



服务器被攻击咋办？10大游戏公司都用这防御

送30G DDOS防御 · 可达1000G · 增强防御CC

立即购买

论坛 搜索

快捷导航

请输入搜索内容

帖子



论坛 > 技术区 > 电脑与单片机 > 精确测量二极管参数应用软件源程序

发帖

回复

返回列表

查看: 1764 | 回复: 9

精确测量二极管参数应用软件源程序 [复制链接]



发表于 2020-2-8 11:22:36 | 显示全部楼层

1# 电梯直达

本帖最后由 gxg0000 于 2020-2-8 11:20 编辑

闲来无事，写上一段用c#编写的程序盼望交流，实际应用见本坛《精确测量二极管RD的方法》一贴



gxg0000
20 主题 | 892 帖子 | 1万 积分

VIP会员



积分 13437

发消息

```
01. using System;
02. using System.Collections.Generic;
03. using System.ComponentModel;
04. using System.Data;
05. using System.Drawing;
06. using System.Text;
07. using System.Windows.Forms;
08. using System.IO;
09. using System.Runtime.InteropServices;
10. using Microsoft.Win32;
11.
12. namespace WindowsApplication1
13. {
14.     public partial class Form1 : Form
15.     {
16.         public Form1()
17.         {
18.             InitializeComponent();
19.         }
20.         Int32 V1, V2, V3, V4;
21.         Double Id1, Id2;
22.         Int32 Index; //
23.         Int64[] R = new Int64[7]; //限流电阻R1, R2, R3, R4, R5, R6, R7
24.         NumericUpDown[] numericUpDown = new NumericUpDown[7];
25.         String[] str1 = { "R1", "R2", "R3", "R4", "R5", "R6", "R7" }; //电阻的编号数组
26.         Decimal[] str2 = { 4700.0M, 1000.0M, 330.0M, 100.0M, 33.0M, 10.0M, 10000M }; //电阻初值数组
27.         Int32 T=25; //温度
28.         Double IS = 171.34388778833204; //饱和电流
29.         Double RD = 160.40886801322117; //零电阻
30.         Double N = 1.0693751430533303; //理想因子
31.         Double VT = 0.025679647254953728; //热电压
```

```

32.         Double[] set1 = { 1.0, 1.0, 1.0, 1.0 }; //IS, RD, N, VT
33.         Double[] set2 = { 1.0, 1.0, 1.0, 1.0 };
34.         Double[] set3 = { 1.0, 1.0, 1.0, 1.0 };
35.         Double[] set4 = { 1.0, 1.0, 1.0, 1.0 };
36.         Double[] set5 = { 1.0, 1.0, 1.0, 1.0 };
37.         Label[] lb1 = new Label[7];
38.         Label[] lb2 = new Label[7];
39.         Label[] lb3 = new Label[6];
40.         Bitmap tup1;
41.         Graphics g1;
42.         private void Form1_Load(object sender, EventArgs e)
43.         {
44.             this.listBox1.LostFocus += new System.EventHandler(this.listBox1_LostFocus); //添加失去焦点事件
45.             this.Controls.Add(panel1);
46.             panel1.BringToFront();
47.             panel1.Left = 522;
48.             panel1.Top = 270;
49.             panel1.Visible = false;
50.             for (int i = 0; i < 7; i++)
51.             {
52.                 numericUpDown[i] = new NumericUpDown();
53.                 numericUpDown[i].Cursor = Cursors.Hand;
54.                 numericUpDown[i].Width = 57;
55.                 numericUpDown[i].Height = 21;
56.                 numericUpDown[i].Top = 33 + 27 * i;
57.                 numericUpDown[i].Left = 102;
58.                 numericUpDown[i].Minimum = 0M;
59.                 if (i == 6)
60.                 {
61.                     numericUpDown[i].DecimalPlaces = 0;
62.                     numericUpDown[i].Maximum = 20000M;
63.                     numericUpDown[i].Increment = 1M;
64.                 }
65.                 else
66.                 {
67.                     numericUpDown[i].DecimalPlaces = 1;
68.                     numericUpDown[i].Maximum = 10000M;
69.                     numericUpDown[i].Increment = 0.1M;
70.                 }
71.                 this.panel1.Controls.Add(numericUpDown[i]); //添加到控件集合
72.                 numericUpDown[i].BringToFront();
73.             }
74.             //读取注册表保存的设置
75.             if (null == Registry.GetValue(@"HKEY_CURRENT_USER\Software\二极管参数计算\Settings", "R1", ""))
76.             {
77.                 //第1次创建注册表子项,并将初值数据写入内存和注册表
78.                 for (Int32 i = 0; i < 7; i++)
79.                 {
80.                     R[i] = Convert.ToInt64(str2[i]);
81.                     numericUpDown[i].Value = str2[i];
82.                     Registry.SetValue(@"HKEY_CURRENT_USER\Software\二极管参数计算\Settings", str1[i],
str2[i]);
83.                 }
84.             }
85.             else
86.             {
87.                 //读取注册表保存的子项内数据,并写入界面
88.                 for (Int32 i = 0; i < 7; i++)
89.                 {
90.                     numericUpDown[i].Value =
Convert.ToDecimal(Registry.GetValue(@"HKEY_CURRENT_USER\Software\二极管参数计算\Settings", str1[i], ""));
91.                 }
92.             }
93.             Index = 3;
94.             comboBox1.SelectedIndex = Index;

```

```

95.         label6.Text = "电阻" + str1[Index] + ":";
96.         listBox1.SelectedIndex = 1;
97.         listBox1.Visible = false;
98.         Drawing_Transfer_Characteristic(); //绘制曲线图形
99.         System.IO.Directory.CreateDirectory("C:\\Documents and Settings\\User\\My Documents\\二极管参数
表"); //创建保存数据的文件夹
100.     }
101.     /// <summary>
102.     /// 计算Is,N,Rd
103.     /// </summary>
104.     /// <param name="sender"></param>
105.     /// <param name="e"></param>
106.     private void button1_Click(object sender, EventArgs e)
107.     {
108.         V1 = Convert.ToInt32(1000 * numericUpDown1.Value);
109.         V2 = Convert.ToInt32(1000 * numericUpDown2.Value);
110.         V3 = Convert.ToInt32(1000 * numericUpDown3.Value);
111.         V4 = Convert.ToInt32(1000 * numericUpDown4.Value);
112.         for (Int32 i = 0; i < 7; i++)
113.         {
114.             R[i] = Convert.ToInt64(1000 * numericUpDown[i].Value);
115.         }
116.         for (Int32 i = 0; i < 6; i++)
117.         {
118.             R[i] = (R[i] * R[6]) / (R[i] + R[6]); //并联10M电阻后，修正DVM带来的误差
119.         }
120.         T = Convert.ToInt32(numericUpDown5.Value);
121.         VT = ((273 + T) * 1.380649 / 1.602177) / 10000; //计算热电压
122.         Id1 = 1000.0 * V2 / R[comboBox1.SelectedIndex] - 1000.0 * V1 / R[6]; //计算二极管电流（单位nA）
123.         Id2 = 1000.0 * V4 / R[comboBox1.SelectedIndex] - 1000.0 * V3 / R[6];
124.         Int32 n = 0;
125.         Double Is1=0, Is2=0;
126.         for (Int32 i = 0; i < 16; i++) //16位逐次逼近法
127.         {
128.             n |= (1 << (15 - i)); //预置转换位
129.             N = 0.5 + n / 21845.0; //N(0.5~3.5)
130.             Is1 = Id1 / (Math.Exp(V1 / (VT * 1000000 * N)) - 1);
131.             Is2 = Id2 / (Math.Exp(V3 / (VT * 1000000 * N)) - 1);
132.             if (Is1 > Is2) //
133.             {
134.                 n &= ~(1 << (15 - i)); //清零该位
135.                 N = 0.5 + n / 21845.0;
136.             }
137.         }
138.         if (Is1 > Is2)
139.         {
140.             if ((Is1 - Is2)/Is1 > 0.001)
141.                 MessageBox.Show("方程无解！"); //误差超过0.1%，表示方程无法解出
142.         }
143.         else
144.         {
145.             if ((Is2 - Is1) / Is2 > 0.001)
146.                 MessageBox.Show("方程无解！"); //显示错误
147.         }
148.         IS = (Is1 + Is2) / 2;
149.         RD=1000000 * (VT * N / IS);
150.         label26.Text = Convert.ToString(Convert.ToInt32(IS));
151.         label27.Text = Convert.ToString(Convert.ToInt32(RD));
152.         label28.Text = String.Format("{0:0.00}", N);
153.         label16.Text = String.Format("{0:样本数据1: Vd1=0.0 mV}",
Convert.ToDouble(numericUpDown1.Value));
154.         label38.Text = String.Format("{0:Id1=0 nA}", Id1);
155.         label17.Text = String.Format("{0:样本数据2: Vd2=0.0 mV}",
Convert.ToDouble(numericUpDown3.Value));
156.         label39.Text = String.Format("{0:Id2=0 nA}", Id2);

```

```

157.         label18.Text = String.Format("{0:热电压:    VT=0.0 mV}", VT * 1000);
158.         label29.Text = Convert.ToString(numericUpDown5.Value) + "°C";
159.         Drawing_Transfer_Characteristic(); //绘制转移特性曲线
160.     }
161.     /// <summary>
162.     /// 绘制二极管转移特性函数
163.     /// </summary>
164.     /// <param name="x">电压</param>
165.     /// <param name="y">电流</param>
166.     private void Drawing_Transfer_Characteristic()
167.     {
168.         tup1 = new Bitmap(201, 201);
169.         g1 = Graphics.FromImage(tup1);
170.         String[] str = new String[] { "正", "向", "电", "流", "IF", " " };
171.         //写垂直数字刻度
172.         for (int i = 0; i < 6; i++)
173.         {
174.             lb1[i] = new Label();
175.             lb1[i].BackColor = this.tabPage5.BackColor; //背景颜色(取底色)
176.             lb1[i].Width = 30;
177.             lb1[i].Height = 15;
178.             lb1[i].TextAlign = ContentAlignment.MiddleRight;
179.             if (i == 5)
180.                 lb1[i].Top = 20 + (40 * i - 3);
181.             else
182.                 lb1[i].Top = 20 + 40 * i;
183.             lb1[i].Left = pictureBox1.Left - lb1[i].Width;
184.             lb1[i].Font = new Font("宋体", 9, FontStyle.Regular);
185.             lb1[i].Text = Convert.ToString((5 - i) * Convert.ToInt32(numericUpDown6.Value) / 5);
186.             this.tabPage5.Controls.Add(lb1[i]); //添加到控件集合
187.             lb1[i].BringToFront();
188.         }
189.         for (int i = 0; i < 6; i++)
190.         {
191.             lb3[i] = new Label();
192.             lb3[i].BackColor = this.tabPage5.BackColor; //背景颜色(取底色)
193.             lb3[i].Width = 35;
194.             lb3[i].Height = 15;
195.             lb3[i].TextAlign = ContentAlignment.MiddleCenter;
196.             lb3[i].ForeColor = Color.Green;
197.             lb3[i].Top = 80 + 15 * i;
198.             lb3[i].Left = 0;
199.             lb3[i].Font = new Font("宋体", 9, FontStyle.Bold);
200.             if (i == 5)
201.                 lb3[i].Text = "(" + button2.Text.Substring(0, 2) + ")";
202.             else
203.                 lb3[i].Text = str[i];
204.             this.tabPage5.Controls.Add(lb3[i]); //添加到控件集合
205.             lb3[i].BringToFront();
206.         }
207.         //写水平数字刻度
208.         for (int i = 0; i < 7; i++)
209.         {
210.             lb2[i] = new Label();
211.             lb2[i].BackColor = this.tabPage5.BackColor; //背景颜色(取底色)
212.             if (i != 6)
213.             {
214.                 lb2[i].Width = 30;
215.                 lb2[i].Height = 15;
216.                 lb2[i].Top = 235;
217.                 lb2[i].Left = 47 + 40 * i;
218.                 lb2[i].Text = Convert.ToString(i * Convert.ToInt32(numericUpDown7.Value) / 5);
219.                 lb2[i].Font = new Font("宋体", 9, FontStyle.Regular);
220.                 lb2[i].ForeColor = Color.Black;
221.             }

```

```

222.         else
223.         {
224.             lb2[i].Width = 100;
225.             lb2[i].Height = 15;
226.             lb2[i].Top = 255;
227.             lb2[i].Left = 120;
228.             lb2[i].Text = "正向电压VF(mV)";
229.             lb2[i].Font = new Font("宋体", 9, FontStyle.Bold);
230.             lb2[i].ForeColor = Color.Green;
231.         }
232.         lb2[i].TextAlign = ContentAlignment.MiddleCenter;
233.         this.tabPage5.Controls.Add(lb2[i]); //添加到控件集合
234.         lb2[i].BringToFront();
235.     }
236.     g1.Clear(Color.AliceBlue);
237.     //横线
238.     for (int j = 0; j <= 200; j += 20)
239.     {
240.         g1.DrawLine(new Pen(Color.Gainsboro), 0, j, 200, j);
241.     }
242.     //竖线
243.     for (int j = 0; j <= 200; j += 20)
244.     {
245.         g1.DrawLine(new Pen(Color.Gainsboro), j, 0, j, 200);
246.     }
247.     if (checkBox1.Checked == true) //绘制叠加1
248.     {
249.         Draw_Line(set1[0], set1[2], set1[3], button5.BackColor);
250.     }
251.     if (checkBox2.Checked == true) //绘制叠加2
252.     {
253.         Draw_Line(set2[0], set2[2], set2[3], button6.BackColor);
254.     }
255.     if (checkBox3.Checked == true) //绘制叠加3
256.     {
257.         Draw_Line(set3[0], set3[2], set3[3], button7.BackColor);
258.     }
259.     if (checkBox4.Checked == true) //绘制叠加4
260.     {
261.         Draw_Line(set4[0], set4[2], set4[3], button8.BackColor);
262.     }
263.     if (checkBox5.Checked == true) //绘制叠加5
264.     {
265.         Draw_Line(set5[0], set5[2], set5[3], button9.BackColor);
266.     }
267.     Draw_Line(IS, N, VT, Color.Green); //绘制当前二极管伏安曲线
268.     pictureBox1.Image = tup1;
269. }
270. private void Draw_Line(Double Is, Double n, Double vt, Color color) //绘制二极管伏安线函数
271. {
272.     g1.DrawImage(tup1, 0, 0);
273.     Double I1, I2, Vd; ;
274.     Int32 k = Convert.ToInt32(Convert.ToString(listBox1.SelectedItem).Substring(4));
275.     for (int V = 0; V < 200; V++)
276.     {
277.         Vd = (V / 200.0) * Convert.ToUInt32(numericUpDown7.Value); //mV
278.         I1 = 200 * (Is * (Math.Exp(Vd / (1000 * vt * n)) - 1)) /
(Convert.ToInt32(numericUpDown6.Value) * k); //nA
279.         if (I1 > 200)
280.             I1 = 200;
281.         Vd = ((V + 1) / 200.0) * Convert.ToUInt32(numericUpDown7.Value); //mV
282.         I2 = 200 * (Is * (Math.Exp(Vd / (1000 * vt * n)) - 1)) /
(Convert.ToInt32(numericUpDown6.Value) * k); //nA
283.         if (I2 > 200)
284.             I2 = 200;

```

```

285.         g1.DrawLine(new Pen(color), (float)V, (float)(200 - I1), (float)(V + 1), (float)(200 -
I2));
286.         if ((I1 == 200) | (I2 == 200))
287.             break;
288.     }
289. }
290. /// <summary>
291. /// 绘图XY坐标单位调整
292. /// </summary>
293. /// <param name="sender"></param>
294. /// <param name="e"></param>
295. private void button2_Click(object sender, EventArgs e)
296. {
297.     listBox1.Visible = true; //显示电流单位列表
298.     listBox1.Focus(); //设置listBox1控件焦点
299. }
300. private void listBox1_LostFocus(object sender, EventArgs e)
301. {
302.     listBox1.Visible = false; //失去焦点隐藏
303. }
304. private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
305. {
306.     button2.Text = listBox1.Text;
307.     listBox1.Visible = false;
308.     Drawing_Transfer_Characteristic();
309. }
310. private void numericUpDown6_ValueChanged(object sender, EventArgs e)
311. {
312.     Drawing_Transfer_Characteristic();
313. }
314. private void numericUpDown7_ValueChanged(object sender, EventArgs e)
315. {
316.     Drawing_Transfer_Characteristic();
317. }
318.

```

[复制代码](#)

公开源代码;

补充内容 (2020-2-8 17:12):

谢谢老师们点赞!



[二极管参数计算.zip](#)

261.81 KB, 下载次数: 73

0 评分

2

[查看全部评分](#)



peiguoqing



老虎哥01

分享到: QQ好友和群

收藏

评分

顶

踩

回复

使用道具 举报

gxc0000



20 892 1万
主题 帖子 积分

VIP会员



积分 13437

发消息

楼主 | 发表于 2020-2-8 11:23:38 | 显示全部楼层

2#

```
01.  /// <summary>
02.      /// 保存二极管参数文件
03.  /// </summary>
04.  /// <param name="sender"></param>
05.  /// <param name="e"></param>
06.  private void button4_Click(object sender, EventArgs e)
07.  {
08.      saveFileDialog1.Filter = "文本文件 (*.txt) | *.txt | 全部文件 | *.*";
09.      saveFileDialog1.FilterIndex = 1; //指定第1个过滤器（默认的打开的方法）
10.      saveFileDialog1.InitialDirectory = "C:\\Documents and Settings\\User\\My Documents\\二极管参数
    表"; //打开起始目录
11.      if (saveFileDialog1.ShowDialog() == DialogResult.OK)
12.      {
13.          File.Delete(saveFileDialog1.FileName); //首先删除文件
14.          StreamWriter MyWriter = new StreamWriter(new FileStream(saveFileDialog1.FileName,
    FileMode.Append, FileAccess.Write)); //若无则创建新文件
15.          MyWriter.WriteLine(Convert.ToString(numericUpDown5.Value)); //写入温度
16.          MyWriter.WriteLine(Convert.ToString(IS)); //写入饱和电流IS
17.          MyWriter.WriteLine(Convert.ToString(RD)); //写入RD
18.          MyWriter.WriteLine(Convert.ToString(N)); //写入N
19.          MyWriter.WriteLine(Convert.ToString(VT)); //写入VT
20.          MyWriter.WriteLine("存放顺序第一行开始：温度T，饱和电流IS，零电阻RD，理想因子N，热电压VT");
    //写入注解
21.          MyWriter.Close(); //关闭文件
22.      }
23.  }
24.  /// <summary>
25.  /// 读取二极管数据文件到内存
26.  /// </summary>
27.  /// <param name="sender"></param>
28.  /// <param name="e"></param>
29.  private void button5_Click(object sender, EventArgs e)
30.  {
31.      OpenFileDialog(1);
32.  }
33.  private void button6_Click(object sender, EventArgs e)
34.  {
35.      OpenFileDialog(2);
36.  }
37.  private void button7_Click(object sender, EventArgs e)
38.  {
39.      OpenFileDialog(3);
40.  }
41.  private void button8_Click(object sender, EventArgs e)
42.  {
43.      OpenFileDialog(4);
44.  }
45.  private void button9_Click(object sender, EventArgs e)
```

```

46.         {
47.             OpenFileDialog(5);
48.         }
49.     private void OpenFileDialog(Int32 k)    //读数据文件到数组
50.     {
51.         openFileDialog1.Filter = "文本文件 (*.txt) |*.txt|全部文件|*.*";
52.         openFileDialog1.FilterIndex = 1; //指定第1个过滤器（默认的打开的方法）
53.         openFileDialog1.InitialDirectory = "C:\\Documents and Settings\\User\\My Documents\\二极管参数
表"; //打开起始目录
54.         if (openFileDialog1.ShowDialog() == DialogResult.OK)
55.         {
56.             StreamReader MyWriter = new StreamReader(new FileStream(openFileDialog1.FileName,
FileMode.OpenOrCreate, FileAccess.Read)); //若无则创建新文件
57.             String n0 = Path.GetFileNameWithoutExtension(openFileDialog1.FileName) + "(" +
MyWriter.ReadLine() + "°C)"; //读出文件名和温度
58.             Double n1 = Convert.ToDouble(MyWriter.ReadLine()); //读IS饱和电流
59.             Double n2 = Convert.ToDouble(MyWriter.ReadLine()); //读RD
60.             Double n3 = Convert.ToDouble(MyWriter.ReadLine()); //读N
61.             Double n4 = Convert.ToDouble(MyWriter.ReadLine()); //读VT
62.             MyWriter.Close(); //关闭文件
63.             if ((n1 == 0.0) | (n3 == 0.0) | (n4 == 0.0))
64.             {
65.                 MessageBox.Show("二极管数据文件格式不正确", "警告!");
66.             }
67.             else
68.             {
69.                 switch (k)
70.                 {
71.                     case 1:
72.                         checkBox1.Text = n0;
73.                         set1[0] = n1;
74.                         set1[1] = n2;
75.                         set1[2] = n3;
76.                         set1[3] = n4;
77.                         checkBox1.Enabled = true;
78.                         break;
79.                     case 2:
80.                         checkBox2.Text = n0;
81.                         set2[0] = n1;
82.                         set2[1] = n2;
83.                         set2[2] = n3;
84.                         set2[3] = n4;
85.                         checkBox2.Enabled = true;
86.                         break;
87.                     case 3:
88.                         checkBox3.Text = n0;
89.                         set3[0] = n1;
90.                         set3[1] = n2;
91.                         set3[2] = n3;
92.                         set3[3] = n4;
93.                         checkBox3.Enabled = true;
94.                         break;
95.                     case 4:
96.                         checkBox4.Text = n0;
97.                         set4[0] = n1;
98.                         set4[1] = n2;
99.                         set4[2] = n3;
100.                        set4[3] = n4;
101.                        checkBox4.Enabled = true;
102.                        break;
103.                     case 5:
104.                         checkBox5.Text = n0;
105.                         set5[0] = n1;
106.                         set5[1] = n2;
107.                         set5[2] = n3;

```



```

108.                set5[3] = n4;
109.                checkBox5.Enabled = true;
110.                break;
111.            }
112.        }
113.    }
114.}
115./// <summary>
116./// 复选框叠加绘图
117./// </summary>
118./// <param name="sender"></param>
119./// <param name="e"></param>
120.private void checkBox1_CheckedChanged(object sender, EventArgs e)
121.{
122.    Drawing_Transfer_Characteristic();
123.    if (checkBox1.Checked == true)
124.        radioButton3.Enabled = true;
125.    else
126.    {
127.        radioButton3.Enabled = false;
128.        if (radioButton3.Checked == true)
129.            radioButton8.Checked = true;
130.    }
131.}
132.private void checkBox2_CheckedChanged(object sender, EventArgs e)
133.{
134.    Drawing_Transfer_Characteristic();
135.    if (checkBox2.Checked == true)
136.        radioButton4.Enabled = true;
137.    else
138.    {
139.        radioButton4.Enabled = false;
140.        if (radioButton4.Checked == true)
141.            radioButton8.Checked = true;
142.    }
143.}
144.private void checkBox3_CheckedChanged(object sender, EventArgs e)
145.{
146.    Drawing_Transfer_Characteristic();
147.    if (checkBox3.Checked == true)
148.        radioButton5.Enabled = true;
149.    else
150.    {
151.        radioButton5.Enabled = false;
152.        if (radioButton5.Checked == true)
153.            radioButton8.Checked = true;
154.    }
155.}
156.private void checkBox4_CheckedChanged(object sender, EventArgs e)
157.{
158.    Drawing_Transfer_Characteristic();
159.    if (checkBox4.Checked == true)
160.        radioButton6.Enabled = true;
161.    else
162.    {
163.        radioButton6.Enabled = false;
164.        if (radioButton6.Checked == true)
165.            radioButton8.Checked = true;
166.    }
167.}
168.private void checkBox5_CheckedChanged(object sender, EventArgs e)
169.{
170.    Drawing_Transfer_Characteristic();
171.    if (checkBox5.Checked == true)
172.        radioButton7.Enabled = true;

```

```

173.         else
174.         {
175.             radioButton7.Enabled = false;
176.             if (radioButton7.Checked == true)
177.                 radioButton8.Checked = true;
178.         }
179.     }
180.     /// <summary>
181.     /// 鼠标指针即时Rd显示数据
182.     /// </summary>
183.     /// <param name="sender"></param>
184.     /// <param name="e"></param>
185.     private void pictureBox1_MouseMove(object sender, MouseEventArgs e)
186.     {
187.         Int32 Rd=0;
188.         Double Is,n,v;
189.         Bitmap tup2 = new Bitmap(201, 201);
190.         Graphics g2 = Graphics.FromImage(tup2);
191.         Pen pen;
192.         Brush brush1;
193.         Brush brush2 = new SolidBrush(pictureBox1.BackColor);
194.         //判断跟踪曲线
195.         if (radioButton3.Enabled & radioButton3.Checked)
196.         {
197.             pen = new Pen(button5.BackColor);
198.             brush1 = new SolidBrush(button5.BackColor);
199.             Is = set1[0];
200.             n = set1[2];
201.             vt = set1[3];
202.         }
203.         else if (radioButton4.Enabled & radioButton4.Checked)
204.         {
205.             pen = new Pen(button6.BackColor);
206.             brush1 = new SolidBrush(button6.BackColor);
207.             Is = set2[0];
208.             n = set2[2];
209.             vt = set2[3];
210.         }
211.         else if (radioButton5.Enabled & radioButton5.Checked)
212.         {
213.             pen = new Pen(button7.BackColor);
214.             brush1 = new SolidBrush(button7.BackColor);
215.             Is = set3[0];
216.             n = set3[2];
217.             vt = set3[3];
218.         }
219.         else if (radioButton6.Enabled & radioButton6.Checked)
220.         {
221.             pen = new Pen(button8.BackColor);
222.             brush1 = new SolidBrush(button8.BackColor);
223.             Is = set4[0];
224.             n = set4[2];
225.             vt = set4[3];
226.         }
227.         else if (radioButton7.Enabled & radioButton7.Checked)
228.         {
229.             pen = new Pen(button9.BackColor);
230.             brush1 = new SolidBrush(button9.BackColor);
231.             Is = set5[0];
232.             n = set5[2];
233.             vt = set5[3];
234.         }
235.         else
236.         {
237.             pen = new Pen(Color.Green);

```

```

238.         brush1 = new SolidBrush(Color.Green);
239.         Is = IS;
240.         n = N;
241.         vt = VT;
242.     }
243.     Double Vd=(e.X / 200.0) * Convert.ToInt32(numericUpDown7.Value);    //正向电压mV
244.     Double Id = Is * (Math.Exp(Vd / (1000 * vt * n)) - 1); //正向电压对于的正向电流nA
245.     Int32 k = Convert.ToInt32(Convert.ToString(listBox1.SelectedItem).Substring(4)); //纵坐标单位
246.     Int32 I = (int)(200 * (Id / (Convert.ToInt32(numericUpDown6.Value) * k))); //纵坐标
247.     if (I > 200) I = 200;
248.     if (Id == 0.0) Id = 1;
249.     //鼠标靠近曲线悬停字符串显示
250.     if (((e.Y - (200 - I)) < 5)&((200 - I) - e.Y) < 5))
251.     {
252.         Rd = (Convert.ToInt32(1000000000 * ((n * vt / Id) * Math.Log((Id / Is) + 1, Math.E))));
//计算动态电阻Rd
253.         String str;
254.         if (Rd < 1000)
255.             str = "Rd=" + Rd.ToString() + "Ω";
256.         else
257.         {
258.             Rd = Rd / 1000;
259.             str = "Rd=" + Rd.ToString() + "kΩ";
260.         }
261.         //绘制悬停字符串
262.         g2.DrawImage(tup1, 0, 0); //复制
263.         int x1 = 0;
264.         int y1 = 0;
265.         SizeF sizeF = g2.MeasureString(str, new Font("宋体", 9, FontStyle.Regular)); //测量字符串长度
266.         if (e.X > (int)sizeF.Width) //区分在鼠标左或右显示字符串
267.             x1 = e.X - (int)sizeF.Width;
268.         else
269.             x1 = e.X + 5;
270.         if (e.Y < (int)sizeF.Height) //区分在鼠标上或下显示字符串
271.             y1 = e.Y;
272.         else
273.             y1 = e.Y - (int)sizeF.Height;
274.         g2.DrawString(str, new Font("宋体", 9, FontStyle.Regular), brush1, x1, y1);
275.         g2.FillEllipse(brush2, e.X - 2, (200 - I) - 2, 4, 4); //绘制实心圆
276.         g2.DrawEllipse(pen, e.X - 2, (200 - I) - 2, 4, 4); //绘制圆圈
277.
278.         pictureBox1.Image = tup2;
279.     }
280.     else
281.     {
282.         pictureBox1.Image = tup1; //回复原图
283.     }
284. }
285. private void pictureBox1_MouseLeave(object sender, EventArgs e)
286. {
287.     pictureBox1.Image = tup1;
288. }
289. private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
290. {
291.     if (comboBox1.Text == "校电阻")
292.     {
293.         panel1.Visible = true;
294.         panel1.Focus();
295.         for (Int32 i = 0; i < 7; i++)
296.         {
297.             numericUpDown[i].Value =
Convert.ToDecimal(Registry.GetValue(@"HKEY_CURRENT_USER\Software\二极管参数计算\Settings", str1[i], ""));
298.         }
299.         groupBox1.Enabled = false;
300.     }

```

```
301.         else
302.         {
303.             Index = comboBox1.SelectedIndex;
304.             label6.Text = "电阻" + str1[Index] + ":";
305.         }
306.     }
307.     /// <summary>
308.     /// 保存退出
309.     /// </summary>
310.     /// <param name="sender"></param>
311.     /// <param name="e"></param>
312.     private void button10_Click(object sender, EventArgs e)
313.     {
314.         //数据写入注册表子项
315.         for (Int32 i = 0; i < 7; i++)
316.         {
317.             Registry.SetValue(@"HKEY_CURRENT_USER\Software\二极管参数计算\Settings", str1[i],
numericUpDown[i].Value);
318.         }
319.         panel1.Visible = false;
320.         groupBox1.Enabled = true;
321.         comboBox1.SelectedIndex = Index;
322.         label6.Text = "电阻" + str1[Index] + ":"; //恢复原来的comboBox1选项
323.     }
324.     /// <summary>
325.     /// 放弃
326.     /// </summary>
327.     /// <param name="sender"></param>
328.     /// <param name="e"></param>
329.     private void button11_Click(object sender, EventArgs e)
330.     {
331.         panel1.Visible = false;
332.         groupBox1.Enabled = true;
333.         comboBox1.SelectedIndex = Index;
334.         label6.Text = "电阻" + str1[Index] + ":";
335.     }
336. }
337. }
```

[复制代码](#)

完

○ 评分

2



[查看全部评分](#)

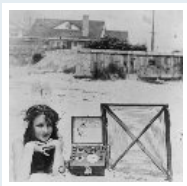
peiguoqing

老虎哥01

[回复](#) [支持](#) [反对](#)

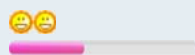
[使用道具](#) [评分](#) [举报](#)

gxg0000



20 892 1万
主题 帖子 积分

VIP会员



积分 13437

发消息

楼主 | 发表于 2020-2-8 17:09:52 | 显示全部楼层

3#

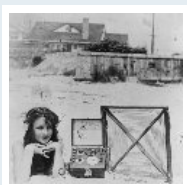
“ 分立元件 发表于 2020-2-8 13:04
下位机是使用单片机测量吗？然后传到电脑上处理数据？ ”

独立的软件，与测试装置不产生通信👉

回复 支持 反对

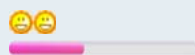
使用道具 评分 举报

gxg0000



20 892 1万
主题 帖子 积分

VIP会员



积分 13437

发消息

楼主 | 发表于 2020-2-9 11:30:20 | 显示全部楼层

4#

“ peiguqing 发表于 2020-2-9 10:33
lz的编程水平很高，借机会学习啦，谢谢！ ”

业余的，与.net高手比起来漏洞百出👉

回复 支持 反对

使用道具 评分 举报

gxg0000



20
主题

892
帖子

1万
积分

VIP会员

积分

13437

发消息

楼主 | 发表于 2020-5-5 16:42:34 | 显示全部楼层

5#

“ drzhu 发表于 2020-3-23 09:58
好厉害！我也想学学c编程，就是贪玩每次都找理由干其他事了。lz值得我学习！ ”

有基础的学起来比较容易，但学会都是一些皮毛的东西，精通则需要长年累月😁

回复

支持

反对

使用道具



评分

举报

发帖

回复








B A   “ ” < > 😊


发表回复




☐ 回帖后跳转到最后一页

高级模式









本版积分规则

Powered by **Discuz!** X3.4
© 2004-2019 caoyin.com

小黑屋 | 手机版 | 矿石收音机 (蒙ICP备05000029号-1)



蒙公网安备 15040402000005号

GMT+8, 2020-7-4 09:18