Fldigi

Help version 3.12 - 19 July 2009

License

Contents

Recognitions

This software would not have been possible without the contribution of many programmers who have given their best to the open source community. The application is built upon the foundation of the Fast Light Tool Kit (http://www.fltk.org), a wonderfully fast and efficient graphical user interface design library. Many have asked what the Fast Light means. There are probably as many answers as there are programmers using the toolkit. I prefer to think of it as lightning fast and light on the code size. Take a look at the size of the executable for fldigi and then compare it with similar applications. I think you will be surprised by how small it is for what it does.

The current development team consists of:

- Dave Freese - W1HKJ
- Stelios Bounanos - M0GLD
- Leigh Klotz - WA5ZNU
- Stephane Fillod - F8CFE

Several authors have placed their digital modem code and signal processing code in the public domain and their source was either an inspiration or in some cases formed the backbone of the code used in Fldigi.

- AE4JY - WinPsk - a windows application
- Takuya OOURA - a generic Fast Fourier Transform for real valued data streams - http://momonga.t.u-tokyo.ac.jp/~ooura/fft.html
- Tomi Manninen, OH2BNS - gmfsk - a great digital modem program for Linux
- Hamish Moffatt, VK3SB - dominoEX code originally for gmfsk
- Joe Veldhuis, KD8ATU - Olivia 2 tone and other mods.
- Dr. Steven W. Smith - author of "Digital Signal Processing", who has kindly placed an entire book on digital signal processing on the internet. (http://www.dspguide.com)

If you make a side-by-side comparison between gmfsk and fldigi source code you will see that they follow the same general structure. The primary difference is that gmfsk is written in the C language and uses the gnome/gtk libraries for the user interface. Fldigi is a C++ application that uses the Fast Light Tool Kit (Fltk) gui library. The design of Fldigi puts emphasis on separating the user interface from the sound card and transceiver input/output operations. Nearly all modern digital modem programs use a programming paradigm called "threads." Threads are light weight processes that share the same memory space, but each has its own stack. The use of threads makes the program look and feel responsive to the user while a lot of code is being executed in the background.

Many of the modem source code files are C to C++ rewrites from the gmfsk application. They say that copying is the best form of flattery and gmfsk simply had the best explanations and the easiest source code to read and understand. The author had also spent several months creating improvements and fixing bugs in the original gmfsk application. That exercise was the impetus to create Fldigi. Gmfsk was becoming too difficult to modify without great effort. I also wanted to learn more about coding with threads ... fools rush in.

The Fast Fourier Transform used by Fldigi is a rewrite of Takuya Ooura's C code. The rewrite is in C++ but you will see the strong resemblence to Takuya's original if you study both. Takuya's FFT code was also used in the Winpsk program. Some of the signal processing algorithms used in Fldigi are from Dr. Smith's book. His on-line publication is sufficient to allow you to become fluent in fft analysis and the creation of digital filters. I printed the relevant pdf files and then purchased the hard bound copy. Improvements to the original gmfsk signal processing algorithms can all be attributed to this excellent source.

And last but certainly not least, I must thank the crew who perform alpha testing of the application. These are stalwart amateurs who risk their operating system and radio equipment in testing, testing and more testing. Their only reward is in being able to influence the design of the application and the fun of seeing it work and the bugs disappear. Thank you to:

Call	Name	Call	Name	Call	Name	Call	Name
4Z5ST	Boris	K3GAU	David	KU1T	Zibi	VA3DB	Dianne
AA0HW	Chuck	K4XTT	Victor	KV9U	Rick	VE3IXI	Dave
AC7JN	Dave	K6KAR	Kirk	N0NB	Nate	VK2TMG	Brett
CT1DRB	David	K7BRK	Chris	N2AMG	Rick	VK4BDJ	David
CX7BF	Walter	K4RE	Brian	N4UM	Tim	W3NR	Ed
DF4OR	Ekki	K9AO	Rick	N4ZNV	Mike	W4ROS	Ross
DK1JBE	Tom	KB3FN	Lynn	N6WFL	Jason	W6JVE	Jim
DL6XAZ	Fred	KD0AR	Mike	N8FQ	Joe	WA3VPZ	Marshal
DL8FCL	Walter	KD4O	Phil	NN8B	Don	WA4SXZ	Rich
G0UZP	Paul	KD8DKT	Mike	NT1G	Skip	WB8ROL	Gary
G3TDJ	Andy	KE3Y	Travis	OZ4KK	Erik	WD4FDW	Steve
G6CKR	Roger	KH6TY	Skip	PA0R	Rein	WD4FNY	Bill
G8SQH	David	KL7NA	Rob	PA3GWH	Richard	WU9Q	Bob

and many others whose names are not listed, please accept my apology.

All the testers were on different platforms and used different Linux distributions and Windows versions. They represent users that have transceivers capable of hamlib support and users who do not. They have varying interests from very slow speed CW to high speed keyboard full break-in CW, from RTTY contesters to PSK rag chewers. They have insisted that fldigi perform well under all of those operations. I have been amazed by the global distribution of the testing team. It is easy to think that the internet will be the death of amateur radio. On the contrary it opens up so many additional ways for us to be cooperative.

Contents

Installing Fldigi

Precompiled Binary

The precompiled binary is available with and without a dependency on PulseAudio. Unless you know that your system uses PulseAudio for its sound card service you should not download that version.

You will need three shared libraries on your system, hamlib-1.2.7; libsamplerate; and libportaudio2. Use the libraries available for your linux distribution. Most current distributions use either deb or rpm files and can be accessed from a global repository. Building the libraries from source should be a last resort unless you are a knowledgable Linux user and have performed a library build from source in the past.

Hamlib required

You will need to have hamlib-1.2.7 installed on your system before fldigi can be executed. Most current distributions either have a deb or rpm distribution file for hamlib-1.2.7. If you must compile from source you can find it at:

http://www.hamlib.org

Follow the instructions in the source code top directory to compile, link and install the library.

libsamplerate required

You will need to have libsample installed on your system before fldigi can be executed. Most current distributions either have a deb or rpm distribution file for libsamplerate. If you must compile from source you can find it at:

libsamplerate source

Follow the instructions in the source code top directory to compile, link and install the library.

PortAudio2 required

You will need to have libportaudio2 installed on your system before fldigi can be executed. Most current distributions either have a deb or rpm distribution file for libportaudio2. If you must compile from source you can find it at:

http://www.portaudio.com

Follow the instructions in the source code top directory to compile, link and install the library.

Installing fldigi

The static executables are tested on as many distributions as possible to insure that they work "out-of-the-box", but there are always a few Linux distributions that may have a missing link or library. The precompiled binaries have been tested and work correctly on all of the Debian and Ubuntu/Kubuntu distributions. They have also been tested and confirmed to work on Suse 10.1, and Mandriva 2007.

Download the tarball for the binary version and unpack to a directory on your HD such as $HOME/bin or some other convenient directory of your choosing. The least common denominator for unpacking a tarball is to download the file and save it to a convenient directory such as $HOME/downloads. Then open up a terminal window. Assuming you will be installing the executable to $HOME/bin do the following and that you have downloaded the tarball to $HOME/downloads

```
cd
cd bin
tar xzf ../downloads/fldigi-D.dd.npa.bin.tgz
```

where D.dd is the current version number as in 3.10

You can create a link to the fldigi executable on your desktop using the fldigi.png icon located at

http://www.w1hkj.com/fldigi-distro/fldigi-psk.png

Creating a desktop link to an application is different for each desktop manager, so please refer to the documentation for your specific manager.

The first time that you execute fldigi either from the command line or by clicking on the executable in a file manager or the desktop icon it will create a new directory and file:

- $HOME/.fldigi
- $HOME/.fldigi/macros.mdf

You can edit the macros.mdf file to create custom macros that meet your operating requirements, but it is easier to use the built-in macro editor that fldigi provides.

Contents

Installing Fldigi on Windows

The port of fldigi to the Windows operating system is built using cross-compilers on Linux. The cross-compilation environment is created using mingw32.

Installing fldigi on windows is very simple. Simply execute the installer program and both fldigi and flarq will be installed in the default programs directory structure for the specific Microsoft version that is being targeted. Desktop icons and desktop menu items will be created. An uninstaller link will be created on the desktop menu.

Click on the desktop icon to start the application. Resize the main dialog to suit your screen. Adjust the Rx/Tx divider to your liking. Then set up the operator and sound card configuration items; configuring fldigi. When you have fldigi receiving and decoding signals you can exit the application which will allow you to save your configuration settings.

Now open up the following folder using your windows file explorer if you do not have a login name and password:

On XP	C:\"Documents and Settings\<urlogin>\fldigi.files\"
On W2K	C:\"Documents and Settings\<urlogin>\fldigi.files\"
On Vista	C:\User\<urlogin>\fldigi.files\

where <urlogin> is the name with which you log onto the computer.

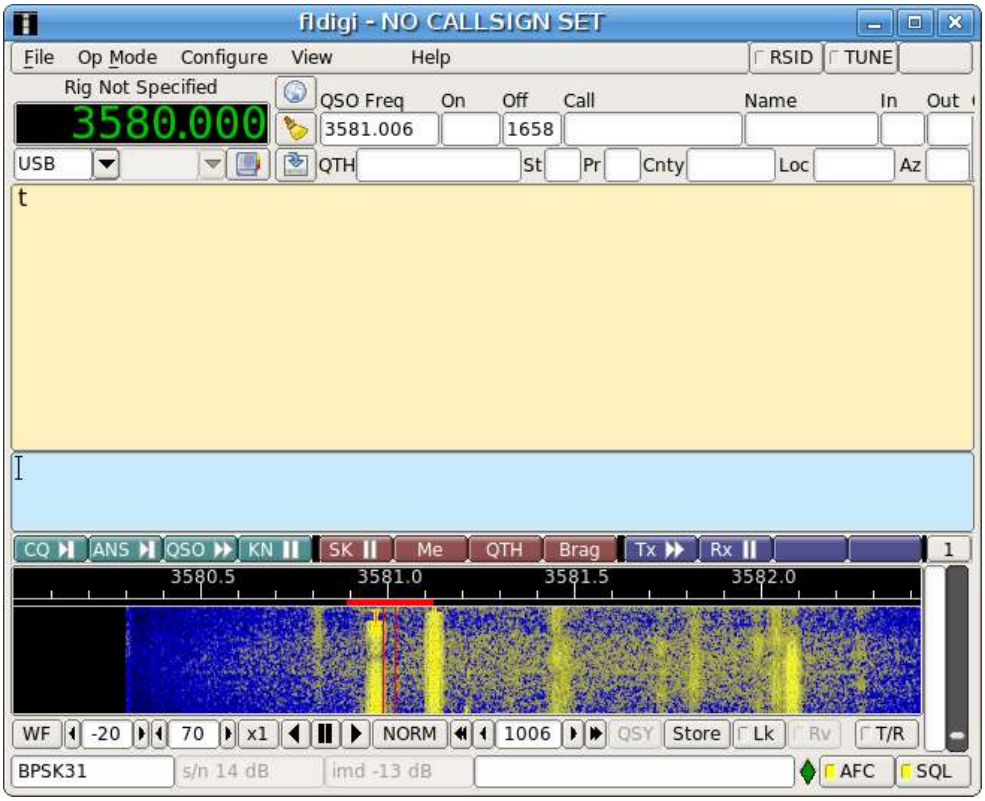
All of these files were generated by fldigi when it first started. The files with the extension pal are palette definition files. The file "macros.mdf" contains the macro definitions which you can change using the macro editor. fldigi.status and fldigi_def.xml are used for storing the application state and configuration items respectively. With the exception of the location of this folder the operation of fldigi on windows is identical to linux. In all instances where the help files make reference to $HOME/.fldigi you should be substitute the appropriate directory for XP or Vista.

Please take the time to read and reread the on-line help file. Better yet download the Adobe Reader file so that you can view the help locally without needing access to the internet. Fldigi is a large complex program with many ways for the user to customize its operation to his or her hardware environment.

Contents

New Installation

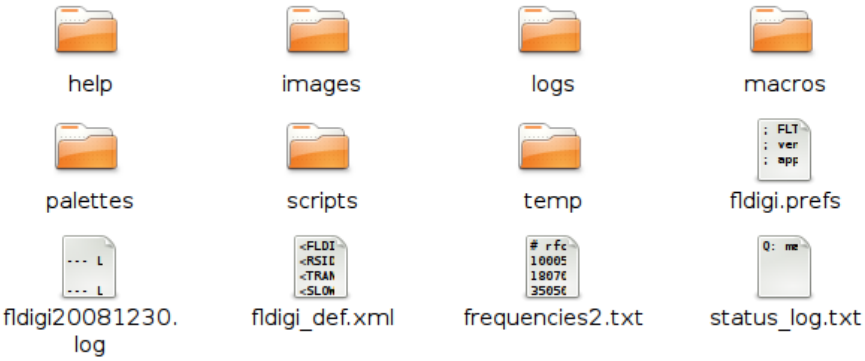
fldigi's opening screen looks like the following when starting fldigi for the first time or when setting up a second or subsequent instance using the --config-dir command line switch.



fldigi will create a working files folder, multiple sub folders and also populate them with a set of default files. The working files folder is different on the different OS.

XP / W2K	C:\Documents and Settings\<username>\fldigi.files
Vista	C:\Users\<username>\fldigi.files
Linux	/home/<username>/.fldigi

After closing the application the working folder will contain the following folders and files:

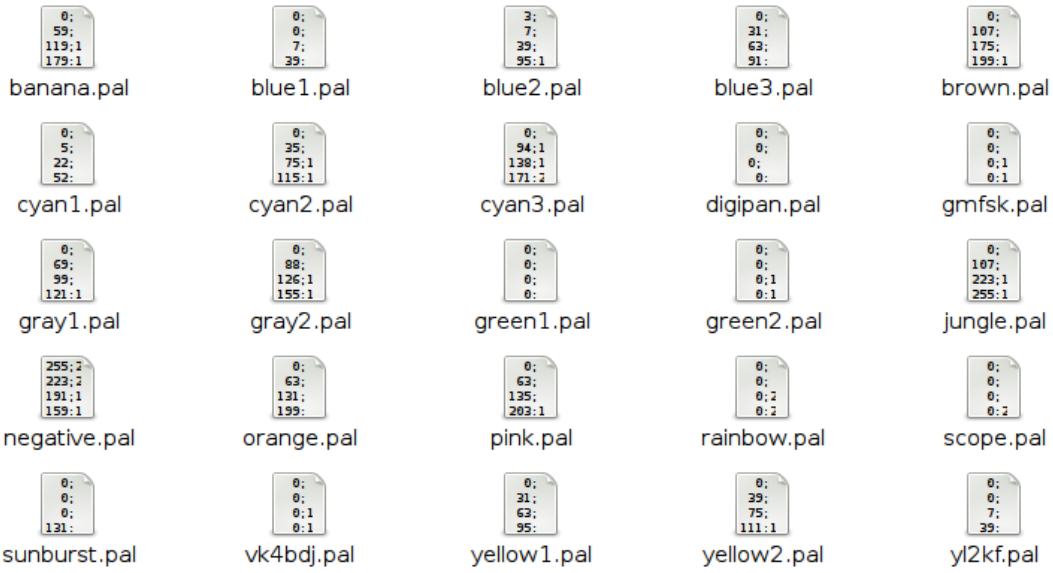


The help, images, logs, scripts and temp folders will be empty. They will contain program created files as you use the program or you may post files in those folders for use by fldigi. Images to be sent with the MFSKpic mode should be placed in images. Your logbook database will appear in logs. If you are running on Linux then you can use various scripts to enhance the macro language that fldigi supports. The temp directory holds files that are transitory and you can safely delete those files between sessions. The 5 files that appear initially are:

fldigi.prefs	contains variables that describe the status of fldigi when last used. This is an ASCII text file that you can safely read. You should not edit or change this file.
fldigiYYYYMMDD.log	this is an historical log of all the received and transmitted text during the day for which the log refers
fldigi_def.xml	contains variables that relate to all of fldigi's configurable items. This is an ASCII text file that conforms with the XML specification. You can safely read this file but should not edit or change it.
frequencies2.txt	an ASCII text file that contains the default (and / or modified) entries for fldigi's rig control process
status_log.txt	a log of events for the most current fldigi execution. This file will contain information relative to any errors that my occur and is important for debugging purposes.

The macros folder contains a single file: macros.mdf. This is an ASCII text file that contains the default macro definitions. After running fldigi for a while and creating your own sets of macro definitions there will be additional *.mdf files located here.

The palettes folder contains the following files:



Each of these is a palette definition file that is used to modify the appearance of the waterfall. Fldigi has a palette editor that enables you to modify these default files or to create your own. The file format of these files is identical to the palette files used by DigiPan. The final color rendition might be a little different as a result of using different painting functions. The file digipan.pal contains:

```
  0;  0;  0
  0;  0; 62
  0;  0;126
  0;  0;214
145;142; 96
181;184; 48
223;226;105
254;254;  4
255; 58;  0
```

Don't bother trying to modify these using an editor. The palette editor is much easier to use and will keep you from wrecking havoc with the program.

The easiest way to find the working files folder is to start fldigi and then select the menu item File/Show config.

Contents

Configuring

The first time you execute fldigi you should resize the main window to suit your screen dimensions. Then adjust the divider line between the Rx and Tx text widgets..

Fldigi contains many configurable items, to specify operator data, user interface, and modem characteristics. The application also saves many state variables between executions. It will start up in the state that it was last used.

You should initially configure the following:

Operator UI Waterfall Modems Rig Audio Id Misc Callsign DB

and

Colors & Fonts

When the program receives and transmits digital signals and your rig control is satisfactory then you can continue configuring other aspects of the program:

Operator UI Waterfall Modems Rig Audio Id Misc Callsign DB

You can configure each modem type to suit your particular operating needs, but the defaults should be satisfactory for most users.

CW DominoEX FeldHell MT-63 Olivia Psk Rtty Thor

To learn more about the characteristics of specific digital modes look here: Digital Modes, Sights & Sounds.

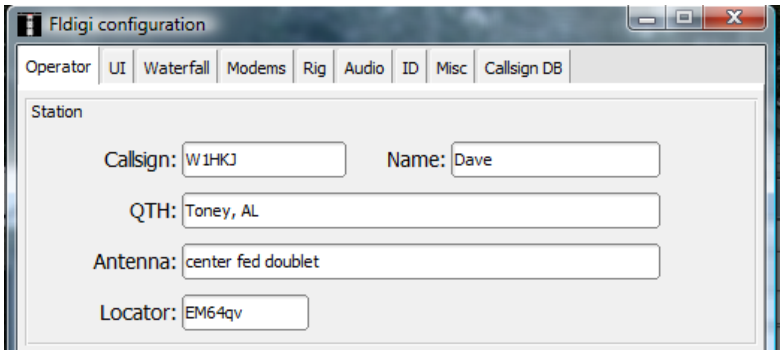
When you have completed the configuration go to the Configure menu and select Save config or press the "Save Config" button on the configure dialog box. The program will write the file ~/.fldigi/fldigi_def.xml.

Exit the program and restart it to test that your configuration was saved and is working correctly.

Your fldigi install is now ready for you to start receiving and transmitting digital signals.

Fldigi recognizes if any configuration changes are made and not saved. You will then be prompted to save the configuration when exiting the program.
Contents

Configure Operator



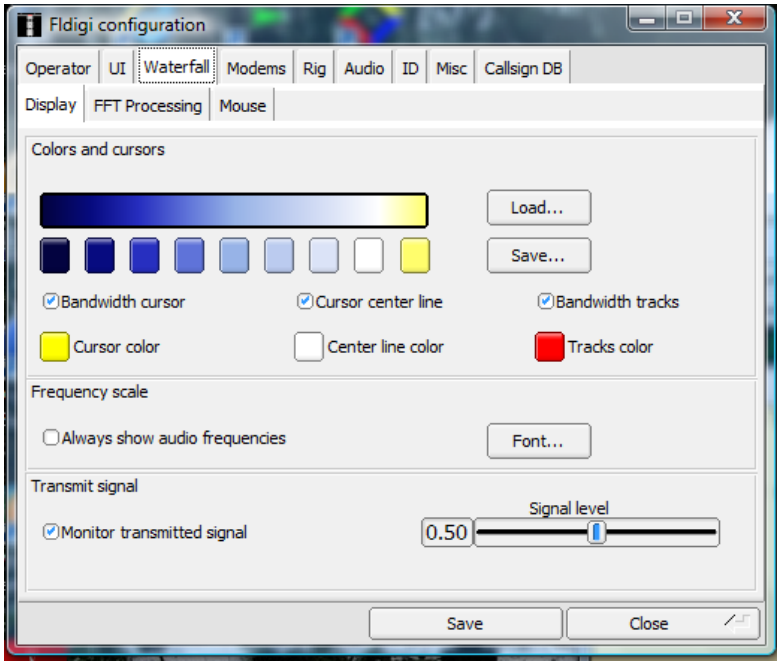
Enter your personal information on the Operator tab of the configuration dialog. This information is used by some of the macro expanders.

The antenna information is required if you elect to report to the spotting web site, http://report.psk.gladstonefamily.net

Your locator data is also used for automatically computing Azimuth to a remote locator when that is available from an on-line database Call query.

Waterfall Configuration

Waterfall Configuration

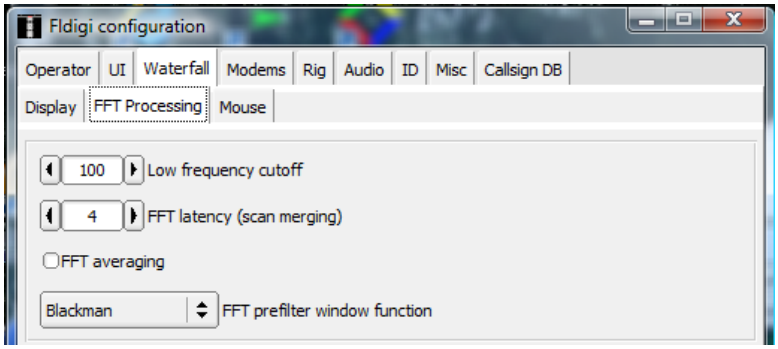


The waterfall palette or color scheme can be altered to suit your personal tastes and visual needs. When fldigi is first started it creates a wide range of pre-built palettes in the $HOME/.fldigi folder. The "Load" button gives you access to those palettes. You may change any palette by clicking on the various color buttons beneath the palette sample. A color picker opens for you to select the color by various means including specifying the RGB values. If you create a palette that suits you better than any of the prebuilt ones you can "Save" the palette.

The waterfall cursor is a set of markers on the frequency scale that are spaced a signal bandwidth apart. You can add a pair of lines that drop down from those two markers for the full height of the waterfall by selecting Cursor BW. You can add a center line cursor to this pair of BW line by selecting Cursor Center line. You can also add a set of BW lines that straddle the received signal tracking point by selecting Bandwidth tracks. All three of these options are color selectable. Click on the colored button below the check box and a color selection dialog will open.

The frequency scale defaults to RF frequency. You can select to show audio frequencies

You can monitor the transmitted audio waveform and also set the level of the monitored signal. This IS NOT your final transmitted signal!

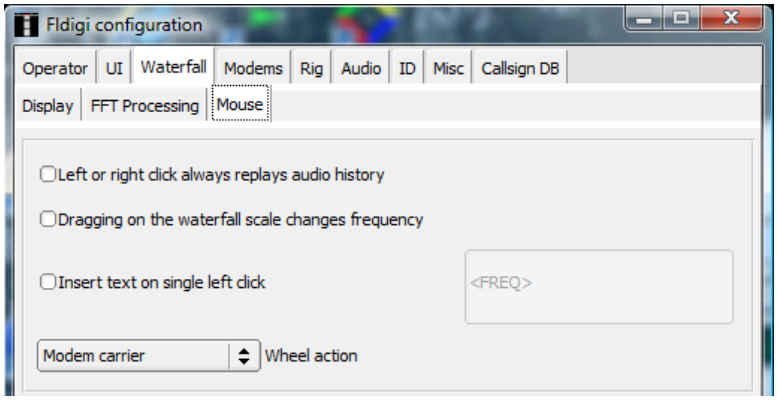


You can extinguish the display of received signals below a particular audio frequency.

Fldigi's waterfall FFT has a bin size of 1 Hz. With an FFT of 8192 and a sampling rate of 8000 it takes almost a second to accumulate enough data to perform the full FFT. A waterfall that dropped at one scan line per second would be hard on the viewer, so fldigi uses a first-in-first-out (FIFO) 8192 byte buffer for the FFT data. 512 byte audio blocks move through the buffer with each successive read of the sound card. The full buffer of 8192 samples is used to compute the FFT. That means that data in the FFT can have a latency of 8 scans. This provides excellent frequency resolution but poor time resolution (the vertical waterfall appearance). The latency control allows you to select the number of 512 byte blocks that are used for the FFT. The default latency is set to 4. You should be able to achieve a reasonable compromise between the time and frequency domain resolutions.

FFT averaging can be used to smooth the waterfall display in the frequency domain.

The FFT Prefilter or window function is used to reduce aliasing in the FFT computation. The default prefilter for the Fast Fourier Transform associated with the waterfall is Blackman. You can try the other windowing filter. Under some conditions you might prefer one of those. The Blackman window has proven best for my setup.



The mouse behavior in the waterfall panel can be controlled to suit your particular operating style. You might want to replay the saved audio history everytime you either left click to select or right click to preview a particular signal. You can move the transceiver frequency in increments of 100 Hz by dragging the waterfall scale. You can also choose to insert a line of text into the Rx panel each time you left click a waterfall signal. The text can include expandable macro tags.

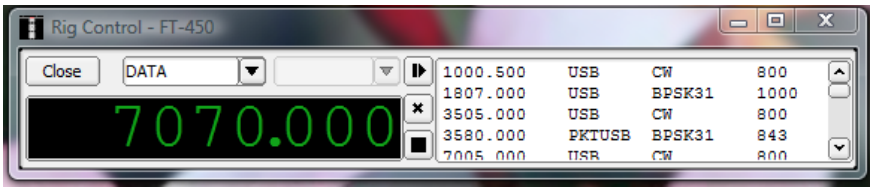
The mouse wheel behavior can also be tailored to your liking:

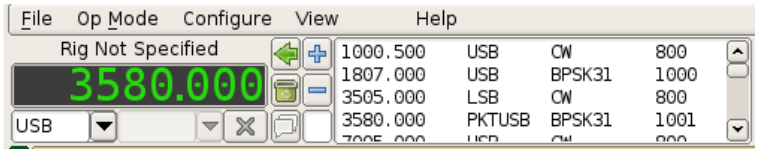
- None - no mouse wheel activity in waterfall panel
- AFC range or BW - adjust the AFC range/BW up/down
- Squelch level - adjust the squelch level up/down
- Signal search - search up / down for next signal in current mode
- Modem carrier - adjust the audio tracking point +/- Hz increments
- Modem - select modem type from a full rotary of available modems
- Scroll - move the waterfall left/right in 100 Hz increments (for 2x, 4x expanded waterfall view)

Contents

Rig Control

If your transceiver is capable of a serial stream command link then you may elect to use one of several rig control methods. Rig control is accomplished by either a separate rig control dialog or one which is docked:





Both are shown on Windows Vista controlling the author's FT-450. The same control is used when the transceiver does not support CAT. When no CAT is available the control is simply a convenient way of keeping track of the transceiver USB/LSB frequency, the mode and the audio tracking point.

The buttons control selecting, adding and deleting entries in the frequency/mode list.

- add the current frequency / mode / audio track point to the list
- select the current list entry
- delete the highlighted entry from the list
- delete all entries from the list (a warning prompt will appear)

The docked rig control has these additional controls:

- show active frequencies based on either the entry field to the right or the stations locator, see pskreporter/spotter.
- entry field for active frequencies search, for example "EM."

The browser list contains frequency, sideband, modem type and audio frequency. The list is saved when fldigi is shut down.

The combo box on the left will allow the selection and control of the operating mode of the transceiver.

The combo box on the right will allow the selection and control of the transceiver bandwidth.

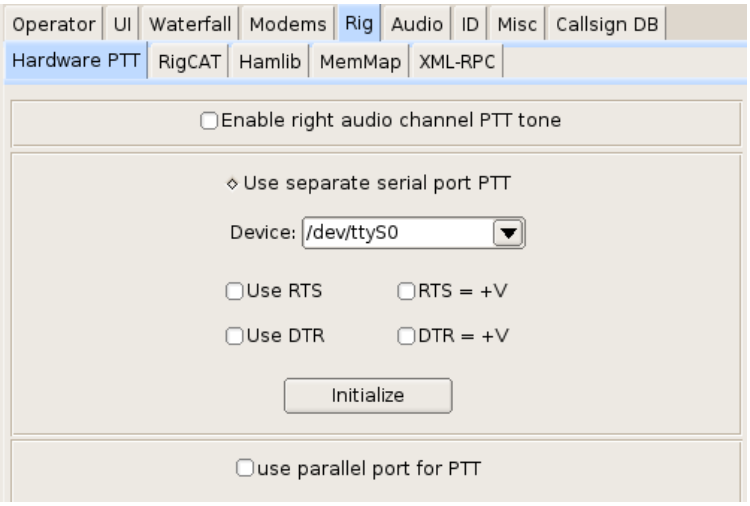
The frequency display is in fact a set of special buttons. Each digit may be left-clicked to increment in frequency by that digit value, or right clicked to decrement by that digit value. The leading digits will follow suit if a decade rollover occurs. You can also place the mouse cursor on a digit and then use the mouse wheel to roll the frequency up and down.

Manual entry of frequency can be accomplished by clicking on any digit and then entering the numeric value in KHz. Don't forget the decimal point if you are entering a fractional KHz value.

The mode combobox, the bandwidth combobox and the frequency display also annunciate the current transceiver status. If you change operating mode on the transceiver, that will be annunciated in the respective combobox and fldigi will adjust any internal parameters accordingly. Fldigi queries the transceiver 10 times per second to maintain a lock step with the transceiver.

Rig Configuration

Hardware PTT control



Right Channel VOX Signal

Fldigi can generate a 1000 Hz tone for the duration of the PTT keydown period. A simple tone detector/filter and transistor switch can be used to generate a PTT signal from this sound card output. The circuit will be similar to that used for QSK control. This might be a convenient way to create a PTT signal for a small notebook or netbook computer that does not have a serial or a parallel port.

Serial Port using DTR or RTS

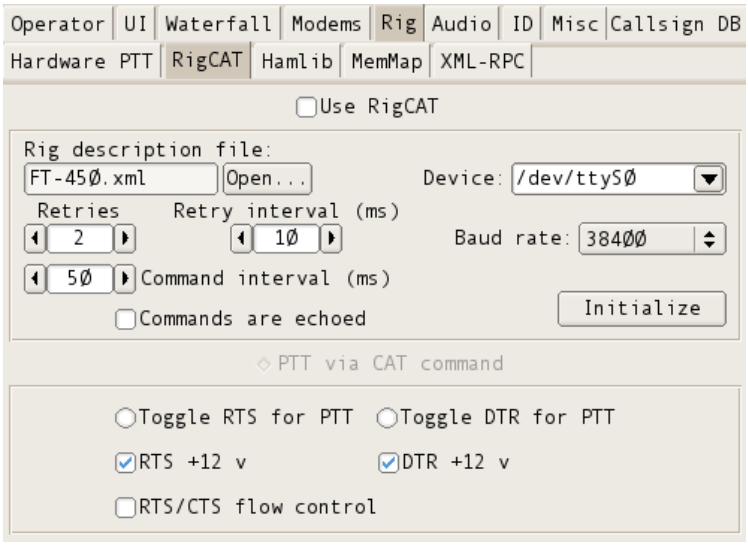
The simplest rig control is just being able to control the push to talk via an external transistor switch. You set this type of control on the first configuration tab for rig control.

You select this operation by checking the "Use serial port PTT". Select the serial port from the list (fldigi will have searched for available ports). Then specify whether the h/w uses RTS or DTR and whether a + or - voltage is required to toggle PTT on. You can use a serial port for control with the RTS and DTR pins configured for you particular interface. The program allows you to use RTS, DTR or BOTH for the PTT signal. Press the Initialize button to start the serial port.

Parallel Port (Linux and Free BSD only)

Fldidi sets and clears the parallel port pin, PARPORT_CONTROL_INIT, pin 16 on the 25 pin parallel port connector. Keydown sets Pin 16 to +5 volts and keyup sets the voltage to zero.

RigCAT control



RigCAT is a rig control system similar to hamlib that was developed specifically for fldigi. It uses command / response definitions that are found in various rig.xml files. You can use a rig.xml file specific for your transceiver or write and test one yourself. The easiest way is to adapt an existing rig xml file for a rig that is similar to your own. ICOM almost identical command/response strings for all of its transceiver line. Yaesu rigs have nearly all used unique command/response structures until just recently. The TS-450, TS-950 and others share a similar set of commands and responses.

RigCAT commands and responses are defined in a rig specific xml file which contains all of the required queries and responses in extended markup language format. Please read the specification document rigxml to learn more about this new way of building generic rig interface definitions and how they are used with fldigi. fldigi will look for a file in the $HOME/.fldigi/rigs directory for all files with extension ".xml". These contain definitions for the transceiver indicated by the file name, ie: FT-450.xml, IC-756PRO.xml, etc. You can download the appropriate xml files from the resource directory tree http://www.w1hkj.com/xmls or from the archives web page. Place the file in your rigs directory and fldigi will find it.

You will need to specify how your PTT will be triggered. This can be using a CAT command, the RTS or DTR pins or none. None would be appropriate if you are using the rig's VOX or an outboard sound card interface such as the SignalLink SL-1+ which produces its own VOX type of PTT. In that case simply leave all of the PTT options unselected.

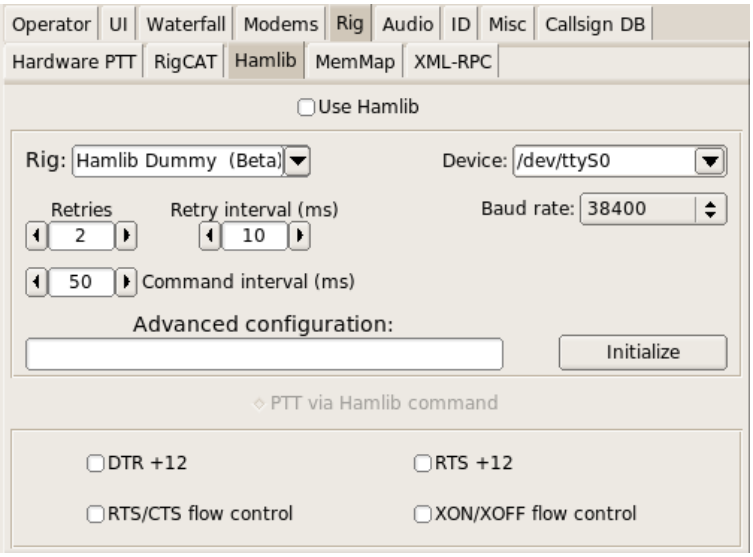
If you are using a transceiver or a rig interface such as CI-V that echos all serial data you check off the "Commands are echoed" box. That will suppress fldigi trying to respond to a command it just sent to the transceiver.

You may need to try various values of retries, retry interval, and command interval to achieve consistent rigcat control.

Press the Initialize button after setting all of the parameters. If the settings are all correct fldigi should start receiving frequency information from the rig and annunciating them on the rig control frequency display.

Hamlib CAT control

Hamlib is a set of standard libraries for interfacing to a large number of transceivers. The hamlib library system consists of a front end which acts on behalf of all rigs and backends which are specific to each rig. The fldigi implementation of hamlib differs on the various OS for which it is targeted. On the Unix/Linux based systems the hamlib is a shared library which the user must have installed on his or her system. This is the standard way of handling hamlib on Unix/Linux systems. On Windows the entire hamlib library has been compiled and statically linked into the application executable. No additional dynamic link libraries are necessary. This approach simplifies the installation of fldigi on Windows platforms.

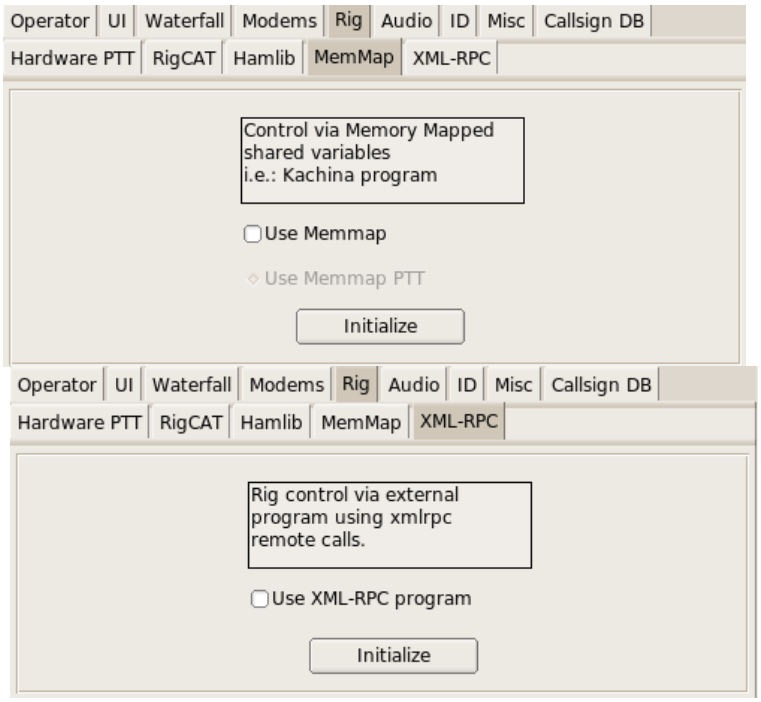


Select your transceiver from the list of supported units. Then select the serial port and baud rate. If you are familiar with the hamlib library you can send various startup sequences to the rig using the advanced configuration. PTT control can be achieved using CAT commands or via DTR / RTS on the same port as the control comms. You might also need to specifiy whether RTS/CTS flow control is uses (Kenwood rigs use this quite often) or if Xon/Xoff flow control is used.

You may need to try various values of retries, retry interval, and command interval to achieve consistent hamlib control.

Press the Initialize button after setting all of the parameters. If the settings are all correct fldigi should start receiving frequency information from the rig and annunciating them on the rig control frequency display.

Memory Mapped CAT & Xml-Rpc CAT



Memory mapped control is selected if you are operating a Kachina 505DSP using the W1HKJ control software for that rig.

Xml-Rpc allows third party software to control various aspects of fldigi operation including but not limited to rig control. If you are using a third party interface such as DxKeeper Bridge you might want to select this method of CAT.

Contents

Sound Card Configuration

Sound Card Configuration

A few words about sound I/O on the PC. "You are in a maze of twisty little passages, all alike".

PortAudio, PulseAudio and OSS are different ways in which fldigi can access your sound card through the various sound systems.

OSS was the first audio backend in fldigi. It works with the Linux sound system of the same name, which has now been replaced by ALSA but is still supported via an emulation layer. Its only advantage, as an audio backend, is that it's simple and doesn't require any external libraries.

The PortAudio backend was written subsequently to support OSS on Linux and FreeBSD, ALSA and JACK on Linux, CoreAudio on OS X, and also the various sound APIs on Windows -- all through the same PortAudio library.

PulseAudio is more than an audio hardware access layer; refer to its website for a summary of what it does. Fldigi supports it mainly because many Linux distributions are now integrating it with their desktops, but also because it has a few interesting features:

- it can take care of the resampling and volume control for us,
- it can stream audio over the network, and
- it makes it easier to run multiple fldigi instances (all accessing the same sound card).

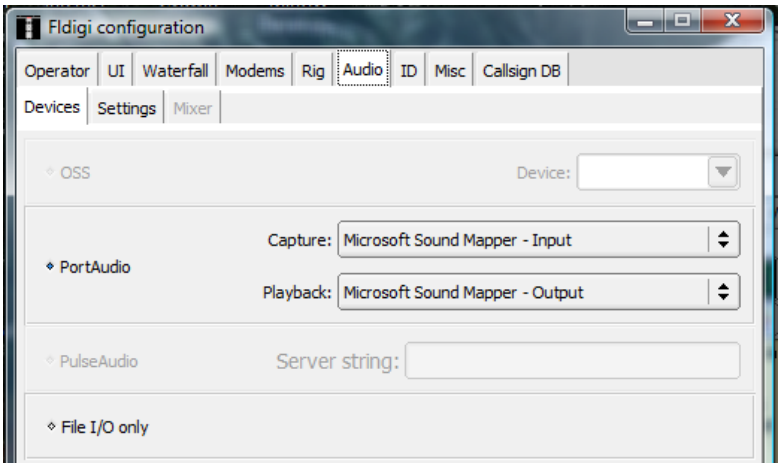
In the future it might be possible to replace all of these with a single backend, without any loss of functionality, performance, sound system or platform support. That'll be the day! Until then:

On Linux:

- Use PulseAudio if your Linux distro ships it, and you already have the pulseaudio daemon running (this is the case with Fedora 8/9 and Ubuntu 8.04, probably also with openSUSE 11.0). Or if you want networked audio, etc. etc.
- Otherwise, use PortAudio and select a device from the list(s). PortAudio is also the best way to access JACK, through which you can use other programs as audio sources/sinks -- particularly useful with SDR software. As with PulseAudio, you can select different capture and playback audio devices.
- The OSS backend should be used only as a last resort. Note that it has not been updated to support user-configurable sample rates.

On Windows:

- Use the PortAudio and select the device from the list(s).



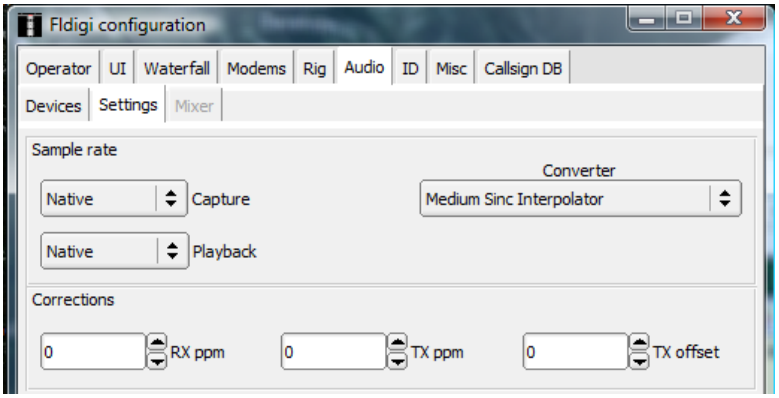
Select the SndCrd tab on the configuration dialog.

On Linux Fldigi can interface to the sound card using either the OSS, the Portaudio, or the PulseAudio. Each of the appropriate libraries must be present on the computer to use that particular sound i/o.

On Windows Fldigi uses the Portaudio sound driver only.

It is also possible to configure Fldigi with File I/O only, which is useful for testing the application without an interface to the sound card. In the File I/O only configuration you can record and playback audio files in a number of different formats including the "wav" format associated with the Windows operating system.

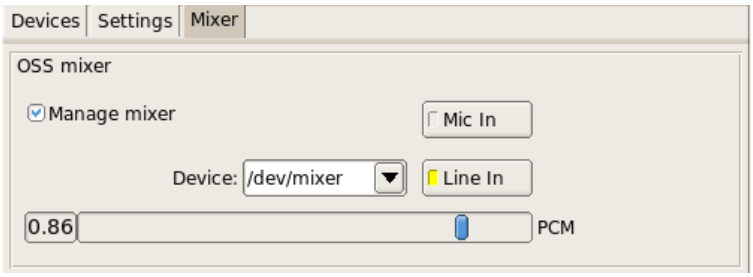
The program will find all active sound cards and the associated drivers for both. Select the sound card and driver type that will be used with the program. I recommend using the PortAudio device driver if that is available on your Linux distribution.



If PortAudio is selected then you can either allow the program to use the auto detect to determine the best sound card sampling rate, or you can pick from the drop down list. If you know your RX and TX sound card oscillator correction factors you can enter them now. If not you can determine the RX rate correction using a special WWV modem built into Fldigi. The decoder and encoder logic for each of the various modems require a specific sound card sample rate which may not be the the actual sound card sample rate. The conversion between the modem sample rate and the sound card sample rate is accomplished by one of a set of sample rate converters.

Sound card oscillators may have a slight error in frequency that causes their sampling rate to not be the value specified. This error is usually small enough to be measured in a parts per million. Fldigi uses a technique called rate conversion to correct the sampled waveform for this error. The error can be measured and the correction

factor determined by using the WWV calibration modem. The supporting library used for the converter provides several different levels of conversion. The default, "Medium Sinc Interpolator" will be satisfactory for most sound cards. If you are running fldigi on a computer with limited cpu power you might find it necessary to select one of the more cpu efficient converters, either "Fastests Sinc Interpolator", "ZOH Interpolator", or "Linear Interpolator". Each gives progressively poorer performance but use fewer cpu cycles to perform the frequency conversion. For really problematic computers you might want to completely disable the converter. This is accomplished by setting the "RX ppm" and "TX ppm" to zero. You should also be sure that the cpu type is set to "Slow cpu" on the miscellaneous configuration tab.

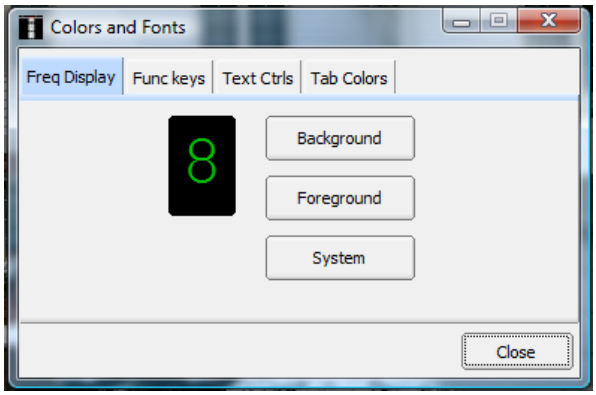


Mixer controls are only active on Linux. Select whether you will be using Line-In or Mic-In for the audio connection from the receiver output. Fldigi ALWAYS expects to use the Line-Out for driving the transmitter audio. Set the PCM level for your soundcard. If you check "Manage mixer" then the Tx and Rx "volume" controls on the main fldigi dialog will be active.

Colors & Fonts

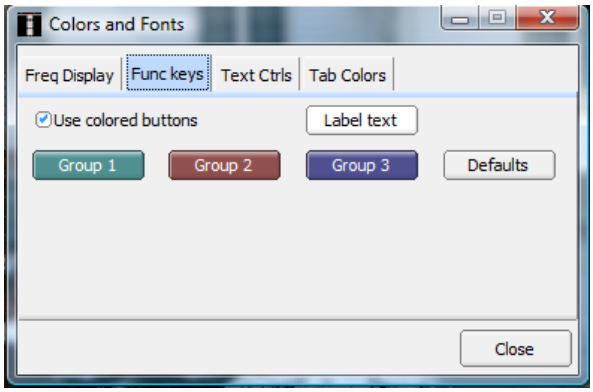
"System colors" are set by command line switches. The default is black on a white background.

From the Menu Configure/Defaults select the menu item Colors and Fonts and then select one of the following tabs. The text widget used for Rx, Tx and Event log displays has been improved to give better performance with proportional fonts. Fixed width fonts still give better performance and are not as demanding on the cpu. There are several very good fixed width fonts that include a slashed zero which are available for both Windows and Linux. If you are using a proportional font and find that the Rx text display gets unresponsive with large amounts of text then you should change to a fixed width font. Do a search on the internet for "Andale Mono". It is an excellent font for this use.



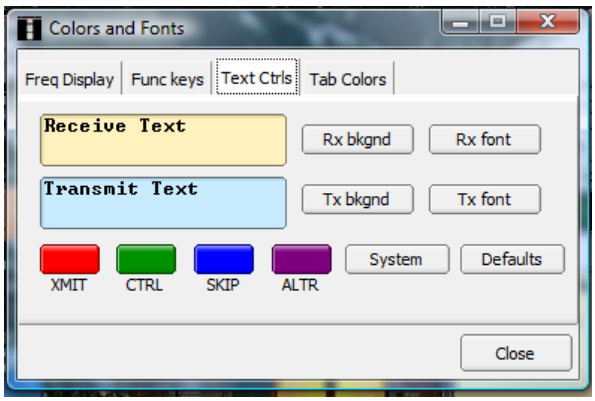
Freq Display

The rig control panel uses a special button for each digit the represents the transceiver frequency. The buttons are responsive to mouse clicks on the upper and lower half with corresponding changes it that unit's value. Unit value is also controlled by the mouse wheel when the cursor is over a particular digit. Select the background and foreground colors to please your overall color scheme and for best visual acuity. The System colors are the same ones that are used by all input and output text controls.



Func keys

You can color code the macro (function key) buttons in groups of 4, F1-F4, F5-F8, and F9-F12. The background color for each group is adjusted by clicking the respective Bkgnd button. The text color for the buton labels is adjusted by clicking on the Label Txt button. The colors will change on these buttons and also on the main dialog as you make these adjustments. The Defaults button restores the colors as shown in this view.

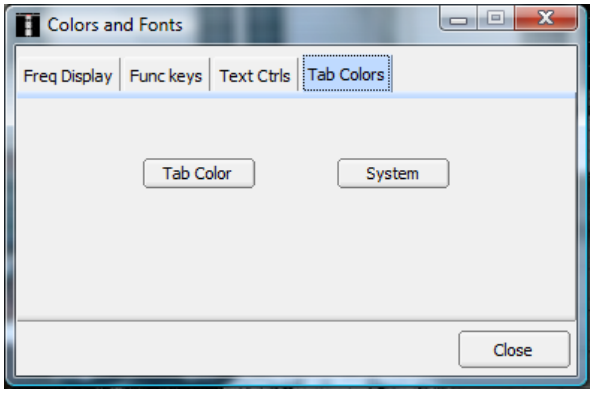


Text Ctrls

The initial color, font and font-size for the Rx and Tx panel are the default values. You can always return to these by pressing the Defaults button. The background color, font and font-size are independently selectable. The Rx panel displays text in one of 5 colors:

- normal received text - "Rx font" button
- transmitted text - XMIT button
- control characters - CTRL button
- skipped characters (Tx ON/OFF in Tx pane) - SKIP button

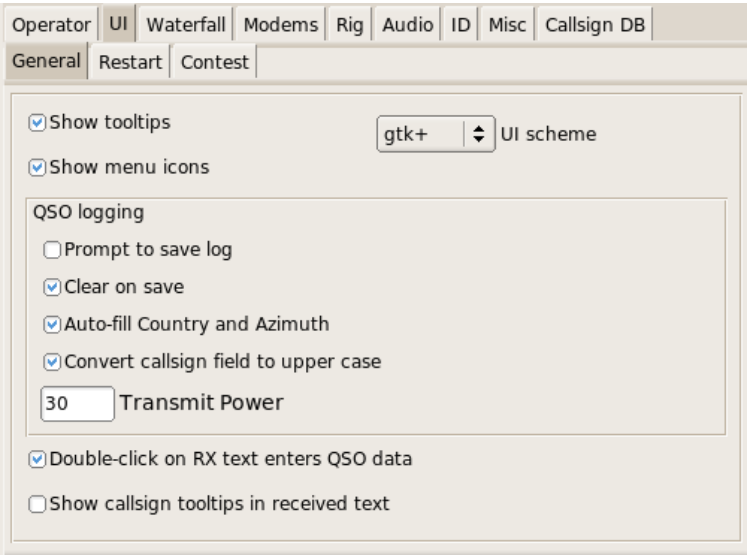
- quick view characters - ALTR button



Tab Colors

Adjust the color of all tabs to suit your personal taste.

User Interface Configuration



Fldigi offers tips on the use of nearly every aspect of its operation. These are particularly useful when you first use the program, but after you are familiar with it's operation they tend to get in the way. You can turn them off by de-selecting "Show tooltips"

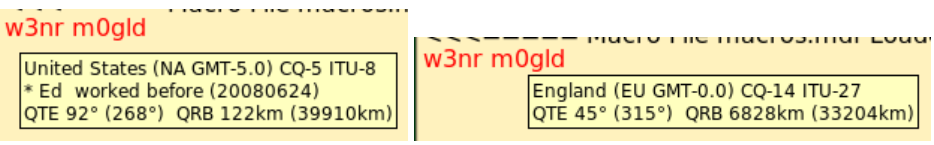
Some users prefer to not have icons on the menu system. You can turn them off also.

Fldigi offers three different look and feel based on the parent Fast Light Toolkit graphics interface; "base", "gtk+" and "plastic". These can be combined with command line specifiers for the default background and foreground colors to create a user unique look to fldigi. You will probably discover that the default colors and the gtk+ UI scheme are to be preferred.

Fldigi has a built in logbook. You can request to be prompted whenever there is an unsaved entry in the qso log area. You can also elect to whether to clear all of the qso fields when the log is saved or to leave them intact. Auto-fill Country and Azimuth uses the data found in the file "cnty.dat" that you should download and place in the fldigi default folder. You can force the callsign field to be upper case independent of capture or keyboard entry. You enter your default Transmit Power which is used for the logbook record.

Fldigi has various ways to transfer data in the Rx panel to the qso logging fields. The default is to use a Shift-Left-Click paradigm. You can also use a double click method if you prefer. The Shift-Left-Click will still function.

If you check the "Show callsign tooltips in received text" then the Rx text area will popup an information box whenever the mouse is held over a callsign for more than 2 seconds. The popup will look like one of the following:



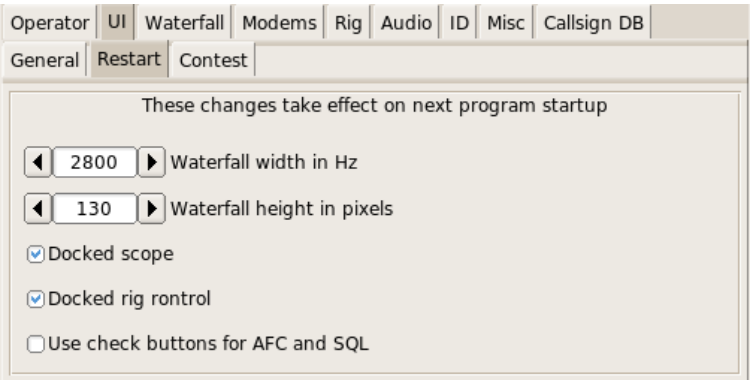
The data is derived by parsing the callsign and referral to both the logbook and the "cty.dat" file. If the station

was previously worked the operator's name and azimuth/distance will be computed from the logbook gridsquare entry (Loc). Otherwise the azimuth/distance is computed from the data in the cty.dat file.

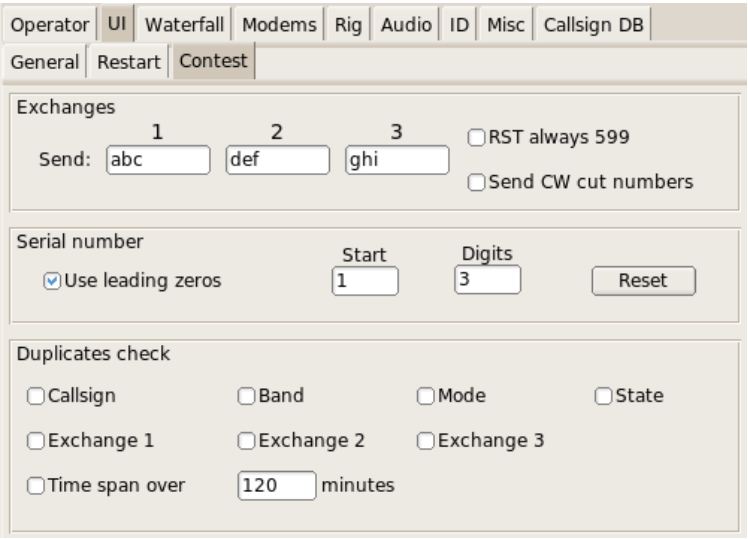
This file is maintained by, and can be downloaded from the following web site:

http://www.country-files.com/

Download the file and put it in your fldigi default files folder.



Some aspects of the user interface will need a program restart to take effect. These include the waterfall width and the waterfall height. The digiscope can be docked as well as floating. It is used for traditional views of the digital signal, such as phase vector, rtty cross hairs etc. The rig control can be either docked or floating. Most users seem to prefer the docked rig control. Finally you can select to use check buttons (boxes) for the AFC and Squelch controls.



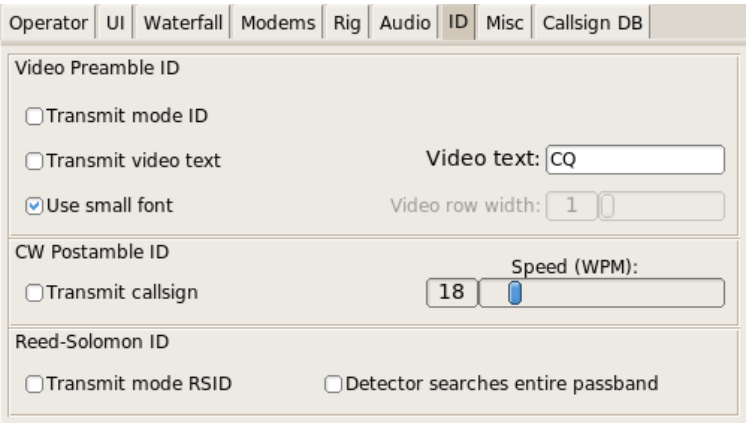
Fldigi supports a generic but robust set of contest functions. In addition to serial-in and serial-out you can capture and transmit three exchange sequences unique to a specific contest. Enter the exchange you want to send for each of the three. You can force the RST in/out to always be 599. That seems to be a norm for many contests. When operating in a CW contest you can have fldigi send cut numbers, T for 0, N for nine.

The serial number can be set to use leading zeros. You can specifiy the starting number for the sequence and how many digits are sent, ie: 0024. Pressing Reset will set the starting number to the qso logging serial out field. See Contest How To for more info.

You can check for duplicates by any combination of the specified named fields. You can also specify that the duplicate had to occur with a given time interval. Some VHF contests allow a duplicate CALL after a given time interval.

Contents

Id Configuration



Fldigi offers several ways to identify the operator or mode that is being used. This is particularly useful when using a hard to recognize mode such as Thor, Olivia or MT63.

Video Text

Transmitted video text will appear as a sequence of characters on the waterfall. The text can be a brief mode identifier or some user specified text. You can use a small font that always appears as a 2 character wide sequence or a larger font that can be 1 to 4 characters wide. You should be aware that the video signal is a constant energy signal and the content will be spread across multiple characters. The highest s/n at the receiving end will be for 1 character wide video. Small font at 2 character width is next in s/n performance followed by 2 character large font etc.

Cw postamble

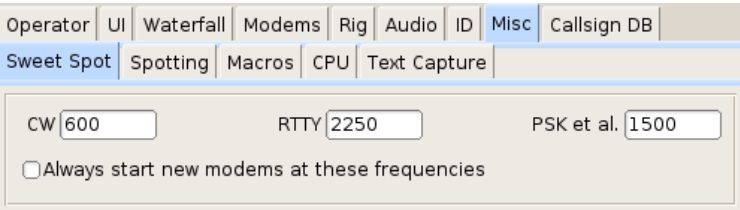
You can transmit your callsign in CW as a postamble to all modes except of CW (a bit redundant to do that).

Reed Solomon Identifier

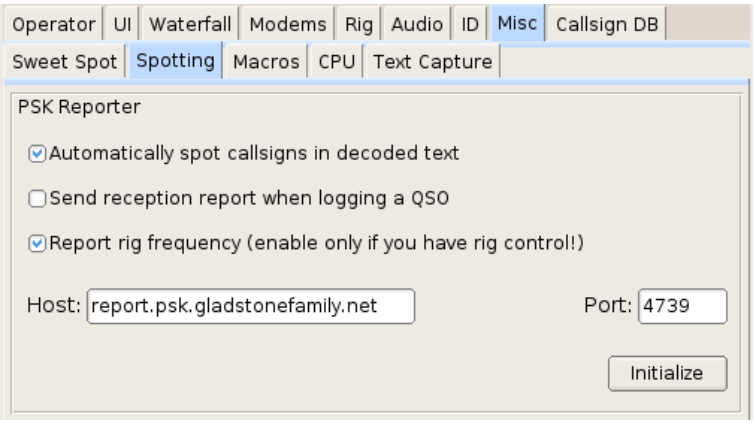
RSid, Reed Solomon Identifier, is a special transmission designed by Patrick Lindecker, F6CTE, for the modem program MultiPsk. It has been adapted to other modem programs. Fldigi's implementation is compatible with the MultiPsk RSid, but provides a slight variation. If you enable the transmission of RSid by selecting that check box., it will occur at both the beginning and the end of a transmission. The detection of RSid normally only occurs in the near vicinity of the current waterfall tracking point. This cuts down on extraneous RSid detections when the band is crowded and several RSid signals might be present. If you want fldigi to search the entire waterfall for RSid signals you can do so by enabling the "Wide Search Detector". You start the search for a signal based on RS Id by using the main panel switch. You MUST click this button to disable the RS Id search if so RS Id signal is discovered. Fldigi will not transmit while the RS Id button in lit. RS Id Search is turned off when a signal is found. When this occurs the waterfall cursor will move the signal, the mode will change if necessary, and signal decoding will begin.

Contents

Miscellaneous Configuration



The sweet spot is the audio frequency at which your transceiver provides the best filtering for a particular signal type. You can specify the value of the sweet spot for CW, RTTY and all others. You can also elect to have the audio cursor placed at the sweet spot when changing modes. The sweet spot is used for the QSY function.



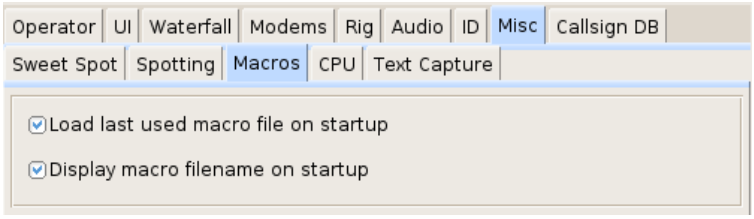
Fldigi allows you to automatically participate in a spotting network maintained by Philip Gladstone. You can see what the web based reporter looks like by visiting this web site: http://pskreporter.info/pskmap?W1HKJ or by simply selecting the menu item "Help / Reception reports..."

Fldigi will continuously scan for spotted callsigns in the decoded text and send reports in the background if you check the "Automat..." option.

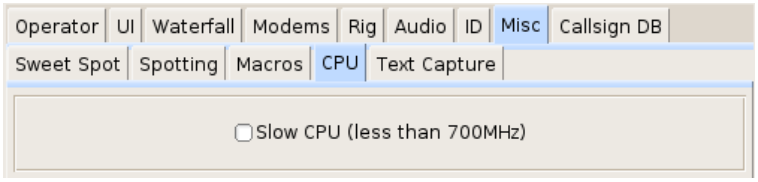
Reports will also (or only) be sent when you log the QSO into the logbook.

If you have rig control enabled the reported rig frequency will also be sent to the spotting network. Do not change the Host and Port numbers unless these are changed by Philip.

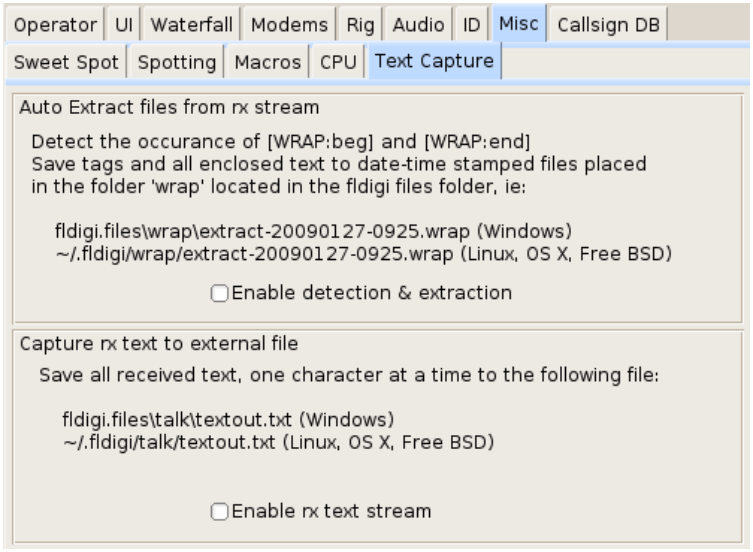
You need to press the Initialize to begin reporting spot information. You will receive a warning message if you did not enter your antenna information on the Operator tab.



Fldigi manages multiple files that contain macro definitions. You may want to have the last used macro file be the one available the next time you start fldigi. If so, simply enable the "load last used Macro file on startup" check box. You can also choose to display which macro file was loaded at startup or when a new macro file is loaded. A brief message indicating which file was loaded will be written to the Rx text area if this option is selected.



When fldigi is executed for the first time it does some tests to determine the performance factor for your central processor unit. If it determines that the cpu is below a critical speed it tries to compensate by modifying some of its timing and algorithms. If you are using a "slow" cpu the "Slow cpu" check box will be enabled. You can also manually check this box if you find that fldigi is not performing well on some of the more esoteric modes such as PSK250, MFSK32, etc.



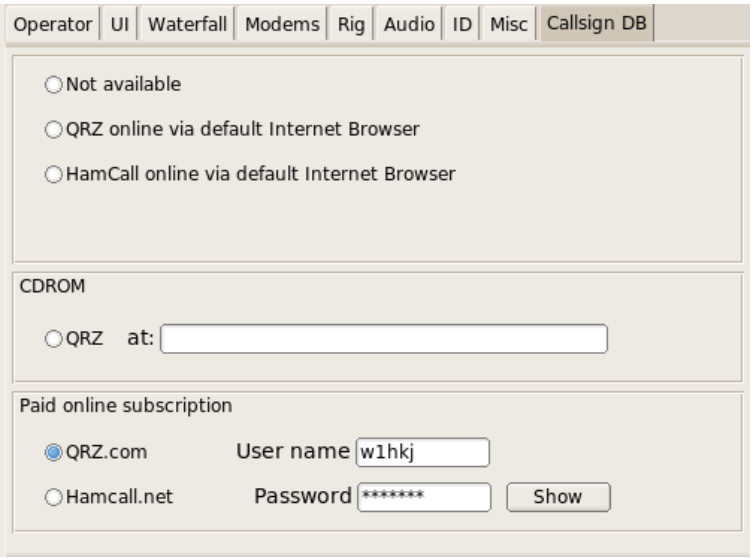
Fldigi can perform automatic capture of the Rx text stream. The simplest is to simply capture all incoming text to a file. Select this from the lower of the two frames. The Rx file is named "textout.txt" and is written to the directory as shown above. The file can be used to review an execution session, or it can be accessed by an external program. For example it could be parsed to provide a text to speech conversion.

The NBEMS suite of programs, fldigi, flarq and wrap provide the emergency operator with a set of tools to assist in the transfer of data files over HF and VHF radio. Additional information on flarq is available here: flarq help system. The program wrap is described here, wrap help system. The reception of a wrapped file can be automated by selecting the "Enable detection & extraction" option. Broadcast messages on fixed channels such as are used by MARS can be received and stored without operation intervention. The wrap program can then be

used to test for validity and data extraction at some later time.

Contents

Callsign DB Configuration



Fldigi will open a web browser to either QRZ.com or Hamcall.net with the contents of the QSO Call field used as a query string to the on line service. You may find that your default browser needs to be triggered twice on the first such query. That behavior seems to be associated with IE7 but not IE6 for example.

If you have a CD with the QRZ database installed you can use that CD or its' stored contents on a hard drive. Simply specify where the CALLBK directory can be found and enable the QRZ radio button.

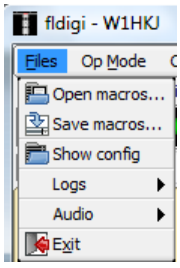
If you are a paid subscriber to either QRZ or Hamcall xml database service then you can specify that fldigi use that service for all Callsign data base queries.

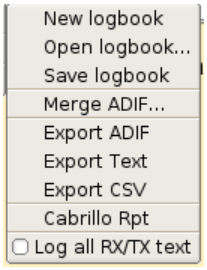
Contents

Menus

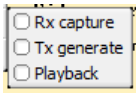
The menu heirarchy is:

- Files
  - ♦ Open Macros - open a macro definition file ... changes the MACRO keys immediately
  - ♦ Save Macros - save the current macro definitions to a designated file
  - ♦ Show config - open the OS native file explorer to the folder containing the fldigi operating files
  - ♦ Logs
    - ◊ Create a new logbook
    - ◊ Open an existing logbook
    - ◊ Save the current logbook
    - ◊ Merge current log with an ADIF file from another source
    - ◊ Export selected or all logbook records to an ADIF formated file - see Log Exports.
    - ◊ Export selected or all logbook records to a text file suitable for printing
    - ◊ Export selected or all logbook records to a tab delimited file
    - ◊ Create a Cabrillo contest report. - see Cabrillo Reports.
    - ◊ write all received and transmitted text to the file "fldigi.log" which will be in the $HOME/.fldigi directory



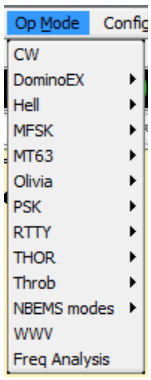


  - ♦ Audio
    - ◊ Rx Capture - allows capturing the incoming audio to a wav file
    - ◊ Tx Generate - allows capturing the generated tx audio to a wav file
    - ◊ Playback - playback a previously captured or generated wav file

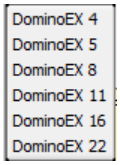


  - ♦ Exit - exit the program closing down the various interfaces in a nice controlled manner.

- Op Mode - the current operating mode will show as a highlighted menu item.



- CW - receive CW 5 to 200 WPM and transmit on any audio frequency using AFCW
- DominoEX

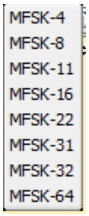


  - dominoex 4
  - dominoex 5
  - dominoex 8
  - dominoex 11 - the default calling mode for dominoEX
  - dominoex 16
  - dominoex 22

- Feld

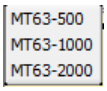
  - Feld-Hell
  - Slow-Hell
  - Feld-Hell X5
  - Feld-Hell X9
  - FSK-Hell (also called FM-Hell by some programs)
  - FSK-Hell105
  - Hell-80

- MFSK



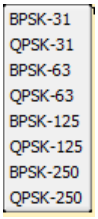
  - mfsk 4 (4 tones
  - mfsk 8
  - mfsk 11
  - mfsk 16
  - mfsk 22
  - mfsk 31
  - mfsk 32
  - mfsk 64

- MT-63



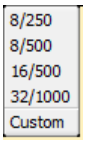
  - MT63-500 - interleave & extended characters set on configuration tab
  - MT63-1000 "
  - MT63-2000 "

- PSK



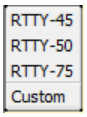
  - psk 31 - phase shift keying - 31.625 baud
  - qpsk 31 - quadrature phase shift keying - 31.25 baud
  - psk 63 - phase shift keying - 63.25 baud
  - qpsk 63 - quadrature phase shift keying - 63.25 baud
  - psk 125 - phase shift keying - 126.5 baud
  - qpsk 125 - quadrature phase shift keying - 126.5 baud
  - psk 250 - phase shift keying - 253 baud
  - qpsk 250 - phase shift keying - 253 baud

- Olivia



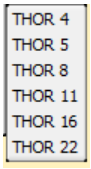
◊ 8/500 8 tone, 500 Hz wide signal format
◊ 16/500 16 tone, 500 Hz wide signal format
◊ 32/1000 32 tone, 1000 Hz wide signal format
◊ Custom - tones and bandwidth configurable on Olivia tab

♦ RTTY



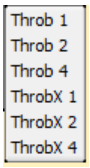
◊ RTTY-45 45 Baud Baudot, 170 Hz shift, used primarily in U.S.
◊ RTTY-50 50 Baud Baudot, 170 Hz shift, used primarily in Europe
◊ RTTY-75 75 Baud Baudot, 800 Hz shift
◊ Custom - Baud Rate, Baudot/ASCII, Shift etc configurable on RTTY tab

♦ Thor



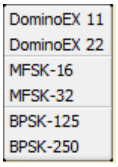
◊ Thor-4
◊ Thor-5
◊ Thor-8
◊ Thor-16
◊ Thor-22

♦ Throb



◊ Throb1
◊ Throb2
◊ Throb4
◊ ThrobX-1
◊ ThrobX-2
◊ ThrobX-4

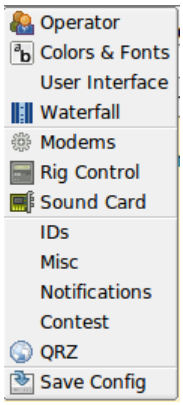
♦ NBEMS modes



These are the recommended modes to use when fldigi is used with flarq to form the NBEMS, Narrow Band Emergency Message System.

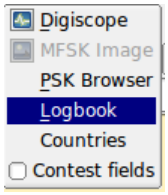
♦ WWV - special receive only modem used for calibrating sound card
♦ Freq Anal - used for carrier detection and frequency measurement

• Configure



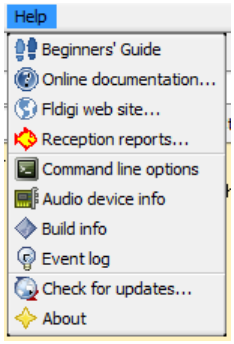
♦ Operator - open the operator configuration tab
♦ Colors - Fonts - select the colors and fonts for various main dialog controls
♦ User Interface - configure various aspects of the main fldigi dialog
♦ Waterfall - open the waterfall configuration tab
♦ Modems - open up modem configuration to the current modem tab
♦ Rig Control - open the rig control configuration tab
♦ Sound Card - open the sound card configuration tab
♦ IDs - configure various IDentification signals that can be sent and received by fldigi
♦ Misc - open the miscellaneous configuration tab (sweet spot definitions)
♦ QRZ - open the QRZ/Hamlog access configuration tab
♦ Save Config - write the current configuration to the file ~/.fldigi/fldigi_def.xml

• View



- ♦ Digiscope - Opens up a resizeable, moveable scope display
- ♦ MFSK Image - Opens the MFSK picture image (if being received)
- ♦ PSK Browser - open the psk viewer dialog to display up to 30 simultaneously decoded psk signals
- ♦ Rig Control - Opens up a dialog for controlling the transceiver interface. NOTE: this menu item is not visibile if program is configured to use the docked rig control.
- ♦ Logbook - Opens up the logbook dialog
- ♦ Countries - Opens a dialog which displays the DXCC list
- ♦ Contest fields - Display alternate 2nd line in qso logging area; provides access to contest logging fields

• Help



- ♦ Beginners' Guide
- ♦ On line documentation... - open up default browser to the on-line Help site
- ♦ Fldigi web site... - open up default browser to the www.w1hkj.com primary web page
- ♦ Reception reports... - open up browser to the http://pskreporter.info web page preset to your callsign

---

- ♦ Command line options - display a list of all command line switches available to the fldigi user
- ♦ Audio device info - displays information about all audio devices detected on the computer system
- ♦ Build info - displays all relevant information regarding the compilation and link for the application - build info
- ♦ Event log - opens a text display window that records various events depending on the level of reporting depth selected. This is a useful window for reporting problems with the program to the developers.

- ♦ Check for updates... fldigi silently opens a download web site, checks and reports on whether a new version is available.
- ♦ About - Version number and a little about the programmers

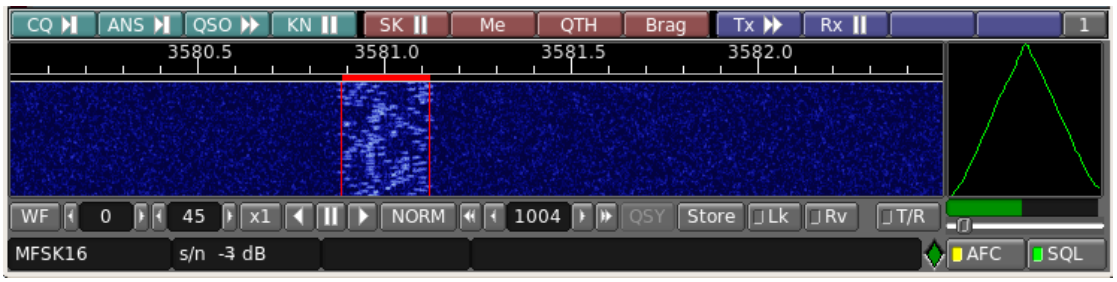
Spot button - The "Spot" light button is visible if callsign spotting is enabled. Use this button to toggle the callsign spotting reporter on and off. It is automatically turned off when playback is selected in the Files menu. The main window text is not searched if the viewer is active, i.e., if it is displayed and the current modem is PSK. See PskReporter and Notifier.

RSID button - Pressing the "RSID" button puts fldigi into a special Reed Solomon Identification detection mode.

Tune button - Pressing the "Tune" button causes fldigi to insert a tone at the current waterfall frequency. The peak-to-peak amplitude of this tone is the standard by which you should set your transmitter drive or adjust your antenna matching network.

Contents

Operating Controls & Displays



The main display for fldigi is the waterfall display shown above in color and in scale x1. The above display shows fldigi configured by invoking the following command line switches:

fldigi -bg2 black -fg white -bg grey40 --wfall-height 150 --wfall-width 3000 --font sans:12

The macro button colors are set to the default on the colors-fonts dialog.

You don't have to remember all of those switch settings every time you start fldigi. Just enter them on the Command Line, Launcher tab for the desktop icon properties (Gnome desktop).

Or from Windows XP on the Target Line, of the Shortcut tab for the properties dialog associated with the fldigi desktop icon. The fg, bg and bg2 specification on Windows is not the same as Linux. On Windows those three command line parameters need to be:

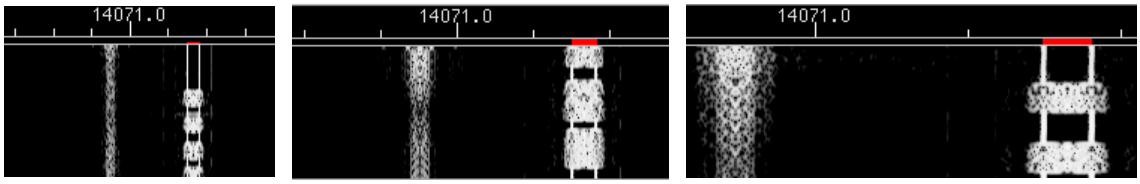
-bg2 FFFFFF -fg 000000 -bg 606060

where each color is specified by its RRGGBB component as a hexadecimal value.

The button Wtr toggles the display between a waterfall and a spectrum display. This button acts as a rotary. Left clicking moves the display selection in one direction and right clicking in the other direction. The three display modes are Wtr - waterfall, FFT - spectrum (Fast Fourier Transform) and Sig - oscilloscope time domain. Let the mouse cursor hover over any one of the controls and a small hint box will open to help you navigate the various controls.

The Norm button controls the speed of the waterfall drop. This is also a rotary type of button control. The speeds available are SLOW, NORM, FAST and PAUSE. The load on the cpu will be directly proportional to this selection. If your cpu is slow you might want to select the SLOW or PAUSE option for the waterfall.

The scale control (X1, X2, X4) expands or contracts the view into the fast fourier transform that is displayed on the waterfall or the FFT display. fldigi always computes the FFT to a 1 Hz resolution, and displays the results according to the scale control.



X1 scale X2 scale X4 scale

The next three controls are positional conrols for the waterfall. The waterfall can display 4096 data points, where each one can be thought of as a spectral line at the equivalent Hertz. The ratio is actually 8000/8192 and is related to the ratio of sound card sampling rate to Fast Fourier Transform length. This ratio changes for some modems that require a sampling rate other than 8000 Hz. The left arrow key will shift the display to the right (displays a lower section of the spectrum). The right arrow key moves the display higher in frequency. These two buttons are repeating buttons. Hold them down and the display slews at about 20 shifts / sec. The center button with the two vertical block lines is a "center the signal" button. The current cursor (red signal cursor in the waterfall) will be centered in the display area. NOTE: these controls are only functional if the current waterfall or spectrum view is smaller than the full view available. This is usually the case when the X2 or X4 expansion is selected. But it also might be the case when the width of the main dialog is reduced so that the waterfall display does not extend over the entire available width.

Try moving the cursor around in the waterfall area. You will see a set of yellow cursor blocks that show the center point and bandwidth of the current operating mode (psk31 = 31.25 Hz for example). To capture a received signal just click near the signal and the AFC will perform a multi-step acquisition. This will be very fast and should not require additional operator intervention. Casual tuning You can take a look at any received signal on the waterfall by right-clicking and holding the mouse button on or near the signal. The modem will begin to decode that signal if it is in the currently selected mode. The text will be a unique color on the Rx text widget so that you can discern the difference between casual and normal tracking. Release the mouse button and the tracking returns to the previously selected normal tracking point.

Audio History Fldigi maintains a history buffer of the received audio. This buffer is approximately 2 minutes in duration. After tracking commences on a signal you can decode the audio history for that signal. The audio history is invoked by a Ctrl-Left click anywhere on the waterfall. You can also invoke the audio history for the casual tuning mode by pressing Ctrl-Right click on the waterfall.

The next control is your transceive audio frequency. In the display above you can see that the audio signal is 1679 Hz. The red cursor is centered beneath 14071.679 Mhz. The transceiver was set to 14070 Mhz. The arrow key pairs move up/down in cycles and tens of cycles. You can fine tune the receive point using this control.

The next two controls to the right of the audio frequency control are for the receive signal processing. The one that reads -10 is the max signal level for the waterfall/spectrum display. The one that reads 51 is for the range over which that control will display signals. Both of these are in dB. The default of -10 / 40 is a good starting point, but you need to adjust these for band conditions. You can see the impact of these controls most easily by putting the main display area in the spectrum mode. Changes in these controls will effect the waterfall instantly and for all past history displayed on the waterfall. You do not have to wait for new signal data to observe the effect.

The QSY button is very specific to rigs interfaced with either hamlib or the memory mapped i/o. Each rig has a sweet spot associated with its bandwidth controller. For the Argonaut V this is 1100 Hz. For the the Kachina it is 1000 Hz. As the transceivers bandwidth is changed the changes occur centered at this frequency. So .... let's say that I just started copying a rare dx at 1758 Hz and I wanted to put the signal at the sweet spot so I could easily narrow the receiver bandwidth. Click on the signal on the waterfall. Let the AFC capture and then press the QSY button. The tranceiver frequency will be shifted and the fldigi audio tracking point shifted in unison such that the signal is now at the receivers sweet spot. Very fast and very convenient! If you do not have hamlib enabled for your transceiver this button will be dimmed and not activated.

The M> button allows you to store, recall and manage mode/frequency pairs. If you want to save the current

mode and frequency simply left click the button. A right click will enable a popup menu from which you can select a previously stored set. You can quickly move between modes and audio sub carrier using this technique. A shift-left click will clear the memory. When the popup menu is visible you left click on an entry to select it. You can shift-left click on an entry to delete that single entry.

The T/R button should be self-explanatory. It's your transmit/receive button. Action is immediate, so if you were transmiting some text and hit the button the PTT is disabled, the transmit text area cleared and the program returned to receive mode. The T/R button is a "lighted button" that shows RED when transmitting. All other lighted buttons show YELLOW when they are in the active state.

The Lck button locks the transmit audio frequency to its present value. You can then continue to QSY around your transmit position. I have used this to reply to a DX station that wanted a +500 Hz response. The DX was at 690 Hz audio, and wanted a response at +500. I moved the display cursor (or the audio frequency control) to 1190 Hz. Hit the Lck button and then went back to 690 with the waterfall cursor. Now the program is receiving on 690 Hz and transmitting on 1190 Hz. Caught him on the first try. Use this button also as a Master Station control. Not all rigs are equal in their VFO performance. Some exhibit a shift between receive and transmit. If this occurs then the stations find themselves chasing each other with every t/r exchange. Locking your transmit frequency with this control will inhibit that from happening. Be sure to disable the control when that qso is over or you may forget and transmit over top of another qso!

If the "Lck" is enabled the TX frequency does not follow the AFC action applied to the RX frequency.

For transceivers which are either hamlib or memmap enabled, if the "Qsy" button is pressed BOTH the RX and TX frequencies are changed to synchronize to where the RX was positioned.

Perhaps some numbers will help to make that a little clearer.

"Lck"	Before "Qsy"		After "Qsy"	
	RX	TX	RX	TX
OFF	1002 / 7071.002	1002 / 7071.002	1500 / 7071.002	1500 / 7071.002
ON	1002 / 7071.002	1000 / 7071.000	1500 / 7071.002	1500 / 7071.002
ON	1000 / 7071.000	1800 / 7071.800	1500 / 7071.000	1500 / 7071.000

With "Lck" off the TX audio frequency is always synchronized with the RX frequency.

With "Lck" on the TX audio frequency is fixed with respect to the RX frequency UNLESS the "Qsy" button is pressed in which case it shifts to the RX frequency, the Transceiver VFO is shifted and both the RX and TX audio frequencies are shifted to put both into the middle of the transceiver passband. The TX continues to be locked, but at the new audio frequency.

If the "Lck" is ON moving the cursor around will ONLY EFFECT the RX frequency and NOT the TX frequency.

The AFC and SQL buttons enable or disable the respective function in the software. The slider just above the AFC & SQL controls is the squelch level control. The bar indicator just above it is the equivalent of received signal level and relates on a 1:1 basis with the squelch level slider. The SQL button illuminates YELLOW when the SQL is selected, but the signal is below the squelch level. It illuminates GREEN when the the SQL is selected and the signal is above the squelch level.

The indicator just to the left of the AFC button is the overload indicator. It will be GREEN if your audio drive to

sound card is satisfactory, YELLOW if the audio signal is marginally high and turn red when it is in overload. Back down the mixer control or the audio pad from the rig to computer. Fldigi will not perform well if the sound card is over driven. You will see ghost signals on the waterfall and the modem decoders will not work correctly.

Receive audio level should be adjusted so that the overload indicator does not illuminate red. When observing the received signals on the oscilloscope view you should expect that they do not exceed a peak-to-peak amplitude of 3/4 of the full display height.

Mode Status Indicators

The lower left corner of the main display (MFSK-16) in the view above is actually a button disquised as a status panel. This button responds to the mouse in several ways:

- Left Click - opens a quick pick list of associated modem types; you can switch to a new modem type from this popup menu
- Right Click - opens the configuration dialog at the tab associated with the current modem type
- Scroll Wheel - rotates forward and backwards through the various modem types in accordanced with the modem menu heirarchy. Stop at the one you want and you are now in that mode

The next status indicator to the right provides information relative to the current modem, for PSK it indicates the received signal strength in dB.
The third status indicator from the left provides additional information relative to the current modem, IMD for PSK measured in dB.
Note that for PSK these values are only measured during periods when the PSK idle signal is being received.

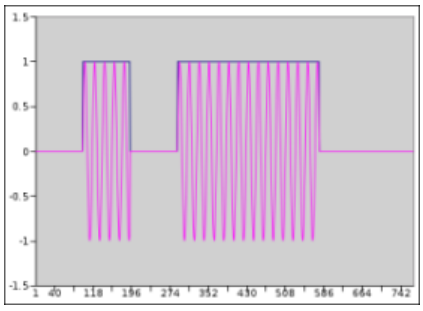
Contents

CW Mode

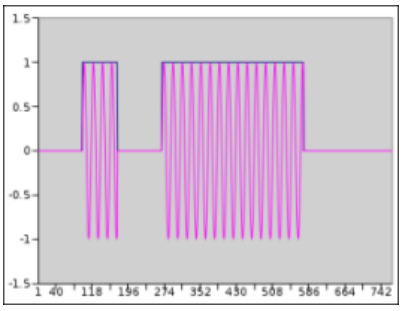
Fldigi generates CW using AFCW (A2).

AFCW, A2 is generated by tone insertion into a transceiver operating in either USB (preferred) or LSB mode. The actual transmitted signal will be at the USB carrier + the audio frequency, or the LSB carrier - the audio frequency. If fldigi is tracking and receiving a CW signal on the waterfall your transmitted signal will be exactly on the frequency of the other operator. The CW generated this way has a nearly ideal attack and decay time, controlled by the software modem. But ... a caveat ... your transmitter must never be overdriven and it should have excellent opposite sideband suppression. Overdriving the transmitter can cause multiple audio signals within the SSB passband, and cause unwanted interference to other ops. The same is true for a poorly designed or adjusted transmitter with bad sideband suppression. I recommend having a trusted and knowledgable operator assist you when first trying A2 CW. Have them carefully look for evidence of your signal above and below your primary signal (by at least +/- 3 Khz). If there is no evidence of extra signals then your are set to go. If there is you might want to have the transceiver adusted for sideband suppression, or check to be sure you are not over driving the audio.

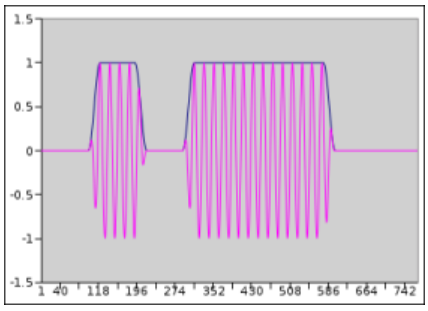
This is what the A2 signal should look like with various settings of weight, Dash/Dot and Edge. The audio frequency is 400 Hz and the TxWPM is 100 WPM.



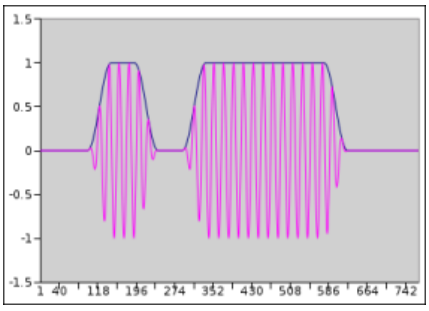
Dash/Dot = 3.0, Edge = 0.0



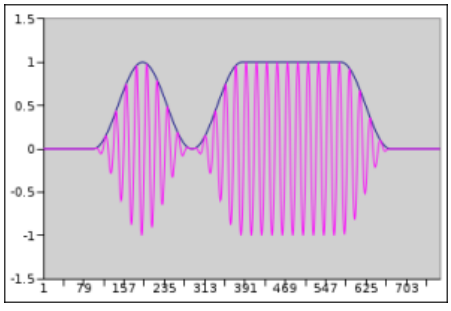
Dash/Dot = 4.0, Edge = 0.0



Dash/Dot = 3.0, Edge = 3 msec



Dash/Dot = 3.0, Edge = 6 msec



Dash/Dot = 3.0, Edge = 12 msec

Changing the weight, dash/dot or edge of the waveform does not change the WPM at which the code is generated. When a conflict occurs between the various settings WPM takes first priority, and Edge second. In the above examples, the Edge setting could not exceed 12 msec even if the control were set higher than 12.0. The figures were generated by capturing the output data being sent to the sound card and then formatting it using Gnumeric. An oscilloscope photo of the signal is virtually identical.

The setting for inter-character and inter-word spacings are fixed at 3 and 7 respectively. The 3 is achieved by sending a silent period of 1 dot (element) length at the beginning of each character and 2 at the end of each character (shown in the figures). This silent period is sufficient for most transceivers to respond to the PTT signal which occurs at the beginning of the transmission so that the first dit or dash is not lost in transmission.QRQ (high speed CW operation)

You may wonder why fldigi can go as high as 200 WPM. It's hard to believe but there are CW operators who can decode 100+ WPM in their head. These operators also usually operate QSK (full breakin). A2 CW and PTT operation and QRQ/QSK are not a natural mix. But fldigi can be used for this type of operation if an external keyer is used. For that purpose the A2 Tx output from fldigi is full wave rectified and detected to create a keyline control. The outboard conversion from A2 to keyline requires a nearly square wave pulse output of audio at the CW keying rate. Setting the Edge control to 0.0 and then the audio frequency to about 1000 Hz provides the needed signal to effect this type of keyline control.

If you are operating QSK with a separate receiver / transmitter you can very quickly stop your transmit signal with the TAB key. In the CW mode only the TAB key causes the program to skip over the remaining text in the transmit text buffer. The text that is skipped will be color coded blue. The program remains in the transmit mode (PTT enabled), but since the buffer is now empty no A2 CW signal is generated. Code transmission will then restart with the very next keyboard closure of a valid CW character. The Escape and Pause/Break keys still can be used to respectively abort and pause transmission.

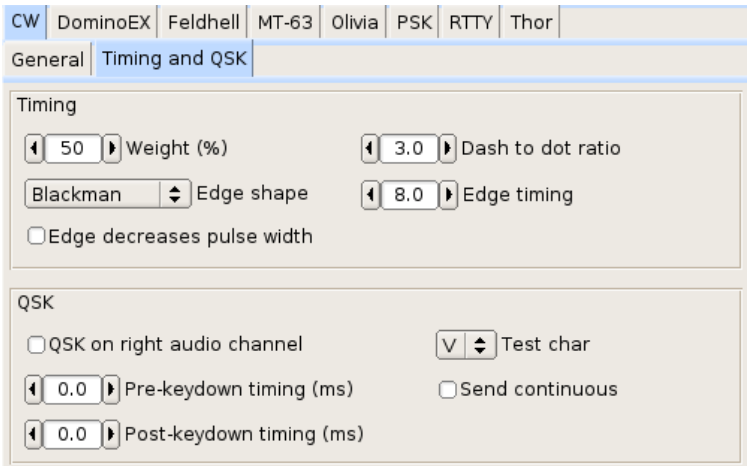
CW Prosigns

The prosigns available in the CW mode are:

PROSIGN	KEYBOARD	DISPLAYED AS
AA	~	<AA>
AR	}	<AR>
AS	%	<AS>
HM	^	<HM>
INT	&	<INT>
SK	>	<SK>
KN	<	<KN>
VE	{	<VE>

Contents

Using the FLdigi Qsk(Rt. Channel) function to key a transmitter



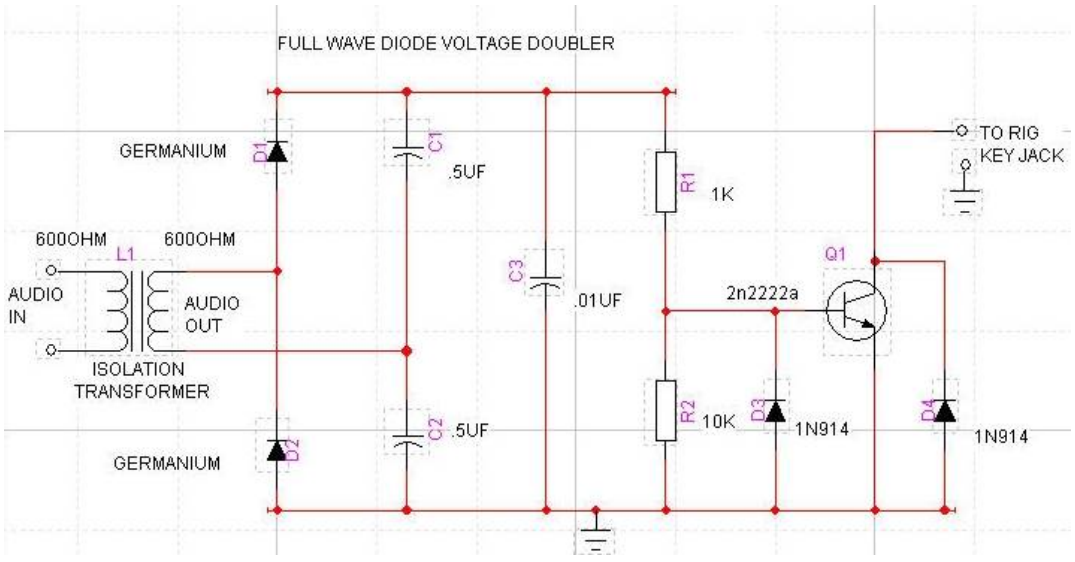
Click on the Configure menu item to open the fldigi configuration dialog. Click on the Modems tab and then on the QSK tab.

1. set the Pre Timing and Post Timing to zero,
2. activate the QSK feature by clicking "QSK on right channel"

Click the CW tab and adjust the CW settings to your preference. Use the Test char and the Send continuous controls to ease the adjustment process.

FLdigi is now ready to generate a 1600 hertz CW tone on the right channel of the stereo audio out of your sound card. The left channel will be the normal raised cosine shaped CW wave form that you may use for your side tone.

The following circuit may be used to take the FLdigi QSK OUT signal from the right channel of your SOUND CARD to key your transmitter or a QSK circuit.



NOTE:

L1 - Radio Shack has two items that may be used for this isolation transformer.

- Catalog # 270-054
- Catalog # 273-1374

Attach an audio cable from the Rt. Channel out of the your computer's SOUND CARD to the input of this QSK INTERFACE CIRCUIT (input of L1).

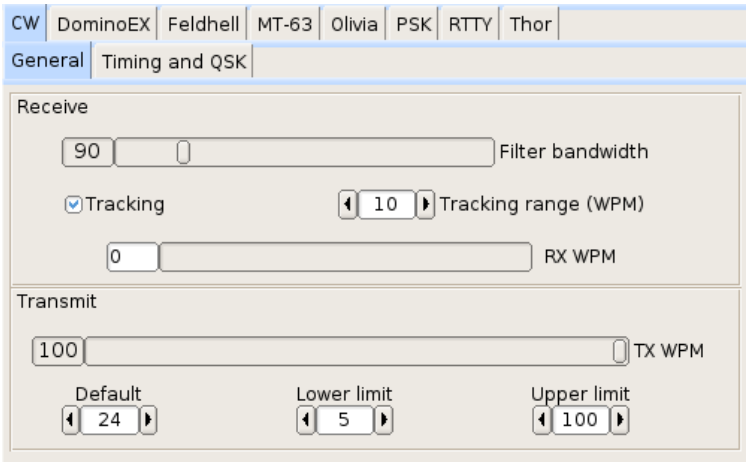
Attach another cable from the output of this circuit to your Rig's Keying Jack.

Every CW tone that is generated by FLdigi is rectified by this FULL WAVE VOLTAGE DOUBLER circuit. The resultant voltage turns the Q1 transistor on and "grounds" the collector, which takes the RIG'S CW KEYING JACK to ground and "keys" the transmitter.

You can adjust the start and stop timing of the QSK circuit relative to the CW waveform with the "pre" and "post" settings.

Contents

CW Configuration



Fldigi can send and receive morse code from 5 wpm to 200 wpm. The operating controls for CW are found on the Config/CW tab. You can open that tab by selecting the "Configure/Modems" menu item and the clicking on the Modems/CW tab. You can also open up the CW tab by first selecting CW as the operating mode and then clicking on the left-most item "CW" on the status bar at the bottom of the fldigi main window. During operation the Rx and Tx WPM settings are annunciated on the status bar in the two boxes next to the mode indicator.

The CW decoder has a DSP filter than is implemented with a sin(x)/x impulse response. This is a very steep sided filter that is centered on the received signal in the audio passband. You can control the bandwidth of this filter using the BW control.

Fldigi can track the incoming signal. Enable Rx WPM tracking by enabling the check box "Enable Tx Trkg". The tracking range (+/- Hz around the TxWPM setting) can be set using the "Rx Trkg Rng" control.

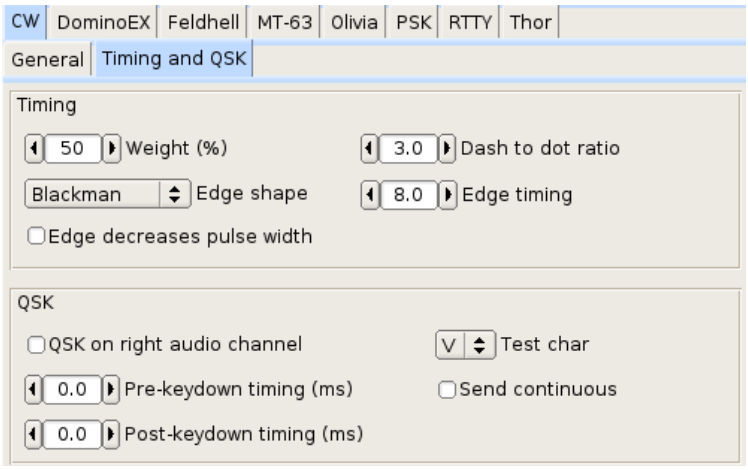
The RxWPM controls are indicators and are not used for setting the operation of the CW decoder.

The TxWPM sliding controller is used to set the transmit WPM. To make the setting easier two additional controls are provided. "Lower" sets the lower limit of the slider and "Upper" sets the upper limit of the slider. The resolution of the TxWPM slider is 1 WPM. The Lower/Upper controls are in in 5 WPM increments.

The transmit encoder settings for WPM can also be adjusted with three hot keys:

- Numeric keypad "+" increases the TxWPM by 1
- Numeric keypad "-" decreases the TxWPM by 1
- Numeric keypad "*" toggles between the selected TxWPM and a default WPM

The "Default" control on the CW tab sets that default value. As shown above the TxWPM is 30 and the default is 18. If during a QSO you needed to slow down to give the other op a better chance to copy what you are sending, just hit the "*" on the numeric keypad and the cw code will immediately switch to sending cw at the set default value (18 wpm in this example). Press the "*" again to return to back to the cw speed that you were previously using.



Fldigi generates CW using AFCW (A2). AFCW, A2 is generated by tone insertion into a transceiver operating in either USB (preferred) or LSB mode. The A2 signal is completely generated in the software so it is possible to control many aspects of the CW signal.

- Wt % control sets the weight of the CW. Normal CW is at 50% weight, ie: a dot is equal to the interval between dots or between code elements. It has a range of 20 to 80 percent.
- Dash/Dot controls the relative weight between a dash and a dot. The standard for CW is 3 to 1. The dash is 3 times the length of a dot. Some operators prefer the sound of either a heavier or lighter sounding CW. This control can be adjusted from 2.5 to 4.0 in 0.1 increments.
- Edge shape provides two leading/trailing edge shapes (1) Hanning, or raised cosine, and (2) Blackman a modified raised cosine with a steeper attack and decay. Both of these edge shapes give a more narrow bandwidth CW signal than the traditional exponential waveform. They are very easy to listen to even at speeds exceeding 100 wpm.
- The Edge control sets the rise and fall times of the CW waveform. It can be set anywhere from 0.0 to 15.0 milliseconds in 0.1 millisecond increments. DO NOT operate A2 CW with the control set below 4 msec. This is the control that sets the effective bandwidth and sound of your CW. If the edge is too steep you will have a clicky signal and be the bane of the CW bands. The purpose of being able to set the edge to 0.0 or a very quick rise/fall time is explained below. A good setting for nice sounding CW at 40 WPM and below is 4 to 6 milliseconds.
- Edge decreases pulse width, when checked will give a slightly narrower dot length as the edge timing is increased. This is useful when operating QSK and listening between the character elements.

QSK

You might ask why fldigi doesn't simply provide a keyline output on one of the parallel port pins or on RTS or DTR via a comm port. The answer is quite simple. Linux is a multi-tasking operating system and the interaction between the OS and the application causes the timing to be adversely effected. The driver implementation of the audio sub system must be responsive and so the OS gives that sub system a very high priority in its multi-tasking structure.

Many QSK operators use high speed diode antenna switching between receiver and antenna. fldigi generates a signal that can be used for that purpose. The left audio channel is always the AFCW signal. When selected the right audio channel can be configured to generate a square wave signal that begins earlier and ends later than each of the CW elements. The square wave signal can be rectified and filtered to provide the diode switching signal for the Rx/Tx antenna switching.

The right audio channel QSK signal is selected by checking the box and then adjusting the pre and post timing in

millisecond increments. See CW Keying for additional information and a circuit which converts this signal into a keying waveform.
Operating CW and additional notes on CW configuration is described in Operating CW.

Setting up a QSK device can be quite difficult. Fldigi helps to ease the adjustment by generating a continuous series of characters. This allows a dual trace scope to be properly synched while making the adjustments to both the software and the associated QSK hardware. You enable continuous characters by selecting the checkbox, and then enabling the T/R button for transmit. The repeated character can be change on the fly with the pick control. It can be one of either E, I, S, T, M, O or V.

Contents

DominoEX - Operating

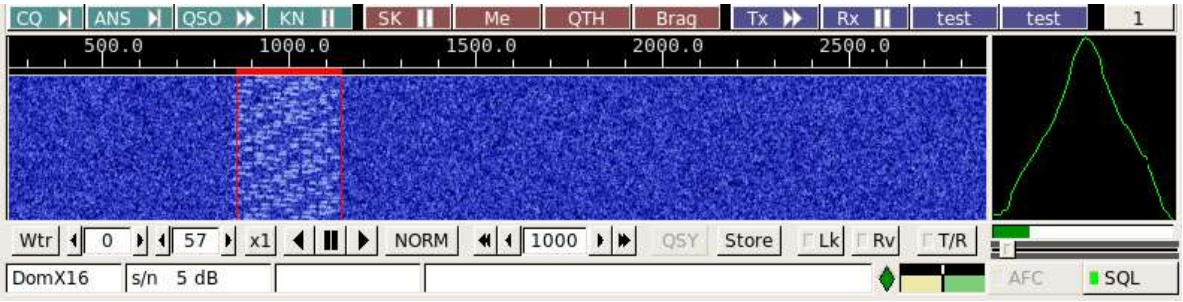
Fldigi can operate in the following DominoEX modes:

- dominoEX-4
- dominoEX-5
- dominoEX-8
- dominoEX-11
- dominoEX-16, and
- dominoEX-22

The sound card sampling rate is 8000 Hz for the 4, 8 and 16 modes. It is 11025 Hz for the 5, 11 and 22 modes. This change in sound card sampling rate will be seen in the drop rate on the waterfall. See DominoEX Technical Description.

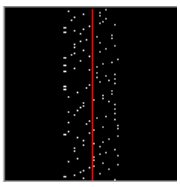
The modem code for dominoEX uses a wide band multiple frequency detector that can lock on and detect the incoming signal even when badly mistuned. Frequency domain oversampling is used to allow proper tone detection with the need for AFC. The AFC control does not alter the decoder in any way.

The waterfall and digiscope will appear as:



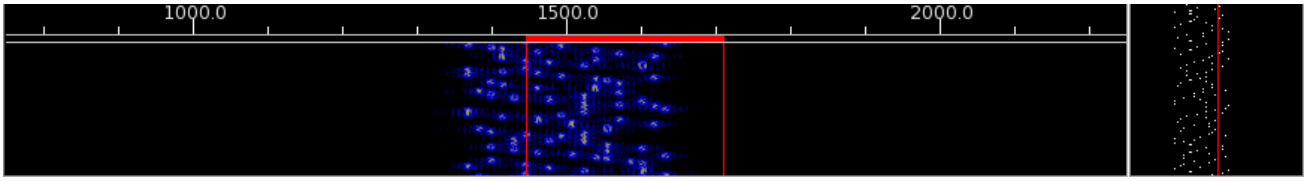
The text displayed in the status area is the secondary text being sent by the transmitting station. When the keyboard buffer is empty the dominoEX modem transmits text from the secondary text buffer. Your secondary text buffer can be edited on the DominoEX configuration tab.

The digiscope display represents the tone pairs moving through the tone filters. You can also use an alternate digiscope display (left click on the digiscope display area).

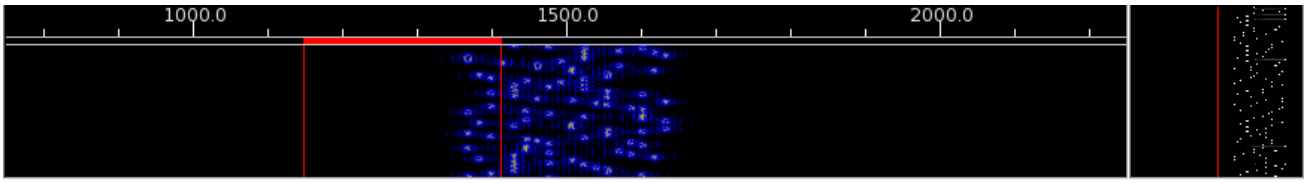


In this display mode the red line represents the center of the multiple tone bins that are in the detector. The dots will be blurry if the AFC is not locked on and become very distinct when AFC lock has been achieved. The tone dots will move from bottom to top (opposite the direction of the waterfall).

This is the same signal mistuned:

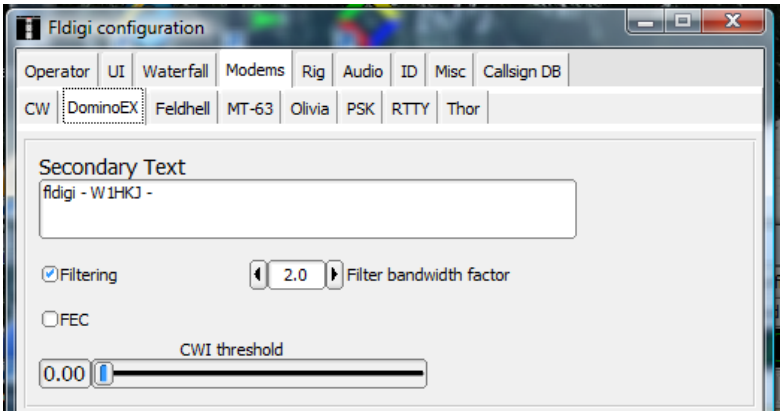


and with the signal badly mistuned:



Contents

DominoEX Configuration



Enter the secondary text. This text will be sent during periods when your keyboard is inactive (between letters for slow typists). The default for this text will be your callsign when you have entered that in the Operator configuration tab.

Set the BW factor for the decoding prefilter. 2.0 should be adequate unless you are experiencing nearby continuous wave interference (CWI). You can enable and disable the prefilter with the checkbox. Please note that the filter requires additional cpu cycles. Older and slower cpu models might give better decoding with the filter disabled.

Fldigi can send and receive FEC in accordance with the DomEX-FEC specification for MultiPsk. This type of FEC is achieved by some loss of non printing characters in the primary character set. It is therefore not usable as an FEC mode for ARQ (automatic repeat request) transmissions.

The DominoEX decoder can detect the presence of CWI within the passband set by the BW factor. Increasing the CWI threshold increases the sensitivity to such interference. When the interference is detected the associated data is culled using a technique called puncturing.

DominoEX operations are described in Operating DominoEX.

Contents

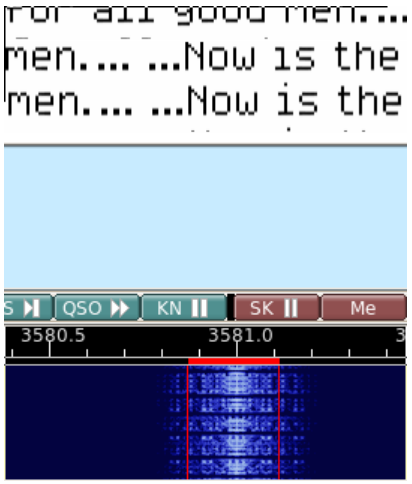
Hell Modes

All Hellschreiber modes are based on character scanning, reproducing characters in a similar way to a dot-matrix printer. This technique uses a digital transmission, yet allows the received result to be interpreted by eye, a similar concept to the reception of Morse by ear. The character is scanned upwards, then left to right. There are typically 14 pixels (transmitted dot elements) per column (although single pixels are never transmitted) and up to seven columns per character including inter-character space.

These remarkably simple modes are easy to use, easy to tune, and although not especially sensitive, are entirely suited to HF/VHF since they use no sync and the eye can discern the text even in high levels of noise. fldigi can operate in the following Hellschreiber modes:

Mode	Symbol Rate	Typing Speed	Duty Cycle	Bandwidth
Feld-Hell	122.5 baud	~ 2.5 cps (25 wpm)	~ 22%	350 Hz
Slow Hell	14 baud	~ 0.28 cps (2.8 wpm)	~ 22%	40 Hz
Feld-Hell X5	612.5 baud	~ 12.5 cps (125 wpm)	~ 22%	1750 Hz
Feld-Hell X9	1102.5 baud	~ 22.5 cps (225 wpm)	~ 22%	3150 Hz
FSK-Hell	245 baud	~ 2.5 cps (25 wpm)	~ 80%	490 Hz
FSK-Hell 105	105 baud	~ 2.5 cps (25 wpm)	~ 80%	210 Hz
Hell 80	245 baud	~ 5.0 cps (50 wpm)	100%	800 Hz

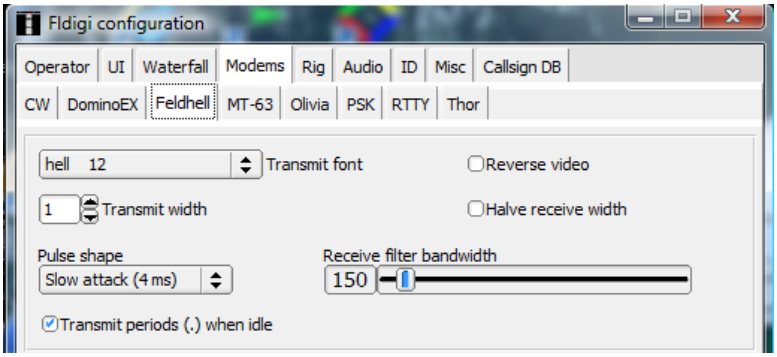
Feld-Hell seems to be the most commonly used and use can usually be found on 80 and 40 meters at the high end of the digital sub bands. All of the Hellschreiber modes are described in detail with both waterfall and sound clips in the Sight and Sounds section. Feld-Hell look like this when being received by fldigi:



Feld-Hell, Slow Hell, Feld-Hell X5, and Feld-Hell X9 are all pulse modes. Extreme linearity is required in the transmit path in order to control the bandwidth of the transmitted signal. Feld-Hell X5, Feld-Hell X9 and Hell 80 should probably not be used on HF in the US. They can be used on VHF and UHF.

Contents

Feld Hell Configuration



The Hellschreiber modes all use a video display that is basically a character-by-character facsimile. The shape of the characters is determined at the transmitting station. You can select from 15 different fonts, all of which have been designed for Feld Hell use. In normal use each dot of the character font is transmitted twice. You can increase the effective video s/n by transmitting the dots 2 or 3 times the normal. That is controlled by the Transmit width.

FeldHell is a pulse amplitude modulated signal. The shape of the pulse is a raised cosine. This helps to control the bandwidth of the transmitted signal. It is customary to use a 4 millisecond risetime for the raised cosine, especially on HF. You can change that to 2 milliseconds. The video edges will be sharper, but the bandwidth twice as large. You might find 2 msec a better choice for VHF and above.

Fldigi provides three controls for the receive function. The video is normally black on white. You can change that to white on black. You can also compress the horizontal scan rate of the video. You might find this effective in displaying received fonts that are broad (or if the other end is using a multiple dot transmit width).

As you change Hellschreiber modes the optimum filter bandwidth will be set. You can change that and might find a narrower filter effective if you are experiencing CWI interference. FeldHell is susceptible to CWI. If you narrow the filter the received video will become blurred.

Further reading on the Hellschreiber modes.

Contents

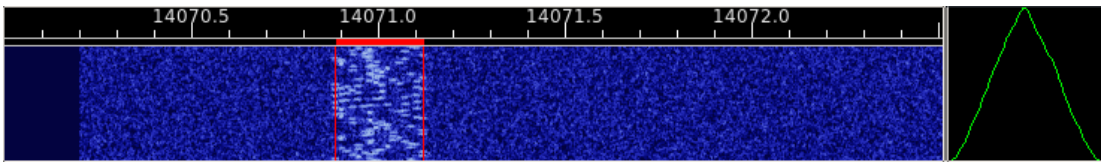
MFSK

MFSK16 and MFSK8 are multi-frequency shift keyed (MFSK) modes with low symbol rate. A single carrier of constant amplitude is stepped (between 16 or 32 tone frequencies respectively) in a constant phase manner. As a result, no unwanted sidebands are generated, and no special amplifier linearity requirements are necessary. The tones selected are set by the transmitted (4 or 5 bit) bit pattern and a gray-code table.

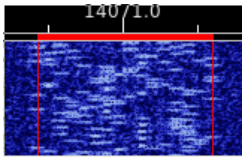
The mode has full-time Forward Error Correction, so it is very robust. Tuning must be very accurate, and the software will not tolerate differences between transmit and receive frequency. The mode was designed for long path HF DX, and due to its great sensitivity is one of the best for long distance QSOs and skeds. MFSK8 has improved sensitivity, but is very difficult to tune, and suffers more from Doppler. It is useful as the band fades out.

MFSK-32 and MFSK-64 are high baud rate and wide bandwidth modes designed for use on VHF and UHF. These are very useful for send large documents or files when some transmission errors are can be tolerated.

This is an example of properly tuned MFSK16 signal with a s/n of approximately 9 dB.



The same signal viewed with the waterfall expanded to the x2 factor.



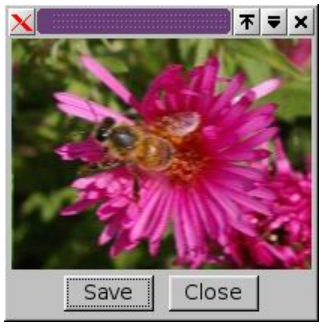
MFSK Picture Mode

Fldigi can send and receive images using all MFSK baud rates. When operating with other modem programs you should limit sending pictures to the MFSK-16 baud rate. The program can send and receive MFSK images in both black and white and in 24 bit color. The transmisssion mode for MFSKpic is similar to FAX.

Reception of an MFSKpic transmission is fully automatic. The MFSKpic transmission has a preamble sent which will be visible on the text screen. The preamble reads as "Pic:WWWxHHH;" or "Pic:WWWxHHHC;" for b/w or color respectively. The WWW and HHH are numbers specifying the width and height of the picture in pixels.

The successful reception of a MFSKpic is highly dependent on s/n conditions. The data is transmitted as an FM modulated signal and is subject to burst and phase noise on the transmission path. It can provide excellent photo transmission on a really good path.

This is an example of a photo received on a bench test. The received image is an exact replica of the transmitted image. The color depth is a full 24 bits.



Images should be carefully selected for size before beginning a transmission. To calculate the transmit time for an image use the following formula:

Time(sec) = W * H / 1000 for black and white

Time(sec) = W * H * 3 / 1000 for color

where the W and H are the dimensions of the photo in pixels. A 200 x 200 image will take 120 seconds to transmit in color and 40 seconds to transmit in b/w. The symbol rate for this mode is 1000 data bytes per second. The color image consists of 3 bytes; red, blue and green for each pixel.

This is an example of a picture received live on 80 meters (thanks K0OG)

Received images are saved in the default folder $HOME/.fldigi/images (Linux) or <defaultpath>/fldigi.files/images (Windows).

Transmitting an Image

You can only transmit an image while in the MFSK-16 mode. The image can be prepared for transmission while in the receive mode. Right click in the transmit text box and select "Send Image" from the popup menu. This will open up the transmit image dialog which will be blank to start.

Press the "Load" button and a file selection dialog will allow you to select a suitable image for transmit. The file selection dialog also has a preview capability so you will see what the image looks like.

You may also open a window manager file browser and drag and drop an image to the center part of the Send image dialog.

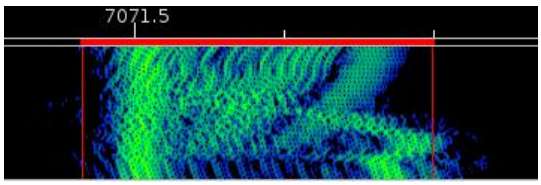
The "X1" button is a three-way toggle that allows you to transmit an image file in
X1 - normal and compatible with other modem programs
X2 - double speed, and
X4 - quadruple speed. X2 and X4 are fldigi specific image modes.

The Send image dialog after the image was drag and dropped onto the dialog.

The properties box said this image was 120 x 119 24 bit color. So it should take 42.8 seconds to transmit in full color. You can send a color or a b/w image in either color mode or b/w mode. If you transmit a color image in b/w the program will convert the image before transmiting. If you transmit a b/w image as full color you are in effect transmitting redundant information, but it can be done. I selected the "XmtClr" button for a trial run. Pressing either the "XmtClr" or "XmtGry" will put the program and the transceiver into the transmit mode if it was in the receive mode. The image is cleared and then repainted as the transmission proceeds. You see the same image progression that the receiving station should see. The main display also displays the % completion on the status bar. Hold the mouse over either the XmtClr or the XmtGry button and the tooltip will tell you the transmit time for this image.

You may abort the transmission at any time by pressing the "Abort Xmt" button. That will return you to the text mode for MFSK. You will then have to toggle the T/R button if you want to return to receive.

The receiving program decodes the "Pic:110x119C;" as a color picture 110 wide by 119 high. Here is shown being received on a computer running Vista Home Premium.



This is what the waterfall will look like during the reception of an MFSK-16 image.

The actual spectrum signature will vary with the image bytes being transmitted. The waterfall scale is in the x4 mode and the above photo was being transmitted in 24 bit color for this screenshot. The waterfall clearly shows that the image transmission is within the bandwidth occupied by MFSK-16.

Picture with a slant

If either the send, receive or both ends of the transmission are using an uncalibrated sound card whose sampling rate is not an exact multiple of 8000 Hz the resulting picture at the receive end will appear slanted. The degree of slant is directly related to the accumulation of the frequency error at both ends of the transfer. Stations wishing to send and receive MFSKpic's should calibrate their sound card. The WWV calibration mode is used to measure and set the parts per million (ppm) correction factor for the sound card.

Your sound system may be fully corrected, but the sending station may have an uncorrected sound card. You can usually correct for small errors in the following way. After the full picture is received move the mouse to bottom left or right corner of the slanted images (the corner that clearly visible). Then left click on that corner. The program will correct for the slant. The correction will not be perfect but it may help to make the image more viewable.

Contents

MT63

MT63 is an Orthogonal Frequency Division Multiplexed mode consisting of 64 paralle carriers each carrying part of the transmitted signal. The tones are differential BPSK modulated. MT63 employs a unique highly redundant Forward Error Correction system which contributes to it robustness in the face of interference and facing. The tones have synchronous symbols, and are raised cosine moduled. This mode requires a very linear transmitter. Over-driving leads to excessive bandwidth and poorer reception.

The mode is very tolerant of tuning and fldigi will handle as much as 100 Hz of mistuning. This is very important since MT63 is often used in very low Signal to Noise ratios. There are three standard modes:

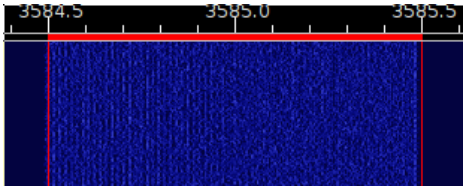
Mode	Symbol Rate	Typing Speed	Bandwidth
MT63-500	5.0 baud	5.0 cps (50 wpm)	500 Hz
MT63-1000	10.0 baud	10.0 cps (100 wpm)	1000 Hz
MT63-2000	20 baud	20.0 cps (200 wpm)	2000 Hz

In addition there are two interleaver options (short and long) which can be set on the MT63 configuration tab. The default calling mode is MT63-1000. If the short interleaver is used then one can expect some compromise in robustness. The long interleaver results in somewhat excessive latency (delay between overs) for keyboard chatting. MT63-1000 with the long interleaver has a latency of 12.8 seconds.

You can change from receive to transmit immediately upon seeing the other stations signal disappear from the waterfall. You do not need to wait until the receive text completes. Any remaining data in the interleaver will be flushed and the associated receive text printed quickly to the Rx pane. Tx will commence right after the buffer is flushed.

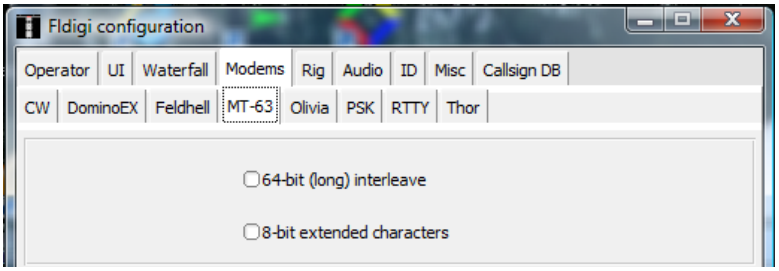
MT63 is the only fldigi mode that does not allow random placement of the signal on the waterfall. Your transmit signal, and also the received signal should be centered at 750 Hz for MT63-500, 1000 Hz for MT63-1000, and 1500 Hz for MT63-2000. If you click on the waterfall to move the tracking point it will be restored to the required position.

The default mode, MT63-1000, looks like this on fldigi's waterfall.



Contents

MT63 Configuration



MT63 is an orthogonal frequency division multiplexed mode consisting of 64 parallel carriers each carrying a part of the transmitted signal. There are 3 bandwidths and baudrates that fldigi implements in MT-63:

500 Hz - 5 baud,
1000 Hz - 10 baud, and
2000 Hz - 20 baud.

The lowest frequency transmitted is always 500 Hz. If you have a scheduled MT63 qso or are trying to copy what you think is MT63 you should tune the signal so that the lowest observable signal is at 500 Hz. Long interleave and 8-bit extended characters are usually agreed upon before a qso exchange begins. The default is to use the long interleave. 8-bit extended characters allow the transmission of Latin-1 accented characters.

MT63 is used extensively in the Military Affiliate Radio System (MARS).

Contents

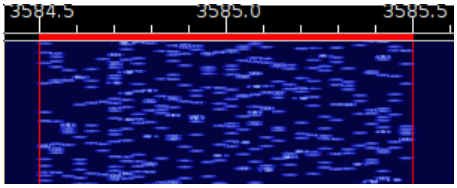
Olivia

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

Mode	Symbol Rate	Typing Speed	Bandwidth
Olivia 8-250	31.25 baud	1.46 cps (14.6 wpm)	250 Hz
Olivia 8-500	62.5 baud	2.92 cps (29.2 wpm)	500 Hz
Olivia 16-500	31.25 baud	1.95 cps (19.5 wpm)	500 Hz
Olivia 32-1000	31.25 baud	2.44 cps (24.4 wpm)	1000 Hz

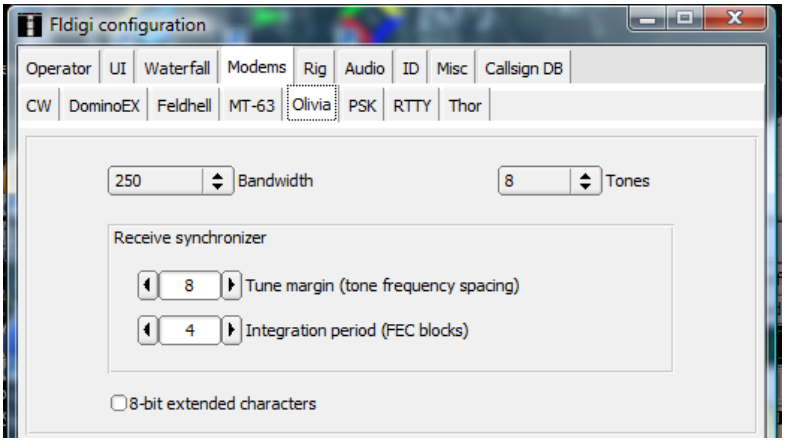
Unusual combinations of symbol rate and bandwidth can be selected using the Olivia configuration tab.

These are unconnected, simplex chat modes with full time Forward Error Correction. Olivia is a very robust mode with low error rates, but the penalty can be an annoyingly slow transfer of information. If you are a one finger typist then Olivia is your cup of tea. The tones are spaced the same as the baud rate, for example 31.25 Hz for the default baud rates. The default calling mode is 32-1000. It has the following appearance on fldigi's waterfall:



Contents

Olivia Configuration



Olivia is a family of MFSK modes with a high redundancy Forward Error Correction system similar to MT63. The family is very large, with 40 or more different options, which can make it very difficult to work out which is which. The mode works well on poor HF paths and has good sensitivity. There are three popular modes, which have 8-FSK, 16-FSK and 32-FSK, thus having three, four or five bits per symbol. These three modes can be selected without additional configurtion. The tone frequency spacing and integration period should always be left at 8 and 4 respectively unless you are experimenting with another station running an Olivia modem that can be changed. These must always be the same at both ends of the Olivia QSO. The modes have two serious shortcomings - excessive bandwidth combined with slow typing rate, and excessive latency which is the apparent typing delay caused by the integration period.

Contents

Psk

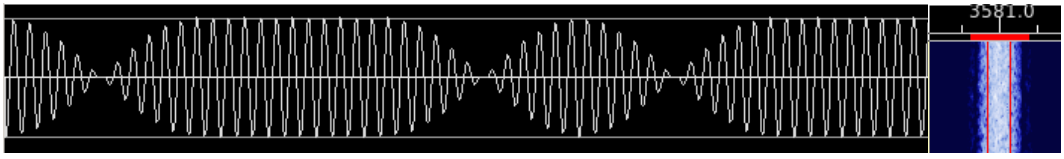
Fldigi supports the following Phase Shift Keying formats:

Mode	Symbol Rate	Typing Speed	Bandwidth
BPSK31	31.25 baud	50 wpm	62.5 Hz
BPSK63	62.5 baud	100 wpm	125 Hz
BPSK125	125 baud	200 wpm	250 Hz
BPSK250	250 baud	400 wpm	500 Hz
QPSK31	31.25 baud	50 wpm	62.5 Hz
QPSK63	62.5 baud	100 wpm	125 Hz
QPSK125	125 baud	200 wpm	250 Hz
QPSK250	250 baud	400 wpm	500 Hz

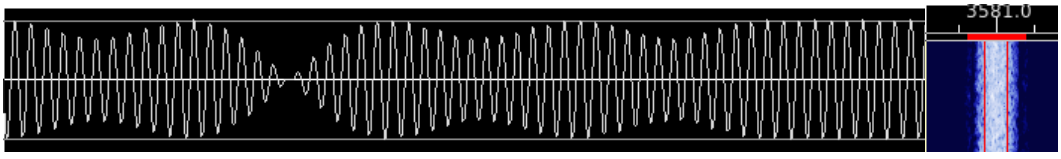
These are narrow band low symbol rate modes using single carrier differential Binary Phase Shift Kying, BPSK, or Quadrature Phase Shift Keying, QPSK. In addition to the binary phase shift keying the signal is 100% raised-cosine amplitude modulated at the symbol rate. This reduces the power to zero at the phase change.

Because of this amplitude modulation, the signal bandwidth is relatively narrow. Synchronization at the receiver is straightforward because it can be recovered from the amplitude information. Differential PSK is used to provide continuous phase changes when idle (to maintain sync), and by allowing the receiver to measure phase difference from symbol to symbol, to reduce the effects of ionospheric Doppler phase changes which modulate the signal. The slower modes are more affected by Doppler, and the QPSK modes are particularly affected.

With no interleaver and limited coding length, the QPSK mode Forward Error Correction coding gain is limited, and under burst noise conditions on HF the performance is usually worse than the BPSK option at the same baud rate. In general the narrow-band BPSK modes work well on a quiet single-hop path, but give poor performance in most other conditions.



PSK63 signal transmitting text data - oscilloscope / waterfall views



QPSK63 signal transmitting text data - oscilloscope / waterfall views

The two oscilloscope views above clearly show the combined phase and amplitude modulation of these modes.

With these modes, a very linear transmitter is required. Over-driven operation results in excessive bandwidth, poorer reception and difficult tuning. Overdrive usually occurs by having the audio signal much too large. The Sights & Sounds section has demonstrations of overdriven PSK signals. These are very sensitive modes and usually very little power is required. QRP operation of 80, 40, 30 and 20 meters can provide nearly 100% copy

over multi-hop paths. In many instances PSK can provide better decoding than CW.

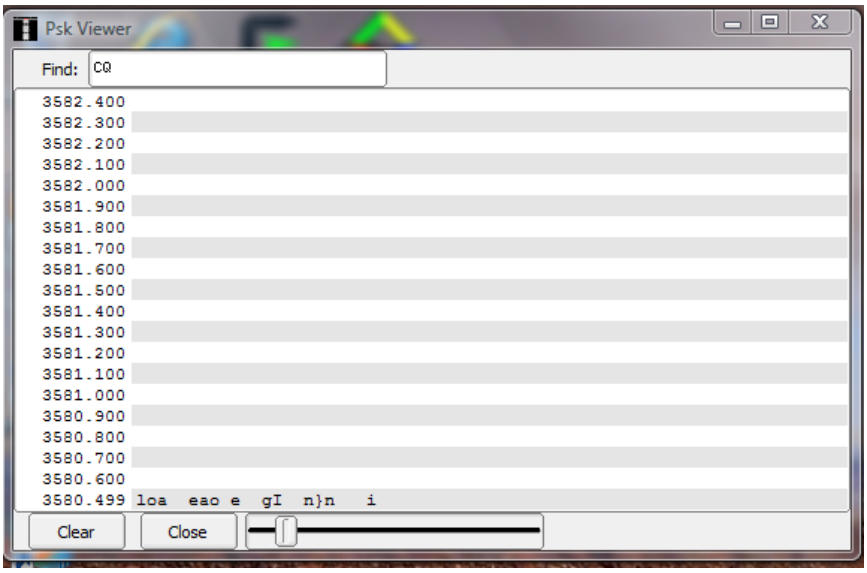
Setting up for a good clean on air signal that will receive the accolades of your qso partners is easy. Follow the instructions on using the tune button and you will have a clean on signal.

Good reception of PSK signals requires that the demodulator be phase locked to the incoming signal. Fldigi has both a fast acquire / slow tracking AFC system. Place the red bandwidth bar (see above) so that it overlies the desired signal and then press the left mouse button. The signal should quickly lock on a decoding should commence immediately. It is almost impossible to visually tell whether a BPSK or QPSK signal is being received. Under very high s/n you might be able to hear the difference, but that is even difficult for most operators. If you are not able to decode a signal that looks like a BPSK and the bandwidth of the signal matches the baud rate then it might be a QPSK signal. Just change mode a try reacquiring the signal.

Contents

Psk Viewer

The psk viewer dialog can help to locate a signal of interest on a busy band. The viewer can be visible at any time, but is only active when fldigi is in one of the psk modes. Open the viewer by clicking on the Viewer menu item.



The psk viewer can decode up to 30 simultaneous signals. As each signal is acquired within a 100 Hz channel width it is printed on the associated line. The radio frequency (or audio if fldigi is not operating with CAT) of the signal is also shown on the line. The default behavior of the line of text is to scroll in a marquee fashion after the line buffer becomes full. The size of the line buffer is controlled by the width of the display.

You can enter any text you want to search for in the "Find:" widget. This text can be a simple text snippit such as "CQ" or any regular expression.

With a regex you can specify a more generic pattern, which means that you can match more things and your search is somewhat noise tolerant. Here is an example for a CQ from a US station (should match most callsigns):

cq.+[aknw][a-z]?[0-9][a-pr-z][a-z]{1,2}

This says "cq followed by at least one character, followed by one A, K, N, or W, followed by an optional letter, followed by a digit, followed by a letter that is not q, followed by one or two letters". The search is case-insensitive.

All plain text is a valid regular expression, unless you really had been looking for these metacharacters:

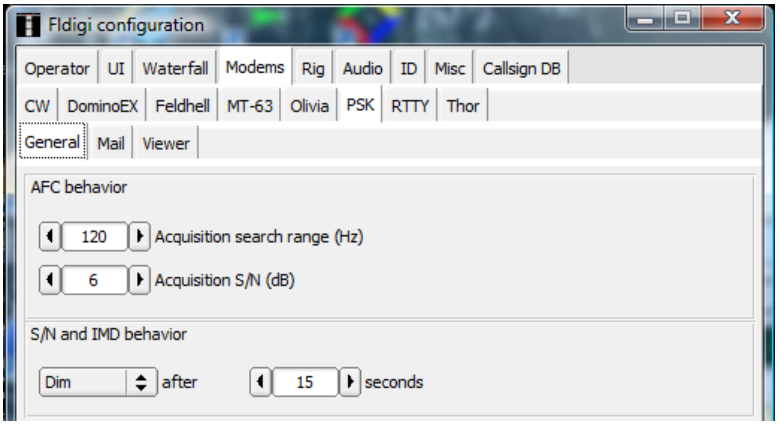
.[{()*+?|^$

These will have to be escaped with a backslash.

If you left click on a line of received text the waterfall frequency is adjusted to that value. The contents of the line of text is transferred to the Rx text widget, and the main Psk loop begins to track and decode that signal.

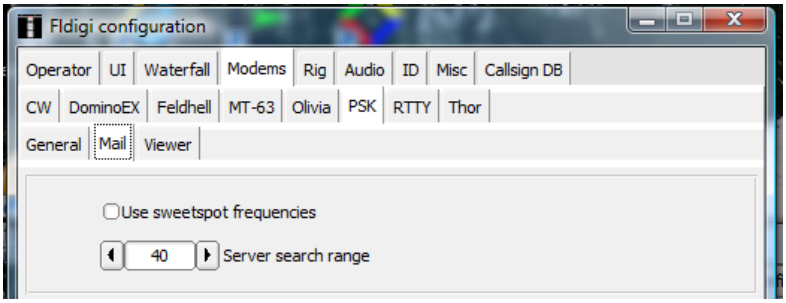
Contents

PSK Configuration

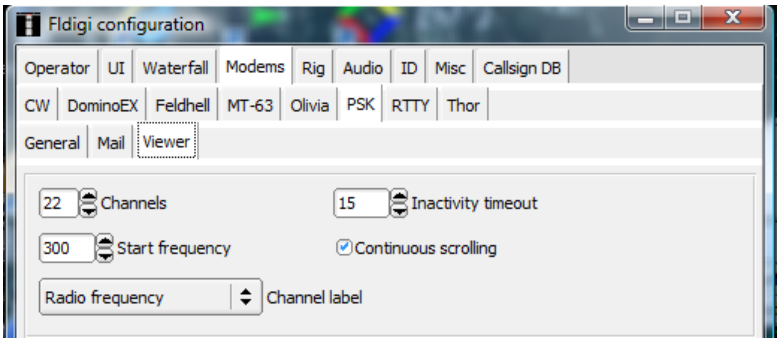


You should set the acquisition search range for waterfall left click action. As you adjust this control you will see the red mode width change on the waterfall scale. You can also adjust this value by pointing the mouse to the waterfall. Hold down the Control key and rotate the mouse wheel. The search routine which finds the psk signal operates on a s/n threshold detector as well as recognizing the Psk phase modulation. You can adjust the acquisition signal to ratio threshold for the search routine.

The PSK decoder estimates the signal to noise ratio, S/N, and the intermodulation distortion, IMD, of the received signal. This measurement is valid during periods when the other station is transmitting the idle signal. The estimates are displayed on the status bar. You can control how these values are displayed; clear or dim after NN seconds. Setting the seconds to 0 disables the clear/dim action.



Fldigi can act as both a server and a client for PskMail, a separate application from fldigi. These two factors are in support of the PskMail server. If you operate fldigi as the modem for a PskMail Server you can specify that fldigi use the psk sweetspot and that its +/- automatic signal search range be limited by the number of Hertz entered in the Server search range control.



Fldigi has a multi channel psk viewer that operates in all of the psk baud rates. You control the look and behavior of the viewer with these settings. The number of channels to be monitored can be set from 5 to 30, spaced by 100 Hz increments. You specify where to start the monitoring with the Start frequency which has a range of 200 to 1000 Hz. Each monitored signal is represented by horizontally scrolling text. Select Continuous scrolling for a marquee format. The text line will extinguish after the Inactivity timeout value in seconds is reached.

The Psk Viewer operation is described here: Psk Viewer

Contents

RTTY

fldigi can operate on a wide range of RTTY symbol rates and bandwidths. The selection of symbol rate and bandwidth is made on the RTTY configuration tab. The three most common in amateur radio use can be selected from the mode menu. These are

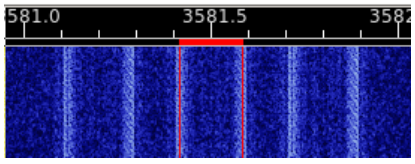
Mode	Symbol Rate	Typing Speed	Bandwidth
RTTY 45	45.45 baud	6.0 cps (60 wpm)	270 Hz
RTTY 50	50.0 baud	6.6 cps (66 wpm)	270 Hz
RTTY 75	75.0 baud	10.0 cps (100 wpm)	370 Hz

These modes were a result of mechanical and electrical designs of the early TTY machines. The 45.45 baud and 75 baud machines were for the US / Canadian market and used 60 Hz synchronous motors. The 50 baud machines were for the European market and used 50 Hz synchronous motors.

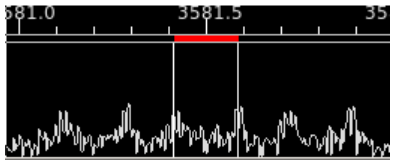
fldigi can encode and decode many other symbol rates and bandwidths. "Custom" combinations are set up on the RTTY configuration tab. You probably will never have to do that unless you like experimenting with unusual RTTY modes.

AFSK is not FSK

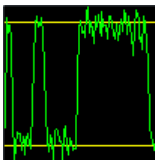
All of the modem signals that fldigi produces are audio signals. That includes the RTTY signal. fldigi can encode and decode an RTTY signal that is anywhere within the passband of the sideband transceiver. It is not limited to the traditional tone pairs around 2100 Hz. The following screen captures clearly show three side-by-side RTTY-45 signals with the middle one being tracked correctly. These signals were generated using fldigi's ability to save audio waveforms. The three RTTY signals were combined and then white noise added to create the three signals with s/n of approximately 10 dB.



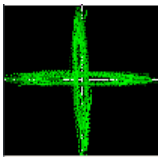
Waterfall centered at 1500 Hz



Spectrum centered at 1500 Hz

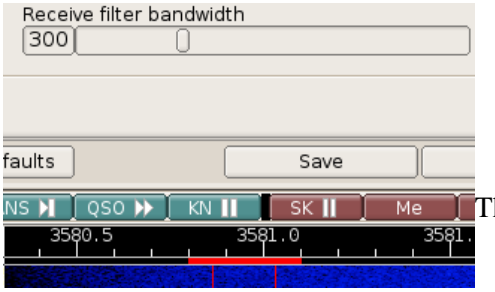


Digiscope Signal View



Digiscope X-Hair View

The decoding on any of the three signals was equal and very near 100% print. The decoder uses a hysterisis detector to help in noise burst rejection. It also uses AFC to track the signal. If a nearby CWI or RTTY signal drags the AFC you can disable it with the AFC button on the fldigi main dialog. Fldigi uses a DSP bandpass filter to reduce interference. The width of the filter can be set on the rtty configuration tab
or by positioning mouse pointer in the waterfall; pressing the control key; and rotating the mouse-wheel.



The slider and the red bar in the frequency scale correspond.

To start decoding a signal simply left click on the signal and the AFC should lock on to the signal.

DSP filtering is applied to the audio before decoding takes place. That process helps to reject nearby interference, other RTTY signals or CWI. When a baud rate / shift combination is selected the optimum filter bandwidth is computed and the Receive filter bandwidth is set to that value. You might find a different value gives better performance, especially if you are using a narrow band transceiver filter. You will need to reset the DSP filter each time you reselect the RTTY modem.

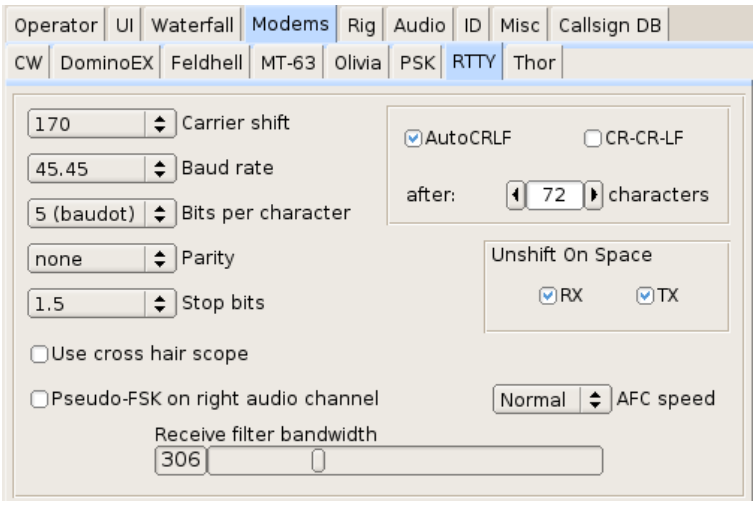
The digiscope display will extinguish when the Rx signal level falls below the squelch setting.

You must operate your transceiver in the USB mode for the RTTY signal to be the correct polarity. You must also observe the requirement to maintain linearity in the transmit path.

It is possible to use fldigi to generate the keying waveform for use with an FSK type of transmitter. See Pseudo FSK for a description of how this can be accomplished.

Contents

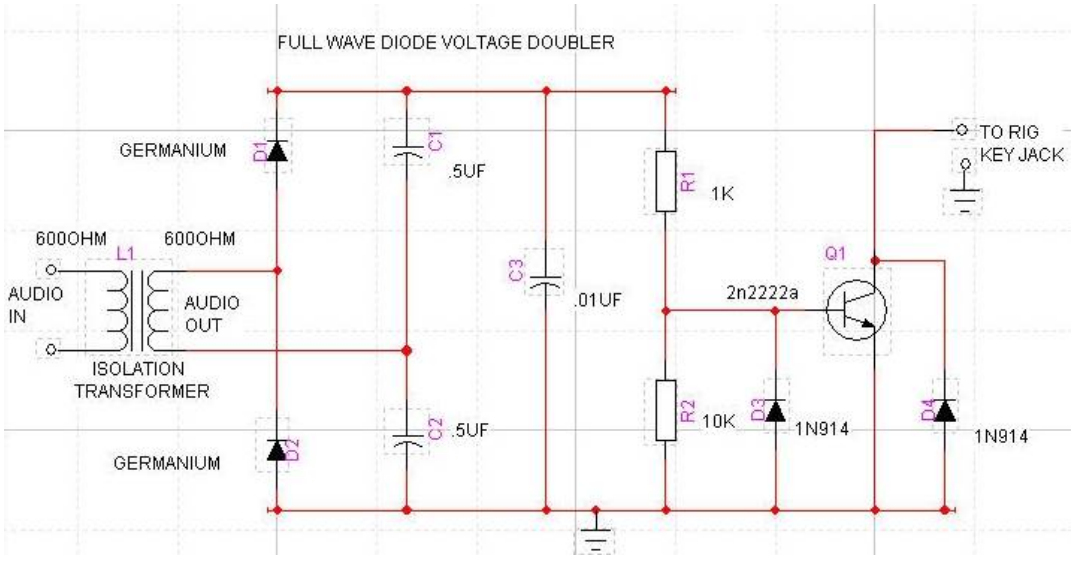
Using the FLdigi Pseudo FSK (Rt. Channel) function to key a transmitter



Select the PseudoFSK check boxes.

FLdigi is now ready to generate a 1000 hertz tone burst signal on the right channel of the stereo audio out of your sound card. This tone burst is on when the RTTY bit is on and off when the RTTY bit is off. The left channel will be the normal AFSK signal.

The following circuit may be used to take the FLdigi PSEUDO-FSK signal from the right channel of your SOUND CARD to key your transmitter's FSK input line. You may find it necessary to invert the sense of the keying signal.



NOTE:

L1 Radio Shack has two items that may be used for this isolation transformer.

Catalog # 270-054, and

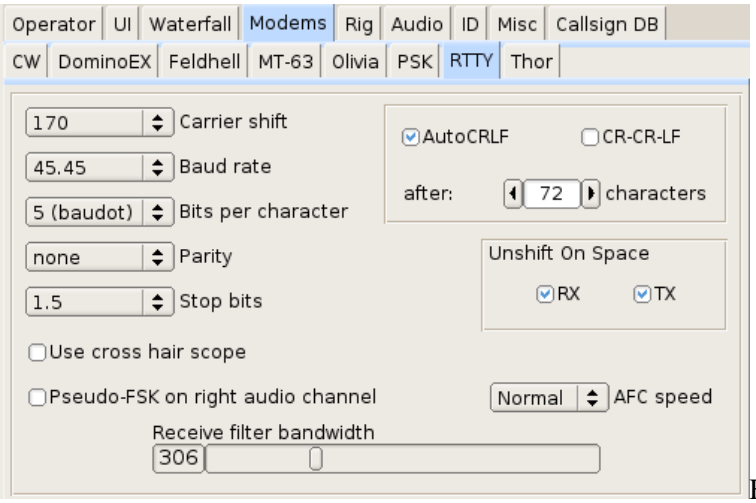
Catalog # 273-1374

Attach an audio cable from the Rt. Channel out of the your computer's SOUND CARD to the input of this FSK INTERFACE CIRCUIT (input of L1).

Attach another cable from the output of this circuit to your Rig's Keying FSK Jack.

Every PSEUDO-FSK tone that is generated by FLdigi is rectified by this FULL WAVE VOLTAGE DOUBLER circuit. The resultant voltage turns the Q1 transistor on and "grounds" the collector.

RTTY / FSK Configuration



Fldigi operates RTTY using AFSK and the transceiver set to USB. The RTTY signal can be transmitted anywhere within the USB passband of the transceiver.

You can select from various Shifts, Bauds, Bits, Parity and Stop Bits for both AFSK and FSK keying of the transmitter. You can elect to have fldigi automatically insert a CFLF when it reaches character 72 on a line. You can also have it insert a CR-CR-LF sequence instead of the standard CR-LF sequence. This is very useful if you are communicating with someone using a hardware TTY printer. The extra carriage return will give the physical device time to move to the left margin before new characters arrive.

The RTTY decoder maintains an internal AFC system for tracking the desired signal. Depending on operating conditions you may need to adjust the action of the AFC loop. Select from the Slow, Normal or Fast AFC loop. You can also disable AFC with the AFC button on the main panel.

The Digiscope display can be defaulted to the X-scope or the Signal scope. The X-scope is similar to older hardware scopes that show the Mark/Space channel signals as quadrature signals.

The DSP bandpass filter preceeds the decoder. You can adjust the bandwidth with the Receive filter bandwidth control. The bandwidth is reflected on the waterfall tracking indicator.

PseudoFSK selection generates an additional audio signal on the right channel. This signal is a burst tone at the FSK keying rate. You can full wave rectify and filter the signal so that it can be used as the FSK keyline signal to a rig that supports FSK transmissions. See Pseudo FSK for additional information and a suitable keying circuit.

Contents

Thor

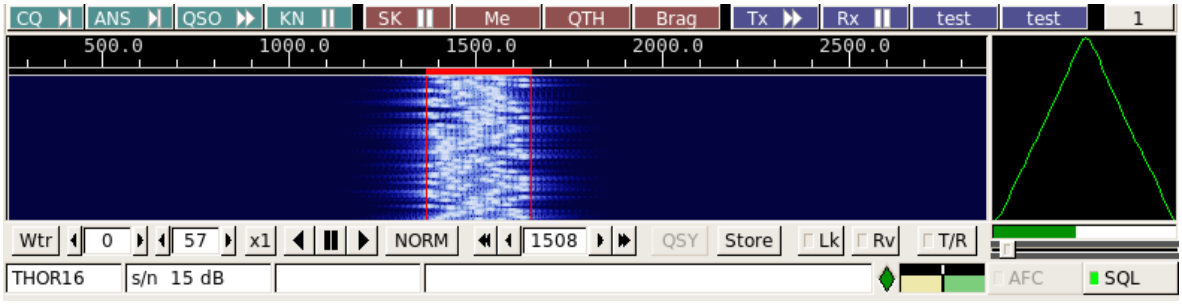
Thor is a new forward error correcting incremental frequency shift keyed communications mode. It was developed specifically to meet the needs of ARQ transfers in the HF spectrum. It is particularly well suited under conditions of atmospheric static noise. Thor borrows from two current modem technologies, MFSK and DominoEX. Fldigi can operate in the following Thor modes:

- Thor-4 - double spaced mode
- Thor-5 - double spaced mode
- Thor-8 - double spaced mode
- Thor-11 - single spaced mode
- Thor-16, - single spaced mode and
- Thor-22 - single spaced mode

The sound card sampling rate is 8000 Hz for the 4, 8 and 16 modes. It is 11025 Hz for the 5, 11 and 22 modes. This change in sound card sampling rate will be seen in the drop rate on the waterfall. See: Thor Technical Description. Thor emits a distinctive double rising tone sequence at the beginning of each transmission. It is used to flush the receive decoder and also provides a visual and audibal clue to its being used.

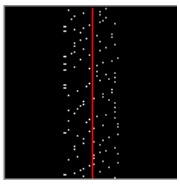
The modem code for Thor uses a wide band multiple frequency detector that can lock on and detect the incoming signal even when badly mistuned. Frequency domain oversampling is used to allow proper tone detection with the need for AFC. The AFC control does not alter the decoder in any way.

The waterfall and digiscope will appear as:



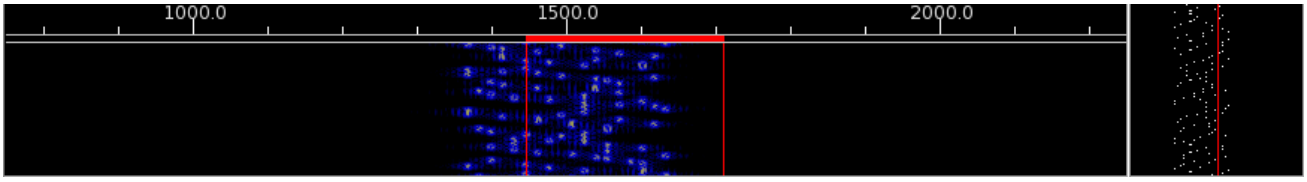
The text displayed in the status area is the secondary text being sent by the transmitting station. When the keyboard buffer is empty the Thor modem transmits text from the secondary text buffer. Your secondary text buffer can be edited on the Thor configuration tab.

The digiscope display is similar to the DominoEX display and represents the tone pairs moving through the tone filters. You can also use an alternate digiscope display (left click on the digiscope display area).

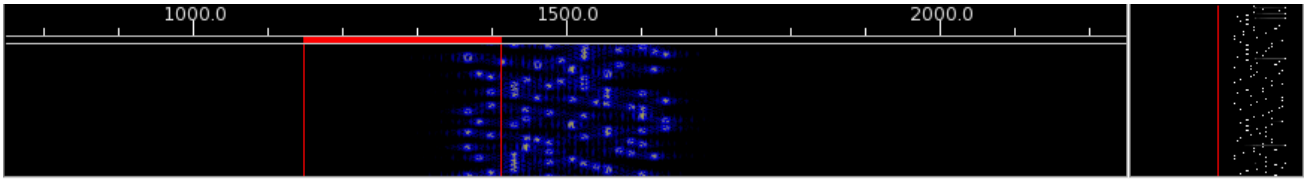


In this display mode the red line represents the center of the multiple tone bins that are in the detector. The dots will be blurry if the AFC is not locked on and become very distinct when AFC lock has been achieved. The tone dots will move from bottom to top (opposite the direction of the waterfall).

This is the same signal mistuned:

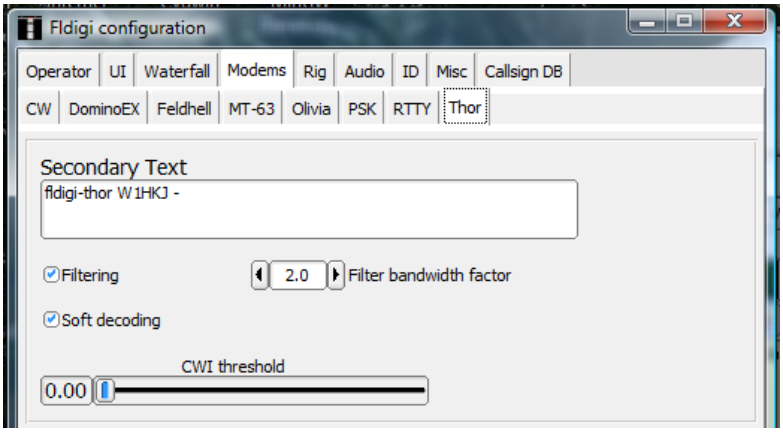


and with the signal badly mistuned:



Contents

Thor Configuration



The decoder can detect and defeat a modest amount of CWI that is within the BW set by the BW factor. Increasing the CWI threshold increasing the sensitivity for this correction. The offending tones are punctured thereby rendering them null to the Viterbi decoder.

Enter the secondary text. This text will be sent during periods when your keyboard is inactive (between letters for slow typists). The default for this text will be your callsign when you have entered that in the Operator configuration tab.

Set the BW factor for the decoding prefilter. 2.0 should be adequate unless you are experiencing nearby continuous wave interference (CWI). You can enable and disable the prefilter with the checkbox. Please note that the filter requires additional cpu cycles. Older and slower cpu models might give better decoding with the filter disabled.

Soft decode provides some additional decoder performance at the expense of more cpu cycles dedicated to the decoder.

The DominoEX decoder can detect the presence of CWI within the passband set by the BW factor. Increasing the CWI threshold increases the sensitivity to such interference. When the interference is detected the associated data is culled using a technique called puncturing.

Thor has been specifically designed to be used with ARQ text transmissions. It is also an easy to use keyboard chat mode. Thor operations are described in Operating Thor.

Contents

Throb

The THROB family of modes use two tones at a time. These tones are also amplitude modulated and can be a single tone for some symbols.

The mode has no Forward Error Correction, and is difficult to tune. It is fairly sensitive and moderately robust. Keyboard-to-keyboard QSOs are reasonably fast. Tuning must be very accurate, and the software will not tolerate differences between transmit and receive frequency.

The amplitude modulation component of THROB is a raised cosine AM modulation of each symbol. This combined with two tones transmitted at the same time, means that a very linear transmitter is required. It also gives the mode its very unique sound. You will never mistake Throb for any other mode.

For THROB, nine tones are used, spaced 8 or 16 Hz. For THROBX, 11 tones are used, spaced 7.8125 or 15.625 Hz.

Fldigi supports the following Throb baud rates and tone spacings:

Mode	Symbol Rate	Typing Speed	Bandwidth
THROB1	1.0 baud	1.0 cps (10 wpm)	72 Hz
THROB2[5]	2.0 baud	2.0 cps (20 wpm)	72 Hz
THROB4	4.0 baud	4.0 cps (40 wpm)	144 Hz
THROBX1	1.0 baud	1.0 cps (10 wpm)	94 Hz
THROBX2	2.0 baud	2.0 cps (20 wpm)	94 Hz
THROBX4	4.0 baud	4.0 cps (40 wpm)	188 Hz

Waterfall appearance and the sound of each throb mode can be seen and heard at Sight & Sound.

Contents

Digiscope Display - WWV mode

The WWV mode is used to measure the offset of the sound card oscillator. It does this by comparing the timing loop for the sound card measurements against the clock tick signal that is transmitted by WWV and WWVH. The sampling rate for the sound card is set to 8000 samples / second. The sound card is also initialized to take samples in 512 blocks. This block sampling is what sets the basic timing mechanism for the thread that reads the sound card, sends data to the waterfall, and sends data to the modem signal processing functions. A process of filtering is used that simultaneously reduces the sampling rate. A single FIR - decimation in time type is used with a total time division of 8. The resulting signal is then power detected and further filtered with a filter called a moving average filter. The moving average is very good at detecting the edge of a pulse such as the 1 second tick transmitted by WWV. This output is then displayed in a manner very similar to a FAX signal. Each scan line represents the received signal over a 1 second interval. The bright white line is the time tick. You can see a very slight slope from left to right as the signal goes from top to bottom of the display.

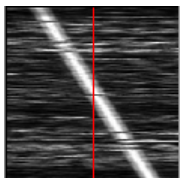
Open the configure dialog box to the "SndCrd" tab. You are going to be adjusting the "Rx corr Rate" while you observe the effect of this control on the slope of the time tick line.

Tune in WWV or WWVH on 2.5, 5.0, 10.0 or 15.0 MHz in the AM mode. This seems to give the best signal view. Select the WWV modem and allow the data to begin to accumulate in the digiscope display. When you can clearly see the bright tick line, move the cursor to the bottom of the line and left click at that position. That will resync the digiscope display and put the ensuing tick marks at the center line red graticule.

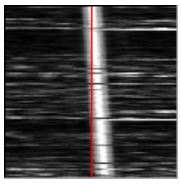
Then right click anywhere in the digiscope display. That changes the zoom level to show more detail regarding the slope of the time tick line. The zoom level increases by a factor of 5. Right clicking again restores the original zoom level. I recommend making the adjustments to the Rx corr Rate control in the x5 zoom level.

If the slope of the time tick line is positive you will need to apply a negative value to the Rx corr Rate. If it is negative then a positive correction is needed.

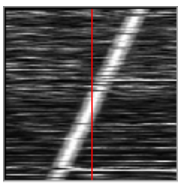
Start with a correction of 0 ppm and observe the slope. Try a value of 1000 ppm and observe the slope. Again, try a -1000 ppm correction and observe the slope. The following are some observations made on 10 MHz WWV, DCF-77 and RWM under less than ideal conditions.



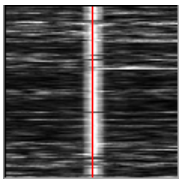
-1000 ppm WWV 5x scale



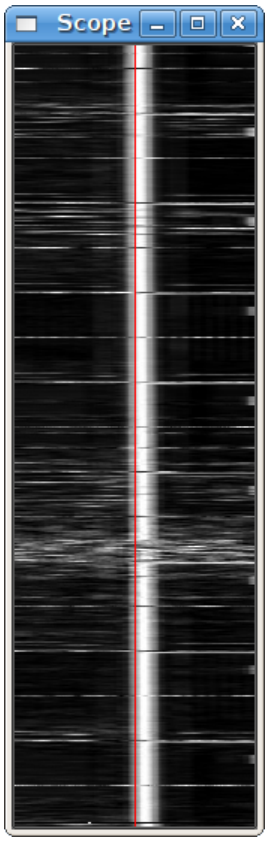
0 ppm WWV 5x scale



+1000 ppm WWV 5x scale

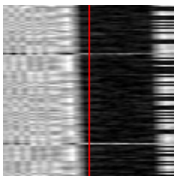


+120 ppm WWV 5x scale

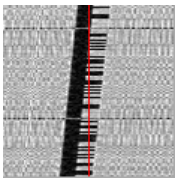


WWV corrected
20 minute trace
5x scale

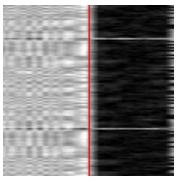
0 ppm DCF-77
1x scale



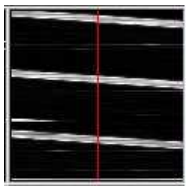
0 ppm DCF-77
5x scale



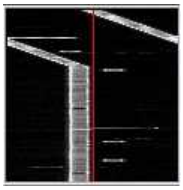
+1000 ppm DCF-77
1x scale



+65 ppm DCF-77
5x scale



RWM uncorrected
1x scale



RWM
+25361 ppm
1x scale

RWM
+25361 ppm
5x scale

You can see that my sound card requires a positive correction since the slope is negative with a 0 ppm entry. The required correction of +120 ppm was determined by guessing the needed correction to be close to 1/10 of the -1000 ppm slope and then adjusting for a steady track along the red graticule. The DCF-77 images were provided by Walter, DL8FCL. The RWM images were provided by Andy G3TDJ.

You can left click on the tick line anytime you want to recenter the signal. That will aid in making your visual observation.

When you are finished, the Rx corr Rate entry is the correct one for your sound card. Save the configuration for future fldigi use.

Andy also provided information on the RWM transmissions:

RWM details extracted from http://www.irkutsk.com/radio/tis.htm

Station RWM - Main characteristics

Location: Russia, Moscow
55 degr. 44' North , 38 degr. 12' East

Standard frequencies : 4996, 9996 and 14996 kHz

Radiated power: 5kW on 4996 and 9996 kHz; 8kW on 14996 kHz

Period of operation: 24 hours per day, except 08.00-16.00 msk for maintenance as below:
on 4996 kHz : 1st wednesday of the 1st month of quater;
on 9996 kHz : 2nd wednesday of the 1st month of the quater;
on 14996 kHz : 3rd wednesday of each odd month;

Coverage: 20 degr. - 120 degr. East
35 degr. - 75 degr. North

Time signals A1X are given every second of 100 ms duration with a frequency of 1 Hz. Minute pip is extended to 500 ms.

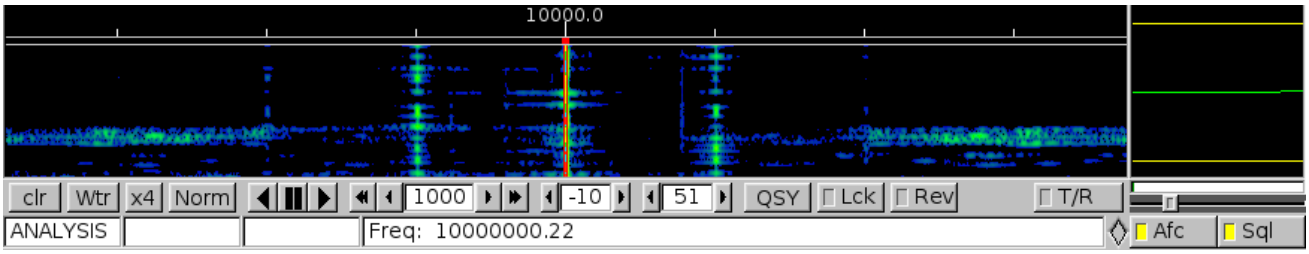
Hourly transmission schedule

m s	m s	
00:00 -	07:55	-- MON signals (no modulation)
08:00 -	09:00	-- transmitter is signed off
09:00 -	10:00	-- station's identification is sent by Morse Code
10:00 -	19:55	-- A1X signals and identification of DUT1+dUT1
20:00 -	29:55	-- DXXXW signals
30:00 -	37:55	-- N0N signals (no modulation)
38:00 -	39:00	-- transmitter is signed off
39:00 -	40:00	-- station's identification is sent by Morse Code
40:00 -	49:55	-- A1X signals and identification of DUT1+dUT1
50:00 -	59:55	-- DXXXW signals

Contents

Frequency Analyzer

Fldigi can be used to accurately measure the frequency of a remote signal that is transmitting a steady carrier.



I have set the sound card up using the WWV modem and have it adjusted for the proper PPM offset on receive. I followed the frequency calibration procedure that Icom recommends for the IC-746PRO, adjusting WWV at 10 MHz for a zero beat.

Then fldigi was used in the "Freq Analysis" mode to track the WWV carrier at 10 MHz. In this mode the decoder is merely a very narrow band AFC tracking filter. The filter bandwidth is set to 2 Hz and the tracking time constants to about 5 seconds. Future releases will probably make both of these user adjustable. When the signal is being tracked the digiscope (right hand display) will be a horizontal line. If the signal is very noisy and tracking difficult the digiscope will jump and become wavy. You can see from the above image that I am tracking about 0.22 Hz high on WWV. I have repeated this measurement at various times during the day and on various days with nearly the same result. So I am comfortable with knowing that my local oscillator is just a little low (that is why the Frequency reads high).

ARRL frequently announces a frequency measurement test (FMT) which takes placde on 160, 80 and 40 meters. This is a chance to test your skills in frequency measurement. You should be able to make a submission to the FMT using this technique. Make corrections to the FMT transmission based upon your WWV measurement. You may have to adjust for other local oscillator effects as well. If you have some good ways to measure and correct for these I would be glad to share them with the other fldigi users.

Contents

TUNE Mode

Too often you see an overdriven signals on the digital sub-bands; multiple audio sidebands on PSK, splatter from overdriven MFSK and RTTY. There is absolutely no reason for a transceiver driven by fldigi to exhibit this type of performance. You can set up your computer / transceiver for good solid performance without excessive drive.

The "TUNE" button generates a continuous single frequency audio signal at the exact frequency to which the waterfall cursor has been set. The peak amplitude of this signal is the peak amplitude of every modem signal generated by fldigi. None will exceed this value, even the simultaneous multi-tone modes like Throb. Every modern SSB transmitter uses some automatic level control ALC for preventing overdrive for SSB voice. A little overdrive on a voice channel can be tolerated to a degree. In fact, that is what an analog RF compressor does, overdrive and then subsequent filtering. But you absolutely cannot tolerate that with the digital modes. Here is the way to set up your transceiver for a clean signal. I recommend starting out with a dummy load, but an "off hour" for a band might work just as well if you do not have a dummy load.

- For Windows users
  - ♦ Set your sound card output level to the minimum on the Windows mixer
- For Linux users
  - ♦ Set your PCM level to about 80%
  - ♦ Set your Transmit Level control for minimum output level.
- Enable the "Tune" mode in fldigi ... you do have CAT or PTT set up ...right?
- Make sure your transceiver's speech compression control is OFF
- Slowly bring up the Mixer audio out until your rig's ALC just starts to function (a light blinking or a meter showing this condition).
- Reduce the Mixer audio output until the ALC is disabled.
- You are now transmitting at maximum output power without distortion.

You can use any level below this and be assured that your output signal will be clean. All digital signals that fldigi generates will be limited to this peak-to-peak voltage. You should always use the minimum power necessary to maintain good comms, remember that even if you are clean at 100 W you signal will be so strong among the QRP signals that it will overpower the AGC on many receivers that are working another digital station within the same SSB bandwidth that you are on. You will appreciate this the first time that you are working a weak PSK DX station and someone blasts through and captures your AGC.

You should try the the above adjustments at different audio frequencies. Transceivers that achieve the SSB filtering with crystal or mechanical filters will have a considerable amount of variation across the passband of the filter. This will show up as a varying amount of ALC that is dependent on the audio frequency. Once you are comfortable with the process you can very quickly repeat the "Tune" and set the power for the frequency to which the waterfall is set.

Contents

Keyboard Operation

The transmit buffer for fldigi is type ahead which means that you can be typing text while the program is sending an earlier part of your transmitted message. Newly entered text appears in black and text which has been transmitted is changed to red. You can backspace into the red area. When you do and the modem in use supports the BS character it will be sent to the receiving station. If you monitor PSK and MFSK signals you will often find operators backspacing over previously sent text. It's probably just as easy to just send XXX and retype that part of the message, but we have gotten used to word processors, email, etc. that allow us to send perfect (right) text, so we expect our digital modems to do the same. Let's see, what was that prosign often used in CW for oooops.

All of the alpha numeric keys perform as you would expect, entering text into the transmit buffer. There is one very important exception:
the caret "^" symbol. This is used in the macro expansion routine and also used by the transmit buffer evaluator. A ^r puts fldigi into receive mode. So you can enter the ^r (caret followed by the r) at the end of your transmit buffer and when the sent character cursor (red chars) gets to that point the program will clear the text and return to the receive mode.

You can load the transmit buffer with any ASCII Text file of your choice. Merely right click in the buffer window and select from the pop-up menu. You can also short cut to the ^r from this popup.

Many ops (including me) do not like to be tied to a mouse. The fldigi text widget supports some short cuts to make your life easier:

- Pause/Break - a transmit / receive - pause button.
  - ♦ if you are in the receive mode and press the Pause/Break key the program will switch to the transmit mode. It will begin transmitting characters at the next point in the transmit buffer following the red (previously sent text). If the buffer only contains unsent text, then it will begin at the first character in the buffer. If the buffer is empty, the program will switch to transmit mode and depending on the mode of operation will send idle characters or nothing at all until characters are entered into the buffer.
  - ♦ if you are in the transmit mode and press the Pause/Break key the program will switch to the receive mode. There may be a slight delay for some modes like MFSK, PSK and others that require you to send a postamble at the end of a transmission. The transmit text buffer stays intact, ready for the Pause/Break key to return you to the transmit mode.
  - ♦ Think of the Pause/Break key as a software break-in capability.
- Esc -
  - ♦ Abort transmission. - immediately returns the program to receive, sending the required postamble for those modes requiring it. The transmit buffer is cleared of all text.
  - ♦ Triple press on Esc - terminates the current transmission without sending a postamble - The PANIC button.
- Ctrl-R will append the ^r (return to receive) at the end of the current text buffer.
- Ctrl-T will start transmitting if there is text in the transmit text window.
- Alt/Meta-R will perform the same function as the Pause/Break key
- Tab moves the cursor to the end of the transmitted text (which also pauses tx). A tab press at that position moves the cursor to the character following the last one transmitted. CW operation is slightly different, see the help for CW.
- Ctrl + three digits will insert the ASCII character designated by that entry.

Function Keys

Keys F1 through F12 are used to invoke the macro F1 - F12. You can also just click on the macro key button associated with that function key. There are 4 sets of 12 macros. If you press the numbered button on the macro button bar the next set of macros are referenced by the F1 - F12. A right click on the numbered button provides a reverse rotation through the 4 sets of macro keys. The respective macro set can be made available by pressing the Alt-1, Alt-2, Alt-3 or Alt-4 key combination. Note that this is not Alt-F1 etc.

Contents

Mouse & Keyboard Shortcuts

Fldigi has a bewildering number of keyboard and mouse shortcuts, some of which may help make your particular style of operation more efficient. You do not need to know them all to make effective use of the program!

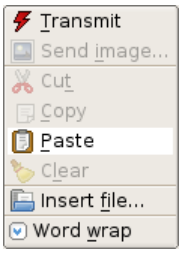
- Main window
  - ♦ Text input fields

    Most text fields use a combination of CUA (PC) and Unix-style keybindings. Text can be marked, copied, pasted, saved to a file as well as transfer to other main panel controls. A right click on any text control will open a context sensitive menu for that particular control. A full list can be found on the FLTK web site.

    The received/transmitted text widgets use CUA key bindings with some modifications:

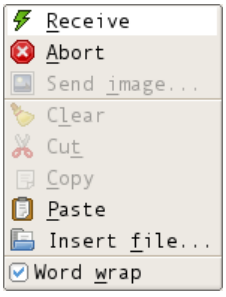
    ◊ RX text
    This widget is read-only and ignores shortcuts that would modify its contents. See logbook for details on the Rx right click popup menu system.

    ◊ TX text
    The text that has already been sent is protected, but can be deleted one character at a time with the Backspace key. Right clicking on the Tx text panel opens the following popup menu:



Fldigi Receiving

Editing Tx text



Fldigi Transmitting

Select:

Transmit	put the program into the transmit operation
Receive	during a transmit or tune, end the transmit and restore receive operation
Abort	during a transmit, receive without waiting for the modem to finish sending
Send image	for MFSK only, send an image using MFSKpic mode
Clear	clear all of the text
Cut	delete the marked text (by left click drag over text)
Copy	copy the marked text to the clipboard
Paste	the clipboard text to the current text insertion point
Insert file	select a file from file browser to insert in text at insertion point
Word wrap	turn word wrap on/off

The Tx panel is fully drag and drop aware. That means you can add a file to the transmit

text by simply opening up a file manager (different for different OS and choice of desktop). Select the file from the manager and then drag and drop it onto the Tx panel. The mouse pointer will move the cursor insert point for the drop.

A number of additional shortcuts can be found in the Keyboard Operation section.

♦ Waterfall display

Most of fldigi's unusual shortcuts are specific to this widget.

Waterfall display - Keyboard

◊ Shift Left/Right move the b/w marker by 1 Hz
◊ Ctrl Left/Right move the b/w marker by 10 Hz

Waterfall display - Mouse

◊ Left click/drag move the b/w marker to, and start decoding at the cursor frequency
◊ Right click/drag as above, but return to previous position on release
◊ Middle click toggle AFC
◊ Ctrl-Left click replay audio history at b/w marker position
◊ Ctrl-Right click replay at cursor frequency and return on button release
◊ Ctrl-Middle click copy the frequency under the cursor to the currently selected (or first) channel in the PSK viewer, and select the next channel
◊ Shift-Left click/drag same as unmodified left click; no signal search
◊ Shift-Left click/drag likewise, with a return to the previous frequency when the button is released
◊ Shift-mouse wheel move the squelch slider
◊ Scroll wheel usage is dependent upon the configuration (see ConfigWaterfall)
· None - no mouse wheel activity in waterfall panel
· AFC range or BW - adjust the AFC range/BW up/down
· Squelch level - adjust the squelch level up/down
· Modem carrier - adjust the audio tracking point +/- Hz increments
· Modem - select modem type from a full rotary of available modems
· Scroll - move the waterfall left/right in 100 Hz increments (for 2x, 4x expanded waterfall view)
· Ctrl-mouse wheel change the AFC search width in PSK modes, or the bandwidth in CW and FeldHell

♦ Waterfall "Store" button

◊ Left click Add a new item for the current frequency and modem
◊ Shift-Left click Delete all items
◊ Middle click Select last item in menu
◊ Right click Pop up menu
· Left/right click Select item (and switch to that frequency/modem)
· Shift-Left/right click Delete item
· Middle click Update (replace) item

♦ Digiscope display

◊ Mouse wheel Change AFC/BW, same as Ctrl-mouse wheel on the waterfall

- Rig control window

There are some shortcuts in addition to those described in the Rig Control

♦ Frequency display

◊ Left/Right arrow key change the frequency by one 1 Hz
◊ Up/Down arrow key change the frequency by 10 Hz

♦ Frequency list

◊ Shift-Left click delete the line under the cursor
◊ Middle click replace the line under the cursor with the current frequency/mode/modem

- PSK viewer window

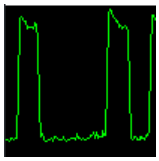
♦ Besides the bindings mentioned in the Psk Viewer section, there are mouse shortcuts to change the nominal frequency of a viewer channel:
◊ Middle click copy the current waterfall b/w marker frequency to the channel under the cursor, overwriting that channel's nominal frequency
◊ Right click restore a channel's nominal frequency
◊ Right click on Clear as above, for all channels

Contents

Digiscope Display

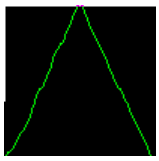
Fldigi provides several different views of the decoded signal with its waterfall, text and a scope displays. The scope display is a separate moveable, resizeable dialog that is opened from the "View/Digiscope" menu item. If you have used a previous version of fldigi you will know that the digiscope was a part of the main display and adjacent to the waterfall. You can still have the digiscope displayed that way by setting the configuration for using a docked digiscope. For those modes in which more than one type of scope view is available you can have different views on the independent digiscopes.

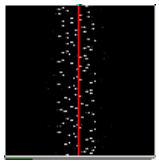
CW



The CW signal will consist of the time domain amplitude detected signal. The horizontal timing is dependent on CW speed, so that the display will appear similar independent of CW speed.

DominoEX / Thor

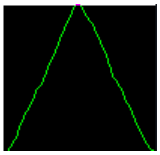




DominoEX and Thor have two alternate views available on the digiscope display. You can toggle between the views by left clicking on the digiscope display area. The triangular view shows data propogation through the interleave filter. As signal s/n degrades this display will become more wavy.

The second view is the decoded data stream viewed in the frequency domain. The dots will be very distinct when the signal is fully acquired and decoding properly. It will be fuzzy when the decoder is not locked or there is interference present.

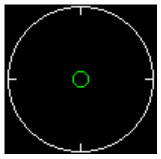
MFSK



This is what you expect to see for all of the MFSK type modes. The number of steps in the slant lines will change with the various modes, but they will all have the same general appearance. If the signal is mistuned the sloped lines will become bowed and distorted.

Psk

The digiscope display just to the right of the waterfall displays signal quality in various formats. The display for PSK modes is the vector scope:



The display with no signal or below squelch

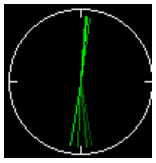
The display with a normal psk31

AFC off and receive carrier

level. If the SQL is off this display will be random vectors driven by noise.

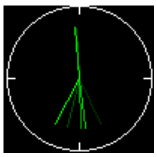
AFC off and receive carrier set above the center of the received signal.

signal. The vector flips between 0 and 6 o'clock



AFC enabled, Fading History Display Mode Selected (left click on scope)

set below the center of the received signal



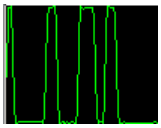
AFC enabled, Fading History / Amplitude Display Mode Selected (2nd left click on scope)

You can see the effect of mistuning by slewing the carrier carrier control moving from low to high over the signal . You must do this with AFC off. Engage the AFC and the vectors will immediately snap to vertical positions.

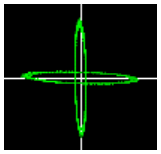
You can alter the appearance of the phase vectors by left clicking on the digiscope display. One click will give you a history of phase vectors that fade with time. A second click will give you a history of phase vectors that both fade with time and are amplitude significant. The third click returns you to the original phase vector display.

The effect is the same with QPSK signals except you will see 4 vectors that are 90 degrees from each other.

RTTY



The signal can be viewed in two different ways on the digiscope. This is the time domain representation of the detected FSK signal. The two yellow lines represent the MARK and SPACE frequencies. This display is for Baudot, 45.45 baud, 182 Hz shift. If the tranmitting station were transmitting at 200 Hz shift the signal extremes would lie above and below the yellow lines. Try tuning across the RTTY signal with the AFC disabled. You will see the signal move above and below the yellow lines as you tune. Then enable the AFC and the signal should rapidly move into the center region of the display. This signal was about 3 - 6 dB above the noise floor. It looked marginal on the waterfall but still gave good copy.



This is the other digiscope display for RTTY. You obtain this view by left clicking anywhere in the digiscope display window. You can toggle back and forth between these views. The MARK / SPACE frequencies are represented by the quadrature ellipses. When the RTTY signal is properly tuned in the lines will be in quadrature and aligned as shown. Tune across the RTTY signal and the MARK/SPACE lines will rotate around the center. If the sending station is using a shift that is smaller than you have the decoder setting then the two lines will close toward the NW/SE quadrants. If the sending station is using a shift that is greater than the decoder setting then the two lines will close toward the NE/SW quadrants.

Contents

Macros

Macros are short text statements that contain imbedded references to text data used by the program fldigi. Macro definition files(s) are located in the $HOME/.fldigi/macros/ directory and all have the extention ".mdf". The default set of macros are contained in the file $HOME/.fldigi/macros/macros.mdf. Fldigi will create this file with a set of default macros on its first execution.

Fldigi supports up to 48 macro definitions in sets of 12. Macro definitions are not recursive, that is; a macro cannot reference another macro or itself.

The imbedded references are similar to those used by DigiPan and other fine modem programs. The imbedded reference is an uppercase plain text descriptor contained with the <> brackets.

Reference definitions:

<MYCALL> my call

<CALL> remote call

<LDT> local date time Zone
format : %x %H:%M%Z

- %x is preferred short form date ie: MM/DD/YY or DD/MM/YYYY etc for your locale
- %H is hour with leading 0
- %M is minute with leading 0
- %Z is abbreviated time zone ie: EDT or GMT

<ZDT> GMT date time Zone
format : %x %H:%M %Z

- %x is preferred short form date ie: MM/DD/YY or DD/MM/YYYY etc for your locale
- %H is hour with leading 0
- %M is minute with leading 0
- %Z is abbreviated time zone ie: EDT or GMT

<QSOTIME> current log time in Zulu HHMM format

<FREQ> my frequency

<ID> send waterfall video mode identifier

<TEXT> send user specified video text

<CWID> send AFCW "DE <MYCALL>" at end of transmission

<MYLOC> my locator

<LOC> remote locator

<MODE> my mode

<MYNAME> my name

<NAME> remote name

<MYQTH> my qth

<QTH> remote qth

<RX> return to receive

<TX> start transmit

<MYRST> my RST

<RST> remote RST

<CNTR> insert current contest serial number into the text stream

<INCR> increment contest serial number

<DECR> decrement contest serial number

<XOUT> contest exchange

<INFO1> append the current data in the 1st status field; s/n for BPSK

<INFO2> append the current data in the 2nd status field; imd for BPSK

<MODEM>name cause program to change operating mode to named mode. Use the shorthand names that are in the pick list.

<LOG> submit QSO data to logbook program & clear the QSO data fields. This macro reference is not constrained to a particular position in the macro. The macro reference action takes place when macro is expanded, so effect is seen immediately after pressing the macro function key which contains this macro reference.

<IDLE:NN> causes fldigi to transmit an idle signal for NN seconds

<TIMER:NN> auto repeat this macro after NN seconds

<WPM:nn> CW words per minute = nn

<PRE:dd.dd> QSK pre-timing = dd.dd msec

<RISE:dd.dd> CW pulse risetime = dd.dd msec

<POST:+/-dd.dd> QSK post-timing = dd.dd msec

<FILE:[fname]> select this from the tag list; a file selection dialog opens for you to choose the file. Select your file to be inserted with this macro. The tag definition will be completed with the fully qualified filename inserted into the macro definition as in:

<FILE:/home/dave/.fldigi/mybrag.txt>

The specified file will be opened and the text inserted into the Tx buffer when this macro is executed.

<MACRO:[fname]> select this from the tag list; a file selection dialog opens for you to choose the file. Select your macro file to be inserted with this macro. The tag definition will be completed with the fully qualified filename inserted into the macro definition as in:

<MACROS:/home/dave/.fldigi/macros/testmacros.mdf>

The new set of macros will be opened and replace the current one when this macro is executed.

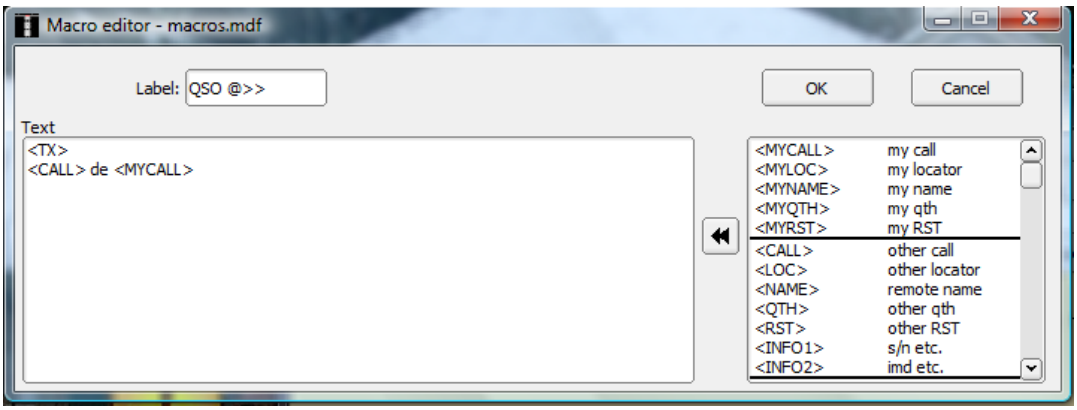
Local references are specified during the program configuration and can be changed during program operation.

Remote references are all part of the qso log field definitions and are routinely changed from contact to contact.

Global references are for items like Greenwich Mean Time.

The macros.mdf file can be edited with any ascii text editor such as kedit, gedit, geany, nano etc. But it is much easier to use the built-in macro editor provided in the program.

Right click on any macro key (or the alternate set) and a macro editing dialog opens with the current copy of that macro and its label. This looks very similar to the DigiPan macro editor at the urging of Skip Teller, KH6TY.



The Text box is a mini-editor with a very limited set of control functions. You can mark, bound and select text for deletion (ctrl-X), copy (ctrl-C), and paste (ctrl-V). Marked text can also be deleted with the delete or the

backspace keys. Marked text modification can also be invoked by using the mouse right click after highlighting.

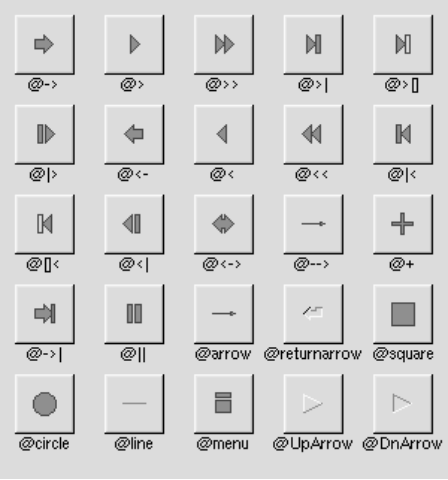
The macro reference in the pick list can be transfered to the current editing cursor location. Highlight the desired macro reference and then press the double << arrow key for each occurance of the reference to be put into the macro text. You can change the label name but any more than 8 characters may exceed the width of the button for the default sized main dialog.

The <TIMER:NN> and <IDLE:NN> macro tags should have the NN replaced with the time interval in seconds.

<TX><IDLE:5>CQ CQ de <MYCALL> <MYCALL> k<RX><TIMER:20>

- will enable the PTT
- cause 5 seconds of idle signal
- send the CQ CQ de W1HKJ W1HKJ k
- disable PTT
- and count down 20 seconds before repeating the macro
- after sending the text the count down timer button (upper right hand corner of main dialog) will display the current timer value in seconds. Press this button to disable the timer.
- the timer be disabled if the Escape key is pressed, the T/R is pressed, and macro key is pressed, or if a callsign is copied from the Rx text area to the callsign logbook entry.
- the time will be disabled if any mouse activity occurs in the waterfall control.

The label associated with each macro key can be individually annotated with a symbol. Here are the symbols that are recognized by the button label drawing routine:



The @ sign may also be followed by the following optional "formatting" characters, in this order:

- '#' forces square scaling, rather than distortion to the widget's shape.
- +[1-9] or -[1-9] tweaks the scaling a little bigger or smaller.
- '$' flips the symbol horizontaly, '%' flips it verticaly.
- [0-9] - rotates by a multiple of 45 degrees. '5' and '6' do no rotation while the others point in the direction of that key on a numeric keypad. '0', followed by four more digits rotates the symbol by that amount in degrees.

Thus, to show a very large arrow pointing downward you would use the label string "@+92->".

Here are my macro buttons suitably annotated:

There are 4 sets of 12 macro functions. You can move between the 4 sets using the keyboard and the mouse.

1. Left click on the "1" button to move to set #2. Right click on the "1" button to move to set #4.
2. Move the mouse to anywhere on the macro buttons. Use the scroll wheel to move forward & backward through the macro sets
3. Press the Alt-1, Alt-2, Alt-3 or Alt-4 to immediately change to that macro set.

- The label for CQ is "CQ @>|", denoting that both <TX> and <RX> are present in the macro text.
- The label for QSO is "QSO @>>", denoting that only <TX> is present in the macro text.
- The label for KN is "KN @||", denoting that only <RX> is present in the macro text.

You could use any label that is symbolic to the function required. Refer to the FLTK web site for a full list of label types.

If you modify the macros and do not save them ("Files/Save Macros" on the main window) fldigi will prompt you to save the macros when you exit the program if you have the "Nag me" option selected.

The <EXEC>...</EXEC> macro tag provides a way to create external shell scripts and programs that can interact with fldigi. See Exec Macro to learn more about the <EXEC> macro tag.

Contents

Exec Macro

The <EXEC> ... </EXEC> macro is designed to be used on the Linux OS as it supports fully functional pipes. Windows' version of file pipes is not fully POSIX compliant, but the function might work in the environment. Consider all that the following allows you to do from within fldigi and you might want to consider changing over to Linux.

The <EXEC> macro defines an external child process (or processes) that will be called by fldigi when the macro key is invoked.

Exported variables

Fldigi exports a set of variables to the child process and adds ~/.fldigi/scripts to the PATH variable before running the shell code. This is the directory location for all executable scripts and programs which you might want to call from within the macro. Some examples will be given later. Open the macro editor for an undefined macro key and enter the following:

- <EXEC>env | grep FLDIGI</EXEC>

Save the macro; call it ENV. Then press the newly defined macro key. All of the exported variables will be shown in the transmit window.
Here is an example of the results:

FLDIGI_RX_IPC_KEY=9876
FLDIGI_LOG_LOCATOR=FM02BT
FLDIGI_TX_IPC_KEY=6789
FLDIGI_LOG_RST_IN=
FLDIGI_LOG_FREQUENCY=3581.000
FLDIGI_AZ=108
FLDIGI_MY_CALL=W1HKJ
FLDIGI_LOG_TIME=2113
FLDIGI_MY_NAME=Dave
FLDIGI_VERSION=3.0preG
FLDIGI_LOG_NOTES=
FLDIGI_LOG_QTH=Mt Pleasant, SC
FLDIGI_MY_LOCATOR=EM64qv
FLDIGI_DIAL_FREQUENCY=3580000
FLDIGI_CONFIG_DIR=/home/dave/.fldigi/
FLDIGI_LOG_RST_OUT=
FLDIGI_MODEM=BPSK31
FLDIGI_LOG_CALL=KH6TY
FLDIGI_MODEM_LONG_NAME=BPSK-31
FLDIGI_AUDIO_FREQUENCY=1000
FLDIGI_LOG_NAME=Skip
FLDIGI_PID=14600
FLDIGI_FREQUENCY=3581000

All of the above envelope variables can be referenced in a shell script that is called from within fldigi.

Detection of existing scripts

In anticipation of a collection of useful "fldigi scripts", the macro browser contains a macro line for each executable file found in the scripts directory.

The EXEC macro allows the text that is read from the child process to be parsed for more fldigi macros. For example, try this macro:

<EXEC>cat foo</EXEC>

where foo is a file that contains:

<MYCALL>

This may have some interesting uses but, if it is undesirable, it can be suppressed with an extra layer of redirection. Instead of <EXEC>command</EXEC>, you would use <EXEC>noexp command</EXEC> where noexp is the following very simple script:

```
snip----------------------------------------
#!/bin/bash

echo -n "<STOP>"
"$@"     # run the command
r=$?     # save its exit code
echo -n "<CONT>"
exit $?
snip----------------------------------------
```

There are three additional MACRO definitions that expand the capability of the <EXEC> command: <STOP>, <CONT> and <GET>.

The <STOP> and <CONT> macros stop and resume the expansion of all <MACRO> strings. For example, <STOP><MYCALL><CONT><MYCALL> would only expand the second <MYCALL>. By wrapping the command output in this way we can be sure that no text will be expanded. You might even use

"$@" | sed "s/<CONT>//g"

if you feel paranoid.

You can "fork and forget" with an exec macro defined as: <EXEC>exec command -args >/dev/null</EXEC>

Any of the text that appears between the <EXEC> and </EXEC> can reference an executable program or shell command found in the ~/.fldigi/scripts directory.

Any text output that is returned by the program or script program (or the result of the in-line command) is always returned to the transmit buffer and appears as appended to the transmit window.

Querying an external database

The <GET> command captures returned text from the external process and parses it for the following content:

$NAMEtext_name$QTHtext_qth

If either $NAME or $QTH is present the trailing text is transferred to the LOG_NAME or LOG_QTH widgets respectively. This means that you can create a script that accesses a local or net based database of callsign data and parse that data to form the above console output. Fldigi will accept that output, parse it and populate the associated log entries. Cool! Now for some examples. Here is a perl script that performs the above for the University of Alabama on-line callsign database, ualr-telnet.

The matching macro key definition for the above is:

<EXEC>ualr-telnet.pl $FLDIGI_LOG_CALL</EXEC><GET>

which I named "ualr ?"

Google Earth Map

Here is a really cool perl script, Google Earth Mapping, that accepts the current "Loc" field in the logging area and generates a Google Earth map which is displayed in your default browser. The macro call is <EXEC>map.pl</EXEC>

Custom dates/times

You can use <EXEC> to create custom date/time entries. For example, BARTG contesters use %H%M, but in other circumstances a user might prefer %H:%M or %H.%M etc.

Create the following script file in the ~/.fldigi/scripts directory, call it mytime:

snip----------

```
#!/bin/sh
date --utc "+%H:%M"
```

snip----------

date calls strftime, the same C function used by fldigi for the ZDT/LDT expansion, so it has an equally vast number of format strings to choose from. Look for them in its manual page.

Give "mytime" execute permissions with a file manager or with chmod: chmod u+x ~/.fldigi/scripts/mytime.

Test it on the command line and make sure it works correctly: ~/.fldigi/scripts/mytime

Restart fldigi. The mytime script will now appear at the end of the list in the macro browser, and can be entered with the << button as usual. Test that macro and you will see that <EXEC>mytime</EXEC> inserts the datetime in the specified format.

Of course you could have entered:

<EXEC>date --utc "+%H:%M"</EXEC>

in the macro body text directly

Many other uses for the <EXEC>...</EXEC> macro pair can be imagined when used the with ENV parameters. For example you could send Azimuth data to an automated antenna rotor. The exported variables should be sufficient for a script writer to create custom loggers and clients.

Contents

QSO Logbook

Fldigi maintains a large set of QSO logbook fields that will probably be sufficient for casual operating, contesting and some certificate logging. All of the fields that are captured in the logbook are maintained in an ADIF database that can be read by any logbook program that can read the ADIF text format.

The complete set of logbook fields are:

ADIF FIELD		USE	ADIF FIELD		USE
BAND		QSO band (computed from frequency)	STATE	*	contacted stations state
CALL	*	contacted stations call sign	QSLRDATE		QSL received date
COMMENT	*	comment field for QSO	QSLSDATE		QSL sent date
COUNTRY	*	contacted stations DXCC entity name	STX	*	QSO transmitted serial number
FRE	*	QSO frequency in MHz	SRX	*	QSO received serial number
GRIDSQUARE	*	contacted stations Maidenhead Grid Square (Loc)	TIME_OFF	*	end time of QSO in HHMM format
MODE		QSO mode	TIME_ON	*	start time of QSO in HHMM format
NAME	*	contacted operators name	TX_PWR	*	power transmitted by this station
QSO_DATE	*	QSO data at start of contact	IOTA		Islands On The Air
RST_RCVD	*	received signal report	VE_PROV	*	2 letter abbreviation for Canadian Province
RST_SENT	*	sent signal report	XCHG1	*	received contest exchange
QTH	*	contacted stations city	MYXCHG		sent contest exchange
DXCC		contacted stations DXCC country code	ITUZ		ITU zone
IOTA		Islands-On-The-Air designator	CQZ		CQ zone

* - these fields are either captured on the main dialog, computed from internal values, or determined by configuration

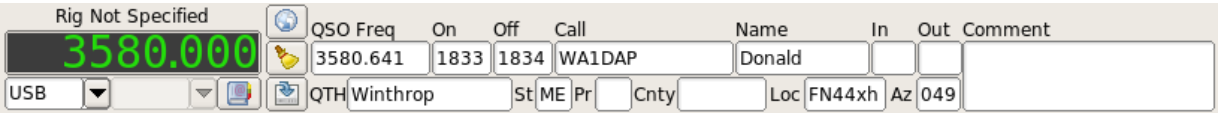
The data in the fldigi logbook can be exported to external text files; ADIF, text, and CSV (comma separated value). The ADIF can be read by any ADIF compatible logbook program. The text output is suitable for use in a wordprocessor and for printing. The CSV can be read into many spreadsheet programs such as Excel, Open Office or Gnumeric.

Digital modes signal reports

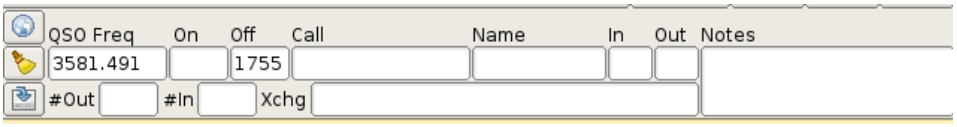
Fldigi does not enforce any rules on signal reporting. It could very well do so for many of the modes in which signal quality is inherently measured as a part of the decoder. Learning how to evaluate a signal, to properly report it, and then help in correcting deficiencies should be the goal of every amateur operator. Please read further on using both RST and RSQ signal reports.

Capturing QSO data

Fldigi supports two QSO capture panels. The first for casual QSO logging

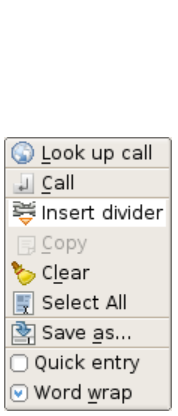


and the second for contest fields

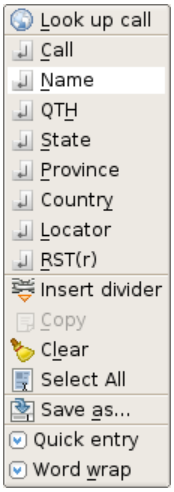


The frequency, Off (time off), and #Out are filled by the program. All the others can be populated by manual keyboard entry or by selection from the Rx panel. The time off, Off, is continuously update with the current GMT. The time on, On, will be filled in when the Call is updated, but can be modified later by the operator.

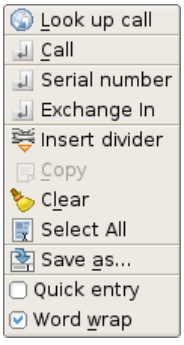
A right click on the Rx panel brings up a context sensitive menu that will reflect which of the two QSO capture views you have open.



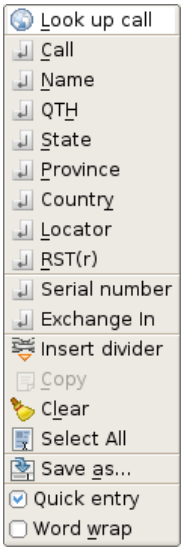
Normal
Short Menu



Normal
Long Menu



Contest
Short Menu



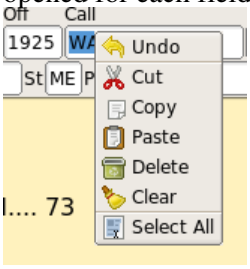
Contest
Long Menu

If you highlight text in the Rx pane then the menu selection will operate on that text. If you simply point to a word of text and right click then the menu selection will operate on the single word.

Certain fields may also be populated with automatic parsing, Call, Name, Qth and Loc. You point to the Rx pane word and then either double-left-click or hold a shift key down and left click. The program will attempt to parse

the word as a regular expression to populate the Call, Name, Qth, and Loc fields in that order. It may place some non standard calls into the Loc field if they qualify as a proper Maidenhead Grid Square, such as MM55CQ. That may be a special event station, but it also looks like a grid square locator value. You need to decide when that occurs and use the pop up menu for those special cases. The first non-Call non-Loc word will fill the Name field and subsequent qualify words will go into the Qth field.

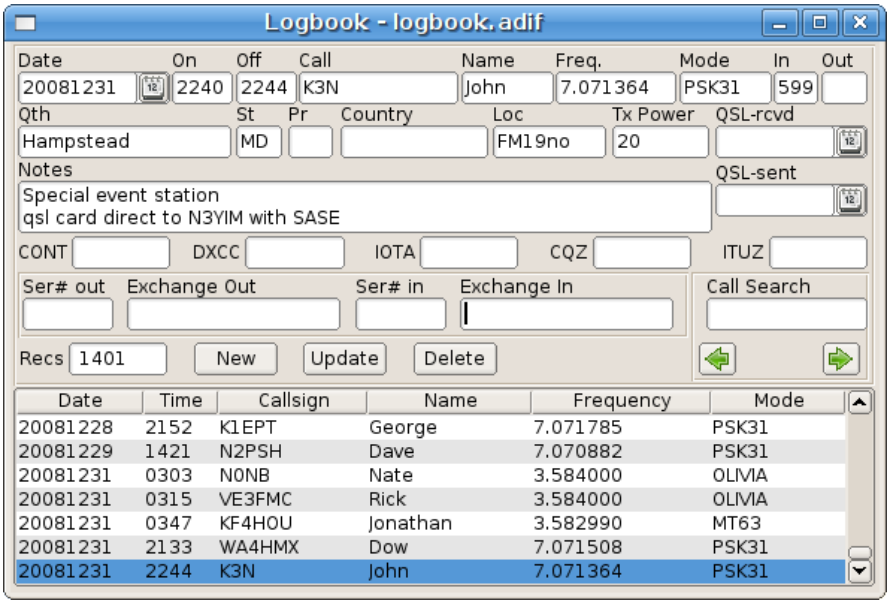
A highlighted section of text, can always be copied to the clipboard for subsequent pasting elsewhere. The Copy menu item will be active when text in the Rx pane has been highlighted. That text can also be saved to a file. Use the "Save as..." menu item for that purpose. All data fields in fldigi share a common set of keyboard shortcuts. Linux users will recognize these as familiar Emacs shortcuts. There is also a small popup menu that can be opened for each field by right clicking the contents with the mouse:



Highlighted text will be overwritten when a paste is selected. Otherwise the clipboard will be pasted at the current cursor position.
You can query on-line and local CD based data base systems for data regarding a Call. Set up your query using the Callsign Db configuration tab. You make the query by either clicking on the globe button, or selecing "Look up call" from the menu. The latter will also move the call to the Call field and make the query.

If you have previously worked a station the logbook will be searched for the most recent qso and fill the Name, Qth and other fields from the logbook. If the logbook dialog is open that last qso will be selected for viewing in the logbook.

You open the logbook by selecting from the View menu; View/Logbook. The logbook title bar will show you which logbook you currently have open. fldigi can maintain an unlimited (except for disk space) number of logbooks.



You can resize the dialog to suit your screen size and operating needs. Fldigi will remember the placement and size for subsequent use.

You can create new entries, update existing entries, and delete entries using this dialog. You can also search for an entry by callsign. The browser can be sorted by Date, Callsign, Frequency or Mode. The sort can be forward or backward with the most recent being the default selected entry after each sort. You execute the sort by clicking on the column button at the top of the column to be sorted. Each click causes the sort to reverse. I like to view my log with the most recent at the top. You might want to view it with the most recent on the bottom.

There are no frills such as keeping track of DXCC worked, fancy printouts etc. Fldigi's logbook is primarily a capture function. You can export your data for use with an external database or for uploading to LOTW or eQSL. Data from those sources can also be used for importing into the logbook.

Exporting logbook data: Log Exports.

Cabrillo reporting: Contest Reports.

Contents

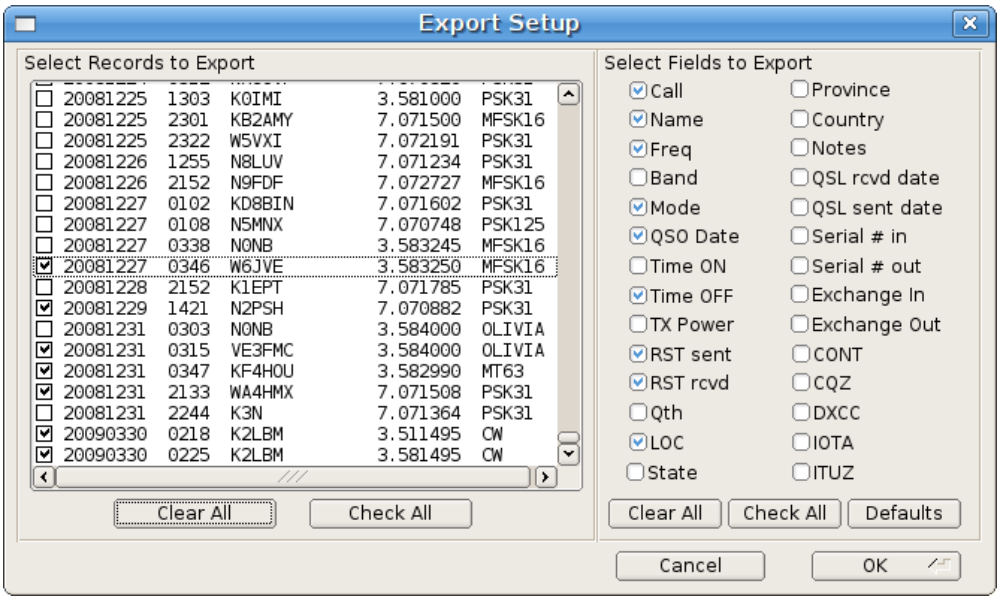
Exporting Logbook Data

Fldigi provides automatic export of log records as they are recorded. On Linux the data is forwarded to Xlog compatible programs using the SysV message queue system. On Windows the records are exported via a temporary file structure and are accepted by Logger32.

The user may also export all or selected records consisting of all or selected fields. Access to this export function of available from the menu "File/Log/Export ADIF", "File/Log/Export Text", and "File/Log/Export CSV".

Export ADIF

Selecting the Export ADIF menu item opens the following dialog:



If you want to export every record press the "Check All" in the left panel. You can also select and deselect individual records. Choose which fields you want to export with the right panel controls. Press the OK button to continue or Cancel to abort the operation. A file chooser dialog will open which allows you to specify the name and location of the exported file. Use the extension ".adi" on Windows and ".adif" on the other OS's.

Export Text / CSV

The same Export Setup dialog is used for Text and CSV exports.

The Text export produces a simple space delimited file with columns set at locations dictated by the field size for each field that is exported. It is suitable for use with a word processing program or for printing a hardcopy of your activities.

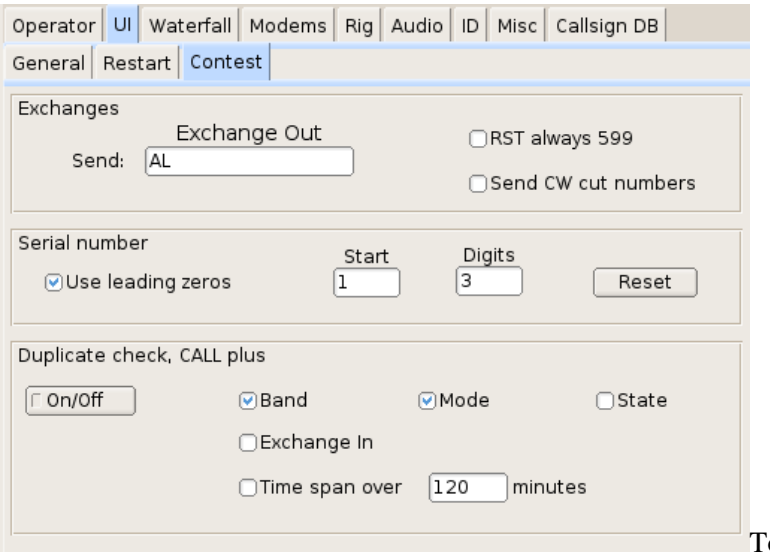
The CSV is a "Character Separated Value" file with the TAB character used as the field separator. This type of file can be imported into nearly all spreadsheet programs such as Gnumeric, Open Office or MS Excel.

Contents

Contest How To

Fldigi supports a basic contesting format. Select the menu item View/Contest fields to see how the qso entry entry fields change for contest data. You will see that fldigi has fields to support received and sent contest numbers as well as generic contest exchange information.

The serial number out (#Out) is automatically initialized and updated by the built-in serial number generator. You can enter the appropriate exchange information via the keyboard or mouse. Text in the Rx pane can be selected by the usual left-click-swipe of highlighting. Then right click anywhere after highlighting the desired text and a popup menu will appear allowing you to select the destination QSO field. Make your selection and the info is placed in the correct text box. Note that the popup menu changes with the QSO logging view and also with a change in "Quick entry". A full description is found in the description of operating the Logbook. The important thing to note for contest operation is that the Call and Serial # are single word captures. The Xchg capture can be either single word or multiple word (mark / right click). If the Xchg field has text contents then the new capture is appended to end of the current text in that field. That means you can point to the word representing the field, right click and select from the menu. You do not need to highlight the text for the word capture. You can very rapidly fill in the serial number and the exchange data (even if multi value) by simply pointing and right clicking on the desired word.



To set up fldigi for contesting you will need to open configure contest. the 1st row contains what info you want sent with the appropriate macro tag. ie...if the contest requires rst and name you would fill in the x1 box with your name. you will also need to check the rst always 599 box as this is the de-facto signal report in contests.

If you are participating in a cw contest you may want to select the "Send CW cut numbers", cut numbers is the norm for a CW contest. The cut numbers will send N for 9 and T for zero.

The next box contains the needed requirements to use serial numbers for a contest. you will always want to use leading zeros, start with 1 and use 3 digits. Press reset to initialize the #Out QSO field to the Start number.

Check the appropriate fields for determining if this is a duplicate call. DUPE will be displayed above the Call in the QSO logging area if a duplicate is detected. There are many choices to alert you to a duplicate contact. The

duplicate is based on the logical AND of all of the fields to be checked. The DUPE is cleared when you press the clear QSO log button (the brush icon).

After you have filled in all the required infomation, make sure you save and close.

Remember YOU MUST click the Reset button in the Serial number panel for the serial number counter to be initialized. You should also press the QSO clear button (broom) in the qso entry widget for the other changes to take effect.

It would be best to create a new log for each contest. You create a new log by selecting the menu item File/Logs/New logbook. The default new log name will be newlog.adif on Linux and newlog.adi on Windows. You can rename the new log file now or later by using the system file manager or when you save the log. The import/export feature of fldigi will allow you to export the log into your everday logging software or the built-in fldigi logbook.

Restarting a contest session

You might have closed down fldigi in the middle of a contest, everyone needs a break now and then. You then start fldigi and want to continue the contest. Here are the steps to insure that you continue operations with no glitches.

- Load your macro file that contains your contest macros (more on that below)
- Select the menu item View/Contest fields
- Select the menu item View/Logbook
- Make sure you have the contest logbook open ... if not then this is the time to open that logbook database.

  Select the menu item "File/Logs/Open logbook..." and find your log data file.
- Look at the last record and check the serial number sent. Enter that number plus one in the Start entry on the config contest tab (see above).
- Press the Reset button in that panel.

You are ready to keep on contesting

Remembering a contact

If you are copying a potential contact but you are not being heard you can save fldigi's modem state using one of two methods

1. double click the signal on the waterfall
2. right click on the Rx panel and select "Insert divider"

A line of text will be inserted at the end of the Rx text buffer. It will appear similar to this:

<<2008-12-30T10:06Z BPSK-31 @ 3580000+0781>>

The date-time, the mode, the transceiver operating frequency and the audio offset will be recorded. The text line is in blue and behaves in a way that you might expect a url reference to behave in a web browser window. Work a few more contacts (even on a different band or frequency) and then scroll the Rx pane to that special divider. Left click on the line of text and fldigi will restore the transceiver to its frequency, change the mode to the saved

mode and put the waterfall cursor at the audio offset frequency. Changing the transceiver frequency will only work if you are using CAT control of your transceiver. If you are not using CAT control the mode and waterfall cursor will still be restored.

There is no limit to the number of divider lines that can be inserted into the Rx pane. They will all be removed when the Rx pane is cleared.

Saving the entire session

Select the menu item "File/Logs/Log all RX/TX text". If this toggle menu is checked your entire session of received and sent text will be saved to a file in the fldigi default files folder. It will be given a name synonomous with the date and time is is started, ie: fldigi20081230.log. You can review this log by selecting the menu item "File/Show config" which will open your OS default file explorer to the fldigi files folder. The file is an ASCII text file.

The format of the daily log is shown in Working Logs.

Contesting Macro Tips

OK, now we have fldigi setup for basic contesting, lets move on to some ideas on macros to use. I tend to make generic one size fits all macros. I recommend that you make a new macro file, mine is named contest.mdf, this will give you 48 macros to use based on the type of contest you are entering. Take a good look at the examples I have listed, you will notice there are no commas, hyphens or other extraneous items. I have seen just about every example of a poorly thought out macro there is or has ever been dreamed up. Classic examples are:

- w3nr you are 599 in Alabama your serial number is 001-001-001 how copy ??
- hello ed thanks for the call you are 599-599-001-001-001 qth Alabama back to you

The list goes on and on. Just think, you have to try and capture the exchange, try it and you will see what I mean.

When you enter a contest you have to decide whether you are going to sit on one frequency and call CQ (Run) or are you going to tune the band looking for stations to work (S&P). So lets set up some macros that should cover both cases.

Several new macro tags have been created to facilitate contesting, these include the following tags.

<LOG>	add QSO data to the logbook & clear the QSO data fields
<CNTR>	insert current contest serial number into the text stream
<INCR>	increment contest serial number
<DECR>	decrement contest serial number
<XOUT>	contest exchange

<QSOTIME>	current log time in Zulu HHMM format
<LDT>	local date time
<ILDT>	LDT in iso-8601 format
<ZDT>	Zulu date time
<IZDT>	ZDT in iso-8601 format

See Macros for additional information on editing and using the fldigi macro system.

RUN Macros

We need just a few, starting with a CQ macro - Put this in the F1 key defintion

<TX>
cq test de <MYCALL> <MYCALL> cq k
<RX>

Notice that I left 2 spaces between my call and 3 spaces at the end before the k. This will make it easier for a station to grab my call and the k on the end elminates garbage characters before my macro finishes. The tx/rx are on seperate lines as I want to be sure my macro is on a line by itself and not mixed in with screen garbage.

Now the exchange macro - Put this in the F2 key definition

<TX>
<CALL> 599 <CNTR> <CNTR> <X1> <X1> <CALL> k
<RX>

Why do I have his call at the beginning as well as the end, to make sure I have copied his call correctly. You will also see that I have not as yet logged the contact,why, well are you sure he does not need to correct his call or ask for a repeat.

You are asked to repeat the exchange, you can just re-send the exchange macro, this verfies all of the information. Now he sends you his info and if you have copied it correctly you need a TU macro. - Put this in the F3 key definition

<TX>
qsl tu qrz test <MYCALL> k
<RX><LOG><INCR>

Here we have done all the necessary items to complete the exchange. Notice that I did not log the contact until after everything was correct. I have fldigi set to clear on save, so when the <LOG> part of the macro executes the QSO area is cleared.

Thats the end of my RUN macro setup, told you it was rather simplistic and generic.

S&P Macros

I rarely if ever use S&P, but there are times I need to, especially if my QSO rate drops while running. Again the macros are very generic with only the needed info. If band conditions warrant you may want to send your call 3 times. Put this in the F5 key definition

<TX>
<MYCALL> <MYCALL> k
<RX>

Why just my call ?? Well I assume the other guy already knows his call !

The exchange macro is basically the same as the RUN macro. Put this one in the F6 key definition

<TX>
599 <CNTR> <CNTR> <X1> <X1> k
<RX>

As you see I have not as yet logged the QSO or incremented the serial number. This is the final S&P macro. Put this one in the F7 key definition

<LOG><INCR>

Now this is the most important macro you will ever need.....trust me. Put it where you won't fail to find it. How about F9 ?

<TX>
agn agn k
<RX>

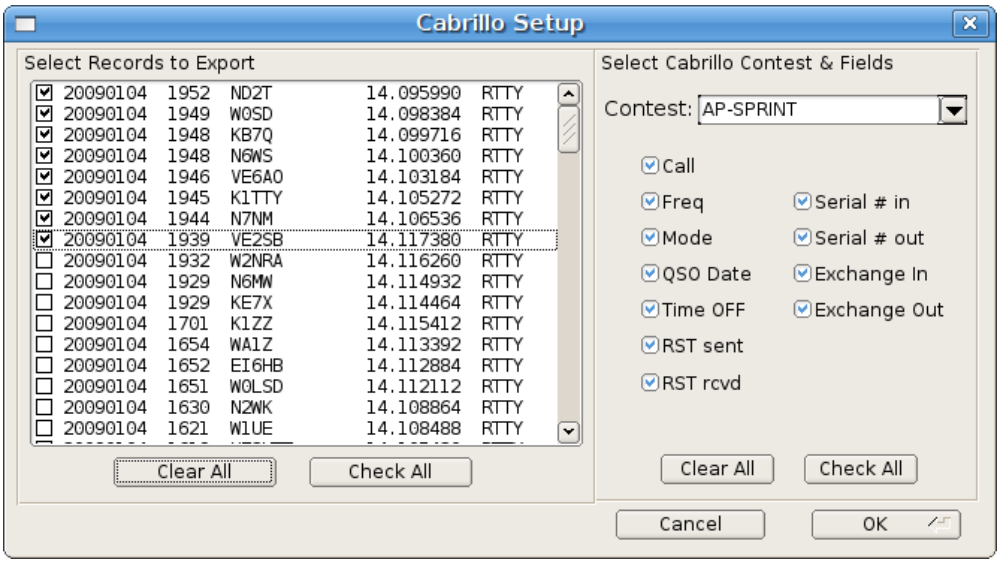
You will see that it is used many times during a contest, especially with weak stations and heavy QRN/QRM.

Contents

Creating a Cabrillo Report

Fldigi can generate a basic Cabrillo report that meets most contest needs.

Selecting the "File/Log/Cabrillo report" menu item opens the following dialog:



If you want to export every record press the "Check All" in the left panel.

Select the Contest type from the pull down menu in the right panel. Fldigi knows how to format the various fields for each contest. When satisfied with the setup press OK. You will then have the opportunity to specify the location and name of the cabrillo output file.

You must then open the file with a plain text editor and modify the appropriate entries. Check with each contest sponsor to see what their requirements are.

Here is an example of a generated cabrillo report format before being edited:

START-OF-LOG: 3.0
CREATED-BY: fldigi 3.11

The callsign used during the contest.
CALLSIGN: W1HKJ

ASSISTED or NON-ASSISTED
CATEGORY-ASSISTED:

Band: ALL, 160M, 80M, 40M, 20M, 15M, 10M, 6M, 2M, 222, 432, 902, 1.2G
CATEGORY-BAND:

Mode: SSB, CW, RTTY, MIXED
CATEGORY-MODE:

```
# Operator: SINGLE-OP, MULTI-OP, CHECKLOG
CATEGORY-OPERATOR:

# Power: HIGH, LOW, QRP
CATEGORY-POWER:

# Station: FIXED, MOBILE, PORTABLE, ROVER, EXPEDITION, HQ, SCHOOL
CATEGORY-STATION:

# Time: 6-HOURS, 12-HOURS, 24-HOURS
CATEGORY-TIME:

# Transmitter: ONE, TWO, LIMITED, UNLIMITED, SWL
CATEGORY-TRANSMITTER:

# Overlay: ROOKIE, TB-WIRES, NOVICE-TECH, OVER-50
CATEGORY-OVERLAY:

# Integer number
CLAIMED-SCORE:

# Name of the radio club with which the score should be aggregated.
CLUB:

# Contest: AP-SPRINT, ARRL-10, ARRL-160, ARRL-DX-CW, ARRL-DX-SSB, ARRL-SS-CW,
# ARRL-SS-SSB, ARRL-UHF-AUG, ARRL-VHF-JAN, ARRL-VHF-JUN, ARRL-VHF-SEP,
# ARRL-RTTY, BARTG-RTTY, CQ-160-CW, CQ-160-SSB, CQ-WPX-CW, CQ-WPX-RTTY,
# CQ-WPX-SSB, CQ-VHF, CQ-WW-CW, CQ-WW-RTTY, CQ-WW-SSB, DARC-WAEDC-CW,
# DARC-WAEDC-RTTY, DARC-WAEDC-SSB, FCG-FQP, IARU-HF, JIDX-CW, JIDX-SSB,
# NAQP-CW, NAQP-RTTY, NAQP-SSB, NA-SPRINT-CW, NA-SPRINT-SSB, NCCC-CQP,
# NEQP, OCEANIA-DX-CW, OCEANIA-DX-SSB, RDXC, RSGB-IOTA, SAC-CW, SAC-SSB,
# STEW-PERRY, TARA-RTTY
CONTEST: ARRL-RTTY

# Optional email address
EMAIL:

LOCATION:

# Operator name
NAME:

# Maximum 4 address lines.
ADDRESS:
ADDRESS:
ADDRESS:
ADDRESS:

# A space-delimited list of operator callsign(s).
OPERATORS:
```

```
# Offtime yyyy-mm-dd nnnn yyyy-mm-dd nnnn
# OFFTIME:

# Soapbox comments.
SOAPBOX:
SOAPBOX:
SOAPBOX:

QSO: 14095 RY 2009-01-04 1952 W1HKJ       599 GA 12345   ND2T        599 CA 67890
QSO: 14098 RY 2009-01-04 1949 W1HKJ       599 GA         W0SD        599 SD
QSO: 14099 RY 2009-01-04 1948 W1HKJ       599 1234567890 KB7Q        599 1234567890
QSO: 14100 RY 2009-01-04 1948 W1HKJ       599 GA         N6WS        599 CA
QSO: 14103 RY 2009-01-04 1946 W1HKJ       599 GA         VE6AO       599 AB
END-OF-LOG:
```

Contents

PSK Reporter

The PSK reporter can generate reception reports from three different sources:

- The decoded text
- The log data
- Data entered manually

The configuration for the PSK reporter in in Misc/Spotting. PSKR needs the following fields from the Oper tab to be non-empty:

1. Callsign (freeform because it's impossible to verify, and because we need to support SWLs without callsigns)
2. Locator (standard 6 character format)
3. Antenna info (freeform, should be kept reasonably short)

Sources (1) and (2) are configurable from Misc/Spotting configuration tab, while (3) is always enabled. To keep the code sane, changing the PSKR options (or the above station info) does not take immediate effect. Instead, the Initialize button changes colour to indicate that the changes have not
been applied. Clicking on the button will do so (or display an error) for the current and future sessions. This is similar to the Initialize buttons in the rig control configuration.

Here are the options in some more detail:

Automatically spot callsigns in decoded text

The text that is sent to the main window or the PSK browser is continuously searched for callsigns. If this option is enabled, the main window gets a "Spot" light button that can toggle the auto-spotter on and off. It is automatically turned off when playback is selected in the Files menu. The main window text is not searched if the viewer is active, i.e., if it is displayed and the current modem is PSK.

Send reception report when logging a QSO

A reception report is queued for every QSO as soon as it's logged

Report QRG (etc.)

This makes the reception reports include the current rig frequency, adjusted for modem audio frequency and rig sideband. It does not need a click on "Initialize" to take effect. This needs to be an option because it is impossible to tell whether the user has real or "fake" rig control with 100% certainty. Besides that, users may want to run a dedicated spotter for a narrow modes sub-band, and in that case they won't have to synchronise fldigi's frequency display with the rig all that often.

Host and port

With the port set to 14739 the reports will not be entered in the main database, but instead will be analysed and displayed here:

http://pskreporter.info/cgi-bin/psk-analysis.pl

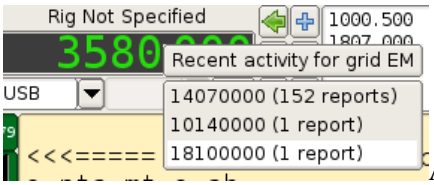
Probably of no interest to anyone who is not hacking on a PSKR client but may be useful for debugging. The PSKR protocol uses UDP with no acknowledgements or retransmissions, so don't be surprised if the occasional report never makes it to the server. There should be enough coverage overlap to make packet loss irrelevant (and save a lot of bandwidth and CPU cycles).

The spotter needs to see a repeated callsign within a short search window, but stations do not always repeat their callsigns. In addition, some operators like to be creative with their macros, and as a result some signals will decode 100% but the callsign will never be auto-captured. Such callsigns can be reported manually.

The manual spotting is done by right-clicking the QRZ "globe" icon. This will generate a report for whatever is in the Call & Loc fields, so make sure that those are correct! You should also verify the frequency (e.g. by placing the waterfall marker on the signal being spotted).

There is a confirmation popup that will open when you right click the "globe" button. The aim of course is to avoid accidentally sending rubbish reports to the PSK reporter database.

Reception reports are filtered for duplicates among all data sources: a report is queued only once every hour for each callsign and frequency band. The queue is flushed every five minutes. You can see what the spotter is doing in the Event Log window or on the terminal if you set the log level to "Info". "Debug" will show all the gory details.



A button and popup text field on the rig control frame give access to the most recent receptions reports in your geographic area.The area is determined by the contents of the field to the right of the button, or by the locator text on the operator tab if the mini field is empty. The first two characters of the locator are used. If the locator is not set, the pskreporter.info uses the current IP geolocation to approximate it.

A popup is displayed when the user clicks the button or presses the Enter key from within the field. The popup shows the frequencies by measure of activity that gives more weight to transmissions. If rig control is active, the user can click on one of the lines to go to that band. Clicking on the 18100000 (1 report) line would immediately QSY the transceiver to 18.1 MHz.

The data is retrieved from http://pskreporter.info/cgi-bin/psk-freq.pl

or with a filled text field, http://pskreporter.info/cgi-bin/psk-freq.pl?grid=TEXT

There is a link to the pskreporter.info map page in the Help menu.

Contents

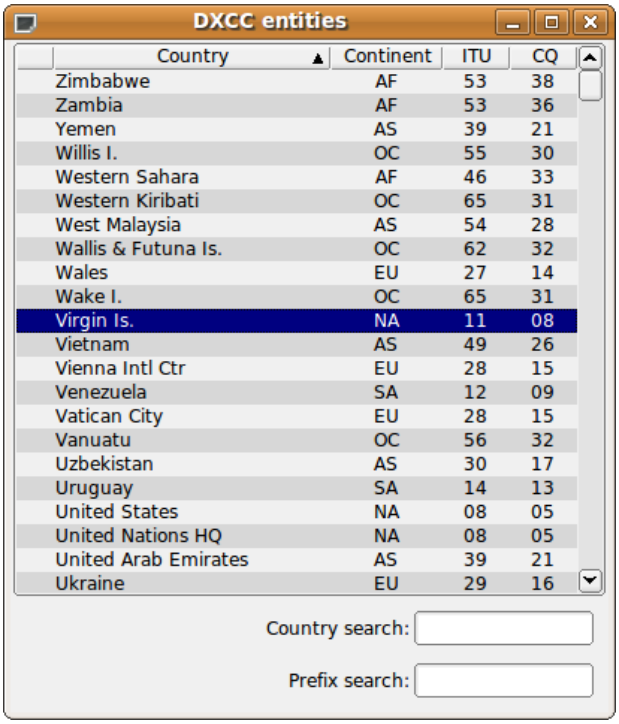
DXCC List

Fldigi uses several data files that are not included with the distribution. These must be downloaded from the list maintenance web sites for the most current data. These lists include:

List Data	List Name	Web source
DXCC	cty.dat	http://www.country-files.com/cty/
LOTW	lotw1.txt	http://www.hb9bza.net/lotw/lotw1.txt
EQSL	AGMemberList.txt	http://www.eqsl.cc/QSLcard/DownloadedFiles/AGMemberList.txt

These files should be downloaded and placed in the fldigi files directory. The most convenient way to open the fldigi files directory is via the menu item "File / Show config".

The DXCC list browser is shown by selecting the menu item "View / Countries".



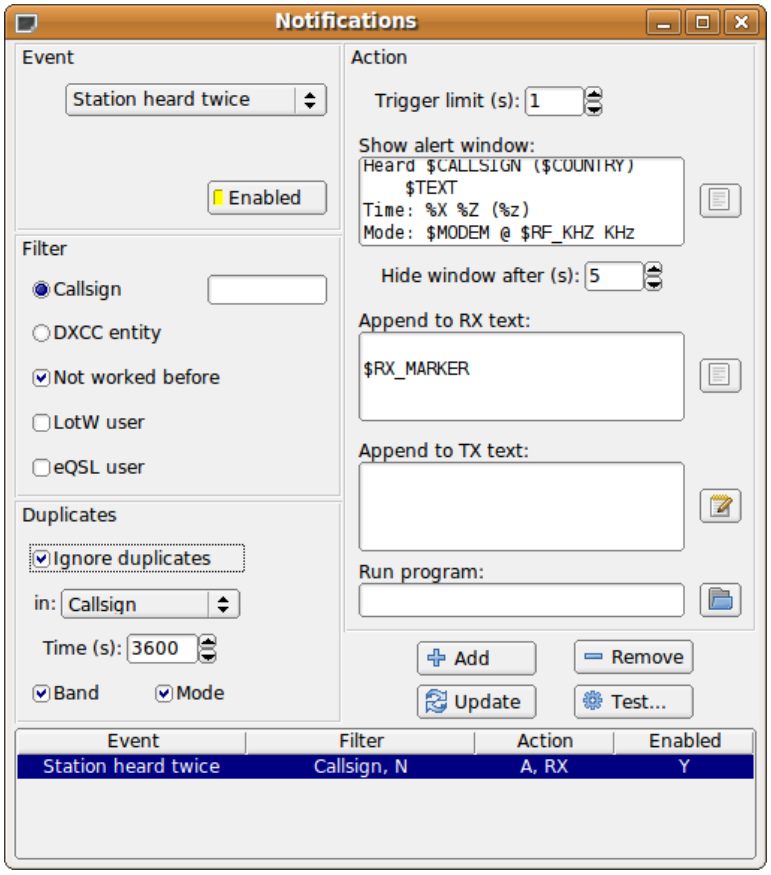
You can sort the list by Country, Continent, ITu or CQ zone by clicking on the various column headers.

Contents

Notifier

This is something that will interest those of you who would like to "grep" the bands.

There is a new dialog available under Configure->Notifications in which you can specify search patterns and alerts that are triggered when the decoded text matches those patterns. This only happens when the Spot button on the main window is activated, as with the PSK Reporter client.



First, here's how it works in general. You specify a regular expression (RE) that contains one or more parenthesised capturing groups. Fldigi's spotter matches it against the incoming text (main window or PSK Browser) and, if the RE matches, it performs one or more of the following:

Displays an alert window with some text and a "go to that frequency" button.

Enters some arbitrary text into the Transmit pane. The text may contain <MACRO>s and these will be expanded as usual.

Runs a program (Unix/Linux only for now).

The text described by the capturing group(s) can be used in all of the above. There is an example of this at the end of this page.

Not everyone is at ease in writing regular expressions for the notifier to act upon. So a few "canned" searches are coded into the notifier and are selected from the event chooser at the upper left of the dialog.

1) My Callsign de CALL. Can be used to alert you when CALL calls you.

2) Station heard twice. Pretty much the same search that the PSK reporter client does.

3) Custom text search. This reveals an input field where you type your own RE.

Both (1) and (2) are special cases of (3), but with some extra processing available because in each case fldigi knows what it has just found.

The Filter pane is available for the first two event types only, i.e. not the custom text search. In this pane you can specify some properties that the spotted callsign must have for the actions to take place:

a) The Callsign radio button reveals a text field when selected. If you enter something in that field, the event will be accepted only if the text matches the spotted callsign (I may change this to a RE match).

b) The "DXCC entity" radio button reveals a button that brings up a list of DXCC entities. Select entities by clicking or dragging. If you select any at all, the spotted callsign's country will have to be one of those or the event will be ignored. Having no entities selected is the same as selecting all of them, i.e. any country, but is a more efficient.

The entity list can be sorted by clicking on the row headers, and there is a right-click context menu that can (de)select by continent and CQ zone. The buttons and search fields at the bottom behave as you'd expect.

The list is also available with the menu item "View / Countries" in the main window

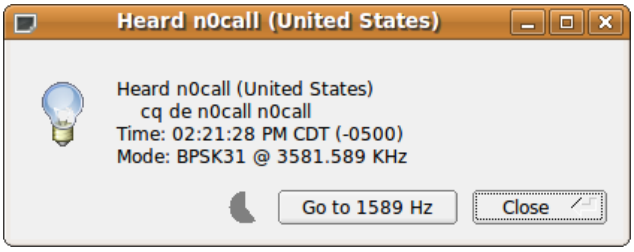
You need cty.dat for all this to work

c) The "Not worked before" check button asserts that, if you have selected (a) above, the callsign must not be in your logbook. Same with (b), but now you must not have had any QSOs with
stations from that country in the log.

d) The "LotW user" and "eQSL user" buttons specify that the callsign must be on one of these two lists (the documentation explains where to get the user lists from and where to put the files).

The Action pane is where you choose how fldigi will alert you when an event matches the filter bits.

a) The text in the "Show alert text" box, if not empty, is shown in a pop-up window. The alert window has a timer and dismisses itself after a configurable time interval (the "Hide after" control). The user can click anywhere inside the window to stop the timer.



The button next to the text box enters the default alert text for the event you have selected. There are a few variables that are substituted when the window is displayed:

For all three event types: $MODEM (modem name), $DF_HZ (dial frequency), $RF_HZ (actual receive frequency), $RF_KHZ, $AF_HZ (modem audio frequency)

For the 1st event type (my call): $CALLSIGN, $TEXT (all matched text).

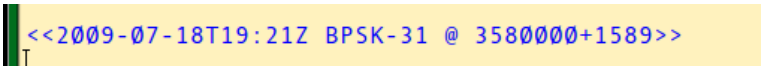
For the 2nd event type (station): $CALLSIGN, $TEXT, $COUNTRY.

For the 3rd event type (custom): you're on your own here, but fldigi will helpfully list all the possible substrings found in your RE.

The whole text is passed through strftime(3) so you can customize the date. Here's a reference for the % characters:

http://www.opengroup.org/onlinepubs/007908799/xsh/strftime.html

b) The "Append to TX text" box -- self explanatory. The same variable substitutions apply, as well as macro expansion. The nearby button shows the macro editor. The appended rx text is clickable. Clicking it will move the waterfall frequency (and transceiver if under CAT) to the detected signal and change to the indicated mode.



c) The "Run program" field and browse button are only available on Unix systems. Field contents are passed to the shell ("/bin/sh -c"), as with system(3). No variable or \backref substitution is done for this field, but all substrings are exported as environment variables, such as FLDIGI_NOTIFY_STR_1. The usual <EXEC> macro variables are also there and your ~/.fldigi/scripts directory will be in the shell's path. Try it out with a test script for the full list of variables.

d) The trigger limit box specifies how much time must pass between subsequent invocations of whatever actions you have specified.

The Duplicates pane has a check button that displays the rest of that group when checked. If you enable this, fldigi will remember what it has seen and ignore the event if it is a duplicate. The other controls in that pane determine what constitutes a duplicate:

a) The menu tells fldigi what to look at. For the first two event types, the menu will display "Callsign", and for the custom search it will contain a list of \X references for the RE.

b) The time box is also essential; it determines how close the events must be in time to be considered duplicates.

c) The Band and Mode check boxes further restrict the comparison.

An example:

You are looking at callsigns, with a dup time of 600s, and both Band and Mode checked. A callsign is found once and fldigi alerts you. Now if this callsign is spotted again, less than 600s later and in the same band and mode, it is a duplicate and will be ignored. With (say) Band and Mode unchecked, it is a duplicate regardless of frequency band or mode as long as it's heard before the 600s elapse.

Three of the four buttons at the bottom left are pretty much self-explanatory. Add to the list an event you have just specified, or select an event from the list and Remove it, or change some of its parameters and Update it.

The Test... button allows you to test an event with some text of your choice. This is particularly useful with the custom text search, as it's too easy to enter a RE that will never match. The dialog will show you the default test string for the two fixed event types. Careful: the "Station heard twice" event type expects a non-alphanumeric character at the end of its input. The default test string has a space at the end.

If nothing happens, it may be because you have not specified any actions, or because the event's filter does not match, or because the trigger limit or dup handling are preventing the actions from happening. In the latter case, updating an event will reset its dup data. But it's better to add the dup and trigger limits at the end, after you've tested the event.

The list at the bottom of the window shows the events you have added. All contents are saved in the file ~/.fldigi/notify.prefs.

The list has a context menu for quick access to Update, Remove, and Toggle. The first two have the same effect as clicking on the button of the same name.

The Toggle item lets you flip the "Enabled" status of an event: this is like selecting an event, clicking on the "Enabled" button in the Event pane to (de)activate it, and then clicking "Update". Disabled events are kept on the list but are not registered with the spotter and so they are never triggered.

If you disable all the events and there is nothing else using the spotter (e.g. PSK Reporter), the Spot button will disappear from the main window.

A 2nd example:

Here's how to do the "my call" event using the custom text search:

a) Select "Custom text search" in the event pane

b) In the RE box, enter (without the quotes or leading white space):

"<YOUR_CALL>.+de[[:space:]]+([[:alnum:]]?[[:alpha:]/]+[[:digit:]]+[[:alnum:]/]+)"

and remember to replace <YOUR_CALL> with your callsign.

c) In the actions pane you can now use \0 for the whole text matched by the above RE, and \1 for the first capturing group (the callsign).

d) Select "\1" in the duplicates menu if you want dup filtering.

e) Test with "<YOUR_CALL> de <SOME_OTHER_CALL>" and you should see the alert window with the text you specified.

Addional examples:

Add a "My callsign de CALL" event with a script that will do something to get your attention when someone calls you.

Add a "Station heard twice" with the DXCC filter and the "Not worked before" option. Also set the LotW or eQSL options if desired.

Add a "Station heard twice" with no callsign/dxcc/etc. filter but with duplicate filtering. Write a script that sends the data to a DX cluster or similar.

Here is a simple Perl script that uses notify-send (in the package libnotify-bin on Debian) to display desktop notification "bubbles". A better version would use the libnotify bindings for Perl or Python directly.
----------snip

```
#!/usr/bin/perl
exec("notify-send", "-t", "5000", "-i", "/usr/share/pixmaps/fldigi.xpm",
    "Heard " . $ENV{"FLDIGI_NOTIFY_CALLSIGN"} . " ($ENV{FLDIGI_NOTIFY_COUNTRY})",
    $ENV{"FLDIGI_NOTIFY_STR_0"});
snip----------------------------
```

Contents

Working Logs

Fldigi maintains a number of working log files that are found in its default folder. The default folder is easy to find, simply select the menu item "File/Show config" and your OS default files explorer will be opened to that location.

Rx/Tx Capture File

Everytime you start or stop fldigi that event is recorded in a daily log file. The daily log is name as:

flidigYYYYMMDD.log

where YYYYMMDD is the current GMT date. This log will also contain your entire session of Rx and Tx data annotated as to activity and timestamped. Here is a small example of the daily log:

--- Logging started at Tue Dec 30 11:37:21 2008 UTC ---

RX (2008-12-30 11:37Z): o ur property. No pwr even for a day is rough.
TX (2008-12-30 11:39Z):
TX (2008-12-30 11:39Z): CQ CQ CQ de W1HKJ W1HKJ W1HKJ
TX (2008-12-30 11:40Z): CQ CQ CQ de W1HKJ W1HKJ W1HKJ pse k

RX (2008-12-30 11:40Z): mG sk

--- Logging stopped at Tue Dec 30 11:48:11 2008 UTC ---

This log is appended to with each start and stop. That means that no data is ever overwritten.

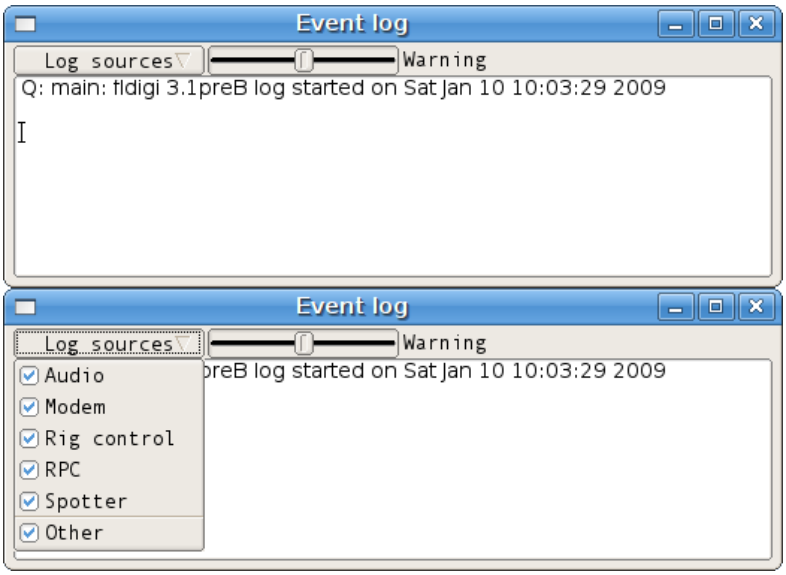
Status log

A log of errors, warnings and status reports is written for each session. This file is overwritten each time the program is opened and subsequently closed. Its format is also ASCII text and will contain data such as:

Q: main: fldigi 3.04BV log started on Tue Dec 30 05:47:10 2008

W: dxcc_open: Could not read contest country file "/home/dave/.fldigi/cty.dat"

This data is identical to that which can be viewed with the event log dialog which is opened using the menu item "Help/Event log"



There are five levels of event logging with increasing depth of reports:

The default level for logging events is "warning."

At the Debug level you will probably see more events than you need. You can select which events to suppress using the "Log sources" menu button. It defaults to all enabled.

Contents

XmlRpc Control

As of version 3.0, various aspects of Fldigi's operation can be controlled remotely using XML-RPC. XML-RPC data is transported via simple HTTP and client implementations exist for most programming languages. A Perl client that can be used as a control script is included in the source tarball as scripts/fldigi-shell.

The following command line arguments become available when XML-RPC support is compiled into fldigi, as described in the build instructions:

--xmlrpc-server-address HOSTNAME
Set the XML-RPC server address. The default is 127.0.0.1.

--xmlrpc-server-port PORT
Set the XML-RPC server port. The default is 7362.

--xmlrpc-allow REGEX
Allow only the methods whose names match REGEX

--xmlrpc-deny REGEX
Allow only the methods whose names don't match REGEX

--xmlrpc-list
List all available methods

The --xmlrpc-deny and --xmlrpc-allow switches can be used as a simple access control mechanism. REGEX specifies a POSIX extended regular expression. This invocation disables the methods that may cause fldigi to transmit:

--xmlrpc-deny 'main\.(tx|tune|run_macro)'

By default all methods are allowed.

The --xmlrpc-list switch outputs the method list and exits the program. If preceded by --xmlrpc-deny or --xmlrpc-allow, it shows the list of
methods as filtered by those switches.

The methods implemented in version 3.1 are listed below. The three columns are method name, signature (return_type:argument_types), and
description. Refer to the XML-RPC specification for the meaning of the signature characters (briefly: n=nil, b=boolean, i=integer, d=double,
s=string, 6=bytes, A=array, S=struct).

fldigi.list	A:n	Returns the list of methods
fldigi.name	s:n	Returns the program name
fldigi.version_struct	S:n	Returns the program version as a struct
fldigi.version	s:n	Returns the program version as a string
fldigi.name_version	s:n	Returns the program name and version
fldigi.config_dir	s:n	Returns the name of the configuration directory

modem.get_name	s:n	Returns the name of the current modem
modem.get_names	A:n	Returns all modem names
modem.get_id	i:n	Returns the ID of the current modem
modem.get_max_id	i:n	Returns the maximum modem ID number
modem.set_by_name	s:s	Sets the current modem. Returns old name
modem.set_by_id	i:i	Sets the current modem. Returns old ID
modem.set_carrier	i:i	Sets modem carrier. Returns old carrier
modem.inc_carrier	i:i	Increments the modem carrier frequency. Returns the new carrier
modem.get_carrier	i:n	Returns the modem carrier frequency
modem.get_afc_search_range	i:n	Returns the modem AFC search range
modem.set_afc_search_range	n:i	Sets the modem AFC search range. Returns the old value
modem.inc_afc_search_range	n:i	Increments the modem AFC search range. Returns the new value
modem.get_bandwidth	i:n	Returns the modem bandwidth
modem.set_bandwidth	n:i	Sets the modem bandwidth. Returns the old value
modem.inc_bandwidth	n:i	Increments the modem bandwidth. Returns the new value
modem.get_quality	d:n	Returns the modem signal quality in the range [0:100]
modem.search_up	n:n	Searches upward in frequency
modem.search_down	n:n	Searches downward in frequency

main.get_status1	s:n	Returns the contents of the first status field (typically s/n)
main.get_status2	s:n	Returns the contents of the second status field
main.get_sideband	s:n	Returns the current sideband
main.set_sideband	n:s	Sets the sideband to USB or LSB
main.get_frequency	d:n	Returns the RF carrier frequency
main.set_frequency	d:d	Sets the RF carrier frequency. Returns the old value
main.inc_frequency	d:d	Increments the RF carrier frequency. Returns the new value
main.get_afc	b:n	Returns the AFC state
main.set_afc	b:b	Sets the AFC state. Returns the old state
main.toggle_afc	b:n	Toggles the AFC state. Returns the new state
main.get_squelch	b:n	Returns the squelch state
main.set_squelch	b:b	Sets the squelch state. Returns the old state
main.toggle_squelch	b:n	Toggles the squelch state. Returns the new state
main.get_squelch_level	d:n	Returns the squelch level
main.set_squelch_level	d:d	Sets the squelch level. Returns the old level
main.inc_squelch_level	d:d	Increments the squelch level. Returns the new level
main.get_reverse	b:n	Returns the Reverse Sideband state
main.set_reverse	b:b	Sets the Reverse Sideband state. Returns the old state
main.toggle_reverse	b:n	Toggles the Reverse Sideband state. Returns the new state
main.get_lock	b:n	Returns the Transmit Lock state
main.set_lock	b:b	Sets the Transmit Lock state. Returns the old state
main.toggle_lock	b:n	Toggles the Reverse Sideband state. Returns the new state

main.get_trx_status	s:n	Returns transmit/tune/receive status
main.tx	n:n	Transmits
main.tune	n:n	Tunes
main.rsid	n:n	Waits for RSID
main.rx	n:n	Receives
main.abort	n:n	Aborts a transmit or tune
main.run_macro	n:i	Runs a macro
main.get_max_macro_id	i:n	Returns the maximum macro ID number
log.get_frequency	s:n	Returns the Frequency field contents
log.get_time_on	s:n	Returns the Time-On field contents
log.get_time_off	s:n	Returns the Time-Off field contents
log.get_call	s:n	Returns the Call field contents
log.get_name	s:n	Returns the Name field contents
log.get_rst_in	s:n	Returns the RST(r) field contents
log.get_rst_out	s:n	Returns the RST(s) field contents
log.get_serial_number	s:n	Returns the serial number field contents
log.get_state	s:n	Returns the State field contents
log.get_province	s:n	Returns the Province field contents
log.get_country	s:n	Returns the Country field contents
log.get_qth	s:n	Returns the QTH field contents
log.get_band	s:n	Returns the current band name
log.get_sideband	s:n	Returns the current sideband
log.get_notes	s:n	Returns the Notes field contents
log.get_locator	s:n	Returns the Locator field contents
log.get_az	s:n	Returns the AZ field contents
log.clear	n:n	Clears the contents of the log fields
log.set_call	n:s	Sets the Call field contents
text.get_rx_length	i:n	Returns the number of characters in the RX widget
text.get_rx	6:ii	Returns a range of characters (start, length) from the RX text widget
text.clear_rx	n:n	Clears the RX text widget
text.add_tx	n:s	Adds a string to the TX text widget
text.add_tx_bytes	n:6	Adds a byte string to the TX text widget
text.clear_tx	n:n	Clears the TX text widget
spot.get_auto	b:n	Returns the autospotter state
spot.set_auto	n:b	Sets the autospotter state. Returns the old state
spot.toggle_auto	n:b	Toggles the autospotter state. Returns the new state
spot.pskrep.get_count	i:n	Returns the number of callsigns spotted in the current session

Contents

Command Line Switches

The following output is available by executing "flidig --help"

Usage:
  fldigi [option...]

fldigi options:

 --config-dir DIRECTORY
  Look for configuration files in DIRECTORY
  The default is: /home/dave/.fldigi/

 --rx-ipc-key KEY
  Set the receive message queue key
  May be given in hex if prefixed with "0x"
  The default is: 9876 or 0x2694

 --tx-ipc-key KEY
  Set the transmit message queue key
  May be given in hex if prefixed with "0x"
  The default is: 6789 or 0x1a85

 --arq-server-address HOSTNAME
  Set the ARQ TCP server address
  The default is: 127.0.0.1

 --arq-server-port PORT
  Set the ARQ TCP server port
  The default is: 3122

 --xmlrpc-server-address HOSTNAME
  Set the XML-RPC server address
  The default is: 127.0.0.1

 --xmlrpc-server-port PORT
  Set the XML-RPC server port
  The default is: 7362

 --xmlrpc-allow REGEX
  Allow only the methods whose names match REGEX

 --xmlrpc-deny REGEX
  Allow only the methods whose names don't match REGEX

 --xmlrpc-list
  List all available methods

 --debug-level LEVEL

Set the event log verbosity

--version
Print version information

--help
Print this option help

Standard FLTK options:

-bg COLOR, -background COLOR
Set the background color
-bg2 COLOR, -background2 COLOR
Set the secondary (text) background color

-di DISPLAY, -display DISPLAY
Set the X display to use DISPLAY,
format is ``host:n.n''

-dn, -dnd or -nodn, -nodnd
Enable or disable drag and drop copy and paste in text fields

-fg COLOR, -foreground COLOR
Set the foreground color

-g GEOMETRY, -geometry GEOMETRY
Set the initial window size and position
GEOMETRY format is ``WxH+X+Y''
** fldigi may override this setting **

-i, -iconic
Start fldigi in iconified state

-k, -kbd or -nok, -nokbd
Enable or disable visible keyboard focus in non-text widgets

-na CLASSNAME, -name CLASSNAME
Set the window class to CLASSNAME

-ti WINDOWTITLE, -title WINDOWTITLE
Set the window title

Additional UI options:

--font FONT[:SIZE]
Set the widget font and (optionally) size
The default is: sans:12

Rig Xml Howto

This document describes the contents of the rig definition file “rig.xml”.

A number of transceivers have rig defintion files written and tested which you may use. These are found in the xmls directory on this site: xml archives. You will find subdirectories by manufacturer which contain files named by rig type, ie: TS-850.xml. If you create, test and verify the proper operation for a transceiver not yet posted please share that with others by sending it as an attachment to w1hkj@w1hkj.com and I will post it on the web site. You are encouraged to study the various rig definition files to learn more about how they are organized.

Comments are contained within the tag pair

<!-
-->

and may appear anywhere in the rig definition file The entire rig definition must be contained within the tag pair

<RIGDEF>
</RIGDEF>

The text within the tag pair <RIG></RIG> specifies the transceiver to which this file applies, as in:

<RIG>Icom 746 PRO</RIG>

The text within the tag pair <PROGRAMMER></PROGRAMMER> is not used by the parser, but should as a minimum say who created and who tested the definition file, as in:

<PROGRAMMER>
Dave Freese W1HKJ Tested by: W1HKJ, Dave
</PROGRAMMER>

The text within the tag pair

<STATUS>
</STATUS>

is not used by the parser, but should as a minimum state whether the definition file has been “Verifed”, is “Alpha”, what the Version and Date of creation or update, as in:

<STATUS> Verified Version: 1.0 Date: 2007 Jan 5 </STATUS>

The

<TITLE>
</TITLE>

tag pair contains the text which will be displayed on the window decoration bar, as in:

<TITLE>Rig Control - IC-746 PRO</TITLE>

The transceiver modes are specified within the <MODES></MODES> tag pair. Each entry or element associated with a mode has a symbol name (text) and a way to specifiy what the data transfer consists of. The data transfer might be a single byte, multiple bytes, or aa string

Example 1, for the Icom-746PRO

```
<MODES>
    <ELEMENT><SYMBOL>LSB</SYMBOL><BYTE>00</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>USB</SYMBOL><BYTE>01</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>AM</SYMBOL><BYTE>02</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>CW</SYMBOL><BYTE>03</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>RTTY</SYMBOL><BYTE>04</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>FM</SYMBOL><BYTE>05</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>CW-R</SYMBOL><BYTE>07</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>RTTY-R</SYMBOL><BYTE>08</BYTE></ELEMENT>
</MODES>
```

Example 2, for the Kenwood 850

```
<MODES>
     <ELEMENT><SYMBOL>LSB</SYMBOL><BYTE>31</BYTE></ELEMENT>
     <ELEMENT><SYMBOL>USB</SYMBOL><BYTE>32</BYTE></ELEMENT>
     <ELEMENT><SYMBOL>CW</SYMBOL><BYTE>33</BYTE></ELEMENT>
     <ELEMENT><SYMBOL>FM</SYMBOL><BYTE>34</BYTE></ELEMENT>
     <ELEMENT><SYMBOL>AM</SYMBOL><BYTE>35</BYTE></ELEMENT>
     <ELEMENT><SYMBOL>FSK</SYMBOL><BYTE>36</BYTE></ELEMENT>
     <ELEMENT><SYMBOL>CW-R</SYMBOL><BYTE>37</BYTE></ELEMENT>
     <ELEMENT><SYMBOL>FSK-R</SYMBOL><BYTE>39</BYTE></ELEMENT>
</MODES>
```

Example 3, for the FT-100

```
<MODES>
     <ELEMENT><SYMBOL>LSB</SYMBOL><BYTE>00</BYTE></ELEMENT>
     <ELEMENT><SYMBOL>USB</SYMBOL><BYTE>01</BYTE></ELEMENT>
     <ELEMENT><SYMBOL>CW</SYMBOL><BYTE>02</BYTE></ELEMENT>
     <ELEMENT><SYMBOL>CW-R</SYMBOL><BYTE>03</BYTE></ELEMENT>
     <ELEMENT><SYMBOL>AM</SYMBOL><BYTE>04</BYTE></ELEMENT>
     <ELEMENT><SYMBOL>DIG</SYMBOL><BYTE>05</BYTE></ELEMENT>
     <ELEMENT><SYMBOL>FM</SYMBOL><BYTE>06</BYTE></ELEMENT>
     <ELEMENT><SYMBOL>W-FM</SYMBOL><BYTE>07</BYTE></ELEMENT>
</MODES>
```

The modes which are supported by lower sideband in the transceiver are specified in the <LSBMODES></LSBMODES> tar pair. The string data for the lsb modes must match those given in the modes id specifier For example in the Icom 746 Pro:

```
<LSBMODES>
     <STRING>LSB</STRING>
     <STRING>RTTY</STRING>
     <STRING>CW-R</STRING>
</LSBMODES>
```

If the transceiver data stream uses identically the same format for the bandwidth data then it is specified in

the <BANDWIDTHS></BANDWIDTHS> tag pair

Example for the Icom 746 Pro:

```
<BANDWIDTHS>
    <ELEMENT><SYMBOL>50</SYMBOL><BYTE>00</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>100</SYMBOL><BYTE>01</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>150</SYMBOL><BYTE>02</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>200</SYMBOL><BYTE>03</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>250</SYMBOL><BYTE>04</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>300</SYMBOL><BYTE>05</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>350</SYMBOL><BYTE>06</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>400</SYMBOL><BYTE>07</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>450</SYMBOL><BYTE>08</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>500</SYMBOL><BYTE>09</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>600</SYMBOL><BYTE>10</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>700</SYMBOL><BYTE>11</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>800</SYMBOL><BYTE>12</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>900</SYMBOL><BYTE>13</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>1000</SYMBOL><BYTE>14</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>1100</SYMBOL><BYTE>15</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>1200</SYMBOL><BYTE>16</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>1300</SYMBOL><BYTE>17</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>1400</SYMBOL><BYTE>18</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>1500</SYMBOL><BYTE>19</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>1600</SYMBOL><BYTE>20</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>1700</SYMBOL><BYTE>21</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>1800</SYMBOL><BYTE>22</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>1900</SYMBOL><BYTE>23</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>2000</SYMBOL><BYTE>24</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>2100</SYMBOL><BYTE>25</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>2200</SYMBOL><BYTE>26</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>2300</SYMBOL><BYTE>27</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>2400</SYMBOL><BYTE>28</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>2500</SYMBOL><BYTE>29</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>2600</SYMBOL><BYTE>30</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>2700</SYMBOL><BYTE>31</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>2800</SYMBOL><BYTE>32</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>2900</SYMBOL><BYTE>33</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>3000</SYMBOL><BYTE>34</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>3100</SYMBOL><BYTE>35</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>3200</SYMBOL><BYTE>36</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>3300</SYMBOL><BYTE>37</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>3400</SYMBOL><BYTE>38</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>3500</SYMBOL><BYTE>39</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>3600</SYMBOL><BYTE>40</BYTE></ELEMENT>
</BANDWIDTHS>
```

If the bandwidth data stream is unique for send and receive data streams then they are specified separately with the <BW-CMD></BW-CMD> tag pair for data sent to the transceiver, and the <BW-REPLY></BW-REPLY> tag pair for data returned to the computer.

Example: FT-100:

```
<BW-CMD>
    <ELEMENT><SYMBOL>300</SYMBOL><BYTE>00</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>500</SYMBOL><BYTE>01</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>2400</SYMBOL><BYTE>02</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>6000</SYMBOL><BYTE>03</BYTE></ELEMENT>
</BW-CMD>

<BW-REPLY>
    <ELEMENT><SYMBOL>300</SYMBOL><BYTE>03</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>500</SYMBOL><BYTE>02</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>2400</SYMBOL><BYTE>01</BYTE></ELEMENT>
    <ELEMENT><SYMBOL>6000</SYMBOL><BYTE>00</BYTE></ELEMENT>
</BW-REPLY>
```

Fldigi can parse and decode message returned from the transceiver that define 4 aspects of the transceiver operation:

OK	data accepted by the transceiver
BAD	data rejected by the transceiver
MODE	current operating mode of the transceiver
BW	current bandwidth setting of the transceiver
FREQ	frequency of the active vfo (might be either A or B for example)

These are all contained within multiple <REPLY></REPLY> tag pairs. This is an example of a fixed format message with no variable fields. It is the OK message sent back by the Icom-746 PRO:

```
<REPLY>
    <SYMBOL>OK</SYMBOL>
    <SIZE>6</SIZE>
    <BYTES>FE FE E0 66</BYTES>
    <BYTE>FB</BYTE>
    <BYTE>FD</BYTE>
</REPLY>
```

The <SYMBOL></SYMBOL> pair and the command defnition are mandatory. The <SIZE></SIZE> field is mandatory and specifies the number of bytes contained in this reply. The above definition could also have been coded as:

```
<REPLY>
    <SYMBOL>OK</SYMBOL>
    <SIZE>6</SIZE>
    <BYTES>FE FE E0 66 FB FD</BYTES>
</REPLY>
```

When the reply contains variable data it is specified in a contained tag pair <DATA></DATA>. This data field contains specifiers that describe the kind and size of the data. The <DTYPE></DTYPE> tag pair may be one of:

BINARY or
DECIMAL

This is an example for the reply to a mode query that is returned by the Icom-746 PRO:

```
<REPLY>
    <SYMBOL>MODE</SYMBOL>     specifies the response name
    <SIZE>8</SIZE>                    8 bytes of data returned
    <BYTES>FE FE E0 66</BYTES>  4 bytes of preamble
    <BYTE>04</BYTE>                  1 additional byte for preample
    <DATA>
        <DTYPE>BINARY</DTYPE>   binary data field of 1 byte
        <SIZE>1</SIZE>
    </DATA>
    <FILL>1</FILL>                    a variable field (data) not used
    <BYTE>FD</BYTE>                  1 byte postamble
</REPLY>
```

Fldigi rigcat will check for both the preample and postamble to insure that a valid reply has been sent by the transceiver.

Google Map

snip ---------- copy the following to ~/.fldigi/scripts/map.pl

```
#!/usr/bin/perl

# Author: Stelios Bounanos, M0GLD
# Date:   20080625

use warnings;
use strict;
use Getopt::Std;

our $VERSION = "0.3141";
our %opts = ( "e" => 0, "m" => 1, "z" => 4);

cmdline();
open(STDOUT, '>', "/dev/null");

my $loc = exists($opts{'l'}) ? $opts{'l'} : $ENV{'FLDIGI_LOG_LOCATOR'};
die "Invalid locator\n" unless ((defined($loc) && length($loc) =~ /[2-6]/));

my $label = exists($opts{'t'}) ? $opts{'t'} : $ENV{'FLDIGI_LOG_CALL'};
$label = $loc if (!defined($label) || $label eq "");

my ($lon, $lat) = map { sprintf("%+.6f", $_) } mtoll($loc);
if ($opts{'m'}) {
   my $url = "http://maps.google.com/maps?q=${lat},${lon}(${label})&t=p&z=$opts{'z'}";
#    $url =~ s/([(),])/sprintf("%%%02X", ord($1))/ge; # encode some chars
   exec("xdg-open", $url);
   die "Could not exec xdg-open: $!\n";
}

exit(0) unless ($opts{'e'});
my $kml = (exists($ENV{'TMPDIR'}) ? $ENV{'TMPDIR'} : "/tmp") .
   "/" . $loc . ".kml";
open(KML, '>', $kml) or die "Could not write $kml: $!\n";
print KML <<EOF
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.2">
 <Placemark>
   <name>$label</name>
   <description>
     $label
     $loc
   </description>
   <Point>
     <coordinates>$lon,$lat,0</coordinates>
   </Point>
 </Placemark>
```

```
</kml>
EOF
   ;
close(KML);

#############################################################################

sub cmdline
{
   $Getopt::Std::STANDARD_HELP_VERSION = 1;
   my $old_warn_handler = $SIG{__WARN__};
   $SIG{__WARN__} = sub { die $_[0]; };
   getopts('t:l:mz:e', \%opts);
   $SIG{__WARN__} = $old_warn_handler;
}


# Convert a 2, 4, or 6-character Maidenhead locator string
# to decimal degrees. Return a (longitude, latitude) pair.
sub mtoll
{
   my $len = length($_[0]);
   $_[0] .= join("", ("A", "A", "0", "0", "A", "A")[$len .. 5]) if ($len < 6);
   $_[0] = uc($_[0]);
   die "Invalid locator\n" unless ($_[0] =~ /[A-R]{2}\d{2}[A-X]{2}/);

   my @digits = split(//, $_[0]);
   my ($lon, $lat) = (-180, -90);

   $lon += (ord($digits[0]) - ord('A')) * 20 +
         (ord($digits[2]) - ord('0')) * 2 +
         (ord($digits[4]) - ord('A') + 0.5) / 12;
   $lat += (ord($digits[1]) - ord('A')) * 10 +
         (ord($digits[3]) - ord('0')) +
         (ord($digits[5]) - ord('A') + 0.5) / 24;

   return ($lon, $lat);
}

sub HELP_MESSAGE
{
print <<EOF

Usage: $0 [-OPTIONS [-MORE_OPTIONS]] [--] [PROGRAM_ARG1 ...]

The following single-character options are accepted:

      -t LABEL  Use LABEL as the marker label
            The default is \$FLDIGI_LOG_CALL

      -l LOC    Place marker at IARU locator LOC
```

```
            The default is \$FLDIGI_LOG_LOCATOR

      -m       Show in Google Maps (default)
      -z       Zoom level (Google Maps only)

      -e       Write a Google Earth kml file in
               \$TMPDIR/LOC.kml
EOF
;
}

snip------------------------------------------------
```

parseUALR

A simple parser to create a formated console output for fldigi's <EXEC> macro:

```
snip----------------------------------------
#include <ctime>
#include <cstdio>
#include <cstdlib>
#include <unistd.h>
#include <string>
#include <iostream>
#include <fstream>

using namespace std;
using std::cout;
using std::cin;

int main(int argc, char *argv[])
{
   size_t pos = 0, pos2 = 0, pos3 = 0, pos4 = 0, pos5 = 0;

   string commandline = "";
   string name = "";
   string qth = "";
   string answer = "";

   char c = cin.get();
   while (!cin.eof()) {
      commandline += c;
      c = cin.get();
   }

   if (commandline.find("No match found") != string::npos)
      goto noresponse;

   pos = commandline.find(", ");
   if (pos == string::npos)
      goto noresponse;

   pos += 2;

   pos2 = commandline.find("\n", pos);
   if (pos2 == string::npos)
      goto noresponse;

   name = commandline.substr(pos, pos2 - pos);

   pos3 = name.find(32);
   if (pos3 != string::npos)
      name = name.substr(0, pos3);
```

```
    for (size_t i = 1; i < name.length(); i++) name[i] = tolower(name[i]);

    answer = "$NAME";
    answer.append(name);

    pos4 = commandline.find(", ", pos2);

    pos4 = commandline.rfind( "\n", pos4);
    pos4 += 1;
    pos5 = commandline.find("\n", pos4);

    qth = commandline.substr(pos4, pos5 - pos4);

    answer.append("$QTH");
    answer.append(qth);
    cout << answer.c_str();

    return 0;

noresponse:
    cout << "$NAME?$QTH?";
    return 0;
}

snip----------------------------------------------
```

save the above as "parseUALR.cxx" and then compile and link as follows:

g++ parseUALR.cxx -o parseUALR

copy the "parseUALR" executable to a directory on your shell exec PATH.

snip ---------- copy the following to ~/.fldigi/scripts/ualr-telnet.pl

```
#!/usr/bin/perl

# Author: Stelios Bounanos, M0GLD
# Date: 20090103
#
# ualr-telnet is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 3 of the License, or
# (at your option) any later version.
#
# ualr-telnetl is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <http://www.gnu.org/licenses/>.
# ----------------------------------------------------------------------------

use strict;
use warnings;

die "Usage: $0 CALLSIGN\n" unless (@ARGV == 1);

use Net::Telnet ();

sub error { print "\$NAME?\$QTH?\n"; exit(1); }
my $t = new Net::Telnet( Host => "callsign.ualr.edu", Port => 2000, Timeout => 10,
        errmode => \&error );
$t->open();
$t->waitfor('/LOOKUP>.*$/');
$t->print($ARGV[0]);

$_ = $t->getline();     # blank line
$_ = $t->getline();     # call
error() if (m/No match found/);

$_ = $t->getline();     # name
chomp; s/.+,\s+//; s/\s.+$//;
print "\$NAME$_";
$_ = $t->getline();     # addr
$_ = $t->getline();     # qth
chomp;
$_ =~ ",";
$_ = $`;

print "\$QTH$_\n";

$t->waitfor('/LOOKUP>.*$/');
```

```
$t->print("bye");

snip------------------------------------------------
```

RST & RSQ Reporting

RST

Is the traditional Readability, Strength, Tone reporting system used for CW operations for nearly as long as amateurs have enjoyed the airwaves.

READABILITY

1. Unreadable
2. Barely readable, occasional words distinguishable
3. Readable with considerable difficulty
4. Readable with practically no difficulty
5. Perfectly readable (that is 100% print in todays jargon)

SIGNAL STRENGTH

1. Faint signals, barely perceptible
2. Very weak signals
3. Weak signals
4. Fair signals
5. Fairly good signals
6. Good signals
7. Moderately strong signals
8. Strong signals
9. Extremely strong signals

TONE

1. Sixty cicle ac or less, very rough and broad
2. Very rough ac, very harsh and broad
3. Rough ac tone, rectified but not filtered
4. Rough note, some trace of filtering
5. Filtered rectified ac, but strongly ripple modulated
6. Filtered tone, definite trace of ripple modulation
7. Near pure tone, trace of ripple modulation
8. Near perfect tone, slight trac of modulation
9. Perfect tone, no trace of ripple, or modulation of any kind

RSQ

Give the report as RSQ for digiital modes, but especially BPSK and QPSK; see:
http://www.psb-info.net/RSQ-Reporting-Table.html

READABILITY

1. 0% undecipherable
2. 20% occassional words distinguishable
3. 40% consideralbe difficulty, many missed characters
4. 80% practically no difficyulty, occasional missed characters
5. 95%+ perfectly readable

STRENGTH

1. Barely perciptible trace
2. Weak trace
3. Moderate trace
4. Strong trace
5. Very strong trace

QUALITY

1. Splatter over much of the visible waterfall
3. Multiple visible pairs
5. One easily visible pair
7. One barely visible pair
9. Clean signal - no visible unwanted sidebar pairs

Build Info

The following is obtained by executing "fldigi --version"

fldigi 3.10
Copyright (c) 2008 Dave Freese, Stelios Bounanos, Leigh Klotz, and others
License GPLv2+: GNU GPL version 2 or later <http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

System: Linux dell 2.6.24-22-generic #1 SMP Mon Nov 24 18:32:42 UTC 2008 i686

Built on Tue Dec 30 15:29:17 CST 2008 by dave@dell with:
gcc version 4.2.4 (Ubuntu 4.2.4-1ubuntu3)
CFLAGS=-DLOCALEDIR="/usr/local/share/locale" -I. -I./include -I./irrxml -I./fileselector -DNDEBUG
-pthread -I/usr/local/include -I/usr/local/include -I/usr/include/freetype2 -D_THREAD_SAFE -D_REENTRANT
-I/usr/local/include -D_REENTRANT -I/usr/local/include -I/usr/include -pipe -Wall -fexceptions -O2
-ffast-math -finline-functions -g -O2
LDFLAGS=-L/usr/local/lib -lportaudio -lm -lpthread /usr/local/lib/libfltk_images.a -lpng -lz -ljpeg
/usr/local/lib/libfltk.a -lXft -lpthread -ldl -lm -lXext -lX11 -L/usr/local/lib -lsndfile -lsamplerate -lpulse-simple
-lpulse -L/usr/local/lib -lhamlib -lm -Wl,-Bstatic -L/usr/lib -lxmlrpc_server_abyss++ -lxmlrpc_server++
-lxmlrpc_server_abyss -lxmlrpc_server -lxmlrpc_abyss -lpthread -lxmlrpc++ -lxmlrpc -lxmlrpc_util
-lxmlrpc_xmlparse -lxmlrpc_xmltok -Wl,-Bdynamic -ldl -lrt
Libraries:
FLTK 1.1.9
libsamplerate-0.1.2
Hamlib version 1.2.7.1
PortAudio V19-devel 1899
libsndfile-1.0.17

Installing from Source

You should only be building fldigi locally if the binary distribution does not work on your Linux distribution. Unless you are modifying the code or need to compile for debugging you will not have any additional functionality over the binary files. The source code for fldigi is very large and has a number of dependencies that must be satisfied before a successful compile. If you are not familiar with compiling and linking source code you should probably practice on a simpler package before treading these waters.

To compile fldigi you will need:

- A recent C++ compiler. The GNU C++ compilers in the 4.x series are known to work. Building with g++ 3.x requires the development headers for the Boost C++ library.
- Version 1.1.x of the Fast Light Tool Kit (FLTK), with its development library and headers. Versions 1.1.7 to 1.1.9 are known to work. FLTK's multi-threading support is required.
- The samplerate (a.k.a. secret rabbit code) library.
- The Independent JPEG Group's JPEG library, or the PNG library.

You should also install the libraries and headers for PortAudio, the Portable audio I/O library.

Additional features are enabled if the corresponding libraries are present on your system:

- Support for rig control via hamlib is enabled if the hamlib development files are installed.
- Audio file generation, capture and playback support is enabled if `configure' can find the sndfile library.
- The PulseAudio sound backend is compiled if the development files for libpulse-simple, the PulseAudio simple API library, are present.
- The embedded XML-RPC web server is enabled if `configure' can find the C++ bindings for the libxmlrpc-c3 library.

Once you have installed the required packages, the following commands should be sufficient to compile fldigi and install it under /usr/local:

```
./configure
make
make install    (you may need superuser privileges for installation)
```

The `install' target installs the executable, an icon, and a .desktop file. After installation, an fldigi launcher should appear somewhere in
your applications menu.

Use ./configure -help to see all of the possible compile configurable items that are available. It is possible to compile fldigi for doing benchmark testing and for batch decoding. This is obtained by configuring with the --enable-benchmark switch. The binary is then suitable for measuring the modem's decoding speed.

Mac OS X installation notes

Mac OS X support was added in version 2.10. To run fldigi on OS X, you will need to create an app bundle. For this purpose, the Makefile has an `appbundle' target that can be used instead of `install':

```
make appbundle
```

This target will generate two bundles inside the build directory (by default src/):

1. mac-bundle/fldigi.app, which only contains the bare minimum that is required to run fldigi on the build system
2. mac-libs-bundle/fldigi.app, which also includes copies of non-system libraries (such as PortAudio and FLTK) that the binary links to

The configure script has some support for building universal x86/ppc binaries. Pass the following additional arguments to enable it:

```
--enable-mac-universal --disable-dependency-tracking
```

Fldigi has not been well tested on OS X. Any feedback, suggestions, or patches will be very welcome.

Windows installation notes

As of version 3.0, fldigi supports win32 systems via the Cygwin POSIX environment. To install cygwin, use the following installer link:

http://www.cygwin.com/setup.exe

Refer to the list of library requirements and install the corresponding cygwin packages. You will need at least the following:

1. Base: (all base packages)
2. Devel: binutils boost-devel fltk-devel gcc-g++ make pkg-config
3. Graphics: libjpeg-devel libpng12-devel
4. Libs: zlib

Use the default "Curr" option for the package selection.

PortAudio and libsamplerate are not available from Cygwin and must be compiled from source. If you wish to build easily redistributable binaries, configure these libraries for static linking, e.g. with

`--enable-static --disable-shared'.

After you have installed them in the desired location, remember to tell `pkg-config' how to find them, e.g. with

```
export PKG_SEARCH_PATH=/usr/local/lib/pkgconfig
```

To configure fldigi itself for static linking, use the following command:

```
./configure --enable-static LDFLAGS=-static
```

This will produce a binary that only depends on the cygwin library, which can be found in /bin/cygwin1.dll.

At the time of writing, fldigi has been built with the latest release of Cygwin, version 1.5.25. The binary runs on Windows 2000 and XP, however, a newer version of the cygwin1.dll library may be required for Vista. Version 20080530 of the 1.7 pre-release snapshot has been tested and is known to work:

```
http://cygwin.com/snapshots/
```

Free Software Foundation Generic Installation Instructions.

Contents