

# 'WEAK SIGNAL' - Kommunikation

Von **CW** über **JT65** und **WSPR**  
bis **FT8**

10.11.2018 (HB9AUR)

SWISS-ARTG GV 2018

1

Weak Signal Kommunikation:  
Sehr aktuell im Sonnenflecken-Minimum

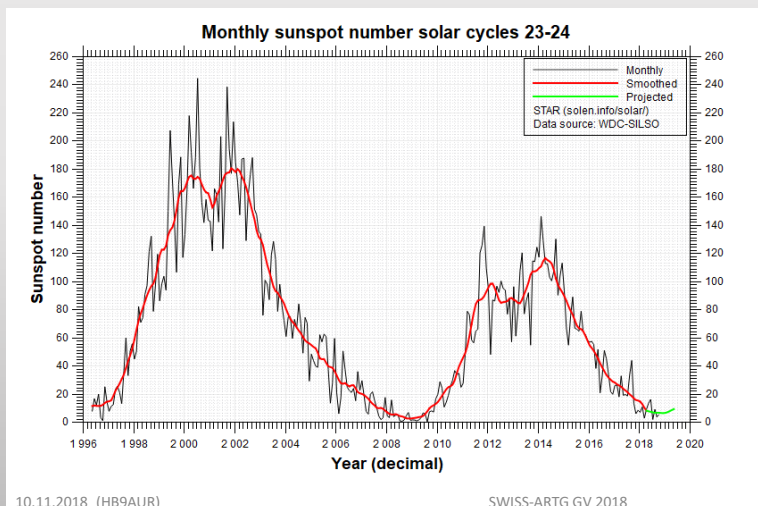


Tabelle: [www.solen.info](http://www.solen.info)  
(by Jan Alvestad)

10.11.2018 (HB9AUR)

SWISS-ARTG GV 2018

2

- Trotz schlechten Bedingungen: tolle DX dank neuen Weak Signal Verfahren, ermöglicht durch cleverere **digitale Signalverarbeitung**
- Besonders bekannt und beliebt: **FT8**
- **WSPR**: schmalbandige Baken (6 Hz) mit geringer Leistung (0.1 – 5 W) geben ein Abbild der aktuellen Ausbreitungsbedingungen auf Kurzwelle
- Verfahren für **EME** machen die riesigen Parabolschüsseln obsolet
- Andere Verfahren erlauben **Meteor-Scatter** QSOs fast rund um die Uhr
- Ursprünglich für **EME** ausgeheckt, wird **WS**-Kommunikation nun verbreitet auch für **ionosphärische Verbindungen** eingesetzt

## Schwerpunkte meiner Präsentation

Einige der 'langsamen' Weak Signal Verfahren:

**JT65**

**WSPR**

**JT9**

**FT8**

## Wer hat's erfunden?

**Joe Taylor / K1JT** (geb. 1941)  
Astrophysiker, Nobelpreisträger (1993)



## Wo kriege ich die SW?

- Nach **WSJT-X** suchen (beinhaltet FT8, JT4, JT9, JT65, QRA64, ISCAT, MSK144, WSPR)
- oder nach **JT65-HF-HB9HQX**-Edition (von Beat Oehrli)

## CW:

### Der Old Timer unter den 'Weak Signal' Verfahren

- 'Wenn sonst nichts geht, CW funktioniert meistens'
- Wie verbessert man den Empfang schwacher CW-Signale?  
Man wählt ein möglichst schmales Filter.  
=> Schmales Filter heisst:  
weniger Rauschen und Störungen -> besseres S/N  
(funktioniert nur, weil das **Nutzsignal** auch **schmalbandig** ist!)
- Wenn's noch immer nicht reicht? Man wiederholt die Worte (nicht die einzelnen Buchstaben!)  
= **Redundante** Übermittlung, mit **zeitlichem Versatz** der repetierten Information

## CW: Noch ein 'Trick'

- Bei schwachen Signalen langsam tasten -> **Symbolrate reduzieren**
- Generell: Information möglichst optimal in wenige Symbole packen
- Optimierte Zeichen-Codierung in **CW**: Häufig vorkommende Zeichen (e, i, t, s, a, n) werden mit wenigen Symbolen codiert. Für seltenere Zeichen (f, l, c, q, y, p) leistet man sich längere Symbolketten.

**-> optimale 'Quellcodierung'**

(Wie packe ich möglichst viel Info in möglichst wenig Bits?)

## Kanalcodierung

- **CW**: Ein/Aus-Codierung -> digitale Amplitudenmodulation
  - gut geeignet für die gehörmässige Dekodierung
  - einzige technische Möglichkeit zu Zeiten der Einführung
  - schlecht geeignet für automatische Dekodierung
- **Alternativen**:
  - Frequenzumtastung **FSK** (früh schon verwendet für RTTY, aber noch nicht schmalbandig);  
davon abgeleitet: **MFSK**
  - Phasenumtastung **PSK** (erstmal im Afu verbreitet mit PSK31)

## PSK oder MFSK ?

- PSK31, von Peter Martinez/G3PLX gegen Ende der 90er Jahre im Afu eingeführt, läutete das Zeitalter der PC-basierten Übertragungsarten ein.
- Fast zeitgleich erschien MFSK16 von ZL1BPU und IZ8BLY.
- Inzwischen sind viele weitere Verfahren eingeführt worden, jeweils optimiert auf spezifische Anwendungen. Die meisten verwenden **MFSK**.
- Urväter von MFSK waren Piccolo (32FSK, 20 Bd) und Coquelet (8FSK, 13FSK, 13.33 Bd)

## Vor- und Nachteile von MFSK

- Hervorragende Unterdrückung von Puls- und Breitband-Noise dank schmaler Bandbreite für den einzelnen Ton
- Niedrige Baudrate ermöglicht hohe Empfindlichkeit und gute Mehrweg-Unterdrückung
- Unempfindlich auf ionosphärische Effekte wie Doppler, Fading, Multi-Path
- Konstante Sendeamplitude -> keine lineare Endstufe notwendig
- **Benötigt hohe Frequenzstabilität**
- **Benötigt grössere Bandbreite (Hz/bps) als PSK**

## Ein näherer Blick auf MFSK

- Es wird immer EINER von M möglichen Tönen gesendet
- Ein Ton entspricht einem Symbol
- Je grösser M, desto grösser der Informationsgehalt eines Symbols / Tons:

M		Bits / Ton	Symbole
2	FSK	1	0, 1
4	4FSK	2	00, 01, 10, 11 («dibits»)
8	8FSK	3	000, 001, 010, 011, 100, 101, 110, 111
16	16FSK	4	0000, 0001, 0010, 0011, 0100, ... , 1101, 1110, 1111
32	32FSK	5	00000, 00001, 00010, 00011, 00100, ... , 11110, 11111
64	64FSK	6	000000, 000001, 000010, 000011, ... , 111110, 111111

## M: Symbolrate oder Bandbreite optimieren

- Bei gegebenem S/N kann ich einen bestimmten Ton umso zuverlässiger detektieren, je länger er anhält -> Je niedriger die Symbolrate, desto geringer die Fehlerwahrscheinlichkeit
- Bei einer bestimmten Anzahl Töne und halbiertes Symbolrate halbiert sich auch die Bitrate (resp. der Durchsatz). Durch Verdoppelung der Anzahl Töne verdoppelt sich jedoch die Anzahl bits/Ton, und der Durchsatz bleibt gleich.

- Die benötigte Bandbreite wird jedoch grösser!

M		Bits/Ton	Symbolrate [Bd]	Durchsatz [bits/s]
2	FSK	1	10	10
4	4FSK	2	5	10
8	8FSK	3	3.3	10
64	64FSK	6	1.67	10

## Neuere (langsame) Weak-Signal Verfahren von K1JT

- **JT65** (2003, QST June 2005): EME, HF QRP; 65FSK, 2.692 Bd, 48 s
- **JT4** (2007): EME VHF; 4FSK, 4.375 Bd, 48 s
- **WSPR** (2008, QST November 2010): Beacon; 4FSK, 1.46 Bd, 96 s
- **JT9** (2012): HF QRP; 9FSK, 1.736 Bd, 48 s
- **QRA64** (2016): EME, 64FSK, 1.736 Bd, 48 s
- **FT8** (QST October, November 2017): HF QRP; 8FSK, 6.25 Bd, 13 s

## Optimierte Quellcodierung

- Ziel: Nur minimale Anzahl bits übertragen für ein minimales, doch vollständiges QSO
- Minimales Standard-QSO:
  1. **CQ** **HB9ABC** **JN47**
  2. **HB9ABC** **AB3CDE** **IO91**
  3. **AB3CDE** **HB9ABC** **-8**
  4. **HB9ABC** **AB3CDE** **R-17**
  5. **AB3CDE** **HB9ABC** **RRR**
  6. **HB9ABC** **AB3CDE** **73**
- Die Standardmeldungen werden fix mit 72 bit codiert!

## Codierung von Rufzeichen

- Einfache 6-bit Codierung für 6-stelliges Rfz:  $6 \times 6$  bit = **36 bit**
- Für jede Stelle optimierte Codierung: **31 bit**

	1. Stelle	2. Stelle	3. Stelle	4. Stelle	5. Stelle	6. Stelle
Mögliche Zeichen	B, Z, Leer	B, Z	Z	B, Leer	B, Leer	B, Leer
Anzahl mögl. Zeichen	37	36	10	27	27	27
Benötigte bits	6	6	4	5	5	5

- Optimierung über alle 6 Stellen: Es gibt maximal  $37 \times 36 \times 10 \times 27 \times 27 \times 27 = 262'177'560$  mögliche Rufzeichen. Weil mit 28 bit  $2^{28}$ , also über 268 Mio verschiedene solcher Kombinationen dargestellt werden können, müssen **28 bit** reichen! (Nochmals 3 bit gespart!)

## Codierung des Locators

- Einfache 6-bit Codierung für 4-stelligen Locator:  $4 \times 6$  bit = **24 bit**
- Für jede Stelle optimierte Codierung: **18 bit**

	1. Stelle	2. Stelle	3. Stelle	4. Stelle
Mögliche Zeichen	B	B	Z	Z
Anzahl mögl. Zeichen	27	27	10	10
Benötigte bits	5	5	4	4

- Optimierung über alle 4 Stellen: Es gibt maximal  $180 \times 180 = 32'400$  mögliche Locators. Weil  $2^{15} = 32'768$  dazu ausreichen: **15 bit!**



## Codierung der Message

- Type 1 Message: **HB9ABC AB3CDE IO91**  
 $28 + 28 + 15 + 1 = 72 \text{ bit}$
- Es werden immer  $n = 72$  bits in einem Block übertragen!
- Varianten der Type 1 Message:  
**CQ HB9ABC JN47**                      **QRZ HB9ABC JN47**  
**AB3CDE HB9ABC -8**                      **HB9ABC AB3CDE 73**
- Beliebige Message im 72 bit Block: 13 Zeichen aus einem Alphabet von 43 möglichen Zeichen (A-Z, 0-9, ' . , / # ? \$ \_ ')

## FEC Codierung für JT65

- **306** zusätzliche bits für die Fehlerkorrektur (FEC), Total **378 bit**  
 -> FEC Codierrate  $r = 72/378 = 0.19$
- Anders ausgedrückt: Die Redundanz beträgt  $378/72 = 5.25$
- Simpelste Redundanz wäre: 5 x die Meldung wiederholen
- Mathematisch optimierte Redundanz: 'Reed Solomon' Code RS(63,12), macht aus 72 bits deren **378**

## Kanalcodierung für JT65

- Die 378 bits werden zu 63 Symbolen à 6 bit zusammengefasst
- Es gibt 64 mögliche Symbole. Jedes wird einem von 64 Tönen zugeordnet

## Synchronisation in JT65

- Ohne genaue Kenntnis des Anfangs der Übermittlung und der Zeichenwechsel ist keine Dekodierung des Signals möglich  
-> eine **Synchronisation** ist notwendig!
- 'Ungefähre Synchronisation': Start eines Blockes jeweils zur vollen Minute + 1 s (hh:mm:01)
- 'Präzise Synchronisation' mittels 'Synchronisations bits', welche auf einer separaten Synchronisationsfrequenz übertragen werden
- Während der Aussendung eines Blocks werden nebst den 63 Daten-Symbolen 63 Synchronisationssymbole übertragen.
- **Symboldauer: 0.372 s (2.69 Bd); Blockdauer: 126 x 0.372 s = 46.872 s**

## Synchronisationsvektor

Der Synchronisationsvektor definiert, zu welcher Zeit ein Datensymbol oder der Synchronisations-Ton gesendet wird. Er hat also eine Länge von 126 bit, besteht aus 63 '0' und 63 '1' und sieht so aus:

```
1,0,0,1,1,0,0,0,1,1,1,1,1,1,0,1,0,1,0,0,0,1,0,1,1,0,0,1,0,0,
0,1,1,1,0,0,1,1,1,1,0,1,1,0,1,1,1,1,0,0,0,1,1,0,1,0,1,0,1,1,
0,0,1,1,0,1,0,1,0,1,0,0,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,
0,1,0,0,1,0,1,1,0,1,0,1,0,0,1,1,0,0,1,0,0,1,0,0,0,0,1,1,
1,1,1,1,1,1
```

## Demodulation

- Nach der Digitalisierung des empfangenen NF-Signal wird als erstes mittels diskreter Fourier-Transformation (DFT) nach der Synch-Frequenz gesucht, um dann die zeitliche Position des Synch-Musters zu bestimmen. Die Genauigkeit liegt dabei typischerweise bei 1.5 Hz und 0.03 s.
- Als Nächstes erfolgt die Bestimmung der Frequenzdrift
- Nun wird für jedes der 63 Symbole mittels DFT der spektrale Anteil der einzelnen Töne bestimmt. Die Wahrscheinlichkeit jedes Tones wird an den 'Soft'-Decoder weitergegeben (also für jedes der 63 Symbole 64 Wahrscheinlichkeiten)

## Reed-Solomon Decoder

- Die Umkehrung der Reed-Solomon Codierung ist rechnerisch extrem aufwändig. Es existieren jedoch Algorithmen, welche auf recht effiziente Weise der optimalen Decodierung nahe kommen.
- Ergibt die Dekodierung bei ganz schwachen Signalen kein plausibles Resultat, startet ein 'Deep-search': Mit einer Liste möglicher Rufzeichen werden z. Bsp. CQ Rufe codiert und auf Übereinstimmung mit dem empfangenen Signal überprüft.

## JT65 Specs

Mode	FEC	Data, Tot. [bit]	MFSK	Symb./s [Bd]	BW [Hz]	Sync En. [%]	Tx Dauer [s]	min. S/N [dB]
JT65	Reed Solomon	72,378	65FSK	2.692	177.6	50	46.8	-25

## WSPR:

### Weak Signal Propagation Reporter (Bake)

- Text: **HB9ABC JN47 37**  
 $28 + 15 + 7 = 50 \text{ bit}$
- FEC: 'Convolutional Code' (K=32, Rate =  $\frac{1}{2}$ )  
 -> Anzahl bits im **Block**:  $(50 + K-1) \times 2 = 162 \text{ bit}$
- Synchronisation: **162 bit**
- Modulation: **4FSK, 1.4648 Bd** -> Symboldauer 0.683 s  
**Blockdauer**:  $162 \times 0.683 \text{ s} = 110.6 \text{ s}$  -> Startzeit hh:00:01, hh:02:01
- 2 bit pro Symbol: 1 bit Data, 1 bit Sync

## WSPR Töne

	Ton 0	Ton 1	Ton 2	Ton 3
<b>Symbol</b>	00	01	10	11
<b>Data</b>	0	0	1	1
<b>Sync</b>	0	1	0	1



## JT9 – ähnlich JT65, aber schmalbandiger

- Nutzdaten im Block: **72 bit** (JT65: 72)
- FEC mittels 'Convolutional Code' (K=32, r=1/2) (JT65: RS(63,12))
- **Blocklänge** total:  $(72+32-1) \times 2 =$  **206 bit** (JT65: 378)
- Modulation: **9FSK, 1.736** (2.69) **Bd** -> Symboldauer 0.576 s; (0.372 s)
- Mit 3 bit pro Symbol enthält ein Block  $206 / 3 =$  **69** (63) 'Nutz'-Symbole
- Zusätzlich werden **16** (63) 'Synch'-Symbole übertragen
- **Blockdauer:**  $(69+16) \times 0.576$  s = **48.96 s** (46.872 s)
- **Bandbreite:** **15.6 Hz** (177.6 Hz)
- **S/N Grenzwert:** **-27 dB** (-25 dB)

10.11.2018 (HB9AUR)

SWISS-ARTG GV 2018

30

## FT8: Der Renner!

- Warum ist **FT8** so viel erfolgreicher als **JT65** oder **JT9**?
  - > Die QSOs sind 4x schneller abgewickelt: 15 s pro Durchgang anstatt **1 min**! Das Standard-QSO (6 Durchgänge) benötigt damit nur 1.5 min anstatt **6 min**.
  - > etwas weniger empfindlich als **JT65**: Detektionsniveau -21 dB statt **-25 dB**
  - > QSO weitgehend automatisch abgewickelt

10.11.2018 (HB9AUR)

SWISS-ARTG GV 2018

31

## FT8: Die Technik und Methoden

- Nutzdaten im Block: **75 bit** (JT9: 72 bit)
- **Prüfsumme** (CRC): 12 bit (JT9: keine)
- FEC: LDPC ('low density parity check'): **87 bit**
- Blocklänge total: **174 bit** (JT9: 206)
- Modulation: **8FSK, 6.25** (1.736) Bd -> Symboldauer 0.16 s (0.576 s)
- Mit 3 bit/Symbol enthält ein Block  $174/3 = 58$  (69) 'Nutz'-Symbole
- Zusätzlich werden  $3 \times 7 = 21$  (16) 'Synch'-Symbole übertragen
- **Blockdauer**:  $(58+21) \times 0.16 \text{ s} = 12.64 \text{ s}$  (JT9: 48.96 s)
- **Bandbreite**: **50 Hz** (JT9: 15.6 Hz)
- **S/N Grenzwert**: **-21 dB** (JT9: -27 dB)

10.11.2018 (HB9AUR)

SWISS-ARTG GV 2018

32

## Zusammenfassende Übersicht

Mode	FEC	Data, Tot. [bit]	MFSK	Symb./s [Bd]	BW [Hz]	Sync En. [%]	Tx Dauer [s]	min. S/N [dB]
JT65	Reed Solomon	72, 378	65FSK	2.692	177.6	50	46.8	-25
WSPR	K=32, r=1/2	50, 162	4FSK	1.465	5.9	50	110.6	-31
JT9	K=32, r=1/2	72, 206	9FSK	1.736	15.6	19	49.0	-27
FT8	LDPC r = 1/2	87, 174	8FSK	6.25	50.0	27	12.6	-21

10.11.2018 (HB9AUR)

SWISS-ARTG GV 2018

33



Viel Spass mit den Weak Signal Modes!

Und vielen Dank für's Zuhören