

FSQ

FSQ, Fast Simple QSO, is an Incremental-Frequency-Keyed mode using an offset differential modulation scheme similar to DominoEX, and Thor. It is a unique mode for fldigi users in that it is a line by line transmission rather than character by character. FSQ uses 33 tones spaced by 3 times the 3 baud symbol rate or 8.8 Hz. The offset rotation of the IFKP sequence offers improved performance under NVIS conditions, because the rotation significantly reduces the risk of adjacent symbols causing inter-symbol interference.

fldigi can operate on the following FSQ modes without special setup by the operator:

Mode	Symbol Rate	WPM	Bandwidth
FSQ 2	1.95 baud	~20	290 Hz
FSQ 3*	2.93 baud	~30	290 Hz
FSQ 4.5	3.91 baud	~40	290 Hz
FSQ 6	5.86 baud	~60	290 Hz

- FSQ-3 is the base rate.

At the time of this release there is only one other amateur modem application that supports FSQ, FSQCALL, written by Con Wassilieff, ZL2AFP. FSQ is an invention of Murray Greenman, ZL1BPU. Both Murray and Con were very supportive during the development and testing of fldigi's FSQ implementation.

fldigi-fsq supports the undirected, directed and image modes of FSQCALL.

This FSQ help content borrows shamelessly from Murray Greenman's (ZL1BPU) "FSQ Explained" document.

Table of Contents

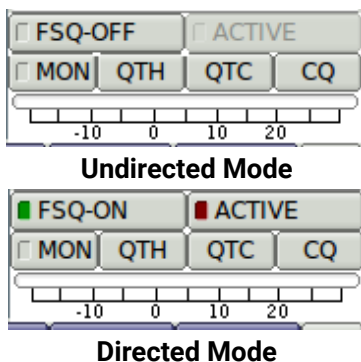
- ↓ [Modulation Scheme](#)
- ↓ [Transmit Functions](#)
- ↓ [FSQ Protocol](#)
- ↓ [Directed Mode](#)
- ↓ [Triggers](#)
- ↓ [Commands](#)
- ↓ [Relaying messages](#)
- ↓ [Special keys](#)
- ↓ [Heard List](#)
- ↓ [Image Mode](#)
- ↓ [Disable RxID](#)



fsq main dialog

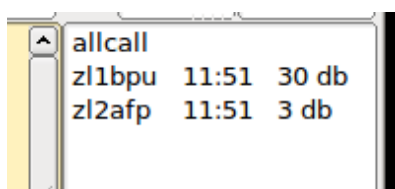
The main dialog fsq user interface provides an Rx-text panel, a Tx-text panel, a "Heard" list, fsq specific control buttons, and a signal to noise indicator.

The Rx-text panel displays all decoded text in the UnDirected Mode. The Tx-text panel is for entering all transmitted text, including directives. These two panels can be resized vertically by dragging the dividing bar which separates them.



The Rx-text panel displays only recognized text in the Directed Mode. The text parser must be satisfied that the sending stations callsign/crc8 agree and that the directed station callsign is your own or "allcall". FSQ is case sensitive, and your sending case must match the senders case, i.e. w1hkj and W1HKJ are not the same for the purposes of FSQ Directed Mode.

- **FSQ-ON**
when on the xmt and rcv are in FSQ directed mode
when off the xmt and rcv are in FSQ undirected mode
- **ACTIVE**
when on the decoder will accept all incoming directed traffic
when off the decocer will only respond to the wake-up * trigger
- **MON**
when on the monitor panel is visible
- **QTH**
appends the myQTH text to the tx panel
- **QTC**
appends the myQTC text to the tx panel
- **CQ**
sends the FSQ cq call 'cqccq'



Heard List Panel

The "Heard" list displays the callsign, s/n, and time of reception of all correctly decoded transmissions.

NOTE: fldigi function keys are inoperative in the FSQ mode!

Modulation Scheme

FSQ is designed to use 32 tone differences between 33 equally spaced tones; 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 a total alphabet of 104 ASCII characters. The extra characters 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.

Symbol(s)	Char	Symbol(s)	Char	Symbol(s)	Char	Symbol(s)	Char
0	SPACE	0-29	@	0-30	~	0-31	=
1	a	1-29	A	1-30	1	1-31	[
2	b	2-29	B	2-30	2	2-31	\
3	c	3-29	C	3-30	3	3-31]
4	d	4-29	D	4-30	4	4-31	^
5	e	5-29	E	5-30	5	5-31	
6	f	6-29	F	6-30	6	6-31	{
7	g	7-29	G	7-30	7	7-31	
8	h	8-29	H	8-30	8	8-31	}
9	i	9-29	I	9-30	9	9-31	`
10	j	10-29	J	10-30	0	10-31	±
11	k	11-29	K	11-30	!	11-31	÷
12	l	12-29	L	12-30	quote	12-31	°
13	m	13-29	M	13-30	#	13-31	×
14	n	14-29	N	14-30	\$	14-31	£
15	o	15-29	O	15-30	%	15-31	
16	p	16-29	P	16-30	&	16-31	
17	q	17-29	Q	17-30		17-31	
18	r	18-29	R	18-30	(18-31	
19	s	19-29	S	19-30)	19-31	
20	t	20-29	T	20-30	*	20-31	
21	u	21-29	U	21-30	+	21-31	
22	v	22-29	V	22-30	-	22-31	
23	w	23-29	W	23-30	/	23-31	
24	x	24-29	X	24-30	:	24-31	
25	y	25-29	Y	25-30	;	25-31	
26	z	26-29	Z	26-30	<	26-31	
27	.	27-29	,	27-30	>	27-31	BS
28	CRLF	28-29	?	28-30	IDLE	28-31	DEL

FSQ Varicode Table

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. MFSK modes such as Piccolo, Coquelet and MFSK16 use direct MFSK modulation, assigning the code directly to a tone number. These modes may suffer from more inter-symbol interference than desirable, cannot tolerate frequency drift, and can be difficult to tune.

FSQ uses very narrow tone spacing. 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+, there is no opportunity for the tones to remain the same for two consecutive symbols, which allows the sync-less process to operate 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.

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.

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.

Transmit Functions

The user types text into the Tx panel, which can be edited before transmission. Transmission starts when an ENTER character is pressed. It will also start as a result of one of the automatic processes, such as ‘Sounding’ or automated replies.

The characters in the Tx panel are then encoded, 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.

BOT Sequence

At the start of transmission in directed and non-Directed Mode, the BOT sequence, a SPACE followed by a LINEFEED, 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 the colon that follows the callsign. This also locates the two callsign checksum characters.

EOT Sequence

At the end of transmission in UnDirected Mode, the transmitter is simply turned off once the last useful character is sent. In Directed Mode, however, a special character sequence is sent just before the transmission stops, to signal EOT.

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 Directed Mode. Print will continue through fades without fldigi 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.

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.

In Directed mode, 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 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 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. This allows all stations on the channel to know what other stations are there with minimum use of channel bandwidth.

Sounding is sent with the lowest message priority. Sounding is essentially random, although the time between Sounds is selectable at 1, 10 and 30 minute intervals. The start time is set by user action, and essentially random.

FSQ Protocol

FSQ is designed 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.

Each transmission starts with the fldigi station callsign (converted to lowercase).

For example, if the user (say z11xyz) types:

Hello John – how are you this evening<ENTER>

The software actually transmits:

z11xyz: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.

Operating 20W on the 40m band during the middle of the day, an automated station 300 km away was used to return transmitted sentences. From 250 words transmitted, at 6 baud, 223 were returned correct (89.2% copy), while at 3 baud, copy was 100%. For practical purposes, 80% copy is considered adequate for a comfortable QSO.

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

Directed Mode

FSQ Directed Mode operates as 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 FSQ mode can be read easily in undirected FSQ mode, and undirected text is visible in the monitor panel which can be viewed in Directed Mode. In Directed Mode, only text directed to your callsign will print in the Rx panel, and only commands directed to your station will be responded to.



FSQ Monitor Panel

The Monitor Panel is only shown during Directed Mode operation. It is hidden when Undirected Mode is in use. All incoming traffic is printed on the Monitor Panel. This includes both Directed and Undirected traffic. The decoding is affected by the fdigi squelch setting.

You may see what appears to be a perfectly good Directed data stream on the Monitor Panel which never appears on the Rx Panel. When this occurs, the data stream has failed to parse correctly:

1. the BOT is missing or corrupt,
2. the CRC8 check has failed,
3. the EOT is missing or corrupt,
4. or it is not directed to your callsign or allcall.

The lower half of the monitor panel displays text that is queued for transmission, for example the response to a ? or a \$ trigger. Automated responses are not sent immediately, but after a small random delay AND when the squelch is not open.

FSQCall Preamble

The FSQCall 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:

z11bpu:b6

The checksum is a standard CCITT CRC8 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 FSQ Directed Mode 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.

Here are some simple examples:

z11bpu:b1z11abc hello john

w1hkj:efw3fqk21bm hello guys

w1hkj:efk21bm?

Color coding:

[] sending station
[] crc8 characters
[] directed station(s)
[] trigger|

Directed Mode Examples

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 FSQ way of asking for a signal report. The queried station will respond with your s/n report as measured by the FSQ program that is being used.

Error Checking

FSQ Directed Mode sentences 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 UnDirected 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

In Directed Mode, FSQ maintains a log of all stations heard, in chronological order. With each entry, verified callsign, the date/time, and measured SNR are recorded in CSV (comma separated value) format. This is suitable for examination as a spreadsheet. The heard log file is named "fsq_heard_log.txt" and is located in the fldigi.files/temp folder.

A complete audit log of transactions is also maintained in Directed Mode. This includes regular time stamps, all raw received text, and all transmitted sentences. The audit log is named "fsq_audit_log.txt" and is located in the fldigi.files/temp folder.

Directed Callsigns

1. Each station responds to three 'callsigns'.
2. The operator callsign (fldigi converts operator call to lowercase).
3. The 'callsign' allcall, to which the software will act upon certain messages, including chat, file and image transfer, and alerts. This mechanism is widely used in nets to send chat to everyone in range.
4. The 'callsign' cqccq is recognised as a legitimate chat 'callsign'.

Use of Lower Case

FSQ is case sensitive, so messages directed (for example) to 'ZL1EE' will not be recognised by the station with its callsign set to 'zl1ee'. Lower case is not mandatory for operator callsigns, but is encouraged for these 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.

Triggers

All the command triggers (currently) consist of one reserved character. These are taken from a pool of characters that are not permitted in callsigns. As far as possible, the characters have a meaning easily associated with the character.

This list 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	Repeat command
"	34	--
#	35	Message or file transfer
\$	36	Request Heard List
%	37	Receive image
&	38	Report QTC message
'	39	--
(40	--
)	41	--
*	42	Restore FSQ to ACTIVE from SLEEP
+	43	Message retrieve
,	44	--
-	45	Message delete (1)
.	46	--
:	58	Reserved, not used.
;	59	Relay command
<	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	Report the squelch setting in dB (1)
`	96	--
{	123	--
	124	Send Alert
}	125	--
~	126	Delayed Repeat command

(1) not supported

Commands

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*** 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?) Addressee station starts to print text. When squelch closes, station responds: **origin_callsign snr=xxdB**
- **callsign*** Switches station to ACTIVE if FSQSEL is in SLEEP. Addressee station starts to print. When squelch closes, station responds: **callsign:CRC-VAL Active**
- **callsign! *message*** (Repeat my message) Addressee station starts to print. When squelch closes, station responds: **callsign:CRC-VAL message**. This is a simple relay mechanism. *message* may contain further trigger commands.
- **callsign~ *message*** (Repeat my message later) Addressee station starts to print. When squelch closes, after a delay of about 15 seconds, station responds: **callsign:CRC-VAL message** This delayed relay is useful when stations mentioned in *message* can hear and respond to the original message.
- **callsign; [*relay_to*]{*trigger*}{*message*}** Relay my message to the '*relay_to*' station. See Relay section for details.
- **callsign!sendto *message*** Relay message to the 'sendto' station. Addressee station starts to print. When squelch closes, station relays the 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* in the same folder as the log/audit files. When squelch closes, station responds **callsign:CRC-VAL Ack** 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* in the same folder as the log/audit files. When squelch closes, station responds **callsign:CRC-VAL** 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 retrieved 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) Addressee station receives the command to record an analog graphic image. The Rx Image Dialog will open and the image painted as it is being received. Images are in color or grey scale and always displayed in native size.. Images can be directed to allcall.
- **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) Following text prints.
- **allcall** text (All stations print) 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-VAL Ack 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]**.

Relaying messages

This is a relay mechanism which preserves the origin callsign as the message propagates so that any response can be relayed back to that station. message can contain further trigger commands.

Originating station: relay_callsign; dest_callsign message

Example 1: Simple relay with no response

```
k1a types: k2a;k3a hi
k1a sends: k1a:11k2a;k3a hi
k2a sees : k1a;;k3a hi
k2a sends: k2a:22k3a [k1a] hi
k3a sees : k2a:[k1a] hi
```

Example 2: Relay with response

```
k1a types: k2a;k3a@
k1a sends: k1a:11k2a;k3a@
k2a sees : k1a;;k3a@
k2a sends: k2a:22k3a [k1a]@
k3a sees : k2a:[k1a]@
k3a sends: k3a:33k2a;k1a vienna, va fm18iw
k2a sees : k3a;;k1a vienna, va fm18iw
k2a sends: k2a:22k1a [k3a] vienna, va fm18iw
k1a sees : k2a:[k3a] vienna, va fm18iw
```

Example 3: Relayed allcall

```
k1a types: k2a;allcall Calling NET. Any Relays?
k1a sends: k1a:11k2a;allcall Calling the NET. Any Relays?
k2a sees : k1a;allcall Calling the NET. Any Relays?
k2a sends: k2a:22allcall [k1a] Calling the NET. Any Relays?
```

A checking in station might type

```
k2a;k1a ka4cdn (mike)
```

k1a would see the check-in relayed through k2a as

```
k2a:[k1a] ka4cdn (mike)
```

```
[ ] sending station
[ ] crc8 characters
[ ] directed station(s)
[ ] trigger
```

Relay Examples

Special keys

Pressing the F1 key will transfer the "last call" selected from the heard list to the Tx panel. "last call" defaults to allcall on program startup.

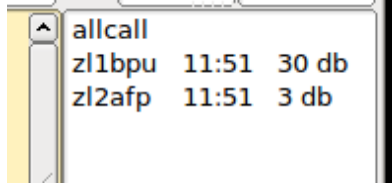
Pressing the F2 key will transfer the last command to the Tx panel. The last command is an empty string on program startup.

Heard List

The FSQ heard list is automatically added to, modified, and aged in accordance with the FSQ user configuration parameters. Each time a valid callsign/crc8 pair is discovered it is either added to the heard list, or it's entry is updated with the last GMT and signal-to-noise ratio.

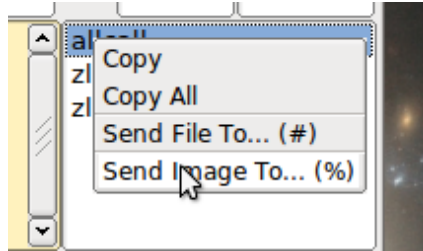
A single click on a heard list entry will make it the "last call" string.

A double click on a heard list entry will also add the call to the Tx panel. If multiple heard list entries are double clicked each one is added to the Tx panel separated by a space.

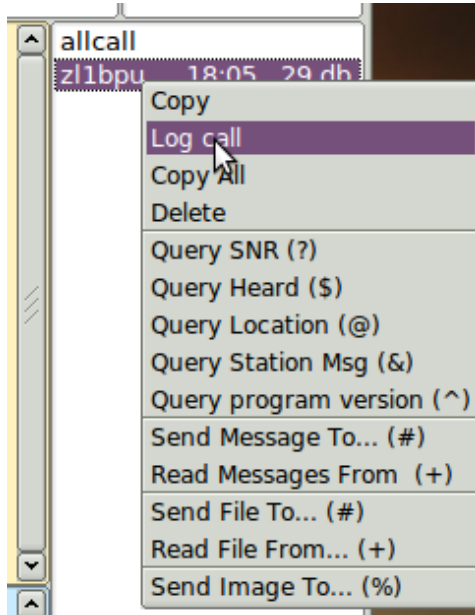


Heard List

Heard list entries have an associated pop-up menu, available by right-clicking on the desired entry:



All Call Menu



Directed Menu

Image Mode

fdigi-fsq has the ability to transmit and receive small digital images selected from those stored on the disk. The image transmission is similar to but not compatible with the MFSKpic modes. FSQ images are always transmitted in a fixed width to height ratio. The W x H format is not limited to a single ratio, nor is it limited to just color, or just gray scale.

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

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

Available image aspect ratios / formats include:

- 160x120 color (1)
- 320x240 color (1)
- 640x480 grey (1)
- 640x480 color (1)
- 240x300 color
- 240x300 grey

- 120x150 color
- 120x150 grey

(1) compatible with the program FSQCALL

Narrow-Band Images

FSQ Image Mode was designed for NVIS propagation on the lower HF bands. Unlike SSTV, the pictures are sent relatively slowly, in a narrow bandwidth (1/4 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.

Image 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 $10 \times P / 12000$ seconds, or the line rate $12000 / (10 \times P)$ lines/sec. In colour, each line is sent three times, in BGR order.

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

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. If you plan to send FSQ images you should calibrate your sound card to WWV.

Receive Image

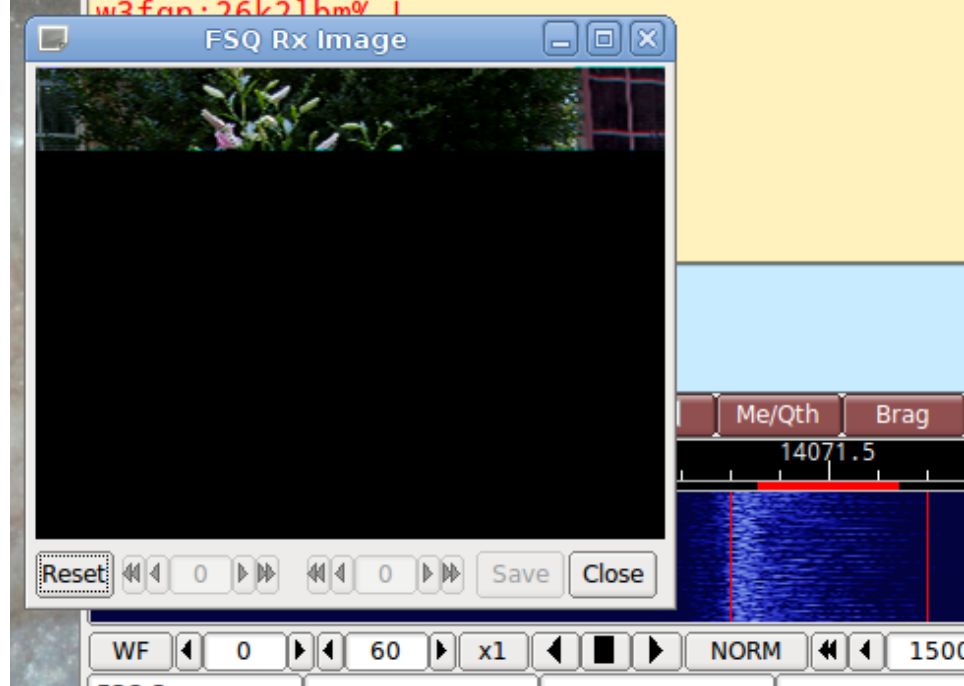


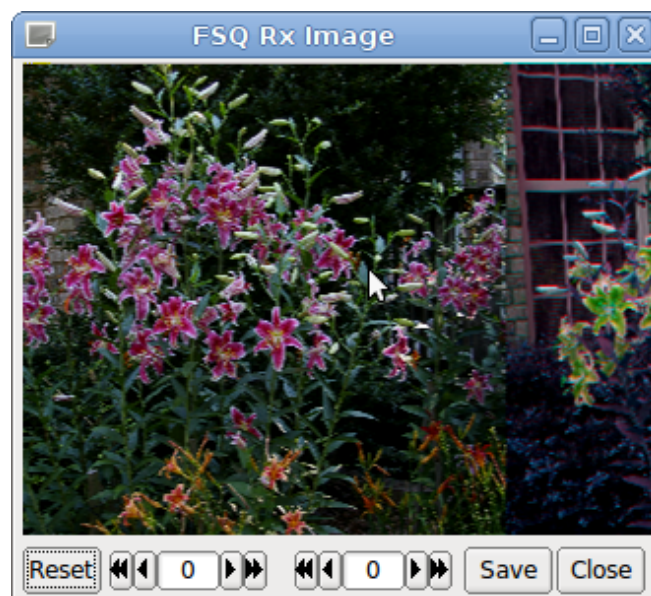
Image Reception

Reception is completely automatic. Provided the software is in Directed Mode, and the sending station uses the correct callsign and trigger command, the decoder will identify the picture start, and record the picture. In doing so, it automatically opens a separate "FSQ Rx Image" dialog.

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 the average over every 10 samples 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.



Received - no correction

The Rx Image dialog controls from left to right are:

- Reset: clears all rx data

- Phase adjustment: moves image left and right
- Slant adjustment: slews image diagonally
- Save: saves the image to the fldigi image folder with a date-time stamped file name
- Close: closes the image dialog



Received - phase corrected

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

Transmit Image

Access to the transmit image functions is only available via the heard list pop-up menus. Right click on the recipient call sign and select the "Send Image to ..." item from the pop-up menu.

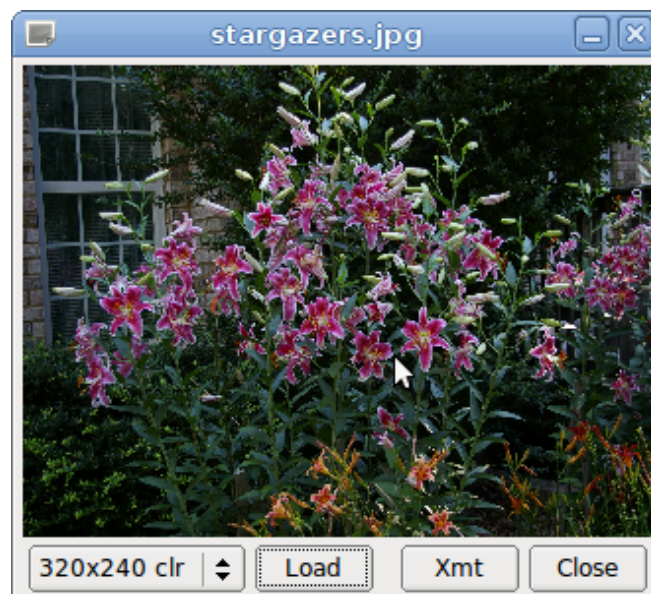


Image Transmission

A file open dialog is displayed when the "Load" button is pressed on the TX Image dialog. This 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.

fldigi will try to force fit the image to the selected aspect ratio. You should try to work with images that are in a 4:3 or a 3:4 aspect ratio. The selected image does not need to be the same size. Fldigi will enlarge or reduce the image using a resampling

technique to fit the selected image size. The example photo of Star Gazer lilies is a 3.2 MB file, 3008 x 2000 pixels. It was taken in my lily patch in June 2014.

Once the image has been loaded, the user starts the transmission by pressing the "Xmt" button on the transmit image dialog. fldigi formats the correct command sequence and starts the transmission.

The recipient callsign for image transmission can be a single callsign, or allcall.

Sending to a single recipient or to allcall is accomplished as described above. To send to multiple addresses you must add their callsigns to the Tx panel, separated by spaces and with a trailing space. Then press the Xmt button on the Tx Image dialog.

fldigi remains in transmit mode when the image transmission is completed and the deadman timer is started.

Disable RxID

You will probably want to disable the RxID detector during FSQ operations. If RxID is enabled and an RsID signal is detected the program will switch to the mode specified by the RxID. That will completely alter the user interface.

[Return to Top of Page](#)

[Return to Main Page](#)