

TheNetNode V1.79

Modifiziert von Oliver Kern DAA531

Hier ist eine Liste mit den Erweiterten und neuen Befehlen.

(AX)IPX - Route setzen oder löschen

Modifiziert:

Dieses Modul habe ich nun automatisiert. Das heißt eingehende neue Routen werden automatisch in die interne Routing Tabelle aufgenommen. Diese werden auch als AUTO-ROUTE bezeichnet. Ändert sich eine IP-Adresse oder UDP-Port, wird das in der Internen Routing Tabelle aktualisiert.

Neu:

Dann habe ich eine Protokoll-Erkennung eingebaut. Damit ist es möglich das sich Routen automatisch an das Node anbinden können (nur auf dem AX25IP-Port).

Flexnet- bzw. Netrom-Protokoll werden automatisch in die LINKS Tabelle eingetragen.

Außerdem können User aus dem Internet sich in das Node einloggen.

Wird eine Route nach einer „gewissen Zeit“ (wird auch als „PINGTIME“ bezeichnet) nicht mehr gelesen, wird diese Route aus der internen Routing-Tabelle gelöscht. Das gilt aber nur für Auto-Routen mit oder ohne Linkanbindung.

ACHTUNG:

AX25IP.CFG wird nicht mehr unterstützt!!! Alle Einträge in der AX25IP.CFG müssen einmal auf der Konsole ausgeführt werden. Mit SP kann man die eingegebenen Daten abspeichern.

Folgende Befehle stehen zur Verfügung:

Den UDP-Port ändern, wo unser TNN-Knoten lauscht:

(Zu beachten wäre, das der AX25IP-Port eingeschaltet sein muss, sonst keine Änderung.)

AX U 93

Loglevel ändern:

AX L 4

```
loglevel 0 - no output
loglevel 1 - config info only
loglevel 2 - major events and errors
loglevel 3 - major events, errors, and AX25 frame trace
loglevel 4 - all events
```

Die Log-Informationen werden in einer Datei ax25ip.log abgespeichert.

Eine AX25IP-Route eintragen:

UDP-Modus:

AX R + DIG531 DIG531.DYNDNS.ORG UDP 93

IP-Modus:

AX R + DIG531 DIG531.DYNDNS.ORG

Eine AX25IP-Route austragen:

AX R - DIG531

Unsere interne Routing-Tabelle:

JO61QH:DIG531> AXIP-Routes:(72/90)

Alias-:Call-----IP-----Mode-Port--PMode---PTime---RX-----TX---Online-
Riesa: DNO531 213.146.112.94 UDP 93 INP 298 1296KB 876KB 19h,0m

AXIP-Routes:(72/90)

```

| |
| |
| | max. Anzahl AXIPR-Routen
| | aktuelle Anzahl der AXIPR-Routen

```

Alias = Aliasname vom Linkpartner

Call = Rufzeichen vom Linkpartner/User

IP = Aktuelle IP-Adresse vom Linkpartner/User

Mode = Protokoll, UDP oder IP.

Port = TX-Port vom Linkpartner/User.

PMode = Protokollanforderung unseres Linkpartners.

PTime = Pingtime, brauchen wir für die Aktualisierung der DNS-Adressen.

RX = empfangene UDP-Packete.

TX = gesendete UDP-Packete.

Online = Connect Zeit

Mit dem PArparameter „PINGTIME“ kann man die Aktualisierung der AXIPR-Routen steuern.
(Weitere Information siehe PArparameter „PINGTIME“)

Es besteht die Möglichkeit die interne Routing-Tabelle als HTML darzustellen. Möchte man dies gerne tun, muss man in der TNN179.PAS das richtige Verzeichnis setzen und im TNN den PArparameter **HTML-STAT** einschalten.

Die Aktualisierung der HTML-Datei „rstat.html“ erfolgt aller 1.Minute.

Außerdem kann man das Aussehen der HTML-Statistik nach seinen wünschen anpassen.

Änderungen sind in der Datei „rstat.css“ vorzunehmen.

(BE)ACON - Bake ein/ausschaltenNeu:

Statusbake mit interessanten Informationen für User oder Sysop.
Die Aussendung der Bake kann man mit dem PArameter Statusbake steuern.
Der Standartwert ist 300 (5 Minuten), dass heißt aller 5 Minuten wird die Bake ausgesendet.

(BE)ACON port mins metric call oder alias via call

Port = Portnummer
Mins = Intervall in Minuten
metric = Erweiterte Baken-Informationen
0 = Keine Metric
1 = Metric
2 = Zusätzliche STAT
3 = Statusbake (neu)

Statusbake einstellen:

(Beispiel für Port 0,Intervall 60, und Statusbake 3)

BEACON 0 60 3 IDENTAusgabe auf dem Monitor

```
T0: fm DNO531 to STATUS UI^ pid F0 – 21.03.04 21:26:13
Links: 32, Convers: 23, Dest/Nodes: 272, Buffers: 19823, Runtime: 44m,55s
DIG531:3 DNO531:28
```

Die Statusbake hat folgende Informationen:

Links = aktuelle Anzahl der L2-Links im Knoten
Convers = aktuelle Anzahl der Convers-User im Knoten
Dest/Nodes = Anzahl erreichbare Dest/Nodes
Buffers = Freie Buffers
Runtime = Laufzeit des Programms
Rufzeichen = alle Linkpartner mit Laufzeitangabe
(außer links ohne Laufzeitmessung)

Diese Statusbake ist erst in der 179mh02 verfügbar!!!

(CONV)ers - Schaltet um in den Conversmodus

Neu:

Ab Version 179mh02 gibt es kleinere Neuerungen im Convers.

1. Jeder Benutzer kann einen Nickname setzen.

Der Nickname wird in einer internen Tabelle gespeichert und erst nach beenden von TNN auf die Festplatte gespeichert, Dateiname ist nickname.txt.

Bei Neustart von TNN wird die Datei nickname.txt einmal eingelesen.

Es werden alle Nicknamen im CONVERS-NETZ eingetragen/aktualisiert.

Das geniale, jeder User egal welche TNN Node er benutzt für Convers, wird beim Login der Nickname gesetzt.

Ist User X eingeloggt im Convers und versucht sich ein weiteres mal in den Convers einzuloggen, ist das nicht mehr möglich !!!

2. Convers-Hostname kann man ändern

Es stehen 10 Zeichen für den Convers-Hostname zur Verfügung.

Wichtig, bei einen LINK Eintrag das Node Call benutzen, nicht den Convers-Hostname verwenden.

Der Convers-Hostname vom Linkpartner wird später automatisch auf den richtigen Namen aktualisiert/geändert.

Wichtig, Austragen des Linkpartner geht nur mit Node Call, Convers-Hostname geht nicht !!!

3. Zusätzlichen CTEXT einbinden.

Wer es brauch, kann noch einen zusätzlich CTEXT beim Login in den Conversmodus einbauen.

4. Passwortschutz

Konfiguration:

Einen Nickname setzen:

(Eingabe im Conversmodus)

/Nickname Olli

Convers-Hostname ändern:

(Eingabe auf der TNN Konsole, nicht im Conversmodus)

CONV M Glaubitz

Zusätzlichen CTEXT einbinden:

1. Öffnet man die Datei conversd.xhf.
2. Geh zum Dateiende bzw. bis @@-----
3. Nun fügt man den neuen Platzhalter ein:
@@CTEXT
@@-----
4. Zwischen den neuen Platzhalter @@CTEXT und @@----- kann man einen beliebigen Text schreiben.

Beispiel:

```
@@CTEXT
*** Looking up DL-Converters
*
* (Hilfe in Deutsch)
*
* ACHTUNG!! Neuer Convers-Befehl Nickname (/NI Name)
*
@@-----
```

Und dieser wird bei jedem Login in den Conversmodus angezeigt.

Passwortschutz:

Der Passwortschutz basiert auf das THENET verfahren.
Min. 5 bzw. Max. 80 Zeichen darf das Passwort enthalten.
Setzt man ein Passwort, wird im Convers-Netz das Passwort
(natürlich kodiert) bekannt geben für alle TNN Nodes.
Der Hintergrund, egal welche TNN Node man benutzt, ist der
Passwortschutz aktiv.

Syntax: pass <neu | zeige | loeschen > [Neues Passwort]

Passwort setzen:

```
/pass neu meinpasswort
```

Passwort zeigen:

```
/pass zeige
```

Passwort loeschen:

```
/pass loeschen
```

(HT)TPD - Ein simpler Httpd-ServerNeu:

Der HTTPD-Server funktioniert nur unter LINUX und WIN32-Systeme.

Das Grundgerüst des HTTPD-Server besteht aus dem Quellcode von der Baycommmailbox (BCM V1.44).

Die Funktion des HTTPD-Server ist sehr ähnlich mit der BCM zu vergleichen.

Beim ersten Login wird nach Benutzername und Passwort gefragt,

Passwort ist optional, also nicht zwingend!

Kommt eine http Anfrage vom User, wird dieser automatisch ins TNN angemeldet,

Befehl ausgeführt und wieder abgemeldet.

Konfiguration:HTTPD-Port definieren in der TNN.INI:

```
device httpd
kisstype 23
port 0
```

TNN-Console:(Httpd-Port oeffnen)

```
PO 0 ON
```

TCP-Port ändern:

```
HT P 80
```

MY-LOGlevel ändern:

```
HT L 0
```

Damit ist die Konfiguration abgeschlossen.

Port **0** ist nur ein Beispiel! Der Httpd-Server kann auch auf andere Ports definiert werden!

Wenn es Probleme beim Httpd-Server geben sollte, ist es hilfreich LOGlevel einzuschalten.

```
HT L 1
```

Es werden nur Fehlermeldungen in die Datei „httpd.log“ geschrieben.

```
HT L 2
```

Es werden alle Informationen in die Datei „httpd.log“ geschrieben, wie z.B. Rufzeichen, IP-Adresse und Socket des User oder welche Befehle ein User an den HTTPD-Server schickt.

Weitere Infos:

Es wird auch drauf geachtet, ob ein User in der SUSPEND-Liste (Rufzeichen sperren) steht!

Interne und Externe HTML-Ausgabe.

*** Interne HTML-Ausgabe: ist eine Standard HTML-Seite die im Programm festgelegt ist.

*** Externe HTML-Ausgabe: jeder Sysop hat die Möglichkeit seine HTML-Dateien selbst zu erstellen. Weitere Infos siehe - **Einen eignen HTTPD-Server (HTML-Dateien) zusammen basteln.**

Einen eignen HTTPD-Server (HTML-Dateien) zusammen basteln

Wer sich ein wenig mit HTML/CSS/JAVA auskennt, kann seinen HTTPD-Server selber zusammen basteln. Das ist Kinderleicht und jeder Sysop kann seine Kreativität voll einsetzen, um das beste rauszuholen (HI).

Konfiguration:

Im Hauptverzeichnis von TNN muss ein Unterverzeichnis „http“ erstellt werden. Alle HTML-Dateien müssen in das http Verzeichnis abgelegt werden.

Die externe HTML-Ausgabe ist dann aktiv, wenn im Verzeichnis http eine „index.html“ gefunden wurde.

HTML-Dateien/Beispiele findet man unter der URL: <http://dig531.dyndns.org/tnn/beispiel.tar.gz>

Platzhalter in einer HTML-Datei:

Sicherlich fragen einige wie ich nun das Login Rufzeichen bzw. die IP-Adresse vom User in das eigene HTML bekomme! Eigentlich ganz einfach, durch einen **Platzhalter**.

Folgende Platzhalter sind z.z. definiert:

- ~a Autor (in eigener Sache)
- ~b Ausgabe eines Befehl (siehe command.html unter HTML-Dateien/Beispiele)
- ~c Counter Zählerstand
- ~i IP-Adresse vom User
- ~l Login String
- ~m Node Rufzeichen
- ~r Rufzeichen vom User

Diese Tabelle kann man noch erweitern!!!
Also, was fehlt noch?!

Platzhalter in HTML-Dateien einfügen:

Es ist Kinderleicht die Platzhalter in die HTML-Dateien einzubinden. Hier ein Beispiel wie ich mein Node Rufzeichen in das title bekomme:

```
<title>~m</title>
```

Oder Login Rufzeichen und die dazugehörige IP-Adresse ausgeben:

```
<tt><b>Login: ~r IP: ~i</b></tt>
```

Alles klar ?!

TNN Befehl in HTML-DATEI definieren:

Es ist möglich, jeden TNN Befehl (für User) in einer HTML-Datei zu definieren. Was Rot markiert ist, ist der eigentliche Befehl! Der Befehl kann gekürzt oder ausgeschrieben sein.

```
<a href="/cmd?cmd=ax">AX-Routes</a>  
<a href="/cmd?cmd=user">User</a>  
<a href="/cmd?cmd=ro">Routes</a>
```

Natürlich kann man die Verweise auch mit CSS/Java verwenden!

(SPE)ECH - Sprache einstellen

Neu:

(TEL)NET - Ein simpler Telnet-Server

Neu:

Der Telnet-Server funktioniert nur unter LINUX und WIN32-Systeme.
Zu beachten wäre, das der TCP-Port (Standard 23) im jeweiligen System freigeschaltet (Firewall) sein muss!

Konfiguration:

Telnet-Port definieren in der TNN.INI:

```
device telnet  
kisstype 22  
port 0
```

TNN-Console:

(Telnet-Port oeffnen)

```
PO 0 ON
```

TCP-Port ändern:

```
TEL P 23
```

MY-LOGlevel ändern:

```
TEL L 0
```

Damit ist die Konfiguration abgeschlossen.

Port **0** ist nur ein Beispiel! Der Telnet-Server kann auch auf andere Ports definiert werden!
Wenn es Probleme beim Telnet-Server geben sollte, ist es hilfreich LOGlevel einzuschalten.

```
TEL L 1
```

Es werden Informationen in die Datei „telnet.log“ geschrieben (für mich sehr hilfreich).

Um den Telnet-Server zu Testen, einfach im jeweiligen System:

```
telnet localhost 23
```

ausführen. Bekommt man kein Zugriff auf den Telnet-Server, prüfen ob alle Konfigurationen korrekt eingetragen sind, auch an die Firewall denken!

Weitere Infos:

Den Login-Text kann man erweitern, in dem man eine Datei „login.txt“ erstellt, mit dem jeweiligen Inhalt. Dieser Inhalt wird dann bei jeden Connect mit ausgegeben.

Ein Telnet-User kann maximal 9 Connects aufbauen!

Die SSID wird automatisch vergeben.

Es wird auch drauf geachtet, ob ein User in der SUSPEND-Liste (Rufzeichen sperren) steht!

Der Parameter „T3-Timer „ sollte sehr niedrig eingestellt werden. Damit wird geprüft ob der Telnet-User noch online ist. Ein sinnvoller Wert wäre z.B. 150.