

```
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 imageUnten = new BrainPad.Image(30, 30);
17     private BrainPad.Image imageOben = new BrainPad.Image(30, 30);
18     private BrainPad.Image imageMitte = new BrainPad.Image(30, 30);
19
20     public void BrainPadSetup()
21     {
22         getImage(imageUnten,                                     ↗
23             BrainPad_Projekt_03.Properties.Resources.BinaryResources.Unten);
24         getImage(imageOben,                                     ↗
25             BrainPad_Projekt_03.Properties.Resources.BinaryResources.Oben);
26         getImage(imageMitte,                                    ↗
27             BrainPad_Projekt_03.Properties.Resources.BinaryResources.Mitte);
28         BrainPad.Display.DrawImage(65, 94, imageUnten);
29         BrainPad.Display.DrawImage(65, 4, imageOben);
30         BrainPad.Display.DrawImage(65, 34, imageMitte);
31         BrainPad.Display.DrawImage(65, 64, imageMitte);
32         BrainPad.Display.DrawFilledRectangle(79, 95, 2, 10,    ↗
33             BrainPad.Color.Red);
34
35         for(int i=0; i<5;i++)
36         {
37             BrainPad.Display.DrawLine(64, 95 - i * 20, 72, 95 - i *    ↗
38                 20, BrainPad.Color.White);
39             BrainPad.Display.DrawText(50, 92 - i * 20, (10 + i * 10).ToString    ↗
40                 (), BrainPad.Color.White);
41         }
42         for (int i = 0; i < 4; i++)
43         {
44             BrainPad.Display.DrawLine(68, 85 - i * 20, 72, 85 - i * 20,    ↗
45                 BrainPad.Color.White);
46         }
47         for (int i = 0; i < 8; i++)
48         {
49             BrainPad.Display.DrawLine(87, 95 - i * 11, 95, 95 - i * 11,    ↗
50                 BrainPad.Color.White);
51             BrainPad.Display.DrawText(98, 92 - i * 11, (50 + i * 10).ToString    ↗
52                 ("D3"), BrainPad.Color.White);
53         }
54
55         BrainPad.Display.DrawLargeLetter(20, 10, 'C', BrainPad.Color.White);
56         BrainPad.Display.DrawLargeLetter(125, 10, 'F', BrainPad.Color.White);
```

```
48     }
49
50     public void BrainPadLoop()
51     {
52         double bar;
53         double temp;
54
55         temp = BrainPad.TemperatureSensor.ReadTemperature();
56         bar = 2 * (temp - 10);
57         BrainPad.Display.DrawFilledRectangle(79, 95 - (int)bar, 2, (int)bar, ↗
58             BrainPad.Color.Red);
59         BrainPad.Display.DrawFilledRectangle(79, 15, 2, 80 - (int)bar - 1, ↗
60             BrainPad.Color.Black);
61         BrainPad.Display.DrawText(15, 109, ↗
62             BrainPad.TemperatureSensor.ReadTemperature().ToString ↗
63             ("N1"),BrainPad.Color.White);
64         BrainPad.Display.DrawText(120, 109, ↗
65             (BrainPad.TemperatureSensor.ReadTemperature()*1.8+32).ToString("N1"), ↗
66             BrainPad.Color.White);
67         BrainPad.Wait.Seconds(1.0);
68     }
69
70     private void getImage(BrainPad.Image image, ↗
71         BrainPad_Projekt_03.Properties.Resources.BinaryResources resource)
72     {
73         int info;
74         byte[] data;
75         BrainPad.Color color = new BrainPad.Color();
76
77         data = (byte[])ResourceUtility.GetObject ↗
78             (BrainPad_Projekt_03.Properties.Resources.ResourceManager, resource);
79         info = data[0xA];
80         for (int Y = image.Height - 1; Y >= 0; Y--)
81         {
82             for (int X = 0; X < image.Width; X++)
83             {
84                 byte h = data[info++];
85                 byte n = data[info++];
86                 color.B = (byte)(h & 0x1f);
87                 color.G = (byte)(((h & 0xe0) >> 5) + ((n & 0x07) << 3));
88                 color.R = (byte)((n & 0xf8) >> 3);
89                 image.SetPixel(Y, X, color);
90             }
91         }
92     }
93 }
```