```
unit Main;

{$mode delphi}

//========================================================================================
//
//  Main.pas
//
//  Calls:  AGCommand : SetVHF_AGValue
//                      SetUHF_AGValue
//          AppConstants
//          AppVariables
//          AppTypes
//          BCCommand
//          ColourSchemes : SetColourScheme
//          COMPort
//          DataEntry : TfrmDataEntry.bbtSaveClick
//          Fav : SetFavChannel(n)
//          Final : Finalize
//          INIStuff : ReadINIFile
//                     WriteINIFile
//          Init : Initialize
//          LCDDisplay : LCDOn
//                       LCD Off
//          Mem : frmMem.Setup
//          Mute : ToggleMute
//          PCCommand : ToggleRFPower
//          psCommand : TogglePowerOnOff
//          RadioStatus : RadioStatusOff
//          ReportTMV7File
//          ResponseParser : ParseResponse
//          Reverse : ToggleReverse
//          SMCommand : SMResponseHandler
//          SplashAbout : TdlgSplashAbout.ShowAbout
//                        TdlgSplashAbout.ShowSplash
//          SQCommand : SetVHF_SQValue
//                      SetUHF_SQValue
//          StatusBar : DisplayTimeStatus
//                      DisplayDateStatus
//                      DisplayCOMPortStatus
//                      DisplayTMV7Status
//          TMVFileReport : PrintTMVFileReport
//          TMVFiles : NewTMVFile
//                     OpenTMVFile
//                     SaveTMVFile
//                     SaveTMVFileAs
//
//  Called By: DataEntry_FAV : DataEntry_FAV_Save
//             Final : Finalize
//             Init : Initialize
//             Mute : SetMuteOn
//                    SetMuteOff
//             Register : TdlgRegister.bbtOkClick
//                        TdlgRegister.bbtCancelClick
//             SerialStuff : OpenPort
//
//  Ver: 1.0.0
//
//  Date: 24 Apr 2014
//
```

```
//========================================================================================

interface

uses
  Classes, SysUtils, FileUtil, PrintersDlgs, RackCtls, uEKnob, ueled,
  uERotImage, SdpoSerial, Forms, Controls, Graphics, Dialogs, Menus, NagScreen,
  ComCtrls, Printers,StdCtrls, ExtCtrls, Buttons, Messages, Windows, DOS, LR_Class,
  LR_DSet,
  // Application units
  AGCommand, AppConstants, AppVariables, AppTypes, BCCommand, ColourSchemes, COMPort,
  DataEntry, Fav, Final, Init, Mem, Mute, PCCommand, PSCommand, Register, ResponseParser,
  Reverse, SplashAbout, SQCommand, StatusBar, TMVFileReport, TMVFiles;

type

  { TfrmMain }

  TfrmMain = class(TForm)
    bbtFav01: TBitBtn;
    bbtFav10: TBitBtn;
    bbtFav11: TBitBtn;
    bbtFav12: TBitBtn;
    bbtFav02: TBitBtn;
    bbtFav03: TBitBtn;
    bbtFav04: TBitBtn;
    bbtFav05: TBitBtn;
    bbtFav06: TBitBtn;
    bbtFav07: TBitBtn;
    bbtFav08: TBitBtn;
    bbtFav09: TBitBtn;
    bbtMute: TBitBtn;
    bbtReverse: TBitBtn;
    btnPower: TButton;
    btnBand: TButton;
    btnRFPower: TButton;
    gbxFavourites: TGroupBox;
    ImageList1: TImageList;
    Label1: TLabel;
    Label2: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    lblVHFOnAirBusy: TLabel;
    lblVHFChannelName: TLabel;
    lblUHFChannelName: TLabel;
    lblVHFChannelNr: TLabel;
    lblUHFChannelNr: TLabel;
    lblVHFDataSource: TLabel;
    lblUHFDataSource: TLabel;
    lblVHFDTSSCode: TLabel;
    lblUHFDTSSCode: TLabel;
    lblVHFFreq: TLabel;
    lblVHFDTSS: TLabel;
    lblUHFDTSS: TLabel;
    lblUHFFreq: TLabel;
    lblUHFOnAirBusy: TLabel;
    lblVHFShift: TLabel;
    lblVHFReverse: TLabel;
    lblUHFReverse: TLabel;
    lblUHFShift: TLabel;
    lblVHFTCTFreq: TLabel;
    lblVHFTCT: TLabel;
    lblVHFRFPwr: TLabel;
    lblVHFPTT: TLabel;
    lblUHFPTT: TLabel;
    lblUHFRFPwr: TLabel;
    lblUHFTCT: TLabel;
    lblUHFTCTFreq: TLabel;
    MainMenu1: TMainMenu;
    mnuConfigClosePort: TMenuItem;
    mnuConfigOpenPort: TMenuItem;
    mnuConfigSelectPort: TMenuItem;
    mnuFilePrintTMVFileReport: TMenuItem;
    mnuMemDTMF: TMenuItem;
    mnuMemUHF: TMenuItem;
    mnuMemVHF: TMenuItem;
    mnuConfigColourScheme1: TMenuItem;
    mnuConfigColourScheme10: TMenuItem;
    mnuConfigColourScheme2: TMenuItem;
    mnuConfigColourScheme3: TMenuItem;
    mnuConfigColourScheme4: TMenuItem;
    mnuConfigColourScheme5: TMenuItem;
    mnuConfigColourScheme6: TMenuItem;
    mnuConfigColourScheme7: TMenuItem;
    mnuConfigColourScheme8: TMenuItem;
    mnuConfigColourScheme9: TMenuItem;
    mnuFileSave: TMenuItem;
    mnuFileNew: TMenuItem;
    mnuHelpAbout: TMenuItem;
    mnuHelpRegistration: TMenuItem;
    mnuHelpContents: TMenuItem;
    mnuHelp: TMenuItem;
    mnuMemory: TMenuItem;
    mnuVFOUHF: TMenuItem;
    mnuVFOVHF: TMenuItem;
    mnuVFO: TMenuItem;
    mnuConfigColourSchemes: TMenuItem;
    mnuConfigCOMPort: TMenuItem;
    mnuConfigure: TMenuItem;
    mnuFileSep2: TMenuItem;
    mnuFilePrint: TMenuItem;
    mnuFilePrinterSetup: TMenuItem;
    mnuFileSep1: TMenuItem;
    mnuFileSaveAs: TMenuItem;
    mnuFileOpen: TMenuItem;
    mnuFileExit: TMenuItem;
    mnuFile: TMenuItem;
    OpenDialog1: TOpenDialog;
    PageSetupDialog1: TPageSetupDialog;
    pbrUHFSMeter: TProgressBar;
    pnlLCD: TPanel;
    pnlLCDBorder: TPanel;
    pbrVHFSMeter: TProgressBar;
    PrintDialog1: TPrintDialog;
    PrinterSetupDialog1: TPrinterSetupDialog;
    SaveDialog1: TSaveDialog;
    SdpoSerial1: TSdpoSerial;
    stxOffLine: TStaticText;
    StatusBar1: TStatusBar;
    tmrSendTimeout: TTimer;
    tmrTimeStatus: TTimer;
    ToolBar1: TToolBar;
    tbtFileOpen: TToolButton;
    tbtFilePrint: TToolButton;
    ToolButton1: TToolButton;
    uekVHFVolume: TuEKnob;
    uekVHFSquelch: TuEKnob;
    uekUHFVolume: TuEKnob;
    uekUHFSquelch: TuEKnob;
    updVHF: TUpDown;
    updUHF: TUpDown;
    procedure bbtFav01MouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: integer);
    procedure bbtFav02MouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: integer);
    procedure bbtFav03MouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: integer);
    procedure bbtFav04MouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: integer);
    procedure bbtFav05MouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: integer);
    procedure bbtFav06MouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: integer);
    procedure bbtFav07MouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: integer);
    procedure bbtFav08MouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: integer);
    procedure bbtFav09MouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: integer);
    procedure bbtFav10MouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: integer);
    procedure bbtFav11MouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: integer);
    procedure bbtFav12MouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: integer);
    procedure bbtMuteClick(Sender: TObject);
    procedure bbtReverseClick(Sender: TObject);
    procedure btnBandClick(Sender: TObject);
    procedure btnPowerClick(Sender: TObject);
    procedure btnRFPowerClick(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure FormClose(Sender: TObject; var CloseAction: TCloseAction);
    procedure FormShow(Sender: TObject);
    procedure mnuConfigClosePortClick(Sender: TObject);
    procedure mnuConfigOpenPortClick(Sender: TObject);
    procedure mnuConfigSelectPortClick(Sender: TObject);
    procedure mnuFilePrintTMVFileReportClick(Sender: TObject);
    procedure mnuConfigColourScheme10Click(Sender: TObject);
    procedure mnuConfigColourScheme1Click(Sender: TObject);
    procedure mnuConfigColourScheme2Click(Sender: TObject);
    procedure mnuConfigColourScheme3Click(Sender: TObject);
    procedure mnuConfigColourScheme4Click(Sender: TObject);
    procedure mnuConfigColourScheme5Click(Sender: TObject);
    procedure mnuConfigColourScheme6Click(Sender: TObject);
    procedure mnuConfigColourScheme7Click(Sender: TObject);
    procedure mnuConfigColourScheme8Click(Sender: TObject);
    procedure mnuConfigColourScheme9Click(Sender: TObject);
    procedure mnuFileExitClick(Sender: TObject);
    procedure mnuFileNewClick(Sender: TObject);
    procedure mnuFileOpenClick(Sender: TObject);
    procedure mnuFilePrinterSetupClick(Sender: TObject);
    procedure mnuFileSaveAsClick(Sender: TObject);
    procedure mnuFileSaveClick(Sender: TObject);
    procedure mnuHelpAboutClick(Sender: TObject);
    procedure mnuHelpContentsClick(Sender: TObject);
    procedure mnuHelpRegistrationClick(Sender: TObject);
    procedure mnuMemDTMFClick(Sender: TObject);
    procedure mnuMemUHFClick(Sender: TObject);
    procedure mnuMemVHFClick(Sender: TObject);
    procedure mnuVFOUHFClick(Sender: TObject);
    procedure mnuVFOVHFClick(Sender: TObject);
    procedure SdpoSerial1RxData(Sender: TObject);
    procedure tmrSendTimeoutTimer(Sender: TObject);
    procedure tmrTimeStatusTimer(Sender: TObject);
    procedure uekUHFSquelchMouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: integer);
    procedure uekUHFVolumeMouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: integer);
    procedure uekVHFSquelchMouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: integer);
    procedure uekVHFVolumeMouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: integer);
//    procedure DisplayTitleBar;
  private
    { private declarations }
  public
    { public declarations }
    procedure DisplayTitleBar;
  end;

procedure SendTimeoutTimerReset;

var
  frmMain: TfrmMain;

implementation

{$R *.lfm}

//========================================================================================
//          FORM ROUTINES
//========================================================================================
procedure TfrmMain.FormShow(Sender: TObject);
begin

  dlgSplashAbout.ShowSplash;

  if not Initialize then
  begin
    MessageDlg('Init Failure', mtInformation, [mbOK], 0);
    Finalize;
    Application.Terminate;
  end;// if not Initialize then

  if not gvblnRegistered then
  begin
    if dlgNagScreen.ShowModal = mrOk then
      dlgRegister.ShowModal;
  end;// if not gvblnRegistered

```

```
end;// procedure TfrmMain.FormShow

//----------------------------------------------------------------------------------------
procedure TfrmMain.FormActivate(Sender: TObject);
begin
  // Update the Title Bar
  DisplayTitleBar;
  // Display the Status Bar fields
  DisplayDateStatus;
  DisplayTimeStatus;
  DisplayCOMPortStatus;
  DisplayTMV7Status;
end;// procedure TfrmMain.FormActivate

//----------------------------------------------------------------------------------------
procedure TfrmMain.FormClose(Sender: TObject; var CloseAction: TCloseAction);
begin
  Finalize;
end;// procedure TfrmMain.FormClose

//========================================================================================
procedure TfrmMain.DisplayTitleBar;
begin
    Caption := gcstrAppName + ' - Ver: ' + gcstrAppVersion;
    if gvblnRegistered then
      Caption := Caption + '  Registered to ' +
        gvstrRegisteredName + ' ' + gvstrRegisteredCall
    else
      Caption := Caption + '  Unregistered';
end;// procedure TfrmMain.DisplayTitleBar

//========================================================================================
//          MENU ROUTINES
//========================================================================================

//----------------------------------------------------------------------------------------
//               FILE MENU
//----------------------------------------------------------------------------------------
procedure TfrmMain.mnuFileNewClick(Sender: TObject);
begin

  if MessageDlg('Thie will erase all data. Continue ?', mtConfirmation,
    [mbYes, mbNo], 0) = mrYes then
  begin

    SaveDialog1.Title := 'Create a New TMV File';
    SaveDialog1.InitialDir := gvstrAppPath + gcstrTMV7FileDir;
    SaveDialog1.Filter := 'TMVFiles|*.tmv';
    SaveDialog1.DefaultExt := 'tmv';

    if SaveDialog1.Execute then
    begin
      gvstrTMVFileName := SaveDialog1.Filename;
      NewTMVFile;
    end
    else
      ShowMessage('Cancelled');

  end;// If MessageDlg('Thie will erase all data. Continue ?'

end;// procedure TfrmMain.mnuFileNewClick

//----------------------------------------------------------------------------------------
procedure TfrmMain.mnuFileOpenClick(Sender: TObject);
begin
  OpenTMVFile('');
end;// procedure TfrmMain.mnuFileOpenClick(Sender: TObject);

//----------------------------------------------------------------------------------------
procedure TfrmMain.mnuFileSaveClick(Sender: TObject);
begin
  SaveTMVFile;
end;// procedure TfrmMain.mnuFileSaveClick

//----------------------------------------------------------------------------------------
procedure TfrmMain.mnuFileSaveAsClick(Sender: TObject);
begin
  SaveTMVFileAs;
end;//procedure TfrmMain.mnuFileSaveAsClick

//----------------------------------------------------------------------------------------
procedure TfrmMain.mnuFilePrinterSetupClick(Sender: TObject);
begin

  PrinterSetupDialog1.Execute;
  showmessage(Printer.PrinterName);

end;//procedure TfrmMain.mnuFilePrinterSetupClick

//----------------------------------------------------------------------------------------
procedure TfrmMain.mnuFilePrintTMVFileReportClick(Sender: TObject);
begin

    if not OpenDIalog1.Execute then
      Exit;

    frmTMVFileReport.CreateTmpFiles(OpenDialog1.FileName);

//    frmTMVFileReport.Show;
    frmTMVFileReport.frTMVFileReport.ShowReport;

    frmTMVFileReport.sdfVHFDataset.Active := False;
    frmTMVFileReport.sdfUHFDataset.Active := False;
    frmTMVFileReport.sdfFAVDataset.Active := False;
    frmTMVFileReport.sdfDTMFDataset.Active := False;

end;// procedure TfrmMain.mnuFilePrintTMV7ReportClick

//----------------------------------------------------------------------------------------
procedure TfrmMain.mnuFileExitClick(Sender: TObject);
begin
  Close;
end;//procedure TfrmMain.mnuFileExitClick

//----------------------------------------------------------------------------------------
//          CONFIGURE MENU
//----------------------------------------------------------------------------------------
procedure TfrmMain.mnuConfigSelectPortClick(Sender: TObject);
begin
  frmCOMPort.ShowModal;
```

```
end;// procedure TfrmMain.mnuConfigureSelectPortClick

//----------------------------------------------------------------------------------------
procedure TfrmMain.mnuConfigOpenPortClick(Sender: TObject);
begin

end;// procedure TfrmMain.mnuConfigureOpenPortClick

//----------------------------------------------------------------------------------------
procedure TfrmMain.mnuConfigClosePortClick(Sender: TObject);
begin

end;// procedure TfrmMain.mnuConfigureClosePortClick

//----------------------------------------------------------------------------------------
procedure TfrmMain.mnuConfigColourScheme1Click(Sender: TObject);
begin
  SetColourScheme('1');
end;// procedure TfrmMain.mnuConfigColourScheme1Click

//----------------------------------------------------------------------------------------
procedure TfrmMain.mnuConfigColourScheme2Click(Sender: TObject);
begin
  SetColourScheme('2');
end;// TfrmMain.mnuConfigColourScheme2Click

//----------------------------------------------------------------------------------------
procedure TfrmMain.mnuConfigColourScheme3Click(Sender: TObject);
begin
  SetColourScheme('3');
end;// TfrmMain.mnuConfigColourScheme3Click

//----------------------------------------------------------------------------------------
procedure TfrmMain.mnuConfigColourScheme4Click(Sender: TObject);
begin
  SetColourScheme('4');
end;// TfrmMain.mnuConfigColourScheme4Click

//----------------------------------------------------------------------------------------
procedure TfrmMain.mnuConfigColourScheme5Click(Sender: TObject);
begin
  SetColourScheme('5');
end;

//----------------------------------------------------------------------------------------
procedure TfrmMain.mnuConfigColourScheme6Click(Sender: TObject);
begin
  SetColourScheme('6');
end;// TfrmMain.mnuConfigColourScheme5Click

//----------------------------------------------------------------------------------------
procedure TfrmMain.mnuConfigColourScheme7Click(Sender: TObject);
begin
  SetColourScheme('7');
end;// TfrmMain.mnuConfigColourScheme7Click

//----------------------------------------------------------------------------------------
procedure TfrmMain.mnuConfigColourScheme8Click(Sender: TObject);
begin
  SetColourScheme('8');
```

```
end;// TfrmMain.mnuConfigColourScheme8Click

//------------------------------------------------------------------------------------------
procedure TfrmMain.mnuConfigColourScheme9Click(Sender: TObject);
begin
  SetColourScheme('9');
end;// TfrmMain.mnuConfigColourScheme9Click

//------------------------------------------------------------------------------------------
procedure TfrmMain.mnuConfigColourScheme10Click(Sender: TObject);
begin
  SetColourScheme('10');
end;// TfrmMain.mnuConfigColourScheme10Click

//------------------------------------------------------------------------------------------
//          VFO MENU
//------------------------------------------------------------------------------------------
procedure TfrmMain.mnuVFOVHFClick(Sender: TObject);
begin
  MessageDlg('VHF VFO', mtInformation, [mbOK], 0);
end;// procedure TfrmMain.mnuVFOVHFClick

//------------------------------------------------------------------------------------------
procedure TfrmMain.mnuVFOUHFClick(Sender: TObject);
begin
  MessageDlg('UHF VFO', mtInformation, [mbOK], 0);
end;// procedure TfrmMain.mnuVFOUHFClick

//------------------------------------------------------------------------------------------
//          MEMORY MENU
//------------------------------------------------------------------------------------------'
procedure TfrmMain.mnuMemVHFClick(Sender: TObject);
begin
  frmMem.Setup(drtVHFMEM);
end;// procedure TfrmMain.mnuMemVHFClick

//------------------------------------------------------------------------------------------
procedure TfrmMain.mnuMemUHFClick(Sender: TObject);
begin
  frmMem.Setup(drtUHFMEM);
end;// procedure TfrmMain.mnuMemUHFClick

//------------------------------------------------------------------------------------------
procedure TfrmMain.mnuMemDTMFClick(Sender: TObject);
begin
  frmMem.Setup(drtDTMF);
end;// procedure TfrmMain.mnuMemDTMFClick

//------------------------------------------------------------------------------------------
//          HELP MENU
//------------------------------------------------------------------------------------------
procedure TfrmMain.mnuHelpContentsClick(Sender: TObject);
begin
  Exec('hh.exe', {'-mapid 0 ' +} gcstrAppName + '.chm');
end;// procedure TfrmMain.mnuHelpContentsClick

//------------------------------------------------------------------------------------------
procedure TfrmMain.mnuHelpRegistrationClick(Sender: TObject);
begin
  dlgRegister.ShowModal;
```

```
end;// procedure TfrmMain.mnuHelpRegistrationClick

//----------------------------------------------------------------------------------------
procedure TfrmMain.mnuHelpAboutClick(Sender: TObject);
begin
  dlgSplashAbout.ShowAbout;
end;// procedure TfrmMain.mnuHelpAboutClick

//========================================================================================
//          TOOLBAR ROUTINES
//========================================================================================

//========================================================================================
//          FAVOURITE ROUTINES
//========================================================================================
procedure ProcessFavButton(vbytFavButton: byte);
begin
  frmDataEntry.vdetDataEntryType := detFAV;
  frmDataEntry.vbytChannelNumber := vbytFavButton;
  frmDataEntry.ShowModal;
end;// procedure ProcessFavButton

//----------------------------------------------------------------------------------------
procedure TfrmMain.bbtFav01MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: integer);
begin
  if (Button = mbLeft) and (gvblnTMV7OnLine) then
      SetFavChannel(1)
  else if Button = mbRight then
    ProcessFavButton(1);
end;// procedure TfrmMain.bbtFav01MouseUp

//----------------------------------------------------------------------------------------
procedure TfrmMain.bbtFav02MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: integer);
begin
  if (Button = mbLeft) and (gvblnTMV7OnLine) then
    SetFavChannel(2)
  else if Button = mbRight then
    ProcessFavButton(2);
end;// procedure TfrmMain.bbtFav02MouseUp

//----------------------------------------------------------------------------------------
procedure TfrmMain.bbtFav03MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: integer);
begin
  if (Button = mbLeft) and (gvblnTMV7OnLine) then
    SetFavChannel(3)
  else if Button = mbRight then
    ProcessFavButton(3);
end;// procedure TfrmMain.bbtFav03MouseUp

//----------------------------------------------------------------------------------------
procedure TfrmMain.bbtFav04MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: integer);
begin
  if (Button = mbLeft) and (gvblnTMV7OnLine) then
    SetFavChannel(4)
  else if Button = mbRight then
    ProcessFavButton(4);
```

```
end;// procedure TfrmMain.bbtFav04MouseUp

//----------------------------------------------------------------------------------------
procedure TfrmMain.bbtFav05MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: integer);
begin
  if (Button = mbLeft) and (gvblnTMV7OnLine) then
    SetFavChannel(5)
  else if Button = mbRight then
    ProcessFavButton(5);
end;// procedure TfrmMain.bbtFav05MouseUp

//----------------------------------------------------------------------------------------
procedure TfrmMain.bbtFav06MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: integer);
begin
  if (Button = mbLeft) and (gvblnTMV7OnLine) then
    SetFavChannel(6)
  else if Button = mbRight then
    ProcessFavButton(6);
end;// procedure TfrmMain.bbtFav06MouseUp

//----------------------------------------------------------------------------------------
procedure TfrmMain.bbtFav07MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: integer);
begin
  if (Button = mbLeft) and (gvblnTMV7OnLine) then
    SetFavChannel(7)
  else if Button = mbRight then
    ProcessFavButton(7);
end;// procedure TfrmMain.bbtFav07MouseUp

//----------------------------------------------------------------------------------------
procedure TfrmMain.bbtFav08MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: integer);
begin
  if (Button = mbLeft) and (gvblnTMV7OnLine) then
    SetFavChannel(8)
  else if Button = mbRight then
    ProcessFavButton(8);
end;// procedure TfrmMain.bbtFav08MouseUp

//----------------------------------------------------------------------------------------
procedure TfrmMain.bbtFav09MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: integer);
begin
  if (Button = mbLeft) and (gvblnTMV7OnLine) then
    SetFavChannel(9)
  else if Button = mbRight then
    ProcessFavButton(9);
end;// procedure TfrmMain.bbtFav09MouseUp

//----------------------------------------------------------------------------------------
procedure TfrmMain.bbtFav10MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: integer);
begin
  if (Button = mbLeft) and (gvblnTMV7OnLine) then
    SetFavChannel(10)
  else if Button = mbRight then
    ProcessFavButton(10);
```

```
end;// procedure TfrmMain.bbtFav10MouseUp

//------------------------------------------------------------------------------------------
procedure TfrmMain.bbtFav11MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: integer);
begin
  if (Button = mbLeft) and (gvblnTMV7OnLine) then
    SetFavChannel(11)
  else if Button = mbRight then
    ProcessFavButton(11);
end;// procedure TfrmMain.bbtFav11MouseUp

procedure TfrmMain.bbtFav12MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: integer);
begin
  if (Button = mbLeft) and (gvblnTMV7OnLine) then
    SetFavChannel(12)
  else if Button = mbRight then
    ProcessFavButton(12);
end;// procedure TfrmMain.bbtFav12MouseUp

//==========================================================================================
//          BUTTON ROUTINES
//==========================================================================================
procedure TfrmMain.btnBandClick(Sender: TObject);
begin
  TogglePTTBand;
end;// procedure TfrmMain.btnBandClick

//------------------------------------------------------------------------------------------
procedure TfrmMain.bbtMuteClick(Sender: TObject);
begin
  ToggleMute;
end;// procedure TfrmMain.bbtMuteClick

//------------------------------------------------------------------------------------------
procedure TfrmMain.btnPowerClick(Sender: TObject);
begin
  TogglePowerOnOff;
end;// procedure TfrmMain.btnPowerClick

//------------------------------------------------------------------------------------------
procedure TfrmMain.bbtReverseClick(Sender: TObject);
begin
  ToggleReverse;
end;// procedure TfrmMain.bbtReverseClick

//------------------------------------------------------------------------------------------
procedure TfrmMain.btnRFPowerClick(Sender: TObject);
begin
  ToggleRFPower;
end;// procedure TfrmMain.btnRFPowerClick

//==========================================================================================
//          MOUSE UP ROUTINES
//==========================================================================================
procedure TfrmMain.uekVHFVolumeMouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: integer);

var
```

```
  vbytTByte: byte;

begin
  vbytTByte := Round(uekVHFVolume.Position);
  if vbytTByte = 0 then
    vbytTBYte := 1;
  if vbytTByte = 32 then
    vbytTBYte := 31;
  SetVHF_AGValue(IntToHex(vbytTByte, 2));
end;// procedure TfrmMain.uekVHFVolumeMouseUp

//========================================================================================
procedure TfrmMain.uekUHFVolumeMouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: integer);

var
  vbytTByte: byte;

begin
  vbytTByte := Round(uekUHFVolume.Position);
  if vbytTByte = 0 then
    vbytTBYte := 1;
  if vbytTByte = 32 then
    vbytTBYte := 31;
  SetUHF_AGValue(IntToHex(vbytTByte, 2));
end;// procedure TfrmMain.uekUHFVolumeMouseUp

//========================================================================================
procedure TfrmMain.uekVHFSquelchMouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: integer);

var
  vbytTByte: byte;

begin
  vbytTByte := Round(uekVHFSquelch.Position);
  if vbytTByte = 0 then
    vbytTBYte := 1;
  if vbytTByte = 32 then
    vbytTBYte := 31;
  SetVHF_SQValue(IntToHex(vbytTByte, 2));
end;// procedure TfrmMain.uekVHFSquelchMouseUp

//========================================================================================
procedure TfrmMain.uekUHFSquelchMouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: integer);

var
  vbytTByte: byte;

begin
  vbytTByte := Round(uekUHFSquelch.Position);
  if vbytTByte = 0 then
    vbytTBYte := 1;
  if vbytTByte = 32 then
    vbytTBYte := 31;
  SetUHF_SQValue(IntToHex(vbytTByte, 2));
end;// procedure TfrmMain.uekUHFSquelchMouseUp

//========================================================================================
//          TIMER ROUTINES
//========================================================================================
procedure TfrmMain.tmrTimeStatusTimer(Sender: TObject);
begin
  DisplayTimeStatus;
end;// procedure TfrmMain.tmrTimeStatusTimer

//========================================================================================
procedure TfrmMain.tmrSendTimeoutTimer(Sender: TObject);
begin

  tmrSendTimeout.Enabled := False;
  gvblnSendTimeout := True;
  MessageDlg('Send Data Timeout', mtError, [mbOK], 0);

end;// procedure TfrmMain.tmrSendDataTimer

//----------------------------------------------------------------------------------------
procedure SendTimeoutTimerReset;
begin
  gvblnSendTimeout := False;
  frmMain.tmrSendTimeout.Enabled := True;
end;// procedure SendTimeoutTimerReset

//========================================================================================
//          SDPOSERIAL1 ROUTINES
//========================================================================================
procedure TfrmMain.SdpoSerial1RxData(Sender: TObject);

const
  vstrBuf: string = ''; // Static variable to retain data between calls
  cchrEORChar: char = #13; // EOR Character (Cr)

var
  vblnRcvBufEmpty: boolean;
  vstrTStr: string;
  vintEORPtr: longint;
  vchrTStr: string;
  vintTemp: integer;

begin

  // We have received data so the buffer is no longer empty
  vblnRcvBufEmpty := False;

  //    Append the recevied data to the buffer
  vstrBuf := vstrBuf + sdpoSerial1.ReadData;

  // Check for End of Response
  vintEORPtr := Pos(cchrEORChar, vstrBuf);

  // DO while we have a complete response. vintEORPtr points to // The end of the Response
  while vintEORPtr > 0 do
  begin
    // Parse the complete Response, includiing the <Cr>.
    ParseResponse(Copy(vstrBuf, 1, vintEORPtr));
    // Remove the complete Response, includiing the <Cr>.
    vstrBuf := Copy(vstrBuf, vintEORPtr + 1, Length(vstrBuf));
    // Check for End of Response. If it is there, we have another response to parse.
    vintEORPtr := Pos(cchrEORChar, vstrBuf);
  end;//While vintEORPtr > 1

  // We have handled all Responses
  vblnRcvBufEmpty := True;

end;// procedure TfrmMain.SdpoSerial1RxData

//========================================================================================

end.// unit Main
```