

Digitale Kommunikation im Amateurfunk



Was man so braucht

- *Eine Amateurfunklizenz zum Senden !!!*
- Einen (alten) PC - selbst ein 386/4 Mbyte tut's noch (für DOS), aber die Windows-Programme sind inzwischen besser, und man kann nebenher noch Logbuch führen oder die Antenne drehen!
- Ein Modem zwischen DM 10,- (HamComm mit einem μ 741) und DM 1500,- (PTC-II voll ausgebaut)
- oder aber eine Creative-Soundblaster (SB16, ... SB128)-kompatible Soundkarte für ISA oder PCI (DM 18,-)
- Eine geeignete Software zwischen 0 DM und 150 DM
- Ein SSB-Funkgerät für KW oder ein FM-Gerät für UKW
- möglichst ein 500 Hz ZF-Filter für KW, DSP schadet nicht...

Zunächst ein wenig Theorie

Codierung



Protokolle

Übertragungstechnik



Analog Digital

- Analoge Übertragungstechnik
 - **Modulation eines Hochfrequenzsignals mit dem niederfrequenten Nutzsignal (kontinuierlicher Frequenzbereich 50Hz - 20kHz):**
 - Sprache: AM, SSB, FM
 - Bildinformation (Helligkeit, Farbe): SSTV, FAX
- Digitale Übertragungstechnik:
 - **Tastung eines Hochfrequenzsignals im Takt des binären Nutzsignals**
 - **Modulation eines Hochfrequenzsignals mit dem bitweise codierten Nutzsignal**
 - **Codierte** Text- und Bildübertragung: CW, RTTY, PSK31, FEC, Amtor, Pactor, AX.25, IP (UDP/TCP)
 - Zwei grundsätzliche Verfahren:
 - **Verbindungsorientiert**: Je zwei Stationen sind verbunden („connected“)
 - **Verbindungslos**: Einer sendet, alle hören zu („broadcast“)

Codierung: Was heißt „codiert“

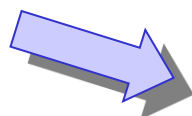
- Darstellung einzelner sicht- und druckbarer Zeichen oder unsichtbarer Steuerzeichen durch eine Kombination aus zwei Zuständen „0“ und „1“ mit variabler oder fester Zeichenlänge
 - Beispiel Morse-Code (variabel): **A** = • — **B** = — • • •
 - Beispiel Baudot-Code (fest, 5 bit): **A** = 11000 **B** = 10011
- Diese Zeichen werden im Computer durch binäre Zahlen dargestellt, meist **8 bit = 1 Byte** pro Zeichen. Das ergibt maximal $2^8 = 256$ verschiedene Zeichen (00000000...11111111)
- Elektronisch werden die Binärzahlen (logisch 0 oder 1) durch zwei „Kennzustände“ repräsentiert:
 - **Strom / kein Strom** (= Mark/Space beim Fernschreiber)
 - **Hi / Lo** (+5V / 0V) in TTL-Logic
 - **Tonfrequenzen**, beispielsweise **Mark / Space** = 1275 Hz / 1445 Hz

Binärzahlen und ihre Darstellung

Binärzahl 2^n	Oktalzahl 8^n	Dezimalzahl 10^n	Hexadezimalzahl 16^n
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F
1 0000	20	16	10
1 0001	21	17	11
....			
1111 1111	377	255	FF

Häufig benutzte Zeichensätze

- Morse (variable Wortlänge, CW)
- Varicode (variable Wortlänge, PSK31)
- Baudot (feste Wortlänge, 5 bit, CCITT Nr. 2, RTTY)
- ASCII (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange, 7 oder 8 bit)
- Huffman-Code (feste Wortlänge, 8 bit, Pactor)
- DOS / OEM („IBM-Zeichensatz“, 8 bit, aber nicht EBCDIC !!)
- Windows ANSI (**A**merican **N**ational **S**tandards **I**nstitute, 8 bit)
- ISO 8859 (**S**ingle **B**yte **C**haracter **S**et, 8 bit)
 - **1: Latin alphabet No. 1 und Nr. 2 (zentraleuropäisch)**
- ISO/IEC 10646 Unicode Standard UCS (**M**ulti **B**yte **C**haracter **S**et, 16 bit)
 - **Architecture and Basic Multilingual Plane, ISO/IEC 10646**
 - **Version 1: 1993**
 - **Version 2: 1996**

 **internationales Webdesign**

Beispiel: Varicode und ASCII-Zeichensatz

PSK31-Varicode
(variabel)

p 11111
q 11011111
r 10101
s 10111
t 101
u 110111
v 1111011
w 1101011
x 11011111
y 1011101
z 111010101
{ 1010110111
| 110111011
.....

0-127

ASCII 0 - 255
(fest, 8 bit)

!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y	z	{		}	~	

128-255

ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	®	¯		
°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ



Übertragungstechnik

oder

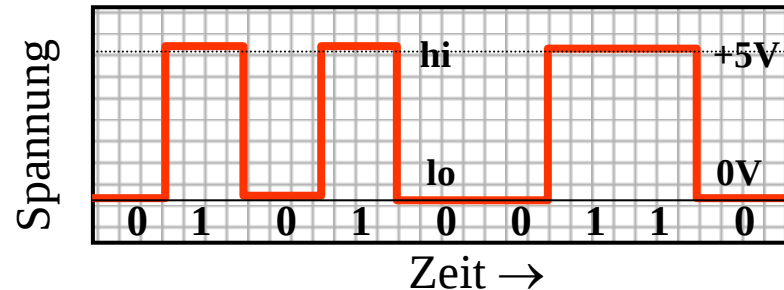
wie kommen die Daten von meiner Tastatur
auf den Bildschirm des QSO-Partners ?

Verbindungsart

- Verbindungslose Kommunikation („broadcast“, 1:n)
 - Beispiele: Sprechfunk, Funkfernschreiben, PSK31, Morsen
 - Einer sendet, alle hören zu (oder senden auch gleichzeitig 😊)
 - **Vorteil:** Jeder kann jederzeit mithören und „einsteigen“, schnell
 - **Nachteil:** Kein „Handshake“, daher keine Flußkontrolle, keine Fehlererkennung (ohne Redundanz) und keine Fehlerkorrektur
- Verbindungsorientierte Kommunikation (1:1)
 - Beispiele: Amtor, Pactor, AX.25 (Packet), TCP (ein IP-Protokoll)
 - Erst wird eine feste Verbindung zwischen zwei Stationen aufgebaut, dann werden über diese Verbindung Daten ausgetauscht
 - **Nachteil:** Langsamer, nur immer zwei Stationen können miteinander „reden“, dritte können bestenfalls mithören
 - **Vorteil:** Automatische Flußkontrolle und Fehlerkorrektur

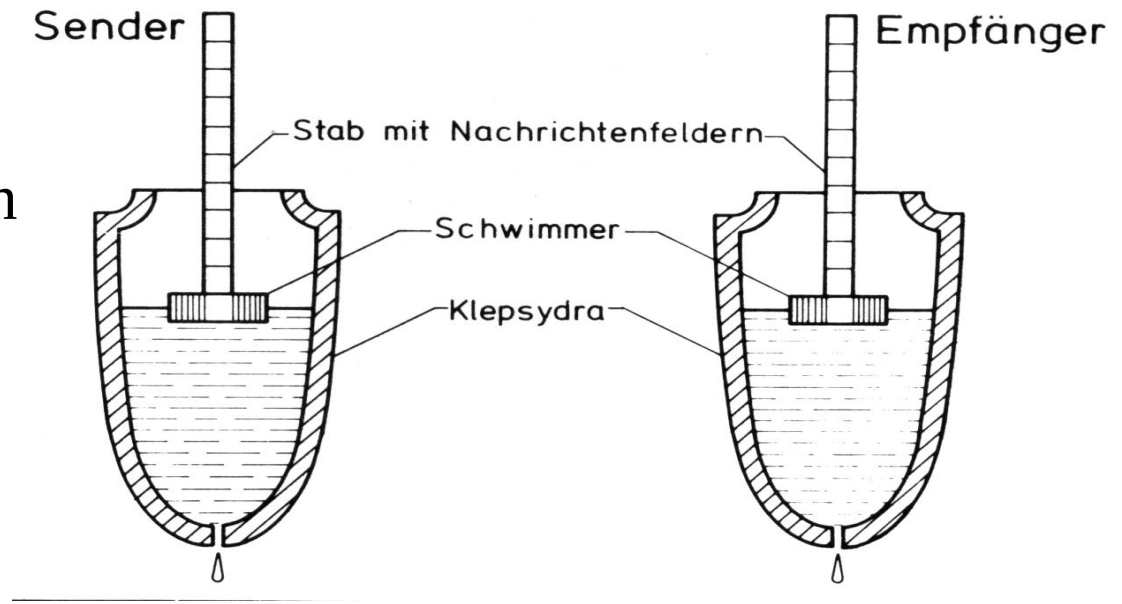
Übertragungstechnik

- Die „0“ und „1“ eines binären Codes werden nacheinander in einem festen Zeitmaß als zwei unterschiedliche Töne ausgesendet:
- Asynchrone Übertragung:
 - Der Sender synchronisiert den Empfänger regelmäßig (z.B. nach jedem Zeichen) durch Aussendung zusätzlicher Start- und Stop-Impulse.
 - Vorteil: Geringe Anforderungen an den Gleichlauf bei Sender und Empfänger, aber geringere Nutzrate durch zusätzliche Sync-Zeichen
- Synchrone Übertragung
 - Sender und Empfänger laufen jeder für sich im gleichen Takt.
 - Vorteil: Höhere Effizienz (Nutzrate), aber auch höhere Anforderungen an den Gleichlauf von Sender und Empfänger



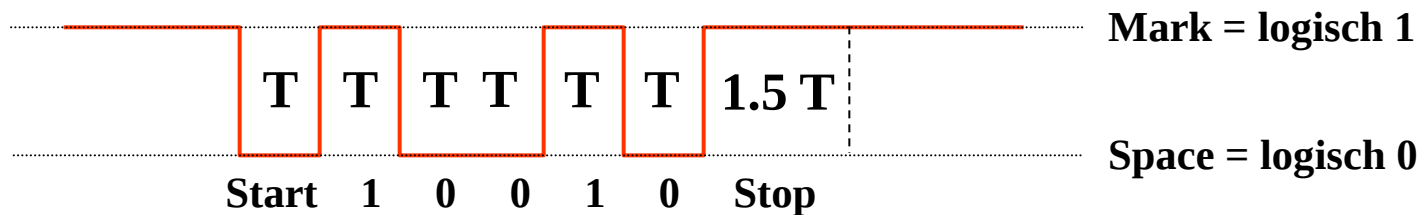
Synchronübertragung - wie geht das ?

- Vorschlag von *Aineias dem Taktiker* (erste Hälfte 4. Jh.v.Chr.)
- Ein Feuerzeichen gibt den Start bei Sender und Empfänger. Damit wird das Tropfloch geöffnet und das Wasser fließt aus.
- Ein weiteres Feuerzeichen zeigt den Stop an, Sender und Empfänger laufen (bei gleichen Töpfen und gleichem Loch) synchron, und am Stab kann die Nachricht abgelesen werden.



Asynchroner Start/Stop-Betrieb

- Serielle Übertragung „bit für bit“, die Synchronisation zwischen Sender und Empfänger erfolgt für jedes übertragene Zeichen neu
 - Logische 1 = „Mark“ = Ruhezustand (Fernschreibrelais angezogen)
 - Logische 0 = „Space“ = aktiv (Relais stromlos, Trommel läuft)



- Schrittgeschwindigkeit (Baudrate) $v = 1/T$ in Baud**
 - Für RTTY (5 bit Code) $\rightarrow$ 1 Zeichen = Start + 5 bit + Stop = 7.5 T
 - Bei 45.45 Baud $\rightarrow T = 1\text{sec}/45.45 = 22 \text{ msec}$
 - Damit wird die **Übertragungsrate** bei RTTY

$$V = 60 \text{ sec} \cdot 45.45 \text{ Schritte pro Sekunde} / 7.5 \text{ Schritte pro Zeichen}$$

$$= 363 \text{ Zeichen pro Minute}$$

Tastung des Senders

- **F**requency **S**hift **K**eying (FSK). Direkte Tastung des VFO zur Frequenzumschaltung durch das Tastsignal (z.B. 0/+5 V TTL)
- **A**udio **F**requency **S**hift **K**eying (AFSK). Tasten des Einseitenbandsignals durch Umschaltung zwischen zwei Tonfrequenzen (SSB) oder direkte Aussendung der Töne (FM)
- Der Frequenzabstand der beiden Töne heißt „**Shift**“
 - **Shift** = **2 • Hub** (Abweichung von der Mittenfrequenz)
- Die erforderliche Bandbreite errechnet sich bei gegebener Shift zu: $B = 2 \cdot (Hub + 1.6 \cdot Baudrate)$ (Kompromißformel)
- Die Zuordnung der Töne zu den Kennzuständen „Mark“ und „Space“ ist je nach Betriebsart (RTTY, Packet) unterschiedlich
- Sie ist auch für die IARU-Region 1 („Lowtones“) verschieden vom Rest der Welt („Hightones“) (wen wunderts?)

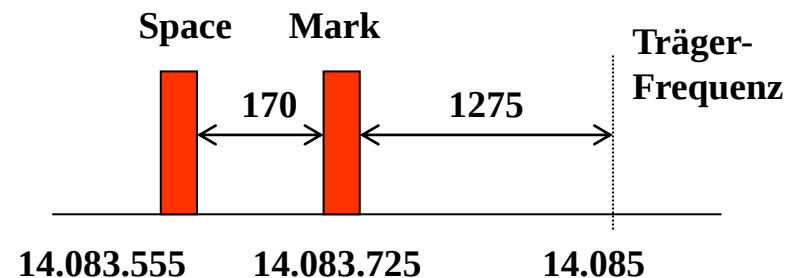
Die Shiftlage „Normal“ und „Invers“

- Die Zuordnung der IARU-Tonfrequenzen zu den Kennzuständen „Mark“ und „Space“ ist bei FSK (Kurzwelle) nur durch ihre HF-Lage definiert !!
- Beispiel RTTY (Lowtones) auf 14.085 MHz in der für RTTY „üblichen“ Sendart LSB in der **Shiftlage „normal“**:

- IARU-Töne für RTTY mit 170 Hz Shift (Region 1)

- „Mark“ = 1272 Hz
- „Space“ = 1445 Hz

- Das **hochfrequenter** HF-Signal (niederer NF-Ton) ist dann „**Mark**“ und liegt bei 14.083.725 Hz
- Das **niedrigerfrequenter** HF-Signal (hoher NF-Ton) ist folglich „**Space**“ und liegt bei 14.083.555 Hz
- Shiftlage „**Invers**“ wird mit vertauschten Mark/Space-Tönen oder durch Umschalten in USB-Mode realisiert

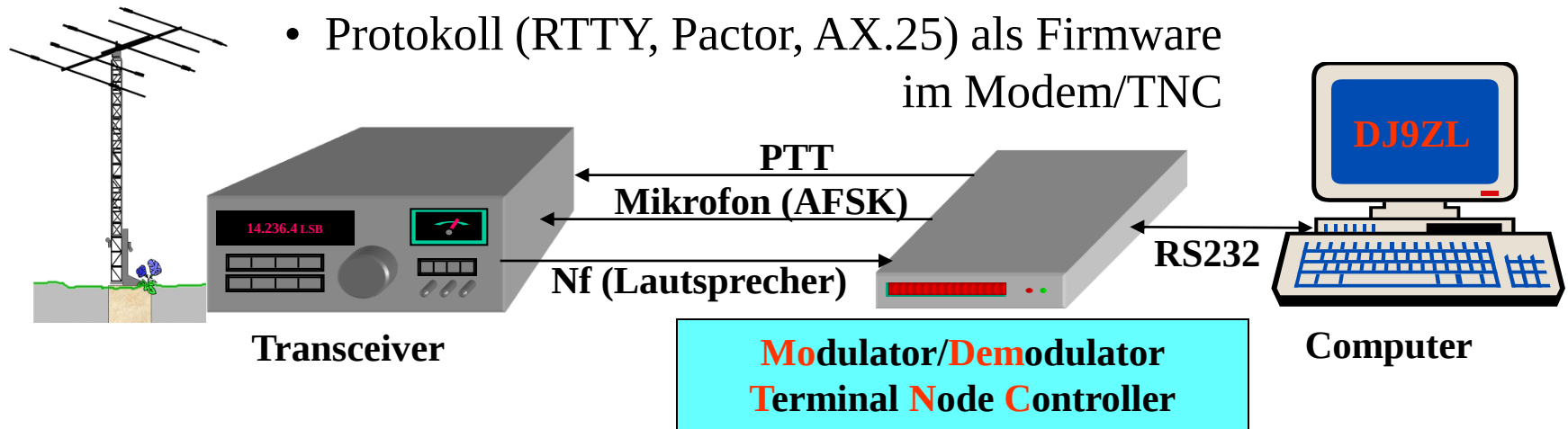


Hardware

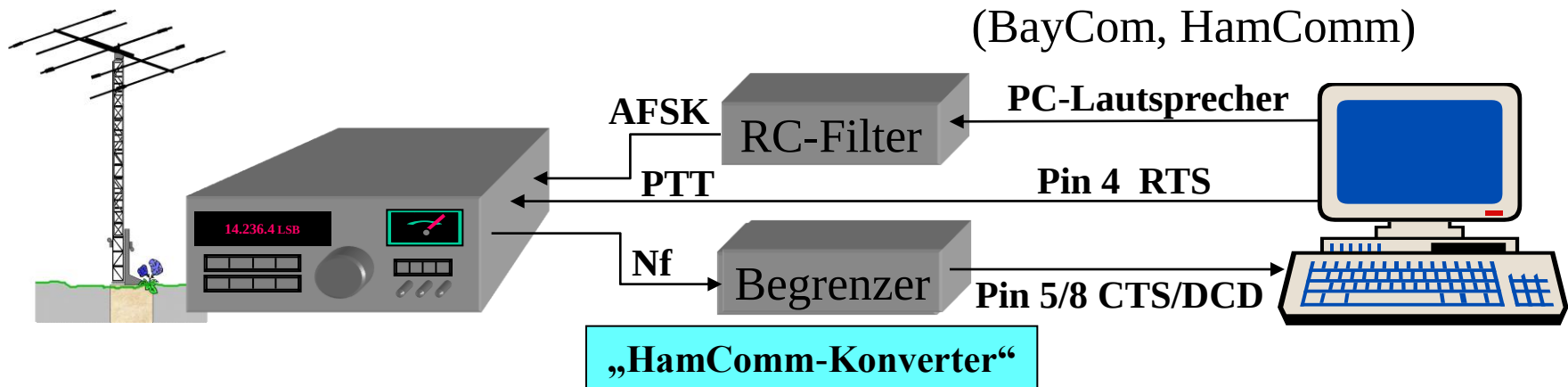
Und
einige Tipps zur Beschaltung

Schema einer Digitalstation

- Protokoll (RTTY, Pactor, AX.25) als Firmware im Modem/TNC



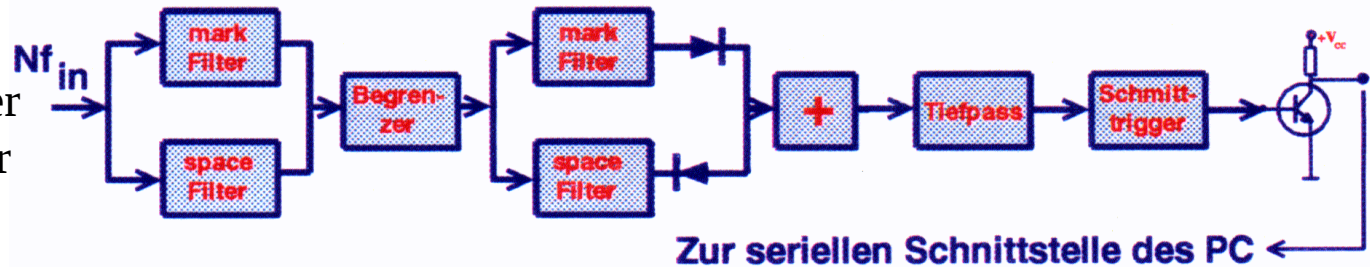
- Protokoll wird vollständig durch Software im PC realisiert (BayCom, HamComm)



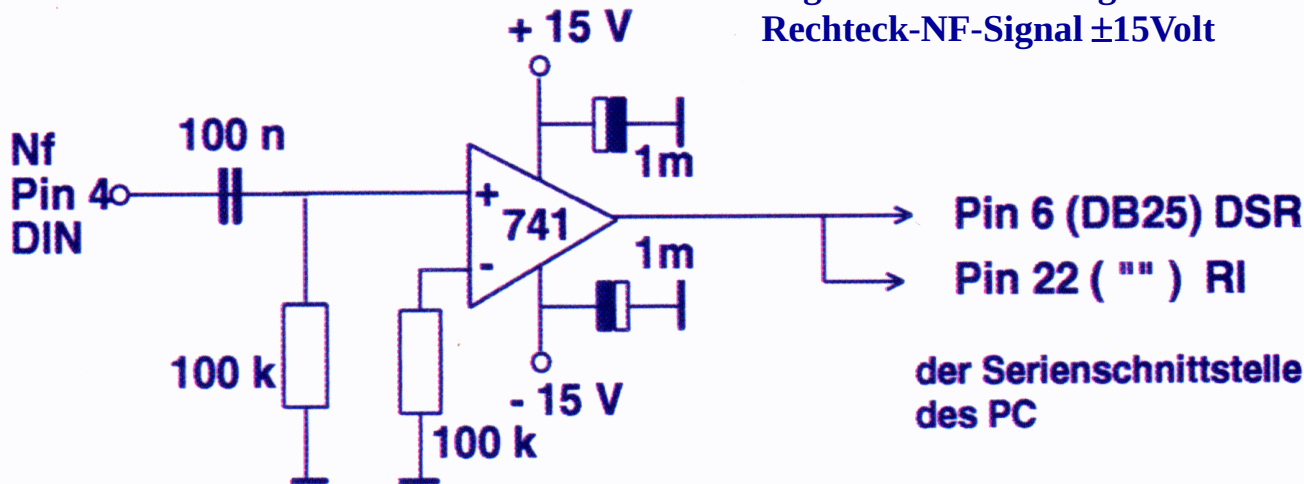
Filter- und HamComm-Konverter

- Prinzip des Filterkonverters nach DJ6HP für RTTY, Amtor, Pactor

Nf vom Empfänger
(Lautsprecher oder
Phonepatch Out)



Begrenzerstufe erzeugt ein
Rechteck-NF-Signal $\pm 15\text{V}$

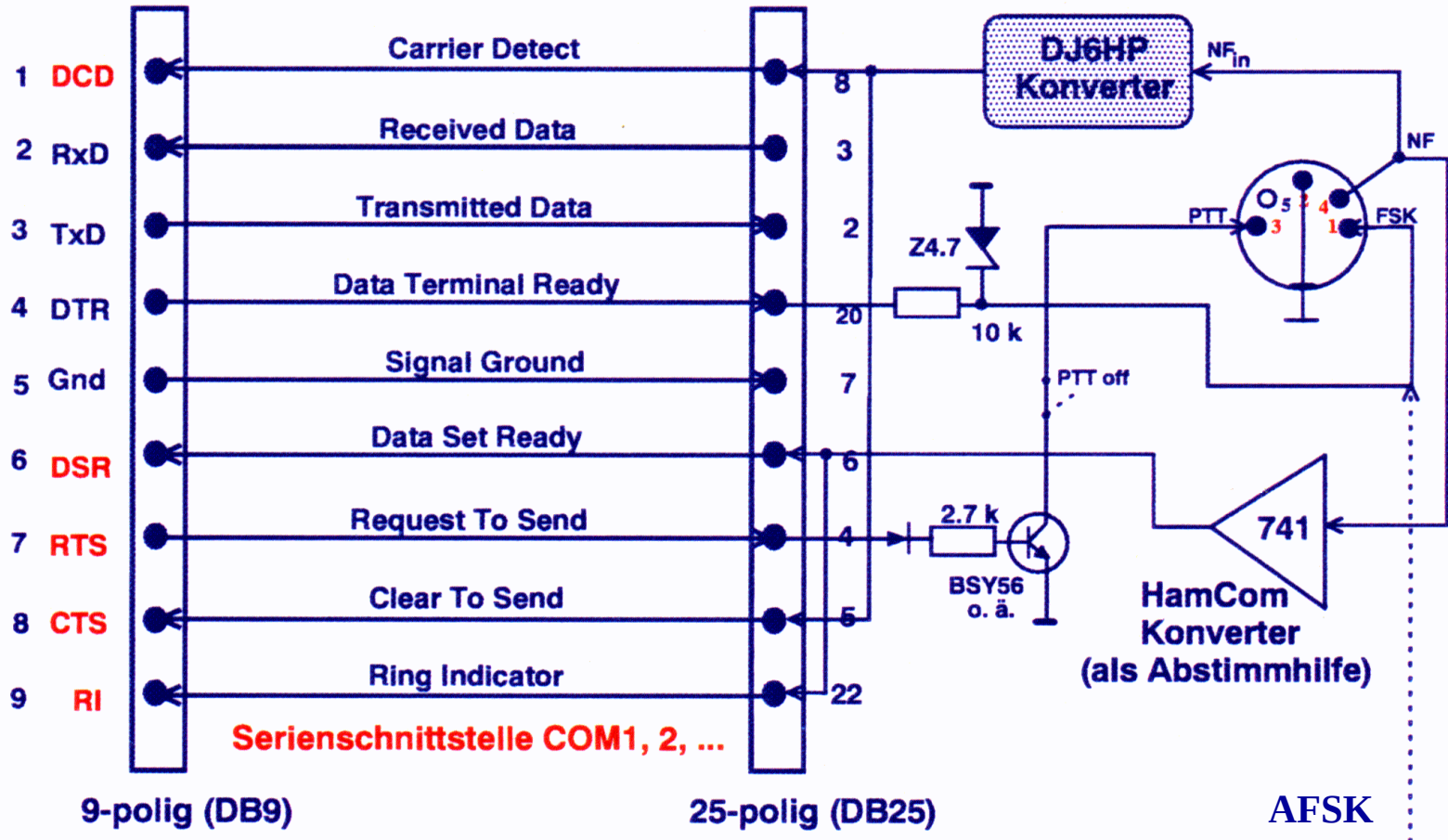


der Serienschnittstelle
des PC

Einfach-Konverter
(HamComm)
für gute Signale bei
RTTY, Amtor, Pactor
und Programme wie
HamComm,
BMKMULTI

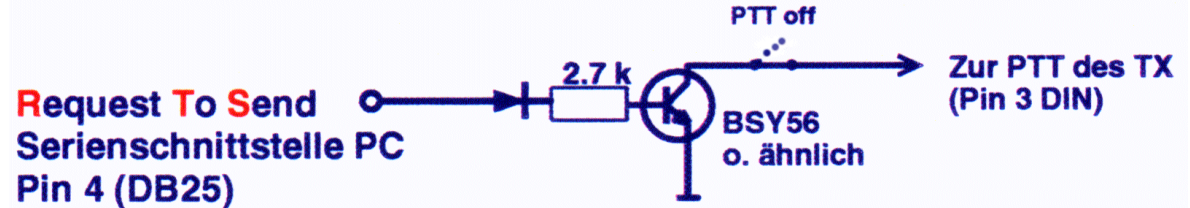
Anschluß der Konverter an den PC

- Serienschnittstelle COM1 oder COM2 (V24/RS232, 9-polig)

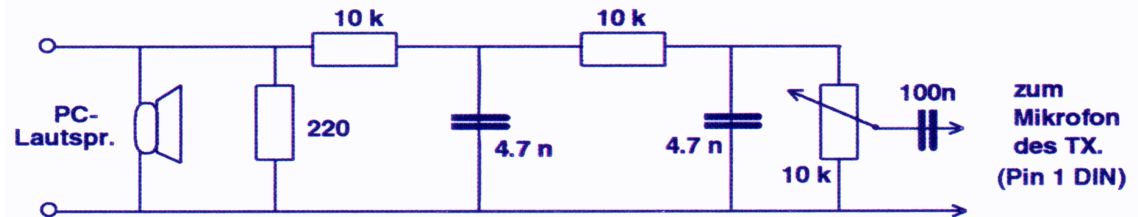


Und ein paar nützliche Schaltungen...

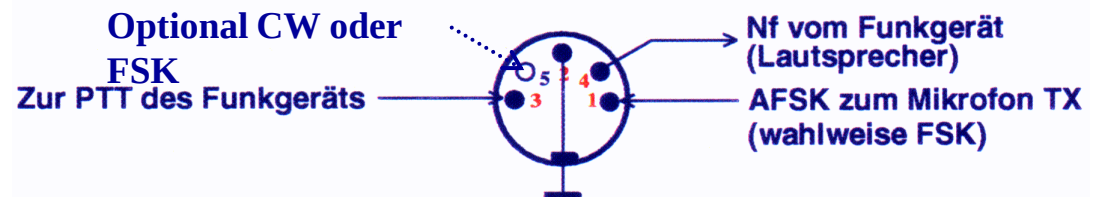
Einfache PTT-Schaltung



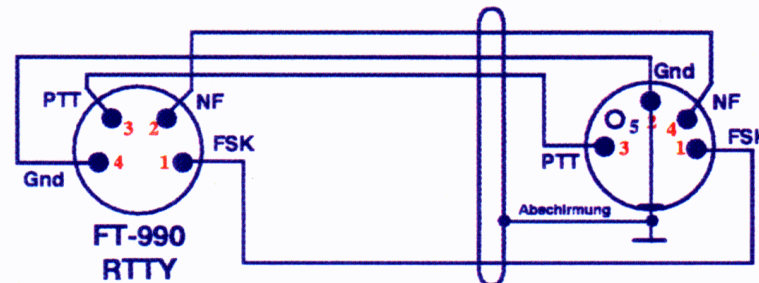
Einfach-AFSK (HamComm)
vom PC-Lautsprecher zum
Mikrofon



Empfohlener Standardanschluß
für alle Modems und TNCs
(DIN-Stecker 5 polig,
DJ9ZL-Norm ;-)



Und ein Anschlußbeispiel für
RTTY/Amtor/Pactor an den Yaesu
FT990 (FSK)



Digitale Betriebsarten

Überblick
und
Eigenschaften

Morsetelegraphie CW (Continuous Wave)

- Morse-Codierung, variable Zeichenlänge
- Codierung optimiert nach Sprachhäufigkeit der Buchstaben
- keine Redundanz (außer der Sprachredundanz)
- keine Fehlererkennung oder Fehlerkorrektur
- Übertragung asynchron und verbindungslos
- Sendermodulation:
 - **getasteter Träger (0/1 = Sender aus/ein), Sendeart A1A**
Sende/Empfangsfrequenz = Trägerfrequenz
 - **Einseitenband durch getastete Tonfrequenz (800 Hz), J2A**
Sendefrequenz = Trägerfrequenz +/- Tonfrequenz (USB/LSB)
 - **weitere Sendearten (selten): A2A, F1A, F2A, G1A, G2A**
- **Coherent CW:** Synchron, schmalbandig, hat sich wegen der sehr hohen Anforderungen an RX/TX-Stabilität vor 20 Jahren nicht durchgesetzt

CW  Prüfung

CW  QSO

Funkferschreiben **RTTY** (**R**adio **T**ele**T**ype)

- Codierung: 5 bit Telegraphenalphabet CCITT Nr. 2
- Patent 1874 durch Jean-Maurice-Emile **Baudot**
- Die 5 bit des Baudot-Codes ermöglichen
 - $2^5 = 32$ mögliche Zeichen. Das reicht nicht!
- Deshalb Umschaltung zwischen zwei Codierungsebenen **B**uchstaben und **Z**iffern (Bu/Zi), das ergibt
 - $2^5 \cdot 2 - 2 \cdot 2 = 60$ mögliche Zeichen.
- Davon sind **CR** (Carriage Return = Wagenrücklauf), **LF** (Line Feed = Zeilenvorschub) und **SP** (Space = Leerzeichen) in beiden Ebenen identisch, also kennt RTTY
 - $60 - 2 \cdot 3 = 54$ verschiedene, darstellbare Zeichen (deshalb werden auch nur Großbuchstaben dargestellt!)



weiter RTTY...

- Wegen der Ebenen-Abhängigkeit ist die Darstellung nicht eindeutig. Ein nicht richtig erkanntes Umschaltzeichen Bu oder Zi führt bis zum nächsten Umschaltzeichen zu Zeichensalat!
- Nationale Unterschiede Europa/USA (wen wunderts?)
 - **Klingel** / ' statt ' / **Klingel**
= / ; statt + / "
werda? Statt \$
- 100% Redundanz auf Bitebene (wenn kein **Mark**, dann **Space**!)
- Code optimiert auf minimale Anzahl von Stromschritten (zur Schonung der mechanischen Relais der alten Fernschreiber)
- Keine Fehlererkennung oder Fehlerkorrektur auf Zeichenebene
- Verbindungsloser, asynchroner (Start/Stop-) Betrieb

Und immer noch RTTY ...

- Die durch die IARU festgelegten Tonfrequenzen für Mark und Space und die gängigen Shifts sind:

Region 1 "Lowtones"	Shift	USA "Hightones"	Kurz  welle
Mark = 1275 Hz		2175 Hz	45.45  170
Space = 1445 Hz	170 Hz	2345 Hz	
= 1700 Hz	425 Hz	2600 Hz	100  850
= 2125 Hz	850 Hz	3025 Hz	

Achtung: Bei TX mit FSK-Eingang entsprechend umschalten, da die Töne intern im Transceiver generiert werden (FT990, IC706 etc.)!

- Die üblichen Baudraten / Shifts für RTTY sind:
 - 45.45 Baud / 170 Hz auf Kurzwelle (LSB) Amateurfunk
 - 50, 75 Baud / 425, 450 Hz Nachrichtenagenturen, Wetterstationen
 - 100 Baud / 425, 850 Hz Amateurfunk, Rundspruch auf UKW (FM)
- Frequenzen in MHz:
 - 3.580-3.620, 7.040-7.045, 14.070-14.099, 21.080-21.120, 28.050-28.150, 144.600, 144.800-144.990, 432.600, 433.300 (AFSK)

Funkferschreiben in ASCII

- Codierung: 7- oder erweiterter 8 bit **A**merican **S**tandard **C**ode for **I**nformation **I**nterschange (keine Parität)
- Alternativ 7 bit Zeichencode mit 1 Paritätsbit
- Eine Codierungsebene, eindeutig (keine Bu/Zi-Umschaltung)
- Auf Kurzwelle normalerweise **110 Baud** (ASR33 Fernschreiber)
- Verbindungsloser, asynchroner Verkehr wie bei RTTY
- Keine Fehlererkennung bei 8 Datenbits, bei 7 bit + Parität
Erkennung von Ein-bit-Fehlern, keine Fehlerkorrektur
- Im Amateurfunk sehr selten, da auf Kurzwelle wegen fehlender Fehlerkorrektur bei relativ hoher Bandbreite extrem störanfällig
- Frequenzbereiche wie bei RTTY

ASCII  ≤110/170

Amtor Mode B (Forward Error Correction FEC)

- Codierung: 7 bit Telegraphenalphabet CCITT Nr. 5, eine Codierungsebene, d.h. 128 mögliche Zeichen
- davon aber nur die Zeichen mit einem 3/4-Verhältnis von Mark und Space benutzt → 38 verschiedene Zeichen
- Übertragung verbindungslos, asynchron
- Fehlererkennung: Jedes Zeichen wird zweimal gesendet

FEC  CQ

Zeichen 1	Zeichen 2	angezeigt
ok	ok	print
ok	Fehler	print 1
Fehler	ok	print 2
Fehler	Fehler	no print
ok	ok, aber ungleich	no print

- Technik aus dem Seefunk, wird für CQ-Nachrichten (Wetter, Sturmwarnung, Bojen) und für Amtor CQ-Ruf benutzt

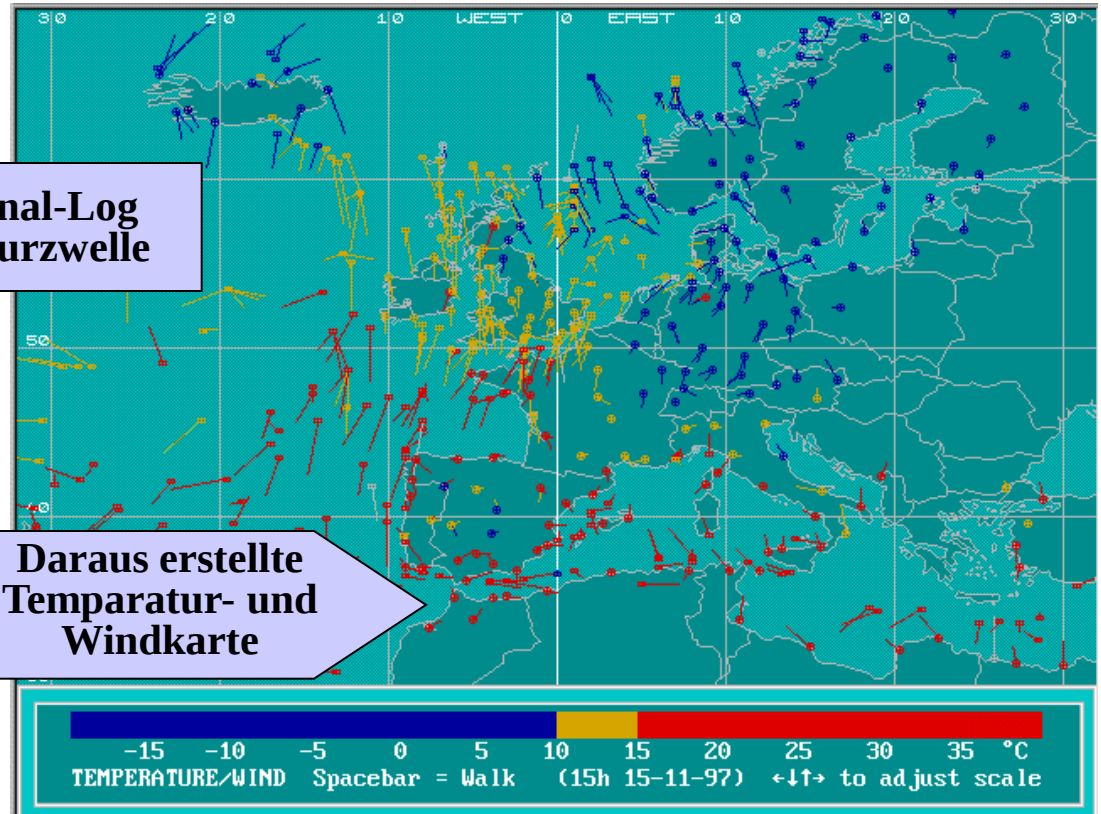
FEC im See- und Wetterfunk: NAVTEX

- Spezielles Nachrichtenformat (SINOP) für Navigation, Sturmwarnungen, Bojen, Eiswarnungen, Decca

clouds:2/8, cumulus, 540m
=
Portugal, 37°00'N 008°57'W
SAGRES (POINT)
manned
cloud height:1000-1500m
visibility:15km
cloud cover:7/8 or more
wind dir:calm, speed:0
air temp:+20.8
dew-point temp:+16.4
pressure at station level:1008.9hPa
pressure at sea level:1011.8hPa
pressure:steady
change in 3h:0.1hPa
maximum temp:+24.2
=

Original-Log
von Kurzswelle

Daraus erstellte
Temperatur- und
Windkarte



SELFEC (Selective FEC)

- Codierung und Technik wie FEC
- Im Gegensatz zu FEC nur nur **an einen Partner gerichtet**, aber dennoch verbindungslos
- Ausstrahlung des Empfänger-SELCALs (4-Zeichen-Rufzeichen, Beispiel DJZL) im 3/4-Verhältnis
- Datenverkehr mit umgekehrtem 4/3-Verhältnis, so daß nur noch der SELCAL-Empfänger mitschreiben kann
- Im Amateurfunk nicht üblich



Amtor Mode A (Automatic Repeat reQuest)

- Amateur TOR (Teletype Over Radio) Amtor  QSO
- Codierung CCITT Nr. 5 (7 bit), davon nur Zeichen im 3/4-Verhältnis von Mark/Space genutzt (also wie FEC)
- Verbindungsorientierte Übertragung zwischen je 2 Stationen. Weitere Stationen können (ohne Fehlerkorrektur) mithören.
- Fehlerkorrektur: Es werden immer Blöcke von je 3 Zeichen gesendet, die vom Empfänger bestätigt oder bei Fehler nochmals angefordert werden (Automatic Repeat ReQuest)
- **Signalisierungsrate 100 Baud** = 33 bit/sec oder $6^{2/3}$ Zeichen/sec
- Sendende Station schickt 210 msec Daten, dann wartet sie 240 msec auf ein „ACKnowledge“ → schnelle TX/RX-Umschaltung
- Kanalabstand 500 Hz wie bei RTTY und FEC



Das SELCAL bei Amtor

- Für den Aufbau einer verbindungsorientierten Kommunikation müssen die Partner über ein eindeutiges Kennzeichen verfügen. Hierzu dient das „SELCAL“
- Definition gemäß CCIR 625 Standard („TOR station identity“):
 - Zeichenkette aus **4 Zeichen**, oder 4 (auch 5)-stellige Dezimalzahl
 - Zeichenkette aus 7 Zeichen (ohne G,H,J,L,N,W), 9 Ziffern, Rufzeichen
 - Beispiel: **DJZL**, OXXKCZK, 899386743, DJ9ZL
 - Im Amateurfunk werden als SELCAL benutzt:
 - Erster und letzte drei Buchstaben: DO1KRM → DKRM
 - Zwei + zwei Buchstaben: DJ9ZL → DJZL
 - Erster Buchstabe doppelt + Rest: W2KL → WWKL
- CQ-Ruf: Senden in Mode B (FEC), nach Connect von der Gegenstation automatisches Umschalten in Mode A

Pactor (Packet-Amtor, der „Vermittler“)

- Halbduplex-Synchronverfahren mit fester Zeitstruktur
- **Verbindungsorientiert** zwischen je 2 Stationen, SELCAL=Call
- Signalisierungsrate **100/200 Baud** mit automatischer Anpassung je nach Übertragungsgüte, **Shift 200 Hz** (1200/1400 Hz AFSK)
- Codierung: 8-bit ASCII, daher auch **für Binärdatenübertragung** (Bilder, Programme) geeignet
- Datenkompression 1.7:1 (Huffman Code) für Textübertragung
- Empfang unabhängig von der Shiftlage durch ständigen Wechsel zwischen beiden Lagen „normal“ und „inverted“
- Sehr gute Fehlererkennung durch 16 bit (!) CRC (**C**yclic **R**edundancy **C**heck)

Weiter mit Pactor....

Pactor  connectPactor  QSO

- Pactor benutzt die sogenannte „Memory-ARQ“:
 - 8 bit AD-Konverter zur 0/1-Erkennung (d.h. 256 Stufen Auflösung)
 - Aufsummierung und Mittelung der Bit-Signale fehlerhafter Frames bis zur korrekten Rekonstruktion gemäß CRC
 - Ausmittlung der Interferenzsignale durch den ständigen Shift-Wechsel
- UNPROTO-Mode (Pactor-FEC) für Rundspruch und CQ-Ruf
- LISTEN-Mode zum Mitschreiben ohne Fehlerkorrektur
- Kontrolliertes Link-Down mit Handshake
- Forcierte Sende/Empfangsumschaltung durch Empfänger
- Geringe Anforderungen an Geräte-Schaltzeiten
- **Longpath-Option** (1.4 sec Zyklus) für Fernverbindungen

Clover

- Codierung variabel: 17....255 Bytes Reed-Solomon Code
- Synchrone Übertragung
- Fehlerkorrektur mit ARQ (Automatic Repeat Request)
- Kombinierte Phasen- und Amplitudenmodulation
- 10 verschiedene, automatisch adaptierende Modulationsarten mit Bitraten von 750 bit/sec bis 31.25 bit/sec (32 msec pro Zeichen)
- Kanalabstand 500 Hz
- Betriebsarten ARQ (Fehlerkorrektur) und FEC (CQ-Ruf)
- Hat sich wegen der Komplexität der Protokolle und der Implementierung in (und deshalb teurer) Spezialhardware nicht durchgesetzt und wurde im wesentlichen durch Pactor verdrängt

Schön  wär's...



PSK31 (*Phase Shift Keying 31*)

- Das „neue“ RTTY mit ähnlichen Betriebseigenschaften
 - Codierung: Varicode mit variabler Wortlänge
 - Jedes Zeichen startet/endet mit „1“, enthält keine Doppelnull, wird aber vom nächsten Zeichen durch Doppelnull „00“ getrennt
 - Code optimiert auf englischsprachige Zeichenhäufigkeit (ähnlich wie beim Morse-Code)
- [Varicode](#)  [Tabelle](#)
- „0“ und „1“ werden nicht als Tonfrequenzen, sondern als Sequenzen festgelegter Phasenwechsel des Sendesignals mit anschließender Glättung durch ein Cosinusfilter erzeugt
 - Signalisierungsrate **31.25 Baud**, im Mittel **50 Zeichen/sec**
 - Damit ergibt sich effektiv ein Zweitonsignal mit **± 15 Hz** Abweichung vom Trägersignal (meist 1000 Hz USB)

Und noch PSK31....

- Ergebnis ist eine **extrem schmalbandige 31Hz-Technik**, ideal für die Anwendung von DSPs (**D**igital **S**ignal **P**rocessor), wie sie z.B. auf modernen Soundkarten zu finden sind
- Verbindungsloser, asynchroner (Start/Stop-) Betrieb
- **2 Modes**: BPSK (Standard) und QPSK (seltener)
- **Binary PSK**: Keine Fehlererkennung oder Fehlerkorrektur
- **Quaternary PSK**: Hinzunahme von $90^\circ + 270^\circ$ Phasensprüngen. Das ergibt zwei unabhängige Sendekanäle (doppelte Datenrate), die bei gleicher Übertragungsrate für eine **Fehlerkorrektur** (convolutional Encoder und Viterbi-Decoder) verwendet werden können. Die Decodierung bedingt eine kleine Verzögerung von 20 bit = 640 msec

PSK31  QSO



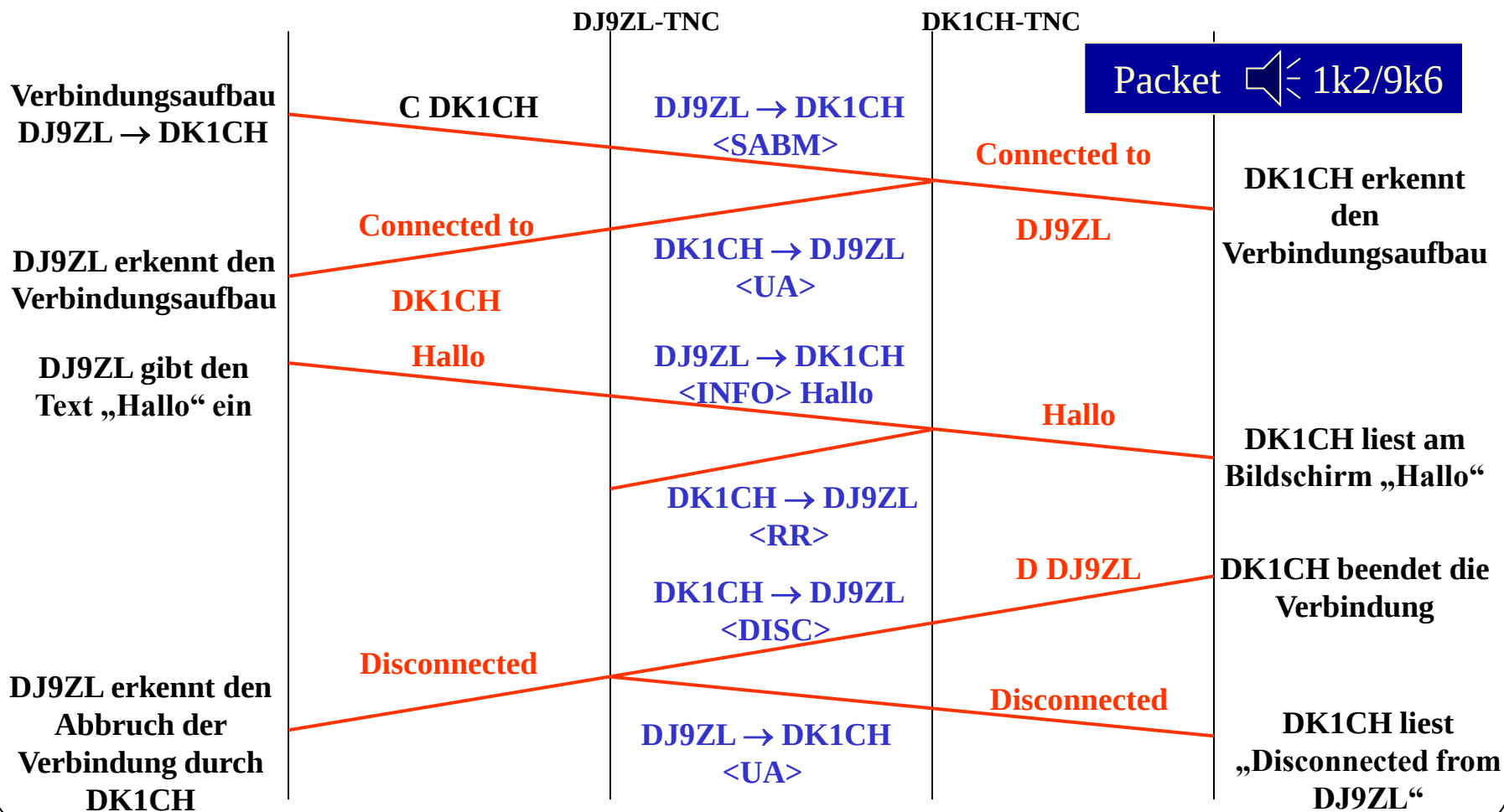
Packet-Radio (Amateur X.25) Standards (brrrr...;-)

- AX.25 „Link Layer Protocol Specification“, abgeleitet vom professionellen Standard X.25 (heute bei „Profis“ ausgestorben)
- Konform zu ISO 3309, 4335 (inklusive DAD 1&2) und 6256 High-level Data Link Control (HDLC)
- Konform zur Beschreibung von ADCCP „balanced mode“ gemäß Spezifikation ANSI X3.66 (LAPB, kein „Master“)
- Entsprechend der CCITT X.25 Empfehlung mit Ausnahme eines erweiterten Adreßfeldes und der Hinzunahme von „Unnumbered Information (UI)“-Frames
- Folgend der CCITT-Empfehlung Q.921 (LAPD) zur Realisierung mehrfacher, Links auf einem einzelnen „shared channel“: „Link Access Protocol Amateur Packet-Radio LAPA“
- Ergänzung des CSMA-Zugriffsverfahren durch Master/Slave Verfahren DAMA („Demand Assigned Multiple Access“)

Eigenschaften von AX.25 (Packet-Radio)

- Codierung 8 bit ASCII (7, 8 bit Datenword, mit/ohne Parität)
 - Verbindungsorientierter (**Connected**) Mode mit Fehlerkorrektur
 - Verbindungsloser (**Unproto**) Mode für CQ und Rundspruch
 - Fehlererkennung mit 16 bit **Frame Check Sequence (FCS)**
 - 7 Byte Adreßfelder mit 4 bit (0 bis 15) „**Station Sub-ID (SSID)**“ für bis zu 16 parallele Verbindungen pro Adresse (Rufzeichen)
 - Signalisierungsrate 300 auf Kurzwelle, 1200, 9600 bis 56000 Baud auf UKW
- Packet 300 📡 Kurzwelle
- Störanfällig gegen Verzerrungen wie selektives Fading, Interferenz oder Signalverformung in Sender und Empfänger
 - Auf Kurzwelle ohne DAMA schwieriger Betrieb durch „**Carrier Sense Multiple Access/Carrier Detect**“-Verfahren (QRM, QSB, Unhörbarkeit der sendenden Stationen je nach Entfernung)

Ablauf einer Verbindung (Connect) bei AX.25


Packet  1k2/9k6

AX.25 Protokollrahmen

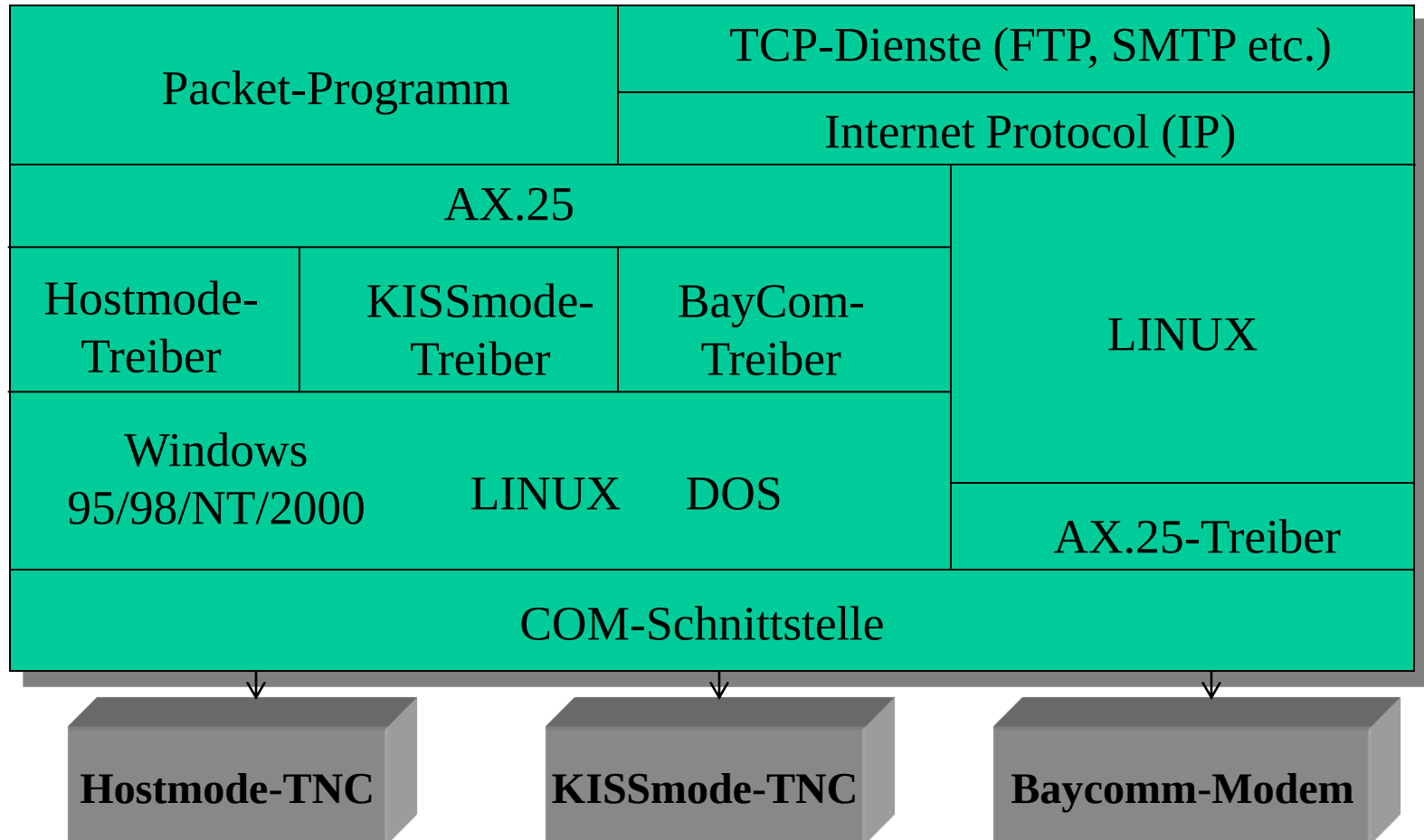
- Bezeichnung der Steuer- und Datenrahmen (Frames):
 - **C/SABM** Verbindungsaufbau (Set Asynchronous Balanced Mode)
 - **UA** Bestätigung ohne Verbindung (Unnumbered Acknowledge)
 - **UI** Informationsdaten ohne bestehende Verbindung (Unnumbered Information, Unproto)
 - **I/INFO** Daten der eigentlichen Nachricht (Information)
 - **RR** Bereit für weiteren Text (Receive Ready)
 - **D/DISC** Verbindung soll abgebaut werden (Disconnect)
 - **FRMR** Zurückweisung eines Frames (Frame Reject, fatal error)
 - **RJ/REJ** Zurückweisung eines Frames (Reject Frame, out of sequence or duplicate)
 - **RNR/NR** Empfangsbuffer voll (Receive Not Ready)
 - **DM** Verbindungsabbau (Disconnect Mode, system busy)

Weitere Eigenschaften AX.25...

- Das Packet-Programm im Computer spricht zum **TNC** (Terminal Node Controller) mit unterschiedlichen Protokollen:
 - **Terminalmode:** Zur Steuerung des TNC und zur Datenübertragung genügt ein einfaches Terminal (VT100) oder eine Terminalemulation auf dem PC (Telix, Windows Hyperterminal)
 - **Hostmode:** Es wird ein Packetprogramm auf dem PC benötigt, das den jeweiligen Hostmode unterstützt. Gängige Hostmode-Firmwares sind:
 - **TheFirmware (TF), 6pack, WA8DED, TAPR**
 - **Passende Packet-Programme: GP, SP, WInGT, TOP, Paxon**
 - **KISSmode** („Keep It Simple Stupid“): Sehr einfaches Protokoll, AX.25-Protokoll wird im wesentlichen im PC-Programm abgewickelt
 - **Packet-Programme: Pr4Win32, KA9Q-NOS, Wnos4 (Wampes-NOS)**
 - **BayCom-Mode:** Keine „Intelligenz“ in Hardware, alle AX.25-Protokollebenen werden vollständig in der Software des PCs abgewickelt (derzeit bis 600 kbps mit schneller Hardware !)
- Binärübertragung im Hostmode oder mit Hilfsprogrammen:
 - **#BIN#, YAPP, 7PLUS, UUENCODE**

AX.25 und TCP/IP im Betriebssystem

PC-Software



TCP / IP (die Internet-Protokolle)

- Transmission Control Program / Internet Protocol (mit Internet Control Message Protocol ICMP)
- Transport der IP-Protokolle über
 - NET/ROM, AX.25, Ethernet, SLIP, PPP
- Jeder „Knoten“ (Rechner) im Internet hat
 - eine eindeutige IP-Adresse (z.B. 44.130.20.55)
 - einen zugehörigen Namen (z.B. dj9zl) und
 - eine zugehörige IP-Domäne (z.B. ampr.org)
- Internet ist ein „Netz der Netze“. Jeder Knoten kennt nur die Rechner im eigenen Netz und ein „Gateway“ zu allen anderen Netzen
- Routing im Internet erfolgt auf IP-Ebene (aber nicht im Amateurfunk, da nicht organisiert !!!)

dj9zl.ampr.org

Webbrowser
(IP-Anwendung)

**TCP / UDP-
Protokoll**

**IP-
Protokoll**

ICMP-Protokoll

**Link-Layer
Protokoll
(z.B. AX.25)**

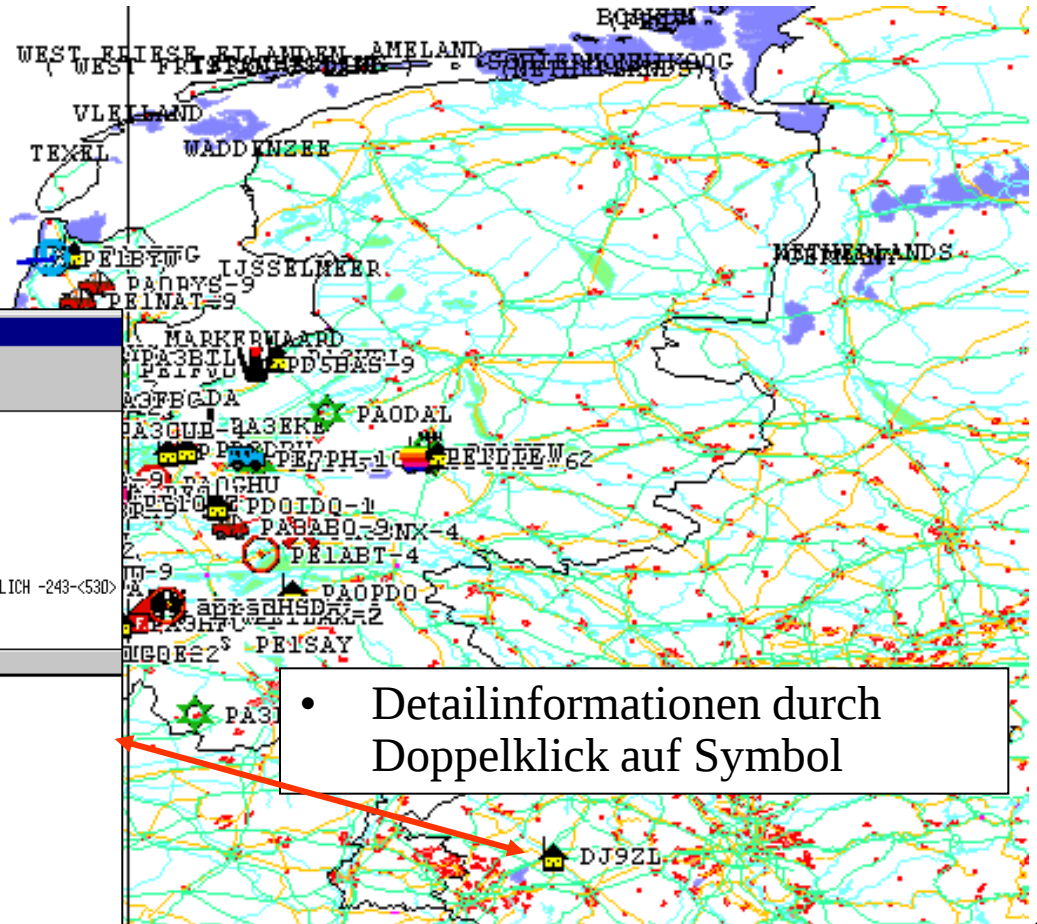
**Physical Medium
Access**

Gängige IP-Dienste (Protokolle)

- TELNET Terminalverbindung zu einem anderen IP-Rechner
- FTP File Transfer Protocol für BIN/ASCII-Dateitransfer mit und ohne Benutzervalidierung (anonymous FTP)
- BIND zur Namens ↔ Adreßübersetzung (DNS)
- SMTP Simple Mail Transfer Protocol (E-Mail)
- POP Post Office Protocol (Mailserver)
- FINGER Information über dem Server bekannte Benutzer
- CHAT Convers-Mode für Gesprächsrunden (n:n)
- NNTP Network News Transfer Protocol, elektronisches Konferenzsystem „NetNews“
- ECHO IP Echo-Test (nicht zu verwechseln: ICMP-„Ping“ !)
- HTML HyperText Markup Language (Web-Protokoll)

APRS (Automatic Position Reporting System)

- Netzwerk GPS-basierender Systeme zur automatischen Positionsverfolgung über Packet und Internet (aprs.net)
- Darstellung über ladbare Maps oder Listen mit Symbolen
- Automatisches Update in Echtzeit
- Station-zu-Station-Messaging





DJ9ZL

Power 0 Watts

Height 0 Feet

Gain 0 db

Direction 0

DJ9ZL

Packets Rcvd 23

3/7/2000 21:37:46

3/7/2000 22:09:01

Lat 50 55' 21"N

Lon 6 21' 57"E

Messages Sent 1

=5055.35N/00621.96E-PHG0000/HinAPRS 2.4.3 -GENRJUELICH -243-<53D>

History

>xmit To TCP>=5055.35N/00621.96E-PHG0000/HinAPRS 2.4.3 -GENRJUELICH -243-<53D>J

>xmit To TCP>:001AAA :hallo,{000J

DJ9ZL>AP4243,TCPXX*:5055.35N/00621.96E-PHG0000/HinAPRS 2.4.3 -GENRJUELICH -243-<53D>

DJ9ZL>AP4243,TCPXX*:001AAA :hallo,{000J

>xmit To TCP>=5055.35N/00621.96E-PHG0000/HinAPRS 2.4.3 -GENRJUELICH -243-<53D>J

DJ9ZL>AP4243,TCPXX*:5055.35N/00621.96E-PHG0000/HinAPRS 2.4.3 -GENRJUELICH -243-<53D>

>xmit To TCP>=5055.35N/00621.96E-PHG0000/HinAPRS 2.4.3 -GENRJUELICH -243-<53D>J

DJ9ZL>AP4243,TCPXX*:5055.35N/00621.96E-PHG0000/HinAPRS 2.4.3 -GENRJUELICH -243-<53D>

>xmit To TCP>=5055.35N/00621.96E-PHG0000/HinAPRS 2.4.3 -GENRJUELICH -243-<53D>J

DJ9ZL>AP4243,TCPXX*:5055.35N/00621.96E-PHG0000/HinAPRS 2.4.3 -GENRJUELICH -243-<53D>

• Detailinformationen durch Doppelklick auf Symbol



TCP / IP: Realität und Ausblick...

- Da keine organisierte IP-Netzstruktur existiert, ist IP-Routing im allgemeinen nicht vernünftig möglich
- Deshalb wird das „Routing“ meist auf Link-Layer (AX.25) abgewickelt und z.B. durch WAMPES weitgehend automatisiert
- IP wird häufig nur benutzt, um (entgegen den strengen Richtlinien für Amateur„*funk*“) über Gateways vom AX.25-Packetnetz ins drahtgebundene Internet zu gelangen
- Interessant wird IP vor allem dort, wo mit hohen Datenraten (56k oder mehr) auch multimediafähige IP-Anwendungen wie MIME-Mail (**M**ultipurpose **I**nternet **M**ail **E**xtension) oder HTML (Web-Protokoll) eingesetzt werden können
- Ein funktionierendes IP-Netz (per Funk !) mit hohen Datenraten (AX.25-Übertragung >57k) wäre sehr attraktiv (**HAM-Web**)!!!