```
1  using Microsoft.SPOT;
2
3  //Nowendig für Visual Studio 2015
4  namespace System.Diagnostics
5  {
6      public enum DebuggerBrowsableState
7      {
8          Never,
9          Collapsed,
10         RootHidden
11     }
12 }
13
14 class Program
15 {
16     private BrainPad.Image imageY = new BrainPad.Image(8, 8);
17     private BrainPad.Image imageG = new BrainPad.Image(8, 8);
18     private BrainPad.Image imageR = new BrainPad.Image(8, 8);
19     private BrainPad.Image imageS = new BrainPad.Image(26, 26);
20     private BrainPad.Image imageZ04 = new BrainPad.Image(26, 26);
21     private BrainPad.Image imageZ05 = new BrainPad.Image(26, 26);
22     private BrainPad.Image imageZ06 = new BrainPad.Image(26, 26);
23     private BrainPad.Image imageZ07 = new BrainPad.Image(26, 26);
24     private BrainPad.Image imageZ08 = new BrainPad.Image(26, 26);
25     private BrainPad.Image imageZ09 = new BrainPad.Image(26, 26);
26     private BrainPad.Image imageZ10 = new BrainPad.Image(26, 26);
27
28     private int position;
29
30     public void BrainPadSetup()
31     {
32         getImage(imageY,
            BrainPad_Projekt_08B.Properties.Resources.BinaryResources.Gelb);
33         getImage(imageG,
            BrainPad_Projekt_08B.Properties.Resources.BinaryResources.Gruen);
34         getImage(imageR,
            BrainPad_Projekt_08B.Properties.Resources.BinaryResources.Rot);
35         getImage(imageS,
            BrainPad_Projekt_08B.Properties.Resources.BinaryResources.Schwarz);
36         getImage(imageZ04,
            BrainPad_Projekt_08B.Properties.Resources.BinaryResources.Zeiger_04);
37         getImage(imageZ05,
            BrainPad_Projekt_08B.Properties.Resources.BinaryResources.Zeiger_05);
38         getImage(imageZ06,
            BrainPad_Projekt_08B.Properties.Resources.BinaryResources.Zeiger_06);
39         getImage(imageZ07,
            BrainPad_Projekt_08B.Properties.Resources.BinaryResources.Zeiger_07);
40         getImage(imageZ08,
            BrainPad_Projekt_08B.Properties.Resources.BinaryResources.Zeiger_08);
41         getImage(imageZ09,
            BrainPad_Projekt_08B.Properties.Resources.BinaryResources.Zeiger_09);
42         getImage(imageZ10,
            BrainPad_Projekt_08B.Properties.Resources.BinaryResources.Zeiger_10);
43
44         BrainPad.Button.ButtonPressed += Button_ButtonPressed;
45         position = 4;
```

```
46          rotation();
47      }
48
49      public void BrainPadLoop()
50      {
51      }
52
53      private void Button_ButtonPressed(BrainPad.Button.DPad button,
         BrainPad.Button.State state)
54      {
55          if (button == BrainPad.Button.DPad.Right)
56          {
57              if (state == BrainPad.Button.State.Pressed)
58              {
59                  if (position < 10)
60                  {
61                      position++;
62                      rotation();
63                  }
64                  else
65                      position = 10;
66              }
67          }
68          if (button == BrainPad.Button.DPad.Left)
69          {
70              if (position > 4)
71              {
72                  position --;
73                  rotation();
74              }
75              else
76                  position = 4;
77          }
78          if (button == BrainPad.Button.DPad.Down)
79          {
80              position = 4;
81              rotation();
82          }
83      }
84
85      private void getImage(BrainPad.Image image,
         BrainPad_Projekt_08B.Properties.Resources.BinaryResources resource)
86      {
87          int info;
88          byte[] data;
89          BrainPad.Color color = new BrainPad.Color();
90
91          data = (byte[])ResourceUtility.GetObject
             (BrainPad_Projekt_08B.Properties.Resources.ResourceManager,
             resource);
92          info = data[0xA];
93          for (int Y = image.Height - 1; Y >= 0; Y--)
94          {
95              for (int X = 0; X < image.Width; X++)
96              {
97                  byte h = data[info++];
```

```
 98                 byte n = data[info++];
 99                 color.B = (byte)(h & 0x1f);
100                 color.G = (byte)(((h & 0xe0) >> 5) + ((n & 0x07) << 3));
101                 color.R = (byte)((n & 0xf8) >> 3);
102                 image.SetPixel(Y, X, color);
103             }
104         }
105     }
106
107     private void drawPoints()
108     {
109         BrainPad.Display.DrawImage(62, 46, imageS);
110         BrainPad.Display.DrawImage(72, 46, imageS);
111         BrainPad.Display.DrawImage(51, 60, imageY);
112         BrainPad.Display.DrawImage(54, 47, imageY);
113         BrainPad.Display.DrawImage(63, 38, imageG);
114         BrainPad.Display.DrawImage(76, 35, imageG);
115         BrainPad.Display.DrawImage(88, 38, imageG);
116         BrainPad.Display.DrawImage(98, 47, imageR);
117         BrainPad.Display.DrawImage(101, 60, imageR);
118     }
119
120     private void drawPointer()
121     {
122         drawPoints();
123         switch (position)
124         {
125             case 4:
126                 BrainPad.Display.DrawImage(62, 46, imageZ04);
127                 break;
128             case 5:
129                 BrainPad.Display.DrawImage(62, 46, imageZ05);
130                 break;
131             case 6:
132                 BrainPad.Display.DrawImage(62, 46, imageZ06);
133                 break;
134             case 7:
135                 BrainPad.Display.DrawImage(72, 46, imageZ07);
136                 break;
137             case 8:
138                 BrainPad.Display.DrawImage(72, 46, imageZ08);
139                 break;
140             case 9:
141                 BrainPad.Display.DrawImage(72, 46, imageZ09);
142                 break;
143             case 10:
144                 BrainPad.Display.DrawImage(72, 46, imageZ10);
145                 break;
146         }
147         BrainPad.Display.DrawText(51, 80, "Speed: " + ((double)
             position*0.1).ToString("F1"), BrainPad.Color.White);
148     }
149
150     private void rotation()
151     {
152         drawPointer();
```

```
153             BrainPad.DcMotor.SetSpeed((double)position * 0.1);
154         }
155     }
156
```