Work the World with *WSJT-X*, Part 2: Codes, Modes, and Cooperative Software Development

Here's how the weak-signal digital protocols in *WSJT-X* work, and an overview of how their software is developed.

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

Part 1 of this two-part article covered topics that highlight the capabilities of weak-signal communication program *WSJT-X*.[1] This software package provides tools for a wide range of Amateur Radio activities including low-power DXing, meteor scatter, moonbounce, and precise frequency measurement — all of them possible with relatively modest station equipment. Based on modern communication and information theory, the *WSJT-X* protocols and software can boost your signal's effective reach by the rough equivalent of 10 – 15 dB of added signal strength.

This concluding Part 2 outlines some digital communication theory fundamentals, including examples to make the discussion accessible to most amateurs. We compare the eight weak-signal protocols in *WSJT-X* and explain how their impressive performance is achieved. Finally, we describe the tools and informal cooperative practices used for creating the *WSJT-X* software. We think it's important that dedicated enthusiasts devote their algorithmic and programming skills and interests to the good of the hobby. We have found that many other hams would like to have a deeper understanding of how these weak-signal protocols work, and how they were developed. We hope this article satisfies that desire.

Digital communication conveys digital information from an originating source to one or more destinations.

Digital Communication Fundamentals

Digital communication conveys digital information from an originating source to one or more destinations. Here the digital information is modulated onto a carrier and transferred over a radio channel. The basic unit of transmitted data is a *channel symbol*. The symbols represent numbers, in turn comprised of bits. The modulator may transmit m information bits in each symbol, using 2^m different waveforms to represent symbol values from 0 up to $2^m - 1$. The different waveforms might have distinct amplitudes, phases, frequencies, or shapes. The *WSJT-X* waveforms are made of sinusoids with constant amplitude. The MSK144 protocol uses *Offset Quadrature Phase-Shift Keying* (OQPSK) with waveforms shaped to maintain a constant envelope. All other modes use frequency-shift keying (FSK), with a different tone frequency to represent each allowed symbol value. Binary modulation ($m = 1$) implies transmitting one bit at a time. Modulation schemes with larger m are used to an advantage in all but one of the *WSJT-X* modes.

Important benefits can be gained by adding controlled redundancy to a digital message so that transmission errors can be recognized and corrected. Simple repetition of each symbol is a trivial form of redundancy. But much more powerful redundancy can be arranged by mapping each sequence of k message symbols in a controlled way into a unique and longer sequence of n symbols called a codeword. This technique is called forward error correction (FEC). The *WSJT-X* protocols use block codes in which the values of n and k are fixed, and labeled as (n,k) codes. An integer parameter q can be used to define the range of available symbol values for a code, analogous to the m we used for modulation schemes. Parameter $Q = 2^q$ is then referred to as the *alphabet size* of the code. The code symbol values range from 0 up to $Q - 1$, and each codeword conveys kq message bits. The amount of redundancy is characterized by the ratio n/k, and its reciprocal k/n is the *code rate*. The mathematics underlying design of such k-to-n mapping schemes and their corresponding n-to-k reverse transformations forms a major branch of modern communication theory.

Reception of transmitted symbols requires accurate synchronization of time and frequency between transmitting and receiving stations. To make this possible with typical amateur station equipment, each *WSJT-X* protocol includes a unique synchronizing pattern: a sequence of known symbols interspersed with those carrying message information. The software demodulation algorithm starts by looking for the known pattern, thereby determining any frequency and time offset, as well as the locations of boundaries between received symbols.

Specifically, the JT65 mode uses a (63,12) code with $q = 6$ and thus $Q = 2^q = 64$. Its code rate is $k/n = 0.19$, and its modulation uses $m = 6$ and thus $2^m = 64$ tone frequency-shift keying, with one additional tone used for synchronization.

Let's divide the process of transmission and reception into a sequence of independent steps (see Figure 1). The steps correspond roughly to identifiable blocks of the *WSJT-X* source code. Steps 1 – 5 take place at the transmitting station, and steps 6 – 9 at the receiving end.

1) Generate a message.

2) Compress message to k symbols of q bits per symbol.

3) Add error-correcting redundancy to produce codeword of n symbols.

4) Add synchronizing pattern and modulate onto a carrier.

5) Transmit modulated waveform over a radio channel.

6) Receive, synchronize, and demodulate to yield n symbols, some of which might be in error.

7) Decode n received symbols to recover k error-free message symbols.

8) Decompress k symbols to recover original message in human-readable form.

9) Deliver message to receiving user.

The most crucial are steps 3 and 7. Step 7 likely requires the most computational resources.

When developing a protocol, we want to choose an efficient code that maximizes the probability of recovering transmitted messages even when the received codeword is corrupted. It's also important to consider likely types of fading, Doppler spread, and interference that may occur on the targeted propagation paths. We need an efficient decoding algorithm that can be executed in reasonable computing time and will ensure that false decodes are rare.

The *WSJT-X* Protocols

Message Structure

Steps 2 and 8 involve lossless compression and decompression of data. This process is called *source encoding* the message. *WSJT-X* protocols JT4, JT9, JT65, QRA64, and MSK144 all use structured messages that source-encode human-readable information for basic contacts into packets of exactly $kq = 72$ bits. The packets contain two 28-bit fields normally used for call signs and a 15-bit field for a grid locator, signal report, acknowledgment, or 73. One additional bit is used to flag packets encoding arbitrary alphanumeric text, up to 13 characters. Special cases allow efficient encoding of other information, such as add-on call sign prefixes (ZA/KA2ABC) or suffixes (G8XYZ/P). The aim is to compress the most common messages used for minimal contacts into fixed-length 72-bit packets. FT8 uses similar source encoding, but provides additional flexibility and room for growth by adding three extra bits that can be used to define up to 14 enhanced message types.

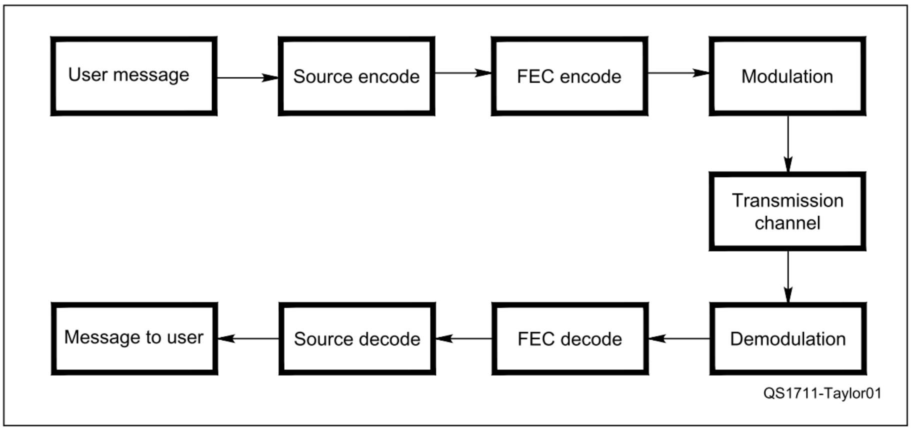


Figure 1 — Block diagram showing steps in a typical digital communication system.

Why 28 bits for a call sign, and 15 for a grid locator? A standard amateur call sign consists of a one- or two-character prefix, at least one of which must be a letter, followed by a digit and a suffix of one to three letters. Within these rules, the number of possible call signs equals 37 × 36 × 10 × 27 × 27 × 27, or somewhat over 262 million. The numbers 27 and 37 arise because in the first and last three positions a character may be absent, or a letter, or perhaps a digit. Because 2^{28} is greater than 268 million, 28 bits are enough. Similarly, the number of four-digit Maidenhead grid locators on Earth is 180 × 180 = 32,400, which is less than 2^{15}, so a grid locator can be encoded uniquely with 15 bits.

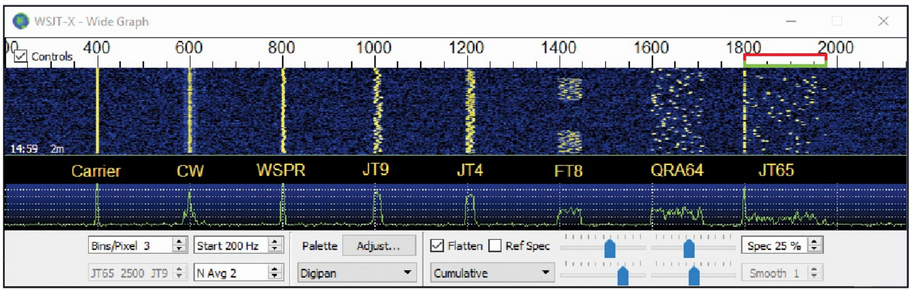
More than six million of the possible 28-bit values are not needed for standard call signs. A few of these slots have been assigned to special message components, such as CQ, DE, and QRZ. CQ may be followed by three digits to indicate a desired call-back frequency. In the meteor-scatter mode MSK144, if KA2ABC transmits on say 50.260, and sends the message "CQ 290 KA2ABC FN20," it means that he or she will listen on 50.290 and respond there to any replies. A numerical signal report of the form ±xx or R±xx can be sent in place of a grid locator. As originally defined in JT65 mode, the numerical signal report values “xx” lie in the range –30 to –01 dB. Recent program versions accommodate reports between –50 and +49 dB for all modes except JT65. A country prefix or portable suffix may be attached to one of the call signs. When this compound call sign feature is used, the additional information is sent in place of the grid locator, or by using some of the six million available slots mentioned above.

Our compression algorithm supports messages starting with CQ AA through CQ ZZ. Such messages are encoded by sending the pseudo-call signs E9AA through E9ZZ in place of the first call sign of a standard message. Upon reception, these calls are converted back to the form CQ AA through CQ ZZ. This allows users to send directed CQ messages, such as CQ DX, CQ EU, or CQ VT.

We want to choose an efficient code that maximizes the probability of recovering transmitted messages even when the received codeword is corrupted.

Error-Correcting Codes

Different codes, modulation schemes, and synchronizing patterns have been adopted for each protocol in *WSJT-X*. The goal has been to optimize each mode’s effectiveness for a particular type of propagation. To some extent, the final code choices also reflect our own incomplete but growing familiarity with historical developments in communication theory. JT65 uses a *Reed-Solomon* code, and JT4, JT9, and WSPR all use a robust *convolutional* code, first implemented for ham radio use by Phil Karn, KA9Q.[2,3] These are among the best-known types of error-correcting codes, and they have been studied thoroughly for over half a century. Our latest modes use state-of-the-art codes that are close to the forefront of this research field. MSK144 and FT8 use *low-density parity check* (LDPC) codes and QRA64 a *Q-ary repeat-accumulate* (QRA) code, a particular type of non-binary LDPC code. Full technical specifications for each mode can be found in the *WSJT-X User Guide* and our openly available source code.[4,5]



Bins/Pixel 3
Start 200 Hz
JT65 2500 JT9
N Avg 2
Digipan
☑ Flatten ☐ Ref Spec
Cumulative
Spec 25 %
Smooth 1

Figure 2 — Simulated signals for an unmodulated carrier, a 25 WPM CW signal, and the *WSJT-X* slow modes WSPR, JT9, JT4, FT8, QRA64A, and JT65. The slow modes are shown in their “A” submode, in increasing order of occupied bandwidth. All signals have S/N of –10 dB in a 2,500 Hz reference bandwidth. The vertical extent of the waterfall corresponds to 50 seconds. Two successive FT8 transmissions are shown.

Protocol Details for Slow Modes

Figure 2 shows an example of each slow mode on the *WSJT-X* waterfall display. This also includes an unmodulated carrier and a 25 WPM CW signal. The signals were generated with a key-down signal-to-noise ratio of –10 dB in a 2,500 Hz reference bandwidth. Among the *WSJT-X* modes, WSPR has the narrowest occupied-bandwidth, 5.9 Hz, and JT65 has the widest at 177.6 Hz. JT4, JT9, JT65, and QRA64 use 1-minute timed sequences of transmission and reception, synchronized approximately with UTC. FT8 uses 15-second sequences, and WSPR uses 2 minutes.

Some design parameters of the slow modes are summarized in Table 1. The type of FEC is denoted by LDPC for low-density parity check, C for convolutional, RS for Reed-Solomon, and QRA for Q-ary repeat-accumulate. Chosen keying rates make the length of a transmission approximately 13 seconds for FT8 and 48 seconds for the other modes. For JT4, JT9, JT65, and QRA64, this leaves plenty of time at the end of a transmission for a message to be decoded and the receiving operator to decide how to reply, before the start of the next minute. With 15-second T/R sequences, FT8 is much more tightly constrained. An optional auto-sequencing feature allows the software to generate suitable messages in response to received information. Exact values for keying rates were chosen so that at 12,000 samples per second the number of digital samples per channel symbol is an integer with no prime factor greater than 7. This advantageous choice makes some of the digital signal processing algorithms more efficient.

In the following, modes are described in chronological order of their development: JT65 was first introduced in 2003, JT4 in 2007, WSPR in 2008, JT9 in 2012, QRA64 in 2016, and FT8 in 2017.

JT65 — A detailed description of the JT65 protocol was published in *QEX*.[6] Half of its channel symbols are used for synchronization, using a pseudo-

Table 1: Parameters of the Slow *WSJT-X* Protocols
Bandwidths (BW) are for the narrowest submodes. S/N threshold is referenced to a 2,500 Hz bandwidth at a 50% probability for decoding of an unfading signal.

Mode	FEC type (n,k)	q m	Modulation	Keying rate, baud	BW, Hz	Sync energy, fraction	TX duration, s	S/N threshold, dB
FT8	LDPC(174,87)	1 3	8-FSK	6.250	50.0	0.27	12.6	–20
JT4	C(206,72)	1 2	4-FSK	4.375	17.5	0.50	47.1	–23
JT9	C(206,72)	1 3#	9-FSK	1.736	15.6	0.19	49.0	–27
JT65	RS(63,12)	6 6#	65-FSK	2.692	177.6	0.50	46.8	–25
QRA64	QRA(63,12)	6 6	64-FSK	1.736	111.1	0.25	48.4	–26
WSPR	C(162,50)	1 2	4-FSK	1.465	5.9	0.50	110.6	–28

#Modulation includes one additional tone used for synchronization.

random pattern at the lowest tone frequency. The other symbols carry encoded information using $2^m = 64$ different tones. Special features (used only for Earth-Moon-Earth — EME) can convey the EME-style "OOO" signal report and short messages interpreted as RO, RRR, and 73. EME submodes JT65B and JT65C use tone spacings two and four times larger than JT65A. JT65 has become very popular for low-power DXing at MF and HF, as well as for EME on VHF and higher bands.

JT4 — Each channel symbol carries one information bit (the most significant bit) and one synchronizing bit. Thus, 50% of the transmitted energy is devoted to synchronization. Submodes JT4A through JT4G have tone spacings at increasing multiples 1, 2, 4, 9, 18, 36, and 72 times the keying rate of 4.375 baud. The wider submodes are useful on propagation paths with large Doppler spread. For example, JT4F is frequently used for EME communication on the 10 GHz band.

WSPR — Designed as a propagation probe rather than for making two-way contacts, it differs from other *WSJT-X* slow modes by using message lengths $k = 50$ bits and 2-minute T/R sequences. Message packets normally include a 28-bit call sign, a 15-bit grid locator, and seven bits to convey transmitter power in dBm. Alternative formats can convey a compound call sign and/or a six-digit grid locator, using a two-transmission sequence. Typical WSPR usage was described in *QST*.[7]

JT9 — Eight tone frequencies are used to convey message information, and one additional tone is used for synchronization. The slow submodes JT9A-H have tone spacings at multiples 1, 2, 4, 8, 16, 32, and 64 times the 1.736-baud keying rate. JT9A (often called simply JT9) uses less than 10% of the bandwidth of JT65, and for steady, undistorted signals is about 2 dB more sensitive than JT65. For these reasons, JT9 is popular for low-power DXing on crowded HF bands.

QRA64 — An experimental mode intended for EME and other extreme weak-signal paths. Its internal code[8] was designed by Nico Palermo, IV3NWV. Synchronization is accomplished by using three 7 × 7 Costas arrays.[9] Submodes QRA64A-E use tone spacings 1, 2, 4, 8, and 16 times the 1.736-baud keying rate. Early tests have shown QRA64A to be very effective for weak-signal work at MF and HF, and for EME on the VHF and UHF bands. The wider submodes QRA64C-E work extremely well for EME on microwave bands up to 24 GHz.

FT8 — Designed especially for propagation conditions, such as multi-hop

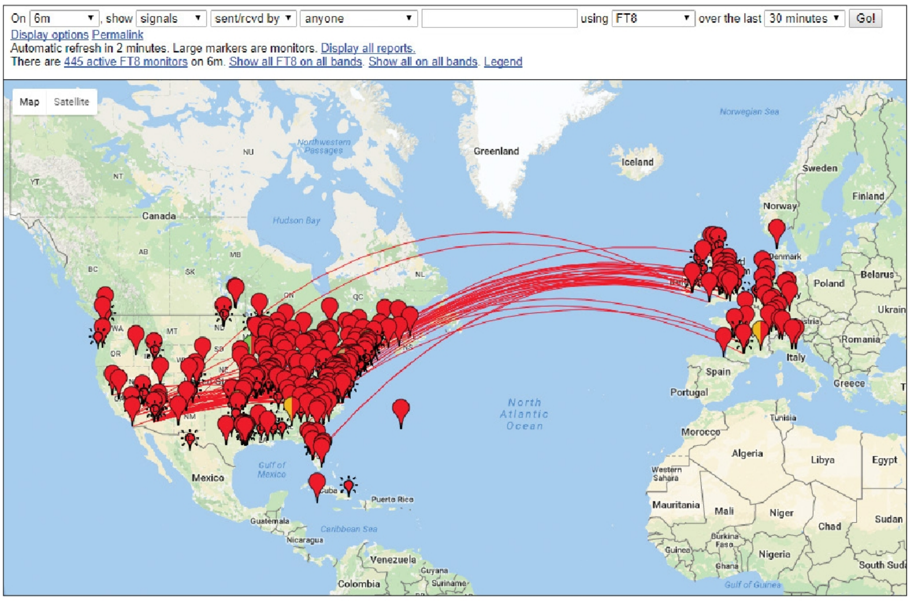

On 6m ▾, show signals ▾ sent/rcvd by ▾ anyone ▾ using FT8 ▾ over the last 30 minutes ▾ Go!
Display options Permalink
Automatic refresh in 2 minutes. Large markers are monitors. Display all reports.
There are 445 active FT8 monitors on 6m. Show all FT8 on all bands. Show all on all bands. Legend

Figure 3 — Screenshot of the *PSK Reporter* website using decoded FT8 transmissions to show summertime multi-hop sporadic-E paths on 50 MHz around 1500 UTC on July 28, 2017.

Table 2: Parameters of the Fast *WSJT-X* Protocols
MSK144-Sh is the optional short-message format in the MSK144 protocol.

Mode	FEC type (n,k)	q m	Mod	Keying rate, baud	BW, Hz	Sync energy	Message duration, s
ISCAT-A	—	—	42-FSK	21.5	905	0.17	1.176
ISCAT-B	—	—	42-FSK	43.1	1,809	0.17	0.588
JT9E	C(206,72)	1 3#	9-FSK	25	225	0.19	3.400
JT9F	C(206,72)	1 3#	9-FSK	50	450	0.19	1.700
JT9G	C(206,72)	1 3#	9-FSK	100	900	0.19	0.850
JT9H	C(206,72)	1 3#	9-FSK	200	1,800	0.19	0.425
MSK144	LDPC(128,80)	1 1	OQPSK	2,000	2,400	0.11	0.072
MSK144-Sh	LDPC(32,16)	1 1	OQPSK	2,000	2,400	0.20	0.020

#Modulation includes one additional tone used for synchronization.

sporadic E at 50 MHz — circumstances where signals are weak and fading, openings short, and quick completion of reliable, confirmable contacts is particularly desirable. Timed sequences for transmission and reception are 15 seconds rather than 1 minute. Shorter transmissions mean that FT8 is about 6 dB less sensitive (for steady signals) than it would be if its duration was commensurate with the other slow modes. Message packets include 75 information bits and a 12-bit cyclic redundancy check (CRC) that helps to ensure a very low false decode rate. Modulation uses eight-tone FSK at 6.25 baud, and synchronization uses three 7 × 7 Costas arrays. As suggested by a snapshot from the *PSK Reporter*[10] website (see Figure 3), FT8 has become popular very quickly after its introduction in early summer of 2017.

Protocol Details for Fast Modes

The *WSJT-X* fast modes take advantage of brief propagation enhancements that bring a signal up to useful levels for a very short time. Keying rates and occupied bandwidths are much larger than for the slow modes, because we want full messages to be conveyed in a very short time. Table 2 lists the essential parameter values for these modes. The last column gives the time required to transmit a message once. In these modes, the transmitted information is repeated for the full duration of a T/R sequence.

ISCAT — Messages are free-form, up to 28 characters in length. The protocol uses no FEC other than repetition.

The* WSJT-X *fast modes take advantage of brief propagation enhancements that bring a signal up to useful levels for a very short time.

ISCAT has proven especially useful for making aircraft-scatter contacts on the microwave bands.

JT9 Fast — Submodes JT9E-H differ from their slow counterparts by using much faster keying rates. Otherwise the coding, modulation, and synchronization schemes are the same as for the slow JT9 modes. JT9 fast modes have proven useful for such propagation types as ionospheric scatter and weak double-hop sporadic E on the 6-meter band.

MSK144 — FEC is implemented by augmenting the 72 message bits with an eight-bit CRC calculated from the message bits. The resulting 80-bit augmented message is mapped to a 128-bit codeword using a (128, 80) LDPC code designed by K9AN specifically for this purpose. Two eight-bit synchronizing sequences are added to make a message frame 144 bits long. Modulation is Offset Quadrature Phase-Shift Keying (OQPSK) at 2,000 baud, so the frame duration is 72 milliseconds. Compared to FSK441, the mode widely used for digital meteor-scatter since its introduction[11] in 2001, MSK144 has the advantages of strong error correction, an effective character transmission rate about 1.7 times faster, and significantly better sensitivity. MSK144 has rapidly become the dominant mode for amateur meteor-scatter contacts.

Decoders and Sensitivities

Together with additional details published in the *WSJT-X User Guide* and the open source code, Tables 1 and 2 and the preceding paragraphs define the various protocols supported in *WSJT-X*. For these protocols to be useful, each one needs an efficient decoder. The algorithms we have implemented use *soft-decision* decoding, and to the best of our knowledge, they are the most sensitive practical algorithm for each code. For JT4, JT9, and WSPR, we use the Fano algorithm, as implemented by KA9Q (see Note 3), and for JT65 we use the Franke-Taylor algorithm, recently published in *QEX*.[12] There is a full description of the QRA64 decoder, written by IV3NWV (see Note 8), and we expect to publish details of the MSK144 and FT8 decoders soon.

Usage patterns of FT8, JT9, JT65, and

WSPR makes it advantageous for the decoders to focus not just on a single frequency, but on a frequency range covering several kilohertz. Our decoders are organized to scan a range of frequencies up to 5 kHz, if the receiving hardware supports it, finding all signals in the specified mode and decoding and displaying the results. For FT8, JT65, and WSPR, the present decoders go one step further, taking advantage of the fact that when a signal with strong FEC has been decoded, we can calculate its transmitted waveform exactly. An amplitude-scaled version of that waveform can be subtracted from the received data and the decoder executed on the remainder, thereby possibly decoding weaker, previously hidden signals. This approach has proved very effective. In these modes, *WSJT-X* frequently decodes weak signals lying within 1 or 2 Hz of much stronger ones.

The various *WSJT-X* modes have better sensitivity than traditional modes such as CW for three main reasons: They use efficient modulation schemes tailored to the targeted types of propagation; they use detection bandwidths matched to the protocol keying rate, and they benefit from *coding gain* provided by each specific error-correcting code. Table 1 shows detection bandwidths for the slow modes range from about 1.5 Hz – 6 Hz. Noise power is proportional to bandwidth, so each of the slow modes has an advantage of 10 dB or more when compared to the typical 50 Hz "ear-and-brain" bandwidth of the very best weak-signal human CW operators.

The MSK144 decoder cannot use such narrow bandwidths because the signal is roughly 2.4 kHz wide. However, it can use *coherent* detection. Meteor-scatter signals generally maintain signal coherence over the duration of a ping. Our MSK144 decoder measures a received signal frequency and phase with enough accuracy to maintain coherence over half a dozen or more of the protocol's 72 millisecond frames. So, the out-of-phase noise power can be rejected and we gain 3 dB over non-coherent detection for single-frame decodes, and up to 7 dB for seven-frame averages.

Summary

Additional details of the software engineering and development effort, as well as a description of the computing hardware and platforms, is available on the *QST*-in-Depth web page at **www.arrl.org/qst-in-depth**.

We hope that our description of the capabilities of *WSJT-X* and its development process will inspire others to join in and contribute to future developments in digital communication techniques for Amateur Radio. We ourselves have many ideas that have not yet reached fruition, but may do so in the future.

Many people have contributed to the development and success of *WSJT-X*. We particularly wish to thank Greg Beam, KI7MT, whose *JTSDK* software development kit has helped many *WSJT-X* users build the program for themselves, from the source code; and Dr. C.W. Suckling, G3WDG; Roger G. Sturtevant, VE1SKY; Rex Moncur, VK7MO, and Roger B. Rehr, W3SZ, for comments that helped us to improve an early draft of this article.

Notes

[1] Joe Taylor, K1JT, Steve Franke, K9AN, and Bill Somerville, G4WJS, "Work the World with *WSJT-X*, Part 1," *QST* Oct. 2017, pp. 30 – 36.

[2] For more details about convolutional codes, see **https://en.wikipedia.org/wiki/Convolutional_code#History**.

[3] Phil Karn, KA9Q, **www.ka9q.net/papers/cnc_coding.html**.

[4] **www.physics.princeton.edu/pulsar/K1JT/wsjtx-doc/wsjtx-main-1.7.1-devel.html**

[5] **physics.princeton.edu/pulsar/k1jt**

[6] Joe Taylor, K1JT, "The JT65 Communications Protocol," *QEX*, Sept./Oct. 2005, p. 3.

[7] Joe Taylor, K1JT, and Bruce Walker, W1BW, "WSPRing Around the World," *QST* Nov. 2010, p. 30.

[8] Nico Palermo, IV3NWV, "Q-ary Repeat-Accumulate Codes for Weak Signals Communications," **microtelecom.it/qracodes/QRACodes-Rev10.pdf**.

[9] **https://en.wikipedia.org/wiki/Costas_array**

[10] Philip Gladstone, N1DQ, **https://pskreporter.info/pskmap**.

[11] Joe Taylor, K1JT, "*WSJT*: New Software for VHF Meteor-Scatter Communication," *QST*, Dec. 2001, pp. 36 – 41.

[12] Steven J. Franke, K9AN, and Joseph H. Taylor, K1JT, "Open Source Soft-Decision Decoder for the JT65 (63,12) Reed-Solomon Code," *QEX*, May/Jun. 2016, pp. 8 – 17.

Joe Taylor was first licensed as KN2ITP in 1954, and has since held call signs K2ITP, WA1LXQ, W1HFV, VK2BJX, and K1JT. He was Professor of Astronomy at the University of Massachusetts from 1969 to 1981, and since then Professor of Physics at Princeton University, serving there also as Dean of the Faculty for 6 years. He was awarded the Nobel Prize in Physics in 1993 for discovery of the first orbiting pulsar, leading to observations that established the existence of gravitational waves. After retirement, he has been busy developing and enhancing digital protocols for weak-signal communication by Amateur Radio, including JT65 and WSPR. He chases DX from 160 meters through the microwave bands. You can reach Joe at **joe@princeton.edu**.

Steve Franke, K9AN, holds an Amateur Extra class license. He was first licensed in 1971 and has previously held call signs WN9IIQ and WB9IIQ. An early and abiding fascination with radio science led to his current position as Professor of Electrical and Computer Engineering at the University of Illinois in Urbana-Champaign. Steve is a member of ARRL and a Fellow of the IEEE. You can reach Steve at **s.j.franke@icloud.com**.

Bill Somerville, G4WJS, read Chemistry at the University of Bristol in the UK, graduating in 1978 with a Bachelor of Science degree. Since graduating, he has worked in computer software and hardware in a variety of industries including defense, software vendors, and financial services, more recently as a freelance consultant providing systems programming and related services to mid- to large-size software tool vendors. Bill has been an active radio amateur since 1981, holding the calls G6IJU and G4WJS. He enjoys HF and VHF bands, contest operating, and DX chasing using CW, phone, and data modes. He is a keen walker and swimmer. Bill participates in open-source software projects for Amateur Radio, with *WSJT-X* being the largest contribution. Bill is happy to provide input, drawing on his extensive software engineering experience, to help those with other important domain skills provide robust, professional standard, software-based Amateur Radio communication tools. You can reach Bill at **bill.8@classdesign.com**.

For updates to this article, see the *QST* Feedback page at www.arrl.org/feedback.