```
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 imageW = new BrainPad.Image(8, 8);
17     private BrainPad.Image imageF = new BrainPad.Image(8, 8);
18     private BrainPad.Image imageS = new BrainPad.Image(26, 26);
19     private BrainPad.Image imageZ000 = new BrainPad.Image(26, 26);
20     private BrainPad.Image imageZ030 = new BrainPad.Image(26, 26);
21     private BrainPad.Image imageZ060 = new BrainPad.Image(26, 26);
22     private BrainPad.Image imageZ090 = new BrainPad.Image(26, 26);
23     private BrainPad.Image imageZ120 = new BrainPad.Image(26, 26);
24     private BrainPad.Image imageZ150 = new BrainPad.Image(26, 26);
25     private BrainPad.Image imageZ180 = new BrainPad.Image(26, 26);
26
27     private int position;
28
29     public void BrainPadSetup()
30     {
31         getImage(imageW,
            BrainPad_Projekt_08A.Properties.Resources.BinaryResources.Kreis_W);
32         getImage(imageF,
            BrainPad_Projekt_08A.Properties.Resources.BinaryResources.Kreis_F);
33         getImage(imageS,
            BrainPad_Projekt_08A.Properties.Resources.BinaryResources.Schwarz);
34         getImage(imageZ000,
            BrainPad_Projekt_08A.Properties.Resources.BinaryResources.Zeiger_000)
            ;
35         getImage(imageZ030,
            BrainPad_Projekt_08A.Properties.Resources.BinaryResources.Zeiger_030)
            ;
36         getImage(imageZ060,
            BrainPad_Projekt_08A.Properties.Resources.BinaryResources.Zeiger_060)
            ;
37         getImage(imageZ090,
            BrainPad_Projekt_08A.Properties.Resources.BinaryResources.Zeiger_090)
            ;
38         getImage(imageZ120,
            BrainPad_Projekt_08A.Properties.Resources.BinaryResources.Zeiger_120)
            ;
39         getImage(imageZ150,
            BrainPad_Projekt_08A.Properties.Resources.BinaryResources.Zeiger_150)
            ;
40         getImage(imageZ180,
            BrainPad_Projekt_08A.Properties.Resources.BinaryResources.Zeiger_180)
```

```
            ;
41
42          BrainPad.Button.ButtonPressed += Button_ButtonPressed;
43          position = 0;
44          rotation();
45      }
46
47      public void BrainPadLoop()
48      {
49      }
50
51      private void Button_ButtonPressed(BrainPad.Button.DPad button,
         BrainPad.Button.State state)
52      {
53          if (button == BrainPad.Button.DPad.Right)
54          {
55              if (state == BrainPad.Button.State.Pressed)
56              {
57                  if (position > 0)
58                  {
59                      position -= 30;
60                      rotation();
61                  }
62                  else
63                      position = 0;
64              }
65          }
66          if (button == BrainPad.Button.DPad.Left)
67          {
68              if (position < 180)
69              {
70                  position += 30;
71                  rotation();
72              }
73              else
74                  position = 180;
75          }
76          if (button == BrainPad.Button.DPad.Down)
77          {
78              if (position != 0)
79              {
80                  position = 0;
81                  rotation();
82              }
83          }
84      }
85
86      private void getImage(BrainPad.Image image,
         BrainPad_Projekt_08A.Properties.Resources.BinaryResources resource)
87      {
88          int info;
89          byte[] data;
90          BrainPad.Color color = new BrainPad.Color();
91
92          data = (byte[])ResourceUtility.GetObject
             (BrainPad_Projekt_08A.Properties.Resources.ResourceManager,
```

```
            resource);
93          info = data[0xA];
94          for (int Y = image.Height - 1; Y >= 0; Y--)
95          {
96              for (int X = 0; X < image.Width; X++)
97              {
98                  byte h = data[info++];
99                  byte n = data[info++];
100                 color.B = (byte)(h & 0x1f);
101                 color.G = (byte)(((h & 0xe0) >> 5) + ((n & 0x07) << 3));
102                 color.R = (byte)((n & 0xf8) >> 3);
103                 image.SetPixel(Y, X, color);
104             }
105         }
106     }
107
108     private void drawPoints()
109     {
110         BrainPad.Display.DrawImage(62, 46, imageS);
111         BrainPad.Display.DrawImage(72, 46, imageS);
112         BrainPad.Display.DrawImage(51, 60, imageF);
113         BrainPad.Display.DrawImage(54, 47, imageF);
114         BrainPad.Display.DrawImage(63, 38, imageF);
115         BrainPad.Display.DrawImage(76, 35, imageF);
116         BrainPad.Display.DrawImage(88, 38, imageF);
117         BrainPad.Display.DrawImage(98, 47, imageF);
118         BrainPad.Display.DrawImage(101, 60, imageF);
119     }
120
121     private void drawPointer()
122     {
123         drawPoints();
124         switch (position)
125         {
126             case 0:
127                 BrainPad.Display.DrawImage(101, 60, imageW);
128                 BrainPad.Display.DrawImage(72, 46, imageZ000);
129                 break;
130             case 30:
131                 BrainPad.Display.DrawImage(98, 47, imageW);
132                 BrainPad.Display.DrawImage(72, 46, imageZ030);
133                 break;
134             case 60:
135                 BrainPad.Display.DrawImage(88, 38, imageW);
136                 BrainPad.Display.DrawImage(72, 46, imageZ060);
137                 break;
138             case 90:
139                 BrainPad.Display.DrawImage(76, 35, imageW);
140                 BrainPad.Display.DrawImage(72, 46, imageZ090);
141                 break;
142             case 120:
143                 BrainPad.Display.DrawImage(63, 38, imageW);
144                 BrainPad.Display.DrawImage(62, 46, imageZ120);
145                 break;
146             case 150:
147                 BrainPad.Display.DrawImage(54, 47, imageW);
```

```
148                 BrainPad.Display.DrawImage(62, 46, imageZ150);
149                 break;
150             case 180:
151                 BrainPad.Display.DrawImage(51, 60, imageW);
152                 BrainPad.Display.DrawImage(62, 46, imageZ180);
153                 break;
154         }
155         BrainPad.Display.DrawText(27, 80, "Rotation: " + position.ToString
             ("D3") + " Grad", BrainPad.Color.White);
156     }
157
158     private void rotation()
159     {
160         drawPointer();
161         BrainPad.ServoMotor.SetPosition(position);
162         BrainPad.Wait.Seconds(0.5);
163     }
164 }
165
```