

```

1: unit BYCommand;
2:
3: {$mode objfpc}{$H+}
4:
5: {=====}
6: {
7: {   BYCommand.pas
8: {
9: {   Calls: AppConstants
10: {           LCDDisplay : DisplayUHFBusyStatus
11: {                               DisplayVHFBusyStatus
12: {                               SerialStuff : SendCommand
13: {           SerialStuff : SednCommand
14: {
15: {   Called By: ResponseParser : ParseResponse
16: {           Init : Initialize
17: {
18: {   Ver: 1.0.0
19: {
20: {   Date: 9 May 2013
21: {
22: {=====}
23:
24: interface
25:
26: uses
27:   Classes, SysUtils,
28:   //      APPLICATION UNITS
29:   AppConstants, LCDDisplay, SerialStuff;
30:
31: procedure BYResponseHandler (vstrKeywordRcvd, vstrParameters : string);
32: procedure GetUHFBYStatus;
33: procedure GetVHFBYStatus;
34:
35: implementation
36: //=====
37:
38: procedure BYResponseHandler (vstrKeywordRcvd, vstrParameters : string);
39:
40: var
41:   vstrBand : string;
42:   vstrBusy : string;
43:
44: begin
45:
46:   // First we parse the reponse (n1,n2) where n1 = Band (0 = VHF, 1 = UHF) and
47:   // n2 = Busy (0 = Not Busy, 1 = Busy).
48:   vstrBand := Copy(vstrParameters, 1,1);
49:   vstrBusy := Copy(vstrParameters, 3,1);
50:
51:   if vstrBand = gcstrUHF then
52:   begin
53:     if vstrBusy = gcstrOn then
54:       DisplayUHFBusyStatus('BUSY')
55:     else
56:       DisplayUHFBusyStatus('');
57:   end
58:   else
59:   begin
60:     if vstrBusy = gcstrOn then

```

```
61:         DisplayVHFBusyStatus('BUSY')
62:     else
63:         DisplayVHFBusyStatus('');
64: end; // if vstrBand = gcstrUHF
65:
66: end; // procedure BYResponseHandler
67:
68: //=====
69:
70: procedure GetUHFBYStatus;
71: begin
72:     SendCommand('BY',gcstrUHF);
73: end; // procedure GetUHFBYStatus;
74:
75: //=====
76:
77: procedure GetVHFBYStatus;
78: begin
79:     SendCommand('BY',gcstrVHF);
80: end; // procedure GetVHFBYStatus;
81:
82: //=====
83:
84: end. // unit BYCommand;
85:
```