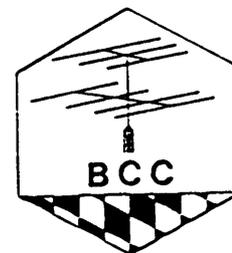


# Bavarian Contest Club



Bernhard Steibl  
DF7RX  
Kelheimwinzerstr. 40  
93309 Kelheim  
Tel. 089/7242379  
(Montag - Donnerstag)

Kelheim, den 21. Oktober 1995

## Siehe BCC Mitglieder !

### WWDX Contest

Es ist mal wieder soweit, die beiden WWDX-Conteste stehen unmittelbar bevor. Wie wir aus der CQ entnehmen können, ist es uns im letzten Jahr gelungen, ganz knapp die Club-Competition zu gewinnen. Alle die daran mitgewirkt haben herzlichen Glückwunsch zu dem Erfolg. Ich hoffe, daß auch heuer, mit genausoviel Begeisterung die Sache angegangen wird wie letztes Jahr und es uns wieder gelingt die Clubwertung zu gewinnen.

#### SSB-Teil

- **TK2C** Multi/Multi mit DF4RD, DF7RX, DK2OY, DK6WL, TK5EP, TK5MH und TK5NN
- **CT3EE** Multi/Single mit CT3EE, DK7YY (ex DL7UTA) und DL3DXX

#### CW-Teil

- **TK2C** Multi/Multi mit DL2NBU, DL4RDJ, DL6RAI, TK5EP, TK5NN, UA2FJ und UA2FM

### WWDX-Logs

Wie alle Jahre können auch heuer wieder die WWDX-Papier- oder Disketten-Logs (mit Angabe der CT-Version 8 oder 9, BIN-File und CTY-File) zu mir geschickt werden.

*Einsendeschluß SSB-Teil: 23. November*

*CW-Teil: 4. Januar*

Ich werde die Logs dann gesammelt zur CQ schicken. Wer sein Log selbst einreicht, sollte nicht vergessen, bei der Club Competition **Bavarian Contest Club** anzugeben und mir die Punktzahl (für Club Competition Summenblatt) melden.

### BCC Meteorscatter Contest

Auch heuer lädt der BCC alle meteorscatterbegeisterten Funkamateure, zu dem in den letzten Jahren zu einer Institution gewordenen Meteorscatter Contest ein. Er findet heuer von 11. Dezember 18.00z bis 15. Dezember 02.00z statt.

## Datenübertragung mit Packet Radio

von Ben Büttner, DL6RAI

"Was soll denn so ein Artikel in unserem Rundbrief?", wird sich so mancher von Euch fragen. "Binärdateien übertragen, das ist doch nur was für die Programmsauger auf DB0MWS. Sowas mache ich nicht!" So, oder so ähnlich wird sich vielleicht mancher zu dem Thema äußern und weiterblättern.

### 1. Verschiedene Arten von Dateien

Nun, die Contester und DXer nutzen immer häufiger EDV, um z.B. Logbuch zu führen, Conteste auszuwerten, Checklisten zu verwalten, Packet Radio zu machen und andere Dinge mehr. Häufig tritt hier das Problem auf, daß Dateien irgendwelcher Art von X zu Y gelangen sollen, z.B. eine neue Programmversion, ein K1EA .BIN-File zur Logauswertung, eine neue CQWW.CTY-Datei für den bevorstehenden Contest usw. Die einfachste Methode ist die Diskette, entweder mit der Post schicken oder persönlich übergeben. Aber was tun, wenn es eilig ist, oder wenn man ein paar Mark Porto für die Sendung sparen will? Da wäre jetzt recht, wenn man wüßte, wie man am schnellsten die Dateien über Packet übertragen könnte, ohne daß etwas verlorenggeht.

Die Schwierigkeit beim Übertragen von Dateien ist immer, daß der Inhalt nicht verfälscht werden darf. Grundsätzlich gibt es in diesem Zusammenhang zwei Typen von Dateien: ASCII-Dateien und Binär-Dateien. ASCII-Dateien können grundsätzlich mit einem EDITOR (z.B. DOS EDIT) bearbeitet und verändert werden. Solche Dateien sind z.B. CQWW.CTY, CTY.DAT oder die .RES-Dateien von K1EA. Solche Dateien kann man mühelos als Nachricht über das Mailbox-Netz verschicken oder auch direkt übertragen. Gelegentlich muß man hinterher mit einem Editor irgendwelche Mailbox-Header oder Prompts löschen, aber die Datei ist komplett und unverfälscht übertragen.

Binär-Dateien sind dagegen nicht so einfach zu handhaben. Dies sind z.B. .BIN-Files von CT oder Programme (.EXE und .COM-Files). Der Unterschied ist, daß ASCII-Dateien nur Zeichen aus einem bestimmten Symbolbereich enthalten (nur die Buchstaben des Alphabets, Ziffern, Satzzeichen jedoch keine Steuerzeichen oder Grafiksymbole). Die Zeichen kommen alle aus dem Bereich ASC 32..127. Binärdateien können dagegen alle im Zeichensatz überhaupt möglichen Zeichen (ASC 0..255) enthalten. Manche dieser darin vielleicht enthaltenen Zeichen würden aber vom Packet-Programm, vom TNC oder von der Mailbox als Steuerzeichen interpretiert werden und können deshalb nicht direkt ausgesendet werden. z.B. wird von den meisten Mailboxen das Zeichen control-Z (ctrl-Z, ASC 26) als Dateiendezeichen interpretiert. Die übertragene Datei würde also genau an dieser Stelle, wo das ctrl-Z vorkäme, beendet werden. ctrl-S und ctrl-Q dienen häufig der Datenflußkontrolle zwischen TNC und Computer und können deshalb auch nicht so übertragen werden. Auch kann man Binärdateien nicht einfach mit einem Editor bearbeiten, denn auch der hätte Probleme mit den Steuerzeichen.

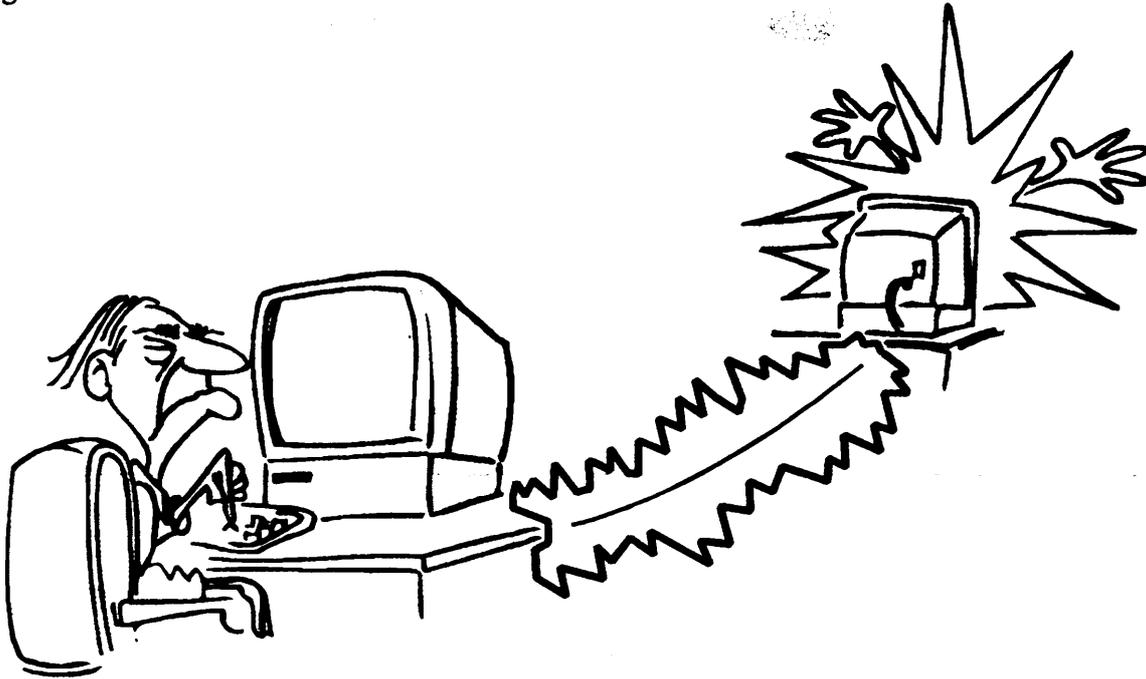
Wir merken uns also: Binär-Dateien muß man vorsichtiger, in jedem Fall anders behandeln wie ASCII-Dateien. ASCII-Dateien erkennt man daran, daß sie mit dem DOS-Editor EDIT bearbeitet werden können. Ggf. kann man ASCII-Dateien mit dem Editor reparieren. Binärdateien kann man nicht editieren.

### 2. Punkt-zu-Punkt-Übertragung mit SP

Ich benutze für meine Beispiele das Programm SP, das bei mir seit vielen Jahren im Einsatz ist. Neuere Programme, wie z.B. GP oder TNT haben ähnliche Funktionen, manchmal sogar die gleichen Steuerkommandos.

Die einfachste Methode, eine Datei zu übertragen ist der direkte Weg. Station X connectet Station Y, beide vereinbaren die Übertragung Y schaltet die Speicherung ein, X beginnt mit der

Aussendung und wenn alles übertragen ist, schaltet Y die Speicherung wieder aus. Bildhaft dargestellt sieht der Dateitransfer so aus:




---

Station X

---

Station Y

---

Connectet Y

Schreibt Y, daß er jetzt die Datei senden will.

Schaltet mit Alt-S das Speichern ein.

Sendet die Datei mit dem SP-Kommando "<ESC> ST". Tippt man nur "<ESC> ST" so erscheint die Liste der vorhandenen Dateien und man kann bequem auswählen. Man kann aber auch schreiben "<ESC> ST CQWW.CTY" um die Datei CQWW.CTY zu senden. Am Ende sendet er ein Klingelzeichen an Y und teilt mit, daß alles übertragen ist.

Schaltet mit Alt-S das Speichern wieder ab

---

Die Datei ist nun übertragen und befindet sich bei Y auf dem Computer im SP-Unterverzeichnis SAVE, bei mir z.B. in C:\APPLIC\SP\SAVE.

Was aber, wenn Y gerade nicht Zuhause ist, oder nicht am Computer sitzt? Bei SP gibt es sog. Fernsteuerkommandos, mit denen X das Programm von Y steuern kann. Somit braucht Y gar nicht mehr direkt am Bildschirm zu sitzen, sondern X kann die Übertragung von seiner Seite aus komplett fernsteuern. Diese Fernsteuerbefehle, manchmal auch hochtrabend "Remote-Befehle"

genannt, werden üblicherweise mit `"/"` eingeleitet. Sehen wir uns nun einen allein von X eingeleiteten Dateitransfer an.

Station X	Station Y
Connectet Y	
Sendet <code>"/write cqww.cty"</code> und wartet auf die Antwort von Y's Programm. Es kommt eine Meldung in etwa mit folgendem Wortlaut: <code>"Bitte Datei CQWW.CTY senden, abschließen mit ctrl-Z"</code> .	
Sendet die Datei wieder mit dem SP-Kommando <code>"&lt;ESC&gt; ST"</code> .	
Beendet die Übertragung mit ctrl-Z.	

Wie wir sehen, hat Y gar nichts zur Übertragung getan. Voraussetzung dafür ist, daß bei Y die Fernsteuerbefehle eingeschaltet sind. Dies wird bei SP in der Datei CONFIG.SP erledigt, dort muß der Parameter REM auf 1 gesetzt sein. Steht er auf 0 so ignoriert SP die `"/`-Kommandos der Gegenseite stillschweigend.

Was tun wir nun bei Binärdateien? Wie wir vorhin gehört haben, kann man diese wegen den darin vorkommenden Steuerzeichen nicht so einfach senden. Bei SP gibt es die Möglichkeit, mit einem speziellen Protokoll, Binärdateien zu transferieren. Dabei werden die Steuerzeichen vom sendenden SP so verändert, daß sie ohne weiteres übertragen werden können. Das empfangende SP wiederum verwandelt alle speziell maskierten Zeichen wieder in ihre Grundform zurück. Damit das korrekt funktioniert, muß die Übertragung von einer Seite aus gesteuert werden, also wieder Fernsteuerung. Dazu dient der Fernsteuerbefehl `"/wprg"`. Hierbei ist zu beachten, daß zwei Dateinamen anzugeben sind, und zwar einmal so, wie die Datei bei mir genannt wird und dann so, wie sie auf der Gegenseite heißen soll. Meistens wird man den gleichen Dateinamen auf beiden Seiten verwenden, also zweimal den gleichen Namen angeben. Wichtig ist auch noch, daß bei Y die Datei in dem SP-Unterverzeichnis USER liegen muß, sonst wird sie nicht gefunden.

Station X	Station Y
Connectet Y	
Sendet den Befehl <code>"/wprg rai.bin rai.bin"</code> Alles weitere geht automatisch. Die korrekte Übertragung der Datei wird am Ende angezeigt. Während der Übertragung kann man an dem Byte-Zähler in der Statusleiste sehen, wieviele Bytes schon Übertragen wurden bzw. noch zu senden sind.	

Nun hat Y in seinem SP-Unterverzeichnis ebenfalls eine Datei mit dem Namen `"RAI.BIN"` stehen, die exakt die gleiche Größe und den gleichen Inhalt hat wie bei X. Übrigens kann man den

Transfer auch von der Gegenseite aus starten, dazu müßte Y eingeben `"/rprg rai.bin rai.bin"`. Alles weitere ginge wieder automatisch, bis die Übertragung beendet ist.

### 3. Datenkompression - wozu brauche ich das?

Wir haben nun schon einiges gelernt, und mit den vorangegangenen Methoden können wir nun schon ASCII- und Binärdateien übertragen. Die zu übertragenden Dateien haben meistens gemeinsam, daß sie verhältnismäßig groß sind. Auf unseren belasteten Packet-Frequenzen sind effektive Transferraten von 100 Baud keine Seltenheit. Deshalb ist froh man um jedes Byte, das nicht übertragen werden muß. Hier kommen nun die sogenannten Kompressionsprogramme ins Spiel. Dies sind üblicherweise als Share- oder Freeware erhältliche Programme, die beliebige Dateien in eine Archivdatei zusammenpacken können und dabei - das ist das wesentliche - die Datenmenge z.T. erheblich schrumpfen lassen.

Ein bekanntes und häufig eingesetztes Kompressionsprogramm ist PKZIP in der neuesten Version 2.04. Logdateien von K1EA lassen sich mit PKZIP um erstaunliche 80% packen. Das heißt, aus einer 200 kB großen Logdatei wird eine 40 kB große ZIP-Datei. Noch krasser werden die Verhältnisse bei ASCII-Dateien. .RES-Files lassen sich beispielsweise auf 10% ihrer Originalgröße bringen.

Wie kommt das? Ein bekanntes und anschauliches Verfahren zur Datenkomprimierung ist die sog. Lauflängenkodierung. Hierbei wird die Datei als Folge von Bytes aufgefaßt. Folgen z.B. zehn Leerzeichen hintereinander, so werden im Archiv nicht zehn sondern nur ein Leerzeichen und ein Zähler mit dem Inhalt "10" gespeichert. ASCII-Dateien enthalten oft viele gleiche Zeichen; insbes. Leerzeichen kommen sehr häufig vor. Deshalb sind diese Dateien auch so gut zu komprimieren. Auch Binärdateien bestehen oft aus gleichen Mustern wenn auch hier andere Regelmäßigkeiten zu finden sind. Man kann sich vorstellen, daß in einem Kontest-Logfile die hintereinander gespeicherten QSOs eine große Ähnlichkeit aufweisen. Datum und Uhrzeit sind bis auf eine Stelle gleich, Band und Betriebsart sind gleich, die beiden VFO-Frequenzen sind gleich usw. Deshalb kann man auch hier gut komprimieren.

Kompressionsprogramme benutzen meist mehrere verschiedene Verfahren, um die Daten kleinzukriegen. Die vorliegenden Daten werden zunächst untersucht und dann das günstigste Verfahren ausgewählt.

Hier ein Beispiel zur Anwendung des Programms PKZIP:  
`C:\USR\DL6RAILOGS\TK5EL>dir * .bin`

Datenträger in Laufwerk C ist /usr/dos  
 Verzeichnis von C:\USR\DL6RAILOGS\TK5EL

```
WW94SSB BIN 208227 11-19-94 11:13p
  1 Datei(en) 208227 Byte
  16924672 Byte frei
```

`C:\USR\DL6RAILOGS\TK5EL>pkzip tk5el ww94ssb.bin`

PKZIP (R) FAST! Create/Update Utility Version 2.04g 02-01-93 Copr. 1989-1993  
 PKWARE Inc. All Rights Reserved. Shareware Version PKZIP Reg. U.S. Pat. and Tm. Off.  
 Patent No. 5,051,745

```
= 80486 CPU detected.
= EMS version 4.00 detected.
= XMS version 3.00 detected.
= Using Normal Compression.
```

Creating ZIP: TK5EL.ZIP  
Creating: WW94SSB.BIN Deflating (79%), done.

C:\USR\DL6RAI\LOGS\TK5EL>dir \*.zip

Datenträger in Laufwerk C ist /usr/dos  
Verzeichnis von C:\USR\DL6RAI\LOGS\TK5EL

TK5EL	ZIP	44402	10-15-95	3:40p
	1 Datei(en)	44402	Byte	
		16875520	Byte frei	

Wie wir sehen, ist soeben die Datei WW94SSB.BIN in dem Archivfile TK5EL.ZIP verpackt worden. Das entstehende Archivfile kann nun übertragen werden und zwar in einem Fünftel der Zeit, die man für die Originaldatei benötigt hätte. Die entstehenden .ZIP-Dateien sind Binär-Dateien. Man kann sie mit den vorher beschriebenen Methoden übertragen.

Auspacken tut man auf der Empfängerseite mit dem Gegenstück zu PKZIP, genannt PKUNZIP.

C:\ASP\USER>pkunzip tk5el.zip

Es entsteht wieder die Originaldatei, der Name des Originals sowie Entstehungsdatum usw. sind in dem Archiv gespeichert und werden beim Auspacken restauriert. Das übertragene ZIP-Archiv kann nach erfolgreichem Auspacken gelöscht werden, es wird nicht mehr benötigt.

PKZIP und PKUNZIP können im Übrigen noch viel mehr interessante Dinge tun. Man kann damit mehrere Dateien oder sogar ganze Verzeichnisstrukturen in ein Archiv verpacken. Näheres entnimmt man der Dokumentation zu PKZIP. Neben PKZIP gibt es noch andere Komprimierungsprogramme. Der Vollständigkeit halber seien genannt: LHARC (\*.ARC), ARJ (\*.ARJ) und ZOO (\*.ZOO).

Der Teil II dieses Beitrags folgt in der nächsten Ausgabe. Es geht um Fernübertragung von Dateien in Mailbox-Netz