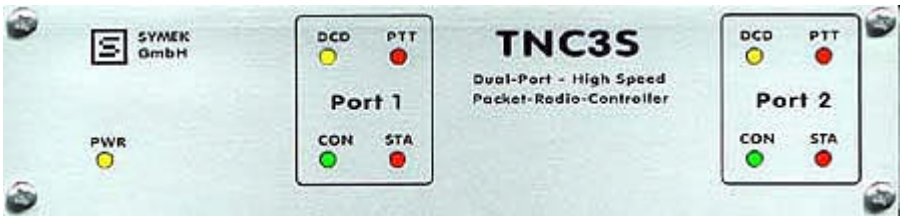
Packet Radio

... *connecting the future* ...

Installation auf ‘302 TNCs



??Installation 3NET
??Token Ring
??KISS/SMACK
??SRPM
??HighSpeedBus
??IPKISS

Autoren
Software: Jimy, DL1GJI
Dokumentation: Manfred DL2GWA

Inhaltsverzeichnis

Allgemeines zu MC68302 TNCs

(X)NET wurde ursprünglich für den TNC3 entwickelt und ist auf dessen moderne Hardwarearchitektur (MC68302 = MC68000 + RISC Communications Controler) zugeschnitten. Der '302-Prozessor von Motorola wurde speziell für Telekommunikationsanwendungen entworfen. Er besitzt drei universelle DMA-gestützte serielle Einheiten (SCC), die von (X)NET voll genutzt werden.

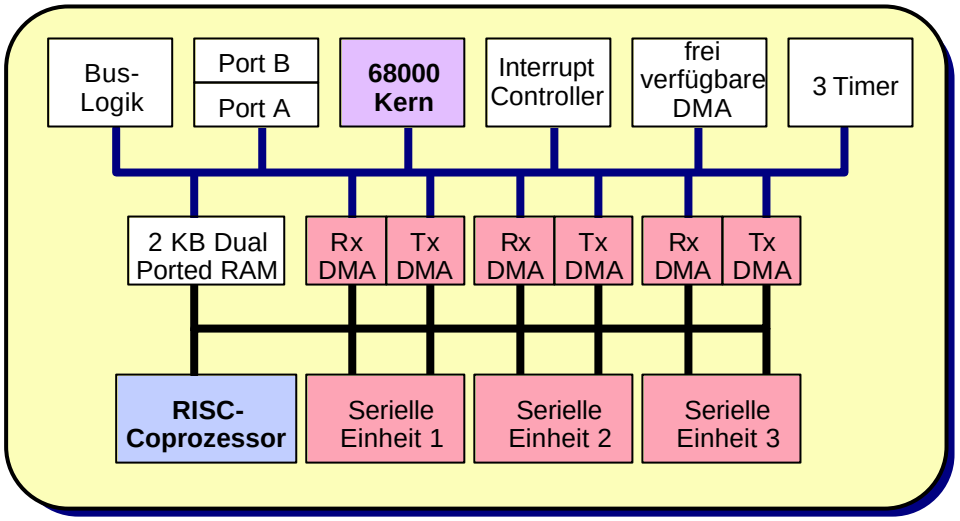


Abbildung 1: Schematischer Aufbau des MC68302 Kommunikations-Prozessors

Jeder seriellen Einheit (SCC) kann eine der folgenden Betriebsarten zugeordnet werden:

?? RSKISS - KISS bis 115200 Baud
?? SMACK - CRC-KISS bis 115200 Baud
?? RMNC - RMNC-CRC-KISS bis 115200 Baud
?? SLIP - bis 115200 Baud
?? AX.25 - bis 1,6 Megabaud
?? HSKISS - bis 1,6 Megabaud
?? TRKISS - RS232 Token-Ring KISS - bis 115200 Baud
?? TRSMACK - RS232 Token-Ring KISS mit CRC bis 115200 Baud
?? Hostmode (nach WA8DED) - bis 115200 Baud

Selbstverständlich laufen diese Betriebsarten auf mehreren SCCs parallel in beliebiger Kombination. Der TNC-Hostmode (nach WA8DED) läuft allerdings nur auf einer SCC.

(X)NET für TNC3 (3NET) ist für alle ‘302- TNCs identisch. Wegen der unterschiedlichen Hardware ist jedoch für jeden TNC-Typ (TNC31, TNC3, TNC4e) ein eigener Flash-Inhalt notwendig.

Um den Start mit 3NET möglichst einfach zu gestalten, sind häufig verwendete Hardwarekonfigurationen bereits im Flash-File („EPFLASH.ABS„) enthalten. Durch die Wahl des entsprechenden Programm DIP-Schalters am TNC können diese aktiviert werden.
Dadurch entfällt ein langwieriges Editieren von Konfigurationsdateien um erst einmal die Hardware zum Laufen zu bekommen.

3NET auf dem TNC3

Die SCCs im TNC3 können bei (X)NET in verschiedenen Modi betrieben werden. Im normalen TNC-Betrieb sieht die Konfiguration so aus:

SCC1 = Port 1 = Modem 1
SCC2 = Port 2 = Modem 2
SCC3 = serielle Schnittstelle zum PC

Der Modus in der eine SCC betrieben wird, kann durch den zugeordneten Treiber festgelegt werden:

ID	Treiber	Beschreibung	Anz. Ports
A	AX25	direkt angeschlossenes Modem	1
H	HSKISS	HighSpeedBus	16
K	KISS		16
R	TERM	RS232-Terminal-Schnittstelle (nur SCC3)	1
S	SMACK	KISS mit Prüfpolynom	8
T	TRKISS	Token-Ring	16
P	SRPM	Serial-Ring-Protocol	8
U		Benutzerdefinierte Zuordnung	-
X		Kein Treiber zugeordnet	-

Tabelle 1: SCC-Treiber

Die Grundeinstellung und Zuweisung der Treiber und Ports erfolgt durch DIP-Schalter-Einstellung. Hierbei werden die zugewiesenen Batch-Dateien (NETxxx.NET) aufgerufen und die darin enthaltenen Anweisungen abgearbeitet. Die Batchdateien sind im ASCII-Format gespeichert und können auf eigene Bedürfnisse abgeändert werden. Zu beachten ist jedoch, daß bei einem Reset aus dem EPROM (oder Flash-EPROM) die Defaultwerte eingestellt sind.

DIP	Konf	SCC1	SCC2	SCC3	Beschreibung
1	NETAAR.NET	AX25	AX25	TERM	TNC-Konfiguration
2	NETAAA.NET	AX25	AX25	AX25	3-Port Single-Board
3	NETAAK.NET	AX25	AX25	KISS	NOS-Frontend
4	NETAAS.NET	AX25	AX25	SMACK	DieBox-Digi-Frontend
5	NETAAT.NET	AX25	AX25	TRKISS	Low Cost Digi
6	NETHXR.NET	HSKISS		TERM	HighSpeedBus
7	NETTXR.NET	TRKISS		TERM	Token-Ring
8	NETHTR.NET	HSKISS	TRKISS	TERM	Mixed
9	NETAAH.NET	AX25	AX25	Highspeedbus	High-Speed-Bus über Arbiter
10	NETAAP.NET	AX25	AX25	SRPM	SRP 115200 Baud
15	NETUUU.NET				Benutzerdefiniert

Tabelle 2: TNC3 Programm-DIP-Schalter

1	2	3	4	5	6	7	8	DIP-Schalter-Stellung
x	x	x	?	?	?	?	?	Dip1
x	x	x	?	?	?	?	?	Dip2
x	x	x	?	?	?	?	?	Dip3
x	x	x	?	?	?	?	?	Dip4
x	x	x	?	?	?	?	?	Dip5
x	x	x	?	?	?	?	?	Dip6
x	x	x	?	?	?	?	?	Dip7
x	x	x	?	?	?	?	?	Dip8
x	x	x	?	?	?	?	?	Dip9
x	x	x	?	?	?	?	?	Dip10
x	x	x	?	?	?	?	?	Dip15

Tabelle 3: Einstellung der TNC3 DIP-Schalter

DIP 1: AAR Single-Board (normale TNC-Konfiguration)

Diese Betriebsart erfordert keine Hardwareänderung am TNC3. Sie wird so einfach bedient, wie ein normaler TNC3 mit der Turbo-Firmware oder der TNC3BOX.

SCC1 = Port1 = Modem1 = Logischer Port 1
SCC2 = Port2 = Modem2 = Logischer Port 2
SCC3 = serielle Schnittstelle zum Terminal/PC

Dieser Konfiguration zugewiesene Batch-Datei (NETAAR.NET) enthält:

```
# 2 * AX.25 + Terminal
attach scc1 ax25 1 1
attach scc2 ax25 2 1
```

DIP 2: AAA Single-Board (3-Port)

Konfiguration für einen Kleinzellendigi bestehend aus einem TNC3. Anstatt der seriellen Schnittstelle zum PC wir ein weiteres Modem an SCC3 angeschlossen.

SCC1 = Port1 = Modem1 = Logischer Port 0
SCC2 = Port2 = Modem2 = Logischer Port 1
SCC3 = Port3 = Modem3 = Logischer Port 2

Dieser Konfiguration zugewiesene Batch-Datei (NETAAA.NET) enthält:

```
# 3 * AX.25
# First detach terminal from SCC3
detach scc3
attach scc1 ax25 0 1
attach scc2 ax25 1 1
attach scc3 ax25 2 1
```

DIP 3: AAK (Zwei Modems und KISS)

Diese Betriebsart erfordert keine Hardwareänderungen am TNC3. Sie ist als AX.25-Frontend für NOS einsetzbar.
Achtung: Das Rufzeichen kann über die KISS-Schnittstelle nicht eingestellt werden. Es muß mit der DIP1-Konfiguration voreingestellt werden.

SCC1 = Port1 = Logischer Port 0
SCC2 = Port2 = Logischer Port 1
SCC3 = Multiport KISS 19200 Baud, Ports 2 bis 4

Dieser Konfiguration zugewiesene Batch-Datei (NETAAK.NET) enthält:

```
# 2 * AX.25 + Kisslink
# First detach terminal from SCC3
detach scc3
attach scc1 ax25 0 1
attach scc2 ax25 1 1
attach scc3 kiss 2 2 19200
```

DIP 4: AAS (Zwei Modems und SMACK)

Wie DIP3, nur SMACK statt KISS

SCC1 = Port1 = Logischer Port 0
SCC2 = Port2 = Logischer Port 1
SCC3 = Multiport SMACK 19200 Baud, Ports 2 bis 4

DIP 5: AAT (Zwei Modems und Token-Ring)

Zwei direkt angeschlossene Modems und ein Token-Ring mit 19200 Baud und 8 Ports.

SCC1 = Logischer Port 0
SCC2 = Logischer Port 1
SCC3 = Multiport Token-Ring-Kiss 19200 Baud, Ports 2 bis 9

Dieser Konfiguration zugewiesene Batch-Datei (NETAAT.NET) enthält:

```
# 2 * AX.25 + Token Ring KISS
attach scc1 ax25 0 1
attach scc2 ax25 1 1
detach scc3
attach scc3 trkiss 2 8 19200
```

DIP 6: HXR (Klassische HighSpeedBus-Konfiguration)

SCC1 = HighSpeedBus Ports 0 bis 7
SCC2 = frei
SCC3 = serielle Schnittstelle zum Terminal/PC

Dieser Konfiguration zugewiesene Batch-Datei (NETHXR.NET) enthält:

```
# HighSpeedBus
attach scc1 hskiss 0 8
```

DIP 7: TXR (Klassische Token-Ring-Konfiguration)

SCC1 = Token-Ring Ports 0 bis 7
SCC2 = frei
SCC3 = serielle Schnittstelle zum Terminal/PC

Dieser Konfiguration zugewiesene Batch-Datei (NETTXR.NET) enthält:

```
# Token Ring KISS
```

```
attach scc1 trkiss 0 8 19200
```

DIP 8: HTR (HighSpeedBus und Token-Ring)

SCC1 = HighSpeedBus Ports 0 bis 7
SCC2 = Token-Ring Ports 8 bis 13
SCC3 = serielle Schnittstelle zum Terminal/PC

Dieser Konfiguration zugewiesene Batch-Datei (NETHTR.NET) enthält:

```
# HighSpeedBus + Token Ring
attach scc1 hskiss 0 8
attach scc2 trkiss 8 8 19200
```

DIP 9: AAH (Highspeedbus)

SCC1 = Port 0
SCC2 = Port 1
SCC3 = High-Speed-Bus

Dieser Konfiguration zugewiesene Batch-Datei (NETAAH.NET) enthält:

```
# 2 * AX.25 + Highspeedbus
attach scc1 ax25 0 1
attach scc2 ax25 1 1
detach scc3
attach scc3 hskiss 2 16
```

DIP 10: AAP (Serial-Ring-Protocol)

SCC1 = Port 0
SCC2 = Port 1
SCC3 = Serial-Ring-Protocol

Dieser Konfiguration zugewiesene Batch-Datei (NETAAP.NET) enthält:

```
# 2 * AX.25 + 8 Ports SRPM
# First detach terminal from SCC3
detach scc3
attach scc1 ax25 0 1
attach scc2 ax25 1 1
attach scc3 srpm 2 8  115200
```

DIP 15: UUU (Benutzerdefinierte Konfiguration)

In diesem Fall startet (X)NET mit einer Konfiguration, die in der Datei AUTOEXEC.NET abgelegt ist.

Beispiel:

```
# DIP 15 = Start mit netuuu.net
#
# HighSpeedBus auf SCC1
attach scc1 hskiss 0 8
# Config SysOp Port
po 8 per 255
po 5 qua 0
po 2 txd 100
```

Der Knoten wird im o.a. Beispiel als High-Speed-Bus-Digi konfiguriert. Die Autoexec.net ist frei definierbar. Es können hierin auch gleichzeitig externe Prozesse gestartet werden, sofern die L7-Programme (XTS oder XTPs) in der RAM-Disk vorhanden sind oder es können entsprechende Parameter eingegeben werden.

3NET Beispielkonfiguration

Ein komplexeres Beispiel einer benutzerdefinierten (X)NET-Konfiguration: DB0XYZ ist ein Knoten mit 6 Funkports und einer Mailbox (DB0XYZ-8). Der Knoten besteht aus einem TNC3 mit 1MB RAM als Master TNC mit (X)NET in den Flash-EPROMs.
Zwei TNC2H hängen über Token-Ring am Master und zwei TNC3 werden über den HighSpeedBus angeschlossen. Die Mailbox soll über SMACK mit 11500 Baud ebenfalls direkt am Master angeschlossen werden.

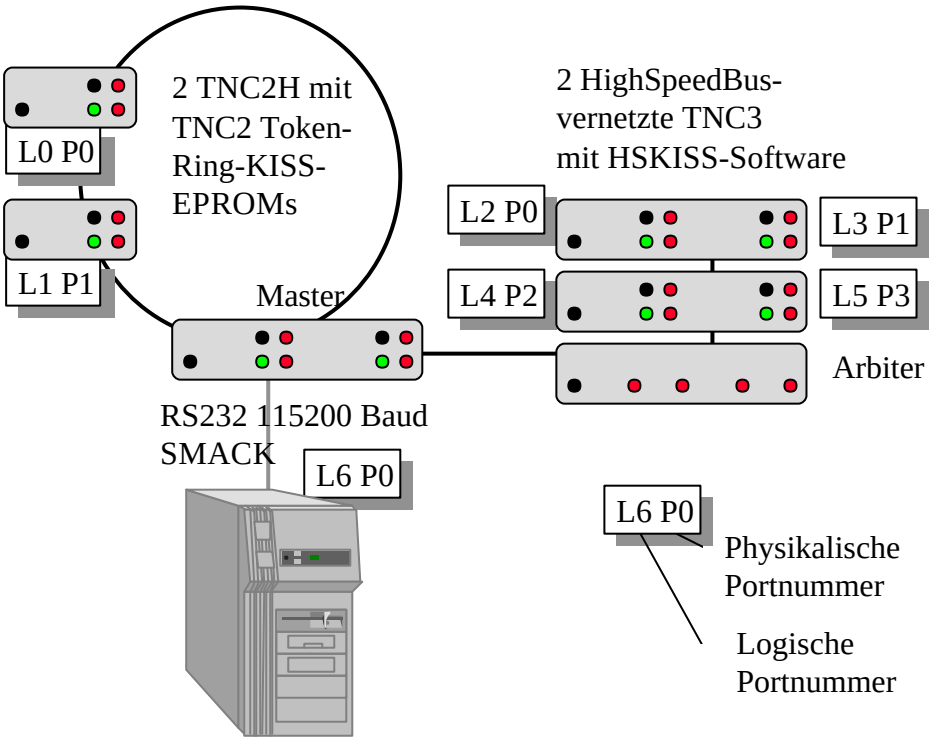


Abbildung 2: Beispielkonfiguration

Diese aufwendige Konfiguration wird nicht standartmäßig von (X)NET unterstützt. Für sie wird nun in der Datei „NETUUU.NET„ eine spezielle Konfiguration hinterlegt:

```
# Erst mal das Terminal abklemmen und die SCC3 freimachen
det scc3
att scc1 trkiss 0 2 19200
#       \      \    \ \    \
#        \      \    \ \    RS232-Baudrate des Token-Rings
#         \      \    \ Anzahl der Token-Ring-Ports
#          \      \   Erste logische Portnummer für den Token
Ring
#           \      Token-Ring KISS-Treiber
#            serielle Schnittstelle 1 des TNC3 (normalerweise
Modem 1)
att scc2 hskiss 2 4
#       \      \    \ \
#        \      \    \ Zahl der Ports auf dem HighSpeedBus
#         \      \   Anzahl der Token-Ring-Ports
#          \      HighSpeedBus KISS-Treiber
```

```
#            serielle Schnittstelle 2 des TNC3 (normalerweise
Modem 2)
att scc3 smack   6 1 115200
#        \       \    \ \    \
#         \       \    \ \    RS232-Baudrate zur Mailbox
#          \       \    \ Anzahl der SMACK-Ports
#           \       \    Erste logische Portnummer für SMACK
#            \       SMACK-Treiber (Checksum-KISS)
#             serielle Schnittstelle 3 des TNC3 (normalerweise
PC)
```

Nachdem diese Datei in die TNC3-Ramdisk kopiert wurde, kann der TNC3 Programm-DIP-Schalter auf 15 gestellt werden. Nach einem Reset des Master-TNC ist die vorliegende Konfiguration unterstützt.

Das Port-Kommando zeigt nun die logischen Ports an:

```
po   baud interface           txd per slt w ret qua dup dam duo

 0   1200  0 SCC1 TRKISS      250  64 100 5  10 128   0   0   0
 1   1200  0 SCC1 TRKISS      250  64 100 5  10 128   0   0   0
 2   9600  0 SCC2 HSKISS      250  64 100 5  10 128   0   0   0
 3   9600  0 SCC2 HSKISS      250  64 100 5  10 128   0   0   0
 4  19200  0 SCC2 HSKISS      250  64 100 5  10 128   0   0   0
 5  38400  0 SCC2 HSKISS      250  64 100 5  10 128   0   0   0
 6 115200  0 SCC3 SMACK       250  64 100 5  10 128   0   0   0
```

Das Kommando „sa 1„ zeigt Infos und Statistikdaten der drei unterschiedlich betriebenen SCCs im Master TNC an:

```
SCC3    : KISS/SMACK/TRKISS Driver Oct 20 1996 RS232: 115200 Baud
       CRC err 0 (12.11.96 20:25:24)
```

3NET auf dem TNC31

Da der TNC31 keine DIP-Schalter besitzt, gibt es hier auch kein Nachdenken über deren Einstellung. „Flasht„ man die EPFLASH.ABS Datei in den TNC, startet dieser mit der Konfiguration:

DIP	Konf	SCC1	SCC2	SCC3	Beschreibung
1	AUTOBOOT.NET	AX25	-	TERM	TNC-Konfiguration

Das Modem wird auf AX.25, die serielle Schnittstelle wird als WA8DED- kompatible Terminalschnittstelle konfiguriert. D.h., zunächst verhält sich die Software wie ein TNC31 mit der Turbo-Firmware oder der TNC3BOX.

Die Datei AUTOBOOT.NET sieht dabei folgendermaßen aus:

```
# 1 * AX.25 + Terminal
attach scc1 ax25 0 1
# Use autouser.net to attach further devices
exec autouser.net
```

Die Datei AUTOUSER.NET ist im Flash-EPROM nicht enthalten. Sie kann verwendet werden um weitere Konfigurationen durchzuführen.

Eine „AUTOUSER.NET„ für einen Token-Ring an der seriellen (TNC-) Schnittstelle sieht beispielsweise folgendermaßen aus:

```
# Attach 38k4 Token Ring with 4 Ports starting with
# (X)NET port number 2.
attach scc3 trkiss 2 4 38400
```

3NET auf dem TNC4e

Ethernet hat sich mittlerweile zur Standard-Vernetzung für LANs etabliert. Die erforderliche PC-Hard- und Software ist überall erhältlich. Die Preise für Ethernet-Karten sind trotz ihrer wesentlich höheren Leistungsfähigkeit auf das Preisniveau von seriellen I/O-Karten abgesunken. PC-Ethernet-Karten in PCI-Technologie sind einserseits exterm schnell und weit wichtiger: sie können Dank DMA den Datentransfer im PC im Hintergrund abwickeln. Spätestens wenn mehrere Server an einen Digi oder TNC angeschlossen werden sollen, zeigt sich der Vorteil von Ethernet: Die Server werden einfach eingesteckt.
Um einen einfachen Start mit dem TNC4e zu ermöglichen sind wie beim TNC3 bereits vordefinierte Konfigurationen im Flash-EPROM enthalten:

DIP	Konf	SCC1	SCC2	SCC3	Beschreibung
1	NETAAR.NET	AX25	AX25	-	TNC-Konfiguration
2	NETAAE.NET	AX25	AX25	-	TNC/AXIP-Konfiguration
3	AUTOBOOT.NET	-	-	-	
16	IPKISS Slave 1				192.168.44.4
17	IPKISS Slave 2				192.168.44.8
18	IPKISS Slave 3				192.168.44.12

Tabelle 4: TNC4 Programm-DIP-Schalter

DIP 1: Ethernet Router

Die Konfiguration 1 startet 3NET.APL und führt die Startdatei NETAAR.NET aus:

```
# (X)NET boot Configuration for TNC4e
#
# Start Ethernet Driver
#
start ether 192.168.44.1
#
# 2 * AX.25
#
attach scc1 ax25  0 1
attach scc2 ax25  1 1
#
# for further user defined attaches
#
exec autouser.net
```

Der Ethernet-Treiber wird geladen und dessen IP-Adresse auf die Nummer 192.168.44.1 festgelegt. Danach werden die SCC1 und SCC2 als normale AX.25-Modems „attached„.

Die Datei „AUTOUSER.NET„ ist nicht im Flash-EPROM enthalten. Sie kann in das RAM geladen werden um weitere Konfigurationen vorzunehmen.

DIP 2: Ethernet Digi (IPKISS)

TNC4e Ethernet-Digi. Vier TNC4 werden zu einem TNC4e-12-Port Digipeater zusammengeschlossen. Dabei wird der (X)NET-Master mit DIP-Schalterstellung 2 gestartet. Die Slaves 1 bis 3 jeweils mit der Schalterstellung 16 bis 18.

Die Konfiguration 2 startet 3NET.APL und führt die Startdatei NETAAE.NET aus:

```
# (X)NET boot Configuration for TNC4e
#
# Start Ethernet Driver
#
start ether 192.168.44.1
#
# 2 * AX.25
#
attach scc1 ax25  0 1
attach scc2 ax25  1 1
#
# 9 AXIP-Ports without AXIP-CRC
#
attach ip0  axip  3 1 cp 192.168.44.4
attach ip1  axip  4 1 cp 192.168.44.5
attach ip2  axip  5 1 cp 192.168.44.6
attach ip3  axip  6 1 cp 192.168.44.8
attach ip4  axip  7 1 cp 192.168.44.9
attach ip5  axip  8 1 cp 192.168.44.10
attach ip6  axip  9 1 cp 192.168.44.12
attach ip7  axip 10 1 cp 192.168.44.13
attach ip8  axip 11 1 cp 192.168.44.14
#
# for further user defined attaches
#
exec autouser.net
```

Der Ethernet-Treiber wird geladen und dessen IP-Adresse auf die Nummer 192.168.44.1 festgelegt. Danach werden die SCC1 und SCC2 als normale AX.25-Modems „attached„. Weitere TNC4e werden über AXIP angeschlossen.
Diese NETAAE-Konfiguration wird beispielsweise bei DB0PRT verwendet.

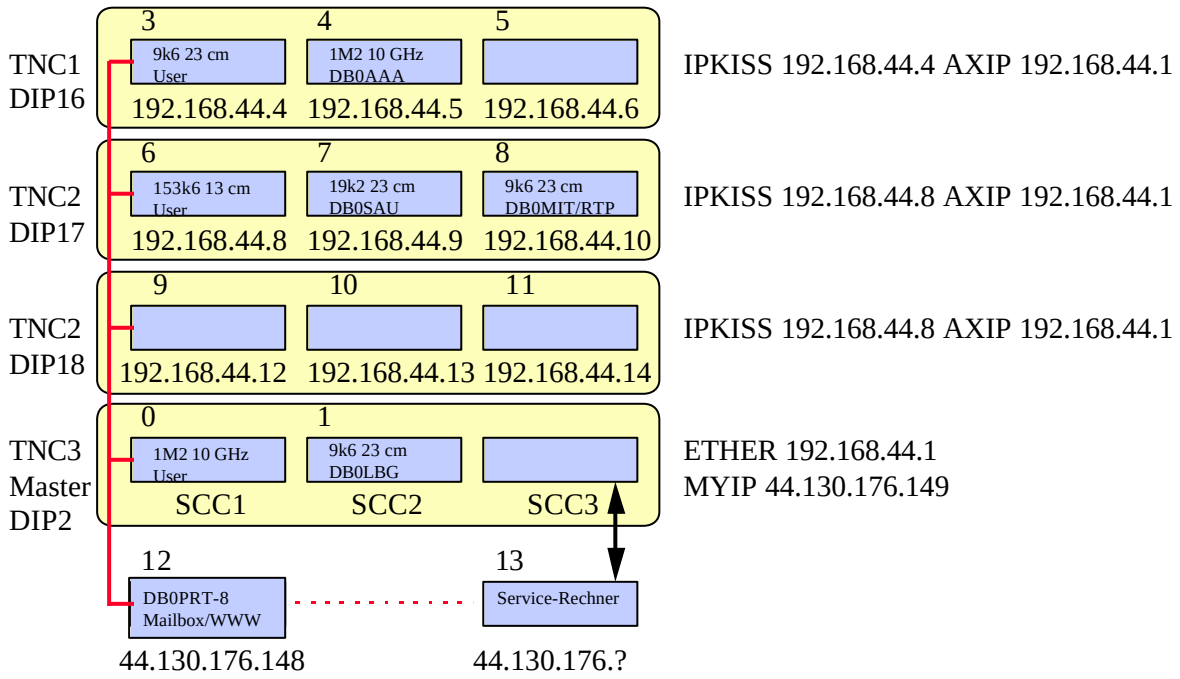


Abbildung 3: Konfiguration von DB0PRT

Der Anschluß der Mailbox erfolgt in der am Schluß aufgerufenen Datei „AUTOUSER.NET„.

DIP 3: 3NET-Start

Die Konfiguration 3 startet 3NET.APL und führt die Startdatei AUTOBOOT.NET aus. Diese Datei ist nicht im Flash-EPROM enthalten. Sie muß ins RAM geladen werden und kann eine völlig freie Konfiguration enthalten.

DIP 16, 17, 18: IPKISS-Mode

Diese Konfigurationen versetzen den TNC4e in den IPKISS-Slave-Mode mit jeweils unterschiedlichen IP-Adressen:

DIP	Software	Port0	Port1	Port2	RS232 (KISS)
16	IPKISS Slave 1	192.168.44.4	192.168.44.5	192.168.44.6	192.168.44.7
17	IPKISS Slave 2	192.168.44.8	192.168.44.9	192.168.44.10	192.168.44.11
18	IPKISS Slave 3	192.168.44.12	192.168.44.13	192.168.44.14	192.168.44.15

Tabelle 5: IPKISS-DIP-Schalter und IP-Nummern

Der Ethernettreiber ETHER.XTS

Auf dem TNC4 wird die Ethernetschnittstelle durch das Treiberprogramm EHTER.XTS aktiviert. Der Treiber ETHER.XTS wird als Hintergrundprozess gestartet. Er installiert insgesamt 16 IP-Devices, die für den Anschluß (attach) von AXIP oder AXUDP-Ports benötigt werden. Zusätzlich installiert der Treiber auch das UART-Device, mit Hilfe dessen sämtliche seriellen Treiber (KISS, SMACK, SLIP etc.) über die TNC4-UART „attached„ werden können.

Anschluss von Stationen über AXIP und AXUDP

Wie bei (X)NET üblich erfolgt der Anschluss eines AXIP/UDP Ports mit Hilfe des „attach„-Befehls. Die Syntax lautet:

```
attach <axip/axudp> <port> <count> {<opt>} <destIP> [<gwIP>]
```

Wobei für <axip/axudp> entweder AXIP oder AXUDP angegeben wird.
Der Port wird bei <port> angegeben <count> ist bei AXIP oder AXUDP immer 1!

Für <opt> können optional angegeben werden:

Option <opt>	Bedeutung
c	Beim Senden keine Checksumme berechnen und beim Empfang nicht prüfen
d<nr>	Angabe der UDP- Port- Nummer der Gegenstation
l<nr>	Angabe der lokalen UDP- Port- Nummer
p	Weitergabe von Parametern (wird für IPKISS.APL benötigt)

Die Destination IP-Adresse <destIP> ist die Adresse der AXIP-Gegenstation. Sofern diese Station nur über einen Router erreicht werden kann, muss die Adresse des Routers (oder Gateway) als <gwIP>] angegeben werden über den die Gegenstation erreicht werden kann.

Beispiel:

```
attach axip 5 1 c 192.168.44.14 192.168.44.65
# Dieser attach trägt eine AXIP- Verbindung zu 192.168.44.14
# über 192.168.44.65 auf Port 5 ein. Es findet keine
# Checksummen-Prüfung und Berechnung statt.
```

Nutzung der TNC4-UART-Schnittstelle

Das UART- Device welches vom Ethernet-Treiber installiert wird, bietet alle asynchronen seriellen Treiber an, die auch für die SCCs des TNC4 existieren. Auch die attach- Syntax ist dieselbe.

Wenn auf dem UART- Device serielle Treiber attached werden, muss das Terminal ausgeschaltet werden (Befehl: term 0)!

Die TNC4-UART kann maximal bis 115200 Baud betrieben werden.

Externe Zusatzprogramme für 3NET

Die folgenden Kommandos sind als ausführbare Programme in die RAM-Disk ladbar. Manche Befehle sind deshalb unter Umständen auf vereinzelten Knoten nicht vorhanden. Der Betreiber des Knotens entscheidet, welche Programme er in den Digi lädt und auf welche Kommandos verzichtet wird, um dadurch evt. RAM-Speicherplatz im TNC3 zu sparen. Welche extern ausführbare Befehle (oder auch Informationstexte) vorhanden sind, kann durch Eingabe von HELP abgefragt werden. Am Ende der Help- Kommandos wird eine Liste der externen Programme ausgegeben. Hierbei ist wiederum zu Unterscheiden zwischen ‚Externals", die nur dem SYSOP zur Verfügung stehen und Kommandos, die User nützen können.
Letztendlich entscheidet der Sysop selbst, welche externen Programme er in den Digi aufnehmen will und wer Zugriff auf die Kommandos haben soll. Die Programmfiles werden binär in die RAM-Disk geladen. Hier entscheidet sich durch die Programm-Endung, ob nur der Sysop auf das Programm zugreifen kann, oder ob es jeder User nutzen kann.
Die Vergabe des Programmnamens kann durch den Sysop ebenfalls frei gewählt werden. Entscheidend ist jedoch die Dateiendung: :XTP (eXTernal Public = für alle nutzbar) oder als .XTS (eXTernal Sysop = nur durch Sysop).

BLINKD

Hintergrundprozess der die LEDs des 3Net-Master TNCs blinken läßt. Für Digis ohne angeschlossenes Terminal kann damit die Betriebsbereitschaft des Digis angezeigt werden.
Verwendung: In der AUTOEXEC.NET als Hintergrundprozess:

```
start blinkd [Millisekunden [LED-Nr]]
```

Optional kann die Frequenz in Millisekunden angegeben werden. Der Standardwert ist 50 ms. Die LED-Nr gibt an bis zu welcher LED „geblinkt„ werden soll. Da einige Digibetreiber die LED-Ausgänge der SCC3 für Steuerungszwecke verwenden, können sie mit diesem Parameter das Blinken auf die LEDs 0 bis 3 (SCC1 und SCC2) beschränken. Beispiel:

```
start blinkd 50 4
```

Blinkt nur auf den CON- und STA-LEDs von SCC1 und SCC2. Die LEDs der SCC3 können für Steuerzwecke verwendet werden.

FLASHCPY

Flashcopy dient zum binären Auslesen der „laufenden„ Software für Flash-EPROMs. Damit kann jeder, der sich ein Update der Software besorgen möchte, die aktuell laufende Software direkt vom entsprechenden TNC3-Digi herunterladen.

LS

Mit LS kann in der Knotenebene das Directory der RAM-Disk abgerufen werden. Das (List-Short)-Kommando gibt das Inhaltsverzeichnis im RAMs in Kurzform aus. Die Ausgabe einer ausführlichen Liste erfolgt mit:

```
LS - L
```

MESZ

Vollautomatische Umstellung von mitteleuropäischer Sommer- und Winterzeit. Das Programm wird ohne Parameter aufgerufen, berechnet die Umstellungszeitpunkte im laufenden Jahr und stellt dann aufgrund des Systemdatums die Sommer- und Winterzeit ein.

Die Verwendung von MESZ ist in Verbindung mit NTPDATE interessant: NTPDATE holt sich die exakte Uhrzeit in UTC von einem Zeit-Server. Der Server liefert allerdings keine Informationen über Winter- und Sommerzeit. MESZ berechnet die Umstellungszeitpunkte selbst und stellt dann die Uhr vollständig richtig.

Beispiel:

```
=>mesz

Umstellung auf Sommerzeit am 27.03.05
Umstellung auf Winterzeit am 30.10.05

Wir haben jetzt Sommerzeit. Eingestellt ist Sommerzeit. OK.
```

OUT

Mit Hilfe des OUT-Befehls können einzelne Portbits des TNC3 zu Fernsteuerzwecken ein- und ausgeschaltet werden.

Syntax:

```
OUT [<port>] <PortBit> [<Value>]

<Port> = a | b
<PortBit> = 0..15
<Value> = 0..1
```

Die Angabe des Prozessorports (a oder b) ist optional. Sofern kein Port angegeben wird, wird Port a verwendet.

Beispiel:

```
=>OUT a 15 0
```

schaltet die CON-LED des TNC3 ein.

Beispiel:

```
=>OUT a 15
PA15 = 0
```

gibt den momentanen Zustand (0 oder 1) des Ausgabeports an.

Folgende I/O-Leitungen koennen über den OUT-Befehl ferngesteuert werden:

PortBit	SCC	Modem	Pin	Modemstecker
0	2	RXD	15	J13
1	2	TXD	13	J13
2	2	RCLK	19	J13
3	2	TCLK	17	J13
4	2	CTS	9	J13
5	2	RTS	11	J13
6	2	CD	7	J13
7	2	BEEPER		
8	3	RXD	15	J10
9	3	TXD	13	J10

PortBit	SCC	Modem	Pin	Modemstecker
10	3	RCLK	19	J10
11	3	TCLK	17	J10
12	3	BEEPER		
13		CON LED	3	(Layout CON 0 LED)
14		STA LED	3	(Layout STA 0 LED)
15		CON LED	1	

Tabelle 6: Von OUT ansteuerbare I/O-Ports

POKE

Durch Poke lassen sich die durch POSTMORT registrierten Ereignisse zurücksetzen.
Eingabe:

```
POKE W 100 0
```

Achtung: Wird ein falsches Poke-Kommando an den Knoten gesendet, kann der Digi zum Absturz gebracht werden - also Vorsicht, sonst ist ein „Digipeater-Besuch„ fällig.

UPDATE

Update vereinfacht das 3NET FLASH- „Updaten„ ganz wesentlich.

```
=> UPDATE

2 Flash EPROM(s) detected. Type: 29F010
Please send EPFLASH.ABS (BIN) file
```

Nach der Eingabe des update Befehls kann direkt die neue Flash-Version zum Digi geschickt werden **(Datei EPFLASH.ABS)**. Direkt nach dem Empfang werden die Flash-EPROMs programmiert und der Digi mit der neuen Software gestartet.
Selbstverständlich wird vor dem Programmieren die Korrektheit der Daten geprüft (CRC). Tritt beim CRC oder während der Übertragung des Updates ein Fehler auf, so wird der Vorgang abgebrochen und der Digi läuft mit der alten Software normal weiter.

KISS/SMACK

KISS entzieht dem TNC die Abarbeitung des AX.25-Protokolls und Befehlssatzes; 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 die vollständige Kontrolle über die auf der HF-Seite uebertragenen HDLC-Frames.

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 uebertragenen Frames zu begrenzen. Jeder Frame beginnt und endet mit einem speziellen FEND (Frame End) Zeichen, ganz aehnlich, wie HDLC-Frames. Die Bildung und Überpruefung der CRC-Pruefsumme wird dabei noch dem TNC ueberlassen. RS-232-Handshake wird nicht verwendet, daher genuegt 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, einen Stopbit und ohne Parity gesendet.
Ein innerhalb eines Frame vorkommendes FEND-Zeichen wird in die Sequenz FESC-TFEND uebersetzt, analog dazu wird ein FESC-Zeichen als FESC-TFESC uebertragen. Der jeweilige Empfaenger (Rechner oder TNC) puffert die Zeichen, ein FEND bedeutet das Ende eines Frame. Der Empfang eines FESC-Zeichen schaltet den Empfaenger 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 regulaeres Zeichen betrachtet. Diese Uebertragungsart mag auf den ersten Blick kompliziert erscheinen, ist aber sehr einfach zu implementieren und macht wenig Synchronisationsprobleme. Sie ist uebrigens identisch zu „SLIP„ (Serial Line IP).
Dem TNC bleibt nur noch, die KISS-Frames in HDLC-Frames umzuwandeln, und umgekehrt, sowie die Frequenz zu ueberwachen 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. Sein hoeheren vier Bit enthalten die Portnummer, die niederen vier Bit die Befehlsnummer. Folgende Befehle sind definiert:

Befehl	Funktion	Bemerkung
0	Daten-Frame	Der Rest des Frame besteht aus Daten
1	TXDELAY	Das naechste Byte gibt die PTT/Datensendebeginn Verzoegerungs-Zeit an. (in 10 ms Schritten). Default ist 50 (d.h. 500ms).
2	P	Das naechste Byte gibt den Persistence-Wert p an (0 - 255).
3	SlotTime	Das naechste Byte ist das Slot-Intervall (in 10 ms Schritten). Default ist 10 (d.h. 100ms).
4	Txtail	Das naechste Byte ist die Zeit, die der Sender nach dem Senden der Prmme noch hochgetastet bleibt (in 10 ms Schritten).

Befehl	Funktion	Bemerkung
5	FullDuplex	Das naechste Byte ist 0 für Halb-Duplex, ungleich 0 für Vollduplex. Default ist 0, d.h. Halb-Duplex.
6	SetHardware	Spezifisch für jeden TNC.
FF	Return	Verlassen des KISS-Modus. (Ueblicherweise wird in den Terminal-Modus zurueckgeschaltet)

Tabelle 7: Grundlegende KISS-Frames

Die urspruengliche Idee eines einfachen Rechner/TNC-Protokolls hatte Brian Lloyd, WB6RQN. Phil Karn, KA9Q, einer der „Vaeter„ des AX.25-Protokolls, organisierte die Spezifikation und legte am 6. August 1986 eine erste KISS-Version vor.
Quelle: The KISS TNC: A simple Host-to-TNC communications protocol Mike Chepponis, K3MC, Phil Karn, KA9Q

KISS mit Prüfpolynom: SMACK

SMACK ist eine Erweiterung des KISS-Modes, bei der an jedes Daten-Frame eine 16-Bit Checksumme angehängt wird. Diese Erweiterung wurde notwendig, als sich herausstellte, daß bei PC und UNIX-Workstations Zeichenverluste beim Empfang serieller Daten auftraten.

Zusammenfassung der TNC3-KISS Kommandos

Name	Beschreibung	Parameter RSKISS	Datenfeld RSKISS [Byte]
Daten folgen		0	0-328
TxDelay	TxDelay	0x1	1 [10 ms]
Persistenz	0 bis 255	0x2	1
Slottime	Slottime	0x3	1 [10 ms]
TxTail	TxTail [ms]	0x4	1
Vollduplex	0=Aus, 1=Ein, n=PTT Hold Time	0x5	1 [s]
DAMA	0=Aus, 1= Ein	-	1
Calibrate	Trägersignal-Aussendung	0x8	1 [min]
Duo-Baud	Software-Sendeverriegelung. 0=Aus, 1= Ein	0x9	1
Reset	TNC-Reset auslösen	0xFF	1
Frames gesendet	Meldung im DAMA-Mode, daß alle Daten ausgesendet wurden	-	1

Tabelle 8: Erweitertes TNC3- KISS

HighSpeedBus

Der HighSpeedBus ist eine einfache und schnelle Vernetzungsoption für den TNC3-Knoten.

Der High-Speed-Bus und (X)NET

Ein (X)NET-Knotenrechner besteht aus einem Master-Rechner (TNC3/TNC4 mit 1MB RAM), und mehreren Dual-Link-Controllern (DLCs). Die DLCs sind minimal ausgestattete TNC3s, sie sind im Wesentlichen für die Aussendung der AX-25-Rahmen (Medium-Access-Control-Ebene MAC) zuständig. Alle Karten werden über den Hochgeschwindigkeitsbus (HS-Bus) vernetzt.

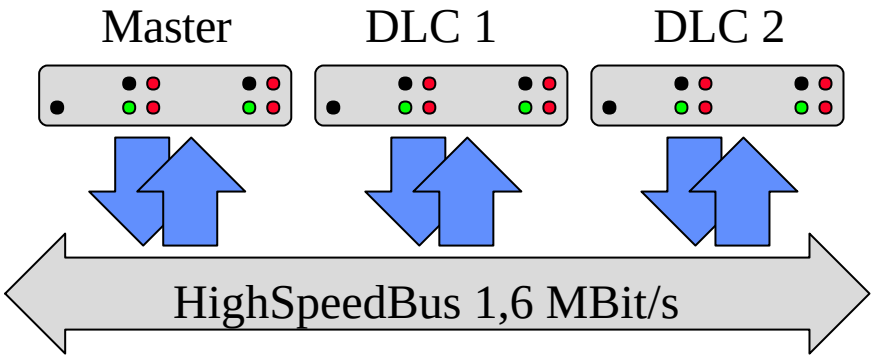


Abbildung 4: Logische Busstruktur

Die maximale Übertragungsrate auf dem HS-Bus liegt bei 1,6 MBit/s (bedingt durch MC68302).

HighSpeedBus-Hardware

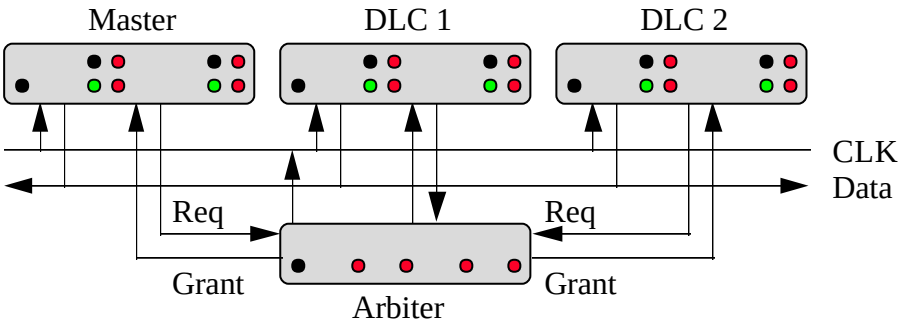


Abbildung 5: Verdrahtung des Busses

Der HighSpeedBus besteht aus einfacher Elektronik. Auf dem TNC3 sieht die Verschaltung des Busses mit dem entsprechenden SCC-Port so aus:

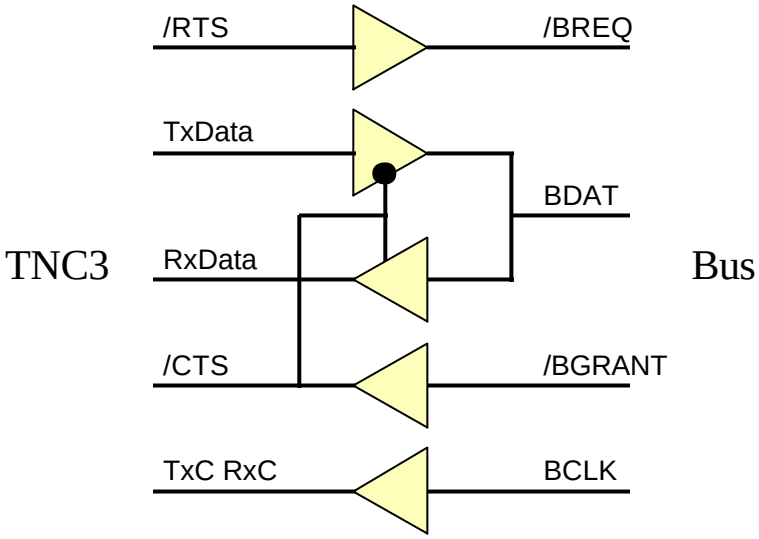


Abbildung 6: HighSpeedBus -Anschaltung

Ein Arbiter ist zum Betrieb des HighSpeedBusses notwendig. Diese Schaltung regelt die Buszugriffe und verhindert Kollisionen.

Das KISS-Busprotokoll

Das primitive Busprotokoll ist dem TNC-KISS-Protokoll nachempfunden. Die KISS-Daten sind jedoch in einen HDLC-Rahmen gepackt. Dadurch können alle Zeichendopplungen (FESC, TFEND,…) aufgrund der Transparenz der HDLC-Rahmen entfallen. Übertragungsfehler sind wegen kurzer Leitungen und der Kollisionsfreiheit selten. Falls Übertragungsfehler dennoch auftreten werden sie mit Hilfe der automatisch generierten Prüfsumme der HDLC-Rahmen (CRC) zu 99,99% erkannt.
Die Adressen sind länger kodiert als bei KISS. Die Adreßerkennung übernimmt der MC68302-RISC-Controller vollständig.

Flag	DEST	SRC	FI	Data (optional)	CRC	Flag

Abbildung 7: KISS-Rahmen

Adreßfelder sind notwendig um die einzelnen DLC-Karten anzusprechen. Die DLC-Software (HSKISS.APL) ist in jedem TNC3 bereits standardmäßig enthalten. Gestartet wird diese Software durch die DIP-Schalterstellung 7. Da keine RS232-Baudrate eingestellt werden muß, können die drei Baudraten-Bits am DIP-Schalter zur Einstellung der DLC-Nummer verwendet werden. Achtung die Bits sind etwas „verdreht„:

DIP-Schalter:	1 2 3 4 5 6 7 8	DLC Nummer	(X)NET-Ports Modem1	Modem2
	0 0 0 0 0 1 1 1	0	0	1[1]
	1 0 0 0 0 1 1 1	2	2	3
	0 1 0 0 0 1 1 1	4	4	5
	1 1 0 0 0 1 1 1	6	6	7
	0 0 1 0 0 1 1 1	8	8	9
...				
	1 1 1 0 0 1 1 1	14	14	15

Durch die Verdrahtung der DIP-Schalter ergibt sich eine Invertierung:

```
1 = Open   = OFF = DIP-Schalter oben
0 = Closed = ON  = DIP-Schalter unten
```

[1] Bei der NET_XS-Version werden diese Port-Nummern schon für die Ports am Master-NC belegt.

Die (X)NET-Karte hat eine fest vorgegebene HighSpeedBus-Adresse:

1	0	0	0	0	0	0	0

Abbildung 8: Adresse des (X)NET-Master

Das FI-Feld ist zwei Bytes groß und hat folgendes Bitmuster

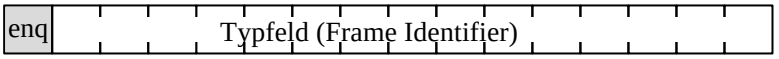


Abbildung 9: HS-Bus Typfeld

Die Kodierung des Typfeldes entspricht KISS

Typfeld	Name	Beschreibung	Größe [Bytes]
0x0	Daten	Daten folgen	0 bis 328
0x1	TxDelay	TxDelay [ms]	2
0x2	Persistenz	0 bis 255	1
0x3	Slottime	Slottime [ms]	2
0x4	TxTail	TxTail [ms]	2
0x5	Vollduplex	0=Aus, 1=Ein, n=PTT Hold Time [s]	1
0x6	DAMA	0=Aus, 1= Ein	1
0xD	Reset	DLC-Reset auslösen	1
0xE	Frames gesendet	Meldung im DAMA-Mode, daß alle Daten ausgesendet wurden	1
0x8001	TxDelay?		2
0x8002	Persistenz?		1
0x8003	Slottime?		2
0x8004	TxTail?		2
0x8005	Vollduplex?		1
0x8006	DAMA?		1
0x8007	Baudrate?	Modem-Baudrate [100 Hz]	2
0x8100	DLC Programmversion?	Zeichenkette	n

Tabelle 9: HighSpeedBus-KISS-Frames

Unterschiede zu KISS:
Alle Zeitparameter sind in Millisekunden angegeben, denn bei zukünftigen TRXen ist mit kleineren TxDelays und Slottimes zu rechnen. Die meisten numerischen Werte werden in zwei Bytes kodiert, 68000-Format. „Escape„-Sequenzen bzw. Zeichendopplung („Character Stuffing„) in den Daten entfallen.
Eine weitere KISS-Ergänzung: Mit Hilfe des Bits enq lassen sich auch Werte abfragen (enq = 1). Die abgefragten Werte werden im gleichen Format an die abfragende Station zurückgesendet.
Die Vorteile gegenüber dem TheNet-Token-Ring liegen auf der Hand:

- 20 bis 40 mal höhere Baudraten auf dem lokalen Bus, wesentlich höherer Durchsatz, höhere Dynamik
- wesentlich geringere Antwortzeiten da keine Zugriffskontrolle mittels Tokens, Verzögerungszeiten im Mikrosekundenbereich
- Kollisionsfreiheit garantiert durch Arbiter; die Bitfehlerrate auf dem Bus ist gering
- Fehlererkennung durch Verpackung aller KISS-Rahmen in HDLC-Rahmen mit CRC-Prüfsumme
- Keine RS232-Pegelwandler notwendig

- Prüfsummenberechnung und Prüfung durch den 68302-RISC-Communications-Controller, keine CPU-Belastung
- Datenaustausch über den Bus belastet die einzelnen DLCs nicht (optimale Unterstützung durch die Adreßerkennung des MC68302 RISC-Communications-Controller).
- Empfang und Senden von KISS-Rahmen auf dem Bus läuft vollständig im Hintergrund; es sind keine Interrupts notwendig. (Empfangen und Senden steuert der RISC-Communications-Controller zusammen mit dem Arbiter)

Die obigen Punkte sprechen für sich. Es sei trotzdem erwähnt, daß es nicht möglich ist mit Hilfe eines parallelen Busses die obigen Leistungsmerkmale an Durchsatz und Antwortzeiten bei gleichzeitig minimaler Prozessorbelastung zu erreichen, es sei den mit Hilfe (teurer) intelligenter Spezialhardware. (1,6 MBit/s entspricht 200 KBytes/s => für die programmgesteuerte Übertragung eines Bytes stünden nur 5 µs zur Verfügung (dies ist mit programmgesteuerter paralleler Übertragung nicht zu erreichen, da auch noch das „Handshaking„ per Programm zu regeln ist), der Prozessor wäre während der Busdatenübertragung voll ausgelastet, es wäre demnach hier keine Parallelarbeit des Prozessors möglich).

HighSpeedBus-KISS-Software

Programme für den HighSpeedBus:

HSKISS.APL

HighSpeedBus-DLC-Software. Dieses Programm ist in jedem TNC3 schon enthalten und ermöglicht es jeden TNC3 am HighSpeedBus als DLC (Dual Link Controller) zu verwenden. Die HighSpeedBus-Portnummern werden direkt an den DIP-Schaltern für die RS232-Baudrate eingestellt.

KTST.APL

KTST.APL ist ein HighSpeedBus-Testmonitor. Das Programm wird auf der Master-Karte gestartet und zeigt alle angeschlossenen HighSpeedBus-DLCs und deren Konfiguration an. Es eignet sich sehr gut um die Funktionsfähigkeit des Busses zu prüfen und die eingestellten Kartenadressen der DLCs anzuzeigen. Das Programm macht sogar die Modem-Baudraten der einzelnen DLC-Ports sichtbar.

```
r:>ktst
======================================================================

Kiss - HighSpeedBus - Testmonitor

        (b)roadcast: Busteilnehmer anzeigen
        (k)iss Konfiguration anzeigen
        (q)uit

======================================================================
```

KTST.APL ist auf der DIGI-Betriebsdiskette und muß von dort geladen werden.

STRESS.APL

STRESS.APL ist ebenfalls ein Testprogramm für den HighSpeedBus. Beim Aufruf wird die Nummer des DLCs angegeben der getestet werden soll. Auch im Dauerbetrieb dürfen keine Fehler auftreten.

```
r:>stress 1
*** HighSpeedBus - Stress-Test ***


Taste...


Statistik DLC [01]:
        Fehler (gesamt) :        0
        Pakete (gesamt) :        0
        Bytes  (gesamt) :        0
        Bitfehlerrate   :        0 E-8
        Pakete/s        :        0
        Bit/s  (Netto)  :        0

        Discarded Frames:        0
        Abort Sequences :        0
        CRC-Errors      :        0
        Retransmissions :        0
        Non matched Adr.:        0
```

RS232 Token Ring

Der Token-Ring ist eine ringförmige Vernetzung von TNC über deren serielle V.24 (RS223) - Schnittstelle. Hierbei hat ein TNC genau zwei Verbindungen: eine zu seinem Vorgänger und eine zu seinem Nachfolger. Diese Vernetzung wurde 1991 erfunden, um über eine serielle PC-Schnittstelle mehrere TNC und damit mehrere Funkports bedienen zu können.

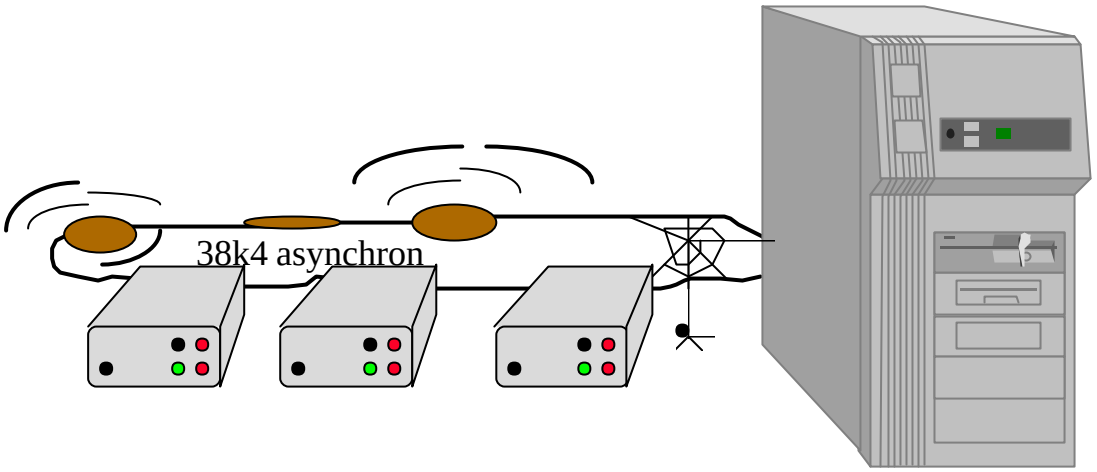
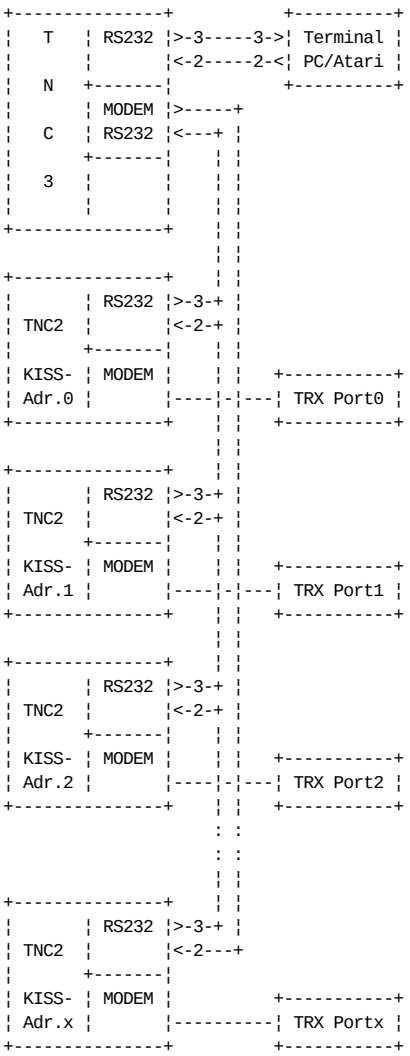


Abbildung 10: RS232-Token-Ring

Um die Sendeberechtigung auf dem Ring zu verteilen, wird ein „Token„ verwendet, welcher vom Master-Rechner erzeugt und an die TNC weitergegeben wird. Die Daten werden als abgewandelte KISS-Frames mit Portnummer über den Ring übertragen. Nachdem ein TNC den Token erhalten hat, darf er seine empfangenen Daten auf den Ring legen. Danach gibt er den Token an seinen Nachfolger weiter.



Token-Ring: Beispiel einer TNC-Verdrahtung

Bis zum Erscheinen des TNC3 war die maximale Baudrate des Token-Ring auf 38400 Baud begrenzt. Dadurch konnten nur wenige Ports oder nur langsame Linkstrecken aufgebaut werden. Ein größeres Problem war, dass oft Zeichen- und Tokenverluste auftraten - was zu Übertragungsfehlern führte. Mit dem TNC3 im Token-Ring sind einige der alten Probleme behoben:

?? keine "Token-Recoveries", keine "Bad Frames" es ist ein fehlerfreier Token-Ring-Betrieb möglich

?? der TNC3 bedient gleich zwei Ports, die Token-Durchlaufzeiten werden reduziert (halb soviel TNCs).

?? die Portnummern lassen sich am DIP-Schalter des TNC einstellen d.h. es wird nur ein EPROM für alle TNC3s (keine EPROM Patches wie beim TNC2) benötigt.

?? der TNC3 mit Token-Ring-Baudraten von 1200 bis 115200 Baud betrieben werden. Ein Ring der ausschliesslich aus TNC3 besteht, kann mit einer Baudrate von 115200 Baud betrieben werden. Falls ein externer Takt verwendet wird, kann die Token-Ring-Baudrate stufenlos bis zu 370000 Baud (370 kBaud) eingestellt werden

?? Funkbaudraten bis 512 kBaud möglich, die max. Baudrate über Funk ist nicht mehr durch den TNC2 auf 19k2 begrenzt

?? Das optionale SMACK-CRC Prüfpolynom kann Übertragungsfehler sicher entdecken

TRKISS auf dem TNC3

TRKISS.APL ist die Token-Ring Software für den TNC3, die vollständig zur entsprechenden TNC2-Software kompatibel ist, jedoch einige Erweiterungen aufweist.

TRKISS-Portnummer

Die Portnummern werden beim Starten des Programms vom Programm DIP-Schalter gelesen. Dabei werden nur die Bits 1-3 der Programmnummer zur Bildung der Portnummer verwendet:

Programm-DIP	KISS-TNC-Port1	KISS-TNC-Port2	trkiss
16	0	1	DIP16.APL
18	2	3	DIP18.APL
20	4	5	DIP20.APL

Achtung: Bei Verwendung von SRP wird die Portnummer vollautomatisch in der Ringreihenfolge der TNC eingestellt! Die Stellung des DIP-Schalters wird ignoriert.

Token-Ring Baudrate

Die Baudrate wird wie üblich an den DIP-Schaltern des TNC:

DIP 1 2 3	Baud
0 0 0	230400
0 0 1	2400
0 1 0	4800
0 1 1	9600
1 0 0	19200
1 0 1	38400
1 1 0	57600
1 1 1	115200

TNC3-Erweiterungen

Duplex-Nachlaufzeit

Der Parameter Fulldup kann im Wertebereich zwischen 0 und 255 angegeben werden (Bisher nur 0 und 1). Damit kann im Vollduplex-Modus ein verzögertes Abfallen von PTT parametriert werden. Dabei geben Werte >= 2 die PTT-Haltezeit in Sekunden an.

Fulldup	Bedeutung
0	Halbduplex
1	Volldupex
2	Vollduplex PTT-Hold-Time 2 s
...	
n	Vollduplex PTT-Hold-Time n s
...	
255	Vollduplex PTT-Hold-Time 255 s

Erweiterte KISS-Kommandos

FEND <port> 8 <min> FEND
Calibrate Nr. 8: Der TNC tastet PTT des angegebenen Ports das
Feld <min> gibt die Tastdauer in Minuten an.

FEND <port> 9 <on> FEND
Duobaud Nr. 9: on = 1 Schaltet die Sendeverriegelung fuer beide
TNC3-Ports ein, on = 0 beendet die Sendeverriegelung.

SMACK

Die Definiton des SMACK-Checksummen-KISS wurde nun auch für das Token-Ring-KISS übernommen. SMACK wird durch den Hostrechner aktiviert, indem dieser bei Sendedaten das 7. Bit der Portnummer auf 1 setzt. D.h. aus der Portnummer 0x00 wird beispielsweise die Nummer 0x80 (s. SMACK-Beschreibung). Daraufhin wird SMACK aktiviert und der TNC3 versieht alle folgenden empfangenen Daten mit einem zusätzlichen SMACK-CRC.
Token-Ring-SMACK kann im Moment nur mit entsprechender Digisoftware ((X)NET) aktiviert werden.

Hardware-Watchdog

Die TRKISS-Software wird nun durch den eingebauten TNC3-Hardware-Watchdog überwacht. Wenn (aus welchen Gründen auch immer) der TNC3 eine Minute lang keinen Token empfängt führt die Software automatisch einen Hardware-Reset des TNC3 aus.

Installationstip

Wird beim Starten des Programms ein Parameter in der Kommandozeile angegeben, meldet sich das Programm folgendermassen:

```
r:>c:trkiss x

TNC3 Token Ring KISS Mode Version 1.14 Jan 10 1998 (C) DL1GJI

Modem 1:  1200 Baud Adresse 0
Modem 2:  1200 Baud Adresse 1
```

Damit koennen die Modem-Baudraten des TNC3 und die aktuell eingestellten Port-Adressen überprüft werden.

Serial-Ring-Protocol

Das Serial-Ring Protokoll dient wie der oben beschriebene Token-Ring oder das FlexNet-6Pack-Protokoll zur Vernetzung von TNCs über die serielle Schnittstelle. Die TNCs sind ebenfalls ringförmig vernetzt - nur das Kommunikationsprotokoll wird ausgetauscht. SRP fasst die Vorteile der beiden Implementierungen (6Pack und Token-Ring) zu einem neuen Protokoll zusammen.

?? Automatische Konfiguration der Portnummern
?? Geringer Protokoll-Overhead
?? Vollduplexfähigkeit
?? Automatische Modem-Baudratenerkennung
?? Geringere Hostrechnerbelastung
?? Vernetzung von Hostrechnern möglich
?? DAMA-Unterstützung
?? Integrierte Prüfsumme
?? Geringere Verzögerungszeiten

Der Aufbau der Frames ist an KISS angelehnt. Der wesentliche Unterschied zum Token-Ring besteht jedoch darin, dass auf den Token komplett verzichtet wurde.

Ab der Slave-Software Version 1.16 kann TRKISS.APL automatisch zwischen Token-Ring-KISS und SRP umschalten.

Vergabe der Portnummern

Die Portnummern werden bei SRP automatisch vergeben. Dabei werden die Ports einfach in Senderichtung durchnummeriert. Dies hat den Vorteil, dass alle Slave-TNCs nun mit identischer DIP-Schalterstellung betrieben werden können.

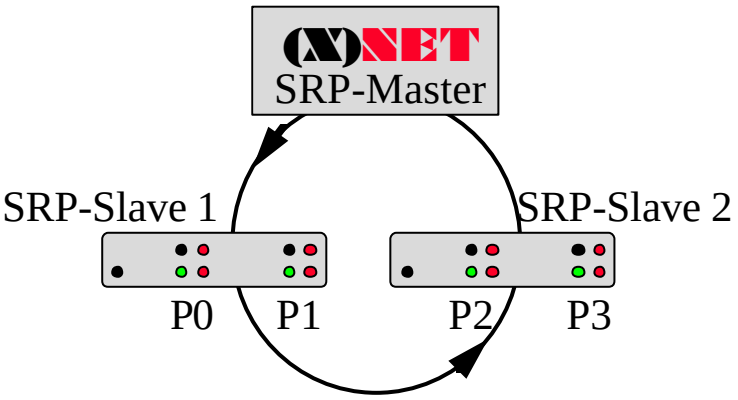


Abbildung 11: SRP: Vergabe der Portnummern

Da die Token-Ring-Software die Portnummer anhand der DIP-Programmschalterstellung ermittelt, kann sich die Portnummerierung bei einer Umschaltung zwischen TRKISS und SRPM ändern.

Protokollbeschreibung

Initialisierungs-Frame

Bei SRP wird kein Token mehr als Sendeberechtigung für den (Slave-) TNC verschickt. Der TNC darf jederzeit seine empfangenen Daten auf dem Ring senden. Er darf dabei natürlich keine Frames die er gerade im Ring weiterleitet, unterbrechen.
Statt des Tokens gibt es jedoch ein Initialisierungs-Frame, welches vom SRP-Master ca. jede Sekunde einmal im Kreis geschickt wird. Das Initialisierungsframe besteht aus folgender Bytesequenz:

0xC0 0xFF 0x10 <Ring-Adresse> 0xC0

Der empfangende TNC übernimmt die Ring-Adresse als seine eigene und leitet das Initialisierungsframe weiter, wobei er die <Ring-Adresse> um eins erhöht. Der Master sendet das obige Frame mit der Ring-Adresse 0 aus. Wenn er das Frame wieder empfängt, weiß er aufgrund der empfangenen Ring-Adresse wie viele Ports im Ring verfügbar sind.

Daten-Frame

Daten -Frames haben den folgenden Aufbau:

0xC0 0x0 <Daten> <CRC16> 0xC0

Sie müssen bei SRP mit SMACK-CRC gesendet werden. Gekennzeichnet wird das CRC-Frame durch gesetztes Bit 7 der Ring-Adresse. Der CRC wird nur über den Datenteil des Frames errechnet. Der CRC wird jedoch je nach Ringadresse unterschiedlich vorinitialisiert.

Ring-Adresse	*CRC-Initialisierung*
0	0xe8c1
1	0x2800
2	0x2800
3	0xe8c1
4	0x2940
5	0xe981
6	0xe981
7	0x2940
8	0x2a80
9	0xea41
10	0xea41
11	0x2a80
12	0xeb01
13	0x2bc0
14	0x2bc0
15	0xeb01

Tabelle 10: CRC-Initialisierungswerte für SRP

Die zwei CRC-Bytes werden wie bei SMACK üblich an das Daten-Frame angehängt.

Steuerungs-Frame

Steuerungs-Frames haben den folgenden Aufbau:

0xC0 <Code> <Parameter> 0xC0

Es existieren die folgenden Steuer-Codes:

Code	*Beschreibung*
$00	Daten-Frame
$01	TXDelay (10ms)
$02	Persistance
$03	Slottime (10ms)
$04	TxTail (10ms)[2]
$05	Vollduplex (Nachlaufzeit in Sekunden)
$06	Host->TNC : DAMA Mode (0=aus) TNC->Host : Debug/Log Meldung folgt (*)
$07	HDLC Baudrate (Meldung an Host) (**)
$08	Host->TNC : Calibrate (Minuten) TNC->Host (nur DAMA Mode) : HDLC Kanalstatus (0=Kanal frei)
$09	Duo-Baud (TNC3): Gegenseitige Verriegelung der Modem-Ports des TNC3. Hat beim TNC2 keine Wirkung.
$0a	LEDD-Flag (Bit1=Con LED, Bit1=Status LED)
$0b	Reserviert
$0c	TNC Reset durchgeführt (Meldung an Host). Datenbyte enthält die Versionsnummer der Software ($75 = 117 = Version 1.17)
$0d	Host->TNC : TNC Reset durchführen
$0e	DAMA Frame gesendet (Meldung an Host)
$0f	DEBUG Mode (0=aus)

Tabelle 11: SRP-Codes

Die Parameter sind 1 Byte lang. Ausnahmen:

(*) Nach $06 folgt ein String mit einer Debug/Log Meldung
(**) Nach $07 folgt die Bitrate als 32Bit Wert, mit dem höchstwertigen Byte zuerst: 0, 0, HiByte, LoByte

[2] Nicht genutzt im TNC2, anstelle "TxTail" wird beim Runtertasten die Sendezeit für 8 Bytes verwendet.

AXIP KISS (TNC4e)

IPKISS ist eine Art Ethernet-Kiss-Mode, der nichts weiter macht, als empfangene AXIP-Frames über einen AX.25 Port des TNC4e auszusenden. Um bestehende Software nicht ändern zu müssen, wird jedem TNC4-Funkport eine eigene IP-Nummer zugewiesen. So ist es jeder AXIP-fähigen Software möglich über den TNC4e schnelle Linkstrecken aufzubauen. IPKISS ist mit den KISS Implementierungen RSKISS, TRKISS, HSKISS vergleichbar - nur findet der Transport der Daten über Ethernet statt.

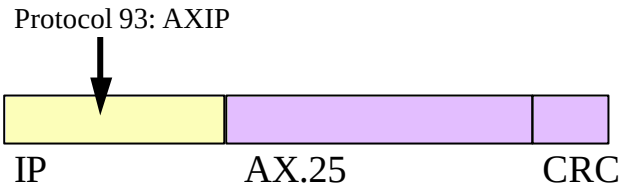


Abbildung 12: AXIP-Frame

Dabei wird das im RFC 1226 beschriebe Protokoll AXIP verwendet (RFC 1226: „Internet Protocol Encapsulation of AX.25 Frames„, B. Kantor). Einige Betriebssystem-Plattformen (Windows 95/98 und NT) erlauben es nicht AXIP-Frames zu versenden und zu empfangen. Um trotzdem mit diesen Plattformen kommunizieren zu können, muß statt AXIP das Protokoll UDP verwendet werden. Anhand der Portnummer 93 kann das AXUDP-Frame erkannt werden. Es hat den folgenden Aufbau:

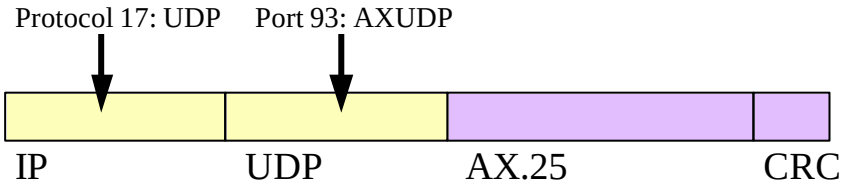


Abbildung 13: AXUDP-Frame

Das im AXIP enthaltene 16-Bit Prüfpolynom (CRC) dient zur Absicherung der AX.25 Dateninhalte. Somit können Verfälschungen oder Zeichenverluste beim Transport über das Internet entdeckt werden. Wenn man AXIP ausschließlich im lokalen Ethernet-Netz nutzt, ist diese Absicherung natürlich überflüssig, da jedes Ethernet-Frame ohnehin schon mit einem 32-Bit CRC-Verfahren geprüft wird.

IPKISS.APL

Um einen TNC4e im IPKISS-Mode zu betreiben wir das Programm IPKISS.APL benötigt. Beim Aufruf von IPKISS.APL können in der Kommandozeile die folgenden Parameter angegeben werden:

Syntax:

```
IPKISS [<opt>] <myip>      (AXIP|AXUDP) [<DestIP>]
```

<myip>	IP-Nummer des ersten TNC4-Ports. **Achtung**! Sie muss durch vier teilbar sein. Die drei weiteren Ports erhalten die nächst folgenden IP-Adressen.
AXIP	der TNC sendet und empfängt im AXIP Protokoll
AXUDP	der TNC sendet und empfängt im AXUDP Protokoll
<DestIP>	Adresse des Rechners, an den die AXIP oder AXUDP-Pakete geschickt werden sollen. Gibt man hier keine Adresse an, so werden die Pakete an die IP-Broadcast-Adresse

255.255.255.255 geschickt. Dies ist i.d.R nicht sinnvoll, da dann alle Stationen im lokalen Netz die Informationen mitlesen müssen und dadurch belastet werden.

<opt> Die Option -c erlaubt es, das im RFC 1226 beschreibene Prüfpolynom zu deaktivieren. Der CRC-Wert wird dann bei der Aussendung weder berechnet, noch beim Empfang gepüft. Die beiden CRC-Bytes werden trotzdem mitgeschickt, enthalten aber undefinierte Werte. Durch das Abschalten der CRC-Prüfung erhöht sich der erreichbare Durchsatz merklich!

Beispiel:

```
ipkiss -c 192.168.44.4 AXIP 192.168.44.1
```

Dieses Kommando versetzt den TNC4 in den AXIP-Modus. Die AXIP-CRC-Berechnung ist durch den Parameter „-c„ deaktiviert. Der TNC erhält für jeden seiner Ports eine separate IP-Nummer:

Port	IP-Nummer	Bemerkung
0	192.168.44.4	Modem 1
1	192.168.44.5	Modem 2
2	192.168.44.6	Modem 3
RS232	192.168.44.7	Serielle Schnittstelle

Mit diesen Nummern kann nun jeder Port des TNC4 getrennt voneinander angesprochen werden.

Die Portnummer 3 kann zum Betrieb der seriellen Schnittstelle des TNC4 im KISS-Mode verwendet werden. Die RS232-Baudrate wird hierbei durch die DIP- Schalter am TNC4 eingestellt. Mit Hilfe des Port-Befehls „port <x> mode 2“ kann zusätzlich die RMNC-Checksumme aktiviert werden.

Alle auf den Modemports empfangenen Daten werden an den Rechner mit der IP-Adresse 192.168.44.1 geschickt.

Parametrierung über AXIP

Anders wie bei den KISS-Modes ist es bei der AXIP-Protokolldefinition nicht vorgesehen Parameter zu übertragen. Um trotzdem Parameter wie TxDelay, Duplex, Persistence etc. setzen zu können werden beim TNC4 alle AXIP-Frames mit einer Länge von 4 Byte als TNC-Parameter aufgefaßt:

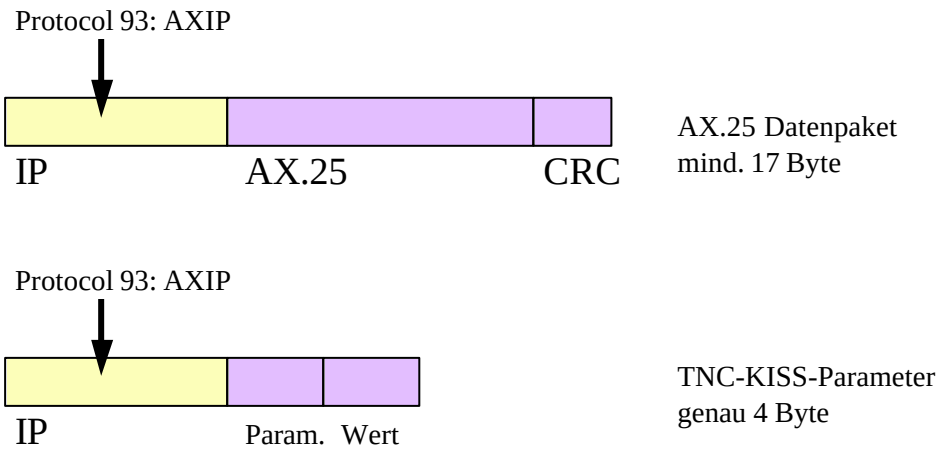


Abbildung 14: Parameterübergabe über AXIP/AXUDP

IPKISS verwendet dabei die folgenden Parameter-Kodierungen:

Name	Beschreibung	Parameter	Datenfeld [Byte]
Daten folgen		0	5-328
TxDelay	TxDelay	0x1	2 [ms]
Persistenz	0 bis 255	0x2	2
Slottime	Slottime	0x3	2 [ms]
TxTail	TxTail [ms]	0x4	2 [ms]
Vollduplex	0=Aus, 1=Ein, n=PTT Hold Time	0x5	2 [s]
DAMA	0=Aus, 1= Ein	0x6	2
Calibrate	Trägersignal-Aussendung	0x8	2 [ms]
Duo-Baud	Software-Sendeverriegelung. 0=Aus, 1= Ein	0x9	2
Reset	TNC-Reset auslösen	0xD	2
Frames gesendet	Meldung im DAMA-Mode, daß alle Daten ausgesendet wurden	0xE	
TxDelay-Abfrage		0x8001	
Persistenz-Abfrage		0x8002	
Slottime-Abfrage		0x8003	
TxTail-Abfrage		0x8004	
Vollduplex-Abfrage		0x8005	
DAMA-Abfrage		0x8006	
Baudrate-Abfrage	Modem-Baudrate[100 Hz]	0x8007	2
IPKISS Programm-Versions-Abfrage	Zeichenkette	0x8100	

Tabelle 12: AXIP/AXUDP-Kommandos

Einschränkungen

IPKISS ist in erster Linie zur Koppelung von (X)NET-Digis mit einem TNC4 über ein lokales Ethernet-Netz gedacht. Daher wurden folgende Vereinfachungen vorgenommen:

IP

IP ist nur minimal implementiert. Es wird keine „IP-Fragmentation„ unterstützt. Somit müssen alle zwischen den beiden Stationen liegende Netze eine MTU 350 Bytes besitzen! Es werden ebenfalls nur IP-Pakete der Version 4 verstanden (die derzeit übliche Version). Die verwendete Protokollnummer ist 93 (wie in RFC1226 vorgeschrieben). Die TTL ist fest auf 16 eingestellt. IP-Optionen werden keine verschickt.

ICMP

Die ICMP Echo Request Message (Ping-Kommando) wird ausgewertet und ein Echo Reply zurueckgeschickt. Andere ICMP-Messages werden weder verschickt noch ausgewertet.

Anhang

Abbildungsverzeichnis

Tabellenverzeichnis