Erweitertes Handbuch

für TNC2/3
Packet-Radio-Controller

Ausgabe 09.11. 2002

Herstellung und Vertrieb: SYMEK GmbH, Datentechnik, Ulf Kumm, DK9SJ
Anschrift: D-70597 Stuttgart (Sonnenberg), Johannes-Krämer-Straße 34
Tel: (0711) 76 78 923 Fax: 76 78 924 Hotline: 76 54 911 Packet: DK9SJ@DB0SAO
Internet: http://symek.de eMail: ulf@symek.de

Inhaltsverzeichnis

Vorwort zu diesem Handbuch

Falls die Informationen im Handbuch des TNC2S / TNC2H / TNC21S oder TNC3S für Ihren Anwendungsfall nicht ausreichen, haben wir hier ein "erweitertes Handbuch" zusammengestellt. Es enthält verschiedene Beiträge aus dem Bereich Packet-Radio-Software, die zu speziell sind, um in den standardmäßigen Benutzerhandbüchern der Geräte abgedruckt zu werden.

09.11.02 Ulf Kumm, DK9SJ

Bedienung des TNC2 mit TAPR-EPROM

Das TAPR-Programm ist im EPROM des TNC2S enthalten und wird aktiviert, wenn der Schalter 6 (von links gezählt) bzw. Schalter 4 des TNC2H nach oben geschaltet ist. Die TAPR Software ist für den Betrieb eines TNC an einem Terminal eingerichtet. Das Terminal hat die Aufgabe, alles, was auf der seriellen Schnittstelle empfangen wird, auf dem Bildschirm anzuzeigen. Außerdem werden alle Zeichen, die von der Tastatur eingegeben werden, zum TNC geschickt.

Diese relativ einfache Aufgabe erledigt am besten ein Terminal. Für praktisch alle Rechner gibt es jedoch auch Terminalprogramme ("Terminalemulatoren"), die den Rechner zum Terminal machen, wie z.B. "Telix" für einen PC oder den "VT52-Emulator" beim Atari. Ein gutes Terminalprogramm hat eventuell noch den zusätzlichen Vorteil, daß man empfangene Texte speichern und später ausdrucken kann.

Die TAPR-Software für TNC wird, nachdem fast alle Programme den Hostmode verwenden, kaum mehr eingesetzt. Die Kommandos sind daher hier nur teilweise aufgelistet.

Im folgenden ist die Bedienung des TNC mit einem solchen Terminal beschrieben.

Vorbereitung

Zuerst muß das Terminal (bzw. der Rechner mit Terminalprogramm) an das TNC angeschlossen werden. Darüber steht im Abschnitt "Anschluß des TNC2S an ..." des Handbuches mehr. Auf jeden Fall muß erst mal die Einschaltmeldung der TAPR-Software richtig auf dem Bildschirm erscheinen. Wenn du etwas eintippst, muß dieser Text auch auf dem Bildschirm erscheinen. Außerdem solltest du das Funkgerät an das TNC angeschlossen haben, sonst kann man ja nichts ausprobieren.

Command-mode und Konvers-mode

Nach dem Einschalten des TNC meldet sich die TAPR-Software mit ihrer Einschaltmeldung, am Ende der Meldung steht 'cmd:'. Das TNC befindet sich im Command-mode, es wartet darauf, daß du ihm ein Kommando sendest, das es versteht und ausführen kann. Das TNC versteht 113 Kommandos, von 'ACKPRIOR' bis 'XON', aber keine Angst: die 10 Kommandos, die zum 'normalen' Packet-Betrieb notwendig sind, kennst du bald auswendig.

Im Konvers-mode sendet man keine Kommandos an das TNC, sondern man sendet Texte an die Gegenstation. Normalerweise funktioniert das also so:

```
CONNECT DF3SO [Kommando: eine Verbindung aufbauen]
```

jetzt schaltet das TNC automatisch in den Konvers-mode um. Alles was du nun eintippst, z.B.

```
Hallo, altes Haus, wie geht es Dir denn so?
```

betrifft nicht das TNC (das könnte wohl auf diese Frage auch keine gescheite Antwort geben), sondern die Gegenstation. Ist das QSO jedoch zu ende, dann muß man dem TNC sagen, daß die Verbindung wieder aufgelöst werden soll. Dazu schaltet man erst in den Command-mode und das TNC meldet sich mit

```
cmd:
```

Nun tippt man

```
DISCONNECT
```

(das versteht das TNC nämlich wieder) und die Verbindung wird getrennt.

Beim Arbeiten im TAPR Mode muß man immer wissen, ob man im Konvers-mode ist, also mit der Gegenstation spricht, oder ob man sich im Command-mode mit dem TNC 'unterhält'.

Transparente Datenübertragung mit TAPR Software

Übertragung binärer Daten

Die TAPR-Software eignet sich zur Übertragung von ASCII-Zeichen, wobei allerdings einige Zeichen eine spezielle Funktion besitzen (z.B. Return, Backspace, Ctl-C etc.). Will man nun z.B. Programme oder andere binäre Daten übertragen, so kann man natürlich nicht ausschließen, daß diese Daten solche Steuerzeichen enthalten. Eine beliebte und praktische Abhilfe ist die 7-plus Codierung: Eine Datei wird durch ein 7-plus Programm so umgewandelt, daß keine Steuerzeichen mehr darin vorkommen. Außerdem werden die binären Daten so codiert, daß 7 Bit zur Übertragung ausreichen. Zusätzlich werden Prüfsummen, Dateinamen etc. in die 7-plus-codierte Datei mit verpackt. Diese Daten kann man mit allen üblichen Packet-Programmen übertragen, in Mailboxen speichern etc. Welche TNC-Software man verwendet ist dabei egal. Notwendig ist ein Rechner mit einem 7-plus Programm.

Falls jedoch binäre Daten direkt zwischen zwei TNC ausgetauscht werden sollen, kann man den in der TAPR-Software vorgesehenen Transparentmode benutzen. Einmal im Transparentmodus, überträgt das TNC **alle** 256 möglichen ASCII Zeichen ohne Ausnahme. Die Daten kommen beim Empfänger genau so wieder zum Vorschein, wie sie von der sendenden Station eingegeben wurden. Eine typische Anwendung wäre z.B. wenn man einen Drucker, ein Meßgerät oder ein Terminal drahtlos an einen Rechner anschließen möchte. Eine transparente Datenverbindung unterscheidet sich nicht von einem direkten Kabelanschluß, mit Ausnahme natürlich der Übertragungsgeschwindigkeit.

Start des Transparentmodus

Eine Packet-Verbindung wird wie üblich aufgebaut, indem man einen connect durchführt:

```
C DF3SO      [Kommando: eine Verbindung aufbauen]
```

ist die Verbindung (im Konvers-modus) hergestellt, schaltet man in den Transparentmodus:

zuerst mit Control-C wieder in den command-mode wechseln und dann

```
T            [Kommando: in Transparentmode wechseln]
```

eingeben. Das Kommando entspricht dem K-Kommando (Wechsel von Command in Konvers), schaltet jedoch statt in Konversmode in den Transparentmode.

Alles, was jetzt eingetippt wird (und nur das), erscheint genau so bei der Gegenstation und umgekehrt. Die eigenen Zeichen werden nicht ge-echot, alle Steuerzeichen werden unverändert übertragen.

Da es im Transparent-mode kein 'Return' als Steuerzeichen gibt, das die Absendung eines Pakets veranlaßt, werden die Daten in regelmäßigen Zeitabständen gesendet. Siehe Kommando "PACTIME"

Beenden des Transparentmodus

Nachdem das TNC im Transparentmodus auf keinerlei Steuerzeichen reagiert, ist es nur über einen 'Trick' möglich, den Transparentmode wieder zu verlassen. Es gibt zwei Möglichkeiten:

1. Der Rechner sendet ein Break-Zeichen (tastet die RS232-Schnittstelle für einige ms auf +10 Volt). Das TNC verläßt den Transparentmode und meldet sich wieder mit cmd:

2. Nach einer Wartezeit (wird mit dem Kommando CMDTIME festgelegt, Default 1 sec.) sendet der Rechner in rascher Folge (weniger als CMDTIME Zeitabstand) dreimal Control-C und wartet danach wieder die in CMDTIME festgelegte Zeit. Danach meldet sich das TNC wieder mit cmd:. Wird CMDTIME =0 gesetzt, ist ein Verlassen des Transparentmodes nur mit Break (Methode 1) möglich.

Für Transparentmodus wesentliche Kommandos

(siehe auch alphabetische Auflistung der Kommandos)

Transpar, PACTime, PARity, Paclen, TRFlow, TXFlow, CMdtime, CONMode, NEwmode, Xflow, CONPerm

Programmierungsbeispiel für Transparentmode

Das folgende Beispiel zeigt, wie man mit zwei TNC2 mit der TAPR Software eine drahtlose transparente Datenverbindung herstellen kann.

1. RESET Kommando ausführen. Schafft definierte Bedingungen.

2. bei beiden TNC2 die Rufzeichen mit dem MYcall Kommando definieren.

3. bei beiden TNC CONMode Transpar setzen

4. ggf. bei beiden TNC AWlen 8 setzen und PARity 0. (muß nicht sein) Je nach Anwendung PACTime, PACLen etc. einstellen.

5. Verbindung herstellen (connect zwischen den beiden TNC)

6. jetzt kann die transparente Datenverbindung getestet werden.

Permanente Verbindungen

Eine einmal bestehende Verbindung kann man mit dem Kommando CONPerm gespeichert werden. Man kann das obige Beispiel fortsetzen:

7. BREAK-Signal senden, damit TNC in Commandmode geht. Dann CONPerm ON bei beiden TNC einschalten. Mit Kommando T wieder zurück in den Transparentmode.

8. Stromversorgung der TNC abschalten. Sobald man die TNC wieder einschaltet, versuchen die Geräte die unterbrochene Verbindung wiederherzustellen, die grünen LED leuchten wieder, der Transparentmode ist wieder wie vor der Unterbrechung aktiv. Die Unterbrechung der Verbindung wird mit der Meldung "*** LINK OUT OF ORDER, possible data loss" kommentiert.

TIP: nach dem Wiedereinschalten der TNCs kommt die (eventuell störende) Einschaltmeldung. Man kann durch eine kleine Änderung im EPROM die Einschaltmeldung unterdrücken. Die Adressen 0290 bis 0297 in der entsprechenden Hälfte des EPROMs lauten $CD, xx, xx, 20, 02, 18, EF, CD. Die Inhalte $20 und $02 der Adressen 0293 und 0294 werden auf $00 und $00 geändert. Beim Einschalten kommt dann lediglich die Anzeige, auf welchem Kanal das TNC steht (|A).

Optimierung der Übertragungsgeschwindigkeit

Bei Packet-Betrieb über normale Digipeater sollte man die Parameter seines TNC möglichst unverändert lassen. Nur dann ist die Chancengleichheit der Benutzer einigermaßen gewährleistet. Für spezielle Übertragungsaufgaben auf einer eigenen Frequenz kann die Übertragungsgeschwindigkeit durch Anpassung der TAPR-Parameter optimiert werden. Dies ist besonders dann wichtig, wenn große Datenmengen gesendet werden.

PACLEN: Damit möglichst viel Daten pro Paket gesendet werden, macht man die Paketlänge möglichst groß. Das Maximum sind 256 Bytes (PACLEN 0). Ist es allerdings häufig, daß während so eines langen Pakets Fehler auftreten, sollte die Paketlänge reduziert werden.

MAXFRAME: Bei MAXFRAME 1 muß jedes Paket erst bestätigt werden, bevor das nächste gesendet wird. Mit MAXFRAME 7 sendet man gleich 7 Pakete am Stück und wartet auf die Bestätigung des 7. Pakets, dann werden sofort die nächsten 7 Pakete gesendet. Verringert die Umschaltung wesentlich und beschleunigt den Datenfluss.

FRACK: Wenn die Gegenstation einmal keine Bestätigung schickt, wartet der Sender diese Zeit bis er das unbestätigte Paket erneut losschickt. Je kürzer FRACK, desto kürzer die Wartezeit bis zur Wiederholung. Bei CONPERM ON versuchen sich die TNCs in FRACK-Zeitabständen neu zu connecten.

TXDELAY: Natürlich muß TXDELAY auf ein Minimum eingestellt werden. Der optimale Wert wird ausprobiert und hängt vom verwendeten Sender und Empfänger ab. TXDELAYC bestimmt die minimale Anzahl Flags, die vor jedem Paket gesendet werden (Minimum 1)

DWAIT: Nach einer Sendung wird erstmal DWAIT lange gewartet bevor das TNC wieder sendet. Falls man eine Strecke ohne Digipeater nur mit 2 TNCs verwendet, kann DWAIT bis auf 0 reduziert werden.

ACKTIME: Verzögerung bis zur Aussendung der Empfangsbestätigung. Kann auf Minimum (testen) eingestellt werden. ACKPRIOR ON gibt den Bestätigungspaketen Priorität.

DEADTIME: wird an die Geschwindigkeit der DCD des Empfängers angepasst. Für eine Funkstrecke zwischen 2 TNC auf einer Frequenz kann DEADTIME auf einen Minimalwert eingestellt werden.

SLOTS: bestimmt die Wahrscheinlichkeit, daß auf den Kanal zugegriffen wird. Mit SLOT 1 ist diese Wahrscheinlichkeit 100%, je größer die Kanalbelegung, desto größer sollte SLOTS gewählt werden. (Default 3, Maximum 127, vernünftige Werte bis 10)

RS232: Die RS232-Verbindung darf natürlich nicht langsamer sein als die Funkstrecke. Im Hostmode z.B. sendet das TNC erst dann die Bestätigung, wenn es seinen Puffer vollständig zum Rechner übertragen hat. Dies halbiert die Übertragungsgeschwindigkeit. Der angeschlossene Rechner muß in der Lage sein, die Daten ohne Verzögerung abzuschicken bzw. zu empfangen. Manche rechenintensiven Packet-Programme brauchen viel Zeit für die Darstellung der Zeichen auf dem Bildschirm (z.B. soft-scroll) und bremsen dadurch die gesamte Transfergeschwindigkeit merklich.

FULLDUP: Hat man die Möglichkeit, gleichzeitig zu senden und zu empfangen (getrennte Sender und Empfänger notwendig), so läßt sich die Geschwindigkeit nochmal deutlich steigern. Falls der Sender ständig getastet wird (PTT dauernd geschaltet), kann man TXDELAY=0 setzen.

Sonderzeichen zur Steuerung des TNC (TAPR)

Für das TNC haben einige der 32 Control-Zeichen eine besondere Bedeutung, wie z.B. Control-C zur Umschaltung Konvers auf Command-mode. Bei der TAPR Software ist es möglich, alle diese Steuerzeichen beliebig anders zu legen. Wenn Sie z.B. mit dem Zeichen % (statt ^C) von Konvers auf Command-mode umschalten wollen, können Sie das Umschaltzeichen von Control-C auf $25 (entspricht dem %-Zeichen nach Tabelle) ändern. Am besten ist meist, wenn man die Steuerzeichen unverändert läßt, denn sonst unterscheidet sich das speziell konfigurierte TNC von der gewohnten Norm und wird anders bedient als allgemein üblich. Trotzdem kann man sich manche Fingerverrenkungen sparen, wenn wichtige Kontrollzeichen auf leicht erreichbare Tastenkombinationen gelegt werden.

Die folgenden Kommandos dienen dazu, die ASCII-Zeichen für einige Spezialfunktionen des TNC zu definieren. Die Zeichen werden durch Angabe des ASCII-Wertes definiert; wenn man das Zeichen in hexadezimaler Schreibweise angeben möchte, setzt man ein $-Zeichen davor, ansonsten wird der Zahlenwert vom TNC als Dezimalzahl interpretiert. Wollen Sie z.B. das Umschaltzeichen von Konvers- in Commandmode von Control-C auf Control-N ändern, dann tippen Sie

```
COMMAND $0E bzw. COMMAND 14
```

das TNC antwortet (in hexadezimaler Darstellung) wie gewohnt mit

```
COMMAND was $03
```

Welche Steuerzeichen es gibt und wie sie gesetzt sind, läßt sich am einfachsten mit dem Kommando

```
DISP C
```

abfragen.

Die Kommandos des TAPR-EPROM

Anschließend sind die wichtigsten TAPR-Kommandos aufgeführt. Die Liste ist nicht vollständig. Einige Kommandos sind nur für den Packet-Betrieb über FM-Umsetzer vorgesehen, das ist bei uns aber nicht üblich. Wer die vollständige Beschreibung aller Kommandos der TAPR-Software braucht, der kann sich das 100-seitige Heft 'TAPR commands-booklet' (englisch) besorgen. In diesem Heft sind sämtliche TAPR-Kommandos umfassend beschrieben.

ACKTime [n] n = 0...250 * 10ms, Default = 14

Legt die Zeit fest, bis wann ein I-Packet bestätigt sein muss, bevor ein Reject ausgesendet wird.

AUtolf [ON|OFF]

ON: Bei Zeilenende wird <CR> und zusätzlich <LF> ans Terminal gesendet. Falls das Terminal bei Empfang von <CR> selbst schon einen Zeilenvorschub macht, erscheint bei ON zwischen zwei Textzeilen immer eine (unnötige) Leerzeile.

AWlen [n] n=7...8 Default: 7

Wortlänge der Schnittstelle TNC-Terminal: 7 oder 8 Bit. Falls hier ein falscher Wert angegeben wird, kann es sein, daß sich das TNC nicht mehr ansprechen läßt. (siehe RESET). Die AWlen-Einstellung wird erst nach einem RESTART oder Wiedereinschalten des TNC aktiv!

Beacon [Every|After n] n = 0...250 *10 s, Default = EVERY 0

Steuert die Aussendung eines Bakentextes (BText). Bei n = 0 ist die Bakenfunktion abgeschaltet. EVERY 6 bedeutet z.B.: alle 60 Sekunden BText senden, AFTER 30 bedeutet: Wenn 5 Minuten lang keine Packets gehört oder gesendet wurden, wird BText einmal ausgesendet. Der BText wird als UI-Paket an die in UNproto definierte Adresse gesendet.

BText [Bakentext] Länge max. 120 Zeichen Default=

Definiert den Text, der als Bake ausgesendet wird. Soll der Text ganz gelöscht werden, so gibt man BT % oder BT & ein. Beispiel für einen BText: BT Hier Peter aus Muenchen mit Übertragungstests 123456789ABCDEFG

BUdlist [ON|OFF]

ON: Alle Rufzeichen, die nicht in der LCAlls-Liste stehen, werden ignoriert.

OFF: Alle Rufzeichen, die in der LCAlls-Liste stehen, werden ignoriert.

CALibra

Startet den Testmodus. Nachdem man CAL eingegeben hat, kann man mit den Tasten <Space>, D, K und Q folgendes bewirken:

Q: Testmodus wird beendet, cmd: erscheint wieder

K: der Sender (PTT) wird getastet. Noch mal K drücken: Sender schaltet wieder ab. (Test der PTT oder der Watchdog-Schaltung). Läßt man den Sender länger als ca. 10 Sekunden getastet, dann schaltet er von alleine wieder ab (Watchdog), obwohl die rote PTT-LED leuchtet. Zweimal K drücken tastet den Sender für weitere 20 Sekunden.

<Space>: Wechselt die Aussendung von Dauereinslage auf Dauernulllage.

D: Es werden abwechselnd eine Null und eine Eins gesendet (01010101010101 ...). Noch mal D schaltet diese Funktion wieder aus. Diese Funktion ist für den Bit-Error-Test des TNC2H wichtig. Mit CAL, K und D bewirkt man dasselbe wie wenn die Brücke BERT gesteckt wird.

CALSet [n]

Abgleichhilfe für TNC mit analogem Modem und eingebautem Zähler. Wird beim TNC2 nicht benötigt, da das Modem quarzgesteuert ist und nicht abgeglichen werden muß.

CBell [ON|OFF]

ON: Wenn man connected wird, sendet das TNC folgende Zeichen zum Terminal: <Bell> *** Connected to call. (das Terminal piept oder klingelt)

OFF: Das <Bell>-Zeichen wird nicht angefügt, Terminal piept dann nicht.

CHeck [n] n = 0...250 *10 s Default: 12 = 120 s

Solange man connected ist, wird spätestens alle n*10 Sekunden ein Checkpacket ausgesendet. Solange die Gegenstation noch da ist, wird sie automatisch auf das Checkpacket antworten und die Verbindung bleibt bestehen. Antwortet sie RETry-mal nicht auf das Checkpacket, so nimmt das TNC an, daß die Gegenstation nicht mehr erreicht werden kann und beendet die Verbindung (retry count exceedet - Disconnected)

CLKADJ [n] n = 0...65535 Default = 0

Reguliert die eingebaute Software-Uhr. Ist n = 0, dann wird die Uhr nicht korrigiert. Ist n ungleich 0, so berechnet sich die relative Geschwindigkeit der Uhr nach: relativer Geschwindigkeit [%] = 100 % -- (9.16667* 1/n). Auch wenn CLKADJ optimal eingestellt ist, wird aus dem TNC keine besonders präzise Uhr.

CMdtime [n] n = 0...250*1 s Default = 1 s

Zum Verlassen des Transparentmodes muß erst CMdtime lang gewartet werden, dann müssen 3 COMMAND-Zeichen (Control-C) ans TNC geschickt werden, zwischen diesen Zeichen darf keine Pause länger als CMdtime sein. Danach muß wieder CMdtime lang gewartet werden. Erst dann verläßt das TNC den Transparentmodus. Werden die Zeiten nicht eingehalten, dann überträgt das TNC die eingegebenen Zeichen unverändert.

CMSg [ON|OFF]

ON: Wenn man connected wird, sendet das TNC zuerst die ConnectMessage (siehe CText) aus.

OFF: keine CText-Meldung bei Connect

CMSGDisc [ON|OFF]

ON: Wenn man connected wird, disconnected das TNC sofort wieder. Ist CMSG ON, so wird vorher der CText gesendet und dann disconnected.

OFF: kein automatisches Disconnect.

COMmand n (Default: $03 = Control-C)

Mit ^C schaltet man vom Konvers- in den Command-mode um. Außer ^C hat BREAK dieselbe Funktion. Anschließend erscheint das Commandprompt 'cmd:'.

CONMode [Convers|Transpar]

Convers: Nach Connect befindet sich das TNC im Konvers-mode (normal)

Transpar: Nach einem Connect befindet sich das TNC im Transparent-mode

Connect [Call [Via Call2 [Call3 [Call4 ...Call 9]]]]

Aufbau einer Verbindung zu Call, ggf. via die Digipeater Call2, Call3 etc. z.B. C DK4SE V DB0IE DB0EQ DB0ID. Zwischen den Rufzeichen kann ein Space oder ein Komma stehen (... VIA Call1, Call2, Call3)

CONOk [ON|OFF]

ON: Connect wird zugelassen

OFF: Connect wird nicht zugelassen. Anrufer bekommt '*** DL1ABC busy'

CONPerm [ON|OFF]

ON: Ein Disconnect ist nicht möglich. Die Verbindung wird sofort wieder aufgebaut.

OFF: Es kann normal connectet und disconnectet werden.

CONStamp [ON|OFF]

ON: Bei Connect wird Uhrzeit angezeigt. (Uhr ist mit DAytime gestellt) Anzeige z.B. *** CONNECTED TO DL4TA [11/01/91 19:45:20]

OFF: keine Anzeige der Uhrzeit nach *** CONNECTED TO ...

CONVers

Schaltet sofort vom Command in den Konvers-mode um. Siehe auch 'K'

CR [ON|OFF]

ON: Im Konvers-mode wird an jedes Paket, das z.B. mit Return abgeschickt wurde, ein <CR> (Wagenrücklauf) angehängt. Das nächste Paket erscheint dann beim Empfänger in einer neuen Zeile.

OFF: Nach Empfang eines Packets (Sender hatte Return gedrückt> geht der Text in derselben Zeile weiter.

CStatus

Anzeige aller 10 Kanäle A bis J in 10 Zeilen, wie z.B. B stream - IO Link state is: CONNECTED to DL3PA P
IO bedeutet: Input und Output auf diesem Kanal
P bedeutet: auf diesem Kanal ist CONPerm ON (kein disconnect möglich)

CText [text]

Eingabe des Begrüßungstextes (maximal 120 Zeichen lang), zum Beispiel: "Servus, hier ist Karl in Ulm. Bitte Mails an mich in DB0SAO ablegen" CText löscht man, indem man CT % oder CT & eingibt.

CWid [ON | OFF

Wenn CWID eingeschaltet ist, wird der in IDTEXT definierte Text alle 9 ½ Minuten im Morsecode gesendet. Je nach verwendetem Modem (NRZ/NRZI, G3RUH) kann diese Kennung jedoch so verändert werden, dass man akustisch nichts mehr erkennt.

DAytime [jjmmtthhmm]

Abfrage bzw. Anzeige der Uhrzeit im eingestellten Format. (siehe DAYUSA) Stellen der Uhr, z.B. DA 9101241123 für 24.1.1991, 11:23 Uhr Die Uhr muß jedesmal gestellt werden, wenn die Stromversorgung des TNC unterbrochen wurde!

DAYUsa [ON|OFF]

ON: Uhrzeitanzeige im Format MM/TT/YY hh:mm:ss , wie in USA üblich

OFF: Uhrzeitanzeige im Format TT-MM-YY hh:mm:ss , wie sonst üblich

DIGipeat [ON|OFF]

ON: Das TNC arbeitet als Digipeater, das heißt es sendet Packets wieder aus, in denen das eigene Rufzeichen als Digipeater enthalten ist.

OFF: kein Digipeating über dieses TNC möglich.

Disconnect

Beenden einer Verbindung. An die Gegenstation wird ein DM-Paket gesendet. Antwortet diese Station darauf, so ist die Verbindung beendet. Antwortet sie auf retry-malige DM-Packets nicht, so wird die Verbindung mit *** retry count exceeded abgebrochen. Wem das zu lange dauert, der kann auch sofort ein zweites Kommando 'D' eingeben, die Verbindung wird dann abgebrochen, ohne auf die Antwort der Gegenstation zu warten.

DISPlay [gruppe]

Anzeige aller Parameter, die man im TNC einstellen kann. DISP ohne weitere Parameter bringt alle gespeicherten Werte (ca. 80 Stück) auf den Bildschirm. Bei Angabe der Gruppe wird nur eine Auswahl an Werten gezeigt.

Async: Parameter der Asynchronen (Terminal-)Schnittstelle
Character: Spezielle Zeichen anzeigen (siehe ASCII-Zeichen)
Health: Statistikzähler anzeigen
Id: Identifikationsparameter (calls, Baken, CTexte etc.) anzeigen
Link: Betriebsartparameter anzeigen
Monitor: Monitorparameter anzeigen
Timing: Timerparameter anzeigen

DWait [n] n = 0...250 *10 ms Default: 33 = 330 ms

Wartezeit zwischen Ende der Kanalbelegung (DCD) und Tastung des Senders. Sollte bei allen Stationen auf 330 ms eingestellt sein, denn bei unterschiedlichen DWait-Zeiten sind Kollisionen zwischen den Packets wahrscheinlicher. DWait 0 darf nicht verwendet werden, da diese minimale Wartezeit den Digipeatern vorbehalten ist. Die Wiederaussendung von Packets hat Vorrang vor der ersten Aussendung neuer Datenpakete.

Echo [ON|OFF]

ON: Jedes Zeichen, das vom Terminal zum TNC gesendet wird, wird vom TNC 'geechot', damit es auf dem Bildschirm erscheint.

OFF: kein Echo. Das Terminal muß dann selbst die eingetippten Zeichen auf den Bildschirm bringen.

EScape [ON|OFF]

ON: Das Zeichen <ESC> ($1B) wird nicht zum Terminal geschickt, sondern als $-Zeichen angezeigt. Mit <ESC> kann man spezielle Funktionen bei den meisten Terminals auslösen, z.B. Seitenformate ändern etc. Damit dies nicht unbeabsichtigt geschieht, filtert man das <ESC> heraus.

OFF: <ESC> wird nicht ausgefiltert. (siehe auch MFilter)

FIRMRnr [ON|OFF]

Wenn der Empfangsbuffer des TNC voll ist, sendet das TNC ein RNR-Frame (RNR-Receiver Not Ready) aus. Sobald der Buffer des TNC nun wieder Platz für weitere Packets hat, reagiert das TNC folgendermaßen:

ON: Das TNC sendet RR (Receiver Ready), und fordert die Gegenstation aktiv auf, mit der Datenübertragung weiterzumachen

OFF: Das TNC reagiert nicht, sondern wartet, bis die Gegenstation die Daten noch mal aussendet.

Außerdem hat FIRMRnr Auswirkung beim Senden von Daten:

ON: Sobald das sendende TNC von der Gegenstation ein RNR empfängt, hört es auf, Daten zu senden bis die Gegenstation wieder ein RR schickt.

OFF: Sobald das sendende TNC von der Gegenstation ein RNR empfängt, wartet es eine Weile und versucht dann noch mal.

FRack [n] n = 0...15 s, Default = 8

Nachdem das TNC ein Datenpacket ausgesendet hat, wartet es eine Weile auf die Empfangsbestätigung der Gegenstation, dann sendet sie das Packet noch mal aus (maximal REtry-mal). Die Wartezeit bis zur nächsten Aussendung stellt man mit FRack ein. Die tatsächliche Wartezeit berechnet das TNC aus FRack, multipliziert mit der zweifachen Anzahl der Digipeater plus 1.

FUlldup [ON|OFF]

ON: Das TNC sendet und empfängt gleichzeitig. Nur möglich, wenn Sender und Empfänger unabhängig arbeiten (Relaisstationen, Digipeater).

OFF: normaler Betrieb, Senden nur wenn DCD aus.

HEALled [ON|OFF]

ON: Die CON und STA Leuchtdioden blinken ständig.

OFF: normale Funktion der CON und STA Leuchtdioden

HId [ON|OFF]

ON: Wenn das TNC digipeated, sendet es alle 9.5 Minuten ein Identifikationspacket aus. Dieses Packet ist an das UNProto-Call gerichtet, das Rufzeichen ist wie in MYcall definiert mit /R angehängt.

OFF: keine automatischen Identifikationspackets.

ID

Ein Identifikationspacket wird ausgesendet. Adresse ist wie in UNProto definiert, an das eigene Call wird ein /R angehängt.

IDTEXT text

Text für Identifikation oder CW-Identifikation

KIss [ON|OFF]

ON: Kissmode wird eingeschaltet. Kissmode wird aktiv, wenn man das TNC an die Stromversorgung anschließt oder das RESTART gibt. Im Kissmode muß fast die gesamte Verarbeitung der ankommenden und gesendeten Packets ein externer Rechner (z.B. IBM-kompatibler mit KA9Q oder Superkiss-Software) erledigen. Rückkehr durch Eingabe von PARAM <tnc> 255 oder hardware-Reset (siehe RESET-Kommando). Der Kissmode kann auch wieder verlassen werden, wenn man die Zeichen $C0, $FF, $C0 nacheinander sendet. (falls das Terminal das kann). Im Kissmode blinkt die CON= und STA LED dreimal auf, wenn das TNC eingeschaltet wird.

OFF: Nach RESTART bleibt das TNC im TAPR-mode (Normalbetrieb)

KISSM

Schaltet TNC sofort in den Kissmode.

LCAlls [call1 [,call2 [,call3 ... ,call8]]]

Es werden nur die Packets dieser Rufzeichen im Monitormode angezeigt, oder es werden alle Packets, außer denen angezeigt, die eines dieser Rufzeichen enthalten. (siehe BUDLIST ON/OFF). Mit % oder & als Rufzeichen läßt sich die Liste löschen.

LCok [ON|OFF]

ON: Das TNC sendet Groß- und Kleinbuchstaben zum Terminal

OFF: Das TNC übersetzt empfangene Kleinbuchstaben in Großbuchstaben. Das kann recht praktisch sein, wenn man die gesendeten Texte in Kleinbuchstaben tippt, und die ankommenden Packets nur mit Großbuchstaben angezeigt werden, kann man die verschiedenen Packets gut unterscheiden.

MAXframe [n] n=1...7 Default = 4 Packets

Das TNC kann mehrere Packets aussenden, ohne daß eine Empfangsbestätigung empfangen wird. Die maximale Anzahl der unbestätigten Packets wird mit MAXframe angegeben.

MCOM [ON|OFF]

ON: Alle Packets werden in Monitormode angezeigt, einschließlich der Packet-Type. (Connect-Pakete, Disconnect-Pakete, Receiver Ready, etc.)

OFF: nur Informationspakete (connected oder unproto) werden im Monitormode angezeigt.

MCon [ON|OFF]

ON: die Monitorfunktion (alles mitschreiben, was auf der Frequenz läuft) ist ständig eingeschaltet, auch wenn man connected ist. (stört meist)

OFF: Monitorfunktion stoppt, sobald man connected ist.

MFilter [n1 [,n2 [,n3 [,n4]]]]

Hier können bis zu 4 Zeichen angegeben werden, die in Monitormode *nicht* ans Terminal weitergegeben werden. Angabe der ASCII-Zeichen dezimal oder hexadezimal mit vorgestelltem $. z.B. kann man mit MF $07 das Klingelzeichen im Monitormode unterdrücken. Es können Zeichen zwischen 0 und 127 unterdrückt werden.

MHClear

Mit MHClear wird die MH-Liste (s.u.) gelöscht.

MHeard

Anzeige der MH-Liste. In dieser Liste sind die Rufzeichen der 18 zuletzt gehörten Stationen (ggf. mit Uhrzeit, Befehl DAYTIME) aufgelistet. Stationen, die nicht direkt, sondern über Digipeater gehört wurden, sind mit einem Sternchen markiert.

MNonax25 [ON|OFF]

ON: Alle Pakete mit gültiger Prüfsumme werden im Monitor angezeigt.

OFF: Nur Pakete mit Protokoll-ID F0 (AX.25 Level 2 Packets) werden im Monitormode angezeigt. Packets mit höheren Levels werden ignoriert, z.B. NET/ROM-Packets, TCP/IP-Packets etc. Diese Packets enthalten oft binäre Daten, die auf dem Terminalschirm oft zu unerwarteten Resultaten führen.

MNONPrint [ON|OFF] Default: off

ON: Alle Pakete werden im Monitor angezeigt.

OFF: Pakete, die nicht-druckbare Zeichen enthalten (die die Bildschirmanzeige durcheinanderbringen könnten) werden unterdrückt.

Monitor [ON|OFF]

ON: Monitormode eingeschaltet. Alle Packets, die empfangen werden, erscheinen auf dem Terminal-Bildschirm.

OFF: Nur Pakete, die an die eigene Station gerichtet sind (connected), werden angezeigt.

MStamp [ON|OFF]

ON: Die angezeigten Pakete im Monitormode enthalten die Uhrzeitangabe

OFF: Monitormode ohne Zeitstempel

MYAlias [call] Default:

Eingabe eines weiteren Rufzeichens, das speziell für die Digipeater-Funktion des TNC bestimmt ist.

MYcall [call] Default: NOCALL

Eingabe des eigenen Rufzeichens, maximal 6 Buchstaben + SSID-Kennung. (siehe 'SSID' weiter hinten)

NEwmode [ON|OFF]

ON: Automatisches Umschalten auf Konvers-mode bei Connect und auf Command bei Disconnect

OFF: kein Umschalten auf Command bei Disconnect

NOmode [ON|OFF]

ON: Kein automatisches Umschalten auf Konvers oder Command-mode.

OFF: Umschalten auf Konvers oder Command wie mit NEwmode definiert.

OUt [nn] n=0...255

Der Wert 'nn' wird auf der Z80-IO-Adresse $0BF ausgegeben. Der letzte Wert von 'nn' wird ständig auf dieser I/O Adresse ausgegeben (mindestens alle 100ms), jedoch nicht sofort nach Empfang des Kommandos.

Paclen [n] n = 0...255 Default = 128 Bytes

Paketlänge. Auch im Transparentmode wird immer dann ein Datenpaket abgeschickt, wenn die in Paclen angegebene Anzahl von Bytes zur Übertragung bereitstehen.

PACtime [Every|After] [n] n = 0...250 *100 ms, Default = AFTER 10

EVERY: Wenn CPActime =ON, dann sendet das TNC alle n Sekunden die in der Zwischenzeit eingetippten Zeichen aus, auch ohne Return.

AFTER: Wenn CPActime =ON, und wenn n Sekunden lang keine weiteren Zeichen eingegeben wurden, dann sendet das TNC die bisher eingetippten Zeichen aus.

PARity [n] n=0...3 Default: 3 (EVEN)

n bestimmt die Einstellung des Parity-Bits der seriellen (Terminal-) Schnittstelle: 0= kein Parity-Bit, 1= ODD (ungerade) Parität, 2= kein Parity-Bit, 3= EVEN (gerade) Parität. Die Parity-Einstellung sollte mit dem Terminal übereinstimmen, sonst ist eine Kommunikation nicht möglich. Die Parity-Einstellung wird erst nach RESTART oder Wiedereinschalten wirksam! Hat man eine falsche Einstellung vorgenommen, dann kann man u.U. das

TNC nicht mehr ansprechen. (siehe RESET-Befehl). Üblich sind die Einstellungen 8 Bit/keine Parität oder 7 Bit/gerade Parität.

PASs n (Default: $16 = Control-V)

Wie kann man z.B. einen mehrzeiligen CText eingeben? Sobald man Return drückt (um in die nächste Zeile zu kommen) wird der Text gespeichert, denn Return ist ja als Ende der Eingabe definiert, hat also eine Sonderfunktion. Lösung: man teilt dem TNC vor dem 'Return' mit, daß das folgende Zeichen keine Sonderfunktion hat, indem man vor dem Return ein ^V tippt. Auch wenn man ein Sonderzeichen senden möchte, auf das das TNC normalerweise unerwünscht reagiert (z.B. ^C), so 'überspringt' man das ^C durch das vorangestellte ^V. Sinnvoll auch, wenn man viele kurze Zeilen als ein Datenpaket senden möchte. Man unterdrückt dann die Aussendung jeder einzelnen Zeile nach Return durch vorheriges ^V. Das ^V wird natürlich nicht mit ausgesendet, es sei denn, ein ^V steht davor!

PASSAll [ON|OFF]

ON: Das TNC akzeptiert auch (gestörte) Packets mit falscher Prüfsumme. Die MH-Liste wird dann nicht weitergeführt, da sonst fehlerhafte Rufzeichen in der Liste auftauchen könnten.

OFF: Das TNC akzeptiert nur ungestörte Packets mit richtiger Prüfsumme

RECOnnect [Call [Via Call2 [Call3 [Call4 ...Call 9]]]]

Aufbau einer Verbindung zu Call, ggf. via die Digipeater Call2, Call3 etc. z.B. RECO DK2GE V DB0IE DB0EQ DB0ID, wenn bereits eine Verbindung zu dieser Station besteht, jedoch die Verbindung auf einem anderen Weg neu aufgebaut werden soll.

RESET

Löscht alle (!) Einstellungen des TNC unwiderruflich. Auch die Uhrzeit geht dabei verloren. Falls sich das TNC einmal überhaupt nicht mehr ansprechen läßt, weil z.B. AWlen verstellt oder KISS ON gesetzt wurde, dann kann man alle Parameter des TNC löschen, indem man 1. Stromversorgung aussteckt, 2. Schalter auf 'WA8DED' (unten) stellt, 3. Stromversorgung einsteckt und nach einigen Sekunden wieder aussteckt, 4. Schalter wieder hoch auf 'TAPR', 5. Stromversorgung wieder einstecken. TNC meldet sich dann mit 'bbRAM failed ...' und ist völlig in Grundstellung. (Diese RESET-Methode gilt natürlich gleichermaßen für TAPR und WA8DED-Betrieb)

RESTART

Hat den gleichen Effekt, wie Aus= und Einschalten des TNC. Alle Werte, die nicht im batteriegepufferten RAM gespeichert sind, werden gelöscht.

REtry [n] n = 0...15, Default = 10

Maximale Anzahl von Wiederholungen eines nicht bestätigten Packets. Bei n=0 wird das Packet unendlich oft wiederholt, das sollte man vermeiden. Wird ein Packet nach n Versuchen nicht bestätigt, erfolgt ein Disconnect.

ScreenIn [n] n = 0...255, Default: 0

Das TNC fügt nach n Zeichen <CR> <LF> in den empfangenen Text ein. Bei ScreenIn = 0 unterbleibt das Einfügen dieser Zeichen.

SEndpac (Default: $0D = Return = Control-M)

Immer wenn Return getippt wird, wird die angefangene Zeile als Datenpaket abgeschickt. Mit SEndpac kann man auch andere Zeichen als Sendebefehl definieren.

SLots [n] n = 0...127, Default: 3

Slot bestimmt die Anzahl der Zeitschlitze, innerhalb derer die Packets gesendet werden. Setzt man beispielsweise SLot auf 3, dann existieren drei Zeitschlitze, von denen jeder mit einer Wahrscheinlichkeit von 1/3 ausgewählt wird. Jeder Zeitschlitz ist DEADtime lang.

Eine ähnliche Methode der Zugriffsteuerung ist die Persistence-Methode. Der Zusammenhang zwischen Persistence und Slot/Deadtime-Methode ist folgendermaßen:

Slots	PPersistence	Persistence	Sende-Wahrscheinlichkeit
1	OFF	255	100%
2	ON	127	50%
3	ON	85	33%
4	ON	63	25%
5	ON	51	20%

STArt (Default: $11 = Control-Q)

Eine mit ^S angehaltene Ausgabe kann man mit ^Q weiterlaufen lassen. Die Steuerung der Ausgabe mit Xon und Xoff nennt man Software-Handshake, manche Rechner oder Terminals senden diese Zeichen automatisch, wenn sie mit der Verarbeitung der ankommenden Daten nicht nachkommen. Setzt man STArt und STOp auf $00, dann läßt sich die Ausgabe durch keine Zeichen mehr anhalten, der Software-Handshake ist dann ausgeschaltet.

STATUS

Zeigt an, ob bzw. wie viele Packets noch nicht vom Empfänger bestätigt sind. Wenn STA-LED nicht leuchtet, kommt 'No Outstanding Packets'.

STOp (Default: $13 = Control-S)

Wenn ^Q getippt wird, hört die Ausgabe von Daten vom TNC zum Terminal sofort auf. Nützlich, wenn die Texte schneller ankommen, als man sie lesen kann. Mit ^Q schaltet man die Ausgabe wieder ein.

STREAMCa [ON|OFF]

ON: Bei Multiconnect wird (außer dem Streamswitch-Zeichen und der Kanalkennzeichnung) das Rufzeichen der Station angezeigt, mit der man auf diesem Kanal connected ist.

OFF: Nur der Streamswitch + Kanal wird angezeigt.

STReamswitch (Default: $7C = | (senkrechter Strich))

Mit diesem Sonderzeichen kann man die 10 Kanäle des TNC umschalten. (siehe Multiconnect) Auf Kanal c schaltet man z.B., indem man |c tippt. Der Kanal der eingehenden Packets wird ebenfalls mit dem Streamswitch-Zeichen und der Kanalnummer gekennzeichnet. Bei manchen Rechnern wird $7C nicht als senkrechter Strich, sondern als kleiner Umlaut ö angezeigt.

TRACe [ON|OFF]

ON: Die empfangenen Packets werden vollständig, das heißt einschließlich der normalerweise unterdrückten Headerinformation angezeigt. Da insbesondere die Header diverse Control-Zeichen enthalten, wird das Packet in Hexadezimaler Schreibweise ausgegeben, für je 16 Zeichen des Packets erscheint eine 80 Zeichen breite Zeile auf dem Schirm. Die Rufzeichen in einem Packet werden um 1 Bit verschoben übertragen, bei TRACe ist für diese Zeichen die Verschiebung in der Spalte 'shifted ASCII' rückgängig gemacht, damit die Calls im Klartext sichtbar werden.

OFF: Normaler Betrieb

Transpar

Schaltet (ähnlich wir K) um in den Transparentmodus. Näheres hierüber siehe TAPR 1.1.8 commands booklet. Zurückschalten: BREAK senden oder nach kurzer Wartezeit dreimal ^C rasch nacheinander tippen.

TRFlow [ON|OFF]

ON: Der Software-Handshake für den Computer kann im Transparent-Mode benutzt werden.

OFF: Nur der Hardware-Handshake kann im Transparent-Mode benutzt werden.

TRIes [n] n = 0...15

TRIes zeigt an, wie oft das letzte Packet schon nacheinander ausgesendet wurde (ohne Empfangsbestätigung). Man kann, indem man n angibt, diese Zahl auf einen bestimmten Wert setzen, z.B. wieder auf 1. Dann wird das Packet noch ein paarmal öfter wiederholt.

TXdelay [n] n = 0...120 *10 ms Default: 33 = 330 ms

Wartezeit vom Hochtasten des Senders (PTT-LED ein) bis zum Beginn der Aussendung von Daten. Möglichst kurze Zeit einstellen, der optimale Wert hängt vom eigenen Funkgerät und dem der Gegenstation ab.

TXDELAYC [n] n = 0...120 *10 ms Default: 2 Character

Wartezeit vom Hochtasten des Senders (PTT-LED ein) bis zum Beginn der Aussendung. Im Gegensatz zu TXdelay wird die Verzögerung in Vielfachen der Übertragungsdauer für ein Zeichen angegeben. Damit erhöht sich die Verzögerung bei niedrigerer Baudrate automatisch.

TXDIddle [ON|OFF]

ON: Während TX-Delay sendet das TNC NRZI 0´s.

OFF: Während TX-Delay sendet das TNC $7E Flags .

TXFlow [ON|OFF]

ON: Der Software-Handshake für as TNC kann im Transparent-Mode benutzt werden.

OFF: Nur der Hardware-Handshake kann im Transparent-Mode benutzt werden.

Unproto [call [Via call2 [call3 ... call9]]] Default: CQ

Eingabe des 'Unproto'-Rufzeichens. Wenn man in den Konvers-mode schaltet und NICHT connected ist, dann sendet man UI-Packets, das sind Datenpackets, bei denen man keine Bestätigung erwartet. Zum Test eines Digipeaters kann man z.B. MONitor ON schalten und UNPROTO TEST V DB0XY setzen. Im Konvers-mode sieht man dann, wie alle eigenen Packets vom Digipeater DB0XY wiederausgesendet werden.

USers [n] n = 0...10 Default: 1

Bestimmt die maximale Anzahl der Kanäle bei Multiconnect. Selbst veranlaßte Verbindungen sind immer auf allen 10 Kanälen möglich.

Xflow [ON|OFF]

ON: Der Datenverkehr von TNC zum Terminal (oder Computer) wird durch die mit XON und XOFF definierten Software-Handshake-Zeichen gesteuert.

OFF: XON/XOFF Handshake ist abgeschaltet, Hardware Handshake ist eingeschaltet (RTS-Leitung)

XMitok [ON|OFF]

ON: Sendertastung ist möglich / OFF: Sender ist grundsätzlich abgeschaltet

XOff (Default: $13 = Control-S)

Wenn das TNC ^S ausgibt, dann kann es gerade keine weiteren Daten annehmen und der Rechner (bzw. das Terminal) sollte aufhören, weitere Daten zum TNC zu schicken. Sobald das TNC wieder Platz für weitere Daten hat, wird es ^Q senden, man kann dann weiter Daten schicken.

XON (Default: $11 = Control-Q)

Hat das TNC den Rechner mit ^S angewiesen, jetzt keine weiteren Daten mehr zu schicken, dann wird es, sobald möglich, den Rechner mit ^Q auffordern, den angehaltenen Datenstrom weiterlaufen zu lassen. Man kann XON und XOff auf $07 (bell, Klingelzeichen) setzen. Wenn man dann mit dem Terminal schneller Daten eingibt, als sie das TNC wieder los wird, dann sendet das TNC das XOFF-Zeichen : Das Terminal klingelt. Sobald das TNC Platz für weitere Daten hat, sendet es wieder das XON-Zeichen (ebenfalls Klingel), dann kann man weitertippen.

Zähler für Statistik

Im TNC werden verschiedene Ereignisse registriert und gezählt. Durch die Abfrage der Zähler läßt sich feststellen, wie häufig diese Ereignisse eingetreten sind. Der Zählbereich geht von 0 bis 65535 (16 Bit), dann springt der Zähler wieder auf 0. Einige Zähler registrieren Fehler, die nicht auftreten dürften, manche Zähler geben Auskunft über die Qualität der empfangenen Signale. Die übrigen Zähler dienen statistischen Zwecken und geben Auskunft über die Menge der gesendeten und empfangenen Packets. Folgende Zähler sind verfügbar: (genaue Beschreibung im original TAPR Handbuch)

ASYFRERR, ASYQOVER, ASYRXOVR, BBFAILED, DIGISENT, HOVRERR, HUNDRERR, RCVDFRMR, RCVDIFRA, RCVDREJ, RCVDRNR, RCVDSABM, RXABORT, RXCOUNT, RXERRORS, RXLENERR, RXRESYNC, SENTFRMR, SENTIFRA, SENTREJ, SENTRNR, TXCOUNT, TXQOVFLW, TXTMO.

Spezialkommandos für GPS

Die TAPR-Software 1.1.9 enthält Kommandos, die für die Positionsdaten-Übertragung eines ans TNC angeschlossenen GPS-Empfängers vorgesehen sind. Bestimmte, programmierbare $GPxxx-Zeilen des GPS

werden im TNC ausgefiltert und in einstellbaren Zeitintervallen ausgesendet. Für Betrieb als NMEA-Bake muss CONMODE auf Converse eingestellt sein. SENDPAC wird auf $0D und CR auf ON gestellt. Für die üblichen GPS-Empfänger sollte die Baudrate auf 4800 Baud, das Wortformat (AWLEN) auf 8 Bit und Parity (PAR) aof No Parity (0) gestellt werden. Da Funkdatenübertragungen nicht 100% zuverlässig sind darf das TNC nicht für Navigationszwecke eingesetzt werden.

NMEABCN (0...254 in 10 s Intervallen = 10...2540s (42 min) Default: 0 = NMEA Bake aus)

Das TNC sendet alle NMEABCN * 10 Sekunden die zuletzt über RS232 empfangenen NMEA Datensätze. Wurde seit dem letzten Einschalten des TNC noch kein gültiger Datensatz empfangen, so sendet das TNC keine Bake. Die Datensätze werden im Unproto-Mode an die mit UNPROTO definierte Adresse gesendet. Für normale TNC-Funktion muss NMEABCN=0 gesetzt sein.

NMEAFLT1 xxxxx (Default: leer)

NMEAFLT1 yyyyy (Default: leer) sssss= NMEA-Kennung

Aus den NMEA-Datensätzen werden diejenigen ausgefiltert, die mit $xxxxx oder $yyyyy beginnen. Sollen z.B. die Zeilen mit $GPRMC oder $GPZDA übertragen werden, so müssen die Filter mit NMEAFLT1 GPRMC und NMEAFLT2 GPZDA definiert werden. Das in allen NMEA-Sätzen am Zeilenbeginn stehende $ Zeichen wird nicht angegeben.

Anleitung zur Benutzung des WA8DED Host Mode

Paul T. Williamson, KB5MU
Übersetzung/Kommentare DJ8RI 23 Mai 1986

Inhaltsangabe:

Dieses Dokument erläutert die Nutzung einer sog. Host Mode Software bei Verwendung der Version 2 Multikanal TNC Firmware für das AX.25 Protokoll, bzw. in der Version 1.1 für den TAPR TNC-1. Die Informationen basieren auf einer Dokumentation von WA8DED (Version 1.0), auf den Informationen durch die Mailbox-Software (Ausgabe vom 13 Mai 86) von W6IXU und dem Terminal Emulator in sog. Split-Screen Technik von WA8DED in der Ausgabe SS vom 28 April 86. Diese Informationen wenden sich an einen Anwendungsprogrammierer der sein Programm in diesem Host Mode mit dem TNC kommunizieren lassen will; es wird vorrausgesetzt, dass er mit dem sog. Benutzer Mode der WA8DED Firmware etwas vertraut ist.

Was ist Host Mode ?

Die Firmware von WA8DED stellt zwei gänzlich unabhängige Benutzer-Oberflächen zur Verfügung. Die voreingestellte Oberfläche ist die zur Benutzung für eine Person am Terminal (Benutzer Mode). Auf der anderen Seite ist der Host-Mode gedacht durch ein spezielles Computer Programm bedient zu werden. Dieser Mode erlaubt dem Computer die eindeutige Kontrolle der Verbindung Computer/TNC und liefert ihm die nötigen Kontrollinformationen, die sonst nicht ohne Schwierigkeiten aus dem Datenstrom entnommen werden könnten.

Die verbreiteste Fehlerquelle im Zusammenspiel Computer/TNC ist üblicherweise die Kontrolle des Datenstroms auf der Verbindung zwischen diesen. Einerseits ist der TNC in der Lage Daten schneller zu senden als der Computer sie verarbeiten kann, andererseits kann der Computer Daten schneller senden als der TNC sie ggf.absetzen kann. Theoretisch ist es möglich die Kontrolle des Datenstroms durch die Signale RTS/CTS als zusätzliche Leitungen in der RS-232 Verbindung zu kontollieren oder aber durch Einfügen der Zeichen XON/XOFF in den Datenstrom. Beide Methoden sind kompliziert und fehleranfällig. Die Methode mit den zusätlichen Signalen (Hardware) wird nicht immer zur Verfügung stehen und bei der Softwarelösung, auch wenn sie genau so gut arbeitet, gibt es Probleme mit der Transparenz der Verbindung. Denn, was passiert,wenn in den Daten ein entsprechendes Zeichen XON/XOFF enthalten ist? Ist etwas nicht in Ordnung dann gehen Daten verloren.

Host Mode von WA8DED behebt dieses Problem dadurch dass:

1. es wird jeweils nur auf Anfrage gesendet
2. die maximale Datenmenge ist limitiert auf 256 Bytes/ Burst

Der erste Punkt erlaubt dem Computerprogramm zwischenzeitlich was anderes zu tun, z.B Daten auf eine Platte zu schreiben,den Bildschirm und Tastatur zu bedienen oder auch Kaffee zu kochen. Dabei besteht nicht die Gefahr,dass Daten vom TNC kommen und verloren gehen.

Der zweite Punkt erlaubt den Computern,die sich mit schnellen seriellen Daten etwas schwer tun (Commodore 64 ohne UART) sich auf das Füllen eines begrenzten Puffers zu konzentrieren, ohne ,dass erneut Daten

abgenommen werden müssen, obwohl die letzten noch nicht verarbeitet sind. Dieses Schema ist sehr grosszügig gegenüber uneffizienten d.h langsamen Routinen für die seriellen Ports. Ist der Computer ein wenig schneller,so reicht die wiederholte Abfrage auf Byte Ebene, d.h. man kann bei einem IBM PC die Routinen verwenden,die für das BIOS zur Verfügung stehen .

Noch ein Wort zu den Routinen zur Bedienung der seriellen Ports: der TNC hat jetzt keinerlei Kontrollfunktionen um den Datenfluss zu steuern, es muss sichergestellt sein, dass der Treiber auf der Computerseite ebenso keinerlei unerwünschte XON/XOFF sendet. Da der TNC nur etwas sendet, falls ihm das vorher gesagt wurde, ist es dann notwendig, dass regelmässing der TNC abgefragt werden muss (polling). Im sog. Benutzer-Mode würde, bei angewähltem Kanal, über das AX.25 Protokoll hereinkommende Daten sofort angezeigt. Im Host-Mode ist es dem TNC nicht erlaubt ohne Anweisung zu senden, das Programm muss also regelmässig die Daten abfragen. Das hat zur Wirkung,dass die Verbindung Computer/TNC dauernd in Betrieb ist. Häufiges Abfragen mit relativ hoher Baudrate (4800 oder 9600 Baud) sind notwendig, falls schnelle Antwortzeiten erwünscht sind. Ein Benutzer,der den Betrieb beobachtet, wird Verzögerungen feststellen, falls die Zeiten zwischen den Abfragen zu gross werden. Das Format, das eingehalten werden muss, wenn zwischen TNC und Computer Informationen ausgetauscht werden sollen muss so gewählt sein, dass:

- keine Zweideutigkeit existiert,
- aber auch nicht zuviel Verwaltungsaufwand getrieben werden muss.

Folgendes Format wird an den TNC gesendet:

```
{channel}{info/cmd}{count}{data...}
```

channel ist die *Kanalnummer* zu der die Sendung gehört.
info/cmd unterscheidet ob die übergebenen Daten *Informationen* sind ,die über AX.25 abgstrahlt werden sollen oder als *Befehle* (command) für den TNC gelten.
count spezifiziert die *Anzahl* (count) der folgenden Bytes
data sind die aktuellen *Daten* oder auch entsprechende Bytes des Befehls.

Jede Datenübergabe vom Computer an den TNC muss in dieser Form erfolgen.

Der TNC sendet an den Computer in einem der möglichen unterschiedlichen Formate:

{channel}{code=0} short format (Kurzformat)

oder

{channel}{code=1-5}{data...}{0} null-terminated format (mit binär Null beendet)

oder

{channel}{code=6-7}{count}{data...} byte-count format (Anzahl der Bytes)

Anhand der Codes lässt sich eindeutig sagen,was dann noch zu folgen hat. Das letzte Format ist ähnlich dem Format bei der Datenübergabe Computer zum TNC.Auf die Einzelheiten dieser Formate soll später eingegangen werden.Jede Datenübergabe vom TNC zum Computer muss einem der vorstehenden Formate genügen.

Einstieg in den Host-Mode und Verlassen des Host-Mode

Der Einstieg in den Host-Mode erfolgt durch Senden des Kommandos JHOST 1 an den TNC in der üblichen Weise. Das klingt recht einfach meint hier: die Zeichenfolge <ESC>JHOST1<CR> wird zum TNC gesendet. Was aber, wenn aus irgend einem Grund (z.B. beim Einschalten) schon ein vermeintliches Zeichen erhalten wurde? Dann wird das Zeichen <ESC> als Date gewertet und die Echofunktion liefert ^[zurück. Man kann sich eine Weile mit diesen Möglichkeiten beschäftigen: man sendet erst ^U oder ^X um den Unsinn los zu werden, oder es kann auch eine gute Idee sein XON zu senden, denn es kann ja auch sein, dass XOFF empfangen wurde. Also senden wir:

```
^Q^X^[JHOST1^M
```

oder , wenn sie ASCII-Schreibweise vorziehen:

```
<DC1><CAN><ESC>JHOST1<CR>
```

das ist:

```
11 18 1B 4A 48 4F 53 54 31 0D

^Q ^X ^[ J  H  O  S  T  1  CR
```

Dieser Befehl bringt den TNC immer in den Host-Mode, falls er im User-Mode war. Ist der TNC aber schon im Host-Mode sollte dieses vorher abgefragt werden und wieder verbunden werden ohne den Host-Mode zu verlassen. Wie dieses geschieht wird später in diesem Papier behandelt.

Um den Host-Mode zu verlassen gibt man den Befehl JHOST0 an den TNC. Dieser bestätigt in der für den Host-Mode üblichen Weise und erst danach erfolgt das Verlassen in den Benutzer-Mode (User-Mode). Es kann also mit der für den Host-Mode normalen Routine gearbeitet werden.

Es muss beachtet werden, dass der Befehl QRES den TNC auch in den User-Mode bringt, aber keine Bestätigung zuvor noch im Host-Mode geschieht. Die Parameter des Host-Mode werden nicht im NOVRAM abgelegt, daher kann nach Einschalten nicht automatisch der Host-Mode erreicht werden. Es lassen sich allerdings andere Parameter durch den Befehl PERM (hier TAPR TNC 1, gilt nicht für den TNC2 und Clone) im nicht flüchtigen Speicher ablegen.

Befehle an den TNC

Nehmen wir also an wir wären im Host-Mode und wollte uns ansehen, wie ein einfacher Befehl übermittelt wird. Dabei gehen wir bei dieser Übersicht nicht ganz ins Detail. Angenommen wir wollten den sog. Unattended-Mode ausschalten. Dann würden wir im User-Mode schreiben : <ESC>U0<CR> d.h. wir senden den Befehl U0 an den TNC. Um das im Host-Mode zu machen, muss eine Zeichenfolge generiert werden, die dem Format *Computer zu TNC* enspricht.

Erinnern wir uns an das Format:

```
{channel}{info/cmd}{count}{data...}
```

Das erste Byte ist die Kanalnummer. Im User-Mode ist es gleichgültig welcher Kanal angewählt ist, wenn sie den Befehl U geben wollen. So ist es auch hier im Host-Mode. Aus rein formalen Gründen wählen wir einen, sagen wir den Kanal 0.

Das zweite Byte bringt die Unterscheidung zwischen Daten und Befehlen. Hier ist es also ein Befehl, folglich ist das Byte eine 1. Der eigentliche Befehl ist U0 , also gleich dem, was im User-Mode zwischen <ESC> und <CR> steht. Das sind zwei Bytes. Da die Aufarbeitung für die *Anzahl* (count) aber sagt: Anzahl=Länge-1 bleibt hier also count = 1.

Die Darstellung der Zeichenkette :

die obere Reihe der Zahlen ist die hexadezimale Darstellung der einzelnen Bytes. Das Byte, was links steht wird als erstes gesendet. Unterhalb der oberen Zeile erfolgt eine Erklärung zu dem Bytes.Dies Schreibweise wird in den folgenden Kapiteln beibehalten.

```
00 01 01 55 30
!  !  !  U  0
!  !  !
!  !  +-- Anzahl   = 1   (zwei Bytes)
!  +----- Info/cmd = 1   (hier ist es ein Befehl)
+-------- Kanal    = 0
```

Achtung ! es ist weder ein <CR> notwendig oder erlaubt!. Sobald entsprechend der *Anzahl* (count) alle Bytes gelesen wurden, (das sollte dann sein, wenn alle Bytes übertragen sind) wertet der TNC die Übertragung als komplett.

Woher wissen wir jetzt,dass der TNC den Befehl richtig erhalten hat? Ganz einfach:der schickt eine Statusmeldung zurück. Der Befehl U0 kann nicht fehlerhaft sein, vermutlich wird der Statusbericht also etwas sein, wie 'mit Erfolg durchgeführt'.

Der Befehl U0 veranlasst auch nicht, dass der TNC irgend welche andere Informationen zurücksendet, so ist dann anzunehmen,dass der TNC in der sog. Kurzform (short format) antwortet:

```
00 00
!  !
!  +--- Code = 0 (Erfolg,keine Daten folgen)
+------ Kanal 0
```

Wir sehen also hier die typische Weise des Zusammenspiels zwischen Computer und TNC : der Computer sendet einen Befehl an den TNC und der TNC schickt eine entsprechende Antwort. Wollen wir also ein Programm für diesen Host-Mode schreiben, so muss dieses in der Lage sein, die rausgehenden Befehle zu generieren und die hereinkommenden Daten zu interpretieren.

Falls wir uns an die Hinweise in den Programmen von WA8DED und W6IXU halten wollen, dann schreiben wir eine einzige Routine, die die grobe Arbeit macht: Sie sendet an den TNC und empfängt die Antwort.

Wird eine Antwort empfangen, dann muss diese Routine aus dem zweiten Byte der Antwort vom TNC feststellen, wieviele Zeichen vom seriellen Port zu lesen sind. Dieser Code sagt in welchem Format die Antwort ist: im Kurzformat (short format), im Format mit dem Wert 0 als Date am Ende (null-terminated format), oder im Format mit der Angabe der Anzahl der folgenden Bytes,die als Information folgen (byte-count format).

Man beachte, dass die Routine,die die Antwort des TNC empfängt, nichts über den momentanen Betriebszustand des TNC oder über ein laufendes Anwenderprogramm wissen muss. Die Aussage in der Antwort des TNC ist eindeutig durch die die gesendete Statusmeldung. Diese Tatsache vereinfacht die Routine, da nichts aus einer möglichen Komplexität eines Anwenderprogrammes zu beachten ist.

Zyklisches Abfragen des TNC: der Befehl G

Der wesentlichste Befehl für das Zusammenspiel Computer/TNC im Host-Mode ist der Befehl G, mit dem die zyklische Abfrage eines bestimmten Kanals erfolgt, um von diesem Ereignisse und Daten abzuholen. Ein typisches Programm für den Host-Mode wird fortwährend den Befehl G an den TNC senden und die Antwort interpretieren. Da der Befehl G eine Frage an einen bestimmten Kanal des TNC ist, muss er an alle Kanäle gesendet werden,die im Gebrauch sind. Daten, die von einer Station stammen, die mit Kanal 1 (oder auch 2,oder 3,oder 4) verbunden ist, werden nur bei Abfrage des betreffenden Kanals an den Computer weitergegeben.

Daten aus der Monitor-Funktion des TNC werden dem Kanal 0 zugeordnet und durch Abfrage des Kanal 0 an den Computer gesendet.

Der Befehl G kennt drei Formate. Der Befehl mit nur dem einfachen G fragt alle erdenklichen Antworten des TNC ab, der Befehl G0 schränkt die Antwort ein auf die Sendung von Daten des entsprechenden Kanals, und die Schreibweise G1 liefert nur die Statusangaben zurück. Diese Unterscheidung ist nützlich, wenn das Anwenderprogramm gerade die Änderung des Status eines Kanals überwacht und keine Daten verarbeiten kann und natürlich auch umgekehrt.

Der sich ändernde Parameter in einem Befehl G ist die Kanalnummer. Es muss sichergestellt sein, dass alle aktiven Kanäle auch abgefragt werden. Falls der Wert z.B. für Y=3 ist (also die maximale Anzahl der logischen Kanäle - oder die maximale Anzahl der gleichzeitigen Verbindungen ist drei) müssen die Kanäle 0 bis 3 abgefragt werden, nicht aber der Kanal 4.

Der uneingeschränkte Befehl G hat folgendes Format:

```
0x 01 00 47
!  !  !  G
!  !  !
!  !  +--- Anzahl   = 0    (ein Byte Daten)
!  +------ Info/cmd = 1    (ein Befehl)
+--------- Kanal    = x    (x ist 0, 1, 2, 3, oder 4)
```

Es gibt nur wenige mögliche Antworten des TNC auf einen Befehl G. Fragt der Computer den TNC zyklisch ab, dann werden die meisten Antworten sein: nichts vorhanden, d. h. es hat sich sei der letzten Abfrage nichts geändert. Das gilt für alle drei Arten des Befehls G. Ändert sich etwas, wird durch G oder G1 z.B. ein 'disconnected' auf der AX.25 Seite erfasst, oder durch ein G oder G0 ein Daten-Frame des Monitor-Mode oder im 'connected'-Mode gelesen. Zu den Einzelheiten des Formats siehe weiter unten.

Die Daten aus dem Monitor-Mode werden auf spezielle Weise behandelt. Die Kopfzeile wird getrennt vom eigentlichen Dateninhalt übertragen. Diese Kopfzeile enthält auch, ob überhaupt jetzt Monitor-Daten folgen. Falls es welche zu der Kopfzeile gibt,sind diese auf Kanal 0 als nächstes vorhanden.

Dieses Verfahren hat zwei Wirkungen: es erspart dem Programmierer eine Routine, die, falls es gewünscht wird, das Trennen von Kopfzeile und Dateninhalt bewerkstelligt und es stellt sicher, dass die Addition der Bytes der Kopfzeile und der maximal möglichen Informationslänge nicht die Buffergrösse übersteigt.

Dieses Schema für den Monitor-Mode mag als Verletzung der Regel angesehen werden, dass jeder Datenaustausch zwischen Computer und TNC für den einzelnen Vorgang als in sich abgeschlossen oder als sich selbst erklärend sein soll. Die Daten aus dem Monitor-Mode sind jedoch eindeutig bestimmt durch die unmittelbar vorher gesendeten Kopfzeile. Vielfach wird es vorkommen, dass die Kopfzeile aus dem Monitor-

Mode als Spezialfall in einem Anwenderprogramm behandelt wird und dann auf die zugehörige Information schnell zugegriffen werden kann.

Einzelheiten zum Format Computer zu TNC

Wie schon oben gesagt,erfolgt der Datenaustausch Computer zu TNC immer in dem gleichen Format: {Kanal}{Info/Befehl}{Anzahl}{Datenfeld...}

Die drei Bytes am Anfang (also Kanal,Info/Befehl und Anzahl) sind als binäre Werte eingetragen. Als Beispiel soll für alle drei Werte 0 angenommen werden (also ^^U,'^Y0' in C oder CHR$(0) in Basic). Hier ist nicht das Zeichen für den Wert 0 im der ASCII Schreibweise gemeint !

In dem Datenfeld kann folgendes enthalten sein:

1. Daten die über die AX.25 Verbindung gesendet werden sollen.
2. Eine Zeichenkette als Befehl d.h. exakt das,was im Benutzer-Mode zwischen den Zeichen <ESC> und <CR> steht.
3. Der Befehl G der zur zyklischen Abfrage benutzt wird.

Falls die Daten über die AX.25 gesendet werden sollen,muss das zweite Byte (Info/Befehl) den Wert binär Null haben. (nicht ASCII 0). Im Datenfeld können Bytes mit jedem Wert stehen,aber nicht mehr als 256. Weiss man wieviele man senden will, dann ziehe man Eins ab und erhält die Anzahl (count). Ist jemand mit mir auf dem Kanal 2 verbunden und ich will ihm ein 'Hallo' senden, dann sieht das wie folgt aus:

```
02 00 05 48 65 6C 6C 6F 0D
!  !  !  H  e  l  l  o  CR
!  !  !                 !
!  !  !                 +---- (dies ist ein Datenbyte!)
!  !  +- Anzahl      = 5    (sechs Bytes als Daten)
!  +---- Info/Befehl = 0    ( das sind Daten )
+------- Kanal       = 2    ( angenommen auf Kanal 2)
```

Achtung:hier ist das <CR> ein Datenbyte und wird auch durch die AX.25 Verbindung übertragen; das <CR> ist nicht Teil des Formates.

Würde diese Information zum Kanal 0 übertragen, bekäme sie die Adresse aus dem Kanal 0 (voreingestellt ist CQ) und würde im 'unproto '-Mode gesendet. Sendet man die Information zu einem Kanal,der nicht 'connected' ist, so geht die Information verloren.

Falls ein Befehl gesendet werden soll muss das Byte zur Unterscheidung Info/Befehl eine binäre 1 sein. Im Datenfeld steht dann der nackte Befehl, also kein <ESC> oder <CR>, nur das,was im User-Mode zwischen den beiden Zeichen steht. Setzen wir z.B. TXDELAY auf 30 so sieht das wie folgt aus:

```
00 01 02 54 33 30
!  !  !  T  3  0
!  !  !
!  !  +  Anzahl      = 2  (drei Datenbytes)
!  +---- Info/Befehl = 1  (ein Befehl)
+------- Kanal       = 0  (belanglos)
```

Die Kanalnummer ist signifikant für die Befehle vom Typ C, D, F, G, L, N, O, und V. Dabei sind die Befehle C, D, F, N, O, und V in gleicher Weise kanalspezifisch wie die Befehle im User-Mode.

Da der Befehl G vorstehend schon diskutiert wurde,bleibt der Befehl L ,der tatsächlich anders im Host-Mode ist. Er wird weiter unten behandelt. Für die anderen Kanäle ist die Kanalnummer nicht signifikant. Eigentlich kann was immer passt genommen werden, WA8DED und W6IXU senden die fraglichen Befehle an Kanal 0.

Das sog. 'byte-count '-Format (also Angabe der Anzahl der folgenden Bytes) wird verwendet um Datentransparenz zu erhalten. Das meint, dass jedes Datenbyte jeden Wert annehmen kann,egal, ob es letztlich als ASCII - Zeichen interpretiert wird oder einen Binärwert darstellt etc. Der TNC akzeptiert eine bestimmte Anzahl von folgenden Bytes und kümmert sich nicht darum, wie das einzelne Byte aussieht. Würde ein spezielles Zeichen z. B. als Ende-Kennung genommen, dann würde dieses Zeichen im Datenstrom nicht erlaubt sein. Was ist aber mit dem Format, was die Null als Abbruchkriterium verwendet? (null-terminated format)

Dieses Format kann folglich nur dann Verwendung finden,falls in der zu sendenden Zeichenkette garantiert keine Null vorkommen kann. Welche Zeichen aber vorkommen können ist ist durch diese Software festgelegt.

Unglücklicher Weise ist das 'byte-count'-Format gegenüber Störungen, die Zeichen einfügen oder bei denen Zeichen verloren gehen (verstümmelt werden) nicht sehr robust (Fehler, die im Bereich der Verbindung TNC/Rechner passieren!). Gerät man 'ausser Tritt' dann bleibt man es auch. Es bedarf also einigen Aufwands um den Verlust der Synchronität festzustellen bzw. die Synchronität zurück zu gewinnen. Nun hat das Format aber auch glücklicher Weise Eigenarten, die dabei nützlich sind. Das erste Byte kann nur fünf gültige Werte annehmen, das zweit Byte nur zwei ;die Prüfung hierauf lässt schon in den meisten Fällen erkennen ob man die Synchronität verloren hat. ebenso können zeitliche Verzögerungen Aussagen zur Nicht-Synchronität ergeben. Nur wenn ein Fehler erkannt wird kann man ihn beheben.

Weitere Diskussion zur Rückgewinnung der Synchronität im Host-Mode erfolgt noch später in diesem Text.

Einzelheiten zum Format TNC zu Computer

Das Format zur Datenübergabe vom TNC zum Computer ist ein wenig komplizierter als für den umgekehrten Weg. Der Schlüssel zur Dekodierung der Nachricht vom TNC liegt im sog. 'code byte', dem Byte, das dem 'Kanalnummer-Byte' folgt. Nachstehende Tabelle zeigt die Werte des Bytes und die Bedeutung.

TNC zu Computer Codes

Code	Bedeutung	Format
0	Erfolg,es erfolgt nichts weiter..nichts vorhanden	Kurzformat (short format)
1	Erfolg, Nachricht folgt	Abbruch durch Null (null-terminated)
2	Fehler, Nachricht folgt	Abbruch durch Null
3	Status der Verbindungen	Abbruch durch Null
4	Kopfzeile Monitor Anzeige keine Daten	Abbruch durch Null
5	Kopfzeile Monitor Anzeige es sind Daten da	Abbruch durch Null
6	Daten des Monitor-Mode	'byte-count' Format
7	Daten aus AX.25 Verbindung	'byte-count' Format

Diese unterschiedlichen Formate werden für unterschiedliche Anwendungsfälle verwendet. Die Kodierungen 0, 1 und 2 sind Antworten auf Informationen oder Befehle die vom Computer übertragen wurden. Wird ein Befehl oder eine Reihe von Befehlen übertragen, die letztlich nicht zu einer Rückantwort mit Daten führen, dann wird die Kodierung 0 verwendet:

```
02 00
!  !
!  +---- Code  = 0   (Erfolg,es folgt nichts weiter)
+------- Kanal = 2   (gleich dem,für den Info oder Befehl war)
```

Ist der Befehl erfolgreich übertragen und erfolgt eine Antwort mit zugehöriger Information (bei jedem Befehl ohne ein Argument ist das der Fall, d.h. eigentlich ist das eine Abfrage) wird die Kodierung 1 verwendet. Die vom TNC übertragenen Informationen haben als Ende- oder Abbruchkennung das letzte Byte mit dem Wert 0. Sehen wir uns den Befehl M an, der Abfrage zum Monitor Mode gehörend, wie das aussieht.Im User-Mode würde geschrieben:

```
<ESC>M<CR>
```

und je nach Terminal-Programm würde etwas ähnliches wie:

```
* IUSC *
```

zurückgeliefert. Im Host-Mode heisst das dann:

```
00 01 00 4D
!  !  !  M
!  !  !
!  !  +---- Count       = 0    (ein Datenbyte)
!  +------- Info/Befehl = 1    (dieses ist ein Befehl)
+---------- Kanal       = 0    (hier belanglos)
```

und die Antwort könnte sein:

```
00 01 49 55 53 43 00
!  !  I  U  S  C  !
!  !              +---- Null als Ende-Kennung
!  !
!  +---- Code   = 1    (Erfolg)
+------- Kanal  = 0    (gleiche Nummer wie in der Abfrage)
```

Falls der Befehl falsch war, keinen Erfolg hatte oder nicht in einer Folge von anderen Daten übertragen werden durfte, dann wird eine Fehlermeldung erzeugt, die entsprechend der Kodierung 2 gesendet wird.

Fehlermeldungen

kein gültiger Befehl	INVALID COMMAND
nicht bereit,Zeile wurde ignoriert	TNC BUSY - LINE IGNORED
anal ist schon belegt	CHANNEL ALREADY CONNECTED
Station ist schon connected	STATION ALREADY CONNECTED

Als Beispiel;es wird diese Datenzeile gesendet:

```
03 00 0C 48 65 6C 6C 6F 20 74 68 65 72 65 2E 0D
!  !  !  H  e  l  l  o     t  h  e  r  e  .  CR
!  !  !                                       !
!  !  !      (dieses ist ein Datenbyte!) ----+
!  !  +---- Anzahl   = 12   (dreizehn Datenbytes)
!  +------- Info/cmd = 0    (dieses sind Informationen)
+---------- Kanal    = 3    (als Beispiel)
```

Hat der TNC kein RAM frei als Puffer zur Aufname der Zeile, dann wird er mit einer Code 2 Fehlermeldung zurückkommen mit der Information :

```
03 02 544C432042555359202D204C494C452049474C4F524544 00
!  !  T N C   B U S Y   -   L I N E   I G N O R E D  !
!  !                                                 !
!  +---- Code = 2 (Fehler)   mit Null abgeschlossen -+
+------ Kanal = 3 (gleich dem der Sendung)
```

Sendet man einen unsinnigen Befehl:

```
00 01 03 4A 55 4C 4B
!  !  !  J  U  N  K
!  !  !
!  !  +---- Anzahl   = 3 (vier Daten-Bytes)
!  +------- Info/cmd = 1 (dies ist ein Befehl)
+---------- Kanal    = 0 (egal welcher Kanal)
```

Dann erhält man die Beschwerde in der Code 2 Form:

```
00 02 49 4E 56 41 4C 49 44 20 43 4F 4D 4D 41 4E 44 00
!  !  I  N  V  A  L  I  D     C  O  M  M  A  N  D  !
!  !                                               !
!  +--- Code  = 2 (Fehler)  mit Null abgeschlossen-+
+------ Kanal = 0 (gleich dem in der Sendung)
```

Die verbleibenden Codes 3 bis 7 werden lediglich als Antwort auf ein Pollen mit dem "G" Befehl verwendet. In der Antwort kann auch ein Code "0" erscheinen, was einfach heisst, dass für den Kanal nichts vorliegt, d.h. nichts hat den Status verändert. Im Betrieb wird dies eine sehr häufige Antwort sein.

```
0x 00
!  !
!  +---- Code    = 0    (es liegt nichts vor)
+------- Channel = x    (wie im entsprechenden "G"-Befehl)
```

Code 3

Der Code 3 identifiziert zum "link status" also zu den bestehenden Verbindungen. Dazu das nächste Kapitel:

LINK STATUS MESSAGES

```
({n}) BUSY fm {call} via {digipeaters}
({n}) CONNECTED to {call} via {digipeaters}
({n}) LINK RESET fm {call} via {digipeaters}
({n}) LINK RESET to {call} via {digipeaters}
({n}) DISCONNECTED fm {call} via {digipeaters}
({n}) LINK FAILURE with {call} via {digipeaters}
CONNECT REQUEST fm {call} via {digipeaters}
({n}) FRAME REJECT (x y z) fm {call} via {digipeaters}
({n}) FRAME REJECT (x y z) to {call} via {digipeaters}
```

busy fm sagt die Station ist erreichbar,sagt aber,dass sie nicht "connected" werden kann. (beschäftigt, belegt!)

connected to sie sind verbunden mit (Rufzeichen)

reconnected to bestandene Verbindung wurde erneut hergestellt (unterbrochene Verbindung)

disconnected fm sie wurden getrennt von (Rufzeichen)

link failure Verbindung konnte nicht hergestellt werden, erbindung wurde unterbrochen

connect request eine (weitere) Station versucht zu "connecten"

frame reject Zurückweisen eines Datenrahmens

Diese Meldungen des TNC sollten bekannt vorkommen; sie sind die gleichen wie im Benutzer-Mode. Das {n} ist die Kanalnummer. Sie ist hier eigentlich redundant und könnte bei späteren Ausgaben der Firmware entfallen.

Nochmals als Beispiel: ich bin mit KB5MU direkt auf dem TNC-Kanal 2 verbunden. Die nächste Abfrage des Kanal 2 (zwischendurch kann noch alles mögliche abgefragt werden !) ergibt dann folgendes:

```
02 03 28322920434F4E4E454354454420746F204B42354D55 00
!  !  ( 2 )   C O N N E C T E D   t o   K B 5 M U  !
!  !                                                !
!  +---- Code  = 3 (Link Status) Null termination --+
+------- Kanal = 2
```

Codes 4 und 5

Beide indentifizieren eine Monitor Kopfzeile.Es handelt sich um eine Meldung der Form "mit Null abgeschlossen".Formal sieht das so aus:

```
fm {call} to {call} via {digipeaters} ctl {name} pid {hex}
```

Hier ist wieder die Monitor Kopfzeile identisch der im "Nutzer Mode". Ist der Code 4, dann ist diese Zeile alles was zu erwarten ist, denn es fehlt das Informations Feld. Als Beispiel wieder: Sie "monitoren" KB6C, der gerade auf ein "connect request" von KB5MU antwortet, dann erhalten aus auf dem Kanal 0 :

```
0004666D204B42364320746F204B42354D552063746C2055612070494420463000

! ! f m   K B 6 C   t o   K B 5 M U   c t l   U A   p i d   F 0 !
! !                                                             !
! +---- Code = 4 (Monitor, keine Info)   mit Null abgeschlossen-+
+------ Kanal= 0 (Monitor Info ist immer auf Kanal 0)
```

Wäre der Code=5 gewesen, dann hätte noch ein Informationsfeld dazu gehört, das man mit einem weiteren G Kommando vom Kanal 0 abholen könnte. Diese Monitor Daten wären mit einem Code von 6 angeliefert worden. Weiter oben wurde jedoch festgelegt, dass die Daten transparent sein sollen, folglich werden die Monitordaten im sog. "Byte-count"-Format gesendet, d.h hier bei den Daten wird nicht in Form "mit Null abgeschlossen" gesendet. Da die Codes 4,5,6 nur im Zusammenhang mit Monitor-Daten definiert sind, gehören sie auch automatisch nur zum Kanal 0.

Also wieder das Beispiel: KB6C sagt zu NK6K "Hi" und man sieht als Monitor Daten dann:

```
0005666D204B42364320746F204E4B364B2063746C2049303020704944204630 00
! ! f m   K B 6 C   t o   N K 6 K   c t l   I 0 0   p i d   F 0 !
! !                                   ! !                       !
```

```
! !                              oder anderes  --+-+                !
! !                                                                 !
! +---- Code = 5 (Monitor, Info folgt)  mit Null abgeschlossen -+
+------ Kanal = 0 (Monitor Info ist immer auf Kanal 0)
```

und die wirklich nächste Abfrage des Kanal 0 liefert:

```
00 06 02 48 69 0D
!  !  !  H  i  CR
!  !  !        !
!  !  !        +-----    (dieses ist ein Daten-byte)
!  !  +---- Anzahl = 2   (drei Daten-byte)
!  +------- Code   = 6   (Daten aus dem Monitorkanal)
+---------- Kanal  = 0   (der Monitorkanal ist der Kanal 0)
```

Ist der Code=7 dann sind damit Daten gekennzeichnet die aus einer bestehenden Verbindung stammen,also aus den Kanälen 1 bis 4. Wegen der Transparenz der Daten wird wieder im "byte-count"-Format gearbeitet. Nehmen wir wieder das "Hi", was als Daten gesendet wurde (auf Kanal 4). Es liegt jetzt im enpfangenden TNC vor.Dann erscheint durch Abfragen des Kanals 4:

```
04 07 02 48 69 0D
!  !  !  H  i  CR
!  !  !        !
!  !  !        +-----    (dieses ist ein Daten-byte)
!  !  +---- Anzahl = 2   (drei Daten-byte )
!  +------- Code   = 7   (Info aus einer bestehenden Verbindung)
+---------- Kanal  = 4   (Info war vom Kanal 4)
```

Abfrage des Kanal-Status: das L Kommando

Eine der wirkungsvollsten Möglichkeiten im sog.Host-mode ist die Abfrage eines Statusreports für einen bestimmten Kanal. Dieser Report ist ähnlich dem des sog. Benutzer-Mode, aber ausführlicher. Man erhält den Report in dem man ein L Kommando an den gewünschten Kanal des TNC sendet, der TNC wird mit einer Antwort mit dem Code 1 zurückkommen. Das Feld mit den Informationen besteht aus 6 Dezimalzahlen dargestellt in ASCII und getrennt durch ein Leerzeichen.

Eine typische Statusanfrage wäre:

```
01 01 00 4C
!  !  L
!  !  !
!  !  +---- Anzahl   = 0    (ein Daten-byte)
!  +------- Info/cmd = 1    (dieses ist ein Befehl)
+---------- Kanal    = 1    (Abruf Statusbericht f}r Kanal 1)
```

Ein typischer Statusreport wäre:

```
01 01 30 20 30 20 30 20 30 20 30 20 30 00
!  !  0     0     0     0     0     0  !
!  !                                   !
!  !           mit Null abgeschlossen -+
!  +------- Code  = 1    (Erfolg mit Info)
+---------- Kanal = 1    (dieses ist ein Report f}r Kanal 1)
```

Für den Kanal 0 gibt es weniger Informationen als für die normalen Kanäle, die für die individuellen gesicherten Verbindungen eingesetzt werden:

```
00 01 00 4C
!  !  !  L
!  !  !
!  !  +---- Anzahl   = 0    (ein Daten-byte)
!  +------- Info/cmd = 1    (dieses ist ein Befehl)
+---------- Kanal    = 0    (Abruf Kanalreport Kanal 0)
```

der Statusreport für Kanal 0 ist dann:

```
00 01 30 20 33 00
!  !  0     3  !
!  !           !
!  !           +---- mit Null abgeschlossen
!  +------- Code  = 1  (Erfolg mit Info)
+---------- Kanal = 0  (dies ist ein Report f. Kanal 0)
```

Nachfolgend die Erklärung zu den vom TNC erhaltenen Daten:

Format des Statusmeldung des jeweiligen Kanals

```
a b c d e f
```

a = Anzahl der Statusmeldungen, die noch nicht angezeigt wurden
(Number of link status messages not yet displayed)

b = Anzahl der erhaltenen Daten-Frames,die noch nicht angezeigt wurden,
i.e. beim Host Mode nicht abgeholt wurden
(Number of receive frames not yet displayed)

c = Anzahl der zu sendenden Daten-Frames, die noch nicht vom TNC abgesandt
werden konnten
(Number of send frames not yet transmitted)

d = Anzahl der schon gesendeten Daten-Frames für die noch die Bestätigung
aussteht
(Number of transmitted frames not yet acknowledged)

e = Anzahl der Versuche zu der derzeitigen Aktivität des TNC
(Number of tries on current operation)

f = Status der Verbindung (Link state)
Möglicher Status zur Verbindung ist:

0 = keine Verbindung / Beendet mit (disconnected)
1 = Verbindungsaufbau (Link Setup)
2 = Zurückweisen eines Daten-Frames (Frame Reject)
3 = Beendigung der Verbindung (Disconnect Request)
4 = Übertragung von Daten (Information Transfer)
5 = Frame mit Zurückweisung eines (Reject Frame Sent)
Daten-Frames wurde gesendet
6 = Warten auf Bestätigung (Waiting Acknowledgement)
7 = Gerät nicht verfügbar (Device Busy)
8 = vorgeschaltetes / fernbedientes (Remote Device Busy)
Gerät nicht verfügbar
9 = Beide Geräte nicht verfügbar (Both Devices Busy)
10 = Warten auf Bestätigung und Gerät nicht verfügbar
(Waiting Acknowledgement and Device Busy)
11 = Warten auf Bestätigung und fernbedientes Gerät nicht verfügbar
(Waiting Acknowledgement and Remote Busy)
12 = Warten auf Bestätigung und beide Geräte nicht verfügbar
(Waiting Acknowledgement and Both Devices Busy)
13 = *Reject Frame* wurde gesendet und Grät ist nicht verfügbar
(Reject Frame Sent and Device Busy)
14 = *Reject Frame* wurde gesendet und das ferbediente Gerät ist nicht
verfügbar
(Reject Frame Sent and Remote Busy)
15 = *Reject Frame* wurde gesendet und beide Geräte sind nicht
verfügbar
(Reject Frame Sent and Both Devices Busy)

Anmerkung 1: es gibt nur a und b für den Kanal 0.
Anmerkung 2: für die Version 1 AX.25 Protokoll gibt es nur den Status 0 bis 4

Was Sie nun mit den Informationen zum Status des jeweiligen Kanals anfangen ist weitgehend von Ihrem Anwender-Programm, bzw. der Aufgabenstellung abhängig. Ein interaktives Terminal-Programm könnte z.B. einen Teil oder alle Informationen anzeigen. Ein Mailbox-Programm könnte erst abwarten bis alle Daten-Frames angekommen d.h. bestätigt sind, um erst dann die Beendigung der Verbindung zu initialisieren. Gleiches gilt für ein Programm, was den Inhalt einer Mailbox (oder bestimmte Teile) an andere Mailboxen weitergibt. Dieses Programm müsste sicherstellen, dass alles angekommen ist.Ebenso ist ein Programm

denkbar, das alle obigen Informationen nutzt um einfach festzustellen, dass der Durchsatz zu gering ist und dann abbricht. Alle diese Möglichkeiten sind im Host-Mode einfach zu realisieren.

Im Falle eines Fehlers

Was passiert, wenn irgendwas auf der Verbindung Rechner/TNC nicht in Ordnung ist? z.B. ein Wackelkontakt oder Hochfrquenz gelangt über die Verbindungsleitung in den Eingang der RS232 Schnittstelle etc. Was immer auch der Grund ist, nehmen wir an es werden Daten verstümmelt. Verbindungen nach dem RS232 Standart sind sehr zuverlässig, aber trotzdem sollte Ihr Anwender-Programm mit einem Fehlerfall fertig werden.

Das Programm oder Programmstück,was den Fehlerfall behandelt, muss davon ausgehen,dass der TNC sich in einem nicht bestimmbaren Zustand befindet. Der TNC wartet vielleich gerade darauf weitere Bytes zu lesen, weil die ihm bekannte Anzahl Bytes (aus dem Byte-count) noch nicht übergeben wurden, oder der TNC wartet eigentlich nur auf irgendwelche Aktivitäten des Rechners. Dieser aber hat durch fehlerhafte Datenübertragung einen Fehlerfall entdeckt und muss versuchen die Synchronität zum TNC wiederherzustellen. Was kann man tun? Man schickt dem TNC eine Anzahl von Befehlen, die dafür sorgen sollen, dass er hoffentlich dann in einem bekannten Zustand ist. Sinnvollerweise sollte dieses aber so durchgeführt werden, dass möglichst wenig unsinnige Sachen auf der AX.25 passieren, also nichts gesendet wird.

Zunächst muss man -jetzt gehen wir davon aus,dass die Synchronität verloren ging- sich darüber im klaren sein,dass alle ankommenden Daten wegzuwerfen sind, selbst die aus einer noch bestehenden Verbindung. Wir wissen nicht, wozu die Daten gehören, also können wir sie nicht verwenden. Das ist zwar schade aber nicht zu ändern.

Dann senden wir eine Folge von ^A's,wobei wir jeweils nach dem Zeichen warten, ob eine Antwort des TNC erscheint.

Spekulieren wir ein wenig: die ersten ^A's werden dazu benutzt eine noch nicht komplette Anzahl von Bytes, die vom vorherigen Befehl dem TNC mitgeteilt wurden, aufzufüllen. Gehören diese Bytes zu einer bestehenden Verbindung, dann werden sie auch gesendet. Ist dann aber die komplette Anzahl der Bytes vom TNC empfangen (es können ja nicht mehr als 256 sein !) wirken die folgenden fünf ^A's als Befehl an den Kanal 1. Die Wahl des Zeichens ^A stellt sicher, dass der TNC irgendwann die Folge der Zeichen als Befehl interpretiert.

```
01 01 01 01 01
!  !  !  ^A ^A
!  !  !
!  !  +-- zwei Bytes
!  +----- ein Befehl
+-------- Kanal 1
```

Nun gibt es den Befehl ^A^A nicht und der TNC kommt mit seiner Antwort :

```
01 02 49 4E 56 41 4C 49 44 20 43 4F 4D 4D 41 4E 44 00
!  !  I  N  V  A  L  I  D     C  O  M  M  A  N  D  !
!  !                                               !
!  +---- Fehler           mit Null abgeschlossen --+
+------- muss Kanal 1 sein,wie im Befehl
```

Eigentlich muss es uns gleich sein,was der TNC genau antwortet, die Fehlermeldung ist nicht eindeutig. Nehmen wir z.B. nur ein einziges ^A , das aber die Anzahl für den *byte count* befriedigt. Dieses ^A gehört dann angenommen zu einem verstümmelten Befehl an den TNC. Der TNC wird dann mit einer Fehlermeldung kommen, die anders aussieht als die vorstehende. Was wir eigentlich suchen ist der Zeitpunkt, wo erstmals nach dem Zeichen eine Erwiderung des TNC vorliegt. Damit ist dann der Zeitpunkt gefunden, an dem der TNC auf einen Befehl wartet.

Dieses Verfahren setzt voraus, dass man lange genug wartet, nachdem man ein ^A an den TNC gesendet hat, um festzustellen, ob eine Antwort vom TNC erfolgt. Das kann den Vorgang langsam machen. Nehmen wir an, der TNC hat bis jetzt folgends erhalten:

00 00 FF ,damit wartet er auf 256 Bytes und eine Antwort

erscheint auch erst nach 256 Bytes. Diese wird den Erfolg melden, dass die Daten übertragen wurden. Man muss jedoch davon ausgehen, dass ein Versuch zur Rückerlangung der Synchronität nicht allzu häufig notwendig sein wird.

Danksagung und Kommentare

Es soll hier Ronald E.Raikes,WA8DED,gedankt werden,der die AX.25 Version 2 Multi-Kanal Firmware für den TNC geschrieben hat. Ebenso für SS,dem sog. split screen Terminal-Emulator, der die Vorteile des Hostmode

nutzt und dafür, dass er dieses Dokument auf Richtigkeit durchgesehen hat. Gedankt muss hier auch Mike Busch werden, W6IXU, dessen Software für die Mailbox mich das erste mal mit Host Mode in Berührung brachte.

Kommentare zu diesem Dokument sind erwünscht. Sie können sie an KB5MU/KA6IQA im WestNet senden, oder durch CompuServe (75265,367), durch Telefon (619-458-1238) oder durch die Post:

Paul T. Williamson, KB5MU; 6583 Edmonton Ave.; San Diego, CA 92122

SS,der split screen Terminal-Emulator ist ein exzellentes Beispiel für den Host Mode. Dieses Programm ist erhältlich für den Eigenbedarf und zur kostenlosen Weitergabe. Lauffähig ist das Programm auf dem IBM PC und sog. Kompatiblen .SS stellt für alle 5 Kanäle eine Übersicht bereit,ob sie 'connected' sind. SS hat einen Buffer für die geschriebenen Zeichen, für die emfangenen und zu sendenen Zeichen und unten im Schirm wird eine Zeile mit den Angaben zum Status der Verbindungen eingeblendet. SS ist in C geschrieben und zwar für den Microsoft C Compiler Version 3.0, mit einem Teil in Assembler,der als Interface zur Konsole fungiert. Schicken Sie mir eine Diskette und ausreichend Rückporto,dann freue ich mich Ihnen das Programm zu senden.

Die W6IXU Mailbox Software steht derzeit nicht für die Allgemeinheit zur Verfügung. Sie wird aber in Zukunft verfügbar sein, wenn sich das Programm stabilisiert hat und ein passendes Verfahren zur Verteilung gefunden wurde.

mini-Hostmode-Demo-Programm

mini-Hostmode-Demo-Programm für TNC2 + WA8DED Firmware;
geschrieben in GFA-BASIC 3.5; DG6UL Juni 90

Muticonnect auf 4 verschiedenen Ports
Getrennte Ein -und Ausgabe der Texte
Gleichzeitiges Monitorwindow

Installation: TNC-Baudrate im Listing eintragen.
In den Preferences XON/XOFF ausschalten und PARITY NONE

Anleitung: Umschalten der Ports mit F1-F4. F10 ist Monitor/Unproto.
Befehle werden mit Doppelpunkt eingeleitet.
TNC-Befehle müssen gross geschrieben werden
Connect und Disconnect-Befehl kann klein geschrieben werden
Verlassen des Programms geschieht mit :x

```
''
GOSUB install              ! Installation des TNC und der Software
'
'                            Hauptschleife
'                           ---------------
'
DO
  FOR kanalnummer|=0 TO 4              ! Kanal-Z_hler
    '
    z$=INKEY$                          ! Eingabe von Tastatur
    IF z$<>""
      IF ASC(z$)=155                   ! Funktionstaste gedrückt?
        GOSUB funktionstasten
      ELSE
        GOSUB eingabe
      ENDIF
    ENDIF
    '
    OUT #1,kanalnummer|,1,0,103        ! "G"-Sequenz
    kanal|=INP(#1)                     ! Antwort der Kanalnummer
    status|=INP(#1)                    ! Antwort Status
    OPENW #kanal|                    ! Ausgaben auf richtigen Schirm
    GOSUB status                       ! Verzweige Status
    '
    IF kanalnummer|>0                ! "L"-Abfrage nicht bei Kanal 0
      llflag|=1                        ! Flag fnr Statusabfrage
      OUT #1,kanalnummer|,1,0,108      ! "L" zum TNC senden
      kanal|=INP(#1)                   ! Antwort der Kanalnummer
      status|=INP(#1)                  ! Antwort Status
      OPENW #kanal|                  ! Ausgaben auf richtigen Schirm
      GOSUB status                     ! Verzweige Status
    ENDIF
  NEXT kanalnummer|
LOOP
'
'
PROCEDURE install
  ' ------------------------------------------------------------
  '   Dieser Programmteil halt folgende Aufgaben:
  '   Dimensionierung der Variablen, definieren und ÷ffnen der Screens,
  '   TNC in der Hostmode schalten und konfigurieren.
  ' ------------------------------------------------------------
  '
  DIM b$(30)                           ! TNC-Befehle
```

```
  DIM call$(5)                          ! Fnr Balkenanzeige
  DIM balken$(5)                        ! Fnr Balkenanzeige
  DIM neu$(5)                           ! Fnr Balkenanzeige
  DIM alt$(5)                           ! Fnr Balkenanzeige
  DIM eingabe$(4)                       ! Eingabestring
  DIM topbalken$(5)                     ! Oberer Balken
  lf$=CHR$(10)
  cr$=CHR$(13)
  sctitel$=" Host-Mode-Demonstration von DG6UL JUNI 1990   Help: siehe
Listing"
  '
  OPENS 5,0,0,640,256,2,&H8000       ! Eigener Screen für programm
  TITLES #5,sctitel$
  OPENW #0,0,11,640,245,0,2048              ! Monitorfenster
  PRINT CHR$(27)+"[42m"+CHR$(27)+"[J"       ! F_rbt Monitorfenster
  OPENW #9,0,11,640,34,0,2048        ! Eingabefenster kanal 4
  OPENW #8,0,11,640,34,0,2048        ! Eingabefenster kanal 3
  OPENW #7,0,11,640,34,0,2048        ! Eingabefenster kanal 2
  OPENW #6,0,11,640,34,0,2048        ! Eingabefenster kanal 1
  OPENW #5,0,11,640,41,0,2048        ! Eingabefenster kanal 0
  OPENW #4,0,45,640,143,0,2049       ! Connect Kanal 4
  PRINT CHR$(27)+"[80u"            ! 80 Zeichen fnr Ausgabewindows
  OPENW #3,0,45,640,143,0,2049       ! Connect Kanal 3
  PRINT CHR$(27)+"[80u"
  OPENW #2,0,45,640,143,0,2049       ! Connect Kanal 2
  PRINT CHR$(27)+"[80u"
  OPENW #1,0,45,640,143,0,2049+4096         ! Connect Kanal 1
  PRINT CHR$(27)+"[80u"
  '
  FOR x=1 TO 4
    topbalken$(x)="("+STR$(x)+")"+" DISCONNECTED"   ! Default der Balken
    TITLEW #(x+5),topbalken$(x),sctitel$
  NEXT x
  topbalken$(0)="(0) Unproto/Monitor"
  TITLEW #5,topbalken$(0),sctitel$
  TITLEW #0," Monitor-Window",sctilel$
  OPENW #0
  a$="      "+STRING$(62,"*")                 ! PRG-Information
  PRINT lf$+lf$+lf$+lf$+a$+lf$+"      *       DEMO-Programm";
  PRINT " fnr den Hostmode von DG6UL 1990       *"+lf$+a$+lf$+lf$
  FRONTW #0
  FRONTW #5
  '
  '
  OPEN "o",#1,"com1:9600,n,8,1"     ! Schnittstelle wird ge÷ffnet
  PRINT #1,CHR$(24);                ! TNC-Eingabepuffer leer
  a$=INPUT$(LOC(#1),#1)             ! SER-buffer leer
  PRINT #1,CHR$(27);"e1";cr$;       ! Terminalmode-Befehl ECHO EIN
  PRINT #1,CHR$(27);"a1";cr$;       !                     AUTOLF EIN
  PAUSE 15                          ! kurz warten
  PRINT #1;CHR$(27);"i";cr$;        !                     MYCALL
  PAUSE 15                          ! kurz warten
  WHILE LOC(#1)<>0               ! warten bis Daten an Schnittstelle
    LINE INPUT #1,a$                ! Zeile mit MYCALL einlesen
  WEND
  PRINT "* Rufzeichen im TNC: ";a$    ! MYCALL ausgeben
  PRINT
  '
  PRINT #1;CHR$(27);"jhost1";cr$;     ! TNC in Hostmode schalten
```

```
  a$=INPUT$(9,#1)                       ! Antwort einlesen
  PRINT a$                              ! Antwort ausgeben
  '
  PRINT "* Hostmode aktiv. TNC wird konfiguriert *"
  b$(1)="T 33"
  b$(2)="U 0"                         ! TNC-Konfigutation im Hostmode
  b$(3)="W 10"
  b$(4)="F 3"
  b$(5)="P 64"
  b$(6)="M UISC"
  b$(7)="Y 4"
  b$(8)="@t2 100"
  FOR x|=1 TO 8
    GOSUB send_to_tnc(0,1,b$(x|))
  NEXT x|
  '
  FOR x|=1 TO 4                         ! Call-Variable leer
    call$(x|)="discon  "
    alt$(x|)=" U0 V0 Disconnected"
  NEXT x|
  GOSUB balken
  PRINT lf$+"OK";STRING$(10,lf$)  ! OK-Meldung: Install erfolgreich
  '
RETURN
'
'
PROCEDURE status
  ' ---------------------------------------------------------------
  ' Auswertung der TNC-Antworten:
  ' 0=Erfolg, 1/2=TNC-Meldung, 3=Linkstatus, 4/5=Header,
6/7=Informationen
  ' Ausgabe auf entsprechendens Window
  ' ---------------------------------------------------------------
  SELECT status|
  CASE 0                                ! Erfolg ohne Meldung
    '
  CASE 4,5                              ! Monitor Header
    a=1                                 ! damit kein 0
    a$=""
    WHILE a<>0                          ! Einlesen bis "0"
      a=INP(#1)
      a$=a$+CHR$(a)
    WEND
    PRINT CHR$(27)+"[33m"+a$+CHR$(27)+"[31m"   ! Ausgabe
    '
  CASE 1,2,3
    a$=" "                              ! Erfolg Fehler Linkstatus
    b$=""
    WHILE a$<>CHR$(0)                   ! Lese bis "0"
      a$=CHR$(INP(#1))
      b$=b$+a$
    WEND
    IF status|=3                        ! Wenn Linkstatus
      topbalken$(kanal|)=LEFT$(b$,LEN(b$)-20)  ! Anzeige Connectstatus
      TITLEW #kanal|+5,topbalken$(kanal|),sctitel$
      IF MID$(b$,5,3)="CON"           ! Bei Con Rufzeichen eintragen
        ruf$=MID$(b$,18,10)
        m=INSTR(ruf$," ")
```

```
        LSET call$(VAL(MID$(b$,2,1)))=LEFT$(ruf$,m-1)  ! Rufzeichen
eintragen
      ENDIF
      IF MID$(b$,5,3)="DIS" OR MID$(b$,5,6)="LINK F"   ! Bei d oder
linkfehler
        call$(VAL(MID$(b$,2,1)))="discon  "  ! Rufzeichen löschen
      ENDIF
      GOSUB balken                              ! -nderung in Balken
    ENDIF
    '
    IF llflag|=1       ! Statusflag: Ausgabe der Linkinformation
      llflag|=0                         ! Flag wieder 0
      nbp|=VAL(MID$(b$,7,1))      ! Anzahl nicht best_tigten pakete
      ver|=VAL(MID$(b$,9,2))      ! Anzahl der Versuche
      sta|=VAL(RIGHT$(b$,2))          ! Status
      SELECT sta|
      CASE 0
        sta$="Disconnected"
      CASE 1
        sta$=" Link Setup "
      CASE 2
        sta$="Frame Error"
      CASE 3
        sta$="Disc Request"
      CASE 4
        sta$="InfoTransfer"
      CASE 5
        sta$="Reject Sent "
      CASE 6
        sta$="Wait for ack"
      DEFAULT
        sta$="Busy Failure"
      ENDSELECT
      neu$(kanal|)=" U"+STR$(nbp|)+" V"+STR$(ver|)+" "+sta$
      IF neu$(kanal|)<>alt$(kanal|) AND kanal|>0    ! Nur bei Änderung
Balken-
        alt$(kanal|)=neu$(kanal|)                   ! Anzeige erneuern
        TITLEW #kanal|,balken$(kanal|)+neu$(kanal|),sctitel$
      ENDIF
      '
    ELSE
      PRINT CHR$(27)+"[33m";b$+lf$+CHR$(27);"[31m";   ! Ausgabe auf
Schirm
    ENDIF
    b$=""
    '
  CASE 6,7                             ! Information-Ausgabe
    l=INP(#1)+1
    a$=INPUT$(l,#1)
    '
    '
    'Routine zur Umwandlung von CR in LF innerhalb eines Strings
    '-----------------------------------------------------------
    start=1
    DO
      x=INSTR(start,a$,cr$)           ! Suche nach CR
      MID$(a$,x,1)=lf$                ! austauschen: CR=LF
      start=x+1
    LOOP UNTIL x=0
```

```
    '
    '      ------------------------------------------------
    '
    IF kanal|=0 AND RIGHT$(a$,1)<>lf$ AND a$<>cr$     !  +LF im Monitor
      a$=a$+lf$
    ENDIF
    PRINT a$;                                  ! INFO-Ausgabe
  DEFAULT                                      ! Falls Status falsch
    PRINT "Falsche TNC-Antwort bei PROCEDURE STATUS: ";status|
    DELAY 3
    STOP                                       ! Programmende
  ENDSELECT
RETURN
'
'
PROCEDURE eingabe
  ' ------------------------------------------------------------
  '  Eingaben von Tastatur werden ausgewertet.
  '  Dabei wird verhindert, da_ die beim Korrigieren eingegebenen
  '  Backspace-Zeichen gesendet werden. Überlange Eingaben werden
  '  nicht abgefangen.
  '
  OPENW #fwin|+5                               ! Eingabewindow ÷ffnen
  IF z$=CHR$(8)                                ! Wenn Eingabe=Backspace
    IF LEN(eingabe$(fwin|))>0                ! Nur wenn Eingabestring > 0
      eingabe$(fwin|)=LEFT$(eingabe$(fwin|),LEN(eingabe$(fwin|))-1)
      PRINT z$;" ";z$;                         ! Letztes Zeichen l÷schen
    ENDIF
  ELSE
    PRINT z$;                                  ! Zeichen ausgeben
    eingabe$(fwin|)=eingabe$(fwin|)+z$  !Eingabestring akualisieren
  ENDIF
  IF z$=cr$                                    ! Wenn RETURN
    PRINT
    t$=eingabe$(fwin|)                         ! Befehlsstring
    IF LEFT$(t$,1)=":"                         ! Wenn Befehl
      GOSUB befehl(fwin|)
    ELSE
      GOSUB send_to_tnc(fwin|,0,t$)            ! Senden
    ENDIF
    eingabe$(fwin|)=""                         ! Eingabestring leer
  ENDIF
RETURN
'
'
PROCEDURE send_to_tnc(k|,mode|,text$)
  ' ------------------------------------------------------------
  '  k| =Kanalnummer;  mode| =Modus;   text$ =zu sendender Text oder
Befehl
  ' ------------------------------------------------------------
  PRINT #1;CHR$(k|);CHR$(mode|);CHR$(LEN(text$)-1);text$;
  kanal|=INP(#1)
  status|=INP(#1)
  IF status|<>0                  ! Wenn kein Erfolg
    OPENW #kanal|
    GOSUB status
  ENDIF
RETURN
'
```

```
'
PROCEDURE befehl(kn|)
' ------------------------------------------------------------
' kn|=Kanalnummer
' Hier wertet das Programm die Befehle aus und unterscheidet
' zwischen TNC-und Programmbefehlen.
' Hier ist nur ein einziger Progr.befehl: x zum Beenden d. Software
  ' ------------------------------------------------------------
  IF t$=":x"+cr$                              ! (x) Prg-Ende
    GOSUB ende
  ELSE                                  ! wenn TNC-Befehl dann...
    t$=MID$(t$,2,LEN(t$)-2)                   ! Befehl an TNC
    x=ASC(LEFT$(t$,1))
    IF (x<91 AND x>63) OR x=99 OR t$="d"   ! Erm÷glicht Kleinschreiben
der
      GOSUB send_to_tnc(kn|,1,t$)     ! connect und discon Befehle
    ELSE                                      ! Wenn Gro_buchstaben
      PRINT "Eingabefehler"
      t$=""
    ENDIF
  ENDIF
RETURN
'
'
PROCEDURE funktionstasten
  ' ------------------------------------------------------------
  '                  Erm÷glicht umschalten der Windows
  ' ------------------------------------------------------------
  u|=ASC(MID$(z$,2,1))           ! liest Funktionstaste
  SELECT u|
  CASE 48                        ! Port 1
    FRONTW #1
    FRONTW #6
    fwin|=1
  CASE 49                        ! Port 2
    FRONTW #2
    FRONTW #7
    fwin|=2
  CASE 50                        ! Port 3
    FRONTW #3
    FRONTW #8
    fwin|=3
  CASE 51                        ! Port 4
    FRONTW #4
    FRONTW #9
    fwin|=4
  CASE 57                        ! Monitor
    FRONTW #0
    FRONTW #5
    fwin|=0
  ENDSELECT
RETURN
'
'
PROCEDURE ende
  ' ------------------------------------------------------------
  '        Windows schlie_en, TNC konfigurieren und in Terminalmode
  ' ------------------------------------------------------------
  b$(1)="M N"             ! Monitorbetriebsart OFF
```

```
  b$(2)="U 1"              ! OFFLINE-Text in TNC einschalten
  FOR x|=1 TO 2
    GOSUB send_to_tnc(0,1,b$(x|))     ! TNC-Befehl
  NEXT x|
  PRINT "TNC in Terminalmode"
  t$="jhost 0"
  GOSUB send_to_tnc(0,1,t$)           ! TNC in Terminalmode
  PRINT "OK"
  FOR x=0 TO 9                        ! Windows schlie_en
    CLOSEW #x
    CLOSE #x
  NEXT x
  EDIT                                ! Ende
RETURN
'
'
PROCEDURE balken
  ' ------------------------------------------------------------
  '                   Aktualisieren der  Balkenanzeige
  ' ------------------------------------------------------------
  FOR x|=1 TO 4
    balken$(x|)="1:"+call$(1)+" 2:"+call$(2)+" 3:"+call$(3)+"
4:"+call$(4)
    TITLEW #(x|),balken$(x|)+alt$(x|),sctitel$
  NEXT x|
RETURN
```

PS: Das Listing ist nur ein Beispiel wie man ein Terminalprogamm für den Hostmode der WA8DED-Software schreiben kann. Man kann sicher noch einiges verbessern.

Viel Spass beim ausprobieren, vy 73 de Holger (DG6UL @ DB0IE)

KISS - Protokollbeschreibung

KISS: "**K**eep **i**t **s**imply **s**tupid", ein einfaches Protokoll für die Kommunikation zwischen TNC und Rechner.

Die Standard-TNC-Software, der sogenannte "Terminal-Modus" wurde urprünglich für den Betrieb mit einfachen Terminals geschrieben; sie ist nicht auf die Leistungsfähigkeit heutiger Personal-Computer zugeschnitten. Das trifft vor allem auf Anwendungen mit Multi-Connect-Möglichkeit, Gateway etc. zu. Dazu kommt, daß Protokollverbesserungen wie "DAMA" oder der auch von DIGICOM her bekannte "Framesammler" ohne Umprogrammierung der TNC-Software unmöglich sind. Die TCP/IP Programme wie WAMPES oder NOS basieren auf dem KISS-Protokoll zwischen TNC und Rechner.

KISS löst dieses Problem, indem es dem TNC die Abarbeitung des AX.25-Protokolls und Befehlssatzes entzieht; der TNC wandelt nur noch das HF-seitige synchrone HDLC-Format in ein spezielles asynchrones auf der seriellen Schnittstelle verwendetes Frame-Format um. Das bedeutet natürlich, daß das AX.25-Protokoll sowie die Benutzerschnittstelle nun auf dem Rechner selbst implementiert werden müssen, der Rechner erhält dadurch aber vollständige Kontrolle über die auf der HF-Seite übertragenen HDLC-Frames; das wiederum steigert die Flexibilität erheblich und stellt für die heutigen leistungsfähigen Rechner keineswegs ein Problem dar.

Die Paketlänge bei üblichen AX.25 Verbindungen ist auf 256 Byte beschränkt. Das KISS im TNC3S unterstützt Datenpakete bis zu 1024 Byte, eine downloadbare Version des TNC3-KISS kann Pakete bis zu 2048 Bytes verarbeiten. (dazu sind 256 kByte RAM notwendig, da das TNC3 Platz für bis zu 16 Frames im RAM bereitstellen muß)

Wie funktioniert nun KISS im Einzelnen?

Das asynchrone Protokoll, mit dem sich Rechner und TNC unterhalten, ist sehr einfach; seine einzige Aufgabe besteht darin, die übertragenen Frames zu begrenzen. Jeder Frame beginnt und endet mit einem speziellen FEND (Frame End) Zeichen, ganz ähnlich, wie HDLC-Frames. Die Bildung und Überprüfung der CRC-Prüfsumme des AX.25 Frames wird dabei noch dem TNC überlassen. Der RS-232-Handshake wird nicht verwendet, daher genügt eine simple Drei-Draht-Verbindung.

Folgende spezielle Zeichen werden bei der Übertragung verwendet:

Abkürzung	Beschreibung	HEX-Wert
FEND	Frame End	C0
FESC	Frame Escape	DB
TFEND	Transposed Frame End	DC
TFESC	Transposed Frame Escape	DD

Frames werden durch das vor- und nachgestellte FEND-Zeichen sicher abgegrenzt, zwei aufeinanderfolgende FEND-Zeichen werden, ebenso wie bei HDLC-Frames, nicht als leerer Frame interpretiert. Die Frames werden transparent mit acht Datenbits, einem Stopbit und ohne Parity gesendet.

Ein innerhalb eines Frame vorkommendes FEND-Zeichen wird in die Sequenz FESC-TFEND übersetzt, analog dazu wird ein FESC-Zeichen als FESC-TFESC übertragen. Der jeweilige Empfänger (Rechner oder TNC) puffert die Zeichen, ein FEND bedeutet das Ende eines Frame. Der Empfang eines FESC-Zeichen schaltet den Empfänger in den "Escaped-Modus", und veranlasst ihn, das darauffolgende TFESC oder TFEND als FESC oder FEND in den Puffer zu schreiben. Ein TFEND oder TFESC, das nicht im Escape-Modus empfangen wird, wird als reguläres Zeichen betrachtet. Diese Übertragungsart mag auf den ersten Blick kompliziert erscheinen, ist aber sehr einfach zu implementieren und macht wenig Synchronisationsprobleme.

Dem TNC bleibt nur noch, die KISS-Frames in HDLC-Frames umzuwandeln, und umgekehrt, sowie die Frequenz zu überwachen und den Sender zu tasten. Der Rechner muss dazu einige wenige TNC-Parameter kontrollieren. Das erste Byte in jedem Frame auf der Verbindung zwischen Rechner und TNC unterscheidet zwischen Kommando- und Daten-Frames. Seine höheren vier Bit´s enthalten die Portnummer, die niederen vier Bit die Befehlsnummer. Das TNC3S nutzt diese 4 Bit zur Zuordnung der Frames zu den Modemkanälen.

Folgende Befehle sind definiert:

Befehl	Funktion	Bemerkung
0	Daten-Frame	Der Rest des Frame besteht aus Daten.
1	Tx-Delay	Das nächste Byte gibt die Zeit für Tx-Delay an. (in 10 ms Schritten). Default ist 50 (d.h. 500ms).
2	Persistence	Das nächste Byte gibt den Persistence-Wert p an (0 - 255). Folgende Formel wird verwendet: P = p * 256 - 1 Default für P ist 63 (d.h. p = 0.25).
3	SlotTime	Das nächste Byte ist das Slot- Intervall (in 10 ms Schritten). Default ist 10 (d.h. 100ms).
4	TX-Tail	Das nächste Byte ist die Zeit, die der Sender nach dem Senden der Frames noch hochgetastet bleibt (in 10 ms Schritten). 0-127 * 10 ms (= max 1.27 s), über 127 wird der Wert nochmal mit 100 multipliziert, d.h. man kann die Sendernachlaufzeit von 0,01-127 s einstellen. (Wichtig für Fullduplex-Digipeater, solange der Sender nachläuft entfällt das TX-Delay)
5	FullDuplex	Das nächste Byte ist 0 für Halb-Duplex, ungleich 0 für Vollduplex. Default ist 0, d.h. Halb-Duplex.
FF	Return	Verlassen des KISS-Modus.

Die ursprüngliche Idee eines einfachen Rechner/TNC-Protokolls hatte Brian Lloyd, WB6RQN. Phil Karn, KA9Q, einer der "Väter" des AX.25-Protokolls, organisierte die Spezifikation und legte am 6. August 1986 eine erste KISS-Version vor.

Quellen:

The KISS TNC, A simple Host-to-TNC communications protocol, Mike Chepponis, K3MC, Phil Karn, KA9Q

SMACK - Protokollbeschreibung

Version 1.0, Stand 27.02.91, von Jan Schiefer, DL5UE/GØTRR und Dieter Deyke, DK5SG/NØPRA leicht überarbeitet von Thomas Kunert, DC3SN im Juni 1993

Einführung

Ende 1990 wurde unter den Stuttgarter Packet-Radio-Amateuren erstmals konkret über die Datensicherung zwischen TNC und WAMPES(*)-Knotenrechner nachgedacht. Da bei anderen Packet-Knotensystemen bereits Datenverluste aufgetreten waren, überlegten wir uns, auf welche Art und Weise eine möglichst kompatible Erweiterung des KISS-Protokolles um eine Prüfsumme vorgenommen werden könnte. Das Resultat dieser Überlegungen bekam den Namen SMACK (**S**tuttgarter **M**odifiziertes **A**mateurfunk-**C**RC-**K**ISS). Dieses Kapitel soll die Unterschiede zwischen SMACK und KISS erläutern und Implementierungen auf anderen Systemen ermöglichen.

Was ist KISS?

KISS wurde im Jahre 1986 von Phil Karn, KA9Q vorgeschlagen [1]. Für seine TCP/IP-Software benötigte er ein Protokoll, das einen einfachen Zugang zu Packet-Radio unterhalb der AX.25-Protokollebene ermöglicht. KISS bietet einen Schicht 2a-Zugang. Aufgaben des TNC sind im wesentlichen nur noch die Zugriffssteuerung auf die Frequenz (Kanal-Belegt-Erkennung, P-Persistence-Verfahren) und die Wandlung der synchronen HDLC-Daten auf dem PR-Kanal in das asynchrone Format der RS232-Schnittstelle. Das KISS-Protokoll regelt die Abgrenzung einzelner Pakete mit Delimitern, die Behandlung eventuell im Datenstrom auftretender Delimiter und definiert einen einfachen Kommandosatz zur Einstellung von TNC-Parametern. Es wurden Vorkehrungen getroffen, um auch TNCs mit mehreren Packet-Kanälen (bzw. Modems) betreiben zu können.

Modifikation des KISS-Protokolles

Der Hostrechner kommuniziert mit KISS in Form von Paketen. Der Anfang eines solchen Paketes wird durch den Delimiter FEND gekennzeichnet. Dann folgt das sogenannte Kommandobyte. Es gibt an, ob es sich um ein Daten- oder ein Kommandopaket handelt und welches Kommando gemeint ist. Bis auf eine Ausnahme (Reset-Kommando = 255) benutzen alle Kommandos nur die unteren 4 Bit dieses Kommandobytes. Dies hat den Sinn, daß die oberen 4 Bit bei Multi-Kanal-TNCs den Kanal angeben können. So können 16 Kanäle einzeln parametriert und angesprochen werden.

Da uns weder 16- noch 8-Kanal-TNCs bekannt waren, haben wir das oberste Bit dieses Kommandobytes zweckentfremdet. Ist es gesetzt, so handelt es sich bei dem fraglichen Paket um ein Datenpaket mit CRC (nur bei Datenpaketen findet eine Prüfsummenberechnung statt!). Bei solchen Paketen wird am Ende des Frames die Prüfsumme angehängt, und zwar das niederwertige Byte zuerst. Die beiden folgenden Abbildungen zeigen das Rahmenformat eines Datenpaketes einmal ohne und einmal mit Prüfsumme für jeweils verschiedene Ports.

FEND	0x00	DATA	DATA	...	DATA	FEND

KISS-Rahmen ohne Prüfsumme für Port 1

FEND	0x10	DATA	DATA	...	DATA	FEND

KISS-Rahmen ohne Prüfsumme für Port 2

FEND	0x80	DATA	DATA	...	DATA	CRC LOW	CRC HIGH	FEND

Smack-Rahmen mit Prüfsumme für Port 1

FEND	0x90	DATA	DATA	...	DATA	CRC LOW	CRC HIGH	FEND

Smack-Rahmen mit Prüfsumme für Port 2

Es soll hier nochmals wiederholt werden, daß nur Datenpakete CRC-gesichert werden. Damit wird verhindert, daß keine Kommandos mehr an den TNC geschickt werden können, wenn sich Host und TNC über CRC / kein CRC uneins sind.

Umschalten von KISS auf SMACK

Ein SMACK-TNC arbeitet nach dem Einschalten im KISS-Modus, erzeugt also keine Prüfsumme. Sobald es das erste Paket mit CRC empfängt, schaltet es seine Senderoutinen ebenfalls auf CRC um. Dieser Betriebszustand kann dann nur noch durch einen Reset verlassen werden. Der Host verhält sich sinngemäss genauso. Ist jedoch ein KISS-TNC angeschlossen, so wird es CRC-Frames vom Host aufgrund des ihm unbekannten Kommandobytes verwerfen und normal weiterarbeiten.

Diese Methode hat den Vorteil, daß sowohl KISS- als auch SMACK-TNCs abwechselnd an einem Host betrieben werden können, ohne daß ein Umkonfigurieren notwendig ist. Das soll durch eine Darstellung der beiden Fälle veranschaulicht werden.

Fall 1: KISS-TNC

Host	TNC
- Sendet ein einziges Paket mit CRC, schaltet dann seinen Sender wieder auf Normal-KISS.	
	- Empfängt einen Rahmen mit dem für ihn unbekannten Kommandobyte 0x80 und verwirft ihn.
- Versendet KISS-Daten ohne Prüfsumme.	
	- Versendet KISS-Daten ohne Prüfsumme.

Fall 2: SMACK-TNC

Host	TNC
- Sendet ein einziges Paket mit CRC, schaltet dann seinen Sender wieder auf Normal-KISS.	
	- Empfängt einen Rahmen mit CRC-Kennzeichnung, schaltet seinen Sender auf CRC um.
- Versendet KISS-Daten ohne Prüfsumme.	
	- TNC sendet den ersten Rahmen mit CRC.
- Empfängt einen Rahmen mit CRC-Kennzeichnung, schaltet seinen Sender auf CRC um. - Versendet SMACK-Daten mit Prüfsumme.	
	- Versendet SMACK-Daten mit Prüfsumme.

Unabhängig vom Sendezustand (CRC / kein CRC) werden empfangene Rahmen immer wie folgt behandelt:

Empfangener Rahmen	Aktion
Kein CRC	Rahmen weiterverarbeiten
Mit CRC, Prüfsumme richtig	Rahmen weiterverarbeiten
Mit CRC, Prüfsumme falsch	Rahmen verwerfen

Dieses Protokoll setzt voraus, daß eine KISS-Implementierung ihr unbekannte Rahmen verwirft. Dies wird in der KISS-Spezifikation [1] gefordert.

CRC-Berechnung und Implementierungstips

Dies ist nicht der richtige Ort, um die Theorie der zyklischen Redundanzüberprüfung (cyclic redundancy check, CRC) zu erläutern. Hierzu sei auf die Arbeit von Michael Röhner, DC4OX [2] verwiesen. Dieser Abschnitt schildert nur die für eine Implementierung notwendigen Details.

Als Prüfpolynom wird das CRC16-Polynom verwendet. Dieses hat die Gestalt

$$X^{16} + X^{15} + X^2 + 1$$

Der CRC-Generator wird mit 0 vorbesetzt. Berechnet wird der CRC über alle Datenbytes einschliesslich des Kommandobytes 0x80.

Bekanntlich wird im KISS-Protokoll die Abgrenzung der Rahmen mit dem FEND-Zeichen (0xC0) durchgeführt. Der Fall, daß dieses Zeichen im Datenstrom vorkommt, wird gesondert behandelt. Dieser Vorgang wird SLIP-Encoding genannt.

Der CRC muß berechnet werden, bevor das SLIP-Encoding stattfindet, und überprüft werden, nachdem das SLIP-Decoding stattgefunden hat. Dafür gib es mehrere Gründe:

- Die CRC Bytes könnten FESC, TFEND, FEND usw. enthalten.

- Der SLIP En/Decoder wird in manchen Host-Implementierungen (z.B. WAMPES) auch unabhängig von KISS benutzt, um beispielsweise die Verbindung zum Unix-Kernel herzustellen. In diesem Fall wären CRC-Überprüfungen zwar auch wünschenswert, werden aber von der anderen Seite nicht verstanden.

Die CRCs gehören also logisch zum KISS Layer.

Die Berechnung findet wie folgt statt:

- CRC-Generator mit 0 vorbesetzen.

- Alle Datenbytes nacheinander in den Algorithmus hineintun, einschliesslich der beiden CRC-Bytes.

- Am Ende muss wieder 0 im CRC-Generator stehen. Ist der Wert ungleich 0, so ist ein Übertragungsfehler aufgetreten und der Rahmen muss verworfen werden.

Verschiedene Algorithmen für den CRC-Generator werden in [2] beschrieben.

Die CRC-Tabelle läßt sich mit folgendem kleinen C-Programm aufbauen:

```
unsigned short Table[256];
const int Rest[8] = { 0xC0C1, 0xC181, 0xC301, 0xC601, 0xCC01, 0xD801, 0xF001, 0xA001 };
main()
{        int i, j;
        unsigned short value;
        for (i = 0; i < 256; i++) {
                value = 0;
                for (j = 0; j < 8; j++)
                        if (i & (1 << j))
                                value ^= Rest[j];
                Table[i] = value;
        }}
```

Wird dieser Algorithmus in Assembler codiert, so benötigt er deutlich weniger Platz als die Speicherung der Tabelle selbst. Die Theorie findet sich wiederum in [2].

Implementierungen

Bisher wurde dieses Protokoll in folgenden Systemen implementiert:

- WAMPES

- SMACK, Version 1.3. Diese Software wurde von Jan Schiefer, DL5UE, aus der von K3MC geschriebenen TNC2-KISS-Implementierung heraus weiter entwickelt. Sie wird insbesondere in den TNCs der WAMPES-Knoten DBØID (Digipeater Stuttgart) und DBØSAO (Mailbox Stuttgart) eingesetzt da sie auch einige WAMPES-spezifische Anpassungen enthält.

- Im KISS der NORD><LINK-Firmware, ab Version 2.4

- Für die TCP/IP-Software NOS gibt es von Thommy Osterried, DL9SAU, einen 'Umrüstsatz', der NOS um die SMACK-Fähigkeiten erweitert.

- Im TNC3S

Diese Protokollbeschreibung hat vorläufigen Character. Die Autoren (dl5ue@dbØsao, dk5sg@dbØsao) sind für Verbesserungsvorschläge, Hinweise auf Fehler oder sonstige Kommentare dankbar.

Literatur

[1] Karn, Phil, KA9Q; Proposed "Raw" TNC Functional Spec, 6.8.1986; veröffentlicht in USENET-News;

[2] Röhner, Michael, DC4OX; Was ist CRC?; veröffentlicht im Packet-Radio Mailbox-Netz, Mai 1988

[3] FTP Software, Inc.; PC/TCP Version 1.09 Packet Driver Specification; Wakefield, MA 1989

[4] Schiefer, Jan, DL5UE; WAMPES - Weiterentwicklung; Vortrags-Skriptum des 5. überregionalen Packet-Radio-Treffens; Frankfurt 1989;

KISS - protocol description

KISS: "**K**eep **i**t **s**imply **s**tupid", a simple communication protocol for communication of TNC and computer.

The standard tnc-software, the so-called "terminal-mode" was originally written for use with simple data terminals, it is not suited to be used with the powerful computers of today. This means especially applications as multi-connect, gateway etc. In addition, there have been some protocol enhancements developed, like "DAMA" or the frame-collecting program known from DIGICOM, ", which are impossible to run on a tnc without ab´dding net program-eproms to the tnc. The TCP/IP programs as WAMPES or NOS are based on the KISS-protokoll between tnc and computer.

KISS solves this problem by passing the processing of AX.25 protocol and command set from the tnc to the computer. The tnc will only convert the analogue synchronous hdlc signal into a special, asynchronous, frame-oriented rs232 signal on the serial interface. But this has the effect, that the ax.25 protocol handling and user interface has to be implemented on the computer but on the tnc. The computer has a complete control on all hdlc data. This increases flexibility significantly and means no severe problem for the fast computers of today.

The packet-length with common ax.25 connections is limited to 256 byte. The kiss of TNC3S supports data packets up to 2048 byte.

What does 'kiss' mean? here some detail:

The asynchronous protocol, which is used for computer - tnc communication, is very simple. Its only task is to delimit the transferred packets. Every frame begins and ends with a special FEND (Frame End) character, similar to the hdlc frames. The generation and verification of the CRC-checksum of the AX.25 frames is done by the tnc. RS232-handshake is not used. So, a simple 2 wire rs232 connection will do.

The following characters are used for the communication:

Abbreviation	**Description**	**hex value**
FEND	Frame End	C0
FESC	Frame Escape	DB
TFEND	Transposed Frame End	DC
TFESC	Transposed Frame Escape	DD

Frames are separated by the trailing leading FEND-characters, two adjacent FEND-characters are (as usual with HDLC-Frames) not interpreted as a empty frame. Frames are sent with eight data bits, one stop bit and no parity.

A FEND character, which appears within a frame, is translated to the sequence FESC-TFEND, analogue to this, the FESC-character is transferred as FESC-TFESC. The receiver (computer or tnc) buffers the characters. A FEND means end of frame, FESC-Zeichen switches the receiver into "escaped-mode", and causes to write the TFESC or TFEND which follows to write as FESC or FEND into the buffer. A TFEND or TFESC, which was received when not in escape mode, is interpreted as a regular character. This This way of communication between tnc and computer may seem complicated, but is easy to implement and causes no problems according synchrionisation.

The only job for the tnc is to translate the KISS frames into HDLC and vice versa, further to control the channel access (check if the frequency is busy) and to key the transmitter. There are few tnc-parameters which can be set by the computer. The first byte of the frame which goes from computer to tnc defines, if the frame is a command or data frame. The upper four bit of the byte contain the port number, the lower four the command number. TNC3S uses the four upper bytes for assigning one of its two modem ports.

The following commands are available:

command	**function**	**remark**
0	data-frame	rest of frame is data
1	tx-delay	the next byte defines the time for tx-delay. 10 ms steps, default ist 50=500ms.
2	persistence	the next byte defines the persistence-value p (0 - 255). The formula P = p * 256 - 1 is used. Default is P= 63 (i.e. p = 0.25).
3	slot time	the next byte defines the slot-interval. (10 ms steps). Default is 10=100ms.

4	tx-tail	the next byte is the time, the transmitter is keeped keyed up after the frame transmission is completed. (10 ms steps, 0-127 * 10 ms = 0,01 to max 1.27 s) for values greater then 127 this value is multiplicated by 100, i.e. for values of 128 to 255 the txtail time is 1 to max 127s. (Inportant for fullduplex digipeaters. As long as the txtail time is not expired, there is no additional tx-delay.
5	full duplex	the next byte is 0 for half-duplex, not equal 0 for full-duplex. Default 0.
FF	return	exit KISS-mode.

The original idea of a simple protocol for computer / tnc communication was published by Brian Lloyd, WB6RQN. Phil Karn, KA9Q, one of the 'fathers' of the ax.25-protocol, organised the specification and presented the first kiss version at 6. August 1986.

Source:

The KISS TNC: A simple Host-to-TNC communications protocol
Mike Chepponis, K3MC, Phil Karn, KA9Q

SMACK - protocol description

Version 1.0, 27. Feb. 91

by Jan Schiefer, DL5UE/GØTRR and Dieter Deyke, DK5SG/NØPRA

1. introduction

In 1990, amateur-packet-radio-amateurs of area of Stuttgart in Germany thought about the idea to secure the data transfer between tnc and the WAMPES node controller against transmission errors on the rs232 connection. As there have been transmission errors or data loss found on other node controllers, we thought about how to enhance the kiss-protocol by a addtional checksum while remaining compatible with the original kiss-mode protocol. The result of these reflections was called SMACK (**S**tuttgarts **M**odified **A**mateur-radio-**C**RC-**K**ISS). This chapter will explain the differences between SMACK and KISS and make it possible to implement SMACK on other systems.

2. What is KISS?

KISS was proposed by Phil Karn, KA9Q in 1986. [1]. For his TCP/IP-software, he needed a protocol which makes it possible to have a simple access to packet-radio below the ax.25 protocol level. KISS offers a level 2a-access. The remaining tasks of the tnc are mainly to control the channel access (channel-busy recognition, p-persistence-procedure) and the translation of synchronous HDLC-data of the packet-channel to the asynchronous format on the rs232-interface. The KISS-protocol controls the separation of single packets by by delimiters, the processing of the delimiters which appear in the data stream and defines a simple set of commands for setting the tnc parameters. Provisions were made to operate tncs with more than one packet-channels (or modems).

3. Modification of the KISS-Protocol

The host computer communicates in KISS-mode by use of packets. The begin of such a packet is marked by a FEND character as delimiter, followed by a so-called command-byte. This byte defines, whether the packet contains data or commands (and which command follows). Except the reset-command (255), all commands use only the lower four bit of the command byte. This makes it possible to use the upper four bytes for the channel-number for multi-channel-tnc. So, it is possible to adress 16 channels and set their parameters individually for 16 separate channels.

As we have never heard about 16- or 8-channel tnc, we used the most significant bit of the command byte for another purpose. If it is set, the packet is a data-packet containing additional a crc checksum (the crc is generated with data packets only). With such packets, the crc checksum is appended at the end of the frame, lower byte first. The following two pictures show the frame format of a data packet without and including the checksum for different tnc-ports.

FEND	0x00	DATA	DATA	...	DATA	FEND

KISS-frame without checksum for port 1

FEND	0x10	DATA	DATA	...	DATA	FEND

KISS-frame without checksum for port 2

FEND	0x80	DATA	DATA	...	DATA	CRC LOW	CRC HIGH	FEND

SMACK-frame with checksum for port 1

FEND	0x90	DATA	DATA	...	DATA	CRC LOW	CRC HIGH	FEND

SMACK-frame with checksum for port 2

Only data packets are protected by the crc checksum. This avoids a deadlock situation if tnc and host disagree on crc / no crc format options.

4. Switching from KISS to SMACK mode

A SMACK-tnc runs in KISS mode after power-on, i.e. it generates no checksum. As soon as the first packet including a crc was received, it changes its transmit routines to crc. The only way to exit this operating mode, is to send a reset command. The host computer operates in a similar way. But if a KISS-tnc (instead of a SMACK-tnc) is connected, it will ignore the CRC-Frames of the host, as the command byte structure is unknown to him and continue in the normal way.

This method has the advantage, that KISS- and SMACK-tnc may be operated with the same host computer without the need to change configuration. This is demonstrated by the following two cases:

case 1: KISS-TNC

Host	KISS-TNC (non-SMACK)
- sends a single packet with crc, switches its transmitter to normal kiss mode	
	- receives a frame with the apparent invalid command byte 0x80 and ignores it.
- transmits KISS-data without checksum only	
	- transmits KISS-data without checksum

case 2: SMACK-TNC

Host	SMACK-TNC
- sends a single packet with crc, switches its transmitter to normal kiss mode	
	- receives a frame with crc-label, switches to crc-transmit mode
- transmits KISS-data without checksum	
	- TNC transmits its first frame *with* crc.
- receives a frame with crc label, switches its transmitter to crc-mode - transmits SMACK-data with crc	
	- transmits SMACK-data with crc

independant from the transmitters state (crc / no crc), all received frames are treated as follows:

received frame	action
no crc	process frame
with crc, checksum ok	process frame
with crc, wrong checksum	ignore frame

this protocol assumes, that the kiss implementation ignores frames with unknown control byte. This is demand is specified in [1].

5. CRC-generation and hints on implementation

We won't try to explain the theory of cyclic redundancy check algorithms. More about that matter is found in a article of Michael Röhner, DC4OX [2]. This paragraph explains only the details which are necessary for the implementations.

as check-polynom, the crc16-polynom is used. The form of this is

$$X^{16} + X^{15} + X^2 + 1$$

The crc-generator is started with a seed value of 0. The crc is calculated for all data bytes, including the command byte 0x80.

As known, for delimiting the KISS frames the FEND-character (0xC0) is used. The case that this character appears in the data stream is explained separately. This method is called SLIP-encoding.

The crc has to be calculated before the SLIP encoding takes place and has to be checked after the SLIP-decoding is completed. There are some reasons for that:

- the crc bytes might contain FESC, TFEND, FEND etc.
- the SLIP en/decoder is used in some host-implementations (e.g. WAMPES) independantly from KISS to establish e.g. the connection to the unix-kernel. For this case, crc checks would be no advantageous, but would be not understood by the other side.

The crc are a logical part of the KISS layer.

The crc calculations is done in that way:

- preset the crc generator with the value of 0.
- put all data byte into the crc algorithm one by one, including the crc bytes.
- at the end of the procedure, a 0 has to be found in the crc generator. If the value is not equal to zero, a transmission error is detected and the frame has to be discarded.

Descriptions of some different crc-generator algorithms are found in [2].

The crc-table may be set up by a little C-program:

```
unsigned short Table[256];
const int Rest[8] = { 0xC0C1, 0xC181, 0xC301, 0xC601, 0xCC01, 0xD801, 0xF001, 0xA001 };
main()
{        int i, j;
        unsigned short value;
        for (i = 0; i < 256; i++) {
                value = 0;
                for (j = 0; j < 8; j++)
                        if (i & (1 << j))
                                value ^= Rest[j];
                Table[i] = value;
        }}
```

If this algorithm is coded in assembler language, it needs much less memory space as the table itself. About the theory see [2].

6. Implementations

The SMACK protocol was implemented on the following systems (spring 1994):

- WAMPES

- SMACK, version 1.3. This software was developed by Jan Schiefer, DL5UE, based on the tnc-kiss program by K3MC. It is used in the tncs of the WAMPES nodes as DBØID (digipeater Stuttgart) or DBØSAO (mailbox Stuttgart), as it contains some WAMPES-specific features.

- KISS-mode of NORD><LINK-tnc-firmware (versions 2.4 and later)

- TCP/IP-software NOS by Thommy Osterried, DL9SAU. A 'enhancement-package' which adds the SMACK features to NOS.

- In the TNC3S kiss.apl.

This protocol description is preliminary and subject th further developments. The authors (dl5ue@dbØsao, dk5sg@dbØsao.#bw.deu.eu) are thankful for any comments.

Literature

[1] Karn, Phil, KA9Q; Proposed "Raw" TNC Functional Spec, 6.8.1986; published in USENET-News;

[2] Röhner, Michael, DC4OX; Was ist CRC?; published in the pr-net in may 1988

[3] FTP Software, Inc.; PC/TCP Version 1.09 Packet Driver Specification; Wakefield, MA 1989

[4] Schiefer, Jan, DL5UE; WAMPES - Weiterentwicklung; Script of the 5. packet-meeting Frankfurt 1989;

Multi Channel KISS TNC3

As KISS mode is used with NOS in most cases, we show a example how to configure the autoexec.nos (or net.rc) file. The TNC3 takes the "port number" from the upper nibble of KISS-type-byte of KISS-protocol. To activate this feature, proceed as follows:

The interface to the tnc is activated as usual by use of the "attach"-line (attach asy..... or attach COM ....). With the predefined parameters, the channel number 0 is used. this refers to port 1 (first modem).

For use of the channel numbers 1-15 you have to add further "attach"-lines:

ATTACH KISS <parent> <channel> <label> ...

<parent> is the physical interface, defined previously with "attach"

<channel> ist the channel number, which will be used by the tnc. Use 0 for the first (1.) modem of TNC3 or 1 for the second (2.) modem.

Example:

ATTACH KISS tnc1 1 ax1

makes the second channel of the TNC3 acessible as "ax1"

AX.25 Version 2 Multi-channel TNC FIRMWARE

(version 2.6) copyright 1990, Ronald E. Raikes (WA8DED)

This firmware supports the full AX.25 link-layer protocol, version 2.0 as described in the ARRL specification dated October 1984, as well as the pre-existing version 1.x. This implementation supports multiple simultaneous link connections with either version protocol. This release has been compiled for a maximum of four connections, although any reasonable number of connections is possible by changing one MAXLNK symbol in the source header file. The firmware is contained in one 26256 EPROM, and is intended to be installed in a TAPR TNC-2 (or equivalent, such as the MFJ-1270 or AEA PK-80) in socket U23. RAM memory is automatically sized, supporting both 16k and 32k configurations.

Commands and information are sent to the TNC in the form of lines. Lines may be up to 256 characters long, including the terminating CARRIAGE RETURN. If the 256th character entered is not a CARRIAGE RETURN, it will be discarded and a BELL character will be output to the terminal. BACKSPACE and DELETE may be used to remove single characters from the line. The entire line may be permanently backspaced out by entering a CONTROL-U or CONTROL- X. A CONTROL-R will temporarily backspace out any partial line to allow incoming frames to be displayed. A second CONTROL-R will then restore the line to allow continuation of entry. During the time a partial line is saved, only another CONTROL-R will be accepted from the keyboard (with the exception of xon/xoff, of course). BELL characters are echoed to the terminal when entered or removed. Lines which begin with an ESCAPE character (echoed as '* ') are interpreted as commands. If a command is issued with no parameter, the current value of that commands parameter is displayed. Lines without a leading ESCAPE character are sent as information.

The firmware provides the operator with five virtual TNC channels, numbered 0 to 4. The terminal is logically attached to only one of these channels at a time, selected by the 'S' command. Information sent on channel 0 is always unproto. The unproto path may be set by issuing a 'C' command when channel 0 is selected. Channels 1 - 4 are also unproto if they are not currently connected. Outgoing connect requests may be issued on any unconnected channel, while incoming connect requests will use the first available channel (provided the maximum number of connections set by the 'Y' command will not be exceeded).

Information received on a connected channel that is not currently selected will remain queued there until that channel is selected. The STA led indicates there is queued information, and the 'L' command may be used to determine the channel(s) where it is located. Information for transmission is sent only to the currently selected channel. When a connection is ended, received information will remain queued until it has been displayed. If a new digipeater path is desired while a connection is being established or is in progress, it is not necessary to disconnect first. Simply re-issuing the 'C' command will re-establish the connection via the new path without any loss of information.

Which protocol version is used to initiate a connection is controlled by the 'V' command, but the version will be changed automatically, if necessary, to conform to the version of the TNC responding. Version 2 protocol is more effecient in terms of network throughput and loading, especially under severe conditions. Version 2 protocol is the default and should be used whenever possible. When version 2 protocol is used, a watch-dog timer is started whenever information is not being transmitted. If the TNC remains idle for three minutes, it will poll the other TNC to determine if the link is still established. If no response is received after the number of tries set by the 'N' command, a link failure is reported. This procedure will also detect the case where someone connects and then leaves without disconnecting. Changing the protocol version during a connection is not permitted.

The 'F', 'I', 'N', 'O', and 'V' commands maintain individual parameters for each channel. The value stored in channel 0 is used to initialize channels 1 - 4 upon power up, and to re-initialize channels 1 - 4 after a disconnect. This allows the values to be changed independently on each channel, prior to and during a connection, and then automatically revert back to the standard values when the connection is ended. A 'D' command issued on a disconnected channel 1 - 4 will also re-initialize that channel.

Frame monitoring is controlled by the 'M' command. The command parameter determines the types of frames monitored, and is a list of desired frames chosen from the letters in the following table:

N	None
I	I frames
U	UI frames
S	Supervisory frames
C	Monitor while connected
+	Call signs to be included (maximum of 8)
-	Call signs to be excluded (maximum of 8)

The '+' and '-' parameters may not be used together. If either is used, it must be the last parameter (followed by one to eight callsigns, if applicable). If no list of callsigns is specified to be included or excluded, all callsigns will be candidates for monitoring. Entering a '+' or '-' with no callsigns will empty the list.

An asterisk displayed after a callsign in the digipeater list indicates the frame was transmitted by that station. The control field displayed will be one of the following:

NAME DESCRIPTION

RRa	Receive Ready
RNRa	Receive Not Ready
REJa	Reject
UI	Unnumbered Information
DM	Disconnected Mode
SABM	Connect Request
DISC	Disconnect Request
UA	Unnumbered Acknowledge
FRMR	Frame Reject
Iab	Information
?ccH	Unknown

a = Next expected frame number (0 - 7)
b = Frame number of this frame (0 - 7)
cc = Hexadecimal value

In addition, one of the following characters will be displayed, reflecting the protocol version, command/response bits, and the poll/final bit:

(blank) = version 1 frame without poll/final bit
! = version 1 frame with poll/final bit
^ = version 2 command frame without poll bit
+ = version 2 command frame with poll bit
- = version 2 response frame with final bit
v = version 2 response frame without final bit

The protocol identifier field is displayed in hexadecimal

An unattended mode, controlled by the 'U' command, provides for sending user supplied text to a connecting station, and then allows that station to leave a brief message. This mode can operate on all channels simultaneously, but in no way limits the operators ability to interact with one of the connected channels or the ability to make outgoing connect requests. When unattended mode is enabled, link status messages are queued to the associated channel and not output to the terminal unless that channel is currently selected. Link status messages will therefore be displayed in chronological order with the information from that channel. In addition, text supplied by the user with the 'U' command will be sent to any station that connects. If channel 0 is left selected, stations may then connect and leave messages on channels 1 - 4 (limited by the 'Y' parameter, of course). The 'L' command may be used to determine if messages have been left on any channel. Selecting a channel containing messages will cause all link status and information from that channel to be displayed. If xon/xoff handshaking is enabled, CONTROL-S and CONTROL-Q may be used to regulate the output to the terminal to allow comfortable reading.

MINI-HOSTMODE-DEMO-PROGRAMM FUER TNC 2 + WA8DED FIRMWARE

geschrieben in GFA-BASIC 3.5 DG6UL Juni 90

Muticonnect auf 4 verschiedenen Ports, Getrennte Ein -und Ausgabe der Texte, Gleichzeitiges Monitorwindow,

Installation: TNC-Baudrate im Listing eintragen.

In den Preferences XON/XOFF ausschalten und PARITY NONE

Anleitung: Umschalten der Ports mit F1-F4. F10 ist Monitor/Unproto.

Befehle werden mit Doppelpunkt eingeleitet.

TNC-Befehle müssen gross geschrieben werden

Connect und Disconnect-Befehl kann klein geschrieben werden

Verlassen des Programms geschieht mit :x

```
GOSUB install              ! Installation des TNC und der Software
Hauptschleife
DO
  FOR kanalnummer|=0 TO 4              ! Kanal-Z_hler
    '
    z$=INKEY$                          ! Eingabe von Tastatur
    IF z$<>""
      IF ASC(z$)=155                   ! Funktionstaste gedrnckt?
        GOSUB funktionstasten
      ELSE
        GOSUB eingabe
      ENDIF
    ENDIF
    OUT #1,kanalnummer|,1,0,103        ! "G"-Sequenz
    kanal|=INP(#1)                     ! Antwort der Kanalnummer
    status|=INP(#1)                    ! Antwort Status
    OPENW #kanal|                      ! Ausgaben auf richtigen Schirm
    GOSUB status                       ! Verzweige Status
    IF kanalnummer|>0                  ! "L"-Abfrage nicht bei Kanal 0
      llflag|=1                        ! Flag fnr Statusabfrage
      OUT #1,kanalnummer|,1,0,108      ! "L" zum TNC senden
      kanal|=INP(#1)                   ! Antwort der Kanalnummer
      status|=INP(#1)                  ! Antwort Status
      OPENW #kanal|                    ! Ausgaben auf richtigen Schirm
      GOSUB status                     ! Verzweige Status
    ENDIF
  NEXT kanalnummer|
LOOP
PROCEDURE install
  '   Dieser Programmteil halt folgende Aufgaben:
  '   Dimensionierung der Variablen, definieren und ÷ffnen der Screens,
  '   TNC in der Hostmode schalten und konfigurieren.
  DIM b$(30)                           ! TNC-Befehle
  DIM call$(5)                         ! Fnr Balkenanzeige
  DIM balken$(5)                       ! Fnr Balkenanzeige
  DIM neu$(5)                          ! Fnr Balkenanzeige
  DIM alt$(5)                          ! Fnr Balkenanzeige
  DIM eingabe$(4)                      ! Eingabestring
  DIM topbalken$(5)                    ! Oberer Balken
  lf$=CHR$(10)
  cr$=CHR$(13)
  sctitel$=" Host-Mode-Demonstration von DG6UL JUNI 1990   Help: siehe Listing"
  OPENS 5,0,0,640,256,2,&H8000             ! Eigener Screen für programm
  TITLES #5,sctitel$
  OPENW #0,0,11,640,245,0,2048             ! Monitorfenster
  PRINT CHR$(27)+"[42m"+CHR$(27)+"[J"      ! F_rbt Monitorfenster
  OPENW #9,0,11,640,34,0,2048              ! Eingabefenster kanal 4
  OPENW #8,0,11,640,34,0,2048              ! Eingabefenster kanal 3
  OPENW #7,0,11,640,34,0,2048              ! Eingabefenster kanal 2
  OPENW #6,0,11,640,34,0,2048              ! Eingabefenster kanal 1
  OPENW #5,0,11,640,41,0,2048              ! Eingabefenster kanal 0
  OPENW #4,0,45,640,143,0,2049             ! Connect Kanal 4
  PRINT CHR$(27)+"[80u"                    ! 80 Zeichen fnr Ausgabewindows
  OPENW #3,0,45,640,143,0,2049             ! Connect Kanal 3
  PRINT CHR$(27)+"[80u"
  OPENW #2,0,45,640,143,0,2049             ! Connect Kanal 2
  PRINT CHR$(27)+"[80u"
  OPENW #1,0,45,640,143,0,2049+4096        ! Connect Kanal 1
  PRINT CHR$(27)+"[80u"
  '
  FOR x=1 TO 4
```

```
    topbalken$(x)="("+STR$(x)+")"+" DISCONNECTED"   ! Default der Balken
    TITLEW #(x+5),topbalken$(x),sctitel$
  NEXT x
  topbalken$(0)="(0) Unproto/Monitor"
  TITLEW #5,topbalken$(0),sctitel$
  TITLEW #0," Monitor-Window",sctilel$
  OPENW #0
  a$="      "+STRING$(62,"*")                      ! PRG-Information
  PRINT lf$+lf$+lf$+lf$+a$+lf$+"      *       DEMO-Programm";
  PRINT " fnr den Hostmode von DG6UL 1990       *"+lf$+a$+lf$+lf$
  FRONTW #0
  FRONTW #5
  OPEN "o",#1,"com1:9600,n,8,1"       ! Schnittstelle wird ge÷ffnet
  PRINT #1,CHR$(24);                  ! TNC-Eingabepuffer leer
  a$=INPUT$(LOC(#1),#1)               ! SER-buffer leer
  PRINT #1,CHR$(27);"e1";cr$;         ! Terminalmode-Befehl ECHO EIN
  PRINT #1,CHR$(27);"a1";cr$;         !                     AUTOLF EIN
  PAUSE 15                            ! kurz warten
  PRINT #1;CHR$(27);"i";cr$;          !                     MYCALL
  PAUSE 15                            ! kurz warten
  WHILE LOC(#1)<>0                    ! warten bis Daten an Schnittstelle
    LINE INPUT #1,a$                  ! Zeile mit MYCALL einlesen
  WEND
  PRINT "* Rufzeichen im TNC: ";a$    ! MYCALL ausgeben
  PRINT
  PRINT #1;CHR$(27);"jhost1";cr$;     ! TNC in Hostmode schalten
  a$=INPUT$(9,#1)                     ! Antwort einlesen
  PRINT a$                            ! Antwort ausgeben
  PRINT "* Hostmode aktiv. TNC wird konfiguriert *"
  b$(1)="T 33"
  b$(2)="U 0"                         ! TNC-Konfigutation im Hostmode
  b$(3)="W 10"
  b$(4)="F 3"
  b$(5)="P 64"
  b$(6)="M UISC"
  b$(7)="Y 4"
  b$(8)="@t2 100"
  FOR x|=1 TO 8
    GOSUB send_to_tnc(0,1,b$(x|))
  NEXT x|
  FOR x|=1 TO 4                       ! Call-Variable leer
    call$(x|)="discon  "
    alt$(x|)=" U0 V0 Disconnected"
  NEXT x|
  GOSUB balken
  PRINT lf$+"OK";STRING$(10,lf$)      ! OK-Meldung: Installation erfolgreich
RETURN
PROCEDURE status
  ' Auswertung der TNC-Antworten:
  ' 0=Erfolg, 1/2=TNC-Meldung, 3=Linkstatus, 4/5=Header, 6/7=Informationen
  ' Ausgabe auf entsprechendens Window
  SELECT status|
  CASE 0                                    ! Erfolg ohne Meldung
  CASE 4,5                                  ! Monitor Header
    a=1                                     ! damit kein 0
    a$=""
    WHILE a<>0                              ! Einlesen bis "0"
      a=INP(#1)
      a$=a$+CHR$(a)
    WEND
    PRINT CHR$(27)+"[33m"+a$+CHR$(27)+"[31m"   ! Ausgabe
  CASE 1,2,3
    a$=" "                                  ! Erfolg Fehler Linkstatus
    b$=""
```

```
  WHILE a$<>CHR$(0)                          ! Lese bis "0"
    a$=CHR$(INP(#1))
    b$=b$+a$
  WEND
  IF status|=3                               ! Wenn Linkstatus
    topbalken$(kanal|)=LEFT$(b$,LEN(b$)-20)  ! Anzeige Connectstatus
    TITLEW #kanal|+5,topbalken$(kanal|),sctitel$
    IF MID$(b$,5,3)="CON"                    ! Bei Con Rufzeichen eintragen
      ruf$=MID$(b$,18,10)
      m=INSTR(ruf$," ")
      LSET call$(VAL(MID$(b$,2,1)))=LEFT$(ruf$,m-1)  ! Rufzeichen eintragen
    ENDIF
    IF MID$(b$,5,3)="DIS" OR MID$(b$,5,6)="LINK F"   ! Bei d oder linkfehler
      call$(VAL(MID$(b$,2,1)))="discon  "            ! Rufzeichen löschen
    ENDIF
    GOSUB balken                                     ! -nderung in Balken
  ENDIF
  IF llflag|=1                        ! Statusflag: Ausgabe der Linkinformation
    llflag|=0                         ! Flag wieder 0
    nbp|=VAL(MID$(b$,7,1))            ! Anzahl nicht best_tigten pakete
    ver|=VAL(MID$(b$,9,2))            ! Anzahl der Versuche
    sta|=VAL(RIGHT$(b$,2))            ! Status
    SELECT sta|
    CASE 0
      sta$="Disconnected"
    CASE 1
      sta$=" Link Setup "
    CASE 2
      sta$="Frame Error"
    CASE 3
      sta$="Disc Request"
    CASE 4
      sta$="InfoTransfer"
    CASE 5
      sta$="Reject Sent "
    CASE 6
      sta$="Wait for ack"
    DEFAULT
      sta$="Busy Failure"
    ENDSELECT
    neu$(kanal|)=" U"+STR$(nbp|)+" V"+STR$(ver|)+" "+sta$
    IF neu$(kanal|)<>alt$(kanal|) AND kanal|>0    ! Nur bei -nderung Balken-
      alt$(kanal|)=neu$(kanal|)                   ! Anzeige erneuern
      TITLEW #kanal|,balken$(kanal|)+neu$(kanal|),sctitel$
    ENDIF
  ELSE
    PRINT CHR$(27)+"[33m";b$+lf$+CHR$(27);"[31m";   ! Ausgabe auf Schirm
  ENDIF
  b$=""
CASE 6,7                               ! Information-Ausgabe
  l=INP(#1)+1
  a$=INPUT$(l,#1)
  '       Routine zur Umwandlung von CR in LF innerhalb eines Strings
  start=1
  DO
    x=INSTR(start,a$,cr$)              ! Suche nach CR
    MID$(a$,x,1)=lf$                   ! austauschen: CR=LF
    start=x+1
  LOOP UNTIL x=0
  IF kanal|=0 AND RIGHT$(a$,1)<>lf$ AND a$<>cr$     !  +LF im Monitor
    a$=a$+lf$
  ENDIF
  PRINT a$;                                         ! INFO-Ausgabe
DEFAULT                                             ! Falls Status falsch
```

```
    PRINT "Falsche TNC-Antwort bei PROCEDURE STATUS: ";status|
    DELAY 3
    STOP                                                ! Programmende
  ENDSELECT
RETURN
PROCEDURE eingabe
  '  Eingaben von Tastatur werden ausgewertet.
  '  Dabei wird verhindert, da_ die beim Korrigieren eingegebenen Backspace-
  '  zeichen gesendet werden. _berlange Eingaben werden nicht abgefangen.
  OPENW #fwin|+5                            ! Eingabewindow ÷ffnen
  IF z$=CHR$(8)                             ! Wenn Eingabe=Backspace
    IF LEN(eingabe$(fwin|))>0               ! Nur wenn Eingabestring > 0
      eingabe$(fwin|)=LEFT$(eingabe$(fwin|),LEN(eingabe$(fwin|))-1)
      PRINT z$;" ";z$;                      ! Letztes Zeichen l÷schen
    ENDIF
  ELSE
    PRINT z$;                               ! Zeichen ausgeben
    eingabe$(fwin|)=eingabe$(fwin|)+z$      ! Eingabestring akualisieren
  ENDIF
  IF z$=cr$                                 ! Wenn RETURN
    PRINT
    t$=eingabe$(fwin|)                      ! Befehlsstring
    IF LEFT$(t$,1)=":"                      ! Wenn Befehl
      GOSUB befehl(fwin|)
    ELSE
      GOSUB send_to_tnc(fwin|,0,t$)         ! Senden
    ENDIF
    eingabe$(fwin|)=""                      ! Eingabestring leer
  ENDIF
RETURN
PROCEDURE send_to_tnc(k|,mode|,text$)
  '  k| =Kanalnummer;  mode| =Modus;   text$ =zu sendender Text oder Befehl
  PRINT #1;CHR$(k|);CHR$(mode|);CHR$(LEN(text$)-1);text$;
  kanal|=INP(#1)
  status|=INP(#1)
  IF status|<>0                  ! Wenn kein Erfolg
    OPENW #kanal|
    GOSUB status
  ENDIF
RETURN
PROCEDURE befehl(kn|)
  ' kn|=Kanalnummer
  ' Hier wertet das Programm die Befehle aus und unterscheidet zwischen TNC-
  ' und Programmbefehlen.
  ' Hier ist nur ein einziger Programmbefehl: x zum Beenden der Software
  IF t$=":x"+cr$                            ! (x) Prg-Ende
    GOSUB ende
  ELSE                                      ! wenn TNC-Befehl dann...
    t$=MID$(t$,2,LEN(t$)-2)                 ! Befehl an TNC
    x=ASC(LEFT$(t$,1))
    IF (x<91 AND x>63) OR x=99 OR t$="d"    ! Erm÷glicht Kleinschreiben der
      GOSUB send_to_tnc(kn|,1,t$)           ! connect und discon Befehle
    ELSE                                    ! Wenn Gro_buchstaben
      PRINT "Eingabefehler"
      t$=""
    ENDIF
  ENDIF
RETURN
PROCEDURE funktionstasten
  '                  Erm÷glicht umschalten der Windows
  'u|=ASC(MID$(z$,2,1))           ! liest Funktionstaste
  SELECT u|
  CASE 48                         ! Port 1
    FRONTW #1
```

```
    FRONTW #6
    fwin|=1
  CASE 49                          ! Port 2
    FRONTW #2
    FRONTW #7
    fwin|=2
  CASE 50                          ! Port 3
    FRONTW #3
    FRONTW #8
    fwin|=3
  CASE 51                          ! Port 4
    FRONTW #4
    FRONTW #9
    fwin|=4
  CASE 57                          ! Monitor
    FRONTW #0
    FRONTW #5
    fwin|=0
  ENDSELECT
RETURN
PROCEDURE ende
  '        Windows schlie_en, TNC konfigurieren und in Terminalmode
  b$(1)="M N"              ! Monitorbetriebsart OFF
  b$(2)="U 1"              ! OFFLINE-Text in TNC einschalten
  FOR x|=1 TO 2
    GOSUB send_to_tnc(0,1,b$(x|))    ! TNC-Befehl
  NEXT x|
  PRINT "TNC in Terminalmode"
  t$="jhost 0"
  GOSUB send_to_tnc(0,1,t$)          ! TNC in Terminalmode
  PRINT "OK"
  FOR x=0 TO 9                       ! Windows schlie_en
    CLOSEW #x
    CLOSE #x
  NEXT x
  EDIT                               ! Ende
RETURN
PROCEDURE balken
  '                    Aktualisieren der  Balkenanzeige
  FOR x|=1 TO 4
    balken$(x|)="1:"+call$(1)+" 2:"+call$(2)+" 3:"+call$(3)+" 4:"+call$(4)
    TITLEW #(x|),balken$(x|)+alt$(x|),sctitel$
  NEXT x|
RETURN
```

PS: Das Listing ist nur ein Beispiel wie man ein Terminalprogamm für den Hostmode der WA8DED-Software schreiben kann. Man kann sicher noch einiges verbessern.

Viel Spass beim ausprobieren, vy 73 de Holger (DG6UL @ DB0IE)

Msg# 162101 To: TNC2 @ALLE From: DB2OS Date: 05Sep91/0339

Subject: TF23.TXT Beschreibung

Bulletin ID: 04910DDK0MAV

Path: DB0CZ!OE9XPI!HB9EAS!DB0GE!DB0IZ!DK0MWX!DB0EAM!DB0BQ!DB0SHG!DB0HOL!DK0MAV

de DB2OS @ DK0MAV

** Diese Beschreibung steht auch im TF23.LZH Eprom-Archiv **

```
The Firmware Version 2.3 DAMA (10 Channel)
   Copyright by NORD><LINK, 03-Sep-91
      Free for non-commercial usage
```

```
Checksum (EB40) = EB40
```

So meldet sich die neue TheFirmware im TERMINAL-MODE auf dem Bildschirm, wenn das mitgelieferte EPROM TF2.3 ordnungsgemäss im TNC-2 (bzw. Einem kompatiblen TNC) installiert wurde. Es handelt sich um die neueste "offizielle" TheFirmware von NORD><LINK.

Die Version TF2.3 ist nun (hoffentlich) fehlerfrei und es wurden gegenüber allen bisherigen Versionen erhebliche Verbesserungen vorgenommen und die bekannten Fehler behoben.

Die neue TheFirmware ist für Betrieb über die neuen "DAMA"-Einstiege der TheNetNode-Software unbedingt erforderlich, hat aber auch eine Reihe von Verbesserungen und Vorteilen bei Betrieb über "normale" Digipeater und im "normalen Direkt-QSO".

Neue Befehle (seit TF2.1)

ESC B [<n>] Zeitspanne in Sekunden, nach der der DAMA Modus abgeschaltet wird, falls kein Poll vom DAMA-Master empfangen wurde. B 0 schaltet den DAMA-Modus generell ab. Dies ist aber nicht erforderlich, da DAMA sowieso automatisch erkannt bzw. abgeschaltet wird. Die Anzeige erfolgt in der Form: "Anfangswert (aktueller Wert)" Beispiel: "120 (93)" Default: 120

ESC F [<n>] Anfangswert für SRTT Berechnung, 10 ms Schritte. (ACHTUNG: Nicht mehr kompatibel mit FRACK bisherigen TheFirmware Versionen!) Default: 700

ESC P [<n>] P-Persistence Einstellung (0..255) Ohne Parameter Anzeige der aktuellen (255=DAMA) und der Vorgabe bei Nicht-DAMA Betrieb. Beispiel: "64 (64)" oder "255 (64)" bei DAMA. Default: 64

ESC W [<n>] Slot-Time (Zeitschlitz) in Millisekunden. Ohne Parameter Anzeige der aktuellen Einstellung (0=DAMA) und der Vorgabe bei Nicht-DAMA Betrieb.Beispiel: "10 (10)" oder "0 (10)" bei DAMA. Default: 10

ESC Y [<n>] Eingabe der maximal zulässigen Kanäle, bis eine anrufende Station "busy" bekommt. Ausgabe in der Form "maximale Anzahl Kanäle (belegte Kanäle)" (funktioniert nur, wenn überall gleiche SSID wie im Monitor-Kanal S0 verwendet wird). Beispiel: "4 (0)" Default: 4

ESC @A1 [<n>] SRTT Berechnung: Wert für a1 eingeben/anzeigen. Default: 7

ESC @A2 [<n>] SRTT Berechnung: Wert für a2 eingeben/anzeigen. Default: 15

ESC @A3 [<n>] SRTT Berechnung: Wert für a3 eingeben/anzeigen. Default: 2

ESC @I [<n>] Wert für max. IPOLL-Framelänge eingeben, bzw. anzeigen. Default: 60

ESC @M [0|1] 7.te Bit im Terminal-Mode zulassen, z.B. für Umlaute. 1=Ja, 0=Nein. Default: 1

Kurzzusammenfassung einiger Änderungen:

1. DAMA-Modus

Ein DAMA-Master wird automatisch erkannt, was dazu führt, dass der TNC nur noch dann sendet, wenn er vom Master zur Sendung aufgefordert wird. Dann allerings mit allen zur Sendung anliegenden Paketen von allen connecteten Kanälen, auch von solchen, die nicht mit einem DAMA-Master verbunden sind.

Mit ESC B kann eine Zeitspanne in Sekunden eingegeben werden, nach der der TNC den DAMA-Modus wieder abschaltet, falls innerhalb dieser Zeitspanne kein Poll vom DAMA-Master gekommen ist. Default: 120 sec.

Wird ein DAMA-Slave (der Benutzer) gleichzeitig als Digipeater von einer anderen Station benutzt, werden dessen Pakete sofort weitergeleitet.

Die vorliegende Firmware kann gleichermassen bei DAMA-Digipeatern, als auch bei NICHT-DAMA-Digis eingesetzt werden. Ein Digipeater, der als DAMA-Master arbeitet wird automatisch erkannt und die Firmware schaltet sich in diesem Fall automatisch in den DAMA-Modus. Erkennbar ist dies bei eingeschaltetem Monitor an den Zeichen "[DAMA]" hinter jedem empfangenem Frame vom DAMA-Master.

Der DAMA-Modus ist in cqDL 4/89 Seite 230ff und in den Mailboxen unter der Rubrik AX25 beschrieben. Bei einigen Digipeatern finden bereits DAMA-Betrieb statt, hierzu gehören u.a. die Netzknoten H:DB0FD, HR:DB0KH, HHOST:DB0HHO.

Nach den ersten sehr erfolgreichen Versuchen werden demnächst viele "TheNetNode"-Knoten die neue Protokoll-Variante einsetzen.

2. Bestätigungszeitgeber T1 (Ackknowledgement Timer, FRACK)

Eine feste FRACK-Einstellung gibt es nicht mehr, stattdessen wird bei jedem QSO die Zeit zwischen Aussenden eines Info-Frames und Empfangen der zugehörigen Bestätigung gemessen und zur Berechnung des Startwertes des Bestätigungstimers T1 herangezogen.

Die gemessene Zeit RTT (round trip time) wird, um zu grosse Schwankungen zu vermeiden, geglättet nach den Formeln:

- bei steigendem RTT: SRTT' = (a1 x SRTT + RTT) / (a1 + 1)

- bei fallendem RTT: SRTT' = (a1 x SRTT + RTT) / (a2 + 1)

Dabei ist SRTT (smoothed round trip time) der aus der letzten Messung ermittelte Wert für geglättete Zeitmessung, RTT die soeben gemessene Zeit, a1 und a2 einstellbare Parameter (Default: 7 und 15) und SRTT' der neue Wert für SRTT.

Der Timer T1 ergibt sich aus SRTT nach:

T1 = a3 x SRTT

wobei a3 einstellbar ist (Default: 2).

Vor Beginn eines QSOs muss SRTT initialisiert werden, da ja noch keine Messung stattgefunden hat. Dies geschieht mit dem Wert, der mit ESC F eingegeben werden kann (10 ms - Schritte, Default: 700). Die RTT-Berechnung beginnt erst nach Aufbau der Verbindung, d.h. nachdem das UA der Gegenstation empfangen worden ist. Bei QSOs über mehrere Digipeater wird die Zeit zwischen den einzelnen SABMs berechnet nach

T1 = (2 x "Anzahl Digis" + 1) x IRTT

A1, a2 und a3 können über ESC @A1, ESC @A2 und ESC @A3 eingestellt werden.

Dies war notwendig, weil z.B. Flexnet-Digipeater das UA solange zurückhalten, bis die Verbindung über die ganze Digi-Strecke aufgebaut ist.

Sofort nach Verbindungsaufbau, also nach Empfang von UA der Gegenstation, setzt die RTT-Berechnung ein, die zu einem sich der Verbindung anpassenden T1-Timers führt.

[Seit TF2.3 wird mit dem RTT/T1 tatsächlich die Zeit vom Aussenden des letzten I-Frame bis zum Empfang der Bestätigung gemessen, hier war noch ein folgenschwerer Fehler in den Versionen TF2.2]

3. DWAIT nach DL4YBG

Vor jeder Aussendung wird grundsätzlich 1-mal Slottime abgewartet, bevor der P-Persistance-Algorithmus eingeleitet wird. Dies gilt sowohl im KISS, als auch im NICHT-DAMA-TheFirmware-Modus. Bei DAMA-Betrieb wird automatisch Slottime auf 0 und P auf 255 gesetzt, damit der TNC ohne Verzögerung sendet. Die Wahrscheinlichkeit von Kollisionen wird durch diese Modifikation spürbar reduziert.

4. Polls mit I-Frames nach DK6PX

Bei kurzen Paketen, die nicht beim Empfänger angekommen sind, kann es sinnvoll sein, anstelle mit RR-Frames eine Bestätigung anzumahnen, dies mit dem nicht bestätigten I-Frame mit gesetzem Poll-Bit zu tun. Dazu kann mit ESC @I die maximale Länge des I-Frames eingestellt werden, bei dem ein IPOLL zu senden ist. ESC @I 0 schaltet IPOLL generell ab.

5. Dynamisches MAXFRAME nach DK6PX

Je nach Länge der I-Frames wird MAXFRAME automatisch erhöht, wenn z.B. nur kurze Info-Frames vorliegen. Bei MAXFRAME 1 wird maximal 1 Frame mit 256 Zeichen gesendet. Liegen im TNC Frames mit je 128 Zeichen an, dann wird MAXFRAME 2 benutzt. MAXFRAME 4 wird automatisch benutzt, wenn z.B. jedes einzelne Frame nicht länger als 64 Zeichen ist, usw.

6. DCD/PTT-Verklemmung

Ein Fehler in einer Interruptroutine konnte unter bestimmten Umständenzu einer "Verklemmung" des TNC's führen. Diese Fehler existierte in allen bisherigen TF-Versionen und wurde nach einem Hinweis von Y51GE behoben. Der KISS-Mode war hiervon jedoch nicht betroffen. Ein noch schwerwiegenderer Fehler in den bisherigen TF2.2 DAMA-Versionen (bekannt als DCD-Hänger) ist ebenfalls behoben worden.

7. DAMA-Parameter

Nach Verlassen der DAMA-Betriebsart werden alle Parameter (P,W,B,@T2) wieder automatisch auf Ihre ursprünglichen Einstellungen zurückgestellt. Während dem DAMA-Betrieb wirkt sich die Neu-Eingabe von P und W nicht direkt auf diese Parameter aus. Dies ist wichtig, wenn z.B. das Terminal Programm während einer

DAMA-Verbindung nach einem Neu-Start die Parameter überschreibt. Diese Parameter werden dann erst nach Verlassen der DAMA-Betriebsart aktiv.

8. Disconnect nach DL1MEN

Wird ein Verbindungsaufbau während dem Link Setup mit ESC D abgebrochen, wird automatisch ein DISC gesendet. Dies soll unnötige Aussendungen verhindern, falls nur der eigene TNC die Antworten der Gegenstation nicht gehört hat.

9. Heard-Liste

Die eingebaute Heard-Liste zeigt nun nicht mehr die zuerst gehörten Rufzeichen, sondern die zuletzt gehörten Rufzeichen. Die ältesten Einträge werden dann überschrieben.

10. Software-Autoren

Die Nennung der Autoren im Signon der Software ist entfallen, da inzwischen einige Co-Autoren erhebliche Verbesserungen an der Software durchgeführt und mithin sehr viel Zeit in die Weiterentwicklung investiert haben, sodass eine lange Rufzeichen-Liste den Rahmen sprengen würde.

Urvater der NORD><LINK-TheFirmware ist Michael, DC4OX. Die DAMA-Implementation stammt im wesentlichen von Frank, DL8ZAW. Verbesserungen wurden vorgenommen von Georg DF2AU, Peter DB2OS, DK6PX, DL9HCJ, DL1MEN, DF7ZE, u.a..Ein besonderer Dank auch an DL1BHO für seine unendliche Geduld bei den Versuchen zum Aufspüren der letzten Softwarefehler.

Vy 73s de Peter DB2OS

Msg# 176619 To: TNC2 @ALLE From: DB8UY Date: 21Oct91/0539

Subject: TF21c.DOC (Beschreibung TheFirmware)

Bulletin ID: 20A104DB0BOX

Path: DB0AAA!DB0MWE!DB0AAB!DB0LNA!DB0RGB!DB0BOX

de DB8UY @ DB0BOX

```
T H E  F I R M W A R E
======================

    AX.25 Version 2
Multi Channel TNC Firmware
     Version 2.1c


Diese  Firmware unterstützt das vollständige AX.25 Protokoll,  Version
2.0,  der Link Ebene,  wie es in der ARRL Spezifikation vom Oktober 1984
beschrieben  ist.  Die  alte Version 1.x wird  auch  verarbeitet.  Diese
Software kann mehrere Verbindungen gleichzeitig in beiden  Protokollver-
sionen bedienen. Das vorliegende Eprom  wurden für maximal vier gleich-
zeitige Verbindungen assembliert.  Es ist aber jede beliebige Zahl durch
Veränderung eines Parameters im Quelltext möglich.
Zusätzlich ist  in  dem  Eprom der KISS-Modus aus dem bekannten TCP/IP-
Paket von  KA9Q implementiert.  Die Parameter der Hostmode-Firmware wer-
den  n i c h t  verändert  (KISS benutzt  eigene  andere  Parameter und
überschreibt nicht  die Firmware-Variablen. Nach  einem Reset sind alle
Parameter unverändert (KISS wurde in  ein linkfähiges Codesegement um-
geschrieben). Das  KISS-Kommando 255, welches  wie  ein  Reset (Aus- und
Einschalten des TNC) wirkt  und eine  programmgesteuerte  Rückkehr  zur
Firmware  ermöglicht, ist  implementiert. Die  Firmware-Uhr  läuft bei
KISS-Betrieb nicht weiter.


Nach  einem  Reset  wird  eine  Prüfsumme  errechnet, die in der Start-
meldung ausgegeben wird. Sie muss mit der in Klammern aufgeführten Zahl
übereinstimmen  (auch  bei  veränderten  Parametern im Eprom). Stimmen
die Zahlen  nicht  überein,  so  muss ein Fehler beim Programmieren des
```

Eprom's passiert sein.

Das TNC-Rufzeichen ist leer und kann mit dem 'I'-Befehl gesetzt werden. Ebenso können alle anderen Standard Parameter geändert werden. Die neuen Werte werden dann im RAM gesichert.

Befehle und Informationen werden dem TNC als Zeilen übergeben. Jede Zeile darf bis zu 256 Zeichen lang sein, wobei das Schlusszeichen <CR> mitzählt. Wenn das 256. Zeichen kein <CR> ist, wird es überlesen und ein <BEL> zum Terminal geschickt. Einzelne Zeichen können mit <BS> = 08 HEX oder <DEL> = 7F HEX gelöscht werden. Mit <CTL-U> oder <CTL-X> kann die gesamte Zeile gelöscht werden. Ein <CTL-R> bringt die Schreibmarke ohne Löschen an den Zeilenanfang, so dass eingelaufene Meldungen angezeigt werden können. Mit einem zweiten <CTL-R> kann dann wieder an das Ende der angefangenen Zeile gegangen werden, um die Eingabe fortzusetzen. Zwischen diesen beiden <CTL-R> werden alle Eingaben ausser <X-OFF> und <X-ON> ignoriert. <BEL> Zeichen werden als <BEL> an das Terminal weitergereicht, sowohl bei der Eingabe als auch beim Löschen. Zeilen, die mit <ESC> beginnen (wird als '*' ausgegeben) werden als Befehlszeilen interpretiert. Wenn ein Befehl ohne Parameter eingegeben wird, wird der augenblickliche Wert dieses Parameters angezeigt. Alle Zeilen, die nicht mit <ESC> beginnen, werden als Information gesendet.

Die Firmware stellt dem Benutzer fünf logische TNC Kanäle bereit, die von 0 bis 4 numeriert werden. Das Terminal ist immer einem dieser Kanäle logisch zugeordnet, die Auswahl erfolgt mit dem 'S'-Befehl. Kanal 0 ist reserviert für nicht protokollierte Sendungen (CQ, Bake). Der Leitweg für Kanal 0 wird wie bei den anderen Kanälen mit dem 'C' Befehl gewählt. Die Kanäle 1-4 senden im Terminal-Modus auch an 'CQ', solange sie nicht mit anderen Stationen verbunden sind. Connect Versuche können auf jedem beliebigen, derzeit nicht belegten Kanal gesendet werden. Einlaufende Connect Versuche anderer Stationen werden auf den ersten freien Kanal gelegt, sofern dadurch nicht die durch den 'Y' Befehl gegebene maximale Zahl gleichzeitiger Verbindungen überschritten wird. Information, die auf einem belegten, derzeit nicht mit dem Terminal verbundenen Kanal einläuft, wird gespeichert, bis dieser Kanal auf das Terminal geschaltet wird. Mit dem 'L'-Befehl kann leicht festgestellt werden, ob auf anderen Kanälen Informationen eingelaufen sind und auf den Abruf warten. Informationen werden nur auf dem gerade angewählten Kanal gesendet. Nach einem Disconnect werden eingegangene Informationen bis zum Abruf im TNC-RAM gespeichert. Wenn der Leitweg oder das Ziel gewechselt werden sollen, während eine Verbindung gerade aufgebaut wird oder besteht, muss nicht erst ein Disconnect ausgelöst werden, es genügt, mit einem neuen 'C'-Befehl die Verbindung neu aufzubauen. Es geht keine Information dadurch verloren. Es ist aber nicht zulässig, die gleiche Station auf mehr als einem Kanal zur gleichen Zeit zu wählen.

Die Protokollversion, mit der die Verbindung aufgebaut wird, wird mit dem 'V'-Befehl gewählt, wird aber automatisch auf die bei der Gegenstation verwendete Version geändert. Version 2 ist besser für Netzwerke geeignet und erlaubt eine bessere Kanalausnutzung, besonders unter erschwerten Bedingungen. Version 2 sollte daher möglichst immer gewählt werden. Bei Benutzung der Version 2 des Protokolls wird ein Timer gestartet, sobald keine Information mehr zu übertragen ist. Bei einer Pause von mehr als drei Minuten wird der TNC der Gegenstation abgefragt, um sicherzustellen, dass die Verbindung noch existiert. Wenn die mit dem 'N'-Befehl vorgegebene Zahl an Versuchen erfolglos geblieben ist, wird versucht, die Verbindung neu aufzubauen. Hierdurch wird auch der Fall abgedeckt, dass jemand eine Verbindung aufbaut und dann ohne Disconnect wieder verschwindet. Es ist nicht zulässig die Protokollversion, während eine Verbindung besteht, zu ändern.

Für die Befehle 'F', 'I', 'N', 'O' und 'V' werden die Parameter für jeden Kanal getrennt gespeichert. Der Wert des Kanals 0 wird im RAM gespeichert und wird zur Initialisierung der Kanäle 1-4 nach Reset und nach Trennen der Verbindung benutzt. Man kann daher vor oder während einer Verbindung diese Werte ändern und erhält automatisch am Ende der Verbindung die Standard Werte wieder zurück. Mit dem 'D'-Befehl kann ein freier Kanal auch wieder initialisiert werden.

Mit dem 'M'-Befehl kann die Kanalaktivität beobachtet werden. Der Parameter für diesen Befehl entscheidet, welche Pakettypen beobachtet werden sollen. Mehrere Parameter können gleichzeitig angegeben werden.

Parameter	Paket-Typ
N	nichts
I	Informationen
U	unprotokolliert
S	Status (Kontroll Pakete)
C	auch beobachten, wenn eine Verbindung besteht
+	nur Pakete von/zu bestimmten Stationen (maximal 8)
-	keine Pakete von/zu bestimmten Stationen (maximal 8)

Die kombinierte Benutzung der '+' und '-' Parameter ist nicht zulässig. Sie müssen als letzter Parameter vor den Rufzeichen eingegeben werden. Die Eingabe von '+' oder '-' ohne Rufzeichen löscht die aktuelle Liste.

Ein '*', der in einer Rufzeichenliste erscheint, kennzeichnet die Station, die das Paket gesendet hat. Der Pakettyp kann aus der folgenden Liste entnommen werden.

Name	Beschreibung
RRa	bereit zum Empfang
RNRa	nicht bereit zu Empfang
REJa	Paket nicht akzeptiert
UI	unprotokollierte Information (an alle)
DM	Verbindung besteht nicht
SABM	Aufforderung zum Verbindungsaufbau
DISC	Aufforderung zum Trennen der Verbindung
UA	Bestätigung eines nicht numerierten Paketes
FRMR	Protokoll Fehler
Iab	Information
?ccH	nicht definierter Typ

Hierbei bedeutet:
a = Nummer des nächsten erwarteten Paketes
b = Nummer dieses Paketes
cc = Hexadezimal Zahl

Ein weiteres Zeichen gibt die verwendete Protokollversion und das Poll/Final sowie das Command/Response Bit an:

<leer> = Protokollversion 1 ohne P/F Bit
! = P/F Bit in Protokollversion 1
^ = Kommando Paket in Protokollversion 2 ohne Poll Bit

+ = Kommando Paket in Protokollversion 2 mit Poll Bit
- = Antwort Paket in Protokollversion 2 mit Final Bit
v = Antwort Paket in Protokollversion 2 ohne Final Bit

Das Protokoll Identifikationsfeld wird in hexadezimaler Form ausgegeben.

Mit dem 'U'-Befehl kann ein Connect-Text ein- bzw. ausgeschaltet werden. Es ist möglich, einen frei wählbaren Text an die anrufende Station zu senden und dann der rufenden Station die Möglichkeit zu geben, eine kurze Nachricht zu hinterlassen. Dies kann auf allen Kanälen gleichzei-

tig geschehen und beeinflusst in keiner Weise die Möglichkeit des Bedieners, Informationen in eine bestehende Verbindung einzugeben oder seinerseits Verbindungen aufzubauen. Wenn der Ctext eingeschaltet ist, werden alle Zustandsmeldungen des betreffenden Kanals gespeichert und erst auf das Terminal gegeben, wenn dieser Kanal angewählt wird. Statusmeldungen werden daher dann in zeitlich richtiger Reihenfolge zusammen mit den eingelaufenen Informationen ausgegeben. Ausserdem wird der mit dem 'U'-Befehl eingegebene Text an jede anrufende Station gesendet. Wenn Kanal 0 als letztes gewählt wurde können Stationen auf den Kanelen 1-4 anrufen und Meldungen hinterlassen, wobei die Zahl der maximal gleichzeitig möglichen Verbindungen mit 'Y' gewählt wurde. Mit dem 'L'-Befehl kann festgestellt werden, welcher Kanal Informationen enthält. Wenn dann dieser Kanal gewählt wird, werden alle gespeicherten Informationen und Meldungen dieses Kanals auf das Terminal gegeben. Wenn mit dem 'Z'-Befehl X-ON/X-OFF-Protokoll gewählt wurde, kann die Ausgabe mit <CTL-S> gestopt und mit <CTL-Q> wieder aufgenommen werden.

Mit dem 'H'-Befehl wird die eingeaute Heardliste aktiviert. Diese Liste unterscheidet sich aber von den bekannten Heardlisten anderer Programme oder TNC-Betriebssoftware in einigen Punkten. Zunächst ist es keine Liste der Art "die letzten 20 gehörten Stationen". Diese Art Liste hat sich bei starkem Betriebsaufkommen als nicht sehr sinnvoll erwiesen. Die bei der Firmware implementierte Liste ist eine dynamische Liste. Die Anzahl der Calls, die sie aufnehmen kann, wird nur durch die Anzahl der freien Buffer bestimmt. Bei Bestückung mit 32k RAM liegt die maximal mögliche Anzahl bei über 600 Rufzeichen. Das ständige Wachsen der Liste bis an die Buffergrenze kann durch Setzen eines Parameters begrenzt werden. Für jedes im Absenderfeld eines AX.25-Frames gehörte Rufzeichen wird in der Liste gespeichert :

- Rufzeichen und SSID
- Datum und Zeit wann zuerst gehört
- Datum und Zeit wann zuletzt gehört
- Anzahl der gehörten I-Frames
- Anzahl der gehörten RR-Frames
- Anzahl der gehörten REJ-Frames
- Anzahl der gehörten RNR-Frames

Die Speicherung des Digipeaterweges ist im Zeitalter von TheNet nicht sinnvoll. Durch die Beobachtung der Frame-Anzahlen können gewisse sehr einfache statistische Betrachtungen gewonnen werden, z.B. was effektive Aktivität oder Qualität von Aussendungen angeht(Verhältnis I/RR/REJ).

Da für jedes gehörte Paket die Liste (linear, weil dynamisch angelegt) durchlaufen werden muss, setzt das Führen einer grossen Liste die Geschwindigkeit der Firmware herab. Mit 'H 0' stellt man das automatische Updaten der Heardliste ab. Die Liste bleibt im Speicher, aber die empfangenen Pakete werden nicht für die Liste untersucht.

Der 'H'-Befehl läuft auch im Hostmodus einwandfrei. Bei der Ausgabe der Heardliste gibt es aber einiges zu beachten. Die erste Zeile (H, Anzahl gehörte, maximal) wird mit dem Hostmodecode 1 (Success, Message follows null terminated) zurückgegeben. Die eigentliche Liste wird dann auf G/G0 auf dem Kanal 0 wie Monitorheader zurückgegeben (Code 5 Monitorheader, null terminated, Info follows). Die letzte Zeile wird mit dem Code 4 (Monitorheader, null terminiert) ausgegeben. Dieses Handling der Ausgabe über mehrere Zeilen funktioniert mit den meisten Terminalprogrammen für den Hostmodus (z.B. TINA) ohne jede Änderung in diesen Programmen.

Die Heardliste ist resident, d.h. sie bleibt bei einem Reset oder einem Abschalten bei Pufferung des RAM mit Batterie erhalten.

Die Firmware 2.1c hat eine 24-Stunden-Uhr und einen Kalender per Software eingebaut. Man kann wahlweise alle Statusmeldungen (also CONNECT REQUEST fm, CONNECTED to, usw.) und Monitormeldungen (Header gemo-

nitorter Pakete) mit einem Datum/Uhrzeit-Stamp versehen lassen. Mit 'K 0' schaltet man die Ausgabe dieses Stamps ab. Trotzdem werden alle Meldungen/Header mit einem Stamp versehen, ein Einschalten des Stamps wirkt also auch richtig bei längere Zeit im Buffer stehenden Meldungen/ Headern mit abgeschaltetem Stamp. Mit 'K 1' schaltet man die Stampausgabe für Statusmeldungen ein, mit 'K 2' die Stampausgabe für Statusmeldungen und Monitorheader. Eingabe von 'K' zeigt die aktuelle Einstellung gefolgt vom Datum und der Uhrzeit. Je nachdem wie man das Datum gesetzt hat, wird es bei Stamps entweder im europäischen oder amerikanischen Format ausgegeben. Das Datum und die Uhrzeit werden bei QRES oder einem Kaltstart gelöscht. Bei einem Warmstart läuft die Zeit ab dem Zeitpunkt weiter, der beim Ausschalten des TNC vorlag. Anhand einer nachgehenden Uhr kann man so leicht feststellen, ob sich ein Stromausfall während des Betriebs ereignet hat. Tests haben ergeben, dass die Uhr ohne Korrekturen sehr genau läuft.

BEFEHLS-ÜBERSICHT

Mit [] sind mögliche, aber nicht notwendige Parameter markiert

BEFEHL	PARAMETER	BESCHREIBUNG
A (1)	0	Auto Linefeed ausgeschaltet
	1	Auto linefeed eingeschaltet
C	Rufz1 [Rufz2 ... Rufz9]	Connect-Weg (in Kanal 0: unproto)
D		Verbindung auflösen
E (1)	0	kein Echo für eingegebene Zeichen
	1	Echo für eingegebene Zeichen
* F (4)	1...15	Wartezeit für Antwort (Sekunden)
G	[0]	Information im Host-Modus holen
	[1]	Status im Host-Modus holen
H (0)		Heardliste anzeigen
	0	Heardlisten-Update ausschalten
	1	Heardlisten-Update einschalten
	2	Heardliste löschen
	n	Maximalanzahl Calls in Heardliste setzen
* I	Rufzeichen	eigenes Rufzeichen
JHOST (0)	0	Terminal-Modus eingeschaltet
	1	Host-Modus eingeschaltet
K (0)		Stamp und Datum/Zeit anzeigen
	0	Stamp abschalten
	1	Stamp Statusmeldungen einschalten
	2	Stamp Status- und Monitormeldungen einschalten
	TT.MM.JJ	Datum setzen, europäische Form
	MM/DD/YY	Datum setzen, amerikanische Form
	HH:MM:SS	Uhrzeit setzen
L	[0...4]	Statusanzeige für die Kanäle
M (IU)	NIUSC+-	Monitor-Betriebsart
* N (10)	0...255	Anzahl der Versuche (0 = unendlich oft)
* O (4)	1...7	Anzahl der unbestätigten Pakete

```
  P (64)          0...255         P-Persistenz Wert

  QRES                            Neu-Start der Firmware

  R (1)             0             Digipeater ausgeschaltet
                    1             Digipeater eingeschaltet

  S (1)           0...4           Kanal-Nummer (0 = unproto)

  T (30)          0...127         Wartezeit von PTT ein bis Daten (10ms)

  U (0)    0      [Text]          kein unbeaufsichtigter Betrieb
           1      [Text]          unbeaufsichtigter Betrieb

* V (2)             1            Protokoll Version 1
                    2            Protokoll Version 2

  W (10)          0...127         Zeitschlitz für P-Persistenz (10 ms)

  X (1)             0             PTT für Sender unterdrückt
                    1             PTT für Sender freigegeben

  Y (1)           0...4           Maximale Anzahl von Verbindungen

  Z (3)             0             Flow  Aus    , Xon/off  Aus
                    1             Flow  Ein    , Xon/off  Aus
                    2             Flow  Aus    , Xon/off  Ein
                    3             Flow  Ein    , Xon/off  Ein

  @  B                            Zeigt Anzahl der freien Puffer

  @  D (0)          0             Full duplex ausgeschaltet
                    1             Full duplex eingeschaltet

  @  K                            TNC in Kiss-Modus schalten

  @  S                            Momentaner Link-Status

  @  T2 (100)     0...255         Timer T2 (10ms)

  @  T3 (18000)   0...32767       Timer T3 (10ms)

  @  V  (0)         0             Rufzeichencheck abgeschaltet
                    1             Rufzeichencheck eingeschaltet
```

Die Grundeinstellungen (aus dem Eprom) stehen in runden Klammern. Mit '*' sind die Befehle markiert, die für jeden Kanal getrennt eingestellt werden können (bei RESET werden aber die gespeicherten Werte von Kanal 0 für alle anderen Kanäle übernommen).

BESCHREIBUNG DER BEFEHLE

Mit dem 'A'-Befehl kann das automatische Einfügen von LINEFEED-Zeichen nach einem CARRIAGE RETURN zum Terminal ein- bzw. ausgeschaltet werden (AUTOLF).

Der 'C'-Befehl wird für den Aufbau einer Verbindung (Connect) benötigt. Man beachte, dass kein 'v' oder 'via' zwischen der Empfängeradresse und den Digipeater-Rufzeichen erforderlich ist. Während einer bestehenden Verbindung ist es mit dem 'C'-Befehl möglich, die Digipeater-Rufzeichen und somit den Connect-Weg zu ändern. Ein Ändern der Empfängeradresse ist nicht möglich. Weiterhin ist es nicht zulässig ein und die selbe Station in mehr als einem Kanal zu connecten. Ein Connect-Befehl der auf Kanal 0 ausgeführt, wird setzt den Weg für UI-

Pakete.

Zum Trennen (Disconnect) einer Verbindung wird der 'D'-Befehl benutzt. Sind bei der Eingabe des 'D'-Befehls noch nicht alle Informationen ausgesendet bzw. bestätigt, dann wird der Disconnect erst nach Eingang der Bestätigung für das letzte Informationspaket ausgeführt. Durch Wiederholung des 'D'-Befehls kann dieser Vorgang abgebrochen werden. Wird der 'D'-Befehl während des Aufbaus einer Verbindung (Connect-Versuch) oder des Beendens (Disconnect) einer Verbindung eingegeben, dann kehrt der TNC sofort in den Disconnect-Zustand zurück. Wird der 'D'-Befehl im Disconnected-Zustand eingegeben, dann werden alle Parameter des gerade in Gebrauch befindlichen Kanals mit den Parameter vom Kanal 0 initialisiert.

Mit dem 'E'-Befehl wird das ECHO von Eingabe-Zeichen (Daten oder Befehle) zum Terminal ein- bzw. ausgeschaltet.

Mit dem 'F'-Befehl wird die Wartezeit für Wiederholungen (FRACK) eingestellt . Ist nach Ablauf dieser Zeit das Paket noch nicht bestätigt, so wird es wiederholt. Die Formel lautet:

Zeit (Sekunden) = FrameAck * (2 * Anzahl der Digipeater + 1)

Das FRACK-Intervall kann für jeden der 4 Connect-Kanäle jeweils getrennt eingestellt werden. Nach einem RESET oder Disconnect wird aber immer der Parameter von Kanal 0 übernommen.

Mit dem 'G'-Befehl können virtuelle Kanäle des TNC im Host-Modus abgefragt werden. Wird kein Parameter spezifiziert, dann wird die chronologisch nächste Mitteilung (Information oder Linkstatus) ausgegeben, vorausgesetzt, dass überhaupt eine Mitteilung bereit steht. Im Terminal-Modus wird dieser Befehl nicht erkannt und es kommt eine Fehlermeldung. Mehr darüber im Kapitel über den Host-Modus.

Mit dem 'H'-Befehl wird die eingebaute Heardliste abgefragt oder Parameter für die Heardliste übergeben. Mit 'H 0' stellt man das automatische Updaten der Heardliste ab. Die Liste bleibt im Speicher, aber die empfangenen Pakete werden nicht für die Liste untersucht. Mit 'H 1' wird das automatische Updaten der Heardliste eingeschaltet. Mit 'H 2' wird die Heardliste gelöscht und der benötigte Buffer freigegeben. Mit 'H n', n eine Zahl grösser 2, wird die maximale Anzahl der Einträge gesetzt. Mit 'H' ohne Parameter wird die Heardliste ausgegeben. Zunächst wird eine Zeile ausgegeben, die angibt, ob die Heardliste ein- oder ausgeschaltet (1 oder 0) ist, die Anzahl Calls in der Liste, die maximale Anzahl möglicher Calls in der Heardliste. Dann folgt die Liste, für jedes Call eine Zeile, alphabetisch sortiert. Die erste Zeile kommt immer, die Liste nur auf Kanal 0. Die Ausgabe der Liste kann im Terminalmodus durch Eingabe eines beliebigen Zeichens abgebrochen werden (abgesehen von XON/XOFF zur Ausgabesteuerung bei eingeschaltetem Flow), im Hostmodus durch Senden eines Paketes an Kanal 0 (ein <CR> bei den meisten Hostmodeterminalprogrammen), das Paket wird nicht ausgesendet.

Beispiele :

H	-	Heardliste anzeigen
H 0	-	Heardlisten-Update ausschalten
H 1	-	Heardlisten-Update einschalten
H 2	-	Heardliste löschen
H 10	-	Maximalanzahl Calls in Heardliste setzen

Die Heardliste ist resident, d.h. sie bleibt bei einem Reset oder einem Abschalten bei Pufferung des RAM mit Batterie erhalten.

ACHTUNG: Bei einer langen Heardliste kann der Neustart nach einem Reset sehr lange dauern, da bei jedem dem Freispeicher zuzuordnenden Buffer erst geprüft werden muss, ob es ein Buffer der Heardliste ist, der nicht zugeordnet werden darf.

Mit dem 'I'-Befehl wird das Absenderrufzeichen des TNC eingegeben (MY-CALL). Nach der ersten Inbetriebnahme ist es mit Leerzeichen gefüllt. Für jeden Kanal kann ein beliebiges Rufzeichen eingegeben werden. Nach einem DISCONNECT wird wieder das Rufzeichen von Kanal 0 übernommen. Bei einer bestehenden Verbindung kann das Rufzeichen nicht verändert werden.
ACHTUNG: Der TNC geht nur mit eingegebenem Rufzeichen auf Sendung!

Mit dem 'JHOST'-Befehl wird zwischen der Betriebsart Terminal-Modus und Host-Modus umgeschaltet.

Mit dem 'K'-Befehl wird die Stamp-Funktion ein- und ausgeschaltet bzw. die 24-Stunden-Uhr oder der Kalender gestellt. Mit 'K 0' schaltet man die Ausgabe dieses Stamps ab. Trotzdem werden alle Meldungen/Header mit einem Stamp versehen. Ein Einschalten des Stamps wirkt also auch richtig bei längere Zeit im Buffer stehenden Meldungen/Headern mit abgeschaltetem Stamp. Mit 'K 1' schaltet man die Stampausgabe für Statusmeldungen ein, mit 'K 2' die Stampausgabe für Statusmeldungen und Monitorheader. Die Eingabe von 'K' ohne Parameter zeigt die aktuelle Einstellung gefolgt vom Datum und der Uhrzeit.
Mit 'K hh:mm:ss' setzt man die Uhrzeit, hh = Stunde, mm = Minute, ss = Sekunde. Mit 'K dd.mm.yy' setzt man das Datum, europäisches Format, dd = Tag, mm = Monat, yy = Jahr. Mit 'K mm/dd/yy' setzt man das Datum im amerikanischen Format. Je nachdem wie man das Datum gesetzt hat, wird es bei Stamps entweder im europäischen oder amerikanischen Format ausgegeben.

Beispiele :

K	-	Stamp und Datum/Zeit anzeigen
K 0	-	Stamp abschalten
K 1	-	Stamp Statusmeldungen einschalten
K 2	-	Stamp Status- und Monitormeldungen einschalten
K 20.02.88	-	Datum setzen, europäische Form
K 02/20/88	-	Datum setzen, amerikanische Form
K 17:36:00	-	Uhrzeit setzen

Mit dem 'L'-Befehl wird der Link-Status eines (Parameter = 0...4) oder aller Kanäle (ohne Parameter) angezeigt. Es werden Informationen über den Weg der Verbindung (Rufzeichen und Digipeaterliste), Anzahl empfangener Frames, Anzahl noch nicht gesendeter Frames, Anzahl noch nicht bestätigter Frames und der jeweilige Retry-Zähler angezeigt. Der jeweils benutzte Kanal wird durch ein '+' Zeichen markiert. Im Host-Modus funktioniert dieser Befehl etwas anders, mehr darüber im Kapitel über den Host-Modus.

Mit dem 'M'-Befehl wird der Monitor-Modus eingestellt. Mit dem nachfolgenden Parameter, in diesem Falle ein oder mehrere Buchstaben aus der folgenden Liste, kann angegeben werden, welche Frames angezeigt werden sollen.

Buchstabe	Paket-Typ
N	keine
I	Informationen
U	unprotokollierte Sendungen
S	Kontroll Pakete
C	Monitor auch an wenn eine Verbindung besteht

+	<Liste von bis zu 8 Rufzeichen>:	nur Pakete dieser Stationen
-	<Liste von bis zu 8 Rufzeichen>:	keine Pakete dieser Stationen

Die kombinierte Benutzung der '+' und '-' Parameter ist nicht zulässig. Sie müssen als letzter Parameter vor dem Rufzeichen eingegeben werden. Die Eingabe von '+' oder '-' ohne Rufzeichen löscht die aktuelle Liste.

Mit dem 'N'-Befehl wird angegeben, wie oft die Zustellung eines Paketes versucht werden soll, bevor eine Fehlermeldung erscheint. Für jeden Kanal kann ein eigener Wert angegeben werden, jedoch wird der Wert aus Kanal 0 nach jedem Disconnect oder RESET für alle Kanäle übernommen.

Mit dem 'O'-Befehl wird die maximale Anzahl von ausstehenden und unbeantworteten I-Frames (MAXFRAME) vorgegeben. Für jeden der Connect-Kanäle kann ein separater Wert vorgegeben werden. Nach jedem Disconnet oder RESET wird aber wieder der Parameter von Kanal 0 übernommen.

Der 'P'-Befehl wird zur Einstellung des P-Persistenz-Wertes benutzt. Dieser Parameter dient als Entscheidungsgrundlage für die Sendersteuerung und zur Kollisionsverminderung. Jedesmal wenn ein Paket ausgesendet werden soll, wird zunächst abgewartet bis der Kanal frei wird (kein DCD). Ist der Kanal frei, dann wird eine Zufallszahl zwischen 0 und 255 erzeugt. Ist diese Zufallszahl kleiner oder gleich der P-Persistenz-Zahl, dann betätigt der TNC die PTT-Leitung und die Informationsübertragung beginnt. Ist die Zufallszahl ausserhalb des Entschlossenheitsbereichs, dann geht der TNC nicht auf Sendung und wartet einen bestimmten Zeitraum, bis der beschrieben Vorgang wiederholt wird. Die Zeitdauer der Verzögerung wird durch den Slot Time Intervall festgelegt (siehe auch 'W'-Befehl).

Mit dem 'QRES'-Befehl wird die Firmware neu gestartet (Kaltstart), ähnlich wie bei einem Hardware-Reset.

Der 'R'-Befehl dient zum Ein- und Ausschalten der Digipeat-Funktion.

Mit dem 'S'-Befehl wird die jeweilige Kanalnummer eingestellt.

Mit dem 'T'-Befehl wird die Verzögerung zwischen Hochtasten des Senders und Start der Datenassendung eingestellt (TXDELAY). Die Einstellung erfolgt in 10 ms-Schritten.

Mit dem 'U'-Befehl hat man die Möglichkeit, eine Meldung an anrufende Stationen zu senden (U 1 Text). Dieser Text bleibt dann auch erhalten, wenn dieser Modus wieder abgeschaltet wird (U 0).

Mit dem 'V'-Befehl wird die Protokoll-Version 1 oder 2 zum Aufbau einer Verbindung vorgegeben, wobei für jeden Kanal ein eigene Version vorgegeben werden kann. Der Parameter für Kanal 0 wird nach Disconnect oder RESET aus dem RAM für alle Kanäle übernommen. Ausserdem kann während einer Verbindung die gerade benutzte Protokollversion abgefragt werden. Während einer Verbindung darf die Protokoll-Version nicht geändert werden.

Mit dem 'W'- Befehl kann die Dauer des Slot Time Intervalls (Zeitschlitzintervall) für die P-Persistenz Steuerung eingegeben werden. Immer wenn der TNC ein Paket austrahlen will und die unter P-Persistenz beschriebene Zufallszahl ausserhalb des P-Persistenz-Bereichs lag, dann

wird für die Dauer des Zeitschlitzes gewartet und die P-Persistenz-Prozedur erneut durchlaufen.

Der 'X'-Befehl kontrolliert die PTT-Leitung des TNC. Falls erforderlich kann hiermit das Einschalten des Senders unterdrückt werden, wenn man z.B. die Frequenz beobachten möchte, aber verhindern will, dass der TNC bei einer Connect-Anfrage ein Busy-Paket zurück sendet.

Mit dem 'Y'-Befehl kann die maximale Anzahl von Verbindungen bezogen auf empfangene Connect-Anfragen eingestellt werden. Der Wert beeinflusst nicht die Connect-Anfragen des Operators an andere Stationen. Bei 'Y 0' ist der TNC immer 'busy'.

Mit dem 'Z'-Befehl wird die Flowsteuerung und das XON/XOFF-Handshaking zum Terminal ein- bzw. ausgeschaltet. Ist die Flowsteuerung eingeschaltet, dann sendet der TNC solange keine Zeichen zum Terminal, wie Daten oder Befehle eingegeben werden. Bei ausgeschalteter Flowsteuerung werden die Zeichen vom TNC sofort ausgesendet, egal ob gerade eine Text-Zeile oder ein Befehl eingegeben wird. Arbeitet der TNC zeitweise ohne Terminal, dann sollte die Flowsteuerung und das XON/XOFF-Handshaking immer abgeschaltet sein, um Probleme mit den internen Buffern zu vermeiden. Ist die XON/XOFF-Steuerung eingeschaltet, so kann die Ausgabe zum Terminal mit CONTROL-S gestopt und mit CONTROL-Q wieder gestartet werden. Im Host-Modus ist die Flowsteuerung und das XON/XOFF-Handshaking immer abgeschaltet.

Über den '@'-Befehl sind weitere Befehle mit Parameterübergabe möglich:

Mit dem 'B'-Parameter wird die Anzahl der freien Puffer angezeigt.

Der 'D'-Parameter dient zum Ein- und Ausschalten des Vollduplexbetriebs des HDLC Ports.

Mit dem 'S'-Parameter kann der momentane Link-Status abgefragt werden.

Mit dem 'T2' Parameter wird die Zeit (in Schritten von 10 ms) eingestellt, die vergehen muss, bis ein eingegangenes Paket bestätigt wird. Durch diese Wartezeit können unter Umständen mehrere Pakete mit einer gemeinsamen Bestätigung bearbeitet werden.

Mit 'T3' wird die Zeit (in Schritten von 10 ms) eingegeben, die der TNC bei einer bestehenden Verbindung auf ein Lebenszeichen der Gegenseite wartet. Nachdem T3 abgelaufen ist, wird beim Partner angefragt, ob er noch empfangsbereit ist. Hierdurch werden auch die Probleme gelöst, die dadurch entstehen, dass der Partner ohne Disconnect einfach abschaltet.

Mit dem Parameter 'K' versetzt man den TNC in den KISS-Modus.

Mit dem Parameter 'V' wird der Rufzeichen-Check ein- bzw ausgeschaltet.

Voreingestellte Parameter

Bei einigen Parametern kann man durch Ändern der Werte im Eprom die Voreinstellung ändern. Die Parameterliste beginnt bei Adresse 0040 hex im Eprom.

TYP	WERT	BESCHREIBUNG
BYTE	1BH	Kommando Zeichen
BYTE	' ',60H	Eigenes Rufzeichen (s.Bemerkung 1)

```
   BYTE    '      '        Mnemonic Rufzeichen
   BYTE      04H           Maximale Anzahl von Connects
   BYTE      03H           Monitor Modus (s.Bemerkung 2)
   BYTE      01H           Digipeaten AUS/EIN
   BYTE      40H           P-Persistenz Wert
   BYTE      0AH           Zeitschlitz Intervall (10ms)
   BYTE      1EH           Sender Verzögerung (10ms)
   BYTE      03H           Flow Steuerung (TNC/Terminal)
   BYTE      01H           Sender PTT AUS/EIN
   BYTE      01H           Automatischer Zeilenvorschub AUS/EIN
   BYTE      01H           Echo-Befehl  AUS/EIN
   BYTE      01H           AX25 Version2 AUS/EIN
   BYTE      04H           Maximal unbestätigte Pakete
   BYTE      0AH           Maximale Anzahl der Retries
   BYTE      04H           FRACK Intervall
   BYTE      00H           Rufzeichen Check AUS/EIN
   BYTE      00H           Vollduplex (HDLC) AUS/EIN
   BYTE      00H           8-Bit Zeichen im Terminalmodus
                           (s. Bemerkung 3)
   BYTE      00H           Einstellung des Stamp-Parameters
                           (K-Befehl)
                            0 - Stamp aus
                            1 - Stamp bei Statusmeldungen
                            2 - Stamp bei Status- und
                                Monitormeldungen
   BYTE      00H           Einstellung des Heard-Parameters
                           (H-Befehl)
                            0 - Heardliste ausgeschaltet
                            1 - Heardliste eingeschaltet
   WORD      64H           Maximale Anzahl Calls in Heardliste
                           (H-Befehl)
   WORD      64H           Timer T2 Intervall (10ms)
   WORD     4650H          Timer T3 Intervall (10ms)

Ausgeschaltet = 00H / Eingeschaltet = 01H

Bemerkung 1: Die SSID muss ein Bit linksgeschoben und mit 60H
             ge'oder't sein.

Bemerkung 2: Der Monitormodus ist abhängig von folgenden Bits:

   BIT       FRAME
   ---       -----
    0        I Pakete
    1        UI Pakete
    2        Kontroll Pakete
    3        Monitor auch bei bestehendem Connect eingeschaltet

Bemerkung 3: 00H = Bit 7 maskiert / 01H =  Bit 7 nicht maskiert


Zusammengestellt aus Beschreibungen von DF2AU, DB2OS und DC4OX.
(DG1AD Juni/1988)
```

Geschichte der digitalen Nachrichtenübertragung im Amateurfunk

Schon seit langer Zeit ist das Funkfernschreiben im Amateurfunk gebräuchlich (RTTY). Die mechanischen Fernschreibmaschinen (Siemens T100, Lorenz Lo15) waren für Amateure billig erhältlich und ließen sich mit einem Konverter ohne Aufwand an einen Transceiver anschließen. Die übliche Geschwindigkeit war 45 Baud, das entspricht ca. 6 Buchstaben pro Sekunde.

Durch die Verwendung von Bildschirmen (Terminals, Hobbycomputer) wurde es für Amateure möglich, wesentlich schnellere Übertragungsverfahren zu verwenden. Mit geringem Aufwand konnten Konverter über eine serielle Schnittstelle an einen Computer angeschlossen werden und ermöglichten Übertragungsgeschwindigkeiten bis zu 2400 Baud (ca. 200 Buchstaben pro Sekunde). Eine Datensicherung (Fehlererkennung bzw. Fehlerkorrektur) fand nicht statt, ein 'Protokoll' existierte nicht.

Aus dem kommerziellen Funkfernschreibverkehr wurde das AMTOR-Verfahren übernommen. Hier werden kleinste Datenpakete mit nur wenigen Buchstaben gesendet und jedesmal die Empfangsbestätigung abgewartet. Bei negativer oder fehlender Empfangsbestätigung wird das Paket wiederholt. Die erzielbare Geschwindigkeit liegt, wie bei RTTY bei etwa 6 Buchstaben pro Sekunde, jedoch sind Übertragungsfehler unwahrscheinlich.

Der Begriff 'Packet-Radio' wurde Ende 1981 in Tucson, Arizona geprägt. Eine Gruppe von Funkamateuren hatte sich zur TAPR (=Tucson Amateur Packet Radio) zusammengeschlossen und plante, ein lokales Datennetz auf Amateurfunkfrequenzen aufzubauen. Dabei war jedoch vorgesehen, dieses Datennetz später international auszuweiten.

Der erste Schritt wurde im Oktober 1982 unternommen. Bei der AMSAT in Washington trafen sich verschiedene Gruppen von Funkamateuren, und legten ein Protokoll für eine Paketdatenvermittlung fest. Hauptzielrichtung war es, ein einheitliches Verfahren zur Datenübertragung von und zu den geplanten Phase III B Satelliten (OSCAR 10) zu entwickeln.

Für die Datenübertragung innerhalb der postalischen Netze war das X.25 Protokoll bereits eingeführt, bei uns ist es unter der Bezeichnung 'Datex-P' gebräuchlich. So lag es nahe, auch für den Amateurfunk ein ähnliches Verfahren zu verwenden. Im Gegensatz zum postalischen Netz, das speziell für Punkt-zu-Punkt-Verbindungen ausgelegt ist, 'hört' ein Funkempfänger alle benachbarten Sender und wird auch von allen Nachbarn gestört. Außerdem ist es beim Amateurfunk wichtig, daß die Rufzeichen der beteiligten Stationen mit ausgesendet werden.

Das erweiterte X.25 Protokoll, das alle Anforderungen für den Amateurfunkbetrieb aufweist, wurde festgelegt und AX.25 (A=Amateur) genannt. AX.25 definiert die Stufe 2 des OSI-ISO-Modells.

Die TAPR entwickelte 1983 eine Rechnerkarte 'TNC1', die die Daten in dem AX.25 Protokoll senden und empfangen konnte. Der Bausatz kostete damals 240 $ und umfaßte eine 16x27 cm große Leiterplatte mit dem Motorola Mikroprozessor 6809 und den Modem-IC XR2211 und XR2201. Die Stromversorgung war auf der Platine enthalten, bei +12, +5 und -12 Volt betrug die Leistungsaufnahme einige Watt.

Während sich das TNC1 in den USA langsam verbreitete, gab es in der Bundesrepublik nur wenige Anwender für diese Betriebsart. Eine Stuttgarter Gruppe machte 1982 Versuche mit 1200 Baud ASCII Übertragung und programmierte verschiedene Kommandos, mit denen man Daten fehlerfrei austauschen konnte.

Ab 1985 kam eine Neuentwicklung heraus: das TNC2. Es hatte den Z80 Mikroprozessor, 8 kByte RAM, 32 kByte EPROM und verwendete größtenteils HCMOS IC um den Stromverbrauch in Grenzen zu halten. Mit dem TNC2 hielt auch die Betriebsart Packet-Radio Einzug in Deutschland.

Es folgten verschiedene Weiterentwicklungen des TNC2, die sich jedoch stark an das von TAPR entwickelte TNC2 anlehnten. Da der Quellcode (Source) des Z80-Programms nicht veröffentlicht wurde, war man darauf angewiesen, TNC-Nachbauten mit gleichen Programmen (EPROM's) zu betreiben wie das Original. Insbesonders das Modem wurde verbessert, indem man die XR2211/2206 gegen ein TCM 3105 austauschte und somit auf den Abgleich der Modemfrequenzen verzichten konnte. Die sehr verbreiteten TNC2C begnügen sich mit 5 Volt Versorgungsspannung und haben einen Leistungsbedarf von weit unter 1 Watt, wovon der größte Teil für den Betrieb der Leuchtdioden benötigt wurde.

Der Betrieb fand in den Anfangszeiten vorwiegend auf dem 2 m Band statt. Die Frequenz 144.675 war in den Ballungsgebieten häufig stark überlastet und der Verkehr über mehrere Digipeater hinweg war nur in den 'ruhigen' Stunden des Tages möglich. Ende 1986 wurden dann die ersten 'offiziellen' Digipeater genehmigt. Diese Umsetzer arbeiten vollduplex im 70cm Band, das heißt, sie können gleichzeitig senden und empfangen. Durch den großen Einzugsbereich dieser Umsetzer war es nun auch mit geringem Stationsaufwand möglich, sichere Packet-Verbindungen über große Entfernungen herzustellen.

Eine Gruppe von Frankfurter Funkamateuren entwickelte einen Steckkartenrechner, der eine Vermittlung der Datenpakete möglich machte. An den RMNC (Rhein-Main-Network-Controller) konnten mehrere Funkgeräte angeschlossen werden und die ankommenden Datenpakete wurden je nach dem Ziel auf dem entsprechenden

Sender ausgegeben. Die Digipeater verfügten meist über einen Duplexkanal (Benutzerzugang) und einen oder mehrere Simplexkanäle (Links), mit denen sie im 23cm-Band mit den benachbarten Digipeatern verbunden waren. Der RMNC hat sich wegen seiner Leistungsfähigkeit bei niedrigen Gerätekosten in DL weit verbreitet.

Eine einfache und billige Möglichkeit, einen Netzknoten aufzubauen, war das 'NET-ROM'. Mithilfe dieses Programms (im EPROM) konnte man mehrere Standard-TNC zu einem Netzknoten zusammenschalten. Der Benutzer des Knotens konnte die Liste der erreichbaren Nachbarknoten abfragen und zu diesen Knoten weiterconnecten. Das NET-ROM war eine kommerzielle Entwicklung aus den USA und verbreitete sich aufgrund der Beschaffungsprobleme nur zögernd. Später wurde ein kompatibles Programm 'TheNet' in Deutschland entwickelt, das für die kostenlose Nutzung durch Funkamateure freigegeben war und sich rasch verbreitete.

Ab 1987 wurden große Anstrengungen unternommen, ein Datennetz aufzubauen, das über 23cm-Richtfunkstrecken die Verbindung aller Digipeater ermöglicht. Mit großem persönlichen Einsatz und erheblichem finanziellem Aufwand wurden an zentralen Stellen Digipeater gebaut und, soweit möglich über 23cm Richtfunkstrecken an das Netz angebunden.

Als sehr nützliche Einrichtung kamen nun zahlreiche Mailboxen ans Netz. In diesen Mailboxen konnten allgemein interessierende Informationen, sowie Nachrichten an bestimmte Funkamateure ('persönliche Mail') abgelegt bzw. abgerufen werden. Die Mailboxen stehen über 'store and forward' miteinander in Verbindung und tauschen Nachrichten untereinander aus. Ein Brief ('Mail'), der z.B. in Kiel in eine Mailbox eingegeben wird, wird von Box zu Box weitergereicht, bis er in der Zielbox z.B. in Stuttgart ankommt. Dort ruft ihn der Adressat dann gelegentlich ab.

Die Verbindung zu Packet-Stationen außerhalb Deutschlands erfolgt durch 'Gateways', das sind Packet-Digipeater, die Nachrichten von Kurzwelle aufnehmen und in das deutsche Netz einspeisen. Auf diese Weise kommen Nachrichten aus der ganzen Welt in jede lokale Mailbox. Momentan (1991) werden täglich etwa 20 bis 50 Nachrichten mit durchschnittlich 2 kByte in einer Mailbox gespeichert. In manchen Mailboxen hat man Zugriff auf einige zehntausend Meldungen, die zum Teil noch aus den Anfangszeiten der Mailbox-Zeit Ende 1987 stammen.

Als experimentelle Betriebsart wurde Packet-Radio von der Deutschen Bundespost nicht beanstandet. Mit dem Überhandnehmen der 'privaten' Digipeater auf dem 2m Band wurde jedoch darauf hingewiesen, daß ein unbeaufsichtigter Betrieb einer Funkanlage nicht gestattet ist. Sofern es zu keiner Beanstandung kam, wurden die 'grauen' Digipeater jedoch geduldet. Durch die aufkommenden lizensierten Digipeater verloren die 'grauen' Digipeater zunehmend an Bedeutung. Die Mailboxbetreiber versuchten mit Erfolg, die Ausbreitung von Nachrichten, die gegen Vorschriften verstoßen, zu verhindern (z.B. Meldungen, die den Charakter von Kleinanzeigen haben)

Ende 1988 wurden erste Versuche mit Modulationsarten unternommen, die mehr als 1200 Baud erlaubten. (z.B. G3RUH-Modem) Diese Modems wurden bevorzugt für die Verbindung der Digipeater untereinander verwendet.

Der Weg der Datenpakete durch das Digipeaternetz mußte man beim Verbindungsaufbau selbst angeben. Ab Mitte 1987 wurde in Stutgart ein Knotenrechner in Betrieb genommen, der die Wege zu den empfangenen Stationen speicherte und die notwendigen Digipeaterrufzeichen automatisch in die Pakete einfügte. Dies vereinfachte die Benützung des Digipeaters wesentlich, da man die Vermittlung der Daten dem Knotenrechner überlassen konnte. ('Autorouting') Einige Zeit später wurden ähnliche Autorouter-Programme auch in den RMNC und OE5XDL Knotenrechnern eingebaut.

Grundsätzlich kann bei Packet-Radio nur eine Verbindung zwischen zwei Stationen bestehen. Um Rund'gespräche' zu ermöglichen, wurden sog. CONVERS-Programme installiert, die eine schriftliche Unterhaltung mehrerer Stationen ermöglichen. Dabei sind alle Stationen mit einem Knotenrechner verbunden, der die Nachrichten empfängt und an alle Stationen (die am CONVERS-Betrieb teilnehmen) weitergibt.

Außer AX.25 werden von einigen Digipeatern auch TCP/IP-Verbindungen unterstützt. Dieses Protokoll eignet sich sehr gut zum automatischen Datenaustausch von Computern. Digipeater vom Typ' WAMPES' (=Württemberger Amateur MultiProtokoll Experimentelle Software) können connected werden und gestatten jedem Amateur, in das unix-System des Rechners einzuloggen und mit dem Rechner zu arbeiten, Programme zu erstellen und zu speichern. Es können viele Benutzer gleichzeitig diese Eigenschaft der WAMPES-Knoten in Anspruch nehmen.

Als sehr nützliche Einrichtung für den Kurzwellen-DX'er sind überall sogenannte 'DX-Cluster' eingerichtet worden. Diese Rechner sind eine Mischung von CONVERS-Knoten und Datenbank. Wenn man solch ein DX-Cluster connected, bekommt man brandaktuelle DX-Informationen (Call, Frequenz, Uhrzeit etc.) laufend auf dem Bildschirm angezeigt. Hat man selbst einen 'seltenen Fisch' auf Kurzwelle geangelt, dann gibt man die Daten in den Clusterrechner ein, der die Meldung sofort an alle angeschlossenen Interessenten weitergibt. Zusätzlich steht eine Datenbank für Prefixe oder QSL-Manager-Adressen zur Verfügung.

Die weitere Entwicklung bei Packet-Radio wird im weiteren Ausbau des Netzes bestehen. Benutzerzugänge mit 9600 Baud werden bestimmt bald die Regel, spezielle Packet-Transceiver für den 9600 oder 19200 Baud-Betrieb werden gebraucht. Immer mehr werden grafische Daten (Bilder etc.) oder digitalisierte Sprache übertragen, was das Datenaufkommen vergrößert. Dies führt natürlich zu einem enormen Bedarf an schnellen Verbindungen zwischen den Digipeatern. Die Knotenrechner werden in Bezug auf Komfort und Geschwindigkeit weiterentwickelt werden. Was die Linkstrecken betrifft bestehen mittlerweile Probleme mit der Genehmigung der 23cm Frequenzen, da Amateurfunk auf 23cm nur auf sekundärer Basis erlaubt ist und z.B. die militärische Nutzung des Bandes dem Frequenzbedarf der Amateure gewisse Grenzen setzt.

Seit Ende 1996 sind Packet-Radio-Linkstrecken in Betrieb, die mit 614000 Baud fullduplex Daten austauschen. Solche Verbindungen werden es längerfristig ermöglichen, sehr große Datenmengen im Netz zu transportieren und dem Packet-user ein Medium zur Verfügung zu stellen, das den 28800 Baud Telefonmodems ebenbürtig ist oder deren Leistung sogar übertrifft. Internet-Zugangänge über Packet wären dann kein Problem.

Weiterführende Schriften

Tucson Amateur Packet Radio TNC Firmware release 1.1.7 commands booklet
ca 100-seitige Beschreibung der TAPR-Befehle (englisch), erhältlich bei TAPR, Tel. 001-602-749-9479 für ca 5 US $ gegen VISA phone order (oder bei SYMEK)

ADACOM Magazin, Zeitschrift des ADACOM-Fachverbands e.V., erscheint vierteljährlich im Rahmen des Mitgliedsbeitrags. Inhalt: Packet-Radio-Technik, Modems, Transceiver für Packet-Radio, Packet und HF-Meßtechnik etc.

Funk-Telegramm, Monatszeitschrift Verlag Rojahn und Kraft, Amateurfunkspezifische Artikel, Umbauvorschläge, Kurzwellen-DX-Nachrichten, Kleinanzeigen

CQ-DL, Monatszeitschrift des DARC e.V.im Rahmen der Mitgliedschaft erhältlich. Amateurfunktechnik, Vereinsnachrichten, Anzeigenteil

Packet-Radio, digitale Betriebstechnik, Günter Grünfeld, DL6YCL, ca. 420 seitiges Buch (deutsch). Enthält alles über Packet-Radio, Digipeater, Satelliten etc. Sehr aktuell. DARC-Verlag ISBN 3-88692-017-1, dritte Auflage 1997, im Buchhandel oder über den DARC-Verlag

Anschriften und Bezugsquellen

SYMEK GmbH (Herstellung und Vertrieb des TNC2S, TNC2H, TNC2M etc.) DK9SJ, Ulf Kumm, Johannes-Krämer-Straße 34, 70597 Stuttgart Telefon: 0711/7678923 (8-18 Uhr), Fax 0711/7678924, Packet: DK9SJ@DB0SAO, eMail ulf@symek.de, Internet http://symek.de

Wolf-Henning Rech, DF9IC@DB0GV, Entwickler des 9600-Baud Modems mit GALs, Verfasser vieler interessanter Artikel über Packet-Radio

ADACOM, unabhängiger Fachverband für den Amateur-Datenfunk e.V., Kaiserstr. 31, 7515 Linkenheim

DARC, Deutscher Amateur Radio Club, Postfach 1155, 3507 Baunatal 1

BayCom, vertreibt Modems nach DF9IC, EPROMs, GALS und das Buch "9600 Baud FSK-Technik nach G3RUH-Standard"