

```

1: unit Main;
2:
3: {$mode delphi}
4:
5: //=====
6: //
7: //  Main.pas
8: //
9: //  Calls:  AGCommand : SetVHF_AGValue
10: //           SetUHF_AGValue
11: //           AppConstants
12: //           AppVariables
13: //           AppTypes
14: //           BCCCommand
15: //           ColourSchemes : SetColourScheme
16: //           COMPort
17: //           DataEntry : TfrmDataEntry.bbtSaveClick
18: //           Fav : SetFavChannel(n)
19: //           Final : Finalize
20: //           INISTuff : ReadINIFile
21: //                       WriteINIFile
22: //           Init : Initialize
23: //           LCDDisplay : LCDOn
24: //                       LCD Off
25: //           Mem : frmMem.Setup
26: //           Mute : ToggleMute
27: //           PCCCommand : ToggleRFPower
28: //           psCommand : TogglePowerOnOff
29: //           RadioStatus : RadioStatusOff
30: //           ReportTMV7File
31: //           ResponseParser : ParseResponse
32: //           Reverse : ToggleReverse
33: //           SMCommand : SMResponseHandler
34: //           SplashAbout : TdlgSplashAbout.ShowAbout
35: //                       TdlgSplashAbout.ShowSplash
36: //           SQCommand : SetVHF_SQValue
37: //                       SetUHF_SQValue
38: //           StatusBar : DisplayTimeStatus
39: //                       DisplayDateStatus
40: //                       DisplayCOMPortStatus
41: //                       DisplayTMV7Status
42: //           TMVFileReport : PrintTMVFileReport
43: //           TMVFiles : NewTMVFile
44: //                       OpenTMVFile
45: //                       SaveTMVFile
46: //                       SaveTMVFileAs
47: //
48: //  Called By: DataEntry_FAV : DataEntry_FAV_Save
49: //           Final : Finalize
50: //           Init : Initialize
51: //           Mute : SetMuteOn
52: //                       SetMuteOff
53: //           Register : TdlgRegister.bbtOkClick
54: //                       TdlgRegister.bbtCancelClick
55: //           SerialStuff : OpenPort
56: //
57: //  Ver: 1.0.0
58: //
59: //  Date: 24 Apr 2014
60: //

```

```

61: //=====
62:
63: interface
64:
65: uses
66:   Classes, SysUtils, FileUtil, PrintersDlgs, RackCtrls, uEKnob, ueled,
67:   uERotImage, SdpoSerial, Forms, Controls, Graphics, Dialogs, Menus, NagScreen,
68:   ComCtrls, Printers, StdCtrls, ExtCtrls, Buttons, Messages, Windows, DOS, LR_Class,
69:   LR_DSet,
70:   // Application units
71:   AGCommand, AppConstants, AppVariables, AppTypes, BCCCommand, ColourSchemes, COMPort,
72:   DataEntry, Fav, Final, Init, Mem, Mute, PCCCommand, PSCommand, Register, ResponseParser,
73:   Reverse, SplashAbout, SQCommand, StatusBar, TMVFileReport, TMVFiles;
74:
75: type
76:
77:   { TfrmMain }
78:
79:   TfrmMain = class(TForm)
80:     bbtFav01: TBitBtn;
81:     bbtFav10: TBitBtn;
82:     bbtFav11: TBitBtn;
83:     bbtFav12: TBitBtn;
84:     bbtFav02: TBitBtn;
85:     bbtFav03: TBitBtn;
86:     bbtFav04: TBitBtn;
87:     bbtFav05: TBitBtn;
88:     bbtFav06: TBitBtn;
89:     bbtFav07: TBitBtn;
90:     bbtFav08: TBitBtn;
91:     bbtFav09: TBitBtn;
92:     bbtMute: TBitBtn;
93:     bbtReverse: TBitBtn;
94:     btnPower: TButton;
95:     btnBand: TButton;
96:     btnRFPower: TButton;
97:     gbxFavourites: TGroupBox;
98:     ImageList1: TImageList;
99:     Label1: TLabel;
100:    Label2: TLabel;
101:    Label4: TLabel;
102:    Label5: TLabel;
103:    lblVHFOnAirBusy: TLabel;
104:    lblVHFChannelName: TLabel;
105:    lblUHFCChannelName: TLabel;
106:    lblVHFChannelNr: TLabel;
107:    lblUHFCChannelNr: TLabel;
108:    lblVHFDataSource: TLabel;
109:    lblUHFDDataSource: TLabel;
110:    lblVHFDTSSCode: TLabel;
111:    lblUHFDTSSTSSCode: TLabel;
112:    lblVHFFreq: TLabel;
113:    lblVHFDTSS: TLabel;
114:    lblUHFDTSSTSS: TLabel;
115:    lblUHFFreq: TLabel;
116:    lblUHFOOnAirBusy: TLabel;
117:    lblVHFShift: TLabel;
118:    lblVHFReverse: TLabel;
119:    lblUHFRReverse: TLabel;
120:    lblUHFSShift: TLabel;

```

```
121:      lblVHFTCTFreq: TLabel;
122:      lblVHFTCT: TLabel;
123:      lblVHFRFPwr: TLabel;
124:      lblVHFPTT: TLabel;
125:      lblUHFPTT: TLabel;
126:      lblUHFRFPwr: TLabel;
127:      lblUHFTCT: TLabel;
128:      lblUHFTCTFreq: TLabel;
129:      MainMenu1: TMainMenu;
130:      mnuConfigClosePort: TMenuItem;
131:      mnuConfigOpenPort: TMenuItem;
132:      mnuConfigSelectPort: TMenuItem;
133:      mnuFilePrintTMVFileReport: TMenuItem;
134:      mnuMemDTMF: TMenuItem;
135:      mnuMemUHF: TMenuItem;
136:      mnuMemVHF: TMenuItem;
137:      mnuConfigColourScheme1: TMenuItem;
138:      mnuConfigColourScheme10: TMenuItem;
139:      mnuConfigColourScheme2: TMenuItem;
140:      mnuConfigColourScheme3: TMenuItem;
141:      mnuConfigColourScheme4: TMenuItem;
142:      mnuConfigColourScheme5: TMenuItem;
143:      mnuConfigColourScheme6: TMenuItem;
144:      mnuConfigColourScheme7: TMenuItem;
145:      mnuConfigColourScheme8: TMenuItem;
146:      mnuConfigColourScheme9: TMenuItem;
147:      mnuFileSave: TMenuItem;
148:      mnuFileNew: TMenuItem;
149:      mnuHelpAbout: TMenuItem;
150:      mnuHelpRegistration: TMenuItem;
151:      mnuHelpContents: TMenuItem;
152:      mnuHelp: TMenuItem;
153:      mnuMemory: TMenuItem;
154:      mnuVFOUHF: TMenuItem;
155:      mnuVFOVHF: TMenuItem;
156:      mnuVFO: TMenuItem;
157:      mnuConfigColourSchemes: TMenuItem;
158:      mnuConfigCOMPort: TMenuItem;
159:      mnuConfigure: TMenuItem;
160:      mnuFileSep2: TMenuItem;
161:      mnuFilePrint: TMenuItem;
162:      mnuFilePrinterSetup: TMenuItem;
163:      mnuFileSep1: TMenuItem;
164:      mnuFileSaveAs: TMenuItem;
165:      mnuFileOpen: TMenuItem;
166:      mnuFileExit: TMenuItem;
167:      mnuFile: TMenuItem;
168:      OpenDialog1: TOpenDialog;
169:      PageSetupDialog1: TPageSetupDialog;
170:      pbrUHFSMeter: TProgressBar;
171:      pnlLCD: TPanel;
172:      pnlLCDBorder: TPanel;
173:      pbrVHFSMeter: TProgressBar;
174:      PrintDialog1: TPrintDialog;
175:      PrinterSetupDialog1: TPrinterSetupDialog;
176:      SaveDialog1: TSaveDialog;
177:      SdpoSerial1: TSdpoSerial;
178:      stxOffLine: TStaticText;
179:      StatusBar1: TStatusBar;
180:      tmrSendTimeout: TTimer;
```

```
181: tmrTimeStatus: TTimer;
182: ToolBar1: TToolBar;
183: tbtFileOpen: TToolButton;
184: tbtFilePrint: TToolButton;
185: ToolButton1: TToolButton;
186: uekVHFVolume: TuEKnob;
187: uekVHFSquelch: TuEKnob;
188: uekUHFVolume: TuEKnob;
189: uekUHFSquelch: TuEKnob;
190: updVHF: TUpDown;
191: updUHF: TUpDown;
192: procedure bbtFav01MouseUp(Sender: TObject; Button: TMouseButton;
193:   Shift: TShiftState; X, Y: integer);
194: procedure bbtFav02MouseUp(Sender: TObject; Button: TMouseButton;
195:   Shift: TShiftState; X, Y: integer);
196: procedure bbtFav03MouseUp(Sender: TObject; Button: TMouseButton;
197:   Shift: TShiftState; X, Y: integer);
198: procedure bbtFav04MouseUp(Sender: TObject; Button: TMouseButton;
199:   Shift: TShiftState; X, Y: integer);
200: procedure bbtFav05MouseUp(Sender: TObject; Button: TMouseButton;
201:   Shift: TShiftState; X, Y: integer);
202: procedure bbtFav06MouseUp(Sender: TObject; Button: TMouseButton;
203:   Shift: TShiftState; X, Y: integer);
204: procedure bbtFav07MouseUp(Sender: TObject; Button: TMouseButton;
205:   Shift: TShiftState; X, Y: integer);
206: procedure bbtFav08MouseUp(Sender: TObject; Button: TMouseButton;
207:   Shift: TShiftState; X, Y: integer);
208: procedure bbtFav09MouseUp(Sender: TObject; Button: TMouseButton;
209:   Shift: TShiftState; X, Y: integer);
210: procedure bbtFav10MouseUp(Sender: TObject; Button: TMouseButton;
211:   Shift: TShiftState; X, Y: integer);
212: procedure bbtFav11MouseUp(Sender: TObject; Button: TMouseButton;
213:   Shift: TShiftState; X, Y: integer);
214: procedure bbtFav12MouseUp(Sender: TObject; Button: TMouseButton;
215:   Shift: TShiftState; X, Y: integer);
216: procedure bbtMuteClick(Sender: TObject);
217: procedure bbtReverseClick(Sender: TObject);
218: procedure btnBandClick(Sender: TObject);
219: procedure btnPowerClick(Sender: TObject);
220: procedure btnRFPowerClick(Sender: TObject);
221: procedure FormActivate(Sender: TObject);
222: procedure FormClose(Sender: TObject; var CloseAction: TCloseAction);
223: procedure FormShow(Sender: TObject);
224: procedure mnuConfigClosePortClick(Sender: TObject);
225: procedure mnuConfigOpenPortClick(Sender: TObject);
226: procedure mnuConfigSelectPortClick(Sender: TObject);
227: procedure mnuFilePrintTMVFileReportClick(Sender: TObject);
228: procedure mnuConfigColourScheme10Click(Sender: TObject);
229: procedure mnuConfigColourScheme1Click(Sender: TObject);
230: procedure mnuConfigColourScheme2Click(Sender: TObject);
231: procedure mnuConfigColourScheme3Click(Sender: TObject);
232: procedure mnuConfigColourScheme4Click(Sender: TObject);
233: procedure mnuConfigColourScheme5Click(Sender: TObject);
234: procedure mnuConfigColourScheme6Click(Sender: TObject);
235: procedure mnuConfigColourScheme7Click(Sender: TObject);
236: procedure mnuConfigColourScheme8Click(Sender: TObject);
237: procedure mnuConfigColourScheme9Click(Sender: TObject);
238: procedure mnuFileExitClick(Sender: TObject);
239: procedure mnuFileNewClick(Sender: TObject);
240: procedure mnuFileOpenClick(Sender: TObject);
```

```

241:     procedure mnuFilePrinterSetupClick(Sender: TObject);
242:     procedure mnuFileSaveAsClick(Sender: TObject);
243:     procedure mnuFileSaveClick(Sender: TObject);
244:     procedure mnuHelpAboutClick(Sender: TObject);
245:     procedure mnuHelpContentsClick(Sender: TObject);
246:     procedure mnuHelpRegistrationClick(Sender: TObject);
247:     procedure mnuMemDTMFClick(Sender: TObject);
248:     procedure mnuMemUHFClick(Sender: TObject);
249:     procedure mnuMemVHFClick(Sender: TObject);
250:     procedure mnuVFOUHFClick(Sender: TObject);
251:     procedure mnuVFOVHFClick(Sender: TObject);
252:     procedure SdpoSerial1RxData(Sender: TObject);
253:     procedure tmrSendTimeoutTimer(Sender: TObject);
254:     procedure tmrTimeStatusTimer(Sender: TObject);
255:     procedure uekUHFSquelchMouseUp(Sender: TObject; Button: TMouseButton;
256:         Shift: TShiftState; X, Y: integer);
257:     procedure uekUHFVolumeMouseUp(Sender: TObject; Button: TMouseButton;
258:         Shift: TShiftState; X, Y: integer);
259:     procedure uekVHFSquelchMouseUp(Sender: TObject; Button: TMouseButton;
260:         Shift: TShiftState; X, Y: integer);
261:     procedure uekVHFVolumeMouseUp(Sender: TObject; Button: TMouseButton;
262:         Shift: TShiftState; X, Y: integer);
263: //     procedure DisplayTitleBar;
264: private
265:     { private declarations }
266: public
267:     { public declarations }
268:     procedure DisplayTitleBar;
269: end;
270:
271: procedure SendTimeoutTimerReset;
272:
273: var
274:     frmMain: TfrmMain;
275:
276: implementation
277:
278: {$R *.lfm}
279:
280: //=====
281: //          FORM ROUTINES
282: //=====
283: procedure TfrmMain.FormShow(Sender: TObject);
284: begin
285:
286:     dlgSplashAbout.ShowSplash;
287:
288:     if not Initialize then
289:     begin
290:         MessageDlg('Init Failure', mtInformation, [mbOK], 0);
291:         Finalize;
292:         Application.Terminate;
293:     end;// if not Initialize then
294:
295:     if not gvblnRegistered then
296:     begin
297:         if dlgNagScreen.ShowModal = mrOk then
298:             dlgRegister.ShowModal;
299:     end;// if not gvblnRegistered
300:

```

```
301: end; // procedure TfrmMain.FormShow
302:
303: //-----
304: procedure TfrmMain.FormActivate(Sender: TObject);
305: begin
306:     // Update the Title Bar
307:     DisplayTitleBar;
308:     // Display the Status Bar fields
309:     DisplayDateStatus;
310:     DisplayTimeStatus;
311:     DisplayCOMPortStatus;
312:     DisplayTMV7Status;
313: end; // procedure TfrmMain.FormActivate
314:
315: //-----
316: procedure TfrmMain.FormClose(Sender: TObject; var CloseAction: TCloseAction);
317: begin
318:     Finalize;
319: end; // procedure TfrmMain.FormClose
320:
321: //=====
322: procedure TfrmMain.DisplayTitleBar;
323: begin
324:     Caption := gcstrAppName + ' - Ver: ' + gcstrAppVersion;
325:     if gvblnRegistered then
326:         Caption := Caption + ' Registered to ' +
327:             gvstrRegisteredName + ' ' + gvstrRegisteredCall
328:     else
329:         Caption := Caption + ' Unregistered';
330: end; // procedure TfrmMain.DisplayTitleBar
331:
332: //=====
333: //          MENU ROUTINES
334: //=====
335:
336: //-----
337: //          FILE MENU
338: //-----
339: procedure TfrmMain.mnuFileNewClick(Sender: TObject);
340: begin
341:
342:     if MessageDlg('This will erase all data. Continue ?', mtConfirmation,
343:         [mbYes, mbNo], 0) = mrYes then
344:         begin
345:
346:             SaveDialog1.Title := 'Create a New TMV File';
347:             SaveDialog1.InitialDir := gvstrAppPath + gcstrTMV7FileDir;
348:             SaveDialog1.Filter := 'TMVFiles|*.tmv';
349:             SaveDialog1.DefaultExt := 'tmv';
350:
351:             if SaveDialog1.Execute then
352:                 begin
353:                     gvstrTMVFileName := SaveDialog1.FileName;
354:                     NewTMVFile;
355:                 end
356:             else
357:                 ShowMessage('Cancelled');
358:
359:         end; // If MessageDlg('This will erase all data. Continue ?'
360:
```

```
361: end; // procedure TfrmMain.mnuFileNewClick
362:
363: //-----
364: procedure TfrmMain.mnuFileOpenClick(Sender: TObject);
365: begin
366:   OpenTMVFile('');
367: end; // procedure TfrmMain.mnuFileOpenClick(Sender: TObject);
368:
369: //-----
370: procedure TfrmMain.mnuFileSaveClick(Sender: TObject);
371: begin
372:   SaveTMVFile;
373: end; // procedure TfrmMain.mnuFileSaveClick
374:
375: //-----
376: procedure TfrmMain.mnuFileSaveAsClick(Sender: TObject);
377: begin
378:   SaveTMVFileAs;
379: end; // procedure TfrmMain.mnuFileSaveAsClick
380:
381: //-----
382: procedure TfrmMain.mnuFilePrinterSetupClick(Sender: TObject);
383: begin
384:
385:   PrinterSetupDialog1.Execute;
386:   showmessage(Printer.PrinterName);
387:
388: end; // procedure TfrmMain.mnuFilePrinterSetupClick
389:
390: //-----
391: procedure TfrmMain.mnuFilePrintTMVFileReportClick(Sender: TObject);
392: begin
393:
394:   if not OpenDialog1.Execute then
395:     Exit;
396:
397:   frmTMVFileReport.CreateTmpFiles(OpenDialog1.FileName);
398:
399: //   frmTMVFileReport.Show;
400:   frmTMVFileReport.frmTMVFileReport.ShowReport;
401:
402:   frmTMVFileReport.sdfVHFDataset.Active := False;
403:   frmTMVFileReport.sdfUHFDataset.Active := False;
404:   frmTMVFileReport.sdfFAVDataset.Active := False;
405:   frmTMVFileReport.sdfDTMFDataset.Active := False;
406:
407: end; // procedure TfrmMain.mnuFilePrintTMV7ReportClick
408:
409: //-----
410: procedure TfrmMain.mnuFileExitClick(Sender: TObject);
411: begin
412:   Close;
413: end; // procedure TfrmMain.mnuFileExitClick
414:
415: //-----
416: //           CONFIGURE MENU
417: //-----
418: procedure TfrmMain.mnuConfigSelectPortClick(Sender: TObject);
419: begin
420:   frmCOMPort.ShowModal;
```

```
421: end; // procedure TfrmMain.mnuConfigureSelectPortClick
422:
423: //-----
424: procedure TfrmMain.mnuConfigOpenPortClick(Sender: TObject);
425: begin
426:
427: end; // procedure TfrmMain.mnuConfigureOpenPortClick
428:
429: //-----
430: procedure TfrmMain.mnuConfigClosePortClick(Sender: TObject);
431: begin
432:
433: end; // procedure TfrmMain.mnuConfigureClosePortClick
434:
435: //-----
436: procedure TfrmMain.mnuConfigColourScheme1Click(Sender: TObject);
437: begin
438:   SetColourScheme('1');
439: end; // procedure TfrmMain.mnuConfigColourScheme1Click
440:
441: //-----
442: procedure TfrmMain.mnuConfigColourScheme2Click(Sender: TObject);
443: begin
444:   SetColourScheme('2');
445: end; // TfrmMain.mnuConfigColourScheme2Click
446:
447: //-----
448: procedure TfrmMain.mnuConfigColourScheme3Click(Sender: TObject);
449: begin
450:   SetColourScheme('3');
451: end; // TfrmMain.mnuConfigColourScheme3Click
452:
453: //-----
454: procedure TfrmMain.mnuConfigColourScheme4Click(Sender: TObject);
455: begin
456:   SetColourScheme('4');
457: end; // TfrmMain.mnuConfigColourScheme4Click
458:
459: //-----
460: procedure TfrmMain.mnuConfigColourScheme5Click(Sender: TObject);
461: begin
462:   SetColourScheme('5');
463: end;
464:
465: //-----
466: procedure TfrmMain.mnuConfigColourScheme6Click(Sender: TObject);
467: begin
468:   SetColourScheme('6');
469: end; // TfrmMain.mnuConfigColourScheme5Click
470:
471: //-----
472: procedure TfrmMain.mnuConfigColourScheme7Click(Sender: TObject);
473: begin
474:   SetColourScheme('7');
475: end; // TfrmMain.mnuConfigColourScheme7Click
476:
477: //-----
478: procedure TfrmMain.mnuConfigColourScheme8Click(Sender: TObject);
479: begin
480:   SetColourScheme('8');
```



```

481: end; // TfrmMain.mnuConfigColourScheme8Click
482:
483: //-----
484: procedure TfrmMain.mnuConfigColourScheme9Click(Sender: TObject);
485: begin
486:   SetColourScheme('9');
487: end; // TfrmMain.mnuConfigColourScheme9Click
488:
489: //-----
490: procedure TfrmMain.mnuConfigColourScheme10Click(Sender: TObject);
491: begin
492:   SetColourScheme('10');
493: end; // TfrmMain.mnuConfigColourScheme10Click
494:
495: //-----
496: //          VFO MENU
497: //-----
498: procedure TfrmMain.mnuVFOVHFClick(Sender: TObject);
499: begin
500:   MessageDlg('VHF VFO', mtInformation, [mbOK], 0);
501: end; // procedure TfrmMain.mnuVFOVHFClick
502:
503: //-----
504: procedure TfrmMain.mnuVFOUHFClick(Sender: TObject);
505: begin
506:   MessageDlg('UHF VFO', mtInformation, [mbOK], 0);
507: end; // procedure TfrmMain.mnuVFOUHFClick
508:
509: //-----
510: //          MEMORY MENU
511: //-----
512: procedure TfrmMain.mnuMemVHFClick(Sender: TObject);
513: begin
514:   frmMem.Setup(drtVHFMEM);
515: end; // procedure TfrmMain.mnuMemVHFClick
516:
517: //-----
518: procedure TfrmMain.mnuMemUHFClick(Sender: TObject);
519: begin
520:   frmMem.Setup(drtUHFMEM);
521: end; // procedure TfrmMain.mnuMemUHFClick
522:
523: //-----
524: procedure TfrmMain.mnuMemDTMFClick(Sender: TObject);
525: begin
526:   frmMem.Setup(drtDTMF);
527: end; // procedure TfrmMain.mnuMemDTMFClick
528:
529: //-----
530: //          HELP MENU
531: //-----
532: procedure TfrmMain.mnuHelpContentsClick(Sender: TObject);
533: begin
534:   Exec('hh.exe', {'-mapid 0 ' +} gcstrAppName + '.chm');
535: end; // procedure TfrmMain.mnuHelpContentsClick
536:
537: //-----
538: procedure TfrmMain.mnuHelpRegistrationClick(Sender: TObject);
539: begin
540:   dlgRegister.ShowModal;

```

```
541: end; // procedure TfrmMain.mnuHelpRegistrationClick
542:
543: //-----
544: procedure TfrmMain.mnuHelpAboutClick(Sender: TObject);
545: begin
546:   dlgSplashAbout.ShowAbout;
547: end; // procedure TfrmMain.mnuHelpAboutClick
548:
549: //=====
550: //          TOOLBAR ROUTINES
551: //=====
552:
553: //=====
554: //          FAVOURITE ROUTINES
555: //=====
556: procedure ProcessFavButton(vbytFavButton: byte);
557: begin
558:   frmDataEntry.vdetDataEntryType := detFAV;
559:   frmDataEntry.vbytChannelNumber := vbytFavButton;
560:   frmDataEntry.ShowModal;
561: end; // procedure ProcessFavButton
562:
563: //-----
564: procedure TfrmMain.bbtFav01MouseUp(Sender: TObject; Button: TMouseButton;
565:   Shift: TShiftState; X, Y: integer);
566: begin
567:   if (Button = mbLeft) and (gvblnTMV7OnLine) then
568:     SetFavChannel(1)
569:   else if Button = mbRight then
570:     ProcessFavButton(1);
571: end; // procedure TfrmMain.bbtFav01MouseUp
572:
573: //-----
574: procedure TfrmMain.bbtFav02MouseUp(Sender: TObject; Button: TMouseButton;
575:   Shift: TShiftState; X, Y: integer);
576: begin
577:   if (Button = mbLeft) and (gvblnTMV7OnLine) then
578:     SetFavChannel(2)
579:   else if Button = mbRight then
580:     ProcessFavButton(2);
581: end; // procedure TfrmMain.bbtFav02MouseUp
582:
583: //-----
584: procedure TfrmMain.bbtFav03MouseUp(Sender: TObject; Button: TMouseButton;
585:   Shift: TShiftState; X, Y: integer);
586: begin
587:   if (Button = mbLeft) and (gvblnTMV7OnLine) then
588:     SetFavChannel(3)
589:   else if Button = mbRight then
590:     ProcessFavButton(3);
591: end; // procedure TfrmMain.bbtFav03MouseUp
592:
593: //-----
594: procedure TfrmMain.bbtFav04MouseUp(Sender: TObject; Button: TMouseButton;
595:   Shift: TShiftState; X, Y: integer);
596: begin
597:   if (Button = mbLeft) and (gvblnTMV7OnLine) then
598:     SetFavChannel(4)
599:   else if Button = mbRight then
600:     ProcessFavButton(4);
```

```
601: end; // procedure TfrmMain.bbtFav04MouseUp
602:
603: //-----
604: procedure TfrmMain.bbtFav05MouseUp(Sender: TObject; Button: TMouseButton;
605:   Shift: TShiftState; X, Y: integer);
606: begin
607:   if (Button = mbLeft) and (gvblnTMV7OnLine) then
608:     SetFavChannel(5)
609:   else if Button = mbRight then
610:     ProcessFavButton(5);
611: end; // procedure TfrmMain.bbtFav05MouseUp
612:
613: //-----
614: procedure TfrmMain.bbtFav06MouseUp(Sender: TObject; Button: TMouseButton;
615:   Shift: TShiftState; X, Y: integer);
616: begin
617:   if (Button = mbLeft) and (gvblnTMV7OnLine) then
618:     SetFavChannel(6)
619:   else if Button = mbRight then
620:     ProcessFavButton(6);
621: end; // procedure TfrmMain.bbtFav06MouseUp
622:
623: //-----
624: procedure TfrmMain.bbtFav07MouseUp(Sender: TObject; Button: TMouseButton;
625:   Shift: TShiftState; X, Y: integer);
626: begin
627:   if (Button = mbLeft) and (gvblnTMV7OnLine) then
628:     SetFavChannel(7)
629:   else if Button = mbRight then
630:     ProcessFavButton(7);
631: end; // procedure TfrmMain.bbtFav07MouseUp
632:
633: //-----
634: procedure TfrmMain.bbtFav08MouseUp(Sender: TObject; Button: TMouseButton;
635:   Shift: TShiftState; X, Y: integer);
636: begin
637:   if (Button = mbLeft) and (gvblnTMV7OnLine) then
638:     SetFavChannel(8)
639:   else if Button = mbRight then
640:     ProcessFavButton(8);
641: end; // procedure TfrmMain.bbtFav08MouseUp
642:
643: //-----
644: procedure TfrmMain.bbtFav09MouseUp(Sender: TObject; Button: TMouseButton;
645:   Shift: TShiftState; X, Y: integer);
646: begin
647:   if (Button = mbLeft) and (gvblnTMV7OnLine) then
648:     SetFavChannel(9)
649:   else if Button = mbRight then
650:     ProcessFavButton(9);
651: end; // procedure TfrmMain.bbtFav09MouseUp
652:
653: //-----
654: procedure TfrmMain.bbtFav10MouseUp(Sender: TObject; Button: TMouseButton;
655:   Shift: TShiftState; X, Y: integer);
656: begin
657:   if (Button = mbLeft) and (gvblnTMV7OnLine) then
658:     SetFavChannel(10)
659:   else if Button = mbRight then
660:     ProcessFavButton(10);
```

```
661: end;// procedure TfrmMain.bbtFav10MouseUp
662:
663: //-----
664: procedure TfrmMain.bbtFav11MouseUp(Sender: TObject; Button: TMouseButton;
665:   Shift: TShiftState; X, Y: integer);
666: begin
667:   if (Button = mbLeft) and (gvblnTMV7OnLine) then
668:     SetFavChannel(11)
669:   else if Button = mbRight then
670:     ProcessFavButton(11);
671: end;// procedure TfrmMain.bbtFav11MouseUp
672:
673: procedure TfrmMain.bbtFav12MouseUp(Sender: TObject; Button: TMouseButton;
674:   Shift: TShiftState; X, Y: integer);
675: begin
676:   if (Button = mbLeft) and (gvblnTMV7OnLine) then
677:     SetFavChannel(12)
678:   else if Button = mbRight then
679:     ProcessFavButton(12);
680: end;// procedure TfrmMain.bbtFav12MouseUp
681:
682: //=====
683: //          BUTTON ROUTINES
684: //=====
685: procedure TfrmMain.btnBandClick(Sender: TObject);
686: begin
687:   TogglePTTBand;
688: end;// procedure TfrmMain.btnBandClick
689:
690: //-----
691: procedure TfrmMain.bbtMuteClick(Sender: TObject);
692: begin
693:   ToggleMute;
694: end;// procedure TfrmMain.bbtMuteClick
695:
696: //-----
697: procedure TfrmMain.btnPowerClick(Sender: TObject);
698: begin
699:   TogglePowerOnOff;
700: end;// procedure TfrmMain.btnPowerClick
701:
702: //-----
703: procedure TfrmMain.bbtReverseClick(Sender: TObject);
704: begin
705:   ToggleReverse;
706: end;// procedure TfrmMain.bbtReverseClick
707:
708: //-----
709: procedure TfrmMain.btnRFPowerClick(Sender: TObject);
710: begin
711:   ToggleRFPower;
712: end;// procedure TfrmMain.btnRFPowerClick
713:
714: //=====
715: //          MOUSE UP ROUTINES
716: //=====
717: procedure TfrmMain.uekVHFVolumeMouseUp(Sender: TObject; Button: TMouseButton;
718:   Shift: TShiftState; X, Y: integer);
719:
720: var
```

```

721:   vbytTByte: byte;
722:
723: begin
724:   vbytTByte := Round(uekVHFVolume.Position);
725:   if vbytTByte = 0 then
726:     vbytTByte := 1;
727:   if vbytTByte = 32 then
728:     vbytTByte := 31;
729:   SetVHF_AGValue(IntToHex(vbytTByte, 2));
730: end; // procedure TfrmMain.uekVHFVolumeMouseUp
731:
732: //=====
733: procedure TfrmMain.uekUHFVolumeMouseUp(Sender: TObject; Button: TMouseButton;
734:   Shift: TShiftState; X, Y: integer);
735:
736: var
737:   vbytTByte: byte;
738:
739: begin
740:   vbytTByte := Round(uekUHFVolume.Position);
741:   if vbytTByte = 0 then
742:     vbytTByte := 1;
743:   if vbytTByte = 32 then
744:     vbytTByte := 31;
745:   SetUHF_AGValue(IntToHex(vbytTByte, 2));
746: end; // procedure TfrmMain.uekUHFVolumeMouseUp
747:
748: //=====
749: procedure TfrmMain.uekVHFSquelchMouseUp(Sender: TObject; Button: TMouseButton;
750:   Shift: TShiftState; X, Y: integer);
751:
752: var
753:   vbytTByte: byte;
754:
755: begin
756:   vbytTByte := Round(uekVHFSquelch.Position);
757:   if vbytTByte = 0 then
758:     vbytTByte := 1;
759:   if vbytTByte = 32 then
760:     vbytTByte := 31;
761:   SetVHF_SQValue(IntToHex(vbytTByte, 2));
762: end; // procedure TfrmMain.uekVHFSquelchMouseUp
763:
764: //=====
765: procedure TfrmMain.uekUHFSquelchMouseUp(Sender: TObject; Button: TMouseButton;
766:   Shift: TShiftState; X, Y: integer);
767:
768: var
769:   vbytTByte: byte;
770:
771: begin
772:   vbytTByte := Round(uekUHFSquelch.Position);
773:   if vbytTByte = 0 then
774:     vbytTByte := 1;
775:   if vbytTByte = 32 then
776:     vbytTByte := 31;
777:   SetUHF_SQValue(IntToHex(vbytTByte, 2));
778: end; // procedure TfrmMain.uekUHFSquelchMouseUp
779:
780: //=====

```

TIMER ROUTINES

```

781: //
782: //=====
783: procedure TfrmMain.tmrTimeStatusTimer(Sender: TObject);
784: begin
785:     DisplayTimeStatus;
786: end; // procedure TfrmMain.tmrTimeStatusTimer
787:
788: //=====
789: procedure TfrmMain.tmrSendTimeoutTimer(Sender: TObject);
790: begin
791:
792:     tmrSendTimeout.Enabled := False;
793:     gvblnSendTimeout := True;
794:     MessageDlg('Send Data Timeout', mtError, [mbOK], 0);
795:
796: end; // procedure TfrmMain.tmrSendDataTimer
797:
798: //-----
799: procedure SendTimeoutTimerReset;
800: begin
801:     gvblnSendTimeout := False;
802:     frmMain.tmrSendTimeout.Enabled := True;
803: end; // procedure SendTimeoutTimerReset
804:
805: //=====
806: //          SDPOSERIAL1 ROUTINES
807: //=====
808: procedure TfrmMain.SdpoSerial1RxData(Sender: TObject);
809:
810: const
811:     vstrBuf: string = ''; // Static variable to retain data between calls
812:     cchrEORChar: char = #13; // EOR Character (Cr)
813:
814: var
815:     vblnRcvBufEmpty: boolean;
816:     vstrTStr: string;
817:     vintEORPtr: longint;
818:     vchrTStr: string;
819:     vintTemp: integer;
820:
821: begin
822:
823:     // We have received data so the buffer is no longer empty
824:     vblnRcvBufEmpty := False;
825:
826:     // Append the received data to the buffer
827:     vstrBuf := vstrBuf + sdpoSerial1.ReadData;
828:
829:     // Check for End of Response
830:     vintEORPtr := Pos(cchrEORChar, vstrBuf);
831:
832:     // DO while we have a complete response. vintEORPtr points to // The end of the Response
833:     while vintEORPtr > 0 do
834:     begin
835:         // Parse the complete Response, including the <Cr>.
836:         ParseResponse(Copy(vstrBuf, 1, vintEORPtr));
837:         // Remove the complete Response, including the <Cr>.
838:         vstrBuf := Copy(vstrBuf, vintEORPtr + 1, Length(vstrBuf));
839:         // Check for End of Response. If it is there, we have another response to parse.
840:         vintEORPtr := Pos(cchrEORChar, vstrBuf);

```

```
841:     end;//While vintEORPtr > 1
842:
843:     // We have handled all Responses
844:     vbInRcvBufEmpty := True;
845:
846: end;// procedure TfrmMain.SdpoSerial1RxData
847:
848: //=====
849:
850: end.// unit Main
```