

```
1 using Microsoft.SPOT;
2 using Microsoft.SPOT.Hardware;
3
4 //Notwendig für Fehlerbeseitigung in Visual Studio 2015
5 namespace System.Diagnostics
6 {
7     public enum DebuggerBrowsableState
8     {
9         Never,
10        Collapsed,
11        RootHidden
12    }
13 }
14 class Program
15 {
16     const int XM = 80;
17     const int YM = 103;
18     private AnalogInput analogIn;
19     private double analogValue;
20     private PWM analogOut;
21     private int count;
22     private BrainPad.Image imageM1 = new BrainPad.Image(50, 50);
23     private BrainPad.Image imageM2 = new BrainPad.Image(50, 50);
24     private int x0, y0, x1;
25     private int x0Old, y0Old, x1Old;
26
27     public void BrainPadSetup()
28     {
29         count = 0;
30         BrainPad.Button.ButtonPressed += Button_ButtonPressed;
31         analogOut = new PWM(BrainPad.Expansion.PwmOutput.PA3, 1000, 0.0, ↗
32             false);
33         analogOut.Start();
34         analogIn = new AnalogInput(BrainPad.Expansion.AnalogInput.PA7); ↗
35         getImage(imageM1, ↗
36             BrainPad_Projekt_06B.Properties.Resources.BinaryResources.M1);
37         getImage(imageM2, ↗
38             BrainPad_Projekt_06B.Properties.Resources.BinaryResources.M2);
39
40         BrainPad.Display.DrawText(38, 8, "Analog-Input", ↗
41             BrainPad.Color.White);
42         BrainPad.Display.DrawImage(XM - 50, YM - 78, imageM1);
43         BrainPad.Display.DrawImage(XM, YM - 78, imageM2);
44         BrainPad.Display.DrawText(75, 80, "V", BrainPad.Color.White);
45         linePoints();
46     }
47
48     public void BrainPadLoop()
49     {
50         analogValue = analogIn.Read() * 3.3;
51         showPointer();
52         BrainPad.Wait.Seconds(1.0);
53     }
54
55     private void Button_ButtonPressed(BrainPad.Button.DPad button, ↗
56         BrainPad.Button.State state)
```

```
52     {
53         if (button == BrainPad.Button.DPad.Right)
54         {
55             if (state == BrainPad.Button.State.Pressed)
56             {
57                 if (count < 10)
58                     count++;
59             }
60         }
61         if (button == BrainPad.Button.DPad.Left)
62         {
63             if (state == BrainPad.Button.State.Pressed)
64             {
65                 if (count > 0)
66                     count--;
67             }
68         }
69         setPWM();
70     }
71
72     private void setPWM()
73     {
74         analogOut.DutyCycle = count * 0.1;
75     }
76
77     private void getImage(BrainPad.Image image, ↗
78         BrainPad_Projekt_06B.Properties.Resources.BinaryResources resource)
79     {
80         int info;
81         byte[] data;
82         BrainPad.Color color = new BrainPad.Color();
83
84         data = (byte[])ResourceUtility.GetObject ↗
85             (BrainPad_Projekt_06B.Properties.Resources.ResourceManager, ↗
86             resource);
87         info = data[0xA];
88         for (int Y = image.Height - 1; Y >= 0; Y--)
89         {
90             for (int X = 0; X < image.Width; X++)
91             {
92                 byte h = data[info++];
93                 byte n = data[info++];
94                 color.B = (byte)(h & 0x1f);
95                 color.G = (byte)(((h & 0xe0) >> 5) + ((n & 0x07) << 3));
96                 color.R = (byte)((n & 0xf8) >> 3);
97                 image.SetPixel(Y, X, color);
98             }
99         }
100     }
101
102     private void showPointer()
103     {
104         x001d = x0;
105         y001d = y0;
106         x101d = x1;
```

```
...06\BrainPad Projekt 06B\BrainPad Projekt 06B\Program.cs 3
105     BrainPad.Display.DrawLine(x0Old, y0Old, x1Old, YM - 29, ↗
        BrainPad.Color.Black);
106     BrainPad.Display.DrawText(67, 95, (count * 10).ToString("D3") + "%", ↗
        BrainPad.Color.White);
107     BrainPad.Display.DrawText(63, 110, analogValue.ToString("F2") + " V", ↗
        BrainPad.Color.Red);
108     linePoints();
109     BrainPad.Display.DrawLine(x0, y0, x1, YM - 29, BrainPad.Color.Red);
110 }
111
112 private void linePoints()
113 {
114     double angle;
115     int dx0, dy0, dx1;
116
117     angle = -(analogValue * 20 - 40);
118     dx0 = (int)(55 * System.Math.Sin(angle / 180.0 * System.Math.PI));
119     dy0 = (int)(55 * System.Math.Cos(angle / 180.0 * System.Math.PI));
120     dx1 = (int)(28 * System.Math.Tan(angle / 180.0 * System.Math.PI));
121     x0 = XM - dx0;
122     y0 = YM - dy0;
123     x1 = XM - dx1;
124 }
125 }
126
```