

```
1 #Event: Connected (Success As Boolean)
2 #Event: StateChanged (State As String)
3 Sub Class_Globals
4     Private Admin As BluetoothAdmin
5     Public Astreams As AsyncStreams
6     Private Serial As Serial
7     Private PH As Phone
8     Private mParent As Object
9     Private mEventName As String
10    Private lblStatus As Label
11    Private ProgressBar1 As ProgressBar
12    Private GenericDeviceName, DeviceName As String
13    Public BluetoothState, ConnectionState, DeviceFound As Boolean
14    Private DeviceName, DeviceMacAdress As String
15    Public CharSet = "UTF-8" As String
16    Private sb As StringBuilder
17    Public State As String
18 End Sub
19
20
21 Public Sub Initialize(Parent As Object, EventName As String, StatusLabel As Label, Pgb As Progress
    ProgressBar)
22     mParent = Parent
23     mEventName = EventName
24     lblStatus = StatusLabel
25     Admin.Initialize("Admin")
26     Serial.Initialize("Serial")
27     ProgressBar1 = Pgb
28     sb.Initialize
29
30     If Admin.IsEnabled = False Then
31         If Admin.Enable = False Then
32             ToastMessageShow("Module Bluetooth non disponible", True)
33         Else
34             ToastMessageShow("Module Bluetooth disponible...", False)
35         End If
36     Else
37         BluetoothState = True
38     End If
39 End Sub
40
41 Public Sub Connect(Name As String)
42     GenericDeviceName = Name
43     Dim success As Boolean = Admin.StartDiscovery
44     If success = False Then
45         lblStatus.Text = "Erreur du processus de recherche"
46     Else
47         lblStatus.Text = "Recherche dispositif en cours"
48         'ProgressBar1.Visible = True
49     End If
50 End Sub
51
52 Public Sub Disconnect
53     If Astreams.IsInitialized Then Astreams.Close
54     If Serial.IsInitialized Then Serial.Disconnect
55 End Sub
56
```

```
57 Private Sub Admin_DeviceFound (Name As String, MacAddress As String)
58     Log(Name & ":" & MacAddress)
59     If Name.Contains(GenericDeviceName) Then
60         Log(GenericDeviceName & " trouvé")
61         DeviceName = Name
62         DeviceMacAddress = MacAddress
63         If PH.SdkVersion <= 24 Then
64             Admin.CancelDiscovery
65             lblStatus.Text = GenericDeviceName & " trouvé"
66         Else
67             lblStatus.Text = GenericDeviceName & " trouvé, veuillez patienter."
68         End If
69     End If
70 End Sub
71
72 Private Sub Admin_DiscoveryFinished
73     If DeviceName = "" Then
74         lblStatus.Text = "Module " & GenericDeviceName & " non détecté"
75         MsgBox2Async("Le Module Bluetooth HC-05 ne semble pas être alimenté, allumez la case
76         créeole, la mini-led rouge du module doit clignoter rapidement puis appuyer de nouveau sur
77         le bouton «Connexion».", "Vérifiez l'alimentation de la Case Créole", "OK", "", "", Null,
78         True)
79     Wait For MsgBox_Result (Result As Int)
80     Else
81         lblStatus.Text = "Connexion à " & GenericDeviceName
82         Serial.Connect(DeviceMacAddress)
83     End If
84 End Sub
85
86 Private Sub Admin_StateChanged (NewState As Int, OldState As Int)
87     Log("Changement de statut: " & NewState)
88     lblStatus.Text = "Statut actuel: " & NewState
89     BluetoothState = NewState = Admin.STATE_ON
90 End Sub
91
92 Sub Serial_Connected (Success As Boolean)
93     Private msg As String
94     If Success = True Then
95         If Astreams.IsInitialized Then Astreams.Close
96         Astreams.Initialize(Serial.InputStream, Serial.OutputStream, "Astreams")
97         msg = "Module Bluetooth HC05 Connecté"
98         ProgressBar1.Visible = False
99     Else
100         Log(LastException.Message)
101         msg = LastException.Message
102     End If
103     lblStatus.Text = msg
104     CallSubDelayed2(mParent, mEventName & "_Connected", Success)
105 End Sub
106
107 Public Sub SendBytes(Buffer() As Byte)
108     Astreams.Write(Buffer)
109 End Sub
110
111 Public Sub SendText(Text As String)
112     Astreams.Write(Text.GetBytes(CharSet))
```

```
111 End Sub
112
113 Public Sub Ast_NewText(Text As String)
114     CallSubDelayed2(mParent, mEventName & "_NewText", Text)
115     Log(Text)
116 End Sub
117
118 Public Sub WriteText(Text As String)
119     Astreams.Write(Text.GetBytes(CharSet))
120 End Sub
121
122 Public Sub WriteBytes(Buffer() As Byte)
123     Astreams.Write(Buffer)
124 End Sub
125
126 Private Sub Astreams_NewData (Buffer() As Byte)
127     Dim newDataStart As Int = sb.Length
128     sb.Append(BytesToString(Buffer, 0, Buffer.Length, CharSet))
129     Dim s As String = sb.ToString
130     Dim start As Int = 0
131     For i = newDataStart To s.Length - 1
132         Dim c As Char = s.CharAt(i)
133         If i = 0 And c = Chr(10) Then '\n...
134             start = 1
135             Continue
136         End If
137         If c = Chr(10) Then
138             CallSubDelayed2(mParent, mEventName & "_NewText", s.SubString2(start, i))
139             start = i + 1
140         Else If c = Chr(13) Then
141             CallSubDelayed2(mParent, mEventName & "_NewText", s.SubString2(start, i))
142             If i < s.Length - 1 And s.CharAt(i + 1) = Chr(10) Then
143                 i = i + 1
144             End If
145             start = i + 1
146         End If
147     Next
148     If start > 0 Then sb.Remove(0, start)
149 End Sub
150
151 Private Sub Astreams_Terminated
152     CallSubDelayed(mParent, mEventName & "_Termin ")
153 End Sub
154
155 Private Sub Astreams_Error
156     Log("erreur: " & LastException)
157     Astreams.Close
158     CallSubDelayed(mParent, mEventName & "_Termin ")
159 End Sub
160
161 Public Sub Close
162     Astreams.Close
163 End Sub
164
```