

```

monrc.man(n)
monrc.man(n)
NAME
    .monrc - Konfigurationsdatei für monitor

HINWEIS
    Die Benutzung des Programms monitor erfolgt ohne Gewähr auf Funktion-
    alität, eventuell auftretende Schäden und mit dem Hinweis auf die
    entsprechende Gesetzeslage hinsichtlich des Abhörens nichtöffentlicher
    Funks.

INHALT
    1. Systemvoraussetzungen und Kompilieren
    2. Starten des Programms
    3. Tasten und ihre Funktion
    4. .monrc
    5. Formatierung von Zeilen
    6. Alias-Angaben, Optionen
    6.1. ZVEI
    6.2. POCSAG
    6.3. FMS
    7. Kommentare
    8. Layout
    8.1. Gestaltung
    8.2. Farben
    9. Aktionen
    10. Logfiles
    11. Ressourcenbelastung
    12. Signalerzeugung

1. Systemvoraussetzungen und Kompilieren
    PC oder etwas anderes

    Soundkarte (16 bit), On-board-Soundsysteme sind im Allgemeinen
    ungeeignet.

    Unix-System (auf SUSE-Linux, RedHat und Sun getestet) /dev/dsp bzw.
    /dev/audio (Sounddevice) muss aktiviert sein

    irgendein geeigneter Funkempfänger (ohne Diskriminatorausgang wird es
    schwierig)

    Kabel

    oder ein paar Audiofiles zum Testen

    Entpacken mit:
    tar xvzf monitor-{version}.tgz
    oder
    gunzip -c monitor-{version}.tgz | tar -xvf -
    dasselbe wie
    gunzip monitor-{version}.tgz; tar xvzf monitor-{version}.tar

    cd monitor-{version}

    make          Compiliert monitor und gen, benötigt werden unter
                  anderem die Pakete make und ncurses
    make scope   Compiliert monitor (mit X-Window-Anbindung für Scope-
                  Modul) und gen, benötigt werden unter anderem die Pakete
                  make, ncurses und XFree86-devel (ehemals xdevel)
    make install (als Super User, root)
                  Installiert monitor und gen nach /usr/local/bin,
                  dieses Manual nach /usr/local/mon/monn (Aufruf mit man
                  monrc)
    make clean   Beseitigt überflüssige Dateien
    make uninstall (als Super User)
                  Deinstalliert monitor, gen und dieses Manual)

2. Starten des Programms
monitor [-s -[L|R] modul][[-a -[L|R] modul] [-f .rc-file] [-t typ test-
file] [-h?]]

-s (subtract)
    Ausschließen von Modulen (nur die anderen werden gestartet)

-a (add)
    Aktivieren von Modulen (die anderen werden nicht gestartet)

    Zusätzlich muss der Kanal angegeben werden, für den das Modul
    gesperrt bzw. aktiviert werden soll. Ohne Parameter werden alle
    Module für alle Kanäle gestartet. Werden keine Parameter
    angegeben, sind alle Module aktiv. Wird auf dem einen Kanal ein
    Modul aktiviert (-a -L fms), wird der andere Kanal gesperrt,
    solange nicht auch dort Module aktiviert werden (-a -R poc-
    sag512). Also soll ein Modul auf allen Kanälen dekodieren, muss
    es für jeden Kanal separat gestartet werden.

-L linker Kanal
-R rechter Kanal

Module:
    (was so mehr oder weniger gut decodiert werden kann):
    pocsag512
    pocsag1200
    pocsag2400
    afsk1200
    afsk2400
    afsk2400_2
    fms
    hapn4800
    fsk9600
    dtmf (Telefontöne, Sirenenauflösungsdoppeltöne)
    zvei (Fünfontfolgen, Melderweckton)
    scope (Signalanzeige, benötigt XWindow)

    läuft also auf der Console mit
    monitor -s -L scope -a -R scope
    am besten
    monitor -a -L fms -a -L zvei -a -L dtmf -a -L pocsag512 -a -L pocsag1200
-s -R all

    oder z.B.:
    links: FMS, ZVEI und DTMF
    rechts: Pocsag1200 und FMS
    monitor -a -L fms -a -L zvei -a -L dtmf -a -R pocsag1200 -a -R fms

    Alternativ dazu können die benötigten Module auch in der .monrc einge-
    tragen werden:
    MOD L FMS
    MOD L ZVEI
    MOD L DTMF
    MOD R Pocsag1200 usw.

    Beim Starten von monitor so entweder alle Module aktiv oder nur die,
    die in der .monrc stehen.

-t format test.format
    decodiert RAW-Files, andere Sound-Formate benötigen sox, (nur

```

```

für Mono-Dateien, werden über die Module des linken Kanals
dekodiert)
z.B. -t raw test.raw -t wav test.wav

-f .my_monrc
    Angabe eines anderen Ressourcenfiles relativ zum $HOME, wird bei
    Nichtvorhandensein mit diesem Namen erzeugt; normalerweise wird
    die .monrc benutzt

Das Allerwichtigste
    Auch wenn in den Lautsprechern das Signal zu hören ist, wird noch
    nichts dekodiert! Zum Überprüfen bitte das Scope-Modul mitstarten; dort
    muss das Signal zu sehen sein. Ist das nicht der Fall, müssen die Mix-
    ereinstellungen angepasst werden. Der Line-in muss als Quelle aktiviert
    sein! Bei einigen Soundkarten ist auch das Anpassen von I-Gain
    notwendig! Eine Eingangslautstärke ab etwa 10% genügt zum Dekodieren.

3. Tasten und ihre Funktion
    Cursor up/down, Page up/down, Pos1, End:
        Rollen des Verlaufs Fensters

    L: Löscht das Layout
    Q: beendet das Programm

4. .monrc
    Beim Starten wird im $HOME (falls nicht durch -f anders angegeben) eine
    editierbare .monrc erzeugt, welche drei Formatierungszeilen enthält,
    eine für FMS, eine für ZVEI und eine für POCSAG. Diese sollten nicht
    fehlen!
    Eine .monrc-beispiel ist enthalten, sie kann gelöscht werden.

5. Formatierung von Zeilen
    Die FMS-Formatierung beginnt mit FMS:
    Danach kann man sich etwas zusammenbasteln, Wörter wie ZEIT, BOS, LAND,
ORT, KFZ, STATUS, DIR, BST (Baustufe), TKI (Taktische Infos) und KANAL
(Soundkarteneingang) in Großbuchstaben, jeweils gefolgt von einer ein-
oder zweistelligen Zahl für die Länge. Diese werden dann bei der Aus-
gabe entsprechend ersetzt. Die Länge der ZEIT ist konstant, eine Zahl
muss trotzdem angegeben werden. z.B.
FMS:ZEIT8 BOS4 ORT5 KFZ25 STATUS28TKI3 KANAL1

    Die ZVEI-Formatierung beginnt mit ZVEI:
    Die Wörter ZEIT, NAME, NUMBER, TYPE und KANAL (Soundkarteneingang) -
    plus Längenangabe werden ersetzt. TYPE ist hier die Alarmierungsart
    (Melder, Sirene). z.B.
ZVEI:ZEIT8 NAME37TYPE15 NUMBER5 KANAL1

    Die POCSAG-Formatierung beginnt mit POCSAG:
    Die Wörter ZEIT, NAME, NUMBER, FUNC und KANAL (Soundkarteneingang) -
    plus Längenangabe - werden ersetzt. FUNC ist hier der Funktionstyp.
    z.B.
POCSAG:ZEIT8 KANAL1 NAME38 NUMBER7 FUNC7 Pager

6. Alias-Angaben, Optionen
    Weil man sich nicht alle Codes merken kann, kann man dafür Aliase fest-
    legen, die dann stattdere ausgegeben werden. Bitte nur TABS und keine
Leerzeichen innerhalb einer Zeile verwenden, es sei denn, sie gehören
    zum Text, der ausgegeben werden soll. Falls der in der Alias-Angabe
    stehende Text die angegebene Länge in der Formatierungszeile überschre-
    it, wird er abgeschnitten.

ZVEI:      ZVNAME ZVTYPE COLZVEI IGNZVEI ZVPAUSE DUMP
POCSAG:   PAGER FUNC COLPOC REPPOC TXTPOC IGNPOC ACCPOC ACCTXTPOC
             TXTSUBST TXTPOCSUBST STEUERZEICHEN CORRPOC

FMS:      BOS LAND ORT KFZ DIR TKI STATUS_F_KFZ STATUS_F_LST STA-
             TUS_P_KFZ STATUS_P_LST FORM COLFMS IGNFMS ACCFMS ACTFMS
             TXTSUBST TXTFMSSUBST QUITTING SPRWLINES PTT DUMP SYNCBITS
             DELAYPT DELAYSPR DELAYALARM CRC STEUERZEICHEN SQUELCHFMS
             EXTORT

    Beispiele zu den Schlüsselwörtern:

6.1. ZVEI
    Fünfontfolgen werden fünfstellig angeben:
    ZVNAME<tab>Nummer<tab>Text

    Joker * sind erlaubt:

    ZVNAME 1910* alle Feuerwehren, die mit 1910 beginnen
    ZVNAME 60*** NS Landkreis Peine

Alarmierungsarten bei ZVEI
    ZVTYPE Typ Text
    ZVTYPE 1 Melderlösung
    ZVTYPE 2 Sirenenalarm (nur mit DTMF-Modul)
    ZVTYPE 3 Sirenenprobe (nur mit DTMF-Modul)

    Typ kann Werte von 1 bis 3 enthalten.

Globale Farben für ZVEI-Folgen
    COLZVEI Nummer Farbe
    COLZVEI 321** 3
    COLZVEI 32200 20

    Zu Farben siehe 8.2.

Ignorieren von ZVEI-Folgen
    IGNZVEI Nummer
    IGNZVEI 546** Unterdrücken aller mit 546 beginnenden Folgen

Wartezeit zwischen ZVEI-Folgen
    Damit ZVEI-Folgen nicht doppelt angezeigt werden und abgewartet
    werden kann, ob ein Melderton oder eine Sirenenauflösung folgt,
    gibt es eine Time-out Zeit. Folgt nichts sinnvolles, wird automa-
    tisch "Melderlösung" ausgegeben und die Auswertung ist beendet.
    Standardmäßig dauern die Pausen 600ms +/-60 ms, sodass mit einer
    Wartezeit von 70ms im Normalfall sicher dekodiert werden kann.
    Sollten die Wartezeiten abweichen, kann die ZVPAUSE-Anweisung
    benutzt werden. Einige Landkreise benutzen größere Pausen zwischen
    den ZVEI-Folgen. Um dies herauszufinden, kann die DUMP-Anweisung
    benutzt werden.

    ZVPAUSE Kennung Zeit
    ZVPAUSE 16 85

```

Mit Kennung sind die ersten zwei Ziffern der ZVEI-Folge gemeint. Die Zeit sollte nicht unter 15 sein, da dann die Folgen selbst nicht mehr angezeigt werden.

Zusätzliche Ausgaben bei ZVEI-Folgen

DUMP 4 Ist diese Option aktiv, werden ZVEI-Folgen im Regelfall doppelt angezeigt. Dahinter steht die jeweilige Zeit bis zur nächsten Folge. Die 4 kann bitweise mit den Einstellungen unter FMS-DUMP verknüpft werden. 4 dezimal entspricht 0100 binär.

6.2. PCSAG

Pocsag-Nummern bestehen aus acht Ziffern, sieben für die Pager-Nummer, die letzte für den Funktionstyp. Aliase werden wie folgt angegeben:

```
PAGER Nummer Text
PAGER 1920193* SA MD ADAC Christoph 36
PAGER 192**** SA MD default
```

Pagertyp

```
FUNC Nummer Text
FUNC *****0 Auftrag
FUNC *****1 Notfall
FUNC *****2 Info
FUNC 13562*1 Einsatz FF
FUNC 13567311 Gefahrgut
```

Globale Farben für PCSAG-Telegramme

```
COLPOC Nummer Farbe
COLPOC ***** 18
COLPOC 1287**** 19
COLPOC 1287701 20
```

Diese Angabe bewirkt dasselbe wie
PAGER Nummer /CFarbe Text
PAGER 1920097 /C5 SA MD FW SEG

Zu Farben siehe 8.2.

Globale Wiederholungsunterdrückung für PCSAG-Telegramme

```
REPPOC Nummer Zeit
REPPOC ***** 30
REPPOC 1287**** 200
```

Die Zeit wird in Sekunden angegeben.
Diese Angabe bewirkt dasselbe wie
PAGER Nummer /RZeit Text
PAGER 19456810 /R30 FW RD Genthin 15

Globale Einstellung der Textzeile für PCSAG-Telegramme

```
TXTPOC Nummer Modus
TXTPOC ***** 0 Text unter Kopfzeile
TXTPOC 087**** 1 Text hinter Kopfzeile
TXTPOC 087**** 2 kein Text
```

Modus kann Werte von 0 bis 2 enthalten.

Unterdrücken und Akzeptieren von Pocsag-Ausgaben

ACCPOC heißt soviel wie "außer" und legt fest, welche Telegramme trotz IGNPOC akzeptiert werden sollen.

```
IGNPOC Nummer
IGNPOC *****2 alle Protokolle mit Funktionswert 2 werden nicht
gezeigt
IGNPOC 01748002 Sperren eines bestimmten Protokolls
```

ACCPOC Nummer

Akzeptieren von Pocsag-Ausgaben mit bestimmten Textteilen

ACCTXTPOC erlaubt die Ausgabe von Protokollen, die mit bestimmten Wörtern beginnen oder bestimmte Wörter enthalten.

```
ACCTXTPOC [*]Wort
ACCTXTPOC Brand Protokolle, beginnend mit "Brand", werden
angezeigt
ACCTXTPOC *VU Protokolle, die "VU" enthalten (*), werden
angezeigt
```

ACCPOC und ACCTXTPOC sind gleichwertig, eine Ausgabe (trotz IGNPOC) geschieht, wenn eines von Beiden zutrifft.

Manipulation von Pocsag-Texten

Hiermit können Textteile ersetzt werden, z.B für Abkürzungen.

```
TXTPOCSUBST "Wort1" "Wort2"
TXTPOCSUBST "ufw" "unbek.FLÜSSIGKEIT (WASSER)"
TXTSUBST "Wort1" "Wort2"
TXTSUBST "374" "374 (84.555 NotfRet H) "
```

Wort1 wird durch Wort2 ersetzt. TXTPOCSUBST gilt nur für Pocsag-Texte, TXTSUBST gilt auch für FMS-Texte.

Steuerzeichen können wahlweise angezeigt werden:

```
STEUERZEICHEN 0 Zeilenumbrüche in <CR> werden durch Semikolon (;)
+ Leerzeichen ersetzt. Andere Steuerzeichen werden
nicht angezeigt
STEUERZEICHEN 1 Alle Steuerzeichen werden in spitzen Klammern aus-
gegeben.
```

Fehlerkorrektur kann wahlweise angezeigt werden:

```
CORRPOC 0 Abbruch bei fehlerhaften Sendungen
CORRPOC 1 Korrektur von Fehlern
Im Allgemeinen genügt eine Auswertung ohne Fehlerkorrektur.
Bei schlechten Verhältnissen können die Ergebnisse mit eingeschalteter Fehlerkorrektur verbessert werden. Das führt allerdings zu vermehrten Fehlerauswertungen, was man wiederum mittels IGNPOC + ACCPOC im Rahmen hält. Vielen Dank an dieser Stelle an Stephan.
```

6.3. FMS

Die **Organisationskennung (BOS)** wird automatisch angezeigt, kann aber umbenannt werden:
BOS Hexzahl Text
BOS d Rettung

Die **Bundesländerkennung (LAND)** wird automatisch angezeigt, kann aber umbenannt werden:
LAND Hexzahl Text
LAND 8 Niedersachsen

Ausnahme bei e und f, weil sich jeweils zwei Länder diese teilen:

die Orte 00-49: 0
die Orte 50-99: 1

```
LAND e0 Mecklenburg-Vorpommern
LAND e1 Sachsen-Anhalt
```

analog Brandenburg/Thüringen

Die **Ortskennung (ORT)** wird zunächst hexadezimal angezeigt, kann aber mit Aliasen versehen werden. Kennungen dreistellig mit vorangehendem Land oder vierstellig mit vorangehendem BOS + Land:
ORT Kennung Text
ORT e80 MD steht für Ort 80 im Land e, also Magdeburg
ORT de62 HBS Rettungsdienst (d), Halberstadt (62), Sachsen-Anhalt (e)

Die **Fahrzeugkennung (KFZ)** wird zunächst hexadezimal angezeigt, kann aber mit Aliasen versehen werden:
KFZ Kennung Text
KFZ 8501 Oschersleben 20-85/1

Die **Richtung (DIR)** wird durch K oder L angezeigt, kann folgendermaßen geändert werden:
DIR Bit Text
DIR 1 ELS

Bit kann 0 oder 1 sein.

Die **Taktische Informationen (TKI)** werden als Römische Zahlen angezeigt, 0..3 entspricht I..IV.
TKI Kennung Text
TKI 0 ohne Wegerecht
TKI 3 mit Wegerecht

TKI von Leitstelle werden nicht ausgegeben.

Die **Status** werden entweder einstellig (0..F) oder vierstellig mit vorangehendem Land und Ort angegeben.

```
STATUS_F_KFZ Kennung Text Status von Feuerwehr-/Rettungsdienst-
fahrzeugen *)1
STATUS_F_LST Kennung Text Direktive von Feuerwehr-/Rettungsdienst-
Einsatzleitstellen *)1
STATUS_P_KFZ Kennung Text Status von Polizei- u.ä. Fahrzeugen *)2
STATUS_F_LST Kennung Text Direktive von polizeilichen Führungszen-
tren *)2
```

*)1 KATS, FW, THW, ASB, DRK, JUH, MHD, DLRG, RD, ZS
*)2 POL, BGS, BKA, ZOLL, FERN

```
STATUS_F_KFZ 3 3-Ausfahrt
STATUS_F_LST f automatische Quittung
STATUS_P_KFZ 1 frei auf Streife
STATUS_F_LST 2 E-Eingesicherung beachten
```

Ausgabe für Hamburg 20 Status 1:
STATUS_F_KFZ 6201 Einsatz übernehmen

Texte, die mit \$ beginnen, werden im Verlaufsfenster nicht angezeigt, jedoch im Layout, im Sprechwunschenster oder hinter "es spricht:". Diese Meldungen erscheinen auch im Logfile.

```
STATUS_P_KFZ f $Sprechtaste
Sprechtaste Polizei-KFZ, wird nur hinter "es
spricht:" angezeigt und ins Logfile geschrieben
STATUS_F_KFZ 1 $frei über Funk
Freimeldung FW/RD, wird nur farbig im Layout
angezeigt und ins Logfile geschrieben
```

Texte, die mit \$\$ beginnen, werden ignoriert und nicht geloggt.

```
STATUS_P_LST f $$
Quittungen Polizei-KFZ werden ignoriert
```

Zu Status und ihren Farben siehe 8.2

Die FORM-Anweisung

Um Kombinationen aus mehreren FMS-Bestandteilen eindeutig mit Aliasen zu belegen, für Falschkodierungen ovm. gibt es die FORM-Anweisung.

FORM Kennung Format

Die Kennung setzt sich zusammen:

```
BLOOXXXXSBRT
| | | | |
| | | | | -- TKI (0 .. 3) für I bis IV
| | | | | -- Richtung (0 oder 1)
| | | | | --- Baustufe (0 oder 1)
| | | | | ---- Status (0 .. f)
| | | | | ----- Fahrzeug (4 Stellen 0 .. f)
| | | | | ----- Ort (2 Stellen 0 .. f)
| | | | | ----- Land (0 .. f)
| | | | | ----- BOS (0 .. f)
```

* (Stern) als Jokerzeichen ist erlaubt

Format setzt sich zusammen:

BOS'Land'Ort'Kfz'Status'Baustufe'Richtung'TKI

Die Bestandteile werden durch Apostrophe ASC(39) getrennt. Die Reihenfolge ist nicht vertauschbar. Einzelne Bestandteile können leer bleiben, die Apostrophe müssen trotzdem stehen. Restliche Apostrophe können weggelassen werden.

Beispiel:

```
FORM ****850***** DRK
Alle KFZ, die mit 850 beginnen (850*), werden mit BOS="DRK"
angezeigt
```

```
FORM 6e810379***** '80
für Fahrzeug FW, LAND SA, KFZ 0379; ORT 81 wird als 80 ausgegeben,
alles andere wird entsprechend der oben erklärten Zuweisungen aus-
gegeben.
```

Autoformatierungen

Innerhalb der FORM-Anweisung können alle Begriffe wie BOS, LAND, KFZ usw. benutzt werden, sie werden dann durch den empfangenen Code ersetzt. An das KFZ muss zusätzlich ein Ausgabeschema angehängt werden, das aus den Zeichen 0¹²³ und gewünschten Trennzeichen besteht. KFZ^{-12/3}, KFZ¹⁻⁴⁻²⁻³ o.ä.

```
FORM 9fff12***** ''Ortsname KFZ1-82/3' STATUS
KFZ wird durch den erkannten Code ersetzt, falls der mit 12
beginnt, z.B. 1232 -> Ortsname 12-83/2, Statusnummer mit vorange-
hendem Leerzeichen; alles andere wird entsprechend der oben
erklärten Zuweisungen ausgegeben.
```

Statusumleitung

Es gibt Landkreise, bei denen werden tatsächlich die Status einiger Fahrzeuge durcheinandergewürfelt. Um wieder Ordnung hineinzubekommen, gibt es zwei Möglichkeiten:

```
FORM 683530101*** ''Ortsname KFZ01-232-Wache
Status 1 wird mit "2-Wache" angezeigt
```

falls das auch noch im Layout mit den richtigen Farben angezeigt werden soll:

```
FORM 683530101*** (1 10 12 "30-10") '' KFZ00-00'>2
als Status 2 ausgeben
FORM 683530102*** (2 10 12 "30-10") '' KFZ00-00'>B
als Status B ausgeben
```

Es werden also mehrere Layouteinträge für das Fahrzeug angelegt und auf dieselbe Stelle geschrieben. Genaueres zum Layout siehe unter 8.
Zu Farben siehe 8.2.

BOS

- 0 nicht vergeben
- 1 Polizei (POL)
- 2 Bundesgrenzschutz (BGS)
- 3 Bundeskriminalamt (BKA)
- 4 Katastrophenschutz (KATS)
- 5 Zoll
- 6 Feuerwehr (FW)
- 7 Technisches Hilfswerk (THW)
- 8 Arbeiter-Samariter-Bund (ASB)
- 9 Deutsches Rotes Kreuz (DRK)
- A Johanniter Unfallhilfe (JUH)
- B Malteser Hilfsdienst (MHD)
- C Deutsche Lebensrettungsgesellschaft (DLRG)
- D andere Rettungsdienste (RD)
- E Zivilschutz (ZS)
- F FERN

Land

- 0 Sachsen (Sax)
- 1 Bund (Bun)
- 2 Baden-Württemberg (BW)
- 3 Bayern 1 (By1)
- 4 Berlin (Ber)
- 5 Hansestadt Bremen (HB)
- 6 Hansestadt Hamburg (HH)
- 7 Hessen (Hes)
- 8 Niedersachsen (NS)
- 9 Nordrhein-Westfalen (NRW)
- A Rheinland Pfalz (RPF)
- B Schleswig Holstein (SH)
- C Saarland (SL)
- D Bayern 2 (By2)
- E Mecklenburg-Vorpommern (MV) / Sachsen-Anhalt (SA)
- F Brandenburg (Bra) / Thüringen (Thü)

Bei der **Farben**-Darstellung wird zwischen der Anzeige im Layout und der Anzeige im Verlaufsfenster unterschieden. **Farben** im Verlaufsfenster können mittels **COLFMS** definiert werden:

```
COLFMS Kennung Farbe
COLFMS 6*****3*1* 5 Status C an FW-Fahrzeuge: Farbe 5
COLFMS d*****3*01 3 ausrückende RD-Fahrzeuge mit TKI II:
Farbe 3
```

Die **Kennung** setzt sich ebenso zusammen wie in der FORM-Anweisung. Mehr zu Farben siehe 8.2.

Ignorieren bzw. Akzeptieren

```
IGNFMS Kennung
IGNFMS ****a***** KFZ a; alle Fahrzeuge, die mit a beginnen, werden nicht angezeigt
```

Die **Kennung** setzt sich ebenso zusammen wie in der FORM-Anweisung.

ACCFMS heißt soviel wie "außer" und legt fest, welche Telegramme trotz **IGNFMS** akzeptiert werden sollen; z.B. Unterdrückung aller Telegramme aus Sachsen, außer der Polizei:

```
IGNFMS *0***** Sachsen wird ignoriert :)
ACCFMS 10***** Polizei Sachsen wird ausgegeben
```

ACCFMS ist also nur sinnvoll in Verbindung mit **IGNFMS**. **ACCFMS** steht in der Hierarchie über **IGNFMS**.

Aktionen (ACTFMS) siehe 9.

Manipulation von FMS-Texten

Hiermit können Textteile ersetzt werden, z.B für Abkürzungen.

```
TXTFMSSUBST "Wort1" "Wort2"
TXTFMSSUBST "KI_LI" "KTW liegend"

TXTSUBST "Wort1" "Wort2"
TXTSUBST "374" "374 (84.555 NotfRet H) "
```

Wort1 wird durch **Wort2** ersetzt. **TXTFMSSUBST** gilt nur für FMS-Texte, **TXTSUBST** gilt auch für Pocsag-Texte.

Ausgabe von automatischen Quittungen

Als "automatische **Quittungen**" werden hier der Status F (ELS) und der Status E (KFZ) bezeichnet.

```
QUITTUNG Modus
QUITTUNG 0 Leitstellenquittungen werden nicht ausgegeben
QUITTUNG 1 Leitstellenquittungen werden normal ausgegeben
QUITTUNG 2 Ausgabe nur, falls Status nicht verstanden wurde
```

Das **Sprechwunschfenster** wird mit **SPRWLINES** definiert und kann zwei bis neun Zeilen enthalten. Ohne diese Angabe wird das Fenster nicht angezeigt.

```
SPRWLINES Zeilen
SPRWLINES 2 Kopfzeile und zwei Zeilen
```

Was im **Sprechwunschfenster** oder hinter **es spricht:** erscheint, wird aus der Zeile direkt unter der Legende extrahiert. Die Formatierung beginnt mit **PTT**, darauf folgen Zahlenpaare, wovon die erste das erste Zeichen, die zweite die Länge markiert. Es werden maximal 16 Zeichen angezeigt.

```
PTT a1,a2[,b1,b2 ... ,n1,n2]
PTT 13,8,26,7
erzeugt aus
"12:29:48 SA MD JUH Lerchenwuhne 81-85/4 Sprechtaete"
dieses: "MD JUH 81-85/4"
```

Temporäre Anzeigen

Dazu gehören Anzeigen hinter **es spricht:**, im **Sprechwunschfenster** oder **Alarmierungen** im Layout. voreingestellte Anzeigedauern, zu ändern mittels:

```
es spricht 30 s DELAYPTT
Sprechwunsch 5 min DELAYSPR
Alarmierung 5 min DELAYALARM
```

```
DELAYPTT Zeit [sec]
DELAYPTT 180
```

```
DELAYSPPR Zeit [sec]
DELAYALARM Zeit [sec]
```

Steuerzeichen können wahlweise angezeigt werden:

```
STEUERZEICHEN 0 Zeilenumbrüche in <CR> werden durch Semikolon (;)
+ Leerzeichen ersetzt. Andere Steuerzeichen werden nicht angezeigt
STEUERZEICHEN 1 Alle Steuerzeichen werden in spitzen Klammern ausgegeben.
```

Auswertepiegel für FMS-Telegramme

Jedes Rauschen erzeugt eine wirre Folge von Nullen und Einsen. Solche Flagen werden gelegentlich als Pseudo-FMS-Telegramme ausgegeben. **SQUELCHFMS** bestimmt dabei, dass jede Null oder Eins aufgrund eines genügend starken Signals zustande gekommen sein muss.

SQUELCHFMS Wert

Wert ist eine Zahl mit einem recht großen Bereich (etwa 0..100000). Das auswertende Signal ist abhängig von der Stärke des Senders, dem Pegel am Scannerausgang und dem Pegel des Line-in-Reglers. Kleine Änderungen haben sehr große Auswirkungen. Wer **SQUELCHFMS** sollte also für konstante Verhältnisse sorgen.

Einen brauchbaren Wert kann man mit Hilfe von **DUMP** herausfinden, wo zu jedem Protokoll der aktuelle Mindestpegel angezeigt wird. **SQUELCHFMS** darf nie über dieses Minimum eingestellt werden, da sonst diese Telegramme nicht mehr angezeigt werden. Wer sicher gehen will, stellt hier einen kleinen Wert ein.

Synchronisation und Fehlerprüfung

Um die Ausgabe eines FMS-Telegramms zu erreichen, müssen zwei Bedingungen erfüllt sein: Zum einen muss die Blocksynchronisation erkannt werden, zum anderen muss das Codewort eine Fehlerprüfung überstehen.

```
FMS-Telegramm:
1111111 00011010|0101 0101 0101 0101 0101 0101 0101 0101 0101010 0
Vorlauf | Sync |BOS |ORT | KFZ |Stat|BRK|CRC |0
verlängertes Sync| 47-Bit-CODEWORD |Sync
```

lussbit

Mit **SYNCBITS** kann man die Länge der Blocksynchronisation frei wählen. Oft ist es so, dass die Relais spät öffnen oder der Scanner zu spät am Kanal stoppt, sodass die Blocksynchronisation nicht mehr empfangen bzw. erkannt wird, das interessante Codewort aber schon. Dafür ist es sinnvoll, **SYNCBITS** kleiner als 8 anzugeben.

```
SYNCBITS Bitzahl {0..16}
SYNCBITS 5 benutzt 11010
```

Die Anweisung **CRC** beeinflusst die Fehlerprüfung. **CRC Modus** {0..1}

Modus kann 0 oder 1 sein. Bei **CRC 0** werden fehlerfreie Telegramme ausgegeben und Ein-Bit-Fehler (blind) korrigiert. Andere werden ignoriert. **CRC 1** lässt nur fehlerfreie Telegramme zu. Trotz alledem kommt es nicht selten vor, dass zufällig Codewörter empfangen werden, die nach **CRC-Check** fehlerfrei sind. Es muss jeder selbst ausprobieren, mit welchem Verhältnis von **SYNCBITS** und **CRC** die besseren Ergebnisse erzielt werden können.

```
SYNCBITS 0 und CRC 0 lassen fast alles durch,
SYNCBITS 16 und CRC 1 hingegen nur fehlerfreies.
SYNCBITS 8 und CRC 1 ist Standard.
```

Außerdem gibt es noch die **IGNFMS**-Anweisung.

Zusätzliche Ausgaben bei FMS-Telegrammen

```
DUMP 1 zusätzliche Ausgabe von Hexdaten (auch $Status)
DUMP 2 Hexdaten für Ignoriertes (IGNFMS) Die 1, die 2 auch die 4
für ZVEI-DUMP können bitweise verknüpft werden. 1 dezimal
entspricht 0001 binär, die 2 entspricht 0010.
```

Erweiterte Ortsausgabe

EXTORT 1

Das ist für jene gedacht, die auf die Ausgabe **LAND** im FMS-Telegramm verzichten; so wird bei Telegrammen, zu denen keine **ORTS**-Alias existiert, der einstellige **LANDES**-Code und der zweistellige **ORTS**-Code angezeigt.

```
22:05:24 DRK MD 61-83/1 7 mit ORTS-Alias
22:05:24 DRK 80 61-83/1 7 ohne ORTS-Alias, ohne EXTORT
22:05:24 DRK E80 61-83/1 7 ohne ORTS-Alias, mit EXTORT
```

7. Kommentare

Kommentare, beginnend mit #, sind immer möglich und werden ignoriert

8. Layout

8.1. Gestaltung

Wenn man will, kann man sich eine Status-Übersicht mit interessanten Fahrzeugen zusammenbasteln. Das **Layout** kann aus mehreren Seiten bestehen, die Seitenzahl ergibt sich aus der größten angegebenen Seitenzahl (**FORM**, **LABEL**). Der Platz dafür wird unten durch die Zeile mit Uhrzeit und Datum, der **Legende**, begrenzt. Sie muss dazu nach unten verschoben werden. **LEGENDE** Zeile

```
LEGENDE 19 Zeit, Datum usw. in der 19. Zeile
```

Mit **LABEL** kann man Text in den Layout-Bereich schreiben. **LABEL Seite Zeile Spalte Farbe Text**

Die erste Seite hat die Nummer 0. Erste Zeile und Spalte haben die Nummer 1.

```
LABEL 10 2 12 16 Feuerwehr #
```

Die Definition der Fahrzeuge im **Layout** geschieht in der **FORM**-Anweisung in der \$HOME/.monrc durch einen Klammersausdruck:

```
FORM Kennung (LayoutNr Zeile Spalte "Text") Format
FORM ****114a**** (33 03 18 "11-42") ''Olvenstedt 11-42
```

Die Nummern dürfen nicht doppelt vergeben werden. Nummern 0-999 erscheinen auf der ersten Layoutseite, 1000-1999 auf der zweiten usw. (Blättern mit Cursor links und rechts)

Fügt man diese Nummer auch bei entsprechenden ZVNAME- oder PAGER-Anweisung ein, wird auch ein Alarmierungsstatus im Layout angezeigt. Auch hier gehören diese Nummern in Klammern, es können auch mehrere Nummern angegeben (mehrere Fahrzeuge gleichzeitig alarmiert) werden.

```
ZVNAME Nummer (LayoutNr) Text
ZVNAME 11504 (110) SA SBK FW Schönebeck 90-82

PAGER Nummer (LayoutNr) Text
PAGER 1920064 (30 31 32)
SA MD FF2 Olvenstedt 11

Der aktuelle Layoutzustand wird in der Datei $HOME/.layout gespeichert.
Die Taste <L> löscht die aktuelle Layoutseite zurück.
```

8.2. Farben

a. Definition

Die Status können im Layout und im Verlaufsfenster durch unterschiedliche Farbgebung sichtbar gemacht werden. Dazu müssen allerdings Farben definiert werden:

COLOR Farbnummer Vordergrund Hintergrund [Optionen]

```
0 schwarz
1 rot
2 grün
3 braun (3 fett: gelb)
4 blau
5 magenta
6 cyan
7 weiß
```

Optionen: NUR in alphabetischer Reihenfolge angeben

```
B blinkend
F fett
R revers/invers
U unterstrichen
```

komplettes Set für die .monrc:

```
COLOR 1 2 7 #Status1
COLOR 2 0 7 #Status2
COLOR 3 7 1 #Status3
COLOR 4 0 3 #Status4
COLOR 5 5 7 B #Alarmierung
COLOR 6 7 0 #Status6
COLOR 7 7 4 #Status7
COLOR 8 0 6 #Status8
COLOR 9 4 7 #Status9
COLOR 10 0 7 #Initialisierung, Layouthintergrund und Reset
COLOR 11 7 4 FU #Sprechwunsch-Fenster + Überschrift
COLOR 12 7 4 B #Sprechwunsch 1. Zeile
COLOR 13 7 4 #Sprechwunsch ab 2. Zeile
```

Farben

1-4, 6-9

werden, falls nicht anders angegeben, automatisch entsprechend dem ankommenden KFZ-Status angezeigt, Leitsteldirektiven werden zunächst nicht im Layout angezeigt.

5 wird als Alarmierungsstatus verwendet.

10 ist der Layouthintergrund und Initialisierung.

11-13 für die Zeilen des Sprechwunschfensters

14 für Peg spricht:"

15 reserviert

16-50 frei definierbar

Die Farbe 0 wird benutzt, um keine Änderung zu bewirken.

Wurde eine Farbe nicht definiert, wird die Einstellung der Konsole/xterm/kvt verwendet.

b. zusätzliche Farben für Status

Abweichende Farben für die Status werden mit /C<Farbnummer> angegeben.

```
STATUS_F_KFZ 5 /C11 Sprechwunsch
Sprechwunsch wird mit Farbe 11 festgelegt
```

```
STATUS_F_LST 3 /C5 C-Für Einsatzübernahme melden
Anzeige des Alarmierungsstatus' 5
```

```
STATUS_F_KFZ 6201 /C3 Einsatz übernommen
Ausgabe für Hamburg 20 Status 1 mit Farbe 3
```

Verwenden zusätzlicher Farben (16-50):

```
STATUS_P_KFZ 7 /C16 über 2-Meter-Funk
```

```
STATUS_F_KFZ 9 /C0 Handquittung
KFZ-Status 9 wird im Layout nicht angezeigt
```

c. farbliche Anzeige im Verlaufsfenster:
Dabei wird die ganze Zeile mit einer Farbe dargestellt.

für ZVEI und POCSAG mit /C<Farbnummer>:
ZVNAME 16*** /C3 SA SDL
PAGER 1920097 /C25 SA MD FW SEG

für FMS wird das mit COLFMS festgelegt:
COLFMS 6*****3*1* 5 #Status C an FW-Fahrzeuge mit Farbe 5
COLFMS d*****3*01 3 #ausrückende RD-Fahrzeuge mit TKI II mit Farbe 3

9. Aktionen

Sollen bei ausgewählten ZVEI- oder POCSAG-Alarmierungen oder DTMF-Folgen andere Programme gestartet werden, können diese in der ZVNAME-, PAGER- bzw. DTMF-Anweisung in eckigen Klammern angegeben werden. Mit der ACTFMS-Anweisung ist das auch für FMS-Telegramme möglich. Diese Programme müssen eventuell einen Pfad enthalten. Mehrere Programme werden durch ein Semikolon getrennt und werden nacheinander ausgeführt. Da einige Programme selbst Ausgaben produzieren, die im monitor-Fenster erscheinen würden, ist es besser, die Aufrufe entweder in einem xterm (unter XWindow) oder über eine Batch-Datei aufzurufen und die Ausgaben umzuleiten.

ACTDTMF Folge [Programm]

Weiterhin ist es bei ZVNAME möglich, an die Nummer den Typ anzuhängen; dieser wird für die Ausführung des Programms benutzt:

ZVNAME Nummer[+Typ] [Programm] Text

Typ ist ein einzelner Buchstabe:

```
M Melder
F Feuer (DTMF)
P Probe (DTMF)
W Warnung (DTMF)
Z Zivilschutzalarm (DTMF)
```

E Entwarnung (DTMF)

```
ZVNAME 17729 [xterm -e "prog1";prog2] SA OK DRK 97-82
oder
ZVNAME 17729 [xterm -e "prog1"] [prog2] SA OK DRK 97-82
ZVNAME 62247M [play bilm.wav] FF H-SND Bilm 46
ZVNAME 62247F [play bilm.wav;play sirene.wav] FF H-SND Bilm 46
ZVNAME 62247E [play bilm.wav] [play sirene.wav] FF H-SND Bilm 46
```

Es ist weiterhin möglich, Durchsagen nach ZVEI-Alarmierungen aufzuzeichnen. Dazu wird als Programm [frec] (in eigenen []-Klammern) aufgerufen. Weiterhin ist dafür das Programmpaket sox notwendig.
ZVNAME 17729 [programm] [frec] SA OK DRK 97-82

In der .monrc empfehlen sich folgende Einstellungen:

```
REC_PATH Pfad
REC_PATH wav-files
Der Pfad wird relativ zum $HOME-Verzeichnis angegeben.
```

```
REC_TIME Zeit in sec
REC_TIME 15
```

```
REC_VOL Lautstärke in %
REC_VOL 100000
```

Achtung! Dies ist eine nette Möglichkeit, sich die Festplatte aufzufüllen. Eine Minute Wav-Datei benötigt zirka 1,3 MB. Vielen Dank an dieser Stelle an Godfried van Berkel aus NL.

In der POCSAG-Anweisung kann zusätzlich %T verwendet werden, um den Pagertext als Argument an Programme zu übergeben. Mit %R wird die Nummer übergeben. Um zu überprüfen, was letztendlich aufgerufen wird, kann man SYSCALLLOG benutzen.

```
SYSCALLLOG Modus
SYSCALLLOG 0 Kein Mitloggen
SYSCALLLOG 1 Aufzurufendes Programm wird im Logfile eingetragen.
```

```
PAGER Nummer [Programm] Text
PAGER 1937920 [echo %R:%T > /dev/lp0] SA ASL DRK Aschersleben 81
```

Bei FMS wird ACTFMS benutzt:
ACTFMS 1*****31** [cat wav/C.wav > /dev/dsp] #Ton bei C (Lst 3) für POL

Wav-Dateien können bei einer Eingangslautstärke von etwa 20% und einer Wav-Lautstärke von etwa 95% abgespielt werden. Auch ohne Verschieben der Balance sollte das Diskriminatorrauschen so nicht mehr stören. Wav-Dateien sollten in 8 Bit mono 22kHz vorliegen.

Batch-Datei z.B.: #!/usr/bin/sh programm > /dev/ttyS1 #ggf. Rechte für Device ändern!

10. Logfiles

Mit der Anweisung

```
LOGPATH Pfad
können Log-Dateien erzeugt werden. Fehlt sie, gibt es kein Logging. Der Pfad wird relativ zum $HOME-Verzeichnis angegeben. "." schreibt die Log-Dateien direkt ins $HOME. Das Erstellungsdatum steht im Dateinamen.
```

11. Ressourcenbelastung

Natürlich benötigt jedes Programm Ressourcen. Vor allem wird die Soundkarte benutzt, welche durch andere Programme nicht mehr oder nur noch eingeschränkt benutzbar ist. Außerdem wird natürlich der Prozessor benutzt. Was ich einst gemessen habe:

```
Pentium 100 Linux ~28%
AMD-K6 233 Linux ~14%
SUN sparc 300 SunOS ~5.2%
(Dekodierung von FMS, ZVEI, DTMF, POCSAG1200, ein Kanal)
```

Je mehr Module aktiv sind, desto höher die Prozessorbelastung.

12. Signalerzeugung

Außer dem Programm monitor gibt es ein weiteres Programm namens gen, mit welchem verschiedene Signale erzeugt werden können. Optionen:

```
-a <ampl> Amplitude auf ein anderes Signal
-d <str> DTMF [0123456789ABCD*#EFPWZ] (E:Entwarnung,
F:Feueralarm,
P:Probealarm,
W:Warnung,
Z:Zivilschutzalarm)
```

```
-p <text> HDLC packet radio text
-f <BLOKKBKSBRYX> FMS-Telegramm
-s <freq> Sinusfrequenz
-z <str> ZVEI [bd1234567890ef] (e: Wiederholton, f: Pause)
-t format test.format: generiert RAW-Files, andere Sound-Formate benötigen sox
```

```
gen -z 1710effffff1710efffff; gen -d F
gen -z 60e43fffff60e43fffff; gen -s 2600; gen -s 2600
gen -f 635991151000
```

monitor basiert auf einem anderen namens multimon, welches unter selbiger URL gefunden werden kann (siehe unten)
Für konstruktive Kritik, Meldungen unentdeckter Fehler und Tips bin ich immer offen. Prinzipiell freue ich mich natürlich mehr über Lob als über Kritik ;-)

Viel Spaß Markus Grohmann markus_grohmann@gmx.de http://home.t-online.de/markus.grohmann/mon

Monitor decoding software configura^H2^H0i^H0o^H2n^H/1f^H1i^H/1^H1e^H3 monrc.man(n)