

```
unit COM_1;

interface

uses
  Classes, Windows, BuffSock;

type
  TCOM1 = class(TThread)
  private
    { Private declarations }
  protected
    procedure Execute; override;
    procedure UpdateForms;
    procedure RxToggle;
  end;

var
  linje : string;

implementation
Uses
  Main;

procedure TCOM1.Execute;
Var
  fSuccess : Boolean;
  NBytes : DWord;
  RawData :
    RECORD
      CASE Boolean OF
        True : (Ch : Char);
        False : (n : byte);
      End;
  SendData : Array[1..255] of Byte;
  T : Integer;
  Trigger : Integer;
  ByteCounter : Byte;

Begin
  FreeOnTerminate := True;
  linje := '';
  Trigger := 0;
  While Not Terminated Do Begin
    // Les en byte fra COM1 og legg den inn i RawData
    ReadFile(hCOM1,RawData.n,1,NBytes,Nil);

    Inc(ByteCounter);
    if ByteCounter > 239 then begin
      ByteCounter := 0; //vis på formen hver gang det er mottatt
      Synchronize(RxToggle); // 240 nye bytes (hvert 500ms)
    end;

    CASE Trigger of
      5 : begin //Hele headeren er mottatt og her kommer
              // selve reply-tegnet (S, C eller U)
              linje := 'UiB: ' + RawData.Ch; //Mottatt streng legges inn i "linje"
              Synchronize(UpdateForms); //Start prosedyren som reagerer på reply
```

```
        trigger := 0;                //Header-triggeren resettes
    end;
4 : begin                            //Sjekker her om tegn nr. 5 matcher Headeren
    if RawData.Ch = ' ' then Trigger := 5 else Trigger := 0;
    end;
3 : begin                            //Sjekker her om tegn nr. 4 matcher Headeren
    if RawData.Ch = ':' then Trigger := 4 else Trigger := 0;
    end;
2 : begin                            //Sjekker her om tegn nr. 3 matcher Headeren
    if RawData.Ch = 'B' then Trigger := 3 else Trigger := 0;
    end;
1 : begin                            //Sjekker her om tegn nr. 2 matcher Headeren
    if RawData.Ch = 'i' then Trigger := 2 else Trigger := 0;
    end;
0 : begin                            //Sjekker her om tegnet matcher Headeren
    if RawData.Ch = 'U' then Trigger := 1;
    end;
END; {..of CASE}
End; {..of tråd-loop}
End; {..of TCOM1.Execute tråd-prosedyren}
```

```
procedure TCOM1.RxToggle;
begin
    //Når programmet mottar vanlig radiostøy på 4800 baud vil denne
    // prosedyren kjøres 2 ggr. i sekundet, dvs CheckBox1 blinker med 1 Hz.
    FrmMain.CheckBox1.Checked := not FrmMain.CheckBox1.Checked;
end;
```

```
procedure TCOM1.UpdateForms;
var
    C, I : ShortInt;
begin
    FrmMain.DataBox.Text := linje;                //Viser mottatt reply fra PIC/nRF
    FrmMain.CheckBox2.Checked := not FrmMain.CheckBox2.Checked;
                                                //Toggle CheckBox2 for å vise at
                                                // en header er mottatt.
    If (linje[6] = 'S') OR (linje[6] = 'C') OR (linje[6] = 'U') OR
        (linje[6] = 's') OR (linje[6] = 'c') OR (linje[6] = 'u') then begin
                                                //Var tegnet som fulgte Headeren
                                                // et gyldig reply-tegn (S,C,U) ?
        FrmMain.Timer1.Enabled := False;        //Stopp timeren (stopp radio TX)
        FrmMain.CheckBox4.Checked := False;     //Vis at vi ikke venter på svar
        FrmMain.SpeedButton1.Enabled := False; //Vi fikk svar, knappen disables
        C := FrmMain.BasicServerSocket1.Clientcount;
        for I := 0 to C-1 do                    //Send reply-tegnet til klientene
            (FrmMain.BasicServerSocket1.Client[I] as TLineSocket).WriteLine
                ( UpCase( linje[6] ) )
        end;
    end;
end;

end.
```

```
unit Main;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, COM_1, ExtCtrls, dWinsock, BuffSock, Buttons;

type
  TFrmMain = class(TForm)
    DataBox: TEdit;
    Timer1: TTimer;
    BasicServerSocket1: TBasicServerSocket;
    Label2: TLabel;
    Button10: TButton;
    ClientEdit: TEdit;
    Shape1: TShape;
    Panel1: TPanel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label1: TLabel;
    RadioButton3: TRadioButton;
    Button1: TButton;
    Button2: TButton;
    RadioButton4: TRadioButton;
    CheckBox1: TCheckBox;
    Panel2: TPanel;
    CheckBox2: TCheckBox;
    CheckBox3: TCheckBox;
    CheckBox4: TCheckBox;
    SpeedButton1: TSpeedButton;

    procedure Timer1Timer(Sender: TObject);
    procedure Button10Click(Sender: TObject);
    procedure BasicServerSocket1Info(Sender: TObject; icode: TSockInfo;
      Socket: TSocketBase);
    procedure BasicServerSocket1Accept(Sender: TObject;
      Socket: TSocketBase);
    procedure BasicServerSocket1Disconnect(Sender: TObject;
      Socket: TSocketBase);
    procedure FormCreate(Sender: TObject);
    procedure FormMouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure RadioButton4Click(Sender: TObject);
    procedure RadioButton3Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
    procedure LineReceived(Sender : TLineSocketBase;
      const line : string; complete : Boolean);
  end;
```

```
var
  FrmMain      : TFrmMain;
  hCOM1        : THandle;
  COM1         : TCOM1;
  TransmitLine : string;

implementation

{$R *.DFM}

procedure TFrmMain.Timer1Timer(Sender: TObject); //Hver gang denne timer'en går
var
  SendData : Array[1..255] of Byte; // sendes det ut en seiell streng
  T        : integer;              // på 9 bytes 4800bd, 8N2.
  NBytes   : DWord;
begin
  If RadioButton3.Checked then begin //Dersom aut. TX/RX switching...
    EscapeCommFunction(hCOM1, CLRRTS); //Legg RTS negativ ==> 5V ==> TX
    CheckBox3.Checked := True;        //Visuell indikasjon på TX
  end;
  Sleep(4);                          //Delay så TX blir klar for data..
  for T := 1 to Length(TransmitLine) do //Putt tegn for tegn inn i et
    SendData[T] := Ord(TransmitLine[T]); // byte-array..
  WriteFile(hCOM1, SendData, T-1, NBytes, Nil); //..send ut dette arrayet på COM1

  Sleep(24);                          //Delay slik at COM-port'en får
  // tid til å sende ut alle bits.

  If RadioButton3.Checked then begin //Dersom aut. TX/RX switching...
    EscapeCommFunction(hCOM1, SETRTS); //Legger RTS positiv ==> 0V ==> RX
    CheckBox3.Checked := False;       //Visuell indikasjon på RX
  end;
end;

procedure TFrmMain.Button10Click(Sender: TObject);
begin
  Close; //Avslutter hele programmet!
end;

procedure TFrmMain.BasicServerSocket1Info(Sender: TObject;
  icode: TSockInfo; Socket: TSocketBase);
begin
  Panell1.Caption := SockInfotoString(icode); //Skriver ut "SocketInfo"
  // nederst på program-vinduet.
end;

procedure TFrmMain.BasicServerSocket1Accept(Sender: TObject;
  Socket: TSocketBase);
begin
  (Socket as TLineSocket).OnLineReceived := LineReceived; //Viser til prosedyre
  // som håndterer mottak av data.
  Label4.Caption := IntToStr(BasicServerSocket1.ClientCount);
  //Viser antall clients på form'en
end;

procedure TFrmMain.LineReceived(
  Sender : TLineSocketBase; const line : string; complete : Boolean);
begin
```

```
TransmitLine := 'abcUiB: U'; //Setter inn default radio-TX data

if (line = 'X') or (line = 'x') then begin //Dersom "STATUS request" mottatt:
  ClientEdit.Text := 'Status request (X)'; //Vis kommando på form'en.
  TransmitLine := 'abcUiB: X'; //Klargjør radio-TX data.
end;
if (line = 'S') or (line = 's') then begin //Dersom "SET" kommando mottatt:
  ClientEdit.Text := 'Skru PÅ LED (S)'; //Vis kommando på form'en:
  TransmitLine := 'abcUiB: S'; //Klargjør radio-TX data.
end;
if (line = 'C') or (line = 'c') then begin //Dersom "CLEAR" kommando mottatt:
  ClientEdit.Text := 'Skru AV LED (C)'; //Vis kommando på form'en.
  TransmitLine := 'abcUiB: C'; //Klargjør radio-TX data.
end;

Timer1.Enabled := True; //Starter timer'en som sender ut
// seriell streng og evt. styrer
// TX/RX linjen (RTS på COM1)

CheckBox4.Checked := True; //Viser på formen at vi venter på
// svar fra PIC'en.

SpeedButton1.Enabled := True; //Enabler knappen som kan
// kansellere kommando-utsendelse

end;

procedure TFrmMain.BasicServerSocket1Disconnect(Sender: TObject;
  Socket: TSocketBase);
begin
  Label4.Caption := IntToStr( BasicServerSocket1.ClientCount -1 );
//Skriver ut på formen hvor mange
// klienter som er tilkoblet
// etter denne frakoblingen.
end;

procedure TFrmMain.FormCreate(Sender: TObject);
Var
  fSuccess : Boolean;
  DCB : TDCB;
  CT : TCommTimeouts;
Begin
  // Klargjør for Socket-tilkoblinger
  BasicServerSocket1.Listen( TLineSocket);

  // Åpne COM1
  hCOM1 := CreateFile(PChar('COM1'), GENERIC_READ OR GENERIC_WRITE, 0, Nil,
    OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0);
  If hCOM1 = INVALID_HANDLE_VALUE Then Begin
    MessageDlg('Fatal feil:'+#10+'Klarte ikke å åpne COM1',mtWarning,[MBok], 0);
    Exit
  End;
  // Les de gamle settingene
  fSuccess := GetCommState(hCOM1, DCB);
  If fSuccess = False Then Begin
    MessageDlg('Fatal feil:'+#10+'Klarte ikke å lese COM1 sine DCB settinger',
      mtWarning,[MBok], 0);
    Exit
  End;
  // Endre COM1 settingene
  DCB.DCBLength := SizeOf(TDCB);
```

```
DCB.BaudRate := CBR_4800;
DCB.Parity := NOPARITY;
DCB.StopBits := TWOSTOPBITS;
DCB.ByteSize := 8;
// Iverksett endringene på COM1
fSuccess := SetCommState(hCOM1, DCB);
If fSuccess = False Then Begin
  MessageDlg('Fatal feil:'+#10+'Klarte ikke å sette COM1 sine "DCB settings"',
            mtWarning,[MBok], 0);

  Exit
End;
// Setter CommMask
fSuccess := SetCommMask(hCom1,EV_CTS);
If fSuccess = False Then Begin
  MessageDlg('Fatal feil:'+#10+'Klarte ikke å sette COM1 sin "CommMask" !',
            mtWarning,[MBok], 0);

  Exit
End;
// Timeout settinger...
CT.ReadIntervalTimeout := 0;
CT.ReadTotalTimeoutMultiplier := 0;
CT.ReadTotalTimeoutConstant := 0;
CT.WriteTotalTimeoutMultiplier := 0;
CT.WriteTotalTimeoutConstant := 0;
fSuccess := SetCommTimeouts(hCOM1, CT);
If fSuccess = False Then Begin
  MessageDlg('Fatal feil:'+#10+'Klarte ikke å sette nye timeout-instillinger '
            +'på COM1 !', mtWarning, [MBok], 0);

  Exit;
End;

SetupComm(hCOM1,250,250); //Sett opp buffers

COM1 := TCOM1.Create(False); //Lager tråden for seriell kommunikasjon
EscapeCommFunction(hCOM1,SETRTS); //Legger RTS positiv ==> 0V ==> RX
end;

procedure TFrmMain.FormMouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin //Info-DialogBox med program-info
  //Kommer frem ved å klikke på form'en
  //samtidig som <Ctrl> tasten holdes nede.
  if (ssCtrl in Shift) then MessageDlg(
    'Fagprøve: Service-elektronikk' +#10 +#10
    +'Erik Grindheim, feb. 2000' +#10
    +' _____' +#10
    +' ' +#10
    +'Inst. for den Faste Jords Fysikk,' +#10
    +'Universitetet i Bergen' +#10
    +' _____' +#10,
    mtInformation, [mbOK], 0);
end;

procedure TFrmMain.RadioButton4Click(Sender: TObject);
begin //Når en enabler manuell TX/RX switching
  Button1.Enabled := True; // må en også enable knappene for å
  Button2.Enabled := True; // skifte mellom TX og RX. (Buttons 1&2)
end;
```

