

```
1 using Microsoft.SPOT;
2 using System.Threading;
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     private const int PAUSE= 100;
18     private BrainPad.Image imageBlau1 = new BrainPad.Image(10, 10);
19     private BrainPad.Image imageBlau2 = new BrainPad.Image(10, 10);
20     private BrainPad.Image imageBlau3 = new BrainPad.Image(10, 10);
21     private BrainPad.Image imageBlau4 = new BrainPad.Image(10, 10);
22
23     public void BrainPadSetup()
24     {
25         getImage(imageBlau1, BrainPad\_Projekt\_02.Properties.Resources.BinaryResources.Blau\_1);
26         getImage(imageBlau2, BrainPad\_Projekt\_02.Properties.Resources.BinaryResources.Blau\_2);
27         getImage(imageBlau3, BrainPad\_Projekt\_02.Properties.Resources.BinaryResources.Blau\_3);
28         getImage(imageBlau4, BrainPad\_Projekt\_02.Properties.Resources.BinaryResources.Blau\_4);
29
30         //Phase 0
31         BrainPad.Display.DrawImage(75, 39, imageBlau1);
32         BrainPad.Display.DrawImage(89, 45, imageBlau4);
33         BrainPad.Display.DrawImage(95, 59, imageBlau4);
34         BrainPad.Display.DrawImage(89, 73, imageBlau4);
35         BrainPad.Display.DrawImage(75, 79, imageBlau4);
36         BrainPad.Display.DrawImage(61, 73, imageBlau4);
37         BrainPad.Display.DrawImage(55, 59, imageBlau3);
38         BrainPad.Display.DrawImage(61, 45, imageBlau2);
39         Thread.Sleep(PAUSE);
40     }
41
42     public void BrainPadLoop()
43     {
44         //Phase 1
45         BrainPad.Display.DrawImage(75, 39, imageBlau2);
46         BrainPad.Display.DrawImage(89, 45, imageBlau1);
47         BrainPad.Display.DrawImage(55, 59, imageBlau4);
48         BrainPad.Display.DrawImage(61, 45, imageBlau3);
49         Thread.Sleep(PAUSE);
50
51         //Phase 2
52         BrainPad.Display.DrawImage(75, 39, imageBlau3);
```

```
53     BrainPad.Display.DrawImage(89, 45, imageBlau2);
54     BrainPad.Display.DrawImage(95, 59, imageBlau1);
55     BrainPad.Display.DrawImage(61, 45, imageBlau4);
56     Thread.Sleep(PAUSE);
57
58     //Phase 3
59     BrainPad.Display.DrawImage(75, 39, imageBlau4);
60     BrainPad.Display.DrawImage(89, 45, imageBlau3);
61     BrainPad.Display.DrawImage(95, 59, imageBlau2);
62     BrainPad.Display.DrawImage(89, 73, imageBlau1);
63     Thread.Sleep(PAUSE);
64
65     //Phase 4
66     BrainPad.Display.DrawImage(89, 45, imageBlau4);
67     BrainPad.Display.DrawImage(95, 59, imageBlau3);
68     BrainPad.Display.DrawImage(89, 73, imageBlau2);
69     BrainPad.Display.DrawImage(75, 79, imageBlau1);
70     Thread.Sleep(PAUSE);
71
72     //Phase 5
73     BrainPad.Display.DrawImage(95, 59, imageBlau4);
74     BrainPad.Display.DrawImage(89, 73, imageBlau3);
75     BrainPad.Display.DrawImage(75, 79, imageBlau2);
76     BrainPad.Display.DrawImage(61, 73, imageBlau1);
77     Thread.Sleep(PAUSE);
78
79     //Phase 6
80     BrainPad.Display.DrawImage(89, 73, imageBlau4);
81     BrainPad.Display.DrawImage(75, 79, imageBlau3);
82     BrainPad.Display.DrawImage(61, 73, imageBlau2);
83     BrainPad.Display.DrawImage(55, 59, imageBlau1);
84     Thread.Sleep(PAUSE);
85
86     //Phase 7
87     BrainPad.Display.DrawImage(75, 79, imageBlau4);
88     BrainPad.Display.DrawImage(61, 73, imageBlau3);
89     BrainPad.Display.DrawImage(55, 59, imageBlau2);
90     BrainPad.Display.DrawImage(61, 45, imageBlau1);
91     Thread.Sleep(PAUSE);
92
93     //Phase 0
94     BrainPad.Display.DrawImage(75, 39, imageBlau1);
95     BrainPad.Display.DrawImage(61, 73, imageBlau4);
96     BrainPad.Display.DrawImage(55, 59, imageBlau3);
97     BrainPad.Display.DrawImage(61, 45, imageBlau2);
98     Thread.Sleep(PAUSE);
99 }
100
101 private void getImage(BrainPad.Image image, ?
    BrainPad_Projekt_02.Properties.Resources.BinaryResources resource)
102 {
103     int info;
104     byte[] data;
105     BrainPad.Color color=new BrainPad.Color();
106
107     data = (byte[])ResourceUtility.GetObject ?
```

```
(BrainPad_Projekt_02.Properties.Resources.ResourceManager,  
resource);  
108 info = data[0xA];  
109 for (int Y = image.Height - 1; Y >= 0; Y--)  
110 {  
111     for (int X = 0; X < image.Width; X++)  
112     {  
113         byte h = data[info++];  
114         byte n = data[info++];  
115         color.B = (byte)(h & 0x1f);  
116         color.G = (byte)(((h & 0xe0) >> 5) + ((n & 0x07) << 3));  
117         color.R = (byte)((n & 0xf8) >> 3);  
118         image.SetPixel(Y, X, color);  
119     }  
120 }  
121 }  
122 }  
123 }
```