FSQ Explained

A description of the FSQ Codec, FSQCall Directed Messaging, and related features.

Murray Greenman ZL1BPU

14 November 2017
Document version 0.03
Reference version FSQCall V0.42

Contents

Introduction

FSQ stands for ***Fast Simple QSO***, which is about as succinct a definition of the system as is possible. FSQ is a radio modem, a new way of conversing by keyboard, and a simple communication technique that enables a range of other add-on features.

Application

FSQ was designed for NVIS[1] propagation, but intended also to be useful on slightly longer distance HF single hop paths and on VHF FM. The main design challenge was to achieve reasonable performance under NVIS conditions, where there is considerable fading, marked multi-path reception, some Doppler shift, and no shortage of pulse interference, either lightning or SSB 'buck-shot'.

The designers have considerable experience in this medium, so chose a codec[2] that uses MFSK,[3] a very low symbol[4] rate, and many tones, to allow a reasonable data throughput and good signal-to-noise ratio. These details will become apparent as the mode is described in detail.

[1] ***Near Vertical Incidence Signals***, short distance, single-hop ionospheric propagation, typically on low HF bands, 3 – 10 MHz.
[2] ***Codec*** Means the technology of coding and decoding used with the modem. The modem is the base-band device which turns data into tones and vice-versa (means modulator-demodulator).
[3] ***Multi-Frequency Shift Keying***
[4] The ***symbol*** is the smallest signalling entity (same phase, frequency, amplitude) used by a modem.

Further, having grown tired of digital transmission modes which suffered from long latencies (slow turn-around between overs), and bad operating habits such as very long transmissions and one-sided QSOs, it was decided that the new mode should depart from the conventional QSO model, and introduce hams to 'chat operation', a concept not unlike cell-phone text-messaging or Skype™ chat.

So the FSQ design is a low HF band, fast and slick QSO mode based primarily on sending single sentences. It is ideal for chatting and conversation among small nets of friends. In most applications no error correction is involved, or needed, and the mode has very quickly proved to be popular for all of the reasons just given.

FSQCall

In early testing, the FSQ codec proved to be so robust under the intended conditions that it made sense to use the inherent sentence-based nature of the design to introduce a measure of directed messaging – like a simple 'Selcall' system, hence the name. The idea grew to become a very effective directed messaging system, with a range of automatic functions.

FSQCall is far more than a Selcall system, since it is capable of much more than 'ringing a bell' at the recipient station. However, it's not a full ALE[5] system, as it is essentially a design which operates on a single frequency at a time, with no scanning. It is however very good at ALE-like '***order wire***' functions (network management, passing commands, schedule information and requests for contact using other media), provided by such systems.

By design, FSQCall is much simpler to deploy than most Selcall and ALE systems, which require type-approved, dedicated radios with special functions and pre-programmed user addresses, require pre-defined operating frequencies and pre-registered and *trained* operators. In contrast, FSQCall is an ***ad hoc*** design – anyone with suitable equipment can operate, on any frequency, at any time, with any callsign. The user callsign acts as the station registration, and is used to facilitate directed commands.

In addition, FSQCall is unique in providing a simple automatic relay facility, along with automated file transmission and storage – and these functions has proved to be a source of great excitement to the Amateur community.

The FSQCall design uses only manual commands (called ***directions***), but in most cases the responses are automatic. The exception is ***directed chat***, where chat text is directed to just one station, or a group of stations, and where replies are manually generated.

Added Functions

With the addition of directed command capabilities, a number of other applications suggest themselves, such as the transmission and reception of error corrected files (achieved by off-line accessories), maintenance of accurate logs, including a channel activity log, and directed analogue image transfer (several formats). Many other minor commands have been implemented and will be described under the **Commands** heading.

[5] ***Automatic Link Establishment*** Largely based on US MIL STD 188-141A and FED-1045.

FSQ Coding and Modulation

FSQ uses 33-MFSK, and in addition the data is encoded using IFK+.[6] IFK+ offers improved performance under NVIS conditions, because the rotation significantly reduces the risk of adjacent symbols causing inter-symbol interference.

Symbol Table

FSQ is designed to use 32 tone differences between these 33 tones (we talk about differences because the IFK+ modulation is differential); so it is possible to allocate 29 different individual differences directly to the most frequently used characters. These characters are all lower case, a-z, plus the most common punctuation symbols: space, full stop (period), and new line (CR/LF). The three 'spare' differences are used to define further code tables.

The total alphabet includes upper case and a reasonable range of symbols, yielding an available alphabet of 104 ASCII characters. The three extra differences are allocated to three additional code tables, which are sent as two sequential tone differences. These are characterized by an ***initial*** difference, describing the character, and a ***continuation*** difference, defining the code table.

When the receiver sees a tone difference in the range 0 – 29, followed by another in the range 0 – 29, it recognises a single-tone character (lower case etc). If the tone difference is followed by a difference in the range 30 – 31, it uses the second difference to decide which code table to select the character from, and the first difference to choose the character from that table. The encoding process is the reverse of this.

CHAR	ASCII	VAR	CHAR	ASCII	VAR	CHAR	ASCII	VAR	CHAR	ASCII	VAR
SPACE	32	0	@	64	0,29	`	96	9,31	CRLF	13/10	28
!	33	11,30	A	65	1,29	a	97	1	IDLE	0	28,30
"	34	12,30	B	66	2,29	b	98	2	±	241	10,31
#	35	13,30	C	67	3,29	c	99	3	÷	246	11,31
$	36	14,30	D	68	4,29	d	100	4	°	248	12,31
%	37	15,30	E	69	5,29	e	101	5	x	158	13,31
&	38	16,30	F	70	6,29	f	102	6	£	156	14,31
'	39	17,30	G	71	7,29	g	103	7	BS	8	27,31
(	40	18,30	H	72	8,29	ĥ	104	8			
)	41	19,30	I	73	9,29	i	105	9			
*	42	20,30	J	74	10,29	j	106	10			
+	43	21,30	K	75	11,29	k	107	11			
,	44	27,29	L	76	12,29	l	108	12			
-	45	22,30	M	77	13,29	m	109	13			
.	46	27	N	78	14,29	n	110	14			
/	47	23,30	O	79	15,29	o	111	15			
0	48	10,30	P	80	16,29	p	112	16			
1	49	1,30	Q	81	17,29	q	113	17			
2	50	2,30	R	82	18,29	r	114	18			
3	51	3,30	S	83	19,29	s	115	19	WSQ Varicode V3.0		
4	52	4,30	T	84	20,29	t	116	20			
5	53	5,30	U	85	21,29	u	117	21			
6	54	6,30	V	86	22,29	v	118	22			
7	55	7,30	W	87	23,29	w	119	23	Copyright (C) Murray Greenman Dec 2013		
8	56	8,30	X	88	24,29	x	120	24			
9	57	9,30	Y	89	25,29	y	121	25			
:	58	24,30	Z	90	26,29	z	122	26			
;	59	25,30	[	91	1,31	{	123	6,31			
<	60	26,30	\	92	2,31	|	124	7,31			
=	61	0,31	]	93	3,31	}	125	8,31			
>	62	27,30	^	94	4,31	~	126	0,30			
?	63	28,29	_	95	5,31	DEL	127	28,31			

Fig. 1. The FSQ Varicode Table

[6] ***Incremental Frequency Keying***, differential MFSK with offset +1. Developed by the author.

The table (Figure 1) on the preceding page illustrates the alphabet in ASCII order. If you inspect Figure 1, you will see that the a – z entries are followed by their ASCII equivalents and then a single number, the ***varicode***[7], ranging from 0 to 28. The same is true of space, full stop (period), and new line (CR/LF).

The other characters have two numbers in the table: the ***initial*** difference, and the ***continuation*** difference. To send these characters, two transmission tones are sent, first the initial, then the continuation.

So there are in effect four code sets, each containing 29 characters, allowing a total possible alphabet of 116 characters. Some combinations remain unused (let's say *reserved*!), leaving 104 allocated characters.

The FSQ Varicode was first developed for a recent LF mode named WSQ (Weak Signal QSO), and is about as efficient as you can get as a coding method for MFSK applications. There are binary Varicodes with greater coding efficiency, however 33-tone 5-bit MFSK does not lend itself to the highest coding efficiencies (such as is achieved in modes such as PSK31). Its efficiency is achieved through sending much more information (five bits) on each symbol. As a result, FSQ will transmit standard lower-case text at six characters per second, which is close to 60 WPM, the same speed as RTTY (Radio Teletype), which operates at nearly eight times the signalling rate.

IFK+ Coding

IFK+ coding is the next step in coding the FSQ signal. Steve Olney VK2XV, who pioneered the technique, named it IFK or Incremental Frequency Keying. Con ZL2AFP and Murray ZL1BPU made the first use of Offset IFK (IFK+) in a QSO mode named DominoEX, 2009.[8]

IFK codes the data (the initial and continuation codes) as differences between two tones, rather than as an absolute tone. This is the MFSK equivalent of differential BPSK, as used in PSK31, etc. Previous MFSK modes, Piccolo, Coquelet and MFSK16[9] used direct MFSK modulation, assigning the code directly to a tone number. These modes suffered from more inter-symbol interference than was desirable, were prone to frequency drift, and were very difficult to tune.

With FSQ needing to use very narrow tone spacing, it was imperative to use an IFK process to counter these interference, drift and tuning issues. IFK, and especially IFK+, reduces inter-symbol interference by ensuring that the chance of the same or an adjacent tone being used for sequential symbols would be very remote.

Since the tones always change due to the differential action and the tone rotation of IFK+ (see later), there is also no opportunity for the tones to remain the same for two consecutive symbols, which allows the sync-less process (described later) to operate

[7] ***Varicoding*** is a process which codes more frequently used characters using fewer transmission symbols that lesser-used characters, thus enabling frequently used characters to be sent faster. While this technique was 'invented' and named Huffman Coding after David A. Huffman (1952), the same concept was included in Morse code (Samuel Morse, 1836) and later used by Amateurs in PSK31 and PACTOR.
[8] See http://www.qsl.net/zl1bpu/MFSK/DEX.htm
[9] The first Amateur MFSK mode, MFSK16, was designed by Murray Greenman ZL1BPU, 1999. See http://www.qsl.net/zl1bpu/MFSK/M16.htm

correctly. For example, without IFK+, repeated space characters could be seen as a single character.

Finally, since the coding is differential, any drift and frequency offset is cancelled out during decoding. IFK+ can handle frequency drift of about a third of the tone spacing per symbol, or in FSQ about 18 Hz drift per second at 6 baud. Frequency error tolerance is a matter of decoder design, but ± 50 Hz is tolerated easily by FSQ in the present configuration. The yellow lines on the tuning waterfall mark this tolerance.

IFK+, an acronym for Offset Incremental Frequency Keying, was suggested by Murray Greenman ZL1BPU (2009), and first used in DominoEX. With each symbol transmitted, as well as adding the difference from the alphabet-coding table to the previous tone number, an addition rotation offset is added. In the case of FSQ, this value is ONE.

Obviously you can't keep adding codes to tone numbers *ad infinitum*, or you'd soon run out of tones! The coding rule for FSQ is that when the tone number reaches or exceeds 33, 33 is subtracted to define the next tone number. Thus the tone numbers are bound within the range 0 – 32, yielding tone differences of 0 – 32, which agrees with the range required from the alphabet coding table.

Once the tone numbers are coded as described, they are multiplied by three, and sent to the modulator, which consists of an NCO[10] which must have a differential resolution of some integer fraction 'n' of 2.9296875 Hz, the same as the receiver FFT bin spacing, and the tones that are used are spaced three times this, 8.7890625 Hz, as a result of the fore-mentioned multiplication. The lowest tone used is about 1300 Hz, and the required tones are achieved by adding some multiple 'n' of the 3x tone number to the NCO value required for 1300 Hz offset, in order to achieve 8.7890625 Hz tone spacing.[11]

Ideal symbol rates for cleanest decoding using a conventional MFSK decoder should be integer ratios of 2.9296875, for example 5.859375 baud (we call it 6 baud) and 2.9296875 baud (3 baud). This rule has been adhered to in most modem designs to date. In practice the strong effect of the ionosphere on symbol timing, and the way sync-less decoding works, makes the distinction rather academic.

Only one tone is transmitted at a time.

In order for the first difference (first character) to be correctly decoded, a dummy tone needs to be transmitted first. This could be the lowest tone, but in fact any permitted tone will suffice since only the difference to the next tone is important. The software simply sends a space character to achieve this (see BOT Sequence, next chapter).

[10] ***Numerically Controlled Oscillator***, a software sine wave direct digital synthesizer.
[11] The modem centres the transmitted signal at 1500 Hz, to ensure that harmonics of the tones are always outside the SSB transmitter pass band.

Transmit Functions

The user types text into a keyboard buffer, and which is displayed in the transmit pane, and can be edited before transmission, using conventional editing functions. Transmission starts when an ENTER character is detected in the keyboard buffer, or the TX button is clicked by the mouse. It will also start as a result of one of the automatic processes, such as 'Sounding' or automated replies (see later).

When the transmission starts, the transmitter is commanded on (this can be via hardware connected to a serial comms port, USB port or via CAT).[12] Then a small delay is introduced to allow the hardware in the transmitter to become active, and for the antenna to change over, and then the first of the tones is applied. The delay required is typically about 150-200ms.

The characters in the keyboard buffer are then encoded as described, and transmitted at the currently set symbol rate. As they are transmitted, the characters are transferred to the receive pane as a record of what has been transmitted.

EOT Sequence

At the end of transmission in non-directed mode, the transmitter is simply turned off once the last useful character is sent. In FSQCall mode, however, a special character sequence is sent just before the transmission stops, to signal EOT.[13] The BS (backspace) character is not required in FSQ, so the corresponding character in the table has been reallocated as an EOT marker.

The purpose of the EOT sequence is to cause the Squelch to close quickly in FSQCall mode. The time constants differ in this mode (fast rise, slow decay), so that print will continue through fades without FSQCall closing and stopping print partway through a sentence. If the sentence transmission were to finish without the EOT, the squelch would close slowly, and junk would print unnecessarily. The EOT 'shorts out' the Squelch time constant, closing the Squelch instantly, preventing junk.

If the user were to type on the end of a sentence while it is being transmitted, this will interfere with the EOT action. The added text will appear on a new line, and some junk may follow. This behaviour is of course to be discouraged, but may still occur from time to time if the operator senses that an error in the sentence needs post-correction and is tempted to do so before the next sentence.

BOT Sequence

At the start of transmission in directed and non-directed mode, the BOT[14] sequence is sent. The space serves as a dummy symbol to allow the next tone difference to be measured. The LF character ensures that reception starts on a new line. In Directed mode, the LF also serves as one delimiter for the callsign search algorithm. The other delimiter is of course the colon that follows the callsign. This also locates the checksum.

---

[12] ***Computer Aided Transceiver***

[13] ***End of Transmission***. The sequence is <SP><SP><BS> (Varicodes 0,0,27,31).

[14] ***Beginning of Transmission***. The sequence is <SP><LF> (Varicodes 0,28)

Symbol Rate

As will be described in the receiver description to follow, the receiver is tolerant of the transmitter symbol rate over a wide range. The tone spacing and the spacing of the receiver FFT bins define the maximum practical rate, at 6 baud. Any arbitrary rate below that can be used, although in practice rates below 2 baud offer no advantage. The FSQ software offers speeds of 2, 3, 4.5 and 6 baud. Defining the number of NCO samples that are sent to the computer sound card sets these speeds, and the timing of transmission is thus controlled by the sound card.

The audio tones generated by the computer sound card are typically fed to the microphone input of an SSB transmitter, resulting in a series of short-duration, closely spaced carrier segments being transmitted. Because the NCO changes frequency without changing instantaneous phase (i.e. it is phase continuous) there is minimal switching noise, and the transmission is very narrow and clean. The tones are also constant amplitude.

The mode is also very effective when applied to a VHF FM transceiver. It has been found from experience that this technique is very sensitive, as the received signal can be decoded even when the signal is too weak to operate the receiver squelch. The advantage over VHF SSB is three-fold: improved sensitivity, better tolerance of drift and frequency error, and wider availability of equipment.

Transmission Bandwidth

The transmission consists of 33 tones spaced 8.7890625 Hz, resulting in a spread of tones of 290.0390625 Hz. Using the ITU-R SM.1138 assessment method, the signal easily meets 300 Hz useful bandwidth, at all signalling speeds, and its ITU Emission Designator is therefore 300HF1B.

Carrier Sense Multiple Access

Stations can send sentences at any time. In FSQCall (Directed) mode, many of the commands include automated responses, and in addition, Sounding messages may be sent automatically. So on a busy channel, the potential for transmissions to clash is high.

The software operates a Squelch system, which determines (by measuring signal to noise ratio) when the channel is free, and can also determine when to stop printing (therefore limiting junk on the screen). The Squelch is also used to determine when a station may transmit. In non-directed mode, the station will transmit after ENTER has been pressed, only if the Squelch is closed. If it has not closed, the operator must wait until it has and try again.

Some intelligent use must be made of the Squelch control setting, as conditions vary over time. A Squelch setting that allows the transmitter to operate under noisy conditions may also prevent weak signal reception when conditions improve.

In Directed mode (the default mode of FSQCall), a conventional network model CSMA protocol decides when the station may have access to the radio channel. This protocol defines three classes of message (chat, responses and soundings), and applies priorities to them. Priority is provided by means of delays after Squelch has closed. In addition, to limit clashes between stations wanting access for the same priority message, random delays are also added.

Priority	Delay	Message Type
1	Short	Chat sentences
2	Med	Automated responses to commands
3	Long	Sounding

Sounding

In FSQCall (Directed) mode, the user may elect to send regular 'Soundings'. These are very short messages with no trigger command and no message body. The message usually consists only of the callsign, colon and checksum of the FSQCall preamble.

These sentences do not print at any station, but appear in the Heard List and are logged in the Heard Log and Message Log. This important function allows all stations on the channel to know what other stations are there with minimum use of channel bandwidth.

In order to facilitate message relaying, stations may use the $ command to request a list of stations heard by another other station.

Sounding is sent with the lowest message priority. Sounding is also essentially random, as although the time between Sounds is fixed (typically 30 minutes) the sequence start time is set by user action, and essentially random.

FSQCall now include APRS-like position reports, where the sounding messages are replaced with specialized position reports. These also will not print, but will appear in logs and may be used by client APRS applications. The use of these messages combined with analysis of the Message Log should allow a future third-party web-based database to be used for propagation study (in the same manner as WSPR).

FSQ Demodulation and Decoding

FFT Operation

The receiver operates an overlapping FFT[15] sampled at a rate of 12000 Hz. This is determined by the sound card sampling clock rate of 12 kHz. No decimation process is used on the samples. The FFT is 4096 samples long, resulting in a bin spacing of 2.9296875 Hz. Each calculation interval, 1/16th of the samples (256 samples) is replaced by new samples, the 4096 sample buffer is windowed[16], and the FFT recalculated. This process takes place at 46.875Hz (12 kHz/256), so there are about 16 calculations per symbol at a nominal 3 baud rate.

The 2.9296875 Hz FFT bin spacing is of course one third of the transmitted tone spacing, so once the frequency differences have been determined by the decoder, they must be divided by three and rounded to an integer. Any error in the frequency difference, up to one FFT bin spacing, is thus removed. These errors can be caused by frequency drift in the transmitter or receiver, manual retuning of the receiver, and ionospheric Doppler shift.

The ability to handle received Doppler shift is limited by the narrow tone spacing, which limits the use of FSQ on higher bands under multi-hop conditions. Using wider shifts to counter this would involve a loss of sensitivity and reduced robustness, and although wider shifts were tested, their use was not justified in terms of the intended application.

The receiver operates the FFT as a software 'integrate and dump' detector. Noise activates all the bins of the FFT, but only the bin that receives the current tone will build up to indicate a significant amount of energy. At each sample point the highest value bin number is noted before the FFT is reloaded and recalculated.

Sync-less Symbol Discovery

After each FFT calculation, the bins are inspected to discover the largest signal (the bin with the most accumulated power). When three inspections in a row at FFT recalculation rate reveal that the same bin is at a maximum, it is recognised as a valid symbol. More than three inspection maxima from the same bin simply confirm that the same symbol is present.

As time passes, subsequent inspections show the power in a certain bin rises and falls in a raised cosine manner as the current symbol shifts from one bin to another (minimum every 8 samples at 6 baud). The symbol peak (relative to the noise in all other bins) may be of short duration, especially if the signal is weak.

When subsequent inspection shows that a different bin is now maximum for at least three inspections, a new determination starts. Thus there is no synchronization process in the conventional sense. Each symbol need only be present for at least three consecutive FFT determinations, and a new symbol is determined when the maximum bin changes. Peaks shorter than three 'hits' are ignored.

---

[15] ***Fast Fourier Transform***

[16] A Blackman-Harris raised-cosine window is used.

Despite conventional wisdom that the symbol determination process should be rather noise-prone, this has not proved to be the case. The receiver operates reliably with a signal-to-noise ratio of down to –15dB in 2.4 kHz bandwidth. This is some 10dB better than Morse code received by ear, and of course is also considerably faster. It is also better than most more conventional digital modes, largely because a very low symbol rate is used, along with narrow FFT resolution.

The ability to track ionospheric changes, and the tolerance of impulse noise, far outweigh any potential disadvantage in sensitivity due to sync-less detection. The receiver can also track changes in transmitted speed on the fly, with no hesitation or error, which allows the operator to unilaterally choose the appropriate sending speed.

This sync-less process was first used in the very slow 17-FSK mode JASON. It is also used in WSQ, but FSQ is the first mode ever to use this concept at conventional typing speeds. With no synchronizing process, decoding is immediate, with no need to track the signal with a phase-locked symbol clock, and no need for transmitted preamble 'training' symbols usually required to acquire sync. This leads to very fast signal acquisition (within one symbol, or as little as 100ms). In addition, the wide tolerance of symbol rate, about 2 to 6 baud, ensures that the mode is completely immune to path difference timing induced by the ionosphere, which can easily amount to 50 ms under NVIS conditions.

IFK+ Decoding

Once the tone numbers from the FFT have been determined (essentially the bin address of each determined symbol FFT result), the tone numbers are divided by three (because the transmitted tone spacing is three times the bin spacing), and rounded to an integer. This is the IFK+ coded difference. The choice of three times spacing (five times has also been tested) is important because the rounding process allows small differential errors (up to one bin) to be eliminated during rounding, as previously described.

In order to determine the character the codes refer to, the previous difference (from the previous transmitted symbol) is subtracted. Then, ONE is subtracted. If the result is less than zero, 33 is added. This now is the decoded Varicode number for the character.

Recall that the code table has in effect four sub-tables: the single-symbol characters, and three further two-symbol sets, defined by the continuation Varicode number received: 29, 30, or 31.

If the current Varicode number is above 28, the previous Varicode number, which has been retained, is used as an index into the code table defined by the current number, to determine the character which was sent, and is to be printed.

If the number is not above 28, the number is retained (and not printed) until the next number arrives. If this in turn is also below 28, the previous number must be a single-symbol character and is selected from the single-symbol table and printed.

Thus printing always lags one symbol behind the received signal. Two consecutive continuation symbols received constitute an error, and result in no print.

The FSQ Protocol

FSQ was designed from the start as a 'chat' mode, not a long-over QSO mode. It is capable of sending multiple sentences in each transmission, but it is easiest and quickest when just one sentence is sent, and the users quickly discover that this is the most enjoyable way to operate: it's more like a face-to-face conversation than a conventional ham QSO (in SSB or any other mode).

Each transmission starts with a callsign followed by a colon. The software inserts this preamble automatically. The user sets the callsign once during installation, and then simply types sentences. To start the transmission, the user then presses ENTER.

For example, if the user (say zl1xyz) types:

> Hello John – how are you this evening?<ENTER>

The software actually transmits:

> zl1xyz:Hello John – how are you this evening?

Nothing could be simpler. Every station within range will print this message.

Performance

There is no error correction involved, so users rely on the inherent robustness of the mode, and if there are errors, either due to 'typos' or reception errors, the users will quickly recognise these. If an error causes confusion, they can always ask for a repeat.

In one set of tests, the error rate was assessed under real conditions, operating 20W on the 40m band during the middle of the day, using an automated station 300 km away to return the transmitted sentences. From 250 words transmitted, at 6 baud, 223 were returned correct (89.2% copy)[17], while at 3 baud, copy was 100%. For practical purposes, 80% copy is considered adequate for a comfortable QSO.

Bearing in mind that no error correction is used by FSQ, and the above test involved transmission over the path in both directions, the results are indeed remarkable.

Tests using an ionospheric simulator bear these measurements out. Reception is essentially 100% for white-noise tests with the signal above –15dB S/N.

In another test on a one-way 3000 km path on 40m at night, much worse copy resulted, as expected. This was because the receiving station was in the Australian sub-tropical zone, and the path correspondingly had marked multi-path and Doppler. At 6 baud there was 37% copy, and at 3 baud 36% copy. Wider tone spacing (5 bins) resulted in a slight improvement, to 49 and 59% copy respectively.

[17] '% copy' is defined as the number of correctly received words, divided by the total number of words transmitted, x100. Missing word spaces are ignored.

Directed Messaging Introduction

As is evident from the results described in the previous section, under the design conditions (low HF band NVIS), reception is remarkably good. As a result, the designers felt it would be worth extending the program capability to provide some limited automation. This started out with simple commands that allowed assessment of performance, such as an automated reply and various message requests.

The results were so encouraging that a comprehensive Directed Messaging System (selective calling) was developed to make best use of the FSQ capability. The most obvious application for this system is emergency and public service communications, where field stations can have instructions, files and messages directed to them. This was always a target application for FSQ, hence the development for NVIS conditions. The system that has evolved, named FSQCall, has been shown to perform effectively for message handling, net control and for simple junk-free directed chat. The Directed mode is now the default mode of FSQCall.

FSQCall Directed mode operates a simple extension of the FSQ protocol. The preamble is changed a little, a direction (address and command) is applied, and finally a short EOT signal is added to the end of each message. These differences are simple to apply, as the user only needs to learn a few simple commands and remember to add a direction and command to each sentence. The rest is automatic.

Further, Directed sentences in FSQCall mode can be read easily in (undirected) FSQ mode, and undirected text is visible in the Monitor facility provided in Directed mode.

In Directed mode, only text directed to your station (essentially, to your callsign) will print, and only commands directed to your station will be responded to.

FSQCall Preamble

The FSQCall Directed mode preamble consists of the sending station's callsign, a colon, and a two character hexadecimal checksum. For example, the simplest transmission (a Sounding message) consists of only this. For example:

zl1bpu:b6

The checksum is a standard CCITT CRC8[18] calculated on the callsign and the colon. The maximum callsign length is 16 characters, and the CRC is capable of detecting up to four errors in this distance reliably. Pseudo-callsigns (such as place names or personal names) can be used, which is very convenient for Public Service and Emergency applications.

Directions and Triggers

All FSQCall messages, apart from the Sounding, carry at least one ***direction*** and a ***trigger***. The ***direction*** is the callsign or callsigns of the recipient station or stations, in other words, where the message is ***directed***. The ***trigger*** is one of a short list of one-letter commands sent to the directed station, which define what it should do with the message. There may or may not be a message body following the directions and triggers.

[18] IEEE 801.16. The algorithm is x^8 + x^2 + x + 1, and the result is expressed in lower-case HEX-ASCII.

Here are some simple examples:

```
zl1bpu:b6zl1abc Hello John.
preamble
         direction
               ^trigger
                message ...

zl1bpu:b6zl1abc zl2cde Hello guys.
preamble       ^trigger
         -directions--
                     ^trigger
                      message ...

zl1bpu:b6zl1abc?
preamble
         direction
               ^trigger
```

In the first example, ZL1BPU is simply chatting to ZL1ABC. He starts typing by giving the direction, the other station's callsign. The trigger used is a space, which is understood by the receiving station as an invitation to print the message that follows.

The second example illustrates how you can send the same sentence to two stations. There is also an 'all stations' direction (***allcall***) that can be used to send messages and some commands to all stations within range.

The third is an example of a simple automated query. It is the FSQCall equivalent of 'zl1abc de zl1bpu hw?' in Morse, i.e. asking for a signal report.

As you will see, these commands are simple to learn, and simple to implement. Note how the callsigns used are expressed in lower case. This preferred arrangement takes advantage of the faster transmission rate and lower error rate of lower case in the FSQ design. The FSQCall software is case sensitive, so ZL1ABC' and 'zl1abc' could be different stations. This is explained further below.

Error Checking

FSQCall sentences (only in Directed mode) are checked at the receiver in two ways. First, the sender's callsign is checked against the checksum received: if this is incorrect, the sentence is rejected. If it is correct, the rest of the sentence is checked to find the recipient's callsign. If this is not received correctly, the sentence is again rejected. If the callsign is received correctly, only the remainder of the message following the recipient's callsign is printed or acted upon.

When (and if) the incoming message is displayed, the preamble is shown without the checksum, so it looks like the preamble in the non-selcall (non-directed) mode. However, the callsign displayed is in this case the callsign verified by the checksum.

Checking the validity of the originator's callsign ensures that the message is correctly attributed, is correctly logged, and most significantly, responses can be sent to the correct station.

Logging

Whether in Directed mode or not, FSQCall from V0.42 maintains a Message Log, which contains every sentence received on the working channel. Each message is recorded with signal to noise ratio and time stamp. The file is in CSV (comma separated value) format, so can be analysed by third-party applications. A new file is opened at local midnight.

In Directed mode, FSQCall also maintains a log of all stations heard, in chronological order (the Heard Log). With each entry, the verified callsign, the date/time, and measured SNR are recorded in CSV format. This file is suitable for examination as a spreadsheet, and is also used by the application FSQplot,[19] which displays station SNR against time of day.

The Monitor tab provided on the user interface in Directed mode shows raw text received, and above it is a list of the most recent stations heard, again in chronological order. The time and quality of latest reception of any of these stations can be determined by checking the Heard Log.

Directed Callsigns

Each station responds to two 'callsigns'. The main one, obviously, is the callsign the user enters into the software. The user may operate several stations on the same frequency, provided they do not have the same exact callsign. 'zl1ee', zl1ee/1' and zl1ee/p' are examples of callsigns that can be used concurrently.

All stations respond to another pseudo 'callsign'. The 'callsign' ***allcall*** is considered to be the station's alternative callsign for some messages, including chat, file and image transfer, and alerts. This mechanism is widely used in nets to send chat to everyone in range.

Automated responses to pseudo callsign ***allcall*** are necessarily restricted, in order to prevent mayhem on the radio channel.

Use of Lower Case

Lower case is not mandatory for operator callsigns, but is encouraged for three reasons:

1. Lower case is quicker and easier to type
2. Lower case is faster to send (most characters are sent in one symbol)
3. The error rate is half that of upper case characters

FSQCall is case sensitive, so messages directed (for example) to 'ZL1EE' will not be recognised by the station with its callsign set to 'zl1ee'. If you wish to refer to 'zl1ee' in a sentence to another station, without arousing station 'zl1ee', use 'ZL1EE' or 'zl.1ee'!

Acknowledging Commands

As a general Directed Messaging design philosophy, messages that require a response do so only for a positive acknowledgement, i.e. the reply contains the word 'ACK'. There is

[19] ***FSQplot*** graphical Signal Monitor: see http://www.qsl.net/zl1bpu/SOFT/FSQplot009.zip

no ‘NACK’ response. The sending station can assume that if no ACK is received, the message either failed to be received, the command was not received or was inappropriate, or the action did not complete correctly.

Some message types (e.g. *allcall#*) result in multiple stations responding with ACK messages. The CSMA channel management protocol is not entirely foolproof, in particular where some stations cannot hear each other, so these ACKs may overlap. Users should exercise discretion when considering such commands on busy channels.

While it is possible to use multiple directions in one sentence, doing so with commands that require response is unwise, as inevitably this will result in two stations transmitting at once. The technique is best reserved for chat sentences where there are no automated replies.

Directed Messages

All the command triggers consist of one reserved character. These are taken from a pool of characters that are not permitted in callsigns (i.e. all except letters, numbers and ‘/’). As far as possible, the characters have a meaning easily associated with the character.

The list below shows all the reserved characters, and their current or planned use. Blank entries are reserved, with no currently planned application.

CHAR	ASCII	USE
Space	32	Directed chat
!	33	Relay command
“	34	
#	35	Message or file transfer
$	36	Request Heard List
%	37	Receive or request image
&	38	Report QTC message
‘	39	
(	40	
)	41	
*	42	Report FSQCall Directed status
+	43	Message retrieve (US versions)
,	44	
–	45	Message delete (US versions)
.	46	
:	58	Reserved, not used. Can be confused with preamble.
;	59	Extended relay command (US versions)
<	60	Reduce sending speed
=	61	Reserved for ECC file transfer
>	62	Increase sending speed
?	63	Report reception
@	64	Report QTH (or position) message
[	91	Used in file transfers
\	92	
]	93	Used in file transfers
^	94	Report FSQ program version
_	95	
`	96	
{	123	
|	124	Send Alert
}	125	
~	126	Delayed Relay command

Commands

As previously mentioned, command sentences consist of a direction followed immediately by a command. Multiple directions can be used in the same sentence under limited circumstances. Two directions in the same sentence, which are understood by the same station, will cause confusion, and only the second will be responded to. For example, one should not send 'zl1xyz@ allcall message', as the first part will be ignored by station zl1xyz.

Command Syntax

The syntax for currently defined commands is listed below.

callsign text (Enable print)
FSQCALL opens. Addressee station starts to print ***text***. No trigger character used except space following ***callsign***. You MUST use a space or the trigger will not be recognised.

callsign? text (Is station hearing me?)
FSQCALL opens. Addressee station starts to print ***text***. When squelch closes, station responds: ***origin_callsign snr=xxdB***

callsign* (Enable FSQCALL)
Switches station to ACTIVE if FSQCALL is in SLEEP. Addressee station starts to print. When squelch closes, station responds: ***callsign:<crc>***

callsign! message (Repeat my message)
FSQCALL opens. Addressee station starts to print. When squelch closes, station responds: ***callsign:<crc> message*** This is a simple relay mechanism. ***message*** may contain further trigger commands.

callsign~ message (Repeat my message later)
FSQCALL opens. Addressee station starts to print. When squelch closes, after a delay of about 15 seconds, station responds: ***callsign:<crc> message*** This delayed relay is useful when stations mentioned in ***message*** can hear and respond to the original message.

callsign#[nnn] (Send file to station)
Addressee station starts to print. Text following ***[nnn]*** will be saved in or appended to text file nnn.txt. When squelch closes, station responds ***callsign:<crc>*** if the message stored OK. There is no response if the message was not received. ***[nnn]*** can be alpha, numeric or mixed.

callsign#[filename.txt] (Send file to station)
Addressee station starts to print. File is opened from menu, and filename appended to message before file body text, File will be saved in or appended to text file filename.txt. When squelch closes, station responds ***callsign:<crc>*** if the message stored OK. There is no response if the message was not received. ***[nnn]*** can be alpha, numeric or mixed.

callsign@ (Request station position or location information)
Addressee station starts to print. When squelch closes, station responds with pre-recorded sentence, typically containing QTH information. This could be GPS position, locator, or physical address. This is the message stored and retreived by the QTH button.

callsign& (Request station message)
Addressee station starts to print. When squelch closes, station responds with pre-recorded sentence, typically containing station information or an 'Out to Lunch' message. This is the message stored by the QTC button.

callsign^ (Request Software Version)
Addressee station starts to print. When squelch closes, station responds with current FSQCALL software version.

callsign% (Send image)
Addressed station receives the command to record an analog graphic image. The picture appears on

the RX Pic tab, which opens automatically. Images are in colour, low or higher resolution, and take about 50 or 200 seconds to send, respectively. Images can be directed to *allcall*. Received images are always displayed as 640 x 480 pixels, the larger size achieved through pixel interpolation.

callsign|message (Send alert)
Addressee station starts to print. Alert pop-up box containing ***message*** is placed on the screen. When the operator closes this dialog, a response transmission is made: ***origin_callsign Alert ack***

cqcqcq text (Call for general chat)
FSQCALL opens if CQ is enabled. Following text prints.

allcall text (All stations print)
FSQCALL opens. Following text prints at all stations.

callsign> or ***callsign<*** (Speed change)
> increases speed, < decreases speed at target station. Response is ***origin_callsign: 4.5 baud*** (or whatever the new speed is). If the speed is already at the maximum or minimum speed, the response is the same, and there is no speed change.

allcall#[nnn] (Send file to all stations)
All stations within range start to print. Text following ***[nnn]*** will be saved in or appended to text file nnn.txt. When squelch closes, station responds ***callsign:<crc>*** if the message stored OK. There is no response if the message was not received. ***allcall#[filename.txt]*** **works the same as** ***callsign#[filename.txt]***

The user FSQCall application software has a Help Menu, which lists several Help files. The command syntax is displayed by the Syntax option. As each version of the software is released, this syntax list is updated, so is always current (whereas the list above may not).

Image Mode

FSQ is sufficiently robust that it works well with directed commands, to which a receiving station can respond automatically. One of the more unusual functions of FSQCall is the ability to transmit and receive small digitally managed and ordered analogue images, either from stored images or a web camera.

Denis UU9JDR first introduced digitally managed narrow-band analogue image transmission in the MIXW software, using MFSK16 mode. A digital command is sent to start reception with parameters defined by the command. Reception in analogue mode continues until sufficient samples have been taken to display the complete image.

The Image Mode offers four sizes and resolutions of image:

Mode	Resolution	Type	Duration
LO	160 x 120	Colour	48s
HI	320 x 240	Colour	192s
HD	640 x 480	Colour	768s
B&W	640 x 480	Black & White	256s

The B&W mode can be used for FAX transmission, grey-scale images or transmission of message forms.

Irrespective of the image resolution, the received image is always displayed at 640 x 480 pixels, enlarged where necessary through interpolation, to give best image appearance

Images can be sent from file, sent from a web camera, remotely requested from a list of files, and even requested from a remote webcam (imagine that at your holiday home!).

This picture was sent and received using FSQCall B&W image mode over a 300 km path on 80 metres at night using just 15 Watts.

Image Modulation

The modulation is similar to SSTV, i.e. it is analogue FM modulation, but that is where the similarity ends. The FSQ image mode uses only 300 Hz bandwidth (the same as FSQ digital transmissions), and like FSQ, there is no synchronization transmitted. Black is transmitted as 1350 Hz, white as 1650 Hz. The ITU Emission Designator for the image mode is 300HF3C.

The start of the picture is signalled digitally, and the receiver simply captures data continuously for L lines of P pixels, as requested, resulting in a LxP pixel image, then reception stops.

The images are approximately in 4:3 aspect ratio. Images should be sent and received in FSQCall ***Directed*** mode, and can be sent to just one recipient, to several recipients, or everyone on the channel, just as with other messages.

Narrow-Band Images

FSQ Image Mode was designed (like FSQ) for NVIS propagation on the lower HF bands. Unlike SSTV, the pictures are sent relatively slowly, in a narrow bandwidth (1/4 that of SSTV), and as a result the picture signal to noise ratio is generally better than SSTV for the same transmission power. As a guideline, FSQ signals (before the picture is started) need to be about +10dB SNR for noise-free picture reception.

FSQ pictures also offer significantly better reception than SSTV under NVIS conditions. Because there is no picture sync, there is no risk of the picture tearing due to timing changes, or breaking up during selective fades. All that happens is that the image may noise up momentarily, but will stay correctly aligned. The mode is also significantly less affected by ghosting.

Picture Format

The image mode sampling rate is 12,000 samples/sec, the same as the text mode. Each pixel consists of 10 samples, and there are P pixels/line. Therefore the line period is 10 x P/12000 seconds, or the line rate 12000/(10 x P) lines/sec. In colour, each line is sent three times, in RGB order. The aspect ratio of the images is 4:3.

As an example, if there were 320 pixels per line, and 240 lines, the line rate would be 3.75 lines/sec, and 240 x 3 (for RGB) lines would take 192 seconds, or 3.2 minutes for a colour picture containing 76,800 pixels. (See table on previous page).

No Sync

As mentioned, there is no horizontal (line) sync sent with the picture, and no vertical sync at the start of the picture. As a result, it is possible for reception to start slightly earlier or later than the standard delay from the time of the command, depending on the relative speed of the computers used at either end. This has two effects: first, the picture may be recorded shifted left or right, and the pixel colour order may be wrong. This is easily fixed after reception using the Phase control.

If the sound cards in the transmitting and receiving computers differ markedly in sample rate, the resulting picture can also be skewed. A small timing difference is enough to cause a noticeable slant in the picture. This is easily fixed after reception using the Slant

control. High quality sound cards should be better than 5ppm, and result in barely noticeable slant, but some cheaper computer built-in sound modules may be rather worse.

Picture Reception

Reception is completely automatic. Provided the software is in FSQCall ***Directed*** mode, and the sending station uses the correct callsign and trigger command, the receiving program will identify the picture start, and record the picture. In doing so, it automatically opens a floating Receive Picture window.

Demodulation uses a phase detection algorithm based on quadrature demodulation using the current and previous samples at 12000 samples/sec. All the samples are converted to pixels and stored, but only every 10^{th} one is displayed.

The raw image is stored in a temporary buffer of expanded size, which allows fine adjustment of the slant and phase to be made before the image is sampled and displayed.

If the transmitting station goes off the air or reception is lost during recording of a picture, the recording will continue until the necessary number of samples has been made, then reception stops. Picture reception can also be started and stopped manually.

Image phase and slant should not be adjusted before the picture reception has finished, as then only part of the image will be corrected.

Picture Transmission

A file open dialog on the TX Image window allows the user to select any image in a number of different standard formats. The image is rendered to the selected image size using a resampling process.

Once the image has been loaded, the user types the necessary direction, using the trigger character '%'. Pressing the Start button on the TX image window starts the transmission. First the digital image reception command for the requested format is appended to the user's direction, then the image is sent.

The recipient callsign for image transmission can be a single callsign, a list of callsigns, or ***allcall***.

In order to preserve privacy, web camera transmissions can only be made when specifically enabled by the user.

File Transfer

Text files can also be sent and automatically received using FSQCall ***Directed*** mode. The process uses the '#' trigger command, and the syntax is well described in the Help files. File transfer opens up a range of possibilities, including:

- Requesting a list of available files from the other station
- Requesting a specific file from another station
- Sending a file to another station or all stations
- Appending further text to a file already sent
- Sending and receiving telemetry data in a very flexible manner

Telemetry devices can be arranged to send data to be appended to their own file at the recipient (or all recipients), or can send data to separate files for different purposes. Multiple telemetry devices can send to the same file, or their own files, and can send bulk reports at once or individual reports as they are generated.

All these features and capabilities are explained in detail in the Help/Syntax Help and Help/Telemetry files.

When a file directed to your station arrives, your station sends an ACK reply, and the MSG RX button lights up. When you subsequently click on the MSG RX button, the program opens the file and turns off the MSQ RX button light.

This action only occurs as described for recognised file types such as '.TXT'. If you wish this file-opening action to occur for other file types (such as '.TLM' telemetry files or .CSV spreadsheet files), the file type needs to be associated with the suffix through the operating system.

If the file type is not associated (or the sender omits a suffix on the file name), the folder containing the file will be opened instead.

Files can also be pre-coded for Error Correction and decoded after reception, using external programs. These functions are explained in detail in a separate document.[20]

In order to preserve privacy, only files in a shared folder can be requested by a remote recipient, although the local user can send any file.

Files can be sent from across a network, and by placing a special file name (data.txt) in the shared folder, a networked user at another computer can automatically have the file sent, since the FSQCall application periodically scans the /shared folder for this file. Once the file (which contains the direction) has been transmitted, the file is deleted.

The user can write their own 'smart sounding' application, which generates bespoke sounding messages (perhaps containing a weather report or club news) which automatically sends its information in this way.

The use of remote third-party 'Helper' programs is described in the Telemetry Help file. The same technique can be applied manually for any text file as just described.

[20] ***ZL1BPU FSQ and Error Correction***, http://www.qsl.net/zl1bpu/DOCS/FSQ and Error Correction.pdf