

# Allgemeines

## Keybindings:

ESC > Neue Verbindung

ALT + C > Neue Verbindung

ALT + D > Disconnect

F1 - F10 > Kanal 1 - 10

F12 > Monitor Mode

SHIFT+F1 – SHIFT+F12 > F-Texte (Macro Texte)

STRG + plus > Textgröße vergrößern

STRG + minus > Textgröße verkleinern

## Datei Endungen (data/usertxt/<USER CALL>):

- \*.ctx > C-Text

- \*.btx > Bye-Text

- \*.atx > News-Text

- \*.itx > Info-Text

- \*.litx > Long Info-Text

- \*.popt > Programm Data Files (Nicht ändern !)

# Vorwort:

Moin moin liebe PR'ler,

nach ein paar Jahren wieder zurück im Hobby "Packet Radio" konnte ich mir das Elend, was die Situation an Terminal Programmen für PR angeht, nicht weiter mir angucken.

Daher habe ich begonnen ein neues PR(AX.25) Terminal Programm zu entwickeln.

Was heißt Elend genau?

Für mich persönlich das größte Elend ist das es kein ordentliches Terminal Programm für Linux gibt (LinPAC, ax25call). Und schon gar kein GUI basierendes.

Und das 2. Elend für Windows Benutzer ist das man etliche Zusatzprogramme wie flexnet32 und oder VirtualComport2TCP (oder wie das heißt) benötigt um entweder via AXIP oder zu Direwolf was z.B: auf nenn Raspberry PI läuft.

Selbst für ne simple Verbindung zu nenn TNC kommt man ( meines Wissens ) nicht an flexnet32 vorbei.

Auch der betrieb rein über AXIP ist sehr eingeschränkt möglich, da für jede AXIP Verbindung ein eigener Port für flexnet32 angelegt werden muss. Und da die Anzahl der Ports auf flexnet32 begrenzt sind, kann man sich ja ausmalen in welche Limitierungen man läuft.

Was kann PoPT oder wird es zukünftig können ?

PoPT ist Python geschrieben und es läuft somit auf Windows sowie auf Linux Systemen bzw. es wird darauf geachtet das beide Systeme unterstützt werden.

Python hat ausserdem den Vorteil das es für Anfänger eine leicht zu erlernende Script Sprache ist (ich bin auch nur Amateur und kein Profi Programmierer) und somit Teile des Quellcodes nach belieben angepasst werden können.

Der nächste Punkt der mir wichtig war bei der Entwicklung von PoPT war Konnektivität. Das heißt das es ohne die o.g. Zusatzprogramme möglich ist sich via AXIP, Direwolf über KISS via TCP oder über Serielle Schnittstelle an KISS Geräte wie TNC oder Direwolf, anzubinden.

Auch ein paar andere Sachen sind noch geplant, wo ich mich jetzt erst einmal nicht weiter äußern will, solange die Grundfunktionen von PoPT nicht einwandfrei laufen.

PoPT ist derzeit noch in einem sehr frühen Stadium ( Wird seit ca. Mitte Feb. entwickelt ) und hat von daher noch ein paar Bugs oder Fehler die wie Bugs erscheinen aber einfach nur Features sind die noch nicht implementiert wurden, wie z.B: die Stationserkennung beim weiter/reconnect.

Allerdings besitzt PoPT jetzt bereits ein paar Features, die es so, meines Wissens nach, in kein weiteren PR Terminal Programm gibt.

- AXIP Multicast
- Speichern und automatisches aufrufen der AXIP Adresse der jeweiligen Stationen auf dem AXIP Port ( IP u PORT der Station wird in MH gespeichert)
- KISS over TCP
- RX-ECHO ein kleines Tool was es unter Linux ax25tools gibt und manchmal ganz nützlich sein kann zum Testen oder experimentieren

Wie gesagt, ist dieses Programm noch recht "jung" und da ich auch noch nenn Job hab, wird es ne ganze weile dauern bis es auch ein wirklich gute PR Programm wir

Das ganze Projekt ist auf GitHub zu finden und ich werde hier im Forum für die Windows benutzer die neueste Version als EXE in ZIP verpackt hochladen.

Allerdings ist auf GitHub immer die aktuellste Version zu finden.

Hier der Link zu GitHub [github.com/DerHirschi/AX25\\_POPT](https://github.com/DerHirschi/AX25_POPT)

Hier der Link für ZIP Dateien [forum.packetradio-salzwedel.de/PoPT/](https://forum.packetradio-salzwedel.de/PoPT/)

Hier der Link zur Telegram Gruppe bzgl. PoPT [t.me/poptsupport](https://t.me/poptsupport)

Viel Spaß beim Testen und Danke fürs Melden der Bugs in der Rubrik BugReports ..

**!!! Achtung !!!**

**!!! Bitte überprüft selbständig die bereitgestellten EXE bzw. ZIP Dateien auf evtl. Schadsoftware !!!**

**Ich für meinen Teil arbeite nach besten Wissen und Gewissen um ein "einschleichen" von Schadsoftware, während des Konvertierung des Python Scriptes in eine ausführbare EXE Datei, zu verhindern.**

**Aber ich kann auch nicht vollständig ausschließen, dass es passiert!**

**Mir persönlich wäre es am liebsten, wenn ihr PoPT direkt von GitHub bezieht und als Python Script laufen lasst oder die Konvertierung selber vor nehmt.**

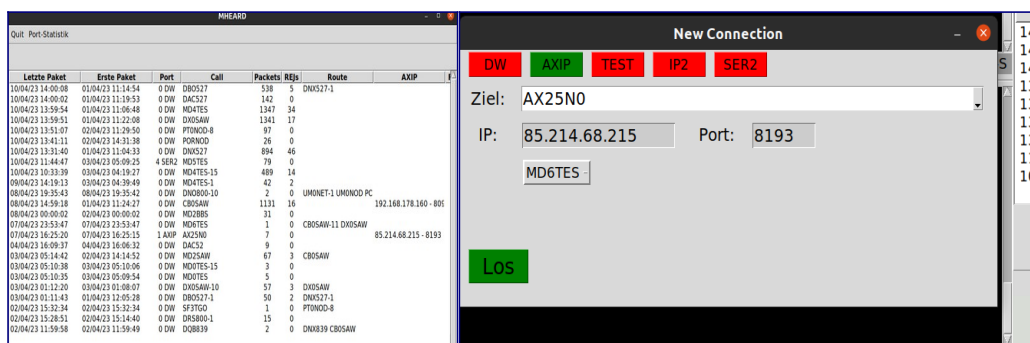
73 ...

## PoPT AXIP

Das AXIP-Verfahren bei PoPT funktioniert etwas anders als bisher von flexnet32 gewohnt.

Die jeweiligen AXIP Adressen sind nicht wie bei flexnet32 fest mit einem Port verknüpft, sondern mit dem dazugehörigen Call.

Dementsprechend werden die AXIP-Adressen ( IP, Port ) zusammen mit Call in der MH Liste gespeichert bzw. müssen, wenn sie dem System noch nicht bekannt sind, beim Aufbau einer Neuen Verbindung mit angegeben werden.



Wenn die AXIP-Adresse dem System bereits bekannt ist, muss sie nicht mehr eingetragen werden. Oder noch einfache, ein einfacher klick auf den MH Listen Eintrag ( egal ob große MH Liste oder die kleine an der Seite ) öffnet das "New Connection" Fenster mit allen notwendigen Daten vorausgefüllt.

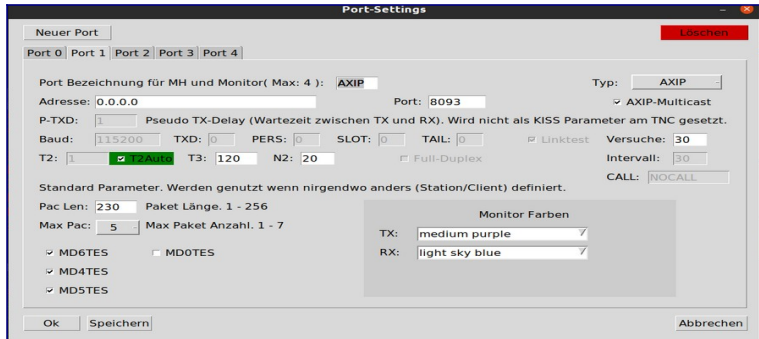
Dementsprechend sind auch die Einstellungen in den "Port Einstellungen" anders zu verstehen. Der AXIP Port entspricht ein geöffneten Port eines Servers.

Die voreingestellte IP Adresse ( 0.0.0.0 ) sagt aus, dass der AXIP Port auf jeder IP Adresse erreichbar ist, die der Rechner hat.

Würde man die IP auf z.B: 127.0.0.1 ändern, wäre der Port von außen nicht erreichbar sondern nur via localhost.

Also intern, von Programmen die auf dem selber Rechner laufen.

Oder ihr habt ein Rechner mit mehreren Netzwerkkarten ( Virtuell oder echt ), also mehreren IP's, so könnt ihr den Port nur für eine IP erreichbar machen.



Aber im allgemeinen kann man die IP bei 0.0.0.0 belassen.

Wichtig ist auch, dass wenn ihr via AXIP vom Internet aus connected werden wollt, müsst ihr den eingestellten Port an euren Router ( Firtzbox oder was auch immer ) öffnen.

## PoPT RX-Echo

RX-Echo ist ein Tool, dass den einen oder anderen Linux User aus den Paket ax25-tools bekannt sein sollte.

Mit RX-Echo ist es Möglich den kompletten oder nach Call gefilterten Verkehr von einem Port zu den anderen zu leiten.

Dies Funktion ersetzt kein Digipeater oder Node, kann aber doch manchmal ganz Sinnvoll sein zu Testzwecken oder um zusammen

mit der AXIP-Multicast Funktion den Verkehr von Direwolf, welches via KISSTCP angebunden ist und zu der flexnet32 Anwendung via AXIP weiterleiten.

So ist es auch möglich sich ein Gerät/Port ( TNC/Direwolf/AXIP ) mit mehreren Anwendungen zu teilen.

Mit dem Tool kann eine "Pipe" zu externen Applikationen/Scripten erstellt werden.

Das Tool überprüft in einstellbaren Abständen eine wählbare Datei nach Inhalten und sendet diese an die angegebene Adresse mittels

UI-Frame ( Im unProto Modus ).

Was von der Adresse (Pipe) empfangen wird, wird in eine andere, wählbare Datei geschrieben.

Auch kann eine Pipe auf eine bestehende Verbindung (Proto Modus) gelegt werden.

So können z.B. von externen Programmen Baken mit z.B. Sensordaten/Wetterdaten/... erzeugt werden, die dann von PoPT gesendet werden.

Auch das "live" Übertragen von Logdateien wäre so möglich.

## unProto Pipe

Unprotokollierten AX.25 Pipe.

Kann im einfachsten Fall dazu genutzt werden um Baken zu senden in den man z.B. via cronjob ein Text das eingestellte Textfile schreibt.

Sobald PoPT die Daten in dieser Datei liest, werden sie an die voreingestellte Adresse mit den voreingestellten Frame Parametern gesendet und aus der Text Datei gelöscht.

Allerdings ist diese "Baken" Funktion einfacher mit der PoPT Baken Funktion umzusetzen, da auch hier die Möglichkeit besteht, die Bake direkt aus einer Textdatei zu lesen.

Als nächstes ist es mit der Pipe Funktion möglich, Daten von einer bestimmten Station mit zu schreiben, in der voreingestellten Datei.

Die oben genannten Anwendungsbeispiele sind nur die einfachsten.

Da die Pipe Tx sowie RX also eingehende Rohdaten ausgeben sowie eingehende Rohdaten senden kann, ist es somit Möglich Applikationen durch das AX25 Protokoll via HF zu "Pipen" oder sogar eigene Applikationen sogar Protokolle zu schreiben.

Um eigene Protokolle implementieren zu können ist von AX.25 Protokoll aus her schon eine extra PID Protokoll ID vorgesehen die man , unter vielen anderen, auswählen kann für die Pipe.

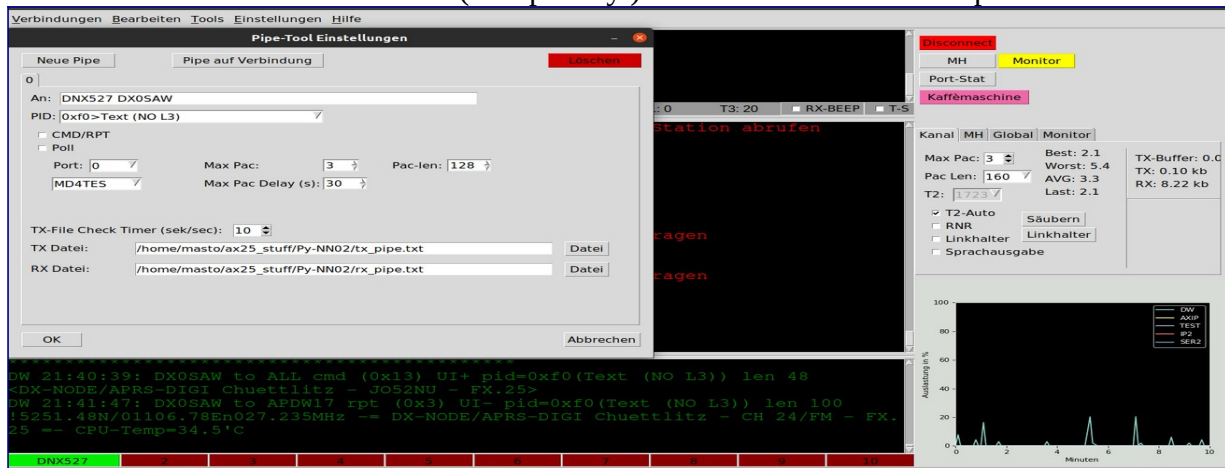
Um die externe Applikation in Zaum halten zu können und aufs AX.25 Protokoll bzw jeweiligen Port Parametern ( Baud usw ) adaptieren zu können, können verschiedene Parameter eingestellt werden.

Max-Pac: Wie viel Pakete auf einmal gesendet werden sollen innerhalb des eingestellten delays.

Max-Pac Delay: Zeitspanne bis die nächsten Pakete gesendet werden.

Pac-Len: Maximale Größe der Pakete

TX-File Check Timer: Zeitabstand ( Loop delay ) in dem das Text File überprüft werden soll.



Dazu ist zu sagen das die eingehenden Daten nacheinander in die jeweiligen Pakete zerlegt und unprotokolliert gesendet werden.

Unprotokolliert heißt, ihr müsst selber dafür sorgen, zu prüfen, ob die Daten auch vollständig am anderen ende ankommen.

Das ganze ist mit einer UDP Verbindung gleich zu setzen, wo verloren gegangene Pakete nicht erneut nachgefragt werden wie bei TCP ( Protokollierte Verbindung / Die das Pipe-Tool ja auch bietet 😊 )

## Proto Pipe

Ist im Endeffekt das selbe wie eine unProt Pipe, nur das ihr euch nicht drum kümmern müsst, ob die Daten beim empfänger ankommen.

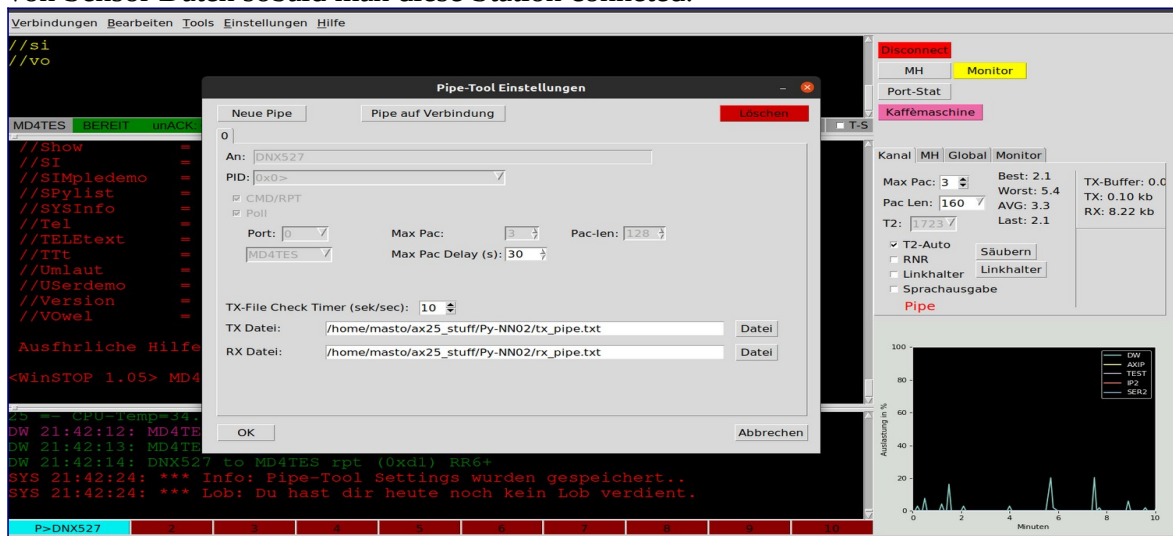
Das wird durch das AX.25 Protokoll sicher gestellt. Also durch die bestehende Verbindung.

Die einfachste denkbare Anwendung hierfür wäre das mitloggen eines QSOs.

Geplant ist noch die Möglichkeit die Pipe direkt auf eine Station(call) legen zu können, was dazu führt das bei connect dieser Station sofort alles durch die Pipe geleitet wird.

Somit kann man quasi eigene Abläufe, Kommandos oder was auch immer hinter ein Call hinterlegen.

Auch die Ausgabe von einer, in Text aufbereiteten, Webseite wären denkbar oder die direkte abfrage von Sensor Daten sobald man diese Station conneted.



Auch ein denkbarer Anwendungsbereich wäre Homeautomation. Im Garten die Pumpe oder Rasensprenger einschalten via PR.

Ich hoffe ich konnte jetzt etwas Licht in die dunkle Pipe bringen..

# Personal Mail System

## - Vorwort

Das PMS dient dazu PR- und Bulletin Mails die von der Heimat-BBS kommen zu Managen (schreiben, lesen, beantworten, speichern ....) und ist kein Ersatz für echte BBS.

Derzeit nutzt das PMS nur das „Forward Protokoll“ im Reverse-Forward Modus, nicht das „TSTHOST“ Protokoll und unterstützt derzeit auch noch nicht das Auslesen von BBS-Baken „unprot Messages“.

Daher kann das PMS vorerst nur nach einstellbaren Zeitplan/Intervallen oder manuell ausgelöst die Heimat-BBS conneten um neuen Mails abzufragen und zu senden. Ähnlich wie das POP3 E-Mail Protokoll früher.

In wie fern weitere Einstellungen von den Heimat-BBS Sysops bzw. an der BBS via Kommandos selber, um auch Bulletin Mails geforwardet zu bekommen, kann ich zunächst nicht weiter sagen, da ich nur ein begrenzte Möglichkeiten wie Zugriff auf andere BBS-Software, nicht das BBS-Netzwerk mit ständig anlegen neuer Testcalls auf anderen BBS oder meine Heimat BBS ändern kann/will.

Das ganze wird aber in kommenden Versionen von PoPT angepasst werden ebenso wird das „TSTHOST“ Protokoll zu Kompatibilität zu anderer BBS Software beitragen

## - Heimat-BBS

Da die Verfahren bei jeder BBS Software anders sind, um den Reverse-Forward auszulösen sind derzeit nur die automatischen Verfahren für FBB und BayCom implementiert.

Allerdings kann der Forward der PMS auch manuell angestoßen werden sobald die Heimat-BBS im entsprechenden Modus ist.

\* automatische verfahren für andere BBS Software wird nach und nach implementiert.

Es ist nicht nötig das Ihr sofort eine eigene BBS aufsetzen müsst allerdings solltet ihr bei der Wahl eurer Heimat-BBS darauf achten das sie in eurer Region(z.B. Bundesland) ist um in der Lage zu sein auch Regionale Bulletin-Mails (z.B. Wetter/Wetterwarnung, Regionale Infos) zu erhalten. Der Call und der Regionalcode der Heimat BBS werden dann zum Teil Eurer PR-Mail Adresse worüber ihr dann „Private Mails“ empfangen/versenden könnt.

Bsp.:

Sysop: MD2SAW

BBS: MD2BBS

Regio der BBS: #SAW.SAA.DEU.EU

Ergibt PR-Mail Adresse: [MD2SAW@MD2BBS.#SAW.SAA.DEU.EU](mailto:MD2SAW@MD2BBS.#SAW.SAA.DEU.EU)

Erklärung Regio:

#SAW(Salzwedel).SAA(Sachsen-Anhalt).DEU(Deutschland).EU(Europa)

Achtung, es gibt kein WW oder WWW in den Regio/Verteiler Adressen um das Verteilen von Interkontinentalen Mails über entsprechende „Gateways“ Routen zu können.

## - Heimat-BBS(FBB)

Um den Reverse Forward bei FBB auslösen zu können muss man als Benutzer den Benutzerstatus „PMS“ haben oder es muss von FBB aus erlaubt sein alle Stationen den Forward zu erlauben.

Hier bitte den Sysop der BBS kontaktieren.

### Verfahren PMS Status setzen:

In der FBB Konsole mit Sysopstatus folgenden Befehl:

EU <BENUTZERCALL>

dann dem Menü folgen.

### Verfahren „Allen Stationen Forward erlauben“:

In der Datei fbb.conf ( ! Kann in anderen FBB Versionen anderen Namen haben )

sicherstellen das **128: Accepts forwarding only from pre-declared BBS**

ausgeschaltet ist.

Das ganze hier zu Erklären würde zu weit gehen.

Übrigens, für CB-Stationen sollte **„4096: Test of callsigns is less strict. Allows all "callsigns" as long as they have one figure (0-9) anywhere in the callsign.“** eingeschaltet werden. Es handelt sich hierbei um ein „AFU-CALL“ Filter.

# New in 5.15c45-51: Parametres:

# 1 : A space is mandatory before the @ in a send message command

# 2 : The length of the fields of a hierarchical address is not  
# tested to be 6 characters

# 4 : The header line of a message is not truncated to the space before  
# the 79th character



```
# 8 : Header MBL/RLI
# 16 : If there is no BBS field, the callsign of the BBS is sent to the PMS
# 32 : Deletes the DATA messages sent to SYSOP
# 64 : Don't use the BID recovered from headers and use a new one
#> 128: Accepts forwarding only from pre-declared BBS
# 256: WP Messages are not held.
# 512: XForwarding protocole has priority on FBB protocole.
# 1024: Generation of an alternate BID like F6FBB-12345 (for dual BBS site)
# 2048: Checksum unvalidated on XFwd.
# 4096: Test of callsigns is less strict. Allows all "callsigns" as long
#       as they have one figure (0-9) anywhere in the callsign.
fbbfwd = OK 5392
```

Nachtrag:

Leider ist mir zu spät aufgefallen das jeder PMS Benutzer, der seine Nachrichten über das „Forwardprotokoll“ senden/empfangen will, auch einen Forwardeintrag benötigt. Das wird später nicht mehr der Fall sein, wenn das „TSTHOST“ Protokoll in PoPT implementiert wurde.

Demnach ist das PoPT PMS zunächst eher für Sysops und/oder feste Benutzer der jeweiligen Heimat BBS geeignet.

Anlegen eines Forwards in LinFBB:

Im fbb Ordner /fwd die Datei <USERCALL>.fwd anlegen die wie folgt aussieht.

```
A <USERCALL>
  F <USERCALL>
  G *
  R
  O 10
#
-----
```

Im fbb Ordner die Datei forward.sys bearbeiten:

```
< fwd/<USERCALL>.fwd
```

einfügen.

Im fbb Ordner die Datei bbs.sys bearbeiten und den USERCALL einfügen hinter einer der laufenden Nummern.

! Die „leeren“ Zeilen wo nur Nummern stehen dürfen nicht gelöscht werden. !

Also:

```
02 MD2BBS
03 <USERCALL>
04
..
```

Bitte <USERCALL> mit den Call des Users ersetzen ohne die <>.

LinFBB wird standardmäßig in den Ordner /usr/local/etc/ax25/fbb „installiert“.

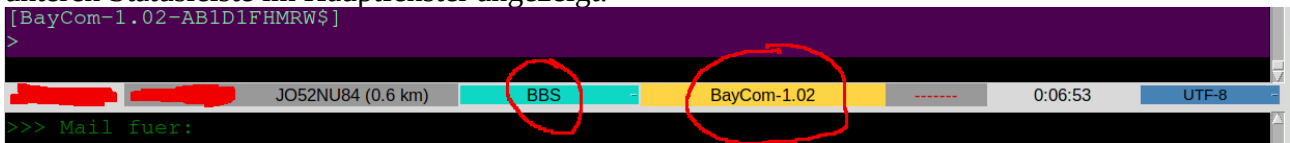
## - Heimat-BBS(BayCom)

Es ist wichtig das PoPT die MailBox/BBS automatisch anhand des „Identifikators/Headers“  
Bsp.: [BayCom-1.02-AB1D1FHMRW\$]  
erkannt hat.

Hierzu muss man einmal „von Hand“ connecten und das Kommando „F>“ an BayCom senden.  
Jetzt müsste folgendes zurück gesendet werden:

```
[BayCom-1.02-AB1D1FHMRW$]  
>
```

Wenn PoPT die Gegenstation als BBS und die Software als BayCom erkannt hat, wird das in der unteren Statusleiste im Hauptfenster angezeigt.



Nun kann testweise ein manueller Forward angestoßen werden im oberen Hauptfenster Menü unter:  
PMS > FWD Starten

## - Heimat-BBS(BayCom Login-Verfahren)

Login auf die Heimat-BBS via „BayCom Login“\* wird derzeit vom „AutoFWD“ Modus nicht unterstützt.

Allerdings unterstützt PoPT das „BayCom Login-Verfahren“ an sich.

Somit muss sich zunächst an die Heimat-BBS angemeldet werden und dann der Forward mit „FWD Starten“ angestoßen werden.

\*Auch FBB, TNN und diverse andere PR Anwendungen nutzen dieses Verfahren.

Bsp.:

```
MD2BBS-0> 16 14 66 34 28 [1701195616]  
Ok
```

```
(1) MD2BBS BBS>
```

## - Manueller Forward (Menüleiste: PMS>FWD Starten)

Die Heimat BBS muss „von Hand“ connected werden.

Ist ein automatischen verfahren für die Heimat-BBS Software verfügbar und bei Baycom die BBS Software von PoPT erkannt worden kann der manuelle Forward gestartet werden.

Ansonsten muss die Box zuvor in den reverse Forward Modus versetzt werden.

## - PMS Einstellungen

Das Forwarding kann erst ausgelöst wenn zuvor die Heimat-BBS und die Benutzerdaten in den PMS Einstellungen hinterlegt wurden.

Menüleiste > PMS > Einstellungen

Python o.ther Packet T.terminal 2.100.22dev

**Eigene Station**

CALL: MD2SAW

Region: #SAW.SAA.DEU.EU

MID: 463

Set MID 463

Ok Speichern Abbrechen

**HomeBBS**

☒ AutoFWD ☒ Single Conn

Neu Löschen

**MD2BBS**

BBS Call: MD2BBS

Regio: #SAW.SAA.DEU.EU

Port ID: 1

VIA: CB0SAW

AXIP:

AXIP-Port: 0 Schedule

Speichern

## - PMS Einstellungen (Region)

RegionsCode/Verteiler der Heimat-BBS. Die in der PR-Mail Adresse.

## - PMS Einstellungen (Abbrechen)

Keine Angst, hier bricht nicht wirklich was aber. Die zuvor gemachten Einstellungen hören nur auf zu existieren. ;-)

## - PMS Einstellungen (MID)

Die MID (Message ID) ist eine laufende Nummer die vom PMS selber erzeugt wird.

**Das setzen der MID wird nur dann notwendig wenn die PMS Datenebank Daten abhanden/gelöscht wurden.**

Sollte das der Fall sein muss darauf geachtet werden das die MID größer ist als die MID/BID der letzten Nachricht die an die Heimat-BBS gesendet wurde.

Aus der MID und dem Call resultiert eine „unique Message/Bulletin ID“ einzigartige Nachrichten ID die im ganzen BBS Netzwerk verwendet wird und wie der Name schon sagt, einzigartig sein muss.

Sollte die beim Forwarden die MID/BID schon bei der Heimat-BBS bzw. im BBS-Netzwerk schon vorhanden sein, wird die Nachricht von der Heimat-BBS beim senden abgewiesen ( Flag: „S-“)

## - PMS Einstellungen (Single Conn)

Sollten mehrere Heimat BSS angelegt sein \* wird sichergestellt das immer nur mit eine BSS gleichzeitig geforwardet wird.

\* Ja das geht auch, man kann z.B. „System-BBS“ oder „Privates BBS“ System betreiben über das z.B. Linux System Meldungen oder der Gleichen geteilt werden. Quasi als „Privater PR-Mail Server“.

## - PMS Einstellungen (AutoFWD)

Die Heimat-BBS wird nach eingestellten Zeitplan/Intervallen connected um evtl. neue Mails zu senden/empfangen

## - PMS Einstellungen (AXIP)

Wenn die AXIP Adresse der Heimat-BBS bzw. des VIAS (NODE) PoPT schon bekannt ist (siehe MH-Liste) dann ist der Eintag hier nicht mehr notwendig.

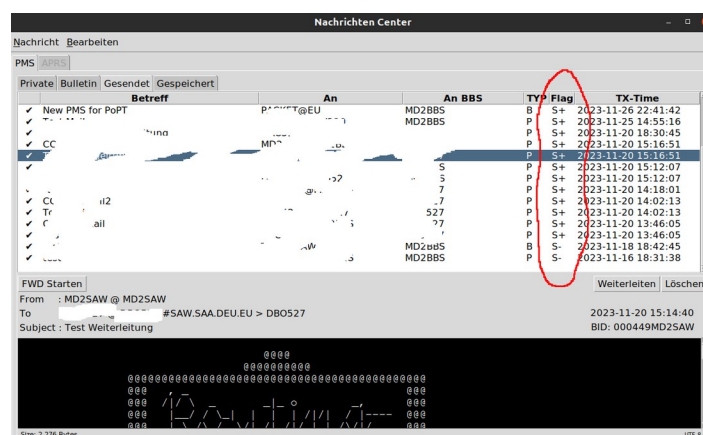
Bitte darauf achten und vorher testen ob die BBS auch über die Route erreichbar ist. Also einmal von Hand connecten über diese Route und AXIP Adresse.

## - PMS Einstellungen (Schedule)

In der derzeitigen Version von PoPT kann es noch vorkommen das die Scheduler Daten beim 1. mal anlegen einer Heimat-BBS nicht übernommen werden.

Hier bitte nach dem Speichern und Schließen des Einstellungs-Fensters bitte das Fenster noch einmal öffnen und die Schedule Einstellungen prüfen.

## - PMS(FLAGS)



*F = Forward (Noch nicht geforwardet)*

*E = Entwurf/Draft (Default)*

*S= = Gesend (Heimat-BBS empfängt diese Nachricht grade von einer anderen Quelle)  
Nachricht wird beim nächsten connect noch einmal versucht zu übertragen.*

*S+ = Gesend (Erfolgreich gesendet)*

*S- = Gesend (Heimat-BBS hat bereit eine Nachricht mit dieser MID/BID)*

*H = Gesend (Nachricht wurde von Heimat-BBS angenommen aber auf „Held“ gesetzt)*

*EO = OFFSET Error not implemented yet TODO*

Das Haupt User-DB Fenster kann auf verschiedenen Wegen geöffnet werden.

## - BayCom Login(Sys-Passwort)

Das BayCom Login Verfahren ist nicht das sicherste, von daher sollten hier einige Dinge beachtet werden.

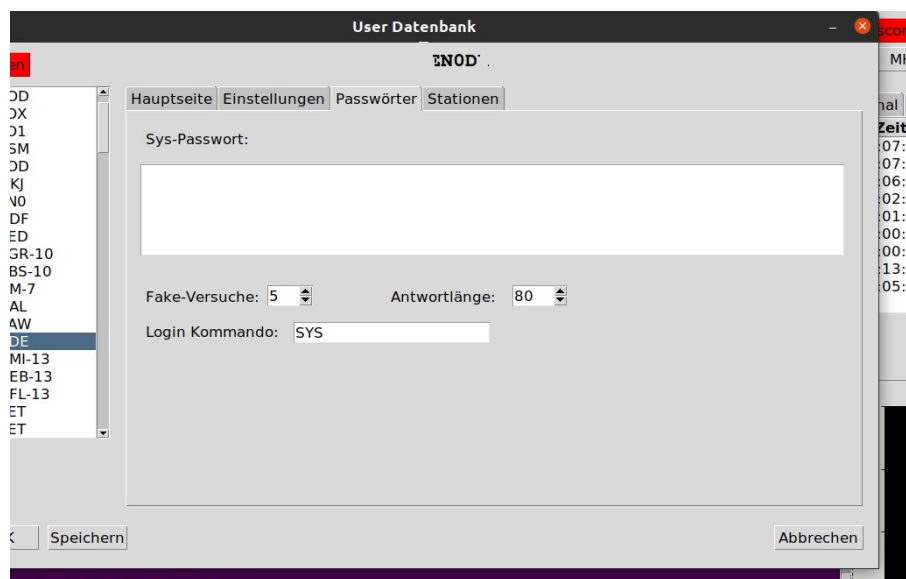
Der einzige Schutz vor eine „Bruteforce“ Attacke besteht darin das die „Leitung“ sehr langsam ist. Spezielle über HF.

Es sollten die Max mögliche Länge des Passworts benutzt werden.

Auch sollte ein Passwort Generator verwendet werden.

Sonderzeichen würde ich aufgrund der Probleme mit der De/Enkodierung vermieden werden und ist für die Sicherheit des Passworts nicht so ausschlaggebend wie die Länge und vor allem zufällige Zeichen die in keinem Wörterbuch vor kommen.

Das Passwort kann in der User-DB in dem jeweiligen Eintrag der Station in dem Reiter „Passwörter“ gesetzt werden.



## - BayCom Login(Fake-Versuche)

Um die Sicherheit zu erhöhen gibt es bei einigen Systemen wie bei TNN die Möglichkeit sich einzuloggen ohne das man eine Bestätigung erhält ob der Login erfolgreich war.

Das kann dazu genutzt werden so genannte „Fake-Logins“ zu senden in dem nicht das Korrekte Passwort übermittelt wird sondern nur eine zufällig generierte Zahl.

PoPT sendet dann beim Login unter einer Serie gesendeter Fake-Logins das echte Passwort (den echten Login)

Das macht es den „Angreifer“ schwerer das echte Passwort zu ermitteln.

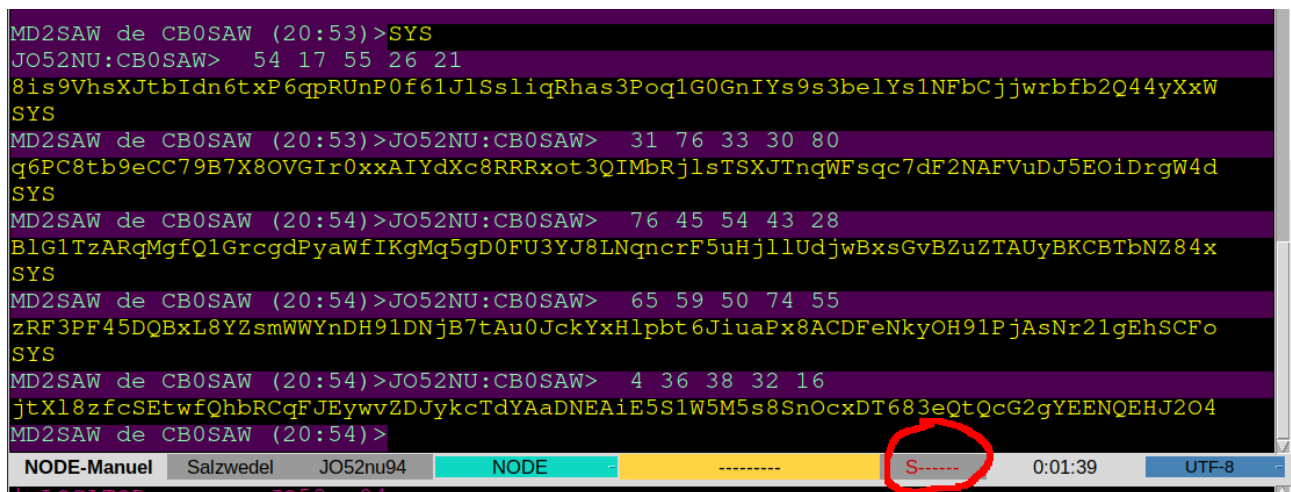
## - BayCom Login(Antwortlänge)

Da das BayCom Login Verfahren immer 5 Stellen aus dem Passwort abfragt ist es trotz Fake-Versuche (was von z.B.: FBB nicht unterstützt wird) relativ einfach das Passwort mit der Zeit „Mitzuschreiben“

Um das zu erschweren kann das eigentliche 5 stellige Passwort in einer zufällig erzeugten Buchstaben/Zahlen Reihe „versteckt“ werden.

Diese verfahren wird von FBB sowie TNN unterstützt ( andere Software unterstützt das wahrscheinlich auch ).

Das ganze sieht dann so aus.



```
MD2SAW de CB0SAW (20:53)>SYS
JO52NU:CB0SAW> 54 17 55 26 21
8is9VhsXJtbIdn6txP6qpRUnP0f61JlSsliqRhas3Poq1G0GnIYs9s3belYs1NFbCjjwrbfb2Q44yXxW
SYS
MD2SAW de CB0SAW (20:53)>JO52NU:CB0SAW> 31 76 33 30 80
q6PC8tb9eCC79B7X8OVGIr0xxAIYdXc8RRRxot3QIMbRjlsTSXJTnqWfsqc7dF2NAFVuDJ5EOiDrgW4d
SYS
MD2SAW de CB0SAW (20:54)>JO52NU:CB0SAW> 76 45 54 43 28
B1G1TzARqMgfQ1GrcgdPyawfIKgMq5gD0FU3YJ8LNqncrF5uHj1lUdjwBxsGvBZuZTAUyBKCBTbNZ84x
SYS
MD2SAW de CB0SAW (20:54)>JO52NU:CB0SAW> 65 59 50 74 55
zRF3PF45DQBxL8YZsmWWYnDH91DNjB7tAu0JckYxHlpbt6JiuaPx8ACDFeNkyOH91PjAsNr21gEhSCFo
SYS
MD2SAW de CB0SAW (20:54)>JO52NU:CB0SAW> 4 36 38 32 16
jtXl8zfcSEtwfQhbRCqFJEyvwZDJYkcTdYAAaDNEAiE5S1W5M5s8SnOcxDT683eQtQcG2gYEENQEHJ2O4
MD2SAW de CB0SAW (20:54)>
NODE-Manuel Salzwedel JO52nu94 NODE ----- S----- 0:01:39 UTF-8
```

Das „S“ in der unteren Statusleiste zeigt euch an ob ihr eingeloggt seid oder nicht.

PoPT kann nicht prüfen ob der Login erfolgreich war, das „S“ ist viel mehr ein Indikator dafür, dass man bereits ein Login durchgeführt hat.

## - BayCom Login(Login-Kommando)

Hier kann das Kommando eingegeben werden, welches an die Gegenstation gesendet werden muss um ein Login auszulösen.

Für Sysop Login ist das in der Regel das Kommando „SYS“.

# Dual Port

Mit dem „Dual Port“ Tool ist es möglich 2 Ports zu einem Port zusammenzufassen.

Im Dual Port Modus werden die beiden zusammengefassten Ports als ein Port ausgewertet, RX-Echos und von beiden Stationen empfangene (doppelt empfangene) Frames ausgefiltert.

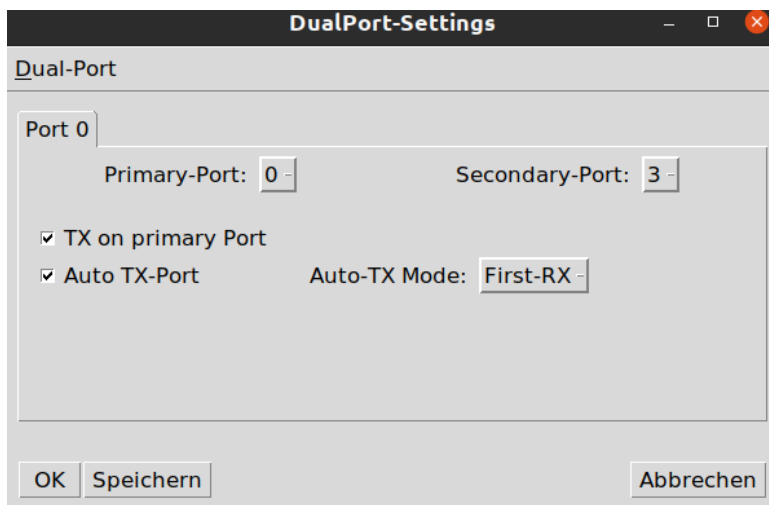
Es gelten die Station Einstellungen des primären Ports.

Mit diesem Tool ist es möglich z.B. 2 RTX an verschiedenen Antennen auf dem selben Kanal zu betreiben ( z.B. Antenne Nord vertikal , Antenne Süd horizontal ) oder ein SDR als zusätzlichen Empfänger zu nutzen.

Grundsätzlich ist das Tool nicht dazu gedacht Gateways zwischen zwei verschiedenen Kanälen/Frequenzen zu realisieren, da PoPT den Dual-Port immer als einen Port handelt (z.B. MH-Liste).

## - Einstellungen

Menüleiste > Einstellungen > Dual-Port



Primär und sekundär Port auswählen.

- TX on primary Port

Auf primären oder sekundären Port senden.

- Auto TX-Port

Der sende Port wird automatisch, je nach eingestellten Modus, ausgewählt.

- Auto-TX Mode First-RX

Es wird auf dem Port gesendet auf dem die Station zuerst empfangen wurde. Wenn die Station noch nicht in der MH-Liste erfasst wurde wird auf dem primären Port gesendet.

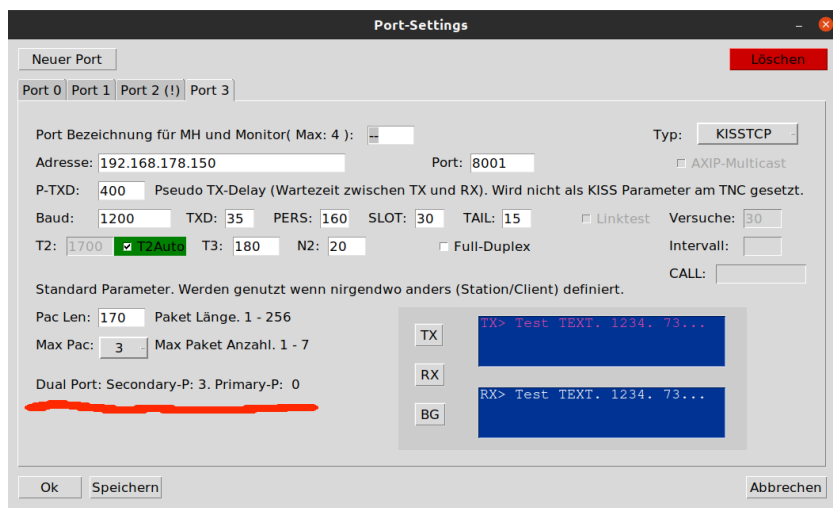
- Auto-TX Mode Last-RX



Es wird auf dem Port gesendet auf dem die Station zuletzt empfangen wurde. Wenn die Station noch nicht in der MH-Liste erfasst wurde wird auf dem primären Port gesendet.

Achtung ! Dieser Modus kann bei hohen Sendeaufkommen zu FRMR führen, da PoPT nicht den Puffer der einzelnen TNC's überwachen kann.

So kann es bei Stationen die über beide Antennen gehört werden, dazu kommen das Pakete plötzlich über den 2. Port gesendet werden auf den der TNC Puffer noch leer ist und er somit Pakete außerhalb der Reihenfolge sendet, da TNC1 die vorherigen Pakete noch im Puffer hat weil es noch nicht zum senden gekommen ist.



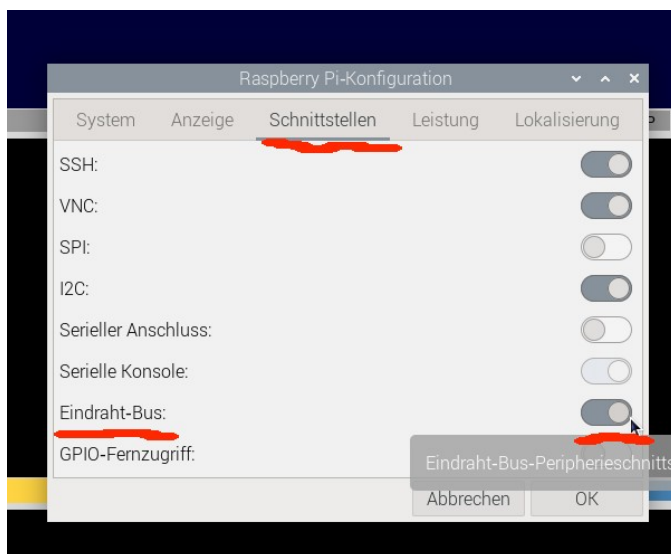
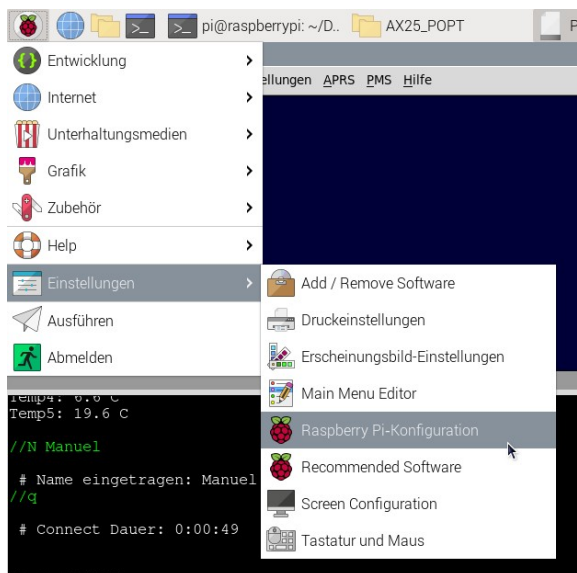
# 1Wire-Sensoren

PoPT bietet die Möglichkeit Sensordaten von Geräten die den 1Wire-Bus verwenden, auszulesen und als Textvariable in C-Text/Bake/Info-Text/usw. zu integrieren.

In den Folgenden Beispielen/Screenshots beziehen sich auf ein Raspberry PI 3 / 4, die 1Wire Funktion kann aber bei jeden anderen Gerät verwendet werde, wo der 1Wire-Bus vorhanden ist.

PoPT bezieht die Daten aus der Ordnerstruktur `./sys/devices/w1_bus_master1`.

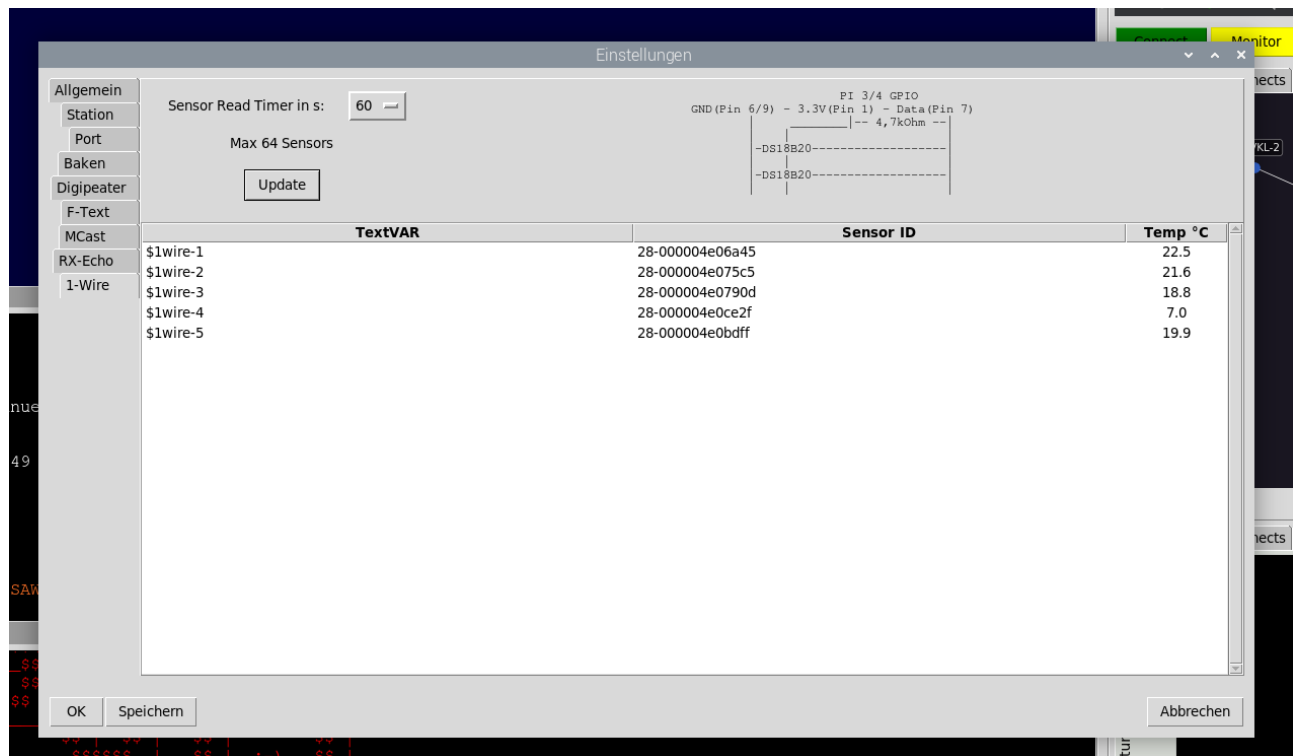
Bei Raspberry-OS muss der 1Wire-Bus erst wie folgt aktiviert werden.



Und anschließend den PI neu starten.

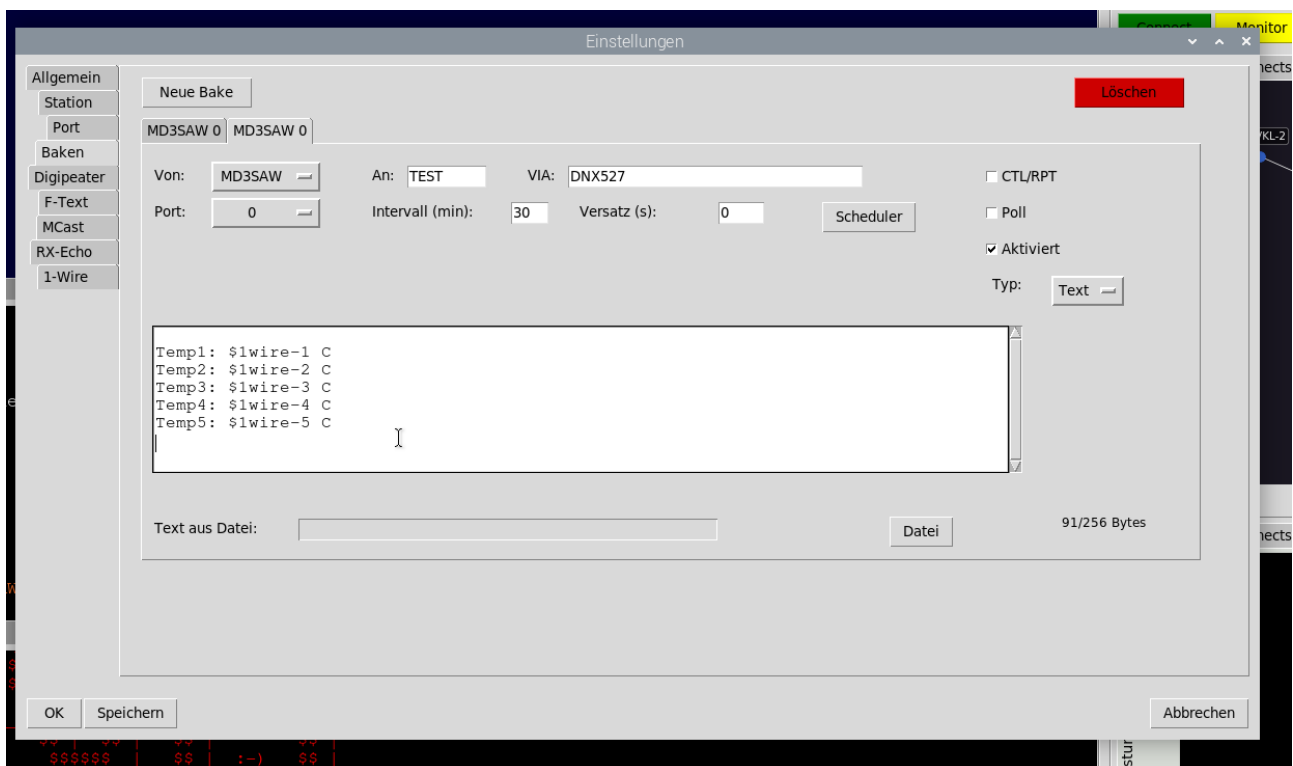
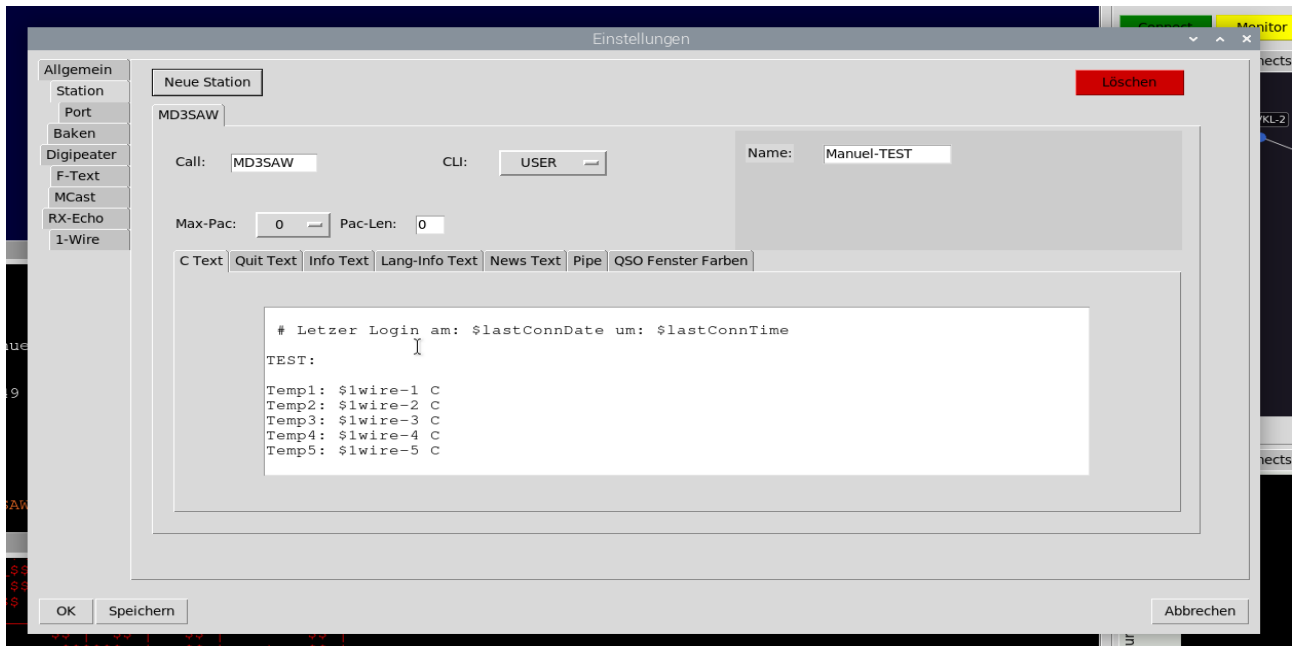
Ob die Aktivierung erfolgreich wart könnt ihr überprüfen ob der Pfad  
/sys/devices/w1\_bus\_master1 vorhanden ist mit,  
ls -la /sys/devices/w1\_bus\_master1  
oder einfach in dem Ihr die Einstellungen in PoPT öffnet.

Dort sollte nun ein neuer Reiter „1-Wire“ erscheinen.



In der 1. Spalte der Tabelle findet ihr die jeweiligen Textvariablen die den Sensoren zugewiesen wurden.

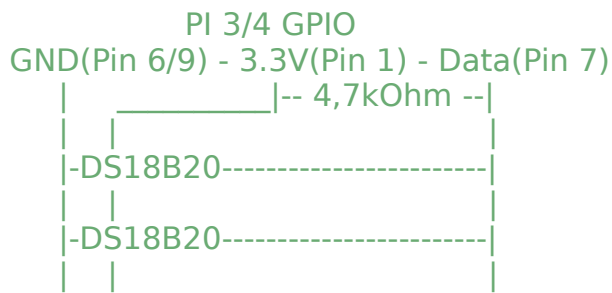
Diese könnt ihr nun, wie Ihr in den folgenden Beispielen sehen könnt, in Euren Texten einfügen.



Da die Abfrage der Sensoren über den 1Wire-Bus sehr einige Sekunden dauern kann, geschieht das Auslesen der Sensoren nicht direkt beim Aufrufen der jeweiligen Texte, sondern in einer ständigen Schleife.

Die Aktualisierungsrate/Abfragerate kann unter ‚Sensor Read Timer in s:‘ eingestellt werden.

Die Sensoren werden wie folgt angeschlossen.



Weiter Informationen hierzu findet Ihr auf :

<https://st-page.de/2018/01/20/tutorial-raspberry-pi-temperaturmessung-mit-ds18b20/>

#### **Stand Version 2.114.x:**

Es werden derzeit nur 1Wire-Temperatur Sensoren unterstützt.

Sollte es noch andere Sensoren geben, wo es Sinnvoll wäre diese zu implementieren, so müsst Ihr Bescheid geben.

## **AXIP MCast-Server**

Die Idee hinter dem MCast ist, daß sich AXIP mehr verhält, wie man es von Funk gewohnt ist. Sprich, jeder kann jeden lesen.

Alle Pakete die an den MCast Server gesendet werden, werden an alle anderen registrierten Teilnehmer die sich im selben virtuellen Kanal weitergesendet.

Somit ist es z.B. möglich, mittels flexnet32 nur einen Port und AXIP Link zum MCast Server anlegen zu müssen und alle Stationen die sich im selben Virtuellen Kanal befinden erreichen zu könne.

Auch UI-Frames, also Baken usw. werden an die anderen Teilnehmer weitergeleitet.

Theoretisch sollte damit auch NetRom/Flexnet/usw-Routing, BBS-FWD und APRS möglich sein.

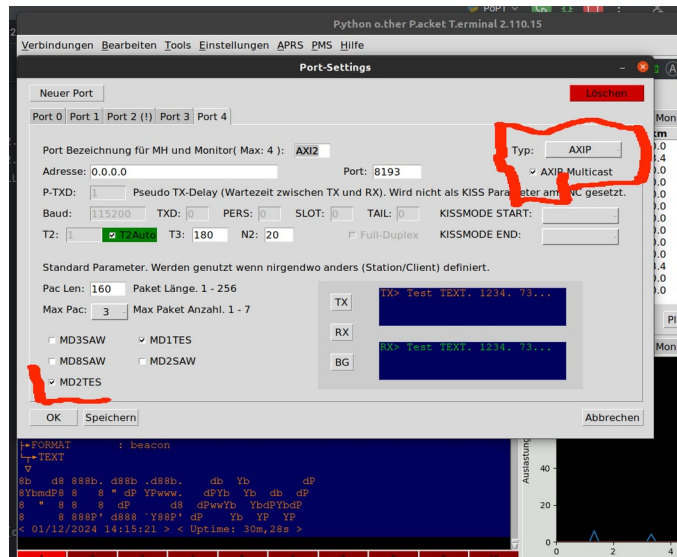
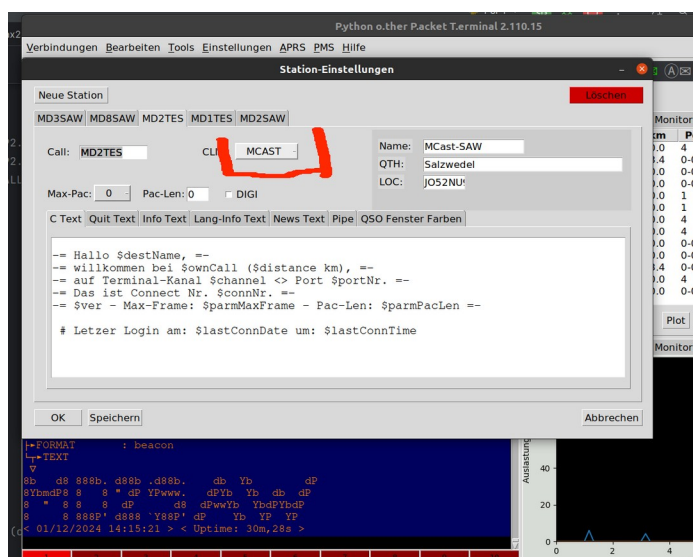
Aber Achtung !! Manche Programme könnten damit nicht klar kommen, wenn eine Station bereits über die eigene IP erreichbar ist und plötzlich die selbe Station über die IP des MCast-Servers liest.

Man kann sich das ganze auch etwas wie ein Convers vorstellen ohne das Ping-Pong Convers Protokoll zu nutzen sondern rein auf AX25 Basis.

Das Registrieren der MCast-/Kanal-Teilnehmer einfach.

Man connected einfach auf den Call des MCast Servers.

Hier kann man auch via Remote Kommandos die Kanäle wechseln, seine AXIP-Adresse (Domain Namen oder IP) ändern und Infos beziehen.



Stand Ver. 2.110.x

Da PoPT noch nicht Performance optimiert ist, ist auch nicht bekannt wie viel Datenverkehr der MCast Server handeln kann.

Auch ist PoPT noch nicht als Serveranwendung ( Dauerläufer ) ausgelegt und somit auch nicht garantiert, dass das Programm ein 24/7/365 Betrieb mit macht ohne Ressourcen wie Ram zu verschlingen.

Gerne könnt ihr mir ein Feedback geben, was das Verhalten und Ressourcen Verbrauch bei einem 24/7 Betrieb angeht.

## Textvariablen

\$ver = PoPT 2.xxx.x

\$time = 20:39:00

\$date = 03/03/2024

\$uptime = Zeit seit Programmstart

\$channel = Kanal NR

\$portNr = Port NR

\$destName = Name der Gegenstation wenn bekannte, ansonsten Call der Gegenstation

\$destCall = Call der Gegenstation

\$ownCall = Eigener Call

\$lastConnDate = Letzter Connect Datum

\$lastConnTime = Letzter Connect Zeit

\$distance = Distanz zur Gegenstation

\$connNr = Connect Nr

\$parmMaxFrame = Max Frame Einstellungen

\$parmPacLen = Pakete Länge Einstellungen

- Bake

- Bake

- Bake

- Bake

- Bake

- Bake

- Bake

- Bake

- Bake

- Bake

- Bake

- Bake

- Bake

- Bake

- Bake