WSJT8 User's Guide

Joe Taylor, K1JT

Last updated: June 18, 2010

Overview

WSJT8 is an experimental version of the familiar weak-signal communication program WSJT. It offers new protocols or "modes" optimized for meteor scatter, ionospheric scatter, EME, microwaves, and QRP at HF. Like the modes in previous versions of WSJT, the new ones are intended for making minimal QSOs, not for rag-chewing. WSJT8 modes are not compatible with those in WSJT7. To use the new modes, transmitting and receiving stations must both use WSJT8.

This introductory User's Guide explains how the new experimental modes differ from the familiar modes **FSK441**, **JT6M**, **JT65**, and **JT4**. It assumes that you are already familiar with installing and using WSJT7. Please consider the Guide is a living document; it will be modified and extended frequently in coming weeks. The online version can be called up with a single keystroke from within WSJT8, so you can easily check for updates.

My working hypothesis is that the new modes in WSJT8 offer significant advantages over those in WSJT7 — and that it will probably make sense to recommend a complete switch-over some months from now. For that transition to happen in an orderly fashion, we'll need a widely held consensus that the change is for the better. If one of the new modes does not live up to present expectations, or for another reason it's decided that some feature of WSJT7 should be retained, that wish can probably be accommodated.

Input from users will be important following the initial beta release of WSJT8. I need to know what works, what doesn't work, what should be improved, and what enhancements should be given highest priority. All comments and suggestions will be greatly appreciated!

Please note: *Before using WSJT8, be sure to read this full document to familiarize yourself with the design goals and motivations for the new modes, and with the available message formats.*

Installation

WSJT8 may be downloaded from the WSJT Home Page, http://www.physics.princeton.edu/pulsar/K1JT/. Click on the WSJT link at the left margin and then on the appropriate download link. Install the program in the usual way for your computer platform. Under Windows, execute the downloaded file and follow the installation instructions. I recommend installing WSJT8 in a different directory from WSJT7, for example in the Windows folder C:\Program Files\WSJT8. For installation under Debian-based Linux distributions, see Appendix D.

Setup

On the **Station parameters** tab of the **Setup | Options** window, select a PTT port and audio devices from the drop-down lists. As in WSJT7, you must exit and restart the program for a new choice of audio devices to take effect. This restriction will be removed in a later release.

The **Message templates** tab offers four suggested formats for basic QSOs. The button labeled **Standard** generates messages normally used for EME and for meteor scatter in North America; **Num Rpts** generates messages with callsigns and numerical signal reports; **Hashed calls** uses hash codes for callsigns already copied (see below for more details); and **Shortest** creates messages in the shortest format conveying all necessary information. Appendix B illustrates QSO formats using messages of different lengths.

The **EME Echo Mode** tab provides entry fields for the parameters needed for testing your own echoes from the moon.

Old Modes, New Modes

Each new mode introduced in WSJT8 is intended as a possible replacement for a mode in WSJT7. The suggested migration path between old and new modes is as follows:

Present mode (WSJT7)	New mode (WSJT8)	Typical applications
FSK441	**JTMS**	Meteor scatter on 2m
JT6M	**ISCAT**	Ionospheric scatter on 6m
JT65	**JT64**	EME at VHF and UHF, QRP at HF
JT4	**JT8**	EME, microwaves, QRP at HF

JT65 offers three sub-modes with different spacings between the audio tones, and **JT4** offers seven. **JT64** and **JT8** will also offer such options, but at this time only the narrowest spacing has been implemented.

Message Formats (Old Modes)

Messages in **FSK441** and **JT6M** are sent character-by-character and have no fixed structure, no start-of-message synchronization, and no error-correcting features. In contrast, **JT65** and **JT4** use structured messages and include synchronization and forward error correction (FEC). The structured messages can have one of three basic formats:

1. Two to four alphanumeric “words” of specified form
2. Arbitrary text, up to 13 characters
3. Shorthand messages for RO, RRR, and 73 (but no shorthands in **JT4**)

The words of a type 1 message normally consist of two standard callsigns, an optional grid locator, and the optional signal report OOO. CQ or QRZ can be substituted for the first callsign. An add-on callsign prefix followed by “/”, a suffix preceded by “/”, a signal report of the form –nn or R–nn, or the message fragments RO, RRR or 73 can be substituted for the grid locator. Each message of type 1 or type 2 conveys exactly 72 bits of user information.

Message Formats (New Modes)

In WSJT8 all modes use structured messages with robust synchronization and strong FEC. The basic message format is similar to that used in **JT65** and **JT4**, but is more flexible because the information content can be 2, 30, 48, or 78 bits. Forty-eight and 78-bit messages allow arbitrary add-on prefixes up to 3 characters, rather than prefixes selected from a limited list. Numerical signal reports may be anywhere in the range –30 to +39, and free-text messages can be up to 14 characters long. The 48-bit, 30-bit, and 2-bit messages convey less information but can be copied at progressively weaker signal levels.

All user messages are automatically source-encoded into the most efficient length. If a message fails to conform to one of the defined formats it is truncated to 14 characters and encoded as free text.

Many of the 48-bit message formats include one full callsign and one "hash-coded" callsign. The hash codes are shortened representations that stand in for callsigns already copied during a QSO. The transmission and reception of the hash-coded forms is indicated by angle brackets surrounding the callsign, as in <K1JT> or <ZA/PA2CHR>. Since hashing is a many-to-one mapping, the process is not uniquely reversible; however, if a full callsign has been decoded in a previous transmission, the decoder may reasonably assume that matching hash codes imply matching callsigns. With a 15-bit hash code, the chances of misidentification are very small, especially within the confines of a particular QSO. If a hash code is received that does not match one already in the program's dynamically maintained hash table, it is displayed as <**…**>.

Examples of the full range of message formats are provided in Appendix B.

Specific Comparisons of Old and New Modes

1. Meteor Scatter on 2m

FSK441 uses continuous-phase 4-FSK (4-tone frequency-shift keying) at 441 baud. With adequate signal-to-noise ratio — minimum ping strength (S+N)/N around 2 dB — the protocol is self synchronizing at the character level, but it provides no beginning-of-message synchronization. The character transmission rate is 147 cps, so reception of a typical message like "WA1ABC 26 KC0XYZ 2626 " requires a ping duration of at least 150 ms. Single-tone shorthand messages for R26, R27, RRR, and 73 are provided for use when appropriate. With no FEC and no message synchronization, **FSK441** typically produces lots of on-screen garbage when attempting to decode a marginal signal.

JTMS uses a K=13, r=1/2 convolutional code and appends a 32-bit synchronizing vector to each encoded message. The number of transmitted symbols is thus 2*(n+K-1)+32, where n=30, 48, or 78 is the number of information bits. Modulation is MSK ("minimum shift keying") at 1500 baud. The raw data rate is 1500 bps and the bandwidth 2250 Hz. Thirty-bit, 48-bit, and 78-bit user messages are transmitted in 77, 101, and 141 ms, respectively, but because of two-fold redundancy in the convolutional code they may sometimes be decoded even in shorter pings. The 32-bit sync vector makes it easy for the decoder to recognize start-of-message and to determine the frequency offset DF to within a few Hz. Strong FEC means that messages are usually copied correctly, or not at all. Shorthands like those in **FSK441** are provided. The sensitivity of **JTMS** is about the same as **FSK441**, but short pings can be used more effectively and garbage decodes are much fewer. There is seldom any need to piece together message fragments to obtain a complete message.

2. Ionospheric Scatter on 6 m

JT6M was designed to take advantage of the mixture of ionospheric scatter and meteor scatter typically seen at distances 800-2000 km, at 50 MHz. It has also proven useful for weak-signal E_s and F_2 propagation. As in **FSK441**, user messages are sent character by character, with no error-correcting code. Modulation is continuous-phase 44-FSK at 21.5 baud. Each tone represents a distinct character from an alphabet of 43 characters, or else a synchronizing tone; the sync tone is sent in every 3rd symbol interval. The character transmission rate is 14.4 cps, and no message synchronization is provided. Messages can sometimes be decoded at S/N down to –10 dB. Decoding of marginal signals often requires considerable user effort and produces much on-screen garbage.

ISCAT uses Reed Solomon codes RS(63,5), RS(63,8), and RS(63,13), with 6-bit symbols. Synchronizing information is appended in the form of a 10×10 Costas array followed by two symbols to distinguish between three possible message lengths. Modulation is continuous-phase 64-FSK keyed at 23.4 baud, and total bandwidth is 64×23.4 = 1500 Hz. Messages are repeated 9.3 times in a 30 s transmission. **ISCAT** signals are decodable at average S/N down to about –15 dB. Decoding is much more reliable than with **JT6M**, and garbage decodes much fewer.

3. EME at VHF and UHF, QRP at HF

JT65 was designed for EME on the VHF bands. It has proven effective also for EME at UHF and for QRP contacts at HF. Messages are source-encoded to 72 bits and carry Reed-Solomon error control using an RS(63,12) code and 6-bit symbols. Synchronizing information is sent as a pseudo-random pattern of 126 ON/OFF intervals of a single tone. Encoded symbols are sent using 64-MSK modulation during the 63 sync-OFF intervals. The keying rate is 2.69 baud, so a full transmission takes 126/2.69 = 46.8 s. Total bandwidth is 178 Hz (sub-mode **JT65A**), 355 Hz (**JT65B**), or 711 Hz (**JT65C**). Decoding in WSJT7 uses the Kötter-Vardy algebraic soft-decision algorithm, and arbitrary **JT65B** messages can be reliably copied down to S/N = –24 dB. The "Deep Search" decoder can identify callsigns found in a local database down to –28 dB. Two-tone shorthand codes are used to convey the simple messages RO, RRR, and 73, and these messages can be copied down to –32 dB.

JT64 builds on the best features of **JT65** and adds further enhancements. Synchronization is achieved by means of three 6×6 Costas arrays instead of a unique sync tone. Each Costas array is followed by two additional symbols to identify the message length. The result is that 28% of transmitted energy is devoted to synchronization, compared with 50% in **JT65** — yet sync performance is more reliable than in **JT65**. Messages of 78 bits can be copied down to –26 dB, and 30-bit messages to –29 dB.

4. Alternative Modes for EME, Microwaves, and QRP at HF

JT4 uses 4-FSK modulation at 4.375 baud and the same message structure and source encoding as JT65. Each transmitted symbol contains a sync bit and a data bit. A number of different tone spacings are offered. Forward error correction uses a long-constraint convolutional code (constraint length K=32, code rate r=1/2), which makes these modes behave rather like JT65: at the receiving end you either get the exact message that was transmitted, or you get nothing at all. The **JT4A** submode has 4.375 Hz tone spacing and 17.5 Hz total bandwidth, just ten percent of the **JT65A** bandwidth. Its measured performance is 1 dB worse than **JT65A**. At the other extreme, the **JT4G** submode uses 315 Hz tone spacing and 1260 Hz total bandwidth. The wider **JT4** submodes have proven very effective for EME on the higher microwave bands and for troposcatter and rain scatter at 10 GHz.

JT8 is proposed as a possible alternative to **JT4**. Modulation is 8-FSK at 2.857 baud. FEC uses convolutional codes with K=14, r=1/4, K=15, r=1/6, or K=16, r=1/8, depending on message length. Synchronization uses 8×8 Costas arrays at the beginning and end of a transmission, followed by two additional symbols to distinguish between 30-bit, 48-bit, and 78-bit messages. Total bandwidth is 23 Hz. At present, only the 78-bit messages have been implemented.

Echo Mode

Echo mode was a popular feature of early versions of WSJT and has been re-introduced in WSJT8. It allows you to make sensitive measurements of your own echoes from the moon, even when they are too weak to be heard. To use it, select **Echo** from the **Mode** menu, aim your antenna at the moon, pick a clear frequency, and toggle the **Auto** button to **ON**. The program will then start cycling through the following loop:

1. Transmit a fixed tone for 2.0 s
2. Wait about 0.5 s for start of return echo
3. Record the received signal for 2.0 s
4. Analyze, average, and display the results
5. Repeat from step 1

At the start of each transmission the frequency of the transmitted tone is offset randomly around a nominal value of 1500 Hz. A number in the entry field **Dither (**on the Setup **|** Options **|** Echo Mode tab) controls the magnitude of the random offset. The observed spectrum of each echo is shifted by the dither amount before being accumulated into the average. This procedure is very effective in minimizing the impact of birdies in the receiver passband. In the average spectrum, a fixed-frequency birdie is smeared out over a wide range while the desired signal remains sharply defined.

Information displayed in the main text box gives the number **N** of completed echo cycles, the average **Level** of receiver background noise in dB, the average echo strength **Sig** in dB, its measured frequency offset **DF** in Hz (after correction for Doppler shift), its spectral **Width** in Hz, and a relative quality indicator **Q** on a 0–10 scale. Background noise level is given with respect to the nominal “0 dB” level used for all WSJT modes. Signal strength is measured in dB relative to the noise power in the full receiver passband, nominally 2500 Hz. Low numbers for **Q** mean that an echo has not been detected or is unreliable; for **Q**=0 the values of **DF** and **Width** are surely meaningless, and **Sig** is an upper limit. Larger values of **Q** imply increasingly believable echo measurements. If you can hear your own lunar echoes, you will see reliable echo numbers within a few seconds after toggling **Auto ON**. If your echoes are 15 to 20 dB below the audible threshold you should get significant results within a few minutes.

By default EME Echo mode assumes that your receiver and transmitter are tuned to the same frequency. The entry field **RIT** on the Echo Mode tab lets you inform the program of any offset receiver tuning — for example, to accommodate a large Doppler shift. Suppose you are running a test on 23 cm and the predicted Doppler shift at the start of the run is –1400 Hz. In that case the 1500 Hz transmitted audio tone would be detected at 100 Hz, probably below the low-frequency cutoff in your receiver's passband. Use your transceiver's RIT control to offset the receiver tuning by the predicted Doppler shift or a nearby rounded value, and enter this offset in the RIT box before starting the echo measurement. The program will track subsequent Doppler changes up to about 800 Hz, if necessary, without further adjustments. Your echo should appear near **DF**=0, as usual. You won't need to use the RIT feature on 6 or 2 meters, where Doppler shifts are much smaller.

The frequency of a valid echo should be well defined, stable, and very close to **DF**=0. If you toggle **Auto** off and then on again to start a new measurement, the echo signal should build up again at the same **DF**. To be absolutely sure that you are seeing you own echo, offset your transmitter frequency by a known amount, say 50 Hz, while holding the receiver frequency constant. A valid echo will shift by the same 50 Hz.

A mode-specific graphical display has yet to be implemented for **Echo** mode, but will be added soon. In the meantime, you can see your strong echoes on the waterfall display.

Request for User Input

In its present form WSJT8 is an experimental release. The program is almost fully functional and has many new features, but (as of June 2010) it still has a few gaps. The decoders work well but may not be protected effectively against QRM, signals in alien modes, and the like. They do not have AFC or other drift-compensating code, and they have not been optimized for speed. Many test QSOs have been made in each of the new modes, but a number of possible circumstances remain untested. Sub-modes such as **JT64B, C**, …, **JT8B, C**, … have not yet been implemented, and there is no mode-specific graphical display in **Echo** mode.

In no particular order, here are some questions on which I would appreciate user input. You can probably think of many others, as well. Please share your views and opinions!

1. Do the new modes perform better than the ones they were designed to replace? I need to hear your own mode-by-mode comparisons and opinions. Don’t be afraid to be critical!
2. Are any of the WSJT7 modes “too good to lose”?
3. Is the lack of a unique sync tone, easily visible on a waterfall, a serious problem for **JT64**?
4. An alternative (perhaps optional?) configuration for **JT64** could use T/R sequences of 30 s rather than 60 s. In EME contest situations this could permit QSO rates up to 20 per hour. Sensitivity would be 3 dB worse, but still nearly as good as **JT65B** with its one-minute sequences. Would this be desirable?
5. How about 15 s sequences in **JTMS** and **ISCAT** modes?

6. What additional program features are most wanted?
7. When you have gained some experience with the new modes, please save and send me any recorded files that you think should have decoded, but did not. Such cases are extremely helpful in the process of optimizing the decoders.
8. If there will be an eventual changeover from WSJT7 to WSJT8, with an entirely new (or mostly new) set of modes, how should it be orchestrated so as to minimize confusion and chaos?

Appendix A. Examples of Minimal QSOs

The following examples illustrate QSO formats selectable from the **Message templates** tab of the **Setup | Options** window. They show proper and effective usage of the various message lengths, always assuring the exchange of all necessary information for a valid QSO. There are many other possible formats, as well.

```
Standard                                    nbits
-------------------------------------------------
1. CQ K1ABC FN42                              48
2.                       K1ABC W9XYZ EM57     78
3. W9XYZ K1ABC FN42 OOO                       78
4.                       RO                    2
5. RRR                                         2
6.                       73                    2

Numerical reports
------------------------------------------------
1. CQ K1ABC FN42                              48
2.                       K1ABC W9XYZ EM57     78
3. W9XYZ K1ABC -21                            78
4.                       K1ABC W9XYZ R-22     78
5. RRR                                         2
6.                       73                    2

Hashed calls
------------------------------------------------
1. CQ K1ABC FN42                              48
2.                       <K1ABC> W9XYZ        48
3. W9XYZ <K1ABC> OOO                          48
4.                       K1ABC <W9XYZ> RO     48
5. RRR                                         2
6.                       73                    2

Hashed calls (JTMS mode)
------------------------------------------------
1. CQ K1ABC FN42                              48
2.                       <K1ABC> W9XYZ        48
3. W9XYZ <K1ABC> 26                           48
4.                       K1ABC <W9XYZ> R27    48
5. RRR                                         2
6.                       73                    2

Shortest
------------------------------------------------
1. CQ K1ABC                                   30
2.                       DE W9XYZ             30
3. W9XYZ OOO                                  30
4.                       K1ABC RO             30
5. RRR                                         2
6.                       73                    2
```

Appendix B. WSJT8 Message Formats

Here's a brief summary of typical contents for the four different message lengths:

- 2-bit – shorthand messages for RO, RRR, and 73
- 30-bit – one callsign together with CQ, DE, OOO, RO; or a selection of canned messages such as "RRR TNX 73", "BEST -19", or a grid locator
- 48-bit – one callsign together with one or more of: a hash-coded callsign, a grid locator, an add-on prefix or suffix, and/or special items like CQ, QRZ, OOO, RO, RRR
- 78-bit – two callsigns with a grid locator, add-on prefix or suffix, and/or signal report, or arbitrary text up to 14 characters

The following table illustrates WSJT8 message formats in a more complete way. Input messages are shown at the left, followed by the number of bits, the source-encoded data in hexadecimal format, and the decoded message as it will appear at the receiving station. An asterisk is present before any decoded message not precisely equal to the original.

Message	Bits	Source-encoded data	OK?	Decoded message
RO	2	40000000 00000000 0000		RO
RRR	2	80000000 00000000 0000		RRR
73	2	C0000000 00000000 0000		73
CQ K1ABC	30	F70C2380 00000000 0000		CQ K1ABC
DE W9XYZ	30	F9777BB4 00000000 0000		DE W9XYZ
W9XYZ OOO	30	F9777BB8 00000000 0000		W9XYZ OOO
K1ABC RO	30	F70C238C 00000000 0000		K1ABC RO
RRR TNX 73	30	FA087420 00000000 0000		RRR TNX 73
TNX 73 GL	30	FA087430 00000000 0000		TNX 73 GL
GRID?	30	FA087040 00000000 0000		GRID?
EM77	30	FA0E4B60 00000000 0000		EM77
BEST -19	30	FA087100 00000000 0000		BEST -19
BEST +29	30	FA087400 00000000 0000		BEST +29
BEST 29	30	FA087400 00000000 0000	*	BEST +29
BEST -1	30	FA087220 00000000 0000	*	BEST -01
BEST 1	30	FA087240 00000000 0000	*	BEST +01
CQ K1ABC FN20	48	F70C238B 39C00000 0000		CQ K1ABC FN20
QRZ K1ABC	48	F70C238F E1E00000 0000		QRZ K1ABC
CQ 118 K1ABC	48	F70C238D D7C20000 0000		CQ 118 K1ABC
CQ VO2/K1ABC	48	F70C2385 28220000 0000		CQ VO2/K1ABC
CQ K1ABC/2	48	F70C238D 4C420000 0000		CQ K1ABC/2
<K1ABC> W9XYZ	48	F9777BB3 2F230000 0000		<K1ABC> W9XYZ
<VO2/K1ABC> W9XYZ	48	F9777BBD 6E630000 0000		<VO2/K1ABC> W9XYZ
<K1ABC/2> W9XYZ	48	F9777BBE 38230000 0000		<K1ABC/2> W9XYZ
DE VO2/W9XYZ	48	F9777BB5 28250000 0000		DE VO2/W9XYZ
DE W9XYZ/2	48	F9777BBD 4C450000 0000		DE W9XYZ/2
DE W9XYZ EM77	48	F9777BBB A9E60000 0000		DE W9XYZ EM77

Appendix B (continued)

```
Message                 Bits  Source-encoded data   OK? Decoded message
-----------------------------------------------------------------------------

W9XYZ <K1ABC> OOO         48  F9777BB3 2F270000 0000   W9XYZ <K1ABC> OOO
W9XYZ <VO2/K1ABC> OOO     48  F9777BBD 6E670000 0000   W9XYZ <VO2/K1ABC> OOO
VO2/W9XYZ OOO             48  F9777BB5 28290000 0000   VO2/W9XYZ OOO
DE VO2/K1ABC OOO          48  F70C2385 282B0000 0000   DE VO2/K1ABC OOO
DE K1ABC FN20 OOO         48  F70C238B 39CC0000 0000   DE K1ABC FN20 OOO

K1ABC <W9XYZ> RO          48  F70C2388 FAED0000 0000   K1ABC <W9XYZ> RO
K1ABC <VO2/W9XYZ> RO      48  F70C238C 874D0000 0000   K1ABC <VO2/W9XYZ> RO
VO2/K1ABC RO              48  F70C2385 282F0000 0000   VO2/K1ABC RO
DE VO2/W9XYZ RO           48  F9777BB5 28310000 0000   DE VO2/W9XYZ RO
DE K1ABC FN20 RO          48  F70C238B 39D20000 0000   DE K1ABC FN20 RO

<K1ABC> W9XYZ RRR         48  F9777BB3 2F330000 0000   <K1ABC> W9XYZ RRR
<VO2/K1ABC> W9XYZ RRR     48  F9777BBD 6E730000 0000   <VO2/K1ABC> W9XYZ RRR
K1ABC <W9XYZ> RRR         48  F70C2388 FAF40000 0000   K1ABC <W9XYZ> RRR
K1ABC <VO2/W9XYZ> RRR     48  F70C238C 87540000 0000   K1ABC <VO2/W9XYZ> RRR
VO2/K1ABC RRR             48  F70C2385 28360000 0000   VO2/K1ABC RRR
DE VO2/W9XYZ RRR          48  F9777BB5 28380000 0000   DE VO2/W9XYZ RRR

73 DE W9XYZ EM77          48  F9777BBB A9F90000 0000   73 DE W9XYZ EM77
73 DE VO2/W9XYZ           48  F9777BB5 283B0000 0000   73 DE VO2/W9XYZ
TNX RUSSELL 73 GL         48  9A924C7A EC5C0000 0000   TNX RUSSELL 73 GL
OP ROBERTO 73 GL          48  988C2B7B 4BBD0000 0000   OP ROBERTO 73 GL

K1ABC W9XYZ EM77          78  F70C238F 9777BBBA 9E00   K1ABC W9XYZ EM77
K1ABC W9XYZ/2             78  F70C238F 9777BBD4 C510   K1ABC W9XYZ/2
K1ABC W9XYZ               78  F70C238F 9777BBFD 2200   K1ABC W9XYZ
VO2/K1ABC W9XYZ           78  F70C238F 9777BB52 8290   VO2/K1ABC W9XYZ
K1ABC VO2/W9XYZ           78  F70C238F 9777BB52 8310   K1ABC VO2/W9XYZ
K1ABC W9XYZ EM77 OOO      78  F70C238F 9777BBBA 9E08   K1ABC W9XYZ EM77 OOO
K1ABC W9XYZ OOO           78  F70C238F 9777BBFD 2208   K1ABC W9XYZ OOO
VO2/K1ABC W9XYZ OOO       78  F70C238F 9777BB52 8298   VO2/K1ABC W9XYZ OOO
K1ABC W9XYZ -21           78  F70C238F 9777BBFD 3600   K1ABC W9XYZ -21
K1ABC W9XYZ +21           78  F70C238F 9777BBFD 8A00   K1ABC W9XYZ +21
K1ABC W9XYZ RO            78  F70C238F 9777BBFE 1800   K1ABC W9XYZ RO
K1ABC W9XYZ R-21          78  F70C238F 9777BBFD B000   K1ABC W9XYZ R-21
K1ABC W9XYZ R+21          78  F70C238F 9777BBFE 0400 * K1ABC W9XYZ R21
K1ABC W9XYZ R21           78  F70C238F 9777BBFE 0400   K1ABC W9XYZ R21
K1ABC W9XYZ R EM77        78  F70C238F 9777BBBA 9E20   K1ABC W9XYZ R EM77
K1ABC W9XYZ RRR           78  F70C238F 9777BBFE 1A00   K1ABC W9XYZ RRR
K1ABC W9XYZ 73            78  F70C238F 9777BBFE 1C00   K1ABC W9XYZ 73
123456789ABCDE            78  4629667D 059FAB1B 6654   123456789ABCDE
123456789ABCDEF           78  4629667D 059FAB1B 6654 * 123456789ABCDE
```

Appendix B (continued)

In **JTMS** mode, 2-bit messages and 48-bit messages involving a signal report are slightly different from those shown above. Where OOO and RO are used in the other modes, **JTMS** uses 26, 27, R26, and R27 in order to be consistent with standard meteor scatter usage.

Message	Bits	Source-encoded data	OK?	Decoded message
R26	2	00000000 00000000 0000		R26
R27	2	40000000 00000000 0000		R27
RRR	2	80000000 00000000 0000		RRR
73	2	C0000000 00000000 0000		73
W9XYZ <K1ABC> 26	48	F9777BB3 2F270000 0000		W9XYZ <K1ABC> 26
W9XYZ <K1ABC> 27	48	F9777BB3 2F3E0000 0000		W9XYZ <K1ABC> 27
W9XYZ <VO2/K1ABC> 26	48	F9777BBD 6E670000 0000		W9XYZ <VO2/K1ABC> 26
VO2/W9XYZ 26	48	F9777BB5 28290000 0000		VO2/W9XYZ 26
DE VO2/K1ABC 26	48	F70C2385 282B0000 0000		DE VO2/K1ABC 26
DE K1ABC FN20 26	48	F70C238B 39CC0000 0000		DE K1ABC FN20 26
K1ABC <W9XYZ> R26	48	F70C2388 FAED0000 0000		K1ABC <W9XYZ> R26
K1ABC <W9XYZ> R27	48	F70C2388 FAFF0000 0000		K1ABC <W9XYZ> R27
K1ABC <VO2/W9XYZ> R26	48	F70C238C 874D0000 0000		K1ABC <VO2/W9XYZ> R26
VO2/K1ABC R26	48	F70C2385 282F0000 0000		VO2/K1ABC R26
DE VO2/W9XYZ R26	48	F9777BB5 28310000 0000		DE VO2/W9XYZ R26
DE K1ABC FN20 R26	48	F70C238B 39D20000 0000		DE K1ABC FN20 R26

Appendix C. Parameters of Digital Modes in WSJT7 and WSJT8

Each group of lines in the following table includes a WSJT7 mode and the corresponding WSJT8 mode designed for similar purposes. All WSJT7 and WSJT8 modes use timed T/R sequences. All S/N values use a 2500 Hz reference bandwidth. WSJT7 uses a 11025 Hz sample rate, while WSJT8 uses 12000 Hz.

Definitions:

Mode – Name of the digital mode
T/R – Length of Tx and Rx sequences (s)
Mod – Modulation type
Bits – Number of information bits per message
FEC – Forward error correction code
Nsps – Number of samples per symbol
Baud – Keying rate
BW – Signal bandwidth (Hz)
Sync – Synchronizing scheme
TxT – Time to transmit encoded message once (s)
S/N – Approximate threshold for successful decoding (dB)

Mode	T/R	Mod	Bits	FEC	Nsps	Baud	BW	Sync	TxT	S/N	Sub-modes
1. Meteor scatter on 2m											
FSK441	30	4-FSK	*	N/A	25	441	1764	*	0.150*	-1	
JTMS	30	MSK	78	K=13, r=1/2	8	1500	2250	1x32	0.141	-1	
			48	K=13, r=1/2	8	1500	2250	1x32	0.101	-1	
			30	K=13, r=1/2	8	1500	2250	1x32	0.077	-1	
			2	N/A	128	93.8	1220	N/A	0.021	-3	
2. Ionospheric scatter on 6 m											
JT6M	30	44-FSK	*	N/A	512	21.5	947	1/3	1.5*	-10	
ISCAT	30	64-FSK	78	RS(63,13)	512	23.4	1500	1x12	3.2	-15	
			48	RS(63,8)	512	23.4	1500	1x12	3.2	-16	
			30	RS(63,5)	512	23.4	1500	1x12	3.2	-18	
			2	N/A	512	23.4	1500	N/A	0.1	-20	

* Messages in FSK441 and JT6M have no fixed length or structure.

Appendix C (continued)

Mode	T/R	Mod	Bits	FEC	Nsps	Baud	BW	Sync	TxT	S/N	Sub-modes
3. EME at VHF and UHF, QRP at HF											
JT65	60	65-FSK	72	RS(63,12)	4096	2.692	178	63x1	46.8	-24	ABC
			2	N/A	16384	0.673	178	N/A	46.8	-32	
JT64	60	64-FSK	78	RS(63,13)	6480	1.852	119	3x8	47.0	-26.2	ABCD?
			48	RS(63,8)	6480	1.852	119	3x8	47.0	-27.6	30 s?
			30	RS(63,5)	6480	1.852	119	3x8	47.0	-28.8	
			2	N/A	16384	0.732	119	N/A	46.4	-32	
4. Alternative Modes for EME, Microwaves, QRP at HF											
JT4A	60	4-FSK	72	K=32, r=1/2	2520	4.375	17.5	1x206	47.1	-23	ABCDEFG
JT4G	(widest sub-mode)						315	1x206	47.1	-17	
JT8	60	8-FSK	78	K=16, r=1/4	4096	2.930	23	2x10	49.0	-24	ABC...
			48	K=15, r=1/6	4096	2.930	23	2x10	49.0	-25	?
			30	K=16, r=1/8	4096	2.930	23	2x10	49.0	-26	?
			2	N/A	16384	0.732	23	N/A	49.2	-30	?

Appendix D: Linux Installation

First make sure you have installed the following Linux packages (can be done with Synaptic Package Manager):

python-numpy
python-tk
python-imaging-tk
libfftw3-3

Download the current release of WSJT8 from the WSJT home page. The filename should be something like wsjt8_8.0_r1940_i386.deb, where "1940" is the current release number.

Move wsjtyyrxxxx_i386.deb into the directory (normally your home directory, e.g., /home/yourname) where you want WSJT to reside. Open a command-line terminal and execute the following command:

```
$ sudo dpkg --instdir=. -i wsjtyyrxxxx_i386.deb
```

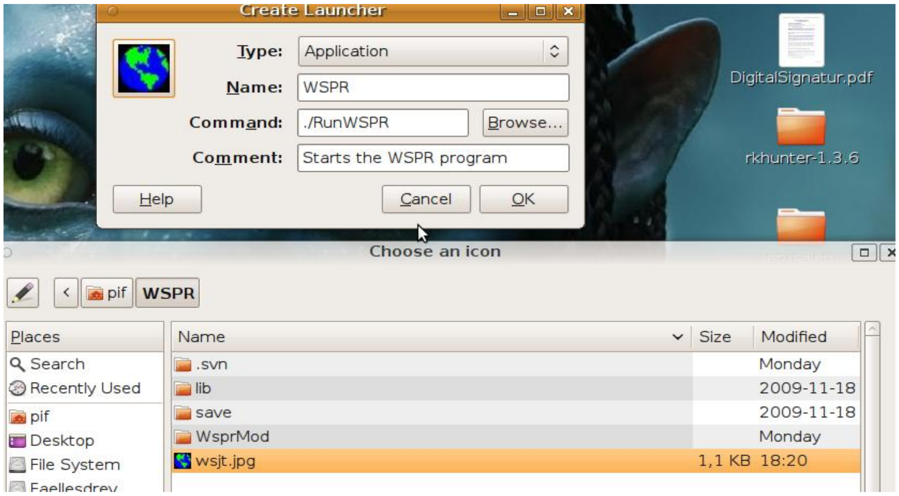
To start WSJT, enter these commands:

```
$ cd WSJT
$ ./wsjt
```

Alternatively, you can create a Desktop-icon for program startup. First create a file in your home folder named "RunWSJT" with this content:

```
#!/bin/sh
cd ~/WSJT
./wsjt
```

Make sure the file is executeable (Filemanager → Properties → Permissions). Then right-click on the Desktop → Create Launcher and fill out the form as appropriate:



Double-Click the the icon and select your preferred Desktop icon – I use wsjt.jpg (converted from wsjt.ico). Press OK and you're done, you can now launch WSJT by double-clicking the desktop icon.

This procedure has been validated on Ubuntu 9.04, 9.10 and 10.04, and should work for all Debian derivatives.