```
 1: unit MEM;
 2:
 3: {$mode objfpc}{$H+}
 4:
 5: //=============================================================================================
 6: //
 7: //  Mem.pas
 8: //
 9: //  Calls: AppConstants
10: //         AppVariables
11: //         DataEntry
12: //         MEM_VHF : LoadVHFStringGrid
13: //                   SetVHFChannel
14: //         MEM_UHF : LoadJUHFStringGrid
15: //                   SetUHFChannel
16: //         MEM_DTMF : LoadDTMFStringGrid
17: //                   SetDTMFCode
18: //         Utilities : GetToneFrequencyFromToneNr
19: //
20: //  Called By: Main : TfrmMain.mnuMemVHFClick
21: //                    TfrmMain.mnuMemUHFClick
22: //                    TfrmMain.mnuMemDTMFClick
23: //
24: //  Ver: 1.0.0
25: //
26: //  Date: 11 Aug 2013
27: //
28: //=============================================================================================
29:
30: interface
31:
32: uses
33:   Classes, SysUtils, FileUtil, ExtendedNotebook, Forms, Controls, Graphics,
34:   Dialogs, ExtCtrls, Grids, Buttons, StdCtrls, Messages,
35:   // Application Units
36:   AppConstants, AppTypes, AppVariables, BCCommand, BUFCommand, DataEntry, LCDDisplay,
37:   MEM_DTMF, MEM_UHF, MEM_VHF, Utilities;
38:
39: type
40:
41:   TfrmMEM = class(TForm)
42:     bbtSelect: TBitBtn;
43:     bbtClose: TBitBtn;
44:     bbtEdit: TBitBtn;
45:     sgrVHF: TStringGrid;
46:     sgrUHF: TStringGrid;
47:     sgrDTMF: TStringGrid;
48:     procedure bbtCloseClick(Sender: TObject);
49:     procedure bbtEditClick(Sender: TObject);
50:     procedure bbtSelectClick(Sender: TObject);
51:     procedure Setup(vdrtDataRecType : TDataRecordType);
52:     procedure sgrUHFDblClick(Sender: TObject);
53:     procedure sgrUHFMouseUp(Sender: TObject; Button: TMouseButton;
54:       Shift: TShiftState; X, Y: Integer);
55:     procedure sgrVHFDblClick(Sender: TObject);
56:     procedure sgrVHFMouseUp(Sender: TObject; Button: TMouseButton;
57:       Shift: TShiftState; X, Y: Integer);
58:   private
59:     { private declarations }
```

```
 60:     vdrtRecType : TDataRecordType;
 61:   public
 62:     { public declarations }
 63:   end;
 64:
 65: var
 66:   frmMEM: TfrmMEM;
 67:
 68: implementation
 69:
 70: {$R *.lfm}
 71:
 72: const
 73:
 74:   cbytVHFColCOunt = 14;
 75:   cbytUHFColCOunt = 14;
 76:   cbytDTMFColCount = 2;
 77:
 78:   cbytDTMFCodeCol = 1;
 79:
 80: //=============================================================================
 81: //          SUPPORT ROUTINES
 82: //=============================================================================
 83: procedure SetVHFUHFHeaders;
 84:
 85: const
 86:   cbytCHNrColWidth = 47;
 87:   cstrCHNrColHdr = 'CH Nr';
 88:   cbytNameColWidth = 160;
 89:   cstrNameColHdr = '             CH Name';
 90:   cbytRXFreqColWidth = 65;
 91:   cstrRXFreqColHdr = ' RX Freq';
 92:   cbytShiftColWidth = 58;
 93:   cstrShiftColHdr = '   Shift';
 94:   cbytOffsetColWidth = 58;
 95:   cstrOffsetColHdr = 'Offset';
 96:   cbytToneCTCSSColWidth = 50;
 97:   cstrToneCTCSSColHdr = ' Tone';
 98:   cbytToneFreqColWidth = 50;
 99:   cstrToneFreqColHdr = ' Freq';
100:   cbytRFPwrColWidth = 65;
101:   cstrRFPwrColHdr = 'RF Pwr';
102:   cstrDTSSColWidth = 50;
103:   cstrDTSSDColHdr = ' DTSS';
104:   cstrDTSSCodeColWidth = 75;
105:   cstrDTSSDCodeColHdr = 'DTSS Code';
106:   cstrReverseColWidth = 60;
107:   cstrReverseColHdr = 'Reverse';
108:   cstrScanColWidth = 40;
109:   cstrScanColHdr = 'Scan';
110:   cstrStepColWidth = 40;
111:   cstrStepColHdr = 'Step';
112:   cstrCommentsColWidth = 375;
113:   cstrCommentsColHdr = '                                    Comments';
114:
115: begin
116:
117:   // Channel Nr
118:   frmMem.sgrVHF.ColWidths[gcbytChMemNrCol] := cbytChNrColWidth;
```

```
119:   frmMEM.sgrVHF.Cells[gcbytChMemNrCol,0] := cstrCHNrColHdr;
120:   frmMem.sgrUHF.ColWidths[gcbytChMemNrCol] := cbytChNrColWidth;
121:   frmMEM.sgrUHF.Cells[gcbytChMemNrCol,0] := cstrCHNrColHdr;
122:   // Ch Name
123:   frmMem.sgrVHF.ColWidths[gcbytNameCol] := cbytNameColWidth;
124:   frmMEM.sgrVHF.Cells[gcbytNameCol,0] := cstrNameColHdr;
125:   frmMem.sgrUHF.ColWidths[gcbytNameCol] := cbytNameColWidth;
126:   frmMEM.sgrUHF.Cells[gcbytNameCol,0] := cstrNameColHdr;
127:   // RX Frequency
128:   frmMem.sgrVHF.ColWidths[gcbytRXFreqCol] := cbytRXFreqColWidth;
129:   frmMEM.sgrVHF.Cells[gcbytRXFreqCol,0] := cstrRXFreqColHdr;
130:   frmMem.sgrUHF.ColWidths[gcbytRXFreqCol] := cbytRXFreqColWidth;
131:   frmMEM.sgrUHF.Cells[gcbytRXFreqCol,0] := cstrRXFreqColHdr;
132:   // Shift
133:   frmMem.sgrVHF.ColWidths[gcbytShiftCol] := cbytShiftColWidth;
134:   frmMEM.sgrVHF.Cells[gcbytShiftCol,0] := cstrShiftColHdr;
135:   frmMem.sgrUHF.ColWidths[gcbytShiftCol] := cbytShiftColWidth;
136:   frmMEM.sgrUHF.Cells[gcbytShiftCol,0] := cstrShiftColHdr;
137:   // Offset
138:   frmMem.sgrVHF.ColWidths[gcbytOffsetCol] := cbytOffsetColWidth;
139:   frmMEM.sgrVHF.Cells[gcbytOffsetCol,0] := cstrOffsetColHdr;
140:   frmMem.sgrUHF.ColWidths[gcbytOffsetCol] := cbytOffsetColWidth;
141:   frmMEM.sgrUHF.Cells[gcbytOffsetCol,0] := cstrOffsetColHdr;
142:   // Tone CTCSS
143:   frmMem.sgrVHF.ColWidths[gcbytToneCTCSSCol] := cbytToneCTCSSColWidth;
144:   frmMEM.sgrVHF.Cells[gcbytToneCTCSSCol,0] := cstrToneCTCSSColHdr;
145:   frmMem.sgrUHF.ColWidths[gcbytToneCTCSSCol] := cbytToneCTCSSColWidth;
146:   frmMEM.sgrUHF.Cells[gcbytToneCTCSSCol,0] := cstrToneCTCSSColHdr;
147:   // Tone Freq
148:   frmMem.sgrVHF.ColWidths[gcbytToneCTCSSFreqCol] := cbytToneFreqColWidth;
149:   frmMEM.sgrVHF.Cells[gcbytToneCTCSSFreqCol,0] := cstrToneFreqColHdr;
150:   frmMem.sgrUHF.ColWidths[gcbytToneCTCSSFreqCol] := cbytToneFreqColWidth;
151:   frmMEM.sgrUHF.Cells[gcbytToneCTCSSFreqCol,0] := cstrToneFreqColHdr;
152:   // RF Powerr
153:   frmMem.sgrVHF.ColWidths[gcbytRFPowerCol] := cbytRFPwrColWidth;
154:   frmMEM.sgrVHF.Cells[gcbytRFPowerCol,0] := cstrRFPwrColHdr;
155:   frmMem.sgrUHF.ColWidths[gcbytRFPowerCol] := cbytRFPwrColWidth;
156:   frmMEM.sgrUHF.Cells[gcbytRFPowerCol,0] := cstrRFPwrColHdr;
157:   // DTSS
158:   frmMem.sgrVHF.ColWidths[gcbytDTSSCol] := cstrDTSSColWidth;
159:   frmMEM.sgrVHF.Cells[gcbytDTSSCol,0] := cstrDTSSDColHdr;
160:   frmMem.sgrUHF.ColWidths[gcbytDTSSCol] := cstrDTSSColWidth;
161:   frmMEM.sgrUHF.Cells[gcbytDTSSCol,0] := cstrDTSSDColHdr;
162:   // DTSS CODE
163:   frmMem.sgrVHF.ColWidths[gcbytDTSSCodeCol] := cstrDTSSCodeColWidth;
164:   frmMEM.sgrVHF.Cells[gcbytDTSSCodeCol,0] := cstrDTSSDCodeColHdr;
165:   frmMem.sgrUHF.ColWidths[gcbytDTSSCodeCol] := cstrDTSSCodeColWidth;
166:   frmMEM.sgrUHF.Cells[gcbytDTSSCodeCol,0] := cstrDTSSDCodeColHdr;
167:   // Reverse
168:   frmMem.sgrVHF.ColWidths[gcbytReverseCol] := cstrReverseColWidth;
169:   frmMEM.sgrVHF.Cells[gcbytReverseCol,0] := cstrReverseColHdr;
170:   frmMem.sgrUHF.ColWidths[gcbytReverseCol] := cstrReverseColWidth;
171:   frmMEM.sgrUHF.Cells[gcbytReverseCol,0] := cstrReverseColHdr;
172:   // SCAN
173:   frmMem.sgrVHF.ColWidths[gcbytScanCol] := cstrScanColWidth;
174:   frmMEM.sgrVHF.Cells[gcbytScanCol,0] := cstrScanColHdr;
175:   frmMem.sgrUHF.ColWidths[gcbytScanCol] := cstrScanColWidth;
176:   frmMEM.sgrUHF.Cells[gcbytScanCol,0] := cstrScanColHdr;
177:   // STEP
```

```
178:   frmMem.sgrVHF.ColWidths[gcbytStepCol] := cstrStepColWidth;
179:   frmMEM.sgrVHF.Cells[gcbytStepCol,0] := cstrStepColHdr;
180:   frmMem.sgrUHF.ColWidths[gcbytStepCol] := cstrStepColWidth;
181:   frmMEM.sgrUHF.Cells[gcbytStepCol,0] := cstrStepColHdr;
182:   // COMMENTS
183:   frmMem.sgrVHF.ColWidths[gcbytCommentCol] := cstrCommentsColWidth;
184:   frmMEM.sgrVHF.Cells[gcbytCommentCol,0] := cstrCommentsColHdr;
185:   frmMem.sgrUHF.ColWidths[gcbytCommentCol] := cstrCommentsColWidth;
186:   frmMEM.sgrUHF.Cells[gcbytCommentCol,0] := cstrCommentsColHdr;
187:
188: end;// procedure SetVHFUHFHeaders;
189:
190: //-----------------------------------------------------------------------------------
191: procedure SetDTMFHeaders;
192:
193: const
194:   cbytChNrColWidth = 57;
195:   cbytNameColWidth = 80;
196:
197: begin
198:   frmMem.sgrDTMF.ColWidths[gcbytChMemNrCol] := cbytChNrColWidth;
199:   frmMEM.sgrDTMF.Cells[gcbytChMemNrCol,0] := 'Mem Nr';
200:   frmMem.sgrDTMF.ColWidths[cbytDTMFCodeCol] := cbytNameColWidth;
201:   frmMEM.sgrDTMF.Cells[cbytDTMFCodeCol,0] := 'DTMF Code';
202:
203: end;// procedure SetDTMFHeaders
204:
205: //===================================================================================
206: //          FORM ROUTINES
207: //===================================================================================
208: procedure TfrmMEM.Setup(vdrtDataRecType : TDataRecordType);
209:
210: Const
211:   cbytVHFUHFTop = 10;
212:   cbytVHFUHFLeft = 10;
213:   cbytVHFUHFWidth = 675;
214:   cbytVHFUHFHeight = 325;
215:   cbytDTMFTop = 60;
216:   cbytDTMFLeft = 260;
217:   cbytDTMFWidth = 157;
218:   cbytDTMFHeight = 246;
219:
220: var
221:   vbytTemp : Byte;
222:
223: begin
224:
225:   sgrVHF.Visible := False;
226:   sgrUHF.Visible := False;
227:   sgrDTMF.Visible := False;
228:
229:   vdrtRecType := vdrtDataRecType;
230:
231:   case vdrtDataRecType of
232:     drtVHFMEM : begin
233:                   Caption := 'VHF Data';
234:                   sgrVHF.Top := cbytVHFUHFTop;
235:                   sgrVHF.Left := cbytVHFUHFLeft;
236:                   sgrVHF.Width := cbytVHFUHFWidth;
```

```
237:                  sgrVHF.Height := cbytVHFUHFHeight;
238:                  sgrVHF.ColCount := cbytVHFColCount;
239:                  sgrVHF.RowCount := gcbytMaxVHFChannels+1;
240:                  SetVHFUHFHeaders;
241:                  LoadVHFStringGrid;
242:                  sgrVHF.Row:=1;
243:                  sgrVHF.Visible := True;
244:                end;
245:     drtUHFMEM : begin
246:                  Caption := 'UHF Data';
247:                  sgrUHF.Top := cbytVHFUHFTop;
248:                  sgrUHF.Left := cbytVHFUHFLeft;
249:                  sgrUHF.Width := cbytVHFUHFWidth;
250:                  sgrUHF.Height := cbytVHFUHFHeight;
251:                  sgrUHF.ColCount := cbytUHFColCOunt;
252:                  sgrUHF.RowCount := gcbytMaxUHFChannels+1;
253:                  SetVHFUHFHeaders;
254:                  LoadUHFStringGrid;
255:                  sgrUHF.Row:=1;
256:                  sgrUHF.Visible := True;
257:                end;
258:     drtDTMF : begin
259:                   Caption := 'DTMF Data';
260:                   sgrDTMF.Top := cbytDTMFTop;
261:                   sgrDTMF.Left := cbytDTMFLeft;
262:                   sgrDTMF.Width := cbytDTMFWidth;
263:                   sgrDTMF.Height := cbytDTMFHeight;
264:                   sgrDTMF.ColCOunt := cbytDTMFColCOunt;
265:                   sgrDTMF.RowCount := gcbytMaxDTMFCodes+1;
266:                   SetDTMFHeaders;
267:                   LoadDTMFStringGrid;
268:                   sgrDTMF.Visible := True;
269:                 end;
270:     end;// case vdrtDataRecType
271:
272:   ShowModal;
273:
274: end;// procedure TfrmMEM.Setup
275:
276: //=============================================================================================
277: //          BUTTON ROUTINES
278: //=============================================================================================
279: procedure TfrmMEM.bbtEditClick(Sender: TObject);
280: begin
281:
282:   case vdrtRecType of
283:     drtVHFMEM : frmDataEntry.vdetDataEntryType := detVHFMEM;
284:     drtUHFMEM : frmDataEntry.vdetDataEntryType := detUHFMEM;
285:  //   drtDTMF :
286:   end;
287:
288:   if gvintSelectedRow = 0 then gvintSelectedRow := 1;
289:   frmDataEntry.vbytChannelNumber := gvintSelectedRow;
290:   frmDataEntry.ShowModal;
291:
292: end;// procedure TfrmMEM.bbtEditClick
293:
294: //---------------------------------------------------------------------------------------------
295: procedure TfrmMEM.bbtSelectClick(Sender: TObject);
```

```
296: begin
297:   SetVHFChannel;
298:   ModalResult := mrClose;
299: end;// procedure TfrmMEM.bbtSelectClick
300:
301: //------------------------------------------------------------------------------
302: procedure TfrmMEM.bbtCloseClick(Sender: TObject);
303: begin
304:
305: end;// procedure TfrmMEM.bbtCloseClick
306:
307: //==============================================================================
308: //          MOUSE ROUTINES
309: //==============================================================================
310: procedure TfrmMEM.sgrUHFMouseUp(Sender: TObject; Button: TMouseButton;
311:   Shift: TShiftState; X, Y: Integer);
312:
313: var
314:   vintCol, vintRow : Longint;
315:
316: begin
317:
318:     sgrUHF.MouseToCell(X, Y, vintCol, vintRow);
319:     if vintRow > 0 then
320:     begin
321:       gvintSelectedRow := vintRow;
322:     end;// if vintRow > 0
323:
324: end;// procedure TfrmMEM.sgrUHFMouseUp
325:
326: //------------------------------------------------------------------------------
327: procedure TfrmMEM.sgrUHFDblClick(Sender: TObject);
328: begin
329:   SetUHFChannel;
330:   ModalResult := mrClose;
331: end;// procedure TfrmMEM.sgrUHFDblClick
332:
333: //------------------------------------------------------------------------------
334: procedure TfrmMEM.sgrVHFMouseUp(Sender: TObject; Button: TMouseButton;
335:   Shift: TShiftState; X, Y: Integer);
336:
337: var
338:   vintCol, vintRow : Longint;
339:
340: begin
341:
342:   sgrVHF.MouseToCell(X, Y, vintCol, vintRow);
343:   if vintRow > 0 then
344:   begin
345:     gvintSelectedRow := vintRow;
346:   end;// if vintRow > 0
347:
348: end;// procedure TfrmMEM.sgrVHFMouseUp
349:
350: //------------------------------------------------------------------------------
351: procedure TfrmMEM.sgrVHFDblClick(Sender: TObject);
352: begin
353:     SetVHFChannel;
354:     ModalResult := mrClose;
```

```
355: end;
356:
357: //============================================================================================
358: end.// unit MEM
359:
```