```
unit Reverse;

//
=================================================================
=============
//
//  REVERSE.PAS
//
//  This module toggles the selected band between "Reverse" and
"Normal"
//
//  Called By: MAIN : TfrmMain.sbtREVClick
//             FINAL : Finalize
//             FAVOURITES : SetFavouriteChannel
//             DATAENTRY : TfrmDataEntry.bbtOkClick
//             MEM : TfrmMEM.dbgVHFDblClick
//                   TfrmMEM.dbgVHFDblClick
//
//  Calls: MYVARIABLES
//         STATUSSTUFF : VHFFreqStatus
//         MAIN
//         BUF_COMMAND : Set_Buffer;
//         BY_COMMAND : Read_BY_VHF;
//                      Read_BY_UHF;
//         SM_CMMMAND : Read_SM_VHF;
//                      Read_SM_UHF;
//         DEBUG : WriteDebugRecord
//
//  Version History:
//
//    1.0.5b1 01 Jan 2010 Initial Release
//
//
=================================================================
=============

interface

uses
  Dialogs, Graphics,
  MyVariables, StatusStuff, BUF_Command, BY_Command, SM_Command,
Debug;

procedure Toggle_VHF_Reverse;
procedure Toggle_UHF_Reverse;

implementation

uses
  Main;
var
  sVHF_Orig_RX_Freq : string[7]; //nnn.nnn
```

```
  sVHF_Orig_TX_Freq : string[7]; //nnn.nnn
  sVHF_Orig_Shift : string[1];   //[s,+,-]
  sVHF_Orig_Tone : string[1];    //[0,1]
  sVHF_Orig_CTCSS : string[1];   //[0,1]

  sUHF_Orig_RX_Freq : string[7]; //nnn.nnn
  sUHF_Orig_TX_Freq : string[7]; //nnn.nnn
  sUHF_Orig_Shift : string[1];   //[s,+,-]
  sUHF_Orig_Tone : string[1];    //[0,1]
  sUHF_Orig_CTCSS : string[1];   //[0,1]

const
  RevHdr = 'REV';

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

procedure Toggle_VHF_Reverse;
begin

    WriteDebugRecord ( RevHdr + '001',
                       'Toggle_VHF_Reversee',
                       'Entered',
                       '' );

  if sVHF_Reverse = '0' then
  begin

        // First set the toggle flag
    sVHF_Reverse := '1';  // This is used as a flag only. It is
not sent to the transceiver
        // Now swap the TX and RX Frequencies to enable reverse
    sVHF_Orig_RX_Freq := sVHF_RX_Freq;
    sVHF_Orig_TX_Freq := sVHF_TX_Freq;
    sVHF_RX_Freq := sVHF_Orig_TX_Freq;  // TX Now is the
opriginal RX
    sVHF_TX_Freq := sVHF_Orig_RX_Freq;  // RX Now is the original
TX
      // Now swap the shift
    sVHF_Orig_Shift := sVHF_Shift;
    if sVHF_Shift = '+' then
      sVHF_Shift := '-'
    else
      sVHF_Shift := '+';
      // Now we turn off the Tone and CTCSS
    sVHF_Orig_Tone := sVHF_Tone;
    sVHF_Tone := '0';
    sVHF_Orig_CTCSS := sVHF_CTCSS;
    sVHF_CTCSS := '0';

    Set_Buffer;
```

```
    frmMain.sbtREV.Font.Color := clRed;

  end
  else
  begin

        // First set the toggle flag
    sVHF_Reverse := '0';  // This is used as a flag only. It is
not sent to the transceiver
        // Now swap the TX and RX Frequencies to re-enable normal
    sVHF_RX_Freq := sVHF_Orig_RX_Freq;  // TX Now is the
opriginal TX
    sVHF_TX_Freq := sVHF_Orig_TX_Freq;  // RX Now is the original
RX
        // Get the shift back
    sVHF_Shift := sVHF_Orig_Shift;
      // And the Tone and CTCSS
    sVHF_Tone := sVHF_Orig_Tone;
    sVHF_CTCSS := sVHF_Orig_CTCSS;

    Set_Buffer;
    frmMain.sbtREV.Font.Color := clBlue;

  end;//if sVHF_Reverse = '0'

  VHFFreqStatus;
  Read_BY_VHF;
  Read_SM_VHF;

end;//procedure Toggle_VHF_Reverse;

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


procedure Toggle_UHF_Reverse;
begin

  WriteDebugRecord ( RevHdr + '002',
                     'Toggle_UHF_Reversee',
                     'Entered',
                     '' );

  if sUHF_Reverse = '0' then
  begin

        // First set the toggle flag
    sUHF_Reverse := '1';  // This is used as a flag only. It is
not sent to the transceiver
        // Now swap the TX and RX Frequencies to enable reverse
    sUHF_Orig_RX_Freq := sUHF_RX_Freq;
```

```
    sUHF_Orig_TX_Freq := sUHF_TX_Freq;
    sUHF_RX_Freq := sUHF_Orig_TX_Freq;  // TX Now is the
opriginal RX
    sUHF_TX_Freq := sUHF_Orig_RX_Freq;  // RX Now is the original
TX
      // Now swap the shift
    sUHF_Orig_Shift := sUHF_Shift;
    if sUHF_Shift = '+' then
      sUHF_Shift := '-'
    else
      sUHF_Shift := '+';
      // Now we turn off the Tone and CTCSS
    sUHF_Orig_Tone := sUHF_Tone;
    sUHF_Tone := '0';
    sUHF_Orig_CTCSS := sUHF_CTCSS;
    sUHF_CTCSS := '0';

    Set_Buffer;
    frmMain.sbtREV.Font.Color := clRed;

  end
  else
  begin

        // First set the toggle flag
    sUHF_Reverse := '0';  // This is used as a flag only. It is
not sent to the transceiver
        // Now swap the TX and RX Frequencies to re-enable normal
    sUHF_RX_Freq := sUHF_Orig_RX_Freq;  // TX Now is the
opriginal TX
    sUHF_TX_Freq := sUHF_Orig_TX_Freq;  // RX Now is the original
RX
        // Get the shift back
    sUHF_Shift := sUHF_Orig_Shift;
      // And the Tone and CTCSS
    sUHF_Tone := sUHF_Orig_Tone;
    sUHF_CTCSS := sUHF_Orig_CTCSS;

    Set_Buffer;
    frmMain.sbtREV.Font.Color := clBlue;

  end;//if sUHF_Reverse = '0'

  UHFFreqStatus;
  Read_BY_UHF;
  Read_SM_UHF;

end;//procedure Toggle_UHF_Reverse;

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

```
end.//unit Reverse;
```