

# Quick-Start Guide to FST4 and FST4W

Steve Franke, K9AN; Bill Somerville, G4WJS; and Joe Taylor, K1JT

*WSJT-X* 2.3.0 introduces FST4 and FST4W, digital protocols designed particularly for the LF and MF bands. On these bands their fundamental sensitivities are better than other *WSJT-X* modes with the same sequence lengths, approaching the theoretical limits for their rates of information throughput. FST4 is optimized for two-way QSOs, while FST4W is for quasi-beacon transmissions of WSPR-style messages. FST4 and FST4W do not require the strict, independent time synchronization and phase locking of modes like EbNaut.

The new modes use 4-GFSK modulation and share common software for encoding and decoding messages. FST4 offers T/R sequence lengths of 15, 30, 60, 120, 300, 900, and 1800 seconds, while FST4W omits the lengths shorter than 120 s. Submodes are given names like FST4-60, FST4W-300, etc., the appended numbers indicating sequence length in seconds. Message payloads contain either 77 bits, as in FT4, FT8, and MSK144, or 50 bits for the WSPR-like messages of FST4W. Message formats displayed to the user are like those in the other 77-bit and 50-bit modes in *WSJT-X*. Forward error correction uses a low density parity check (LDPC) code with 240 information and parity bits. Transmissions consist of 160 symbols: 120 information-carrying symbols of two bits each, interspersed with five groups of eight predefined synchronization symbols.

Basic parameters of all FST4 and FST4W submodes are summarized in the table below. Threshold sensitivity (SNR in 2500 Hz bandwidth yielding 50% probability of decode) was measured for each submode using simulations over the additive white Gaussian noise (AWGN) channel. As with other recently developed modes in *WSJT-X*, a feature called *a priori* (AP) decoding can improve sensitivity by several additional dB as information is accumulated during a standard minimal QSO or FST4W operating session.

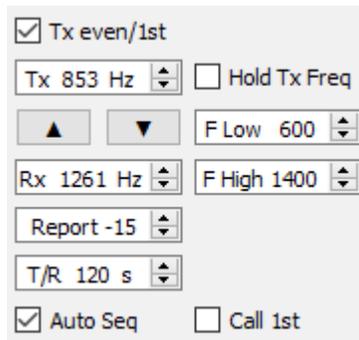
<b>T/R period (s)</b>	<b>Symbol length (s)</b>	<b>Tone Spacing (Hz)</b>	<b>Occupied Bandwidth (Hz)</b>	<b>FST4 SNR (dB)</b>	<b>FST4W SNR (dB)</b>
<b>15</b>	0.060	16.67	67.7	-20.7	
<b>30</b>	0.140	7.14	28.6	-24.2	
<b>60</b>	0.324	3.09	12.4	-28.1	
<b>120</b>	0.683	1.46	5.9	-31.3	-32.8
<b>300</b>	1.792	0.56	2.2	-35.3	-36.8
<b>900</b>	5.547	0.180	0.72	-40.2	-41.7
<b>1800</b>	11.200	0.089	0.36	-43.2	-44.8

FST4-60 is about 1.7 dB more sensitive than JT9, largely because it uses multi-symbol block detection where appropriate. With AP decoding in FST4 the difference can be as much as 4.7 dB. FST4-120

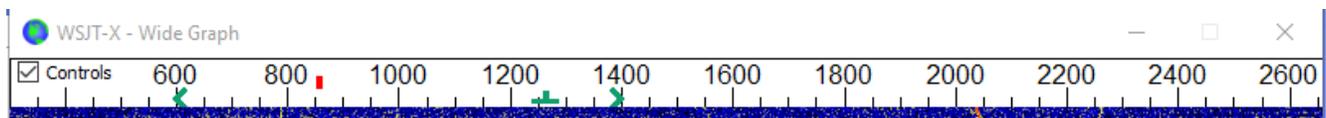
and the longer sequence lengths are proportionally more sensitive. FST4W-120 is about 1.4 dB better than standard WSPR, and with its 30-minute sequences FST4W-1800 reaches a threshold SNR of nearly -45 dB. We strongly recommend that users of JT9 and WSPR on the LF and MF bands should migrate to using FST4 and FST4W, instead.

You may think of other applications for the new modes beyond those discussed here. Keep in mind that these are very narrow-band modes; achieving the sensitivities listed in the table requires that oscillator drifts and path-induced Doppler shifts must be less than the tone spacing, over the full sequence length. As one example of a different application, the short-sequence submode FST4-15 has been found very effective on 50 MHz ionospheric scatter paths. At the opposite extreme of transmission length, VK7MO and VK7ZBX have had good success using FST4W-1800 for non-line-of-sight optical scattering communication, spanning obstructed paths as long as 153 km by using LED arrays, Fresnel lenses, and photo-detectors.

Operators familiar with *WSJT-X* will find using FST4 and FST4W straightforward. Most on-screen controls, auto-sequencing, and other features behave as in other modes. Operating conventions on the LF and MF bands make it useful to have additional user controls to set the active frequency range used by the decoder. When **File** → **Settings** → **General** → **Single decode** is unchecked, spin boxes labeled **F Low** and **F High** set lower and upper frequency limits for the FST4 decoder.



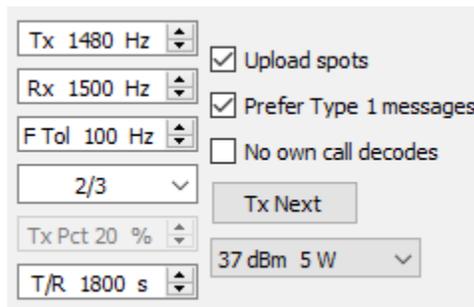
The limits are marked by dark green angle-bracket symbols < > on the Wide Graph frequency scale:



If **Single decode** is checked the **F Low** and **F High** controls and green < > markers disappear, and decoding takes place only in the range **RxFreq** ± **F Tol**.

For FST4W the default **Rx Freq** is 1500 Hz and **F Tol** is 100 Hz, so the active decoding range is the same as for WSPR, 1400 to 1600 Hz. However, you can select different center frequencies and **F Tol** values to conform with operating conventions on LF and MF bands.

A new drop-down control below **F Tol** offers a round-robin mode for scheduling FST4W transmissions:



The screenshot shows a settings panel with the following controls:

- Tx 1480 Hz (dropdown)
- Rx 1500 Hz (dropdown)
- F Tol 100 Hz (dropdown)
- 2/3 (dropdown)
- Tx Pct 20 % (dropdown)
- T/R 1800 s (dropdown)
- Upload spots
- Prefer Type 1 messages
- No own call decodes
- Tx Next (button)
- 37 dBm 5 W (dropdown)

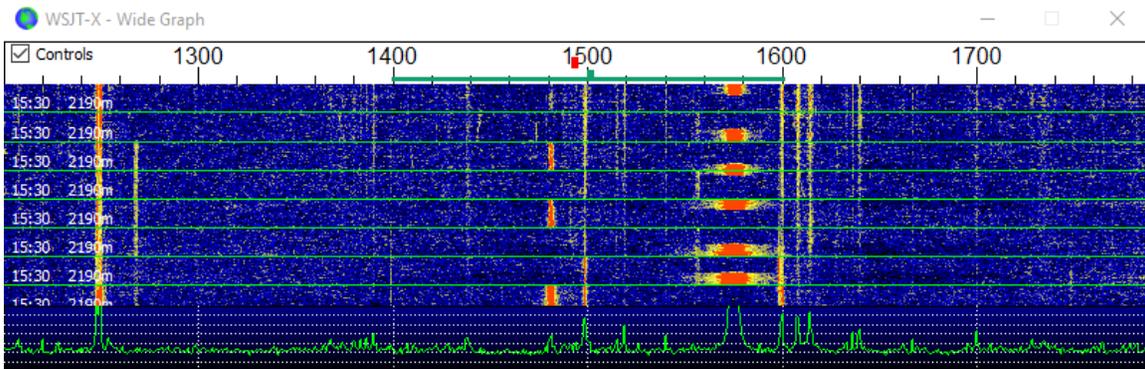
If three operators agree in advance to select the options **1/3**, **2/3**, and **3/3**, for example, their FST4W transmissions will occur in a fixed sequence with no two stations transmitting simultaneously. Sequence 1 is the first sequence after 00:00 UTC. For WSPR-like scheduling behavior, you should select **Random** with this control.

To see the presently suggested default FST4 and FST4W frequencies in the drop-down band-change control you must do a one-time reset. Go to **File** → **Settings** → **Frequencies**, then right-click on the **Working Frequencies** table and select **Reset**.

Optional noise blanking is available and has been found effective in handling atmospheric noise on the LF and MF bands. A spinner control labeled **NB nn %** is located on the FST4 and FST4W main window, just below the band-change control. Set this control to a suggested percentage of data samples to be blanked. We have found that levels in the range 5 – 15% work well in summer conditions on LF/MF bands, but you will probably want to experiment. Data displayed on the waterfall and saved to .wav files do not have noise blanking applied, so you can experiment after the fact.

An experimental feature uses negative settings of **NB** to trigger a “try everything” approach to noise blanking: -1% causes the decoder to try 0, 5, 10, 15, and 20% blanking, and -2% tries 0, 2, 4, ..., 20%. In FST4 mode the trials with nonzero blanking percentages are active only in the frequency range **RxFreq ± FTol**. Negative settings of **NB** can be very effective, but will slow the decoding procedure by significant amounts.

As an example of capabilities of the new modes, the screen shot on the next page shows FST4W-300 signals received on the 2200 m band at NO3M (locator EN91WR) on September 9, 2020. The distances involved are 3501 km to N6LF and 14,976 km to VK4YB. Numbers at the end of each decoded line are measured path Doppler spread in Hz. (To activate this feature, create a file named `plotspec` in the current working directory and start *WSJT-X* from the command line there.) As a general rule, decoding requires Doppler spreads less than the submode tone spacing. Sensitivity is best when Doppler spread is no more than 1/8 of the tone spacing.



Controls    Bins/Pixel 1    Start 1200 Hz    Palette    Adjust...     Flatten     Ref Spec    Spec 20 %  
 Split 2500 Hz    N Avg 50    Digipan    Cumulative    Smooth 2

WSJT-X v2.3.0-level by K1JT, G4WJS, and K9AN

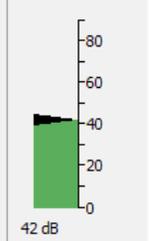
File   Configurations   View   Mode   Decode   Save   Tools   Help

UTC	dB	DT	Freq	Message
1020	-34	0.1	1555	VK4YB QG62 37 0.012
1025	-17	0.1	1480	N6LF CN83 30 0.004
1040	-34	0.0	1555	VK4YB QG62 37 0.025
1100	-34	0.1	1555	VK4YB QG62 37 0.026
1105	-22	0.1	1480	N6LF CN83 30 0.006
1120	-32	0.0	1555	VK4YB QG62 37 0.011
1125	-25	0.1	1480	N6LF CN83 30 0.002
1140	-34	0.0	1555	VK4YB QG62 37 0.020
1145	-33	0.1	1480	N6LF CN83 30 0.005

                            Menus

2190m    **0.136 000**

NB 0 %



**2020 Sep 09  
15:34:33**

Tx 1492 Hz     Upload spots  
 Rx 1500 Hz     Prefer Type 1 messages  
 F Tol 100 Hz     No own call decodes  
 Random      
 Tx Pct 20 %    30 dBm 1 W  
 T/R 300 s

200909\_1145.wav    Testing    **FST4W**    1    0/300

## Appendix A: Message Formats for FST4 and FST4W

Source encoding of FST4 messages is described in reference [1]. All messages are encoded into a 77-bit payload. To avoid transmitting a long string of zeros when sending CQ messages, the assembled 77-bit message is bitwise exclusive-OR'ed with the following pseudo-random sequence before computing the CRC and FEC parity bits:

```
010010100101111101000100110110100101100001000101001111001010101011011111000101
```

The receiving software applies this exclusive-OR procedure a second time, to restore the original 77-bit payload.

A 24-bit cyclic redundancy check (CRC) is calculated from and appended to each 77-bit information packet to create a 101-bit message-plus-CRC word. The CRC algorithm uses the polynomial 0x100065b (hexadecimal) and an initial value of zero.

Forward error correction is accomplished using a (240,101) LDPC code. The generator matrix has 139 rows and 101 columns. It is defined in the file `generator_fst4.dat`. Nonzero values in row  $i$  of the matrix specify which of the 101 message-plus-CRC bits must be summed, modulo 2, to produce the  $i^{\text{th}}$  parity-check bit. The 139 parity bits are appended to the 101 message-plus-CRC bits to create a 240-bit codeword.

Pairs of codeword bits are mapped to tone indices with values in the range 0 - 3 using the Gray encoding given in the third column of Table 3, in reference [1]. The resulting sequence of 120 channel symbols,  $a_n$ ,  $n=0, 1, 2, \dots, 119$  are divided into 4 groups:

$$M_A = \{a_0, a_1, \dots, a_{29}\}$$

$$M_B = \{a_{30}, a_{31}, \dots, a_{59}\}$$

$$M_C = \{a_{60}, a_{61}, \dots, a_{89}\}$$

$$M_D = \{a_{90}, a_{91}, \dots, a_{119}\}.$$

Synchronization is accomplished by embedding five 8-symbol synchronization words in the transmitted frame. The sync words are defined as follows:

$$S_1 = \{0, 1, 3, 2, 1, 0, 2, 3\}$$

$$S_2 = \{2, 3, 1, 0, 3, 2, 0, 1\}.$$

The complete set of 160 channel symbols is assembled as the sequence

$$b_n = \{S_1, M_A, S_2, M_B, S_1, M_C, S_2, M_D, S_1\}.$$

Source encoding of FST4W messages is described in reference [2]. All messages are encoded into a 50-bit payload. A 24-bit cyclic redundancy check (CRC) is calculated from and appended to each 50-bit information packet to create a 74-bit message-plus-CRC word. The CRC algorithm uses the polynomial 0x100065b (hexadecimal) and an initial value of zero.

Forward error correction is accomplished using a (240,74) LDPC code. The generator matrix has 166 rows and 74 columns. It is defined in the file `generator_fst4w.dat`. Nonzero values in row  $i$  of the matrix specify which of the 74 message-plus-CRC bits must be summed, modulo 2, to produce the  $i$ th parity-check bit. The 166 parity bits are appended to the 74 message-plus-CRC bits to create a 240-bit codeword.

The 240-bit codeword is mapped onto 160 channel symbols using the procedure described above for FST4.

#### **References:**

[1] Steve Franke, K9AN, Bill Somerville, G4WJS, Joe Taylor, K1JT “The FT4 and FT8 Communications Protocols,” *QEX*, July/August 2020, pp. 7-17.

[2] Steve Franke, K9AN, Bill Somerville, G4WJS, Joe Taylor, K1JT, in preparation.