nA';
p:=i*u/1000000000;
Prezistor.Caption:='P = '+Format('%.08f',[p])+ ' W ';
p1:=i*u1/1000000000;
Prezistor1.Caption:='P1>= '+Format('%.08f',[p1])+ ' W ';
p2:=i*u2/1000000000;
Prezistor2.Caption:='P2>= '+Format('%.08f',[p2])+ ' W ';
end;//-----

```

end;

end;

procedure TForm1.Timer1Timer(Sender: TObject);

begin

```
if Ukirish1.Position<=U2chiqish1.Position then
U2chiqish1.Position:=U2chiqish1.Position-1;
if RadioButton1.Checked then Xisoblash(1);
if RadioButton2.Checked then Xisoblash(2);
if RadioButton3.Checked then Xisoblash(3);
end;

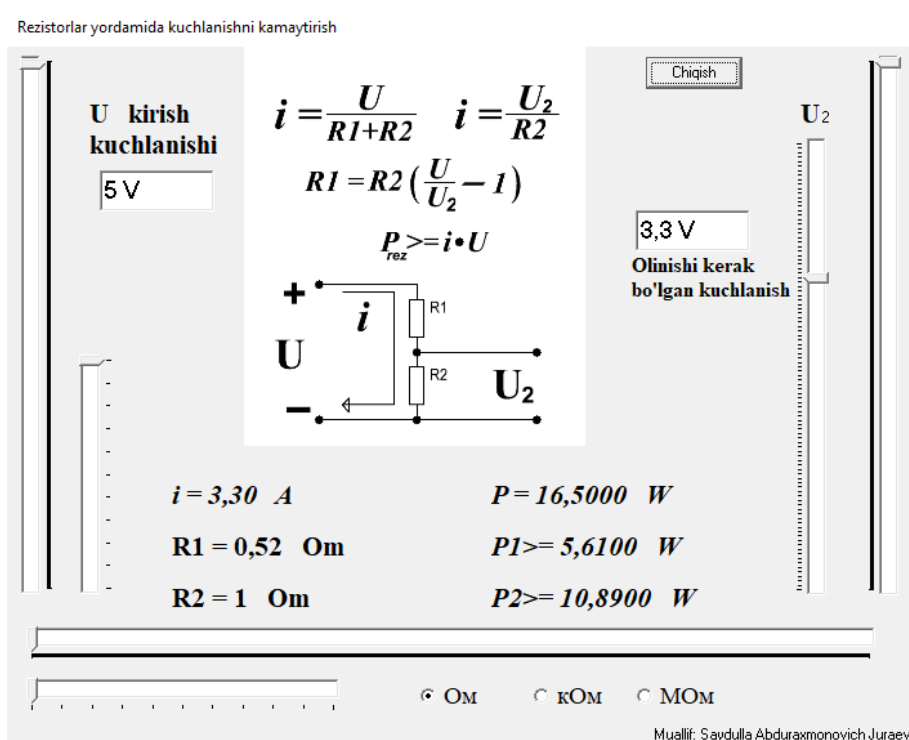
procedure TForm1.U2chiqish1Change(Sender: TObject);
begin
if Ukirish1.Position<=U2chiqish1.Position then
U2chiqish1.Position:=U2chiqish1.Position-1;
if U2chiqish1.Position=0 then U2chiqish01.Position:=1;
end;

procedure TForm1.Ukirish1Change(Sender: TObject);
begin
if Ukirish1.Position<=U2chiqish1.Position then
U2chiqish1.Position:=U2chiqish1.Position-1;
if Ukirish1.Position=0 then Ukirish01.Position:=1;
end;

procedure TForm1.ChiqishClick(Sender: TObject);
begin
Close
end;

end.
```

Delphi dasturlash muhiti yordamida yaratilgan ushbu dasturning kompilyatsiyalangan tayyor ishchi xolatidagi rasmi quyida keltiriladi. Ushbu dasturdan ta'lim jarayonlari yoki nuxandislar tamonidan foydalanish samarali bo'ladi.



XULOSA

O‘tkazilgan tadqiqotlar va dasturiy ishlanmalar natijasida quyidagi xulosalarga kelindi:Elektron qurilmalarni loyihalashda rezistorli kuchlanish bo‘luvchilarni qo‘lda hisoblash ko‘p vaqt va mehnat talab qilishi aniqlandi va ushbu jarayonni raqamlashtirish zarurati asoslandi.Delphi dasturlash muhitida yaratilgan EHM dasturi Om va quvvat qonunlariga tayangan holda qarshilik (Om, kOm, MOM), tok kuchi va quvvat yo‘qotishlarini yuqori aniqlikda, soniyalar ichida hisoblash imkonini berdi.Mazkur dasturiy ta‘minotning ta‘lim jarayoniga joriy etilishi talabalarning laboratoriya va mustaqil ishlarini bajarish samaradorligini oshiradi, vaqtni tejaydi hamde muhandislik hisob-kitoblaridagi inson omili bilan bog‘liq xatolarni bartaraf etadi.

FOYDALANILGAN ADABIYOTLAR

1. Karimov A.S. va b. Elektrotexnika va elektronika asoslari. Masalalar to‘plami va laboratoriya ishlari T. O‘qituvchi. 1991 y.
2. Charls Gross Fundamentals of Electrical Engineering 2012 by Taylor & Francis Group, LLC CRC Press is an imprint of Taylor & Francis Group, an Informa business.
3. GiorgioRizzoniPrinciples and Applications. McGraw-Hill Higher Education 2010g.996r.

4. Planck Collaboration Planck 2013 results. XVI. Cosmological parameters. — arXiv:1303.5076
5. P. Giridhar Kiniand Ramesh C. Bansal, Energy management systems. Published by InTech.JanezaTrdine 9, 51000 Rijeka, Croatia. Copyright © 2011 InTech. (p.61)
6. Gayibov T.SH. Energetika va energiya samaradorlik muammolari. O‘quv uslubiy majmua. TDTU qoshidagi pedagogik kadrlarni qayta tayyorlash va ularni malakasini oshirish tarmoq markazi 2016 y.
7. Infocom.uzelektron jurnali: www.infocom.uz
8. www.press-uz.info
9. www.ziyonet.uz