

```
1 using Microsoft.SPOT;
2 using Microsoft.SPOT.Hardware;
3
4 //Nowendig für Visual Studio 2015
5 namespace System.Diagnostics
6 {
7     public enum DebuggerBrowsableState
8     {
9         Never,
10        Collapsed,
11        RootHidden
12    }
13 }
14
15 class Program
16 {
17     const int XM = 80;
18     const int YM = 103;
19     private AnalogInput analogIn;
20     private double analogValue;
21     private BrainPad.Image imageM1 = new BrainPad.Image(50, 50);
22     private BrainPad.Image imageM2 = new BrainPad.Image(50, 50);
23     private int x0, y0, x1;
24     private int x0Old, y0Old, x1Old;
25
26     public void BrainPadSetup()
27     {
28         analogIn = new AnalogInput(BrainPad.Expansion.AnalogInput.PA7);
29         getImage(imageM1, BrainPad\_Projekt\_06A.Properties.Resources.BinaryResources.M1);
30         getImage(imageM2, BrainPad\_Projekt\_06A.Properties.Resources.BinaryResources.M2);
31
32         BrainPad.Display.DrawText(38, 8, "Analog-Input",
33             BrainPad.Color.White);
34         BrainPad.Display.DrawImage(XM - 50, YM - 78, imageM1);
35         BrainPad.Display.DrawImage(XM, YM - 78, imageM2);
36         BrainPad.Display.DrawLargeText(75, 85, "V", BrainPad.Color.White);
37         linePoints();
38
39     public void BrainPadLoop()
40     {
41         analogValue = analogIn.Read()*3.3;
42         showPointer();
43         BrainPad.Wait.Seconds(1.0);
44     }
45
46     private void getImage(BrainPad.Image image, BrainPad\_Projekt\_06A.Properties.Resources.BinaryResources resource)
47     {
48         int info;
49         byte[] data;
50         BrainPad.Color color = new BrainPad.Color();
51
52         data = (byte[])ResourceUtility.GetObject
```

```
(BrainPad_Projekt_06A.Properties.Resources.ResourceManager,
resource);
53 info = data[0xA];
54 for (int Y = image.Height - 1; Y >= 0; Y--)
55 {
56     for (int X = 0; X < image.Width; X++)
57     {
58         byte h = data[info++];
59         byte n = data[info++];
60         color.B = (byte)(h & 0x1f);
61         color.G = (byte)(((h & 0xe0) >> 5) + ((n & 0x07) << 3));
62         color.R = (byte)((n & 0xf8) >> 3);
63         image.SetPixel(Y, X, color);
64     }
65 }
66 }
67
68 private void showPointer()
69 {
70
71     x0Old = x0;
72     y0Old = y0;
73     x1Old = x1;
74     BrainPad.Display.DrawLine(x0Old, y0Old, x1Old, YM - 29, BrainPad.Color.Black);
75     BrainPad.Display.DrawText(63, 110, analogValue.ToString("F2") + " V", BrainPad.Color.Red);
76     linePoints();
77     BrainPad.Display.DrawLine(x0, y0, x1, YM - 29, BrainPad.Color.Red);
78 }
79
80 private void linePoints()
81 {
82     double angle;
83     int dx0, dy0, dx1;
84
85     angle = -(analogValue * 20 - 40);
86     dx0 = (int)(55 * System.Math.Sin(angle / 180.0 * System.Math.PI));
87     dy0 = (int)(55 * System.Math.Cos(angle / 180.0 * System.Math.PI));
88     dx1 = (int)(28 * System.Math.Tan(angle / 180.0 * System.Math.PI));
89     x0 = XM - dx0;
90     y0 = YM - dy0;
91     x1 = XM - dx1;
92 }
93 }
94
```