```
 1: unit Utilities;
 2:
 3: {$mode objfpc}{$H+}
 4:
 5: //============================================================================================
 6: //
 7: //  Utilities.pas
 8: //
 9: //  Calls: AppConstants
10: //         AppTypes
11: //         AppVariables
12: //
13: //  Called By: DataEntry : TfrmDataEntry.bbtSaveClick
14: //                         TfrmDataEntry.edtRXFrequencyExit
15: //             DataEntry_FAV : DataEntry_FAV_Init
16: //             DataEntry_UHFMEM : DataEntry_UHFMEM_Init
17: //             DataEntry_VHFMEM : DataEntry_VHFMEM_Init
18: //             Fav : SetFAVChannel
19: //             LCDDisplay : DisplayUHFCTStatus
20: //                          DisplayVHFCTStatus
21: //             Mem_VHF : LoadVHFStringGrid
22: //             Mem_UHF : LoadUHFStringGrid
23: //             TMVFiles_FAV : MakeFAVRecord
24: //             TMVFiles_VHF : ParseVHFRecord
25: //
26: //  Ver: 1.0.0
27: //
28: //  Date: 6 Apr 2014
29: //
30: //============================================================================================
31:
32: interface
33:
34: uses
35:   Classes, Dialogs, SysUtils,
36:   // Application Units
37:   AppConstants, AppTypes, AppVariables;
38:
39: procedure DisplayDataArray(vstrArrayType : TDataRecordType; bytRecNr : Byte);
40: procedure DisplayUHFBuffer;
41: procedure DisplayVHFBuffer;
42: function GetStepIndex(vfltStepSize : Real) : Byte;
43: function GetStepSize(vbytStepIndex : Byte) : Real;
44: function GetToneIndexFromToneNr(vbytToneNr : Byte ) : Byte;
45: function GetToneFrequencyFromToneNr(vbytToneNr : Byte ) : string;
46: function GetToneNrFromFrequency(vstrFrequency : string) : Byte;
47: function GetToneNrFromIndex(vbytToneIndex : Byte) : Byte;
48: function ValidUHFFrequency(vstrUHFFrequency : string) : Boolean;
49: function ValidVHFFrequency(vstrVHFFrequency : string) : Boolean;
50:
51: implementation
52:
53: //============================================================================================
54: //     STEP ARRAY ROUTINES
55: //============================================================================================
56: function GetStepSize(vbytStepIndex : Byte) : Real;
57: begin
58:
59: end;// function GetStepSize
60:
```

```
 61: //---------------------------------------------------------------------------------------
 62: function GetStepIndex(vfltStepSize : Real) : Byte;
 63:
 64: var
 65:   vbytTemp : Byte;
 66:
 67: begin
 68:
 69:   for vbytTemp := 0 to 9 do
 70:   begin
 71:     if FloatToStr(vfltStepSize) = gvstrStepArray[vbytTemp] then
 72:     begin
 73:       Result := vbytTemp;
 74:       Exit;
 75:     end;// if FloatToStr(vfltStepSize) = gvstrStepArray[vbytTemp]
 76:   end;// for vbytTemp := 0 to 9 do
 77:
 78: end;// GetStepIndex
 79:
 80: //=======================================================================================
 81: //     TONE ARRAY ROUTINES
 82: //=======================================================================================
 83: function GetToneIndexFromToneNr(vbytToneNr : Byte ) : Byte;
 84: // This function returns the correct Index into the Tone Combobox string table for the
 85: // Tone Number used by the TMV7 passed as vbytToneNr.
 86: //
 87: //  Tone Nr         Index
 88: //     1              0
 89: //     2 (not used)
 90: //     3..39          1..38
 91: begin
 92:
 93:   Case vbytToneNr of
 94:     1 : Result := 0;
 95:    // 2 (not used)
 96:     3..39 : Result := vbytToneNr - 2; // 1 to 37
 97:   end;
 98: end;// function GetToneIndex(vbytToneNr : Byte ) : Byte;
 99:
100: //---------------------------------------------------------------------------------------
101: function GetToneNrFromIndex(vbytToneIndex : Byte) : Byte;
102: // This function returns the correct Tone Number used by the TMV7 for the Tone Combobx
103: // string table Index passed as vbytToneIndex.
104: //
105: //  Index            Tone Nr
106: //    0                1
107: //                     2 Not Used
108: //    1..37            3..39
109: begin
110:
111:   case vbytToneIndex of
112:     0 : Result := vbytToneIndex + 1;
113:     1..37 : Result := vbytToneIndex + 2;
114:   end;
115:
116: end;// function GetToneNrFromIndex(vbytToneIndex) : Byte;
117:
118: //---------------------------------------------------------------------------------------
119: function GetToneFrequencyFromToneNr(vbytToneNr : Byte ) : string;
120: begin
```

```
121:   Result := gvstrToneArray[GetToneIndexFromToneNr(vbytToneNr)]
122: end;// function GetToneFrequency(vbytToneNr)
123:
124: //-----------------------------------------------------------------------------------------
125: function GetToneNrFromFrequency(vstrFrequency : string) : Byte;
126:
127: var
128:   vbytTemp : Byte;
129:
130: begin
131:
132:   // vfltFrequency contains the Tone Frequency as a Real. We have to search the Tone Array to
133:   // determine the array index and then convert that into the correct tone number.
134:   //
135:   //  vbytTemp            Tone Nr
136:   //    0                1
137:   //                     2 Not Used
138:   //    1..37            3..39
139:   for vbytTemp := 0 to gcbytMaxToneIndex do
140:   begin
141:     if vstrFrequency = gvstrToneArray[vbytTemp] then
142:     begin
143:       if vbytTemp = 0 then
144:         Result := vbytTemp + 1
145:       else Result := vbytTemp + 2;
146:       Exit;
147:     end;// if vstrTFreq = gvstrToneArray[vbytTemp]
148:   end;// for vbytTemp := 0 to gcbytMaxToneIndex do
149:
150: end;// function GetToneNrFromFrequency
151:
152: //=========================================================================================
153: //     VALIDATION ROUTINES
154: //=========================================================================================
155: function ValidVHFFrequency(vstrVHFFrequency : string) : Boolean;
156:
157: var
158:   vsngFrequency : Single;
159:
160: begin
161:
162:   Result := True;
163:
164:   if Length(vstrVHFFrequency) <> 7 then
165:   begin
166:     Result := False;
167:     Exit;
168:   end;
169:
170:   vsngFrequency := StrToFloat(vstrVHFFrequency);
171:
172:   if (vsngFrequency < gcsngMinVHFFrequency) or
173:      (vsngFrequency > gcsngMaxVHFFrequency) then
174:       Result := False;
175:
176: end;// function ValidVHFFrequency : Boolean;
177:
178: //-----------------------------------------------------------------------------------------
179: function ValidUHFFrequency(vstrUHFFrequency : string) : Boolean;
180:
```

```
var
  vsngFrequency : Single;

begin

  Result := True;

  if Length(vstrUHFFrequency) <> 7 then
  begin
    Result := False;
    Exit;
  end;

  vsngFrequency := StrToFloat(vstrUHFFrequency);

  if (vsngFrequency < gcsngMinUHFFrequency) or
     (vsngFrequency > gcsngMaxUHFFrequency) then
       Result := False;

end;// function ValidUHFFrequency : Boolean;

//=============================================================================================
procedure DisplayDataArray(vstrArrayType : TDataRecordType; bytRecNr : Byte);
begin

  case vstrArrayType of
    drtVHFMEM :
      ShowMessage('VHF MEMORY array - Record - ' + IntToStr(bytRecNr) +
      #13 +
//      'gcbytChannelNrField = ' + gvstrVHFChannelDataArray[bytRecNr, gcbytChannelNrField] +
//      #13 +
      'gcbytVFOField = ' + gvstrVHFChannelDataArray[bytRecNr, gcbytVFOField] +
      #13 +
      'gcbytRXFrequencyField = ' + gvstrVHFChannelDataArray[bytRecNr, gcbytRXFrequencyField] +
      #13 +
      'gcbytStepField = ' + gvstrVHFChannelDataArray[bytRecNr, gcbytStepField] +
      #13 +
      'gcbytShiftField = ' + gvstrVHFChannelDataArray[bytRecNr, gcbytShiftField] +
      #13 +
      'gcbytReverseField = ' + gvstrVHFChannelDataArray[bytRecNr, gcbytReverseField] +
      #13 +
      'gcbytToneField = ' + gvstrVHFChannelDataArray[bytRecNr, gcbytToneField] +
      #13 +
      'gcbytCTCSSField = ' + gvstrVHFChannelDataArray[bytRecNr, gcbytCTCSSField] +
      #13 +
      'gcbytDTSSField = ' + gvstrVHFChannelDataArray[bytRecNr, gcbytDTSSField] +
      #13 +
      'gcbytToneNrField = ' + gvstrVHFChannelDataArray[bytRecNr, gcbytToneNrField] +
      #13 +
      'gcbytDTSSCodeField = ' + gvstrVHFChannelDataArray[bytRecNr, gcbytDTSSCodeField] +
      #13 +
      'gcbytCTCSSNrField = ' + gvstrVHFChannelDataArray[bytRecNr, gcbytCTCSSNrField] +
      #13 +
      'gcbytShiftOffsetField = ' + gvstrVHFChannelDataArray[bytRecNr, gcbytShiftOffsetField] +
      #13 +
      'gcbytScanField = ' + gvstrVHFChannelDataArray[bytRecNr, gcbytScanField] +
      #13 +
      'gcbytRFPowerField = ' + gvstrVHFChannelDataArray[bytRecNr, gcbytRFPowerField] +
      #13 +
      'gcbytChannelNameField = ' + gvstrVHFChannelDataArray[bytRecNr, gcbytChannelNameField] +
```

```
241:       #13 +
242:       'gcbytCommentsField = ' + gvstrVHFChannelDataArray[bytRecNr, gcbytCommentsField]
243:       );
244:
245:     drtUHFMEM :
246:       ShowMessage('UHF MEMORY array - Record - ' + IntToStr(bytRecNr) +
247:       #13 +
248:  //     'gcbytChannelNrField = ' + gvstrUHFChannelDataArray[bytRecNr, gcbytChannelNrField] +
249:  //     #13 +
250:       'gcbytVFOField = ' + gvstrUHFChannelDataArray[bytRecNr, gcbytVFOField] +
251:       #13 +
252:       'gcbytRXFrequencyField = ' + gvstrUHFChannelDataArray[bytRecNr, gcbytRXFrequencyField] +
253:       #13 +
254:       'gcbytStepField = ' + gvstrUHFChannelDataArray[bytRecNr, gcbytStepField] +
255:       #13 +
256:       'gcbytShiftField = ' + gvstrUHFChannelDataArray[bytRecNr, gcbytShiftField] +
257:       #13 +
258:       'gcbytReverseField = ' + gvstrUHFChannelDataArray[bytRecNr, gcbytReverseField] +
259:       #13 +
260:       'gcbytToneField = ' + gvstrUHFChannelDataArray[bytRecNr, gcbytToneField] +
261:       #13 +
262:       'gcbytCTCSSField = ' + gvstrUHFChannelDataArray[bytRecNr, gcbytCTCSSField] +
263:       #13 +
264:       'gcbytDTSSField = ' + gvstrUHFChannelDataArray[bytRecNr, gcbytDTSSField] +
265:       #13 +
266:       'gcbytToneNrField = ' + gvstrUHFChannelDataArray[bytRecNr, gcbytToneNrField] +
267:       #13 +
268:       'gcbytDTSSCodeField = ' + gvstrUHFChannelDataArray[bytRecNr, gcbytDTSSCodeField] +
269:       #13 +
270:       'gcbytCTCSSNrField = ' + gvstrUHFChannelDataArray[bytRecNr, gcbytCTCSSNrField] +
271:       #13 +
272:       'gcbytShiftOffsetField = ' + gvstrUHFChannelDataArray[bytRecNr, gcbytShiftOffsetField] +
273:       #13 +
274:       'gcbytScanField = ' + gvstrUHFChannelDataArray[bytRecNr, gcbytScanField] +
275:       #13 +
276:       'gcbytRFPowerField = ' + gvstrUHFChannelDataArray[bytRecNr, gcbytRFPowerField] +
277:       #13 +
278:       'gcbytChannelNameField = ' + gvstrUHFChannelDataArray[bytRecNr, gcbytChannelNameField] +
279:       #13 +
280:       'gcbytCommentsField = ' + gvstrUHFChannelDataArray[bytRecNr, gcbytCommentsField]
281:       );
282:
283:     drtFAV :
284:       ShowMessage('FAV MEMORY array - Record - ' + IntToStr(bytRecNr) +
285:       #13 +
286: //      'gcbytChannelNrField = ' + gvstrFAVChannelDataArray[bytRecNr, gcbytChannelNrField] +
287: //      #13 +
288:       'gvstrVFOField = ' + gvstrFAVChannelDataArray[bytRecNr, gcbytVFOField] +
289:       #13 +
290:       'gcbytRXFrequencyField = ' + gvstrFAVChannelDataArray[bytRecNr, gcbytRXFrequencyField] +
291:       #13 +
292:       'gcbytStepField = ' + gvstrFAVChannelDataArray[bytRecNr, gcbytStepField] +
293:       #13 +
294:       'gcbytShiftField = ' + gvstrFAVChannelDataArray[bytRecNr, gcbytShiftField] +
295:       #13 +
296:       'gcbytReverseField = ' + gvstrFAVChannelDataArray[bytRecNr, gcbytReverseField] +
297:       #13 +
298:       'gcbytToneField = ' + gvstrFAVChannelDataArray[bytRecNr, gcbytToneField] +
299:       #13 +
300:       'gcbytCTCSSField = ' + gvstrFAVChannelDataArray[bytRecNr, gcbytCTCSSField] +
```

```
301:       #13 +
302:       'gcbytDTSSField = ' + gvstrFAVChannelDataArray[bytRecNr, gcbytDTSSField] +
303:       #13 +
304:       'gcbytToneNrField = ' + gvstrFAVChannelDataArray[bytRecNr, gcbytToneNrField] +
305:       #13 +
306:       'gcbytDTSSCodeField = ' + gvstrFAVChannelDataArray[bytRecNr, gcbytDTSSCodeField] +
307:       #13 +
308:       'gcbytCTCSSNrField = ' + gvstrFAVChannelDataArray[bytRecNr, gcbytCTCSSNrField] +
309:       #13 +
310:       'gcbytShiftOffsetField = ' + gvstrFAVChannelDataArray[bytRecNr, gcbytShiftOffsetField] +
311:       #13 +
312:       'gcbytScanField = ' + gvstrFAVChannelDataArray[bytRecNr, gcbytScanField] +
313:       #13 +
314:       'gcbytRFPowerField = ' + gvstrFAVChannelDataArray[bytRecNr, gcbytRFPowerField] +
315:       #13 +
316:       'gcbytChannelNameField = ' + gvstrFAVChannelDataArray[bytRecNr, gcbytChannelNameField] +
317:       #13 +
318:       'gcbytCommentsField = ' + gvstrFAVChannelDataArray[bytRecNr, gcbytCommentsField]
319:        );
320:     drtDTMF :
321:       ShowMessage('DTMF array - Record - ' + IntToStr(bytRecNr)
322:       );
323:   end;// case vstrArrayType of
324:
325: end;// procedure DisplayDataArray
326:
327: //---------------------------------------------------------------------------------------
328:
329: procedure DisplayUHFBuffer;
330: begin
331:
332:   ShowMessage('UHF Buffer' +
333:               #13 +
334:               'gvstrUHFDataSource  = ' + gvstrUHFDataSource +
335:               #13 +
336: //              'gvstrUHFChannelNr = ' + gvstrUHFChannelNr +
337: //              #13 +
338:               'gvstrUHFRXFrequency = ' + gvstrUHFRXFrequency +
339:               #13 +
340:               'gvstrUHFStep = ' + gvstrUHFStep +
341:               #13 +
342:               'gvstrUHFShift = ' + gvstrUHFShift +
343:               #13 +
344:               'gvstrUHFReverse = ' + gvstrUHFReverse +
345:               #13 +
346:               'gvstrUHFTone = ' + gvstrUHFTone +
347:               #13 +
348:               'gvstrUHFCTCSS = ' + gvstrUHFCTCSS +
349:               #13 +
350:               'gvstrUHFDTSS = ' + gvstrUHFDTSS +
351:               #13 +
352:               'gvstrUHFToneNr = ' + gvstrUHFToneNr +
353:               #13 +
354:               'gvstrUHFDTSSCode = ' + gvstrUHFDTSSCode +
355:               #13 +
356:               'gvstrUHFCTCSSNr = ' + gvstrUHFCTCSSNr +
357:               #13 +
358:               'gvstrUHFOffset = ' + gvstrUHFOffset +
359:               #13 +
360:               'gvstrUHFScan = ' + gvstrUHFScan +
```

```
361:                #13 +
362:                'gvstrUHFSplitFrequency = ' + gvstrUHFSplitFrequency +
363:                #13 +
364:                'gvstrUHFSplitStep = ' + gvstrUHFSplitStep +
365:                #13 +
366:                'gvstrUHFRFPower = ' + gvstrUHFRFPower +
367:                #13 +
368:                'gvstrUHFChannelName = ' + gvstrUHFChannelName +
369:                #13 +
370:                'gvstrUHFChannelComments = ' + gvstrUHFChannelComments +
371:                #13 +
372:                'gvstrUHFAudioLevel = ' + gvstrUHFAudioLevel +
373:                #13 +
374:                'gvstrUHFSquelchLevel = ' + gvstrUHFSquelchLevel
375:                );
376:
377: end;// procedure DisplayUHFBuffer;
378:
379: //-------------------------------------------------------------------------------------------
380: procedure DisplayVHFBuffer;
381: begin
382:
383:     ShowMessage('VHF Buffer' +
384:                #13 +
385:                'gvstrVHFDataSource = ' + gvstrVHFDataSource +
386:                #13 +
387:    //            'gvstrVHFChannelNr = ' + gvstrVHFChannelNr +
388:    //            #13 +
389:                'gvstrVHFRXFrequency = ' + gvstrVHFRXFrequency +
390:                #13 +
391:                'gvstrVHFStep = ' + gvstrVHFStep +
392:                #13 +
393:                'gvstrVHFShift = ' + gvstrVHFShift +
394:                #13 +
395:                'gvstrVHFReverse = ' + gvstrVHFReverse +
396:                #13 +
397:                'gvstrVHFTone = ' + gvstrVHFTone +
398:                #13 +
399:                'gvstrVHFCTCSS = ' + gvstrVHFCTCSS +
400:                #13 +
401:                'gvstrVHFDTSS = ' + gvstrVHFDTSS +
402:                #13 +
403:                'gvstrVHFToneNr = ' + gvstrVHFToneNr +
404:                #13 +
405:                'gvstrVHFDTSSCode = ' + gvstrVHFDTSSCode +
406:                #13 +
407:                'gvstrVHFCTCSSNr = ' + gvstrVHFCTCSSNr +
408:                #13 +
409:                'gvstrVHFOffset = ' + gvstrVHFOffset +
410:                #13 +
411:                'gvstrVHFScan = ' + gvstrVHFScan +
412:                #13 +
413:                'gvstrVHFSplitFrequency = ' + gvstrVHFSplitFrequency +
414:                #13 +
415:                'gvstrVHFSplitStep = ' + gvstrVHFSplitStep  +
416:                #13 +
417:                'gvstrVHFRFPower = ' + gvstrUHFRFPower +
418:                #13 +
419:                'gvstrVHFChannelName = ' + gvstrVHFChannelName +
420:                #13 +
```

```
421:                'gvstrVHFChannelComments = ' + gvstrVHFChannelComments +
422:                #13 +
423:                'gvstrVHFAudioLevel = ' + gvstrVHFAudioLevel +
424:                #13 +
425:                'gvstrVHFSquelchLevel = ' + gvstrVHFSquelchLevel
426:                );
427:
428: end;// procedure DisplayVHFBuffer;
429:
430: //=============================================================================================
431: end.// unit Utilities;
432:
```