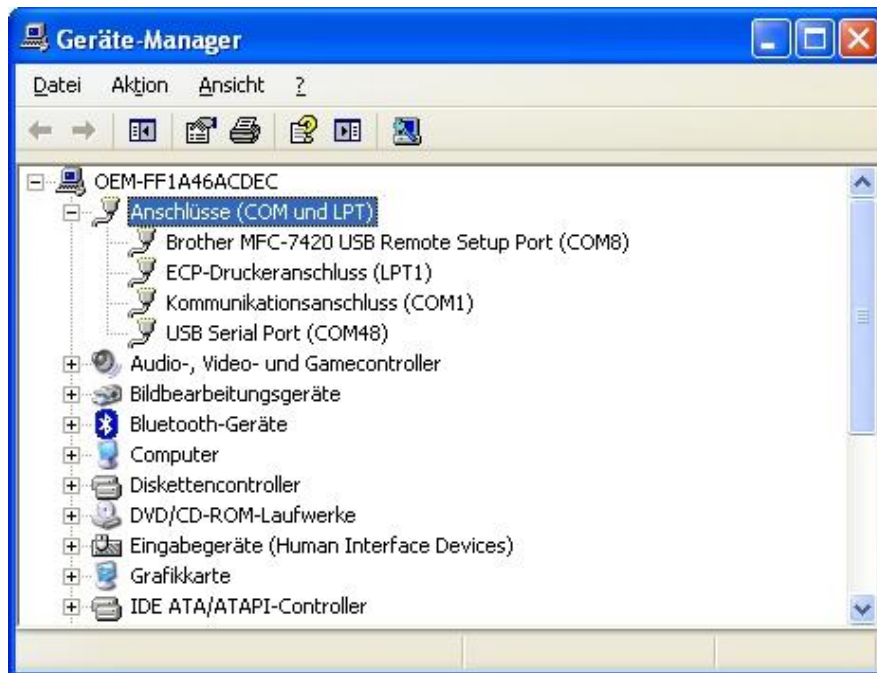


Virtuelle COM-Schnittstelle umbenennen

COM-Nummer eines USB/Seriell-Wandlers verändern

Wenn

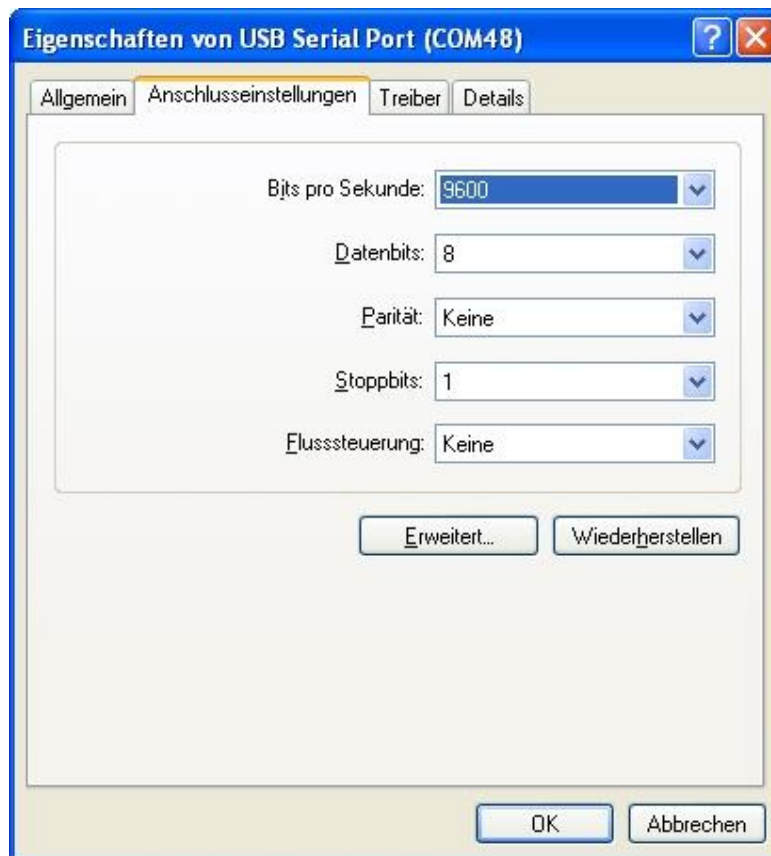
man ein Gerät mit einem USB/Seriell-Wandler neu anschließt, wird meist eine neue virtuelle COM-Schnittstelle gebildet. Hier habe ich das Elektor-DSP-Radio mit einem FT232R angeschlossen. Weil mein PC schon vorher so viele FTDIs gesehen hat, wird die neue Schnittstelle COM48 genannt. Um das herauszufinden und eventuell zu ändern, muss der Geräte-Manager gestartet werden.



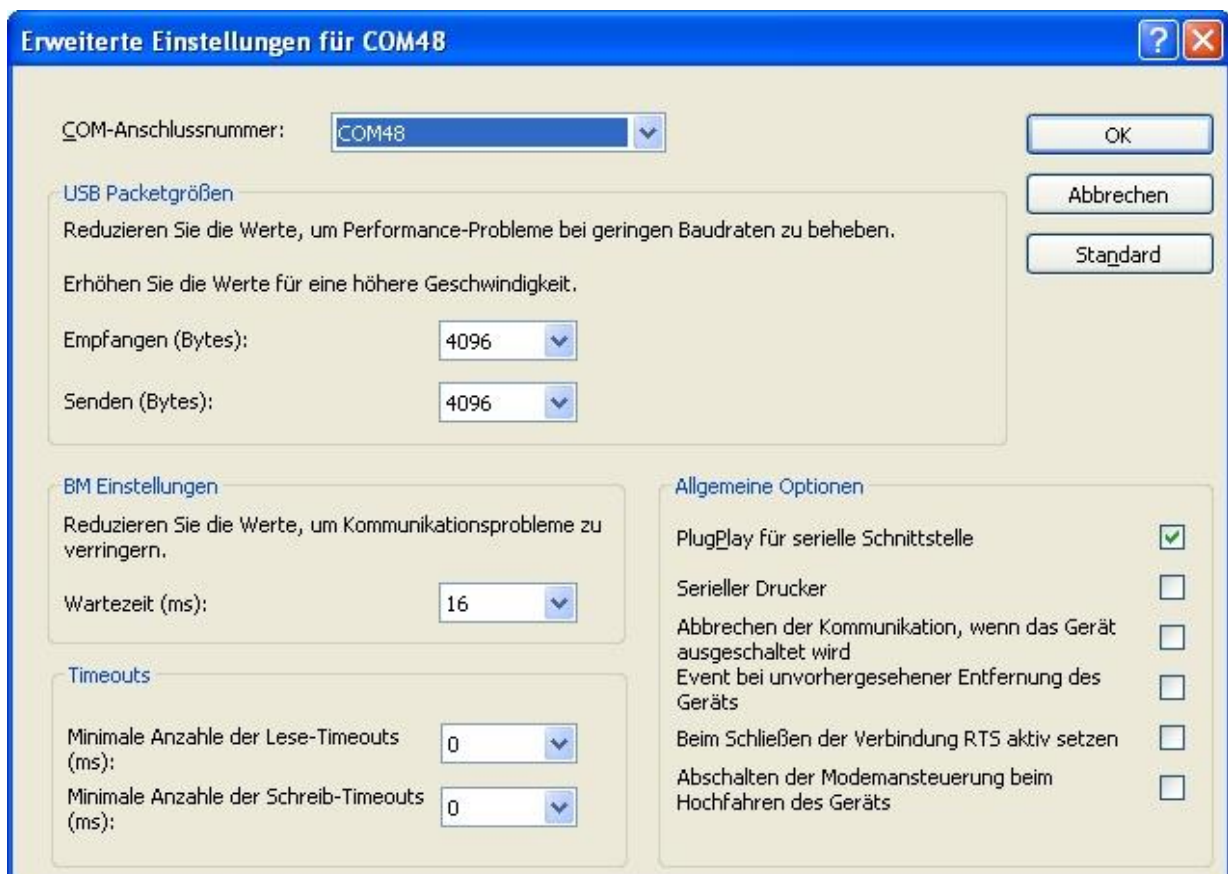
Auf COM48 geklickt...



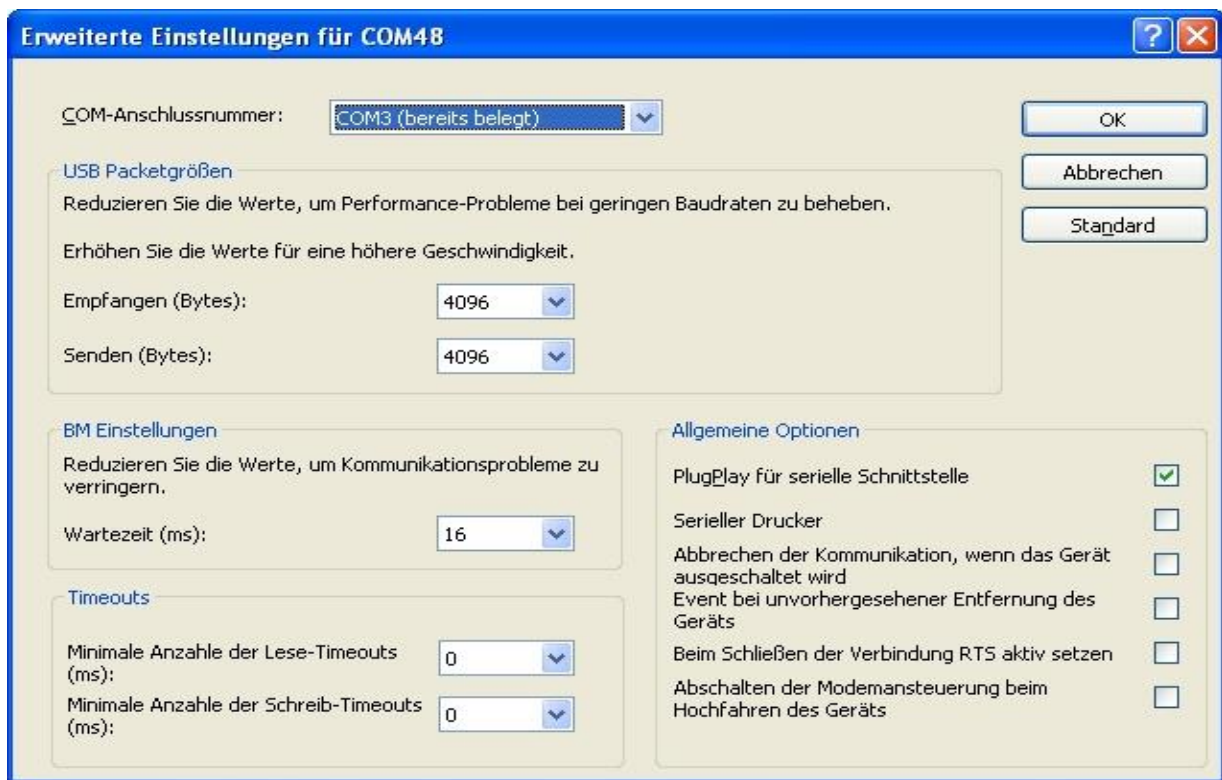
und dann auf die Anschlüsseigenschaften ...



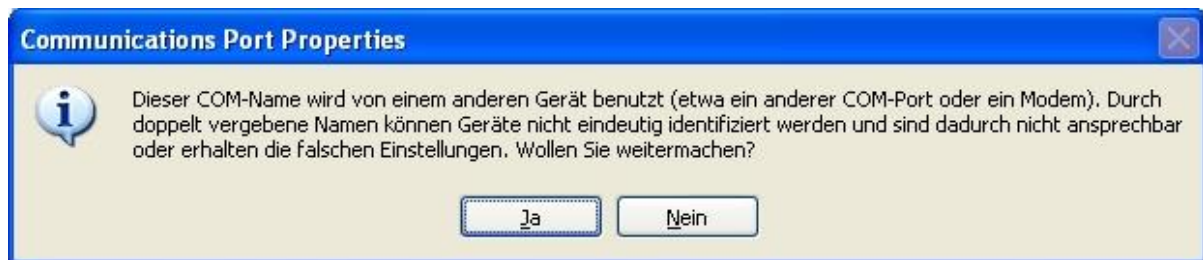
... zeigt zunächst nur unnötige Voreinstellungen, denn Die Baudrate usw. werden von der Anwendersoftware neu festgelegt. Aber unter "Erweitert ..." wird es interessant.



Hier gibt es eine Auswahl für die COM-Nummer. Ich hätte gern COM3.



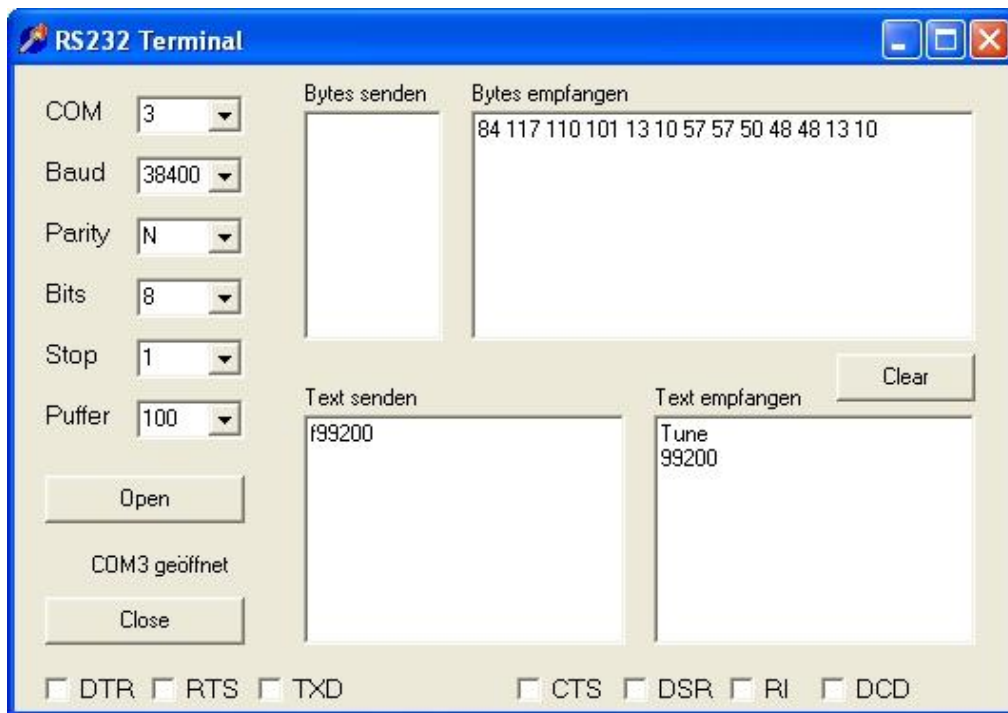
COM3 ist bereits belegt, sagt Windows. Allerdings nicht immer, sondern nur wenn gerade ein entsprechendes Gerät am USB hängt. Ich bestehe also auf COM3. Noch eine Warnung, die ich mit Ja bestätige.



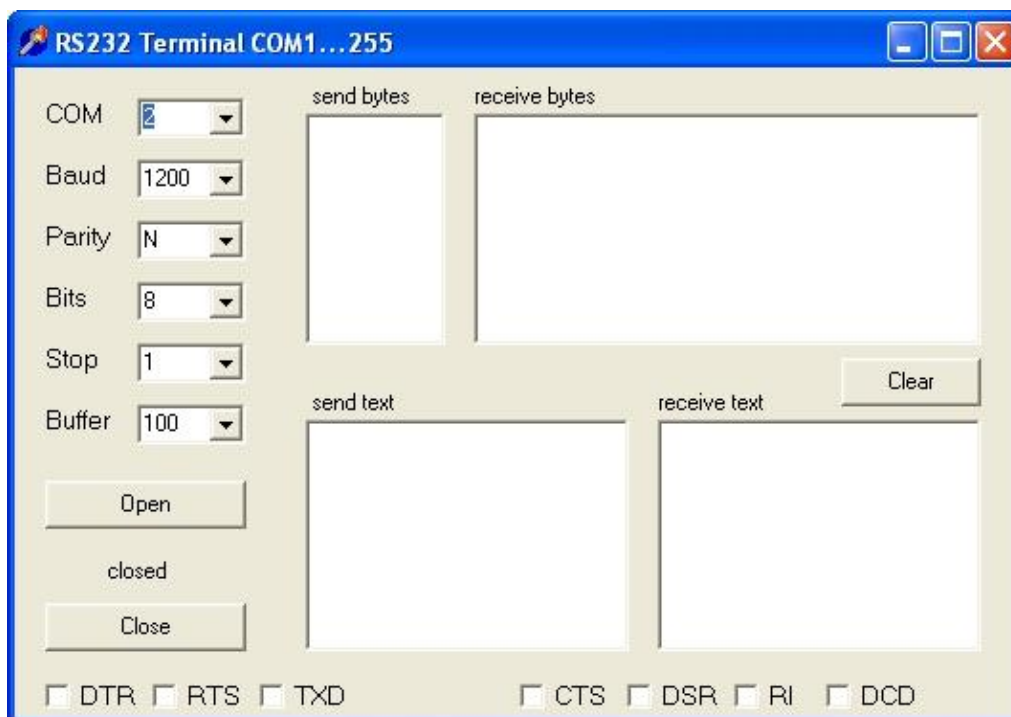
Jetzt ist die COM 3 angenommen. In der Übersicht erscheint der neue Name erst, nachdem ich den Geräte-Manager einmal neu gestartet habe.



Ein Test mit dem Terminalprogramm: Es funktioniert.



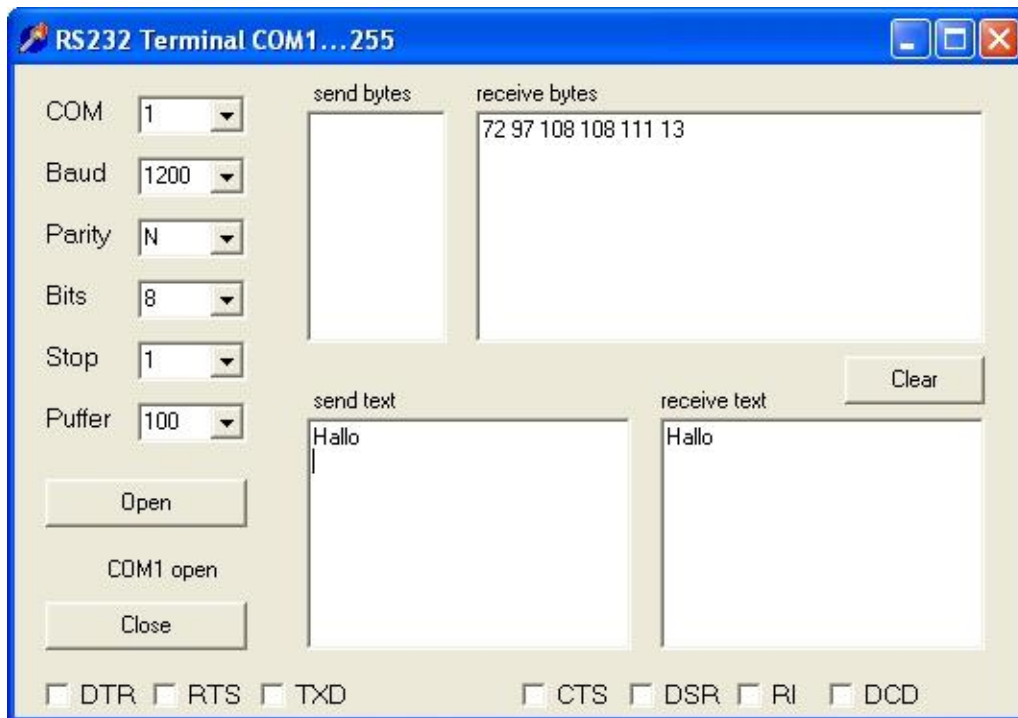
Update des Terminal.exe bis COM255



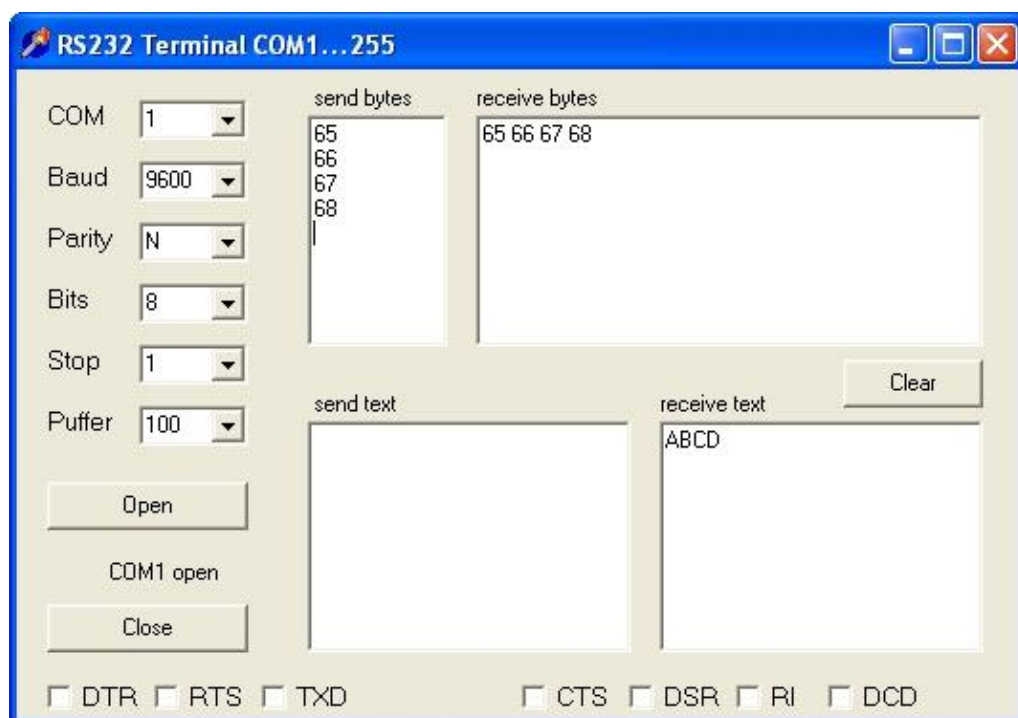
Download: Terminal.exe

Das

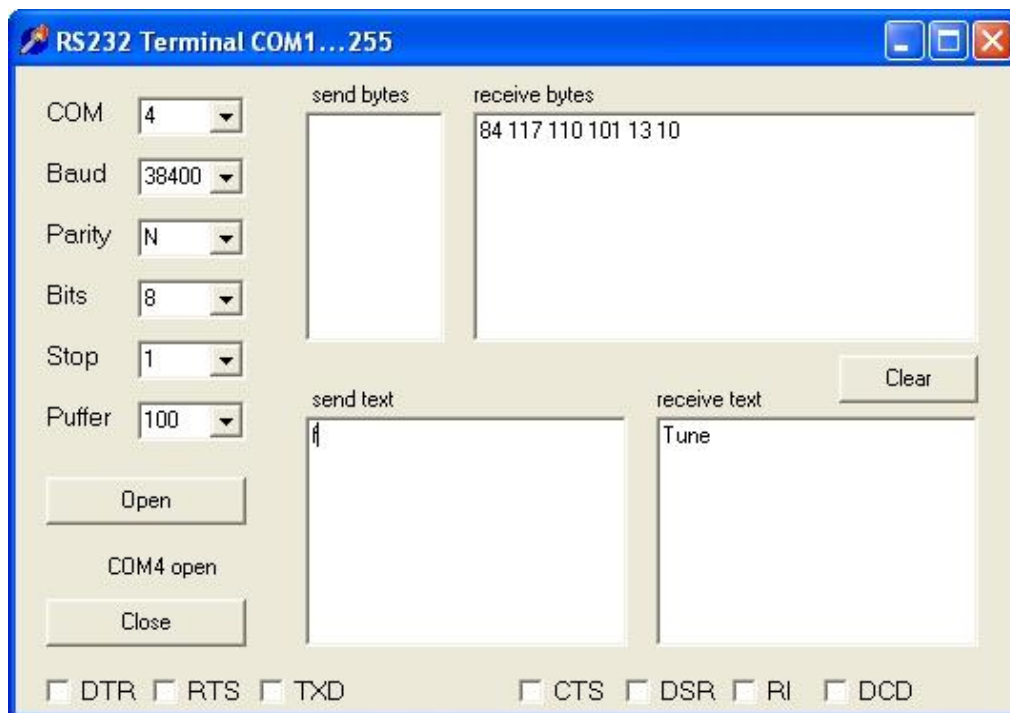
Programm arbeitet stand alone und kann in ein beliebiges Verzeichnis entpackt werden. Beim Start ist die Schnittstelle geschlossen. Wählen Sie die gewünschte COM-Nummer und die Baudrate aus. Für die Com-Nummern 1 bis 4 existieren Einträge im Drop-Down-Menü, höhere Nummern können per Hand editiert werden. Klicken Sie dann auf Open. Achtung, jede Änderung der Einstellungen wird erst wirksam, wenn Sie erneut auf Open klicken.



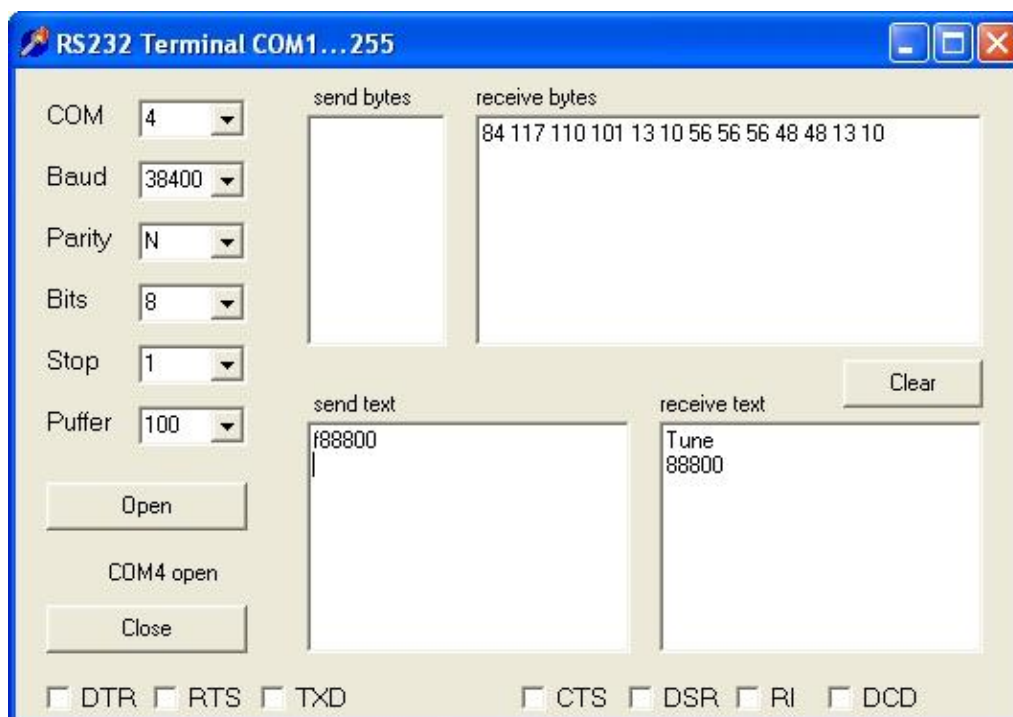
COM1 ist jetzt mit 1200 Baud geöffnet. Verbinden Sie für einen ersten Test TXD (Pin3) mit RXD (Pin 2). Senden Sie einen Text. Jedes in das Text-Fenster eingegebene Zeichen wird sofort gesendet und wieder empfangen. Im Byte-Fenster sehen Sie die Darstellung der ASCII-Zeichen.



In einigen Fällen müssen Einzelbytes gesendet werden. Geben Sie jeweils eine Zahl ins Byte-Fenster ein und bestätigen Sie die Eingabe mit Enter. Das Byte wird gesendet und wieder empfangen. Auch diesmal kommen beide Darstellung, die Bytes im Byte-Fenster und zu zugehörigen Textzeichen im Text-Fenster. Die Zeichen A...D werden bekanntlich durch die ASCII-Werte 65 ... 68 repräsentiert.



Hier wird der Elektor-DSP-Empfänger aus Heft 7/8-2010 angesteuert. Für die Kommunikation ist die Textübertragung mit 38400 Baud definiert. Das Kommando f88800 stellt den Empfänger auf die UKW-Frequenz 88,8 MHz ein. Auf das kleine "f" reagiert der Empfänger mit der Antwort "Tune". Danach folgt der Zahlenwert (Frequenz in kHz) und Enter. Alle eingegebenen Zeichen werden als Echo vom Empfänger zurückgesendet. Man kann also sehen, dass alles richtig angekommen ist.



Und richtig: Der Empfänger wurde auf die gewünschte Frequenz abgestimmt. Zusätzlich sieht am im Display die Signalstärke in dBuV, das SNR in dB, die RDS-Stationskennung und die über RDS empfangene Uhrzeit.



Quelltext

Das Programm Terminal.exe wurde mit Delphi entwickelt. Ältere Vorgänger-Versionen konnten nur bis COM9 eingesetzt werden. Die entscheidende Neuerung liegt in der Unit RSCOM.Pas.

```
function OPENCOM (OpenString: pchar): Integer;  
  
var S, OpenStr, PortStr, Parameter :String;  
  
    DCB: TDCB;  
  
begin  
  
    Result := 0;  
  
    if PortHandle > 0 then CloseHandle(PortHandle);  
  
    OpenStr := OpenString;  
  
    S := copy (OpenStr,5,1);  
  
    if S=':' then begin    //COM1:  
  
        PortStr := copy (OpenStr,1,4);  
  
        Parameter:= copy (OpenStr,6,Length(OpenStr)-6);  
  
    end;  
  
    S := copy (OpenStr,6,1);  
  
    if S=':' then begin    //COM10:  
  
        PortStr := copy (OpenStr,1,5);  
  
        PortStr := '\\.\' + PortStr;  
  
        Parameter:= copy (OpenStr,7,Length(OpenStr)-7);  
  
    end;
```

```

S := copy (OpenStr,7,1);
if S=':' then begin    //COM100:
    PortStr := copy (OpenStr,1,6);
    PortStr := '\\.\' + PortStr;
    Parameter:= copy (OpenStr,8,Length(OpenStr)-8);
end;

PortHandle:=CreateFile(PChar(PortStr),GENERIC_READ or
GENERIC_WRITE,0,NIL,OPEN_EXISTING,0,0);

GetCommState(PortHandle,DCB);
BuildCommDCB(PChar(Parameter),dcb);
DCB.Flags := 1;
if SetCommState(PortHandle,DCB)then Result := 1;
TimeOuts (300);
end;

```

Beim Öffnen wird ein Handle verwendet, also eine Zahl vom Typ Integer, die zur eindeutigen Zuordnung des verwendeten IO-Kanals dient. Das Handle wird vom System entsprechend der Reihenfolge von CreateFile-Aufrufen vergeben. Da WriteFile für sehr unterschiedliche Aktionen und Geräte verwendet werden kann, unterscheidet das Betriebssystem diese mithilfe des Handles. Entsprechend gibt CloseHandle die entsprechende Schnittstelle wieder frei.

Ursprünglich konnten nur die Schnittstellen COM1 bis COM9 geöffnet werden. Höhere Schnittstellen wie z. B. COM51 heißen unter Windows etwas anders, nämlich z. B. "\\.\COM51". Der Gerätenamen wird hier automatisch erweitert. Möglich sind jetzt COM-Nummern bis 255. Von außen kann die Funktion so aufgerufen werden: OPENCOM „COM255:1200,N,8,1“.

Download: Delphi-Quelltext