```
unit BUFCommand;

{$mode objfpc}{$H+}

//=============================================================================================
//
//  BUFCommand.pas
//
// Description: Description: Reads and Sets the VHF and UHF Buffer data consisting of the
// following data items separated by commas:
//
//          0,00145290000,0,2,0,1,0,0,28,000,19,000600000
//          1,00443575000,6,1,0,1,0,0,13,000,19,005000000
//
//          VFO : [0..1] [VHF..UHF]
//          Frequency : [nnnnnnnnnnn] 11 characters - Freq in Hz
//          Step : [n] 1 characters - Frequency Step Size
//          Shift/Offset : [n] [0..2] - 0=None, 1=Plus (Up), 2=Minus (Down)
//          Reverse : [n] 0 = OFF 1 = ON
//          Tone : [n] 0 = OFF 1 = ON
//          CTCSS : [n] 0 = OFF 1 = ON
//          DTSS : [n] 0 = OFF 1 = ON
//          Tone Freq : [nn] Tone Freq Code [01..39]
//          DTSS Code : [nnn] [000..999]
//          CTCSS Freq : [nn] Tone Freq Code [01..39]
//          Offset : [nnnnnnnnn] 9 characters - Repeater Offset in Hz
//
//  Calls: AppConstants
//         AppVariables
//         LCDDisplay : DisplayVHFRXFrequency
//                      DisplayUHFRXFrequency
//                      DisplayUHFShiftStatus
//                      DisplayVHFShiftStatus
//                      DisplayUHFReverseStatus
//                      DisplayVHFReverseStatus
//                      DisplayUHFCTStatus
//                      DisplayVHFCTStatus
//                      DisplayUHFDTSSStatus
//                      DisplayVHFDTSSStatus
//                      DisplayUHFRFPowerStatus
//                      DisplayVHFRFPowerStatus
//                      DisplayUHFDataSource
//                      DisplayVHFDataSource
//                      DisplayUHFChannelNr
//                      DisplayVHFChannelNr
//                      DisplayUHFChannelName
//                      DisplayvHFChannelName
//         Reverse : ToggleReverse
//         SerialStuff : SendCommand
//         StatusBar : DisplayCommentStatus
//         Variables
//
//  Called By: Fav : SetFAVChannel
//      Init : Initialize
//         Mem_VHF : TfrmMEM.bbtSelectClick
//
//  Ver: 1.0.0
//
//  Date: 11 Dec 2013
//
```

```
 61: //=============================================================================================
 62:
 63: interface
 64:
 65: uses
 66:   Classes, Dialogs, SysUtils,
 67:   // Application Units
 68:   AppConstants, LCDDisplay, Reverse, SerialStuff, StatusBar, AppVariables;
 69:
 70: procedure BUFResponseHandler (vstrKeywordRcvd, vstrParameters : string);
 71: procedure GetBufferData(vstrBand : string);
 72: procedure SetBuffer(vstrBand : string);
 73:
 74: implementation
 75:
 76: //=============================================================================================
 77:
 78: procedure BUFResponseHandler (vstrKeywordRcvd, vstrParameters : string);
 79: begin
 80:
 81:   // First we determine the Band is (VHF or UHF) and then save the data
 82:   if Copy(vstrParameters,1,1) = gcstrVHF then
 83:   begin
 84:
 85:     gvstrVHFRXFrequency := Copy(vstrParameters,3,11);
 86:     gvstrVHFStep := Copy(vstrParameters,15,1);
 87:     gvstrVHFShift := Copy(vstrParameters,17,1);
 88:     gvstrVHFReverse := Copy(vstrParameters,19,1);
 89:     gvstrVHFTone := Copy(vstrParameters,21,1);
 90:     gvstrVHFCTCSS := Copy(vstrParameters,23,1);
 91:     gvstrVHFDTSS := Copy(vstrParameters,25,1);
 92:     gvstrVHFToneNr := Copy(vstrParameters,27,2);
 93:     gvstrVHFDTSSCode := Copy(vstrParameters,30,3);
 94:     gvstrVHFCTCSSNr := Copy(vstrParameters,34,2);
 95:     gvstrVHFOffset := Copy(vstrParameters,37,9);
 96:
 97:   end
 98:   else
 99:   begin
100:
101:     gvstrUHFRXFrequency := Copy(vstrParameters,3,11);
102:     gvstrUHFStep := Copy(vstrParameters,15,1);
103:     gvstrUHFShift := Copy(vstrParameters,17,1);
104:     gvstrUHFReverse := Copy(vstrParameters,19,1);
105:     gvstrUHFTone := Copy(vstrParameters,21,1);
106:     gvstrUHFCTCSS := Copy(vstrParameters,23,1);
107:     gvstrUHFDTSS := Copy(vstrParameters,25,1);
108:     gvstrUHFToneNr := Copy(vstrParameters,27,2);
109:     gvstrUHFDTSSCode := Copy(vstrParameters,30,3);
110:     gvstrUHFCTCSSNr := Copy(vstrParameters,34,2);
111:     gvstrUHFOffset := Copy(vstrParameters,37,9);
112:
113:   end;//if Copy(vstrParameters,1,1) = gcstrVHF
114:
115:   // Now we display the data
116:   DisplayUHFRXFrequency;
117:   DisplayVHFRXFrequency;
118:   DisplayUHFShiftStatus;
119:   DisplayVHFShiftStatus;
120:   DisplayUHFReverseStatus;
```

```
121:   DisplayVHFReverseStatus;
122:   DisplayUHFCTStatus;
123:   DisplayVHFCTStatus;
124:   DisplayUHFDTSSStatus;
125:   DisplayVHFDTSSStatus;
126:   DisplayUHFRFPowerStatus;
127:   DisplayVHFRFPowerStatus;
128:   DisplayUHFDataSource;
129:   DisplayVHFDataSource;
130:   DisplayUHFChannelNr;
131:   DisplayVHFChannelNr;
132:   DisplayUHFChannelName;
133:   DisplayVHFChannelName;
134:   DisplayCommentStatus;
135:
136: end;// procedure BUFResponseHandler
137:
138: //=============================================================================================
139:
140: procedure GetBufferData(vstrBand : string);
141: begin
142:
143: end;// procedure GetBufferData
144:
145: //=============================================================================================
146:
147: procedure SetBuffer(vstrBand : string);
148:
149: var
150:   vstrCmdStr : string;
151:
152: begin
153:
154:   if vstrBand = gcstrVHF then
155:   begin
156:
157:   if gvstrVHFReverseState = gcstrOn then
158:        VHFReverseOff;
159:
160:      vstrCmdStr := vstrCmdStr +
161:                    gcstrVHFVFO + ',' +
162:                    gvstrVHFRXFrequency + ',' +
163:                    gvstrVHFStep + ',' +
164:                    gvstrVHFShift + ',' +
165:                    gvstrVHFReverse + ',' +
166:                    gvstrVHFTone + ',' +
167:                    gvstrVHFCTCSS + ',' +
168:                    gvstrVHFDTSS + ',' +
169:                    gvstrVHFToneNr + ',' +
170:                    gvstrVHFDTSSCode + ',' +
171:                    gvstrVHFCTCSSNr + ',' +
172:                    gvstrVHFOffset;
173:   end
174:   else
175:   begin
176:
177:     if gvstrUHFReverseState = gcstrOn then
178:          UHFReverseOff;
179:
180:     vstrCmdStr := vstrCmdStr +
```

```
181:                   gcstrUHFVFO + ',' +
182:                   gvstrUHFRXFrequency + ',' +
183:                   gvstrUHFStep + ',' +
184:                   gvstrUHFShift + ',' +
185:                   gvstrUHFReverse + ',' +
186:                   gvstrUHFTone + ',' +
187:                   gvstrUHFCTCSS + ',' +
188:                   gvstrUHFDTSS + ',' +
189:                   gvstrUHFToneNr + ',' +
190:                   gvstrUHFDTSSCode + ',' +
191:                   gvstrUHFCTCSSNr + ',' +
192:                   gvstrUHFOffset;
193:   end;// if vstrBand = gcstrVHF
194:
195: //***  showmessage(vstrCmdStr);
196:
197:   SendCommand('BUF', vstrCmdStr);
198:
199: end;// procedure SetBuffer
200:
201: //=============================================================================
202:
203: end.// unit BUFCommand;
204:
```