

# FST4/FST4Wクイックスタートガイド

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

WSJT-Xバージョン2.3.0では、FST4とFST4WというLFとMFバンドをターゲットとした新しいモードを導入します。それらのバンドでの感度はいままでWSJT-Xに実装されてきた他のモードより高く、理論限界値に肉薄しています。FST4は2-way QSO、FST4WはWSPRのようなビーコンタイプの送信に最適化されています。FST4とFST4Wは、EbNautのような時間、周波数の厳密な制限を必要としません。

FST4は4-GFSK変調を用い、エンコードとデコードは共通のソフトウェアを使います。送受信サイクルは15、30、60、120、300、900、そして1800秒が用意されています。FST4Wは120秒より短いサイクルを省いています。サブモードの名前は、FST4-60、FST4W-300のように、シーケンス時間を秒で表した数字を後ろにつけます。メッセージペイロードは77ビット（FT4、FT8、MSK144で使われているものと同じ）か、あるいは、50ビット（WSPRで使われているものと同じ）です。前方誤り訂正符号として低密度パリティ検査（LDPC）を採用しており、情報とパリティを合わせて240ビットになります。一回の送信は160シンボルから成り、予め定義された8つの同期シンボルの5ブロックが間に挿入された120シンボル（それぞれ2ビット）で構成されています。

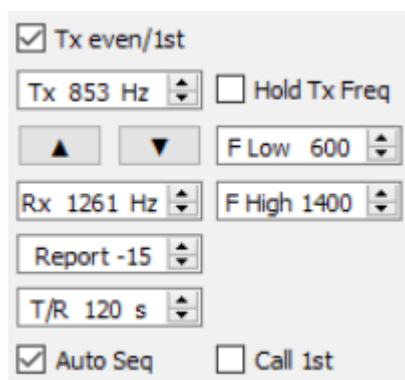
サブモードの基本パラメータを以下に示します。ここで示された感度（バンド幅2500Hz、デコード率50%）は、加算性白色ガウス雑音チャンネル上のシミュレーションで得られたものです。他のモードと同じく、APデコーディングを使うことでさらに数dB感度向上が見込まれます。

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

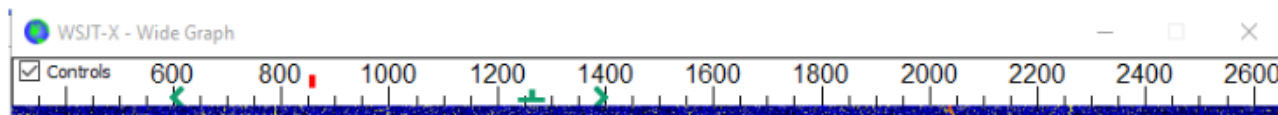
FST4-60はJT9より約1.7dB感度が向上しています。主に、マルチシンボルブロック検出を使う効果によるものです。さらにAPデコーディングを使うことで、最大4.7dB感度が向上します。FST4-120、そしてもっと長い送信時間のサブモードは、時間が長くなるとそれに比例して感度が良くなります。FST4W-120は、WSPRより、およそ1.4dB感度が向上しています。30分シーケンスのFST4W-1800ではSNRが-45dBに達します。LFとMFバンドのJT9とWSPRユーザーは、FST4とFST4Wへ移行するよう、強く推奨します。

ここで議論した以外の応用についてもいろいろ考えられるでしょう。FST4とFST4Wは非常に狭いバンド幅を利用していることを忘れないでください。表に示したような感度を得るためには1回の送信の間じゅう、発振器の周波数ドリフトやドップラーシフトがトーン間隔より十分小さくしなければなりません。他の応用の一例をあげると、50MHzの電離層スカッターで、短いサイクルのFST4-15がとても効果的でした。逆の例をあげると、VK7MOとVK7ZBXがFST4W-180を使い、見通し外光スカッター通信で153kmの交信を成功させています。

WSJT-Xに慣れているユーザーはFST4の使い方もすぐわかるでしょう。画面上のコントロール、自動シーケンス、そして他の機能は、今までのモードとほぼ同じです。LFとMFバンドにおける運用形態に沿って、デコーダの動作周波数範囲の設定を行うとよいでしょう。File → Settings → General → Single decodeのチェックを外し、F Low、F Highとラベル付けされたスピンボックスを使うことでFST4デコーダの動作周波数範囲をセットできます。



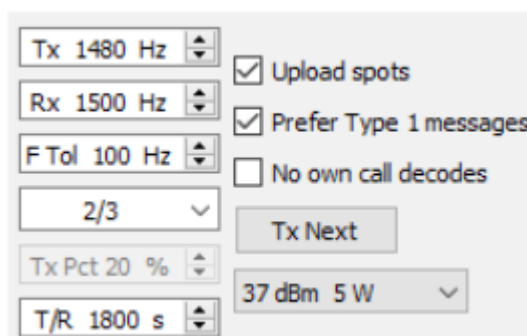
この周波数範囲設定はワイドグラフ上で緑色の〈〉で表示されます。



もし、Single decodeがチェックされていると、F LowとF Highのスピンボックスは消え、〈〉マーカーも表示されなくなります。デコード周波数範囲は、 $RxFreq \pm F Tol$ になります。

FST4WのデフォルトRx Freqは1500Hzに、F Tolは100に設定されます。したがって、FST4Wのアクティブな範囲はWSPRと同じく1400~1600Hzになります。ただし、LF、MFバンドに合った異なるRx FreqやF Tolを選択することも可能です。

FST4Wの送信シーケンス時間の選択の下にラウンドロビンモードが用意されています。



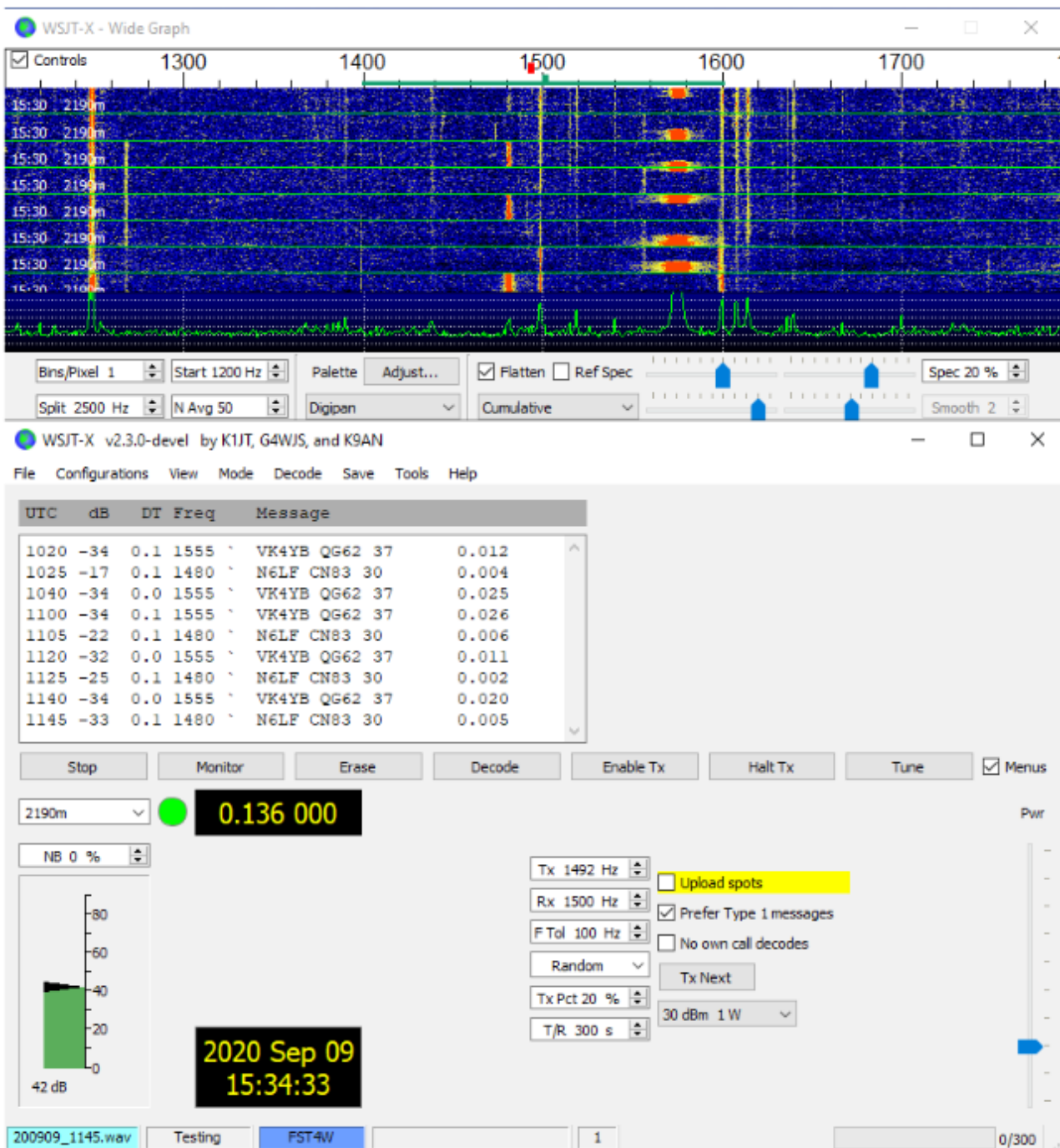
3人のユーザーが予め打ち合わせておき、それぞれが1/3、2/3、3/3を選択しておけば、同時に送信してしまわないようにスケジュールできます。1/3は00:00:00UTCから送信開始することになります。WSPRと同じように送信するときはRandomを選びます。

デフォルトのFST4推奨周波数は、バンド変更ドロップダウンリストから選べます。File → Settings → Frequenciesと進み、右クリックから Working Frequencies、Resetを行ってください。

LF、MFバンドにおける空間ノイズを軽減するためノイズブランクがオプションとして用意されています。FST4とFST4WウィンドウにNB nn%というスピナーコントロールがあり、指定のパーセンテージでデータサンプルをマスクすることができます。われわれの実験によると、夏のLF/MFバンドでは5~15%が良い結果を出していますが、みなさんでいろいろ試してみるとよいでしょう。ウォーターフォール表示、及びそれを録音した.wavファイルはブランキングをかける前の信号です。

NBに負の値を設定することで、「全部試す」的なノイズブランクを実験することができます。-1%にセットするとデコーダーは0、5、10、15、20%のブランキングを実行します。-2%をセットすると、0、2、4、...20%のブランキングを実行します。FST4では $RxFreq \pm F Tol$ の周波数範囲でのみ実行します。非常に効果的ですが、デコードに大変時間がかかるようになります。

新モードの性能を次の図に例示します。2020年9月9日にNO3M（グリッドロケータ-EN91WR）が受信した2200mバンドのFST4W-300の信号です。N6LFとの距離は3501km、VK4YBとの距離は14,976kmです。デコードされたメッセージの右端はドップラースプレッドの周波数がHz単位で表示されます。この機能をオンにするためには、作業フォルダーに plotspec というファイルを作り、コマンドラインからWSJT-Xを起動してください。一般的なルールとしてドップラースプレッドがサブモードのトーン間隔より小さくないとデコードできません。ドップラースプレッドがトーン間隔の8分の1より狭ければ、ベストです。



## 付録A : FST4とFST4Wのメッセージ形式

FST4のエンコーダ詳細については文献[1]をご覧ください。すべてのメッセージは77ビットペイロードに変換されます。CQを送信する際、0が連続することを防止するためCRCとFECを計算する前に次の疑似乱数と排他的論理和をとります。

01001010010111101000100110110100101100001000101001111001010101011011111000101

受信側では、もう一度この疑似乱数を受信データに排他的論理和としてかけ、元の77ビットを算出します。

24ビットのCRCは77ビットのペイロードから算出され、そのペイロードに付加されます。その結果、全体として101ビットのメッセージ+CRCのデータが得られます。CRCアルゴリズムは多項式 $0x100065B$ 、初期値0を使っています。

前方誤り訂正符号は、(240,101)のLDPCを使っています。生成行列は139行101列です。generator\_fst4.datファイルの中に定義されています。i行目の非ゼロの値が101ビットのメッセージ+CRCのどのビットを足し込むか指定し、そしてi番目のパリティチェックビットを生成します。139パリティビットが101ビットのメッセージ+CRCに連結され、240ビットのコードワードが作られます。

コードワードのペアが0から3までのトーンインデックスを指定します。参考文献[1]の表3の3番目の列で指定されるグレーコードでエンコードをかけます。結果として得られた120個のチャンネルシンボル  $a_n$ ,  $n = 0, 1, 2, \dots, 119$ が4つのグループに分けられます。

$$\begin{aligned}M_A &= \{a_0, a_1, \dots, a_{29}\} \\M_B &= \{a_{30}, a_{32}, \dots, a_{59}\} \\M_C &= \{a_{60}, a_{61}, \dots, a_{89}\} \\M_D &= \{a_{90}, a_{91}, \dots, a_{119}\}\end{aligned}$$

同期のため5つの8シンボル同期ワードが送信フレームに埋め込まれます。同期ワードは次のように定義されます。

$$\begin{aligned}S_1 &= \{0, 1, 3, 2, 1, 0, 2, 3\} \\S_2 &= \{2, 3, 1, 0, 3, 2, 0, 1\}\end{aligned}$$

最後の160チャンネルシンボルはそれらを連結して作られます。

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

FST4Wのメッセージ形式は参考文献[2]に説明があります。メッセージはすべて50ビットのペイロードを持ちます。24ビットのCRCがその50ビットから計算され、連結することで74ビットのメッセージ+CRCビット列が作られます。CRCアルゴリズムは多項式 $0x100065B$ 、初期値0を使っています。

前方誤り訂正符号は、(240,74)のLDPCを使っています。生成行列は166行74列です。generator\_fst4w.datに定義されています。i行目の非ゼロの値が74ビットのメッセージ+CRCのどのビットを足し込むか指定し、そしてi番目のパリティチェックビットを生成します。166パリティビットが74ビットのメッセージ+CRCに連結され、240ビットのコードワードが作られます。

240ビットのコードワードはFST4で示したと同じように160チャンネルシンボルへマップされます。

## 参考文献

[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.

日本語訳 大庭 JA7UDE 2020年9月28日公開  
2021年6月5日更新