Contents

Digital Communications

In this operating supplement to the *ARRL Handbook*, updated in early 2021, Steve Ford, WB8IMY, discusses the techniques involved in assembling and configuring station components for operating on the various HF and VHF digital modes. Today's digital communication choices range from keyboard-based modes such as classic RTTY, packet radio, and PSK31, to FT8, FT4, JT65, digital voice, local high-speed multimedia networks and VHF/UHF networks linked by the internet.

The **Modulation** chapter explains the details of how digital signals are created and their characteristics. The **Digital Protocols and Modes** chapter supplies much more information about station-to-station communications and how networks of stations are created and managed. Unless otherwise noted, references to other chapters refer to chapters in the print version of the *ARRL Handbook*.

Amateur digital communication is today a mainstream activity, having far outgrown its history as a niche activity. From the end of World War II until the early 1980s, *radioteletype*, better known as *RTTY*, was *the* amateur radio digital mode. If you had visited an amateur RTTY station prior to about 1977, you probably would have seen a mechanical teletype machine, complete with rolls of yellow paper. The teletype may have been connected to the transceiver through an interface known as a *TU*, or *terminal unit*. An oscilloscope would probably have graced the layout as well, used for proper tuning of the received signal.

When affordable microprocessor technology appeared in the late 1970s, terminal units evolved as well. Some included self-contained keyboards and video displays, making the mechanical teletype obsolete. As personal computers evolved, they became perfect companions for TUs. In this configuration, the PC functioned as a "dumb terminal," displaying the received data *from* the TU and sending data *to* the TU for transmission. TUs of this era offered ultra-sharp receive filters that allowed hams to copy weak signals amid interference.

In the late 1980s, conventional terminal units began to yield to sophisticated microprocessor devices known as *multimode controllers*. As the name suggests, these compact units handle several different digital modes in one package, typically RTTY, packet, AMTOR and PACTOR. Like TUs, multimode controllers are stand-alone devices that communicate with a personal computer acting as a dumb terminal. All the heavy lifting is being done by the controller and its self-contained software known as *firmware*.

In the early 1990s, sound cards appeared for personal computers. As sound cards became more powerful, hams began to realize their potential. With the right software, a sound card could take received audio directly from the radio and translate it into digital information. The same sound card could also create various forms of digital audio modulation for transmission. The first "sound card mode" was PSK31, developed by Peter Martinez, G3PLX. In the years that followed, sound cards became more powerful and versatile. Hams responded by developing new digital modes to take advantage of the advances.

Hardware controllers are still with us, but they are primarily used for modes like packet and PACTOR that require more processing muscle and precision timing than a typical personal computer can provide on its own. Other amateur digital modes such as D-STAR depend on specially designed transceivers that combine the radio hardware with dedicated digital processing firmware.

1 Sound Device Modes

Sound device technology dominates the amateur HF digital communications world. Although the term *sound card* is commonly used, the techniques discussed in this section apply to motherboard-embedded sound chip sets, which are extremely common in computer systems today, and to external sound processing devices, including those found in a number of modern transceivers. We'll use the term "sound device" to refer to any of these hardware implementations.

The sound device modes in use today include FT8, FT4, PSK, RTTY, MFSK16, Olivia, JT65, Hellschreiber, MSK144, WSPR, and many more. There are also sound-device applications for digital voice, discussed later in this supplement, and for slow-scan TV, discussed in the **Image Communications** supplement.

As this was written in 2021, FT8 had become the most popular HF digital communication mode, both for casual contacts and for DXing. The popularity of FT8 notwithstanding, radioteletype (RTTY) remains the chief HF mode for contesting, although FT8 is making serious inroads in this application as well. In fact, a relatively new "cousin" of FT8, known as FT4, is specifically designed for contesting.

All other HF digital modes play minor roles, but each has characteristics that provide benefits depending on the use case. Olivia and MFSK16, for example, provide more robust copy under poor conditions. WSPR is the most popular mode for propagation testing. On VHF and above, the *WSJT-X* software suite offers modes specifically designed for meteor scatter and moonbounce work. *WSJT-X* is free and available for Windows, Linux, and macOS at **physics.princeton.edu/pulsar/k1jt/wsjtx.html**.

Regardless of the mode in question, the sound device functions as the critical link. It is put to work as a digital-to-analog (D/A) and analog-to-digital (A/D) converter. In its A/D role, the sound device takes receiver audio and converts it to digital information. During transmission, the sound device is used as a D/A converter, taking digital information from the software application and creating a corresponding analog signal that is fed to the transceiver. (For more information on A/D and D/A converters, see the **DSP and SDR Fundamentals** chapter of this *Handbook*.)

1.1 The Evolution of Sound Devices

In the early days of amateur radio sound device modes, the computer sound card (or motherboard sound chipset) was *the* critical component in every amateur station. But for most amateur applications today, the sound device has moved out of the computer and into external interfaces, or even into transceivers themselves. See **Figure 1**. Rather than traditional interfaces that merely passed analog audio signals between radios and computers, modern interfaces contain their own A/D and D/A devices. The connection to the computer is strictly for the transfer of digital information, typically through a USB cable.

Transceivers are increasingly taking the next step by bringing the A/D and D/A conversion process into the radio. There is no external interface whatsoever; the radio connects directly to the computer. We can expect to see an increasingly number of transceivers taking this approach.

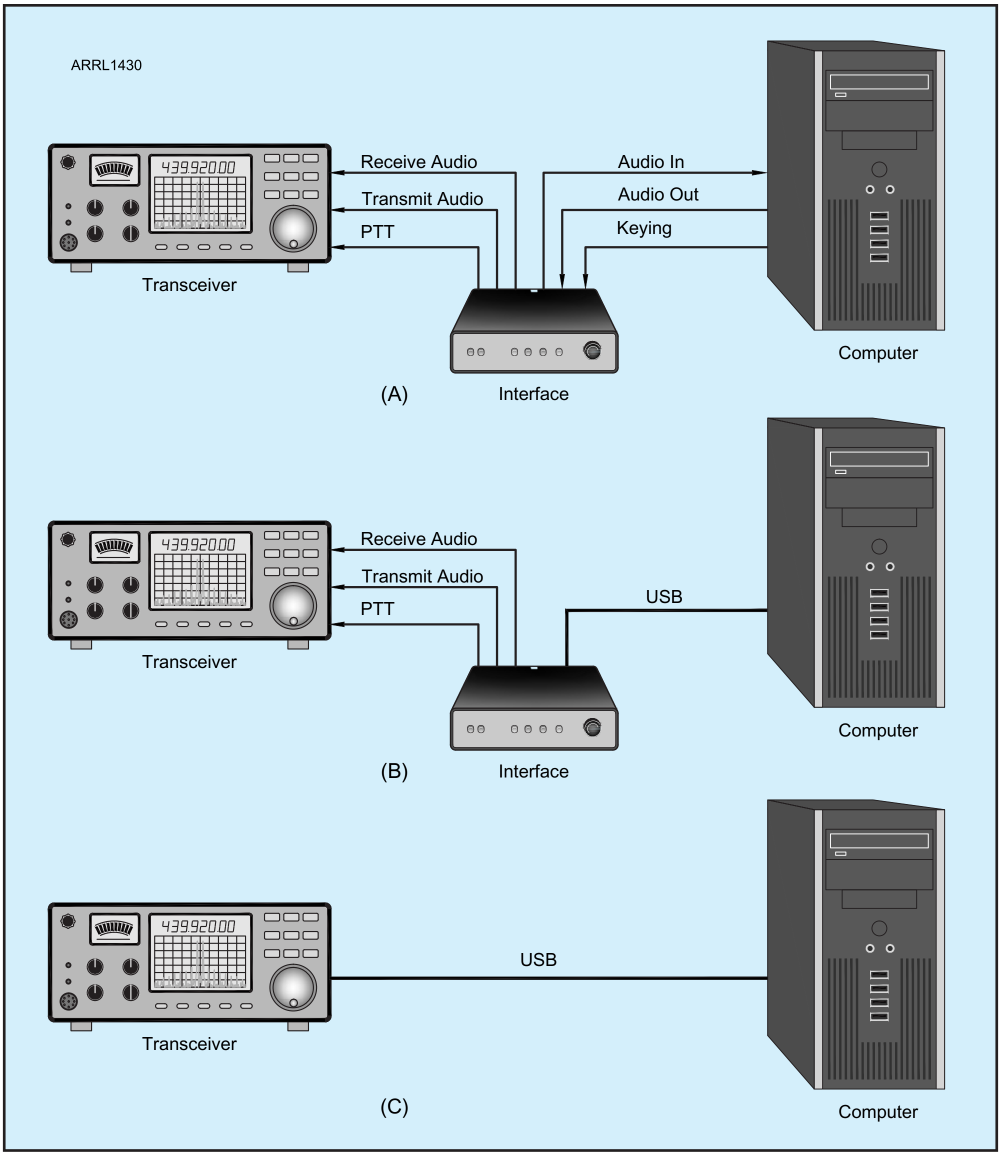


Figure 1 — Three of the most common configurations for interfacing your transceiver with your computer. (A) The most common setup for many years was based on an interface that took transmit and receive audio from the computer sound card, along with transmit/receive keying from one of the computer's COM ports, and passed everything to the transceiver in a way that kept the signal lines isolated. This type of interface is increasingly uncommon. (B) Many interfaces available at the time this was written are USB devices. A single USB cable plugs into the computer. Not only is the computer sound card not used, the interface contains its own sound device. In turn, the interface supplies transmit audio to the radio, processes receive audio from the radio and handles transmit/receive keying. (C) The future will likely see the disappearance of the interface entirely as transceivers incorporate their own computer interfacing. Today, several moderate to higher-priced radios have added this feature.

AFSK AND FSK

Most sound device modes rely on some form of frequency and/or phase-shift keying to create digitally modulated RF signals. This modulation takes place at audio frequencies with the sound device audio output applied directly to an SSB voice transceiver, either at the microphone jack or at a rear-panel accessory jack, and is called *audio frequency shift keying* (*AFSK*).

RTTY, PACTOR I and AMTOR signals can be sent using AFSK, and often are. It is also possible to transmit these modes by applying discrete binary data *directly* to the transceiver. This technique is known simply as direct *frequency-shift keying (FSK)*.

For example, each character in the Baudot RTTY code is composed of five bits. When modulated with AFSK, a "1" bit is usually represented by a 2125-Hz tone and is known as a *mark*. A "0" bit is represented by a 2295-Hz tone called a *space*. The difference between the mark and space is 170 Hz, called the *shift*. When applied to a single-sideband transceiver, this AFSK audio signal effectively generates an RF output signal that shifts back and forth between the mark and space frequencies.

A transceiver that supports FSK, however, can accept mark/space digital data directly from the computer and will use that information to automatically generate the frequency-shifting RF output. No audio signal is applied to the transceiver when operating FSK.

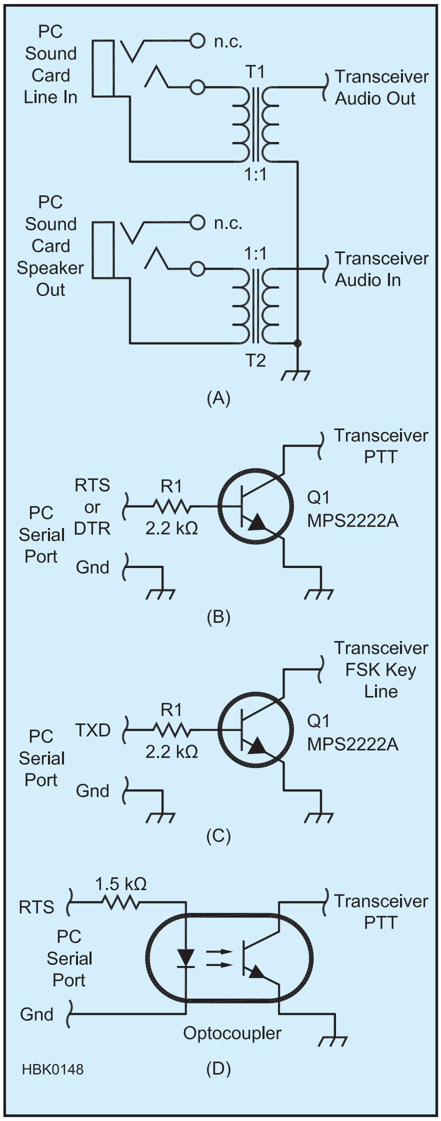


Figure 2 — Some commonly used interface circuits. At A, isolating the sound device and transceiver audio lines. T1 and T2 are 1:1 audio isolation transformers. At B, a simple circuit to use the computer COM port to key your transceiver PTT, and at C, a similar circuit for FSK keying. Q1 is a general purpose NPN transistor (MPS2222A, 2N3904 or equivalent). At D, an optocoupler can be used to provide more isolation between radio and computer. On a DB9 serial port connector: RTS, pin 7; DTR, pin 4; TxD, pin 3; GND, pin 5.

Is there an advantage to using AFSK or FSK? In years past, transceivers that did not support FSK operation often did not allow the use of narrow IF filtering. Narrow filters were reserved for CW, not the SSB voice mode used with AFSK. If you wanted to use RTTY with such a radio, you had to use AFSK and contend with the wider (2.4 kHz or so) SSB IF bandwidth, or else add an external audio filter. FSK-capable transceivers, on the other hand, allowed the RTTY operator to select narrow filters, reducing receive interference in crowded bands.

Many of today's transceivers offer adjustable-bandwidth digital signal processing (DSP) filters in the IF stages that can be used with any operating mode. This has effectively eliminated the FSK advantage, at least for receiving.

The appeal of FSK remains, however, when it comes to transmitting. A *properly modulated* AFSK signal is indistinguishable from FSK, but it is relatively easy to overdrive an SSB transmitter when applying an audio signal from a sound device (more on this in the next section). With FSK this is never an issue. You simply feed data from the computer to the radio; the radio does the rest.

This is why many RTTY operators still use FSK, and it's why transceiver manufacturers still offer FSK modes (sometimes labeled DATA or RTTY) in their products. Several sound device interfaces support FSK by providing a dedicated TTL circuit between the computer COM port, where the FSK data appears, and the transceiver FSK input. When used in this fashion, the sound device does not generate a transmit audio signal at its output. Instead, the RTTY software keys the various lines at the COM port to send the FSK data.

If the interface you've chosen doesn't support FSK, you can build your own TTL interface using the circuit shown in **Figure 2**. This simple circuit uses a transistor that is keyed on and off by data pulses appearing on the COM port TxD pin (pin 3 on a 9-pin COM port). New computers may not have a serial COM port. Several interfaces are available that provide FSK keying from a computer USB port.

It is important to note that FSK transceiver inputs can only be used for modes that are based on binary FSK, typically with 170 or 200-Hz shifts. These modes include RTTY, PACTOR I and AMTOR. You cannot use the FSK input to transmit multi-frequency or phase-shift modulated signals such as MFSK16 or PSK.

1.2 Transmit Audio Levels

When applying digitally modulated audio to a transceiver, it is critical to maintain proper levels. By overdriving the transceiver's audio input, you may create a wide, distorted RF signal that will be difficult, if not impossible, to decode at the receiving end. Such an overmodulated signal will also cause considerable interference on adjacent frequencies.

As you increase the transmit audio output from your sound device, pay careful attention to the ALC indicator on your transceiver. The ALC is the automatic level control that governs the audio drive level. When you see your ALC display indicating that audio limiting is taking place, or if the display indicates that you are exceeding the ALC "range," you are feeding too much audio to the transceiver.

Monitoring the ALC by itself is not always effective. Many radios can be driven to full output without budging the ALC meter out of its "nominal" range. Some radios become decidedly nonlinear when asked to provide SSB output beyond a certain level (sometimes this nonlinearity can begin at the 50% output level). We can ignore the linearity issue to a certain extent with an SSB voice signal, but not with digital modes because the immediate result, once again, is splatter.

The simplest method to tell if your signal is clean is to ask someone to give you an evaluation on the air. For example, PSK31 programs commonly use a waterfall audio that can easily detect "dirty" signals. The splatter appears as rows of lines extending to the right and left of your primary signal, as shown in **Figure 3**. (Overdriven PSK31 signals may also have a harsh, clicking sound.)

If you are told that you are splattering, ask the other station to observe your signal as you slowly decrease the audio level from the sound device or processor. When you reach the point where the splatter disappears, you're all set.

Don't worry if you discover that you can only generate a clean signal at, say, 50 W output. With PSK31 and most other sound device modes, the per-

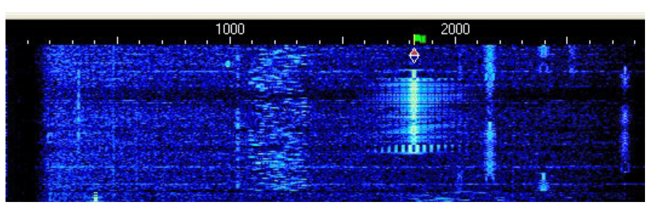


Figure 3 — The waterfall spectrum display built into popular sound device software can help detect overdriven PSK31 signals. Note the lines extending from the signal on the right, near the 2000 Hz marker, indicating splatter. The other signals are from properly adjusted transmitters.

formance differential between 50 W and 100 W is inconsequential.

1.3 Sound Device Software

As you can see in **Table 1**, there is great deal of software available for sound-device-based digital modes. Some of the software applications are free, while others require registration and a fee.

There are applications dedicated to modes, such as *DigiPan* (PSK31) and *MMTTY* (RTTY). The trend in recent years has been to multimode programs that support many different digital modes in a single application.

Most software applications rely on *panoramic reception* where the software processes and displays all signals detected within the bandwidth of the received audio signal. The signal “signatures” are often shown within continuously scrolling *waterfall* displays such as the one shown in **Figure 4**.

Panoramic reception is particularly popular among PSK31 and FT8/FT4 operators because the narrow signals tend to cluster within a relatively small range of frequencies (about 2.5 to 3 kHz, the bandwidth of a typical SSB filter). By using panoramic reception, an operator can simply click the mouse cursor on one signal trace in the waterfall after another, decoding each one in turn.

The weakness of panoramic reception appears when wide IF filtering is used to display as many signals as possible. The automatic gain control (AGC) circuit in the receiver is acting on everything within that bandwidth, working hard to raise or lower the overall gain according to the overall signal strength. That’s fine if all the signals are approximately the same strength, but if a very strong signal appears within the bandwidth, the AGC will *reduce* the gain to compensate. The result will be that many of the signals in the waterfall display will suddenly vanish, or become very weak, as the AGC drops the receiver gain. In cases where an extremely strong signal appears, *all* signals except the rock crusher may disappear completely.

The alternative is to use narrower IF or audio filter settings. These will greatly reduce the waterfall display width, but they will also remove or reduce strong nearby signals.

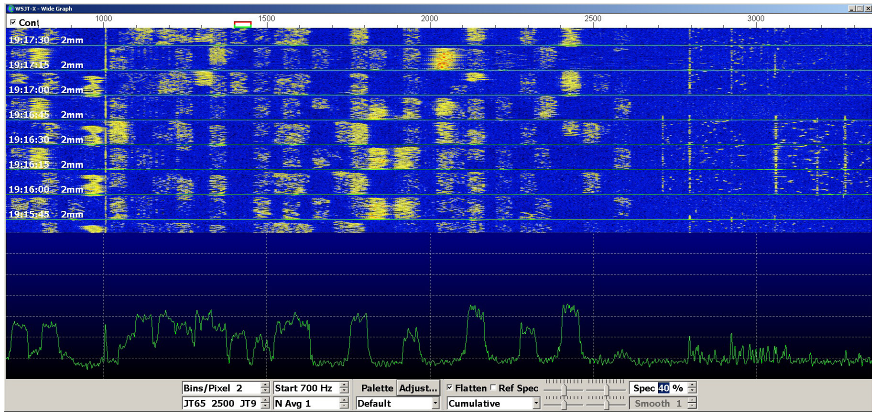


Figure 4 — Many FT8 signals are visible in this view of a *WSJT-X* panoramic waterfall display.

Table 1
Sound Device Software

Windows

RTTY
MMTTY – **hamsoft.ca/pages/mmtty.php**
2Tone – **www.rttycontesting.com/downloads/2tone/**
GRITTY — **www.dxatlas.com/gritty/**

Multimode
MixW — **www.mixw.net**
MultiPSK — **f6cte.free.fr/index_anglais.htm**
Ham Radio Deluxe —**www.hamradiodeluxe.com**
Fldigi – **sourceforge.net/projects/fldigi/files/**

PSK31
DigiPan — **www.apkfollow.com/articles/2020/06/digipan.net.html**
WinWarbler — **www.dxlabsuite.com/winwarbler**
WinPSK — **www.moetronix.com/ae4jy/winpsk.htm**

FT8, FT4
WSJT-X – **physics.princeton.edu/pulsar/k1jt/wsjtx.html**

Digital Voice
FreeDV — **freedv.org**

macOS
MultiMode — **www.blackcatsystems.com/software/multimode.html**
cocoaModem — **www.w7ay.net/site/Applications/cocoaModem/**

Linux
Fldigi — **sourceforge.net/projects/fldigi/files/**

2 Packet Radio

Although the heyday of traditional packet radio has passed, it still survives in the form of networks devoted to specific applications such as emergency communication. Packet radio also finds popular use in the Automatic Packet Reporting System (APRS), addressed in a later section.

2.1 The Packet TNC

In the beginning, there was *X.25*, a protocol for wide-area digital networks that typically communicated over telephone lines. X.25 works by chopping data into strictly defined *packets* or *frames* of information. Each packet is sent to the destination device where it is checked for errors. If errors are discovered, the packet must be sent again to ensure that received data is 100% error free.

In the early 1980s, amateurs began adapting X.25 for over-the-air digital communications. The result was *AX.25*. The new AX.25 protocol worked in much the same way, although it identified each message by sender and destination station call signs and added a Secondary Station ID (SSID) in a range from 0 through 15.

To create and decode these AX.25 packets, hams invented the *terminal node controller*, or *TNC*. TNCs do much more than assemble and disassemble data; they are miniature computers unto themselves. A TNC is also programmed to work within a radio network where there may be other competing signals. For example, to maximize the throughput for everyone on the same frequency, a TNC is designed to detect the presence of other data signals. If it has a packet to send, but detects a signal on the frequency, it will wait until the frequency is clear. TNCs also have a variety of user adjustments and other features, such as a mailbox function to store messages when the operator is away.

Regardless of the changes in packet radio, the TNC is still a vital component. In essence, a TNC functions as a "radio modem." It acts the middleman between your radio and your computer. The TNC takes data from your computer, creates AX.25 packets and then transforms the AX.25 formatted data into audio signals for transmission by the radio. Working in reverse, the TNC demodulates the received audio, changes it back into data, disassembles the AX.25 packets and sends the result to the computer.

For 300 and 1,200-baud applications, TNCs create signals for transmission using AFSK. The most common is 1,200 baud packet, used primarily at VHF. When creating a 1200-baud signal, a *mark* or 1 bit is represented by a frequency of 1200 Hz. A *space* or 0 bit is represented by a frequency of 2200 Hz. The transition between each successive mark or space waveform happens at a rate of 1,200 baud. The frequencies of 1200 and 2200 Hz fit within the standard narrowband FM audio passband used for voice, so that AFSK is accomplished by simply generating 1200 and 2200 Hz tones and feeding them into the microphone input of a standard FM voice transmitter.

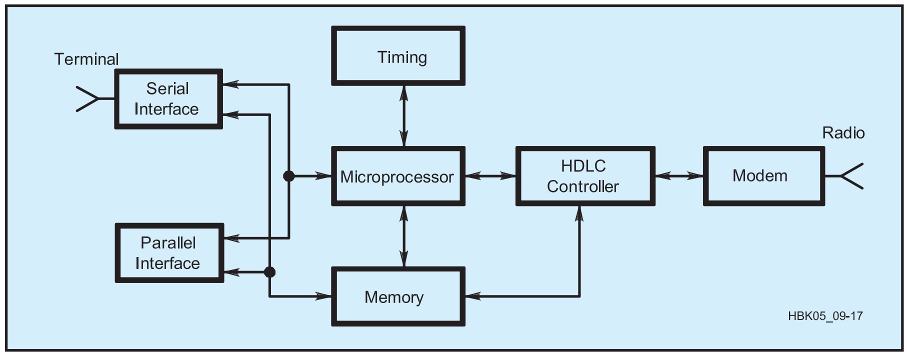


Figure 5 — A block diagram of a typical packet radio TNC.

FSK, not AFSK, is used for 9,600 baud packet. The data signal is filtered and encoded and then applied directly to the FM transmitter, after the microphone amplifier stage, through a dedicated port.

A block diagram of a typical TNC is shown in **Figure 5**. You'll note that it has a serial interface connecting to a "terminal." Most commonly, the terminal is a full-fledged computer running terminal software. Data flows to the computer and vice versa via this interface. At the heart of the TNC is the microprocessor and the attendant *High-level Data Link Controller*, or *HDLC*. The microprocessor is the brain of the unit, but the HDLC is responsible for assembling and disassembling the packets. The modem is simply that — a modulator (changing data to audio tones) and demodulator (changing audio tones back to data).

You can still find TNCs for sale from several manufacturers. There are also several transceivers with packet TNCs built in.

Another solution is a software TNC known as the *AGW Packet Engine*, or *AGWPE*, developed by George Rossopoylos, SV2AGW. The software is downloadable at **www.sv2agw.com/downloads**. With *AGWPE* and a sound device interface as described in this article, it is possible to operate packet radio without a hardware TNC — if your primary packet software application includes the ability to interface with *AGWPE*.

2.2 TNCs and Radios

Tones for 1,200 baud can be fed directly into the microphone input of any VHF FM voice transceiver with an appropriate cable. Packet at 9,600 baud requires an FM transceiver with a dedicated 9,600-baud packet input, as noted earlier.

If you opt to craft your own cable, check your transceiver manual for the wiring diagram of the microphone jack. In most cases, there are separate connections for the audio input and the push-to-talk (PTT) line. (The TNC grounds the PTT line to key your transceiver.) Some transceivers also make receive audio available at the microphone jack for use with speaker/microphone combos. You can use this line to feed audio to the TNC. If it isn't available, you will have to make a separate connection to the transceiver's external speaker jack.

When you are setting up your TNC, be careful about the transmit audio level. Too high a level will create distorted signals that won't be decodable at the receiving station.

An easy way to check your transmitted signal is to use the TNC *calibrate* function. Get to the command mode (CONTROL-C) and enter **cmd: CALIBRATE**. Listen to your transmitted signal with another radio and raise the audio level from the TNC until the received volume seems to stop increasing. Now reduce the audio from the TNC until you can just hear a volume decrease in the receiver. Reduce it a bit more and you're done. Exit the calibrate function.

Some TNCs have an audio output adjustment on the board, some have an adjustment accessible through a hole in the side of the unit, and some have two fixed output levels selectable with a jumper. If one of these does not work, you may have to open the transceiver and find the mic gain control. If this is necessary, be sure you adjust the mic gain control and *not* the deviation control (which can change the bandwidth of your signal).

2.3 TNC Timing

The TNC's TXDELAY parameter specifies the delay interval between the time the TNC keys the radio and the time it starts sending data. Normally 300 – 400 ms are adequate, but some transceivers take longer to settle after the keying line is triggered. If you have a problem being heard and your audio seems normal, try increasing TXDELAY to 400 – 600 ms.

On a busy network, packets and packet acknowledgements fly back and forth at a furious rate. One way to keep interference to a minimum is to manipulate the RESP and DWAIT parameters in conjunction with PERSIST and SLOTTIME to allow staggered transmissions.

RESP is the time delay between reception of a packet and transmission of an acknowledgement. DWAIT sets the delay between the time when activity is last heard on the channel and the moment your radio transmits. You should set values of RESP and DWAIT to the values recommended in your area (check with the person managing the local network or PBBS). Your TNC probably accepts a value in "counts" rather than in milliseconds, so don't forget to convert by the proper value to arrive at the correct timing value in milliseconds. For example, if you have been asked to set DWAIT to 600 ms and the units of DWAIT for your TNC are 10 ms per count, then you would command DWAIT = 60.

Most TNCs contain commands called PERSIST and SLOTTIME, which help enormously in avoiding interference. PERSIST sets the probability that a packet will be transmitted during a given time interval called a SLOTTIME. The parameter SLOTTIME governs the interval between transmission timing "slots." Initially, PERSIST should be set to approximately 64 and SLOTTIME to a value of about 10, which is equivalent to 100 ms.

FRACK (frame acknowledgement) should be set to 6 and RETRY to 10. FRACK sets the number of seconds between retries and RETRY sets the number of times your TNC will try to send a packet and gain acknowledgement of it before it gives up and disconnects

2.4 Monitoring

Start by listening to an active packet frequency in your area. With the radio cable connected, turn on your radio and increase the receiver volume partway. Some TNCs include an LED indicator that shows that the TNC is receiving audio. Turn up the squelch control on the radio until the LED is extinguished. Tune the radio to any odd numbered frequency between 144.91 and 145.09, or between 145.61 and 145.79 MHz, and set the radio for simplex operation. Your best bet may be to search for a DX PacketCluster, or try monitoring APRS activity on 144.39 MHz. When you hear the buzzing packet signals and see text on your screen, you'll know you've hit the jackpot.

Depending on the type of activity you are monitoring, you may see what appears to be nonsense. If you are monitoring APRS, you'll see strings of numbers. These are latitude/longitude position reports. On PacketClusters, you'll see DX call signs and frequencies.

2.5 "Connected" vs "Unconnected"

When discussing TNCs and networks, it is important to understand the difference between connected and unconnected communication.

If you are simply monitoring local packet transmissions, your TNC is in an *unconnected* state. What you see is what you get. If a signal is garbled by noise or interference, you'll see nothing on your screen (unless you've enabled the PASSALL function, in which case you'll see gibberish). If you transmit an unconnected packet, the signal simply leaves your antenna destined for nowhere in particular. Some stations may decode it, some may not.

When your TNC is operating in a *connected* state, everything changes. When you are connected, your station is linked to another station in a "virtual" sense. In a connected state, every packet you send is intended specifically for the receiving station (even though others can see it).

When your TNC transmits a packet, it starts a countdown clock. If the clock reaches zero before your TNC receives an acknowledgement (known as an *ACK*) that the packet arrived without errors, it will send the same packet again. When the packet is finally acknowledged, the TNC will send the next packet. And so it goes, one packet after another. The operator at the other station may also be sending packets to you since this communication process can flow in both directions simultaneously.

The big advantage of the connected state is that data is delivered error-free. One packet station can connect to another directly, or through a series of relaying stations. Error-free can be a disadvantage, too. Specifically, a connected state works best when signals are strong and interference is minimal.

Remember that if too many packets are lost — by either not arriving at all or arriving with errors — the link will fail. That's why AX.25 packet radio tends not to work well on the HF bands. With all the noise, fading and interference, packets are often obliterated in route.

Unconnected packets are ideal for applications where you are transmitting essentially the same information over and over. Since unconnected packets can be decoded by any station, they are an excellent means of disseminating noncritical data (data that doesn't need guaranteed error-free delivery) throughout a given area. If a station fails to decode one packet, it merely waits for the next one. APRS uses exactly this approach.

3 The Automatic Packet Reporting System (APRS)

The Automatic Packet Reporting System, better known as *APRS* (**aprs.org**), is the brainchild of Bob Bruninga, WB4APR. In fact, APRS is a trademark registered by WB4APR. The original concept behind APRS involved tracking moving objects, and that's still its primary use today. However, APRS has been evolving to become an amateur network for other applications such as short text messaging, either from ham to ham, or between hams and non-hams through APRS internet gateways.

Mobile APRS stations communicate their positions based on data provided by on-board Global Positioning System (GPS) receivers. The GPS receivers are attached to either packet radio TNCs or simplified packet devices known as *position encoders*, which in turn are connected to transceivers (see **Figure 6**). At receiving stations, various APRS software packages decode the position information and display the results as icons on computer-generated maps such as the one shown in **Figure 7**. When a station moves and transmits a new position, the icon moves as well.

Virtually all APRS activity takes place today on 144.39 MHz using 1200-baud packet TNCs and ordinary FM voice transceivers. In areas where the APRS network is particularly active, you may hear traffic on 445.925 MHz as well. There is also some activity on HF.

3.1 Setting Up an APRS Station

If you own a 2-meter FM voice transceiver, you already have the primary component of your APRS station. Tune your radio to 144.39 MHz and listen for packet transmissions. If you hear them, it means you have APRS activity in your area. To decode APRS packets,

Figure 6 — A 2-meter FM handheld transceiver attached to an APRS position encoder.

Figure 7 — A snapshot of APRS activity using *UI-View* software. Note the icons representing various mobile and fixed stations.

you'll need a TNC, or a transceiver that has a TNC built in.

There are several packet-ready transceivers that also have built-in GPS receivers. However, if all you want to do is monitor APRS activity, you do *not* need a GPS receiver. If you want to participate in the local APRS network from a fixed (non-moving) station such as your home, you still do not need a GPS receiver. Just determine your home latitude/longitude coordinates and you can use them to establish the location of your home station on the network. There are numerous sites on the internet that will convert your home address to a correct latitude and longitude.

The final component of your APRS station is software. You'll need software to display the positions of APRS stations, along with other information contained in their transmissions. APRS software is also essential if you want to communicate over the APRS network from a fixed or portable station. Note, however, that APRS software is *not* necessary for mobile stations that wish to merely transmit APRS beacons for tracking purposes. That function is carried out automatically using the GPS receiver and APRS-compatible TNC or tracking device; it does not depend on software.

The most popular APRS Windows program is *UI-View*. *UI-View* was created by the late Roger Barker, G4IDE. You'll find it online at **www.ui-view.net/**. The 16-bit version is free for downloading. To use the 32-bit registered version, hams are asked to donate to their local cancer charities. Details are available on the *UI-View* website. Another worthwhile program for Windows is PinPoint APRS, which you will find at **www.pinpointaprs.com**. For the Macintosh, there is *MacAPRS* available from **www.winaprs.com**. For Linux there is *Xastir,* available for download at **sourceforge.net/projects/xastir/**.

APRS software, regardless of the operating system, is designed to talk to the packet TNC, processing the incoming APRS data and creating icons on your computer screen. The application also uses the TNC to transmit APRS data. This means that the software and the TNC must communicate with each other at the same baud rate. Every APRS application has a setup menu to program the correct parameters for various TNCs.

Depending on the software, there may be other features such as logging or messaging. APRS software changes rapidly, so check the help file or manual for specific instructions.

3.2 Maps and APRS

No matter which APRS software you choose, one critical aspect is the mapping function itself. To get the most from APRS, your software maps must be as comprehensive as possible, preferably with the ability to show detail down to street level.

Downloadable APRS software applications generally do not come with detailed maps. Detailed map files are numerous and large, impractical to bundle with every APRS program. Instead, most applications are designed to import user-created custom maps, or to work with existing commercial mapping programs such as Microsoft *Streets*, Delorme *Street Atlas* and *Precision Mapping*.

UI-View, for example, can automatically load and display maps from *Precision Mapping*. You must purchase and install *Precision Mapping* on your PC, then download and install a small *Precision Mapping* "server" application into *UI-View*.

Each APRS transmission includes characters that define the type of map icon that will be displayed at the receiving end. If you are operating a fixed station, your APRS software will allow you to choose your icon. If you are a mobile station using a traditional TNC, you'll need to define your chosen icon in your beacon statement. APRS-compatible TNCs give you the ability to do this. APRS position encoders also allow you to choose your icon when you program the unit. Your mobile icon might be a car, boat or airplane.

3.3 APRS Position Encoders

If you aren't blessed with a transceiver that offers a built-in TNC and GPS receiver, you can create a mobile APRS station with any VHF FM transceiver, a TNC and a GPS receiver. Wire everything together, connect an antenna and dc power and you're set. For hams on the go, however, it's common to replace the full-fledged TNC with an APRS *position encoder*, also referred to as an APRS *tracker*. A position encoder is a compact device designed for one purpose: to receive data from the GPS receiver, assemble APRS packets from the data and create modulated signals for use by the transmitter. Some position encoders include GPS receivers in their designs. You'll even find position encoders that are complete packages incorporating tiny GPS receivers and low power FM transmitters.

To use a position encoder, you must program it the same way that you initially program a TNC. Like TNCs, position encoders connect to computer serial or USB ports for programming and most come with their own programming software. You must enter your call sign and other information such as your beacon interval (how often you want the position encoder to transmit your position). Most position encoders allow you to set the beacon interval to a certain amount of time (say, every two minutes). Some position encoders can be configured to transmit position beacons after a certain distance (every mile), or whenever the vehicle turns a corner.

3.4 APRS Networking Tips

One of the key features of APRS is that while it uses AX.25 to transport its messages, it essentially ignores all the AX.25 connection-oriented baggage. Unlike traditional packet radio, APRS stations do not establish "connections" with each other. Instead, APRS packets are sent to no one in particular, meaning to *everyone*.

Every APRS station can function as a digital repeater, or *digipeater*. So, if it receives a packet, it will retransmit the packet to others. As other digipeaters decode the same packet, they will also retransmit and spread it further. This is known as *flooding* and is illustrated in **Figure 8**.

As an APRS user, you can set up your station to address its packets through specific digipeaters according to their call signs. But when you're traveling, how do you determine which digipeaters you should use?

PATHS AND ALIASES

In the packet world, nodes and digipeaters can have *aliases*. A digipeater call sign may be W1AW-1, but it can also carry an alias, using the MYALIAS command in the digipeater TNC. Perhaps the digipeater alias would be NEWNG (meaning the ARRL HQ home town of Newington). You can route packets through the digipeater by addressing them to W1AW-1, or simply by addressing them to NEWNG. Any station that is set up to respond to an alias can handle your packets automatically, even if you don't know its call sign.

Unlike typical packet use of aliases in which a given single station has a specific alias, APRS specifies standard digipeater aliases that nearly all stations use. This means that you can travel anywhere in the country and still participate in the APRS network without knowing digipeater call signs. (Otherwise, you'd have to reconfigure your TNC whenever you moved from one area to another.)

The most common APRS digipeater alias is *WIDEn-N*. The letters "n" and "N" represent numbers. The first (left-most) "n" designates how many WIDE digipeaters will relay your packets, assuming they can receive them. WIDE2, for instance, is the same as saying that you want your packets relayed through *two* WIDE digipeaters. The second "N" is the *Secondary Station Identifier* (SSID). The SSID is used in APRS networks as a means of limiting how often (and how far) a packet can be repeated.

Here's how it works. Each time your packet traverses a WIDEn-N digipeater, the digipeater subtracts 1 from the SSID as it retransmits. The next digipeater deducts 1 and so on until the SSID reaches zero, at which time the packet will not be repeated. This has the effect of limiting the flood radius. See **Figure 9**.

When you configure a TNC or position encoder for use with APRS, you can use these aliases to set up the paths for the beacon packets you'll be transmitting. In most devices this is accomplished with the UNPROTO parameter, sometimes simply referred to as the "Path." If you are a fixed station (a station at home, for instance), set your path as **WIDE2-2** (or with a traditional TNC UNPROTO statement, set it to APRS VIA WIDE2-2). This designates that your reports will be relayed by *two* WIDEn-N digipeaters and limits the spread beyond those repeaters to just two retransmissions. Set your TNC to beacon once every 30 minutes. That's enough for a fixed station.

If you are running APRS from a car, try **WIDE1-1,WIDE2-1** (or APRS VIA WIDE1-1,WIDE2-1). WIDE1-1 ensures that your packet will be picked up by at least one local digipeater or a home station acting as a fill-in digipeater and relayed at least once. WIDE2-1 gets your packet to another, presumably wider-coverage digipeater, but limits the retransmissions beyond this point. It's wasteful of the network to set up wide coverage for a station that is rapidly changing its position anyway. Mobile stations that are in motion

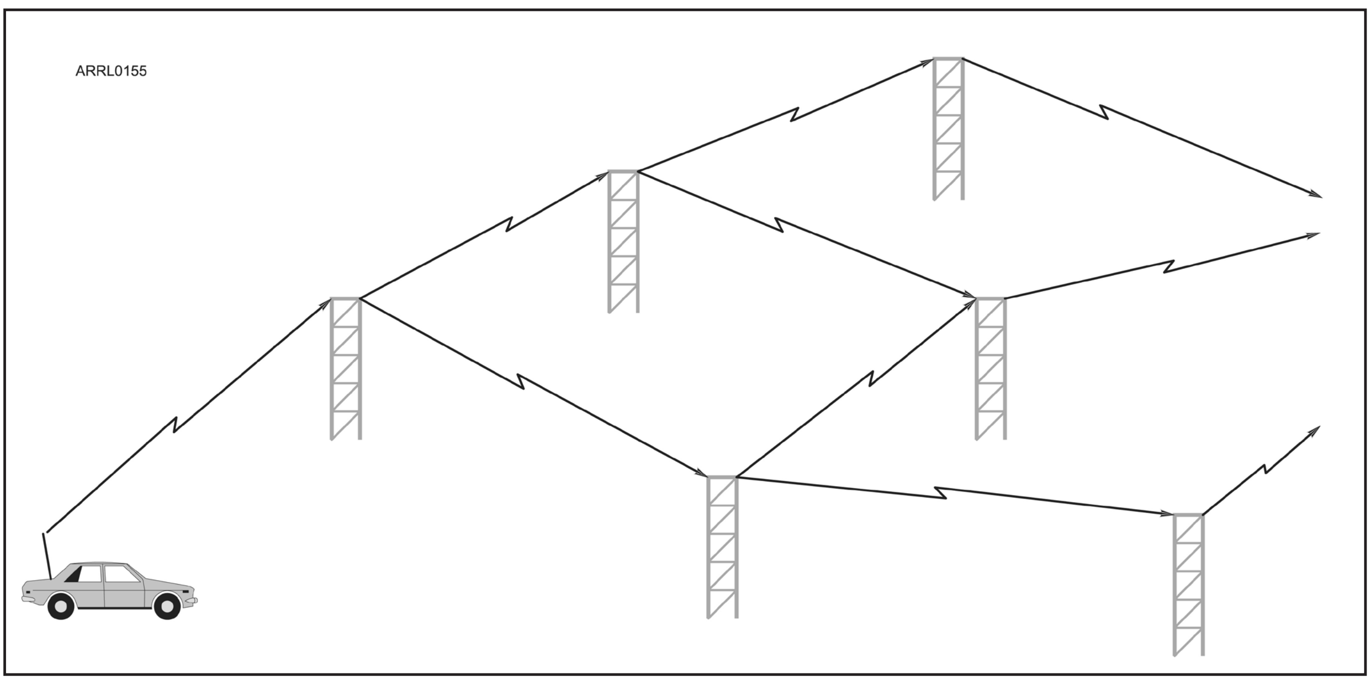


Figure 8 — In this example, an APRS packet is transmitted by a mobile station and is retransmitted by a nearby digipeater. Depending on how the mobile operator configured his TNC or tracker's path, the packet will be picked up and repeated by several other digipeaters. This is known as flooding.

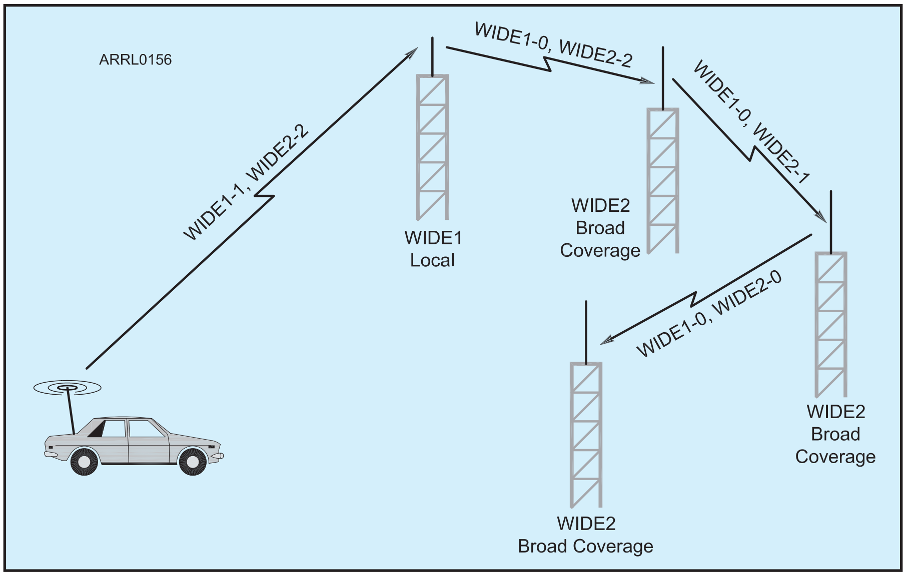


Figure 9 — By using the WIDEn-N system, we can limit packet flooding in a local network and greatly reduce congestion. The mobile station in this example has his path set as WIDE1-1,WIDE2-2. Notice how his packet propagates through the network and how the SSID number is reduced by one each time the packet is repeated through a digipeater with a corresponding alias. When it reaches the third WIDE2 digipeater, the counters all reach zero and digipeating stops.

should also limit their beacon rate to once every 60 seconds, or once per mile, whichever comes first.

Never invoke extremely wide coverage, such as a WIDE5-5 path, unless you are way out in the hinterlands and need every relaying station available to get your packets into the network.

Another popular alias is the "single state" — SSn-N — to limit the spread of your packets to a specific state or area within a state. To keep packets within the state of Connecticut, for example, you could use the SSn-N alias in a path statement, like this: CT1-1, CT2-2. This path assures that local Connecticut stations (CT1-1) will repeat the packets, and that broad-coverage stations (CT2-2) will relay them throughout a large portion of the state. APRS digipeaters outside Connecticut, however, will not respond to these packets because they won't recognize the CT2-2 alias (although they will recognize an alias for their state). The packets will still be heard across state borders but will not be digipeated or add to the packet activity in a neighboring state.

4 PACTOR

PACTOR is a type of HF digital communication that, unlike many other HF modes, offers error-free text and file transfers. PACTOR communication resembles packet radio in that it establishes a "link" between two stations. Data is sent in discrete frames and each frame is acknowledged by the receiving station. This rapid back-and-forth exchange creates PACTOR's distinctive *chirp-chirp-chirp* signal. Through robust coding and sophisticated modulation techniques, PACTOR is often able to maintain the link even in the face of significant noise, fading and interference.

PACTOR I, introduced in the early 1990s, has been largely superseded by PACTOR II and III, which provide faster throughput in difficult conditions. Both PACTOR I and II generate signals that occupy 500-Hz bandwidths. PACTOR III offers even more efficient communication, but its signal occupies a bandwidth in excess of 2000 Hz.

The most recent version of PACTOR is PACTOR IV, which offers much greater throughput even when compared to PACTOR III. However, at the time this was written, FCC rules did not permit signaling rates greater than 300 baud below 28 MHz and PACTOR IV exceeds this limit. The ARRL has petitioned the FCC to change the rules to permit the use of higher signaling rates below 28 MHz, but at the time of this writing, the FCC had yet to act.

To set up a PACTOR station, you must purchase a stand-alone multimode controller that includes the PACTOR mode. Controllers manufactured by Kantronics (**www.kantronics.com**) and Timewave (**www.timewave.com**) include PACTOR I. For PACTOR II or III, you must purchase a controller made by Special Communications Systems (**www.scs-ptc.com**), the original developers of PACTOR.

Figure 10 illustrates a typical PACTOR station built around a multimode controller. The computer is functioning only as a dumb terminal for the controller in this application, so it does not have to be particularly powerful. All that is required is basic terminal software as described in the section on packet TNCs. More sophisticated software applications can provide a smoother, easy-to-use interface, but these require more capable computers.

A PACTOR station requires an SSB transceiver capable of switching from transmit to receive within approximately 30 ms. Most modern HF transceivers can meet this requirement, but not all. *QST* magazine Product Reviews test this specification for all HF transceivers. The transmit audio supplied by the controller can be fed to the microphone or accessory jack to operate AFSK, which is the standard procedure, although PACTOR I can also operate using FSK.

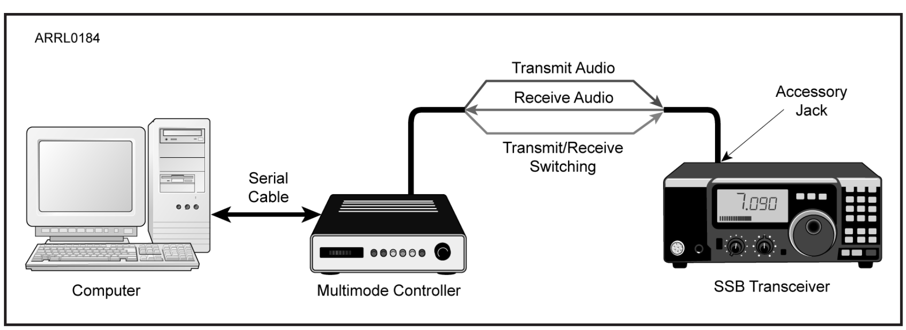


Figure 10 — A PACTOR station with a multimode controller. For PACTOR II and III operation, the controller must be a model manufactured by the SCS Corporation.

5 High Speed Multimedia (HSMM)

Wireless networking using IEEE 802.11 standards has seen explosive growth during the last several years. Coffee shops, fast-food restaurants, hotels, airports and many other high-traffic locations now include wireless internet (WiFi) hotspots. Some hotspots offer free internet access while other charge an access fee, payable with your credit card.

Wireless networks are also popular at home. Establishing a home network is as simple as installing a *wireless router* to manage the data flow from the broadband internet connection. The router allows one or more traditional desktop computers to tap the broadband connection through wired (usually Ethernet) access while simultaneously making the internet available wirelessly to one or more laptops.

All these home networking devices — routers, wireless access cards, and so on — are unlicensed FCC Part 15-regulated transceivers with RF outputs measured in milliwatts. A number of their channel frequencies overlap two amateur radio bands: 2.4 and 5.8 GHz. This means that hams can put them to work as "transceivers" under our Part 97 rules. See **Table 2** for specific frequencies.

By using consumer-grade routers, low-loss coaxial cable and gain antennas, hams can quickly establish high-speed, long-distance wireless networks on these 2.4 or 5.8 GHz frequencies. This type of operating is referred to as *high speed multimedia*, or *HSMM*.

With HSMM amateurs can operate several different modes at the same time, and usually do. HSMM is generally IP-based, and given enough bandwidth, radio amateurs have the capability to do the same things with HSMM that are done on the internet.

- *Audio*: This is technically digital voice, since it is two-way voice over an IP (VoIP) network similar to EchoLink and IRLP networks used to link many amateur radio repeaters over the internet. (VoIP is covered in a later section.)
- *Video*: Motion and color video modes, are called amateur digital video (ADV). This is to distinguish it from digital amateur television (DATV). DATV uses hardware digital coder-decoders (CODEC) to achieve relatively high-definition video like *entertainment quality TV*. The usual practice in HSMM radio is to use a far less expensive (often free) PC-based software video codec to achieve *video communications quality* signals in much smaller bandwidths.
- *Text:* Text exchanges via a keyboard are often used in HSMM radio, but they are similarly called by their internet or packet radio name: *Chat mode*, and if a server is available on the network, e-mail can also be exchanged.
- *Image*: Image file transfers using file transfer protocol (FTP) can also be done, just as on the internet.
- *Motion video*: FTP of MPEG files can provide one-way video streaming of short video clips.
- *Remote control*: Individual devices or even complete stations can be remotely controlled.
- *Mesh networking*: A mesh is a wireless cooperative communication infrastructure in which each station functions as a relay, allowing the entire network to cover a large area. See **Figure 11**. This type of infrastructure can be decentralized (with no central server) or centrally managed (with a central server). Both are relatively inexpensive and very reliable. Individual mesh stations (nodes) act as repeaters to transmit data from nearby nodes. The reliability comes from the fact that each node is connected to several other nodes. If one node drops out of the network, due to hardware failure or any other reason, its neighbors simply find another route. Capacity can be increased by simply adding more nodes. See the IP-based Microwave Networking section of the **Digital Protocols and Modes** chapter for information on current types of HSMM networks.

5.1 A Basic HSMM Radio Station

For the sake of simplicity, we'll discuss an HSMM setup with two stations operating in a host/client configuration. See **Figure 12**.

At the time of this writing, one of the most popular wireless router for HSMM applica-

Table 2
Wireless Network Frequencies in Amateur Bands

Channel	*Center Frequency*
1	2.412 GHz
2	2.417 GHz
3	2.422 GHz
4	2.427 GHz
5	2.432 GHz
6	2.437 GHz
Channel	*Center Frequency*
132	5.660 GHz
136	5.680 GHz
140	5.700 GHz
149	5.745 GHz
153	5.765 GHz
157	5.785 GHz
161	5.805 GHz
165	5.825 GHz

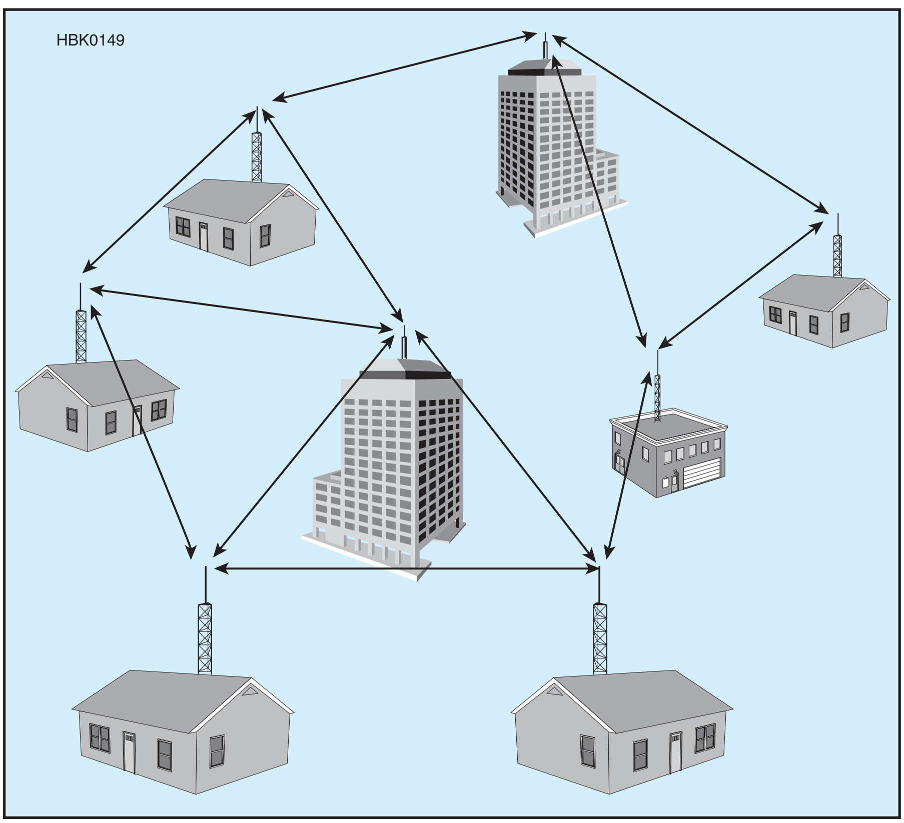


Figure 11 — An HSMM mesh network is highly decentralized. Individual mesh stations (nodes) act as repeaters to transmit data from nearby nodes. If one node drops out of the network, the other nodes will automatically re-route the traffic.

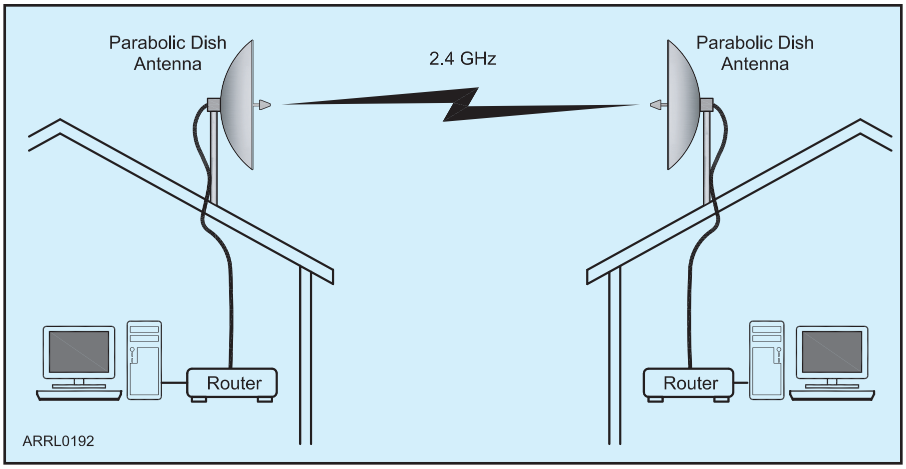


Figure 12 — This is a simplified diagram of a typical HSMM link between two stations. Note that high gain parabolic dish antennas are being used in this example.

Figure 13 — RP connector adapters.

tions is the Linksys model WRT54GL. It is a combination unit consisting of a wireless access point (AP) or hub coupled with a router. As with other routers, your host PC or laptop connects directly to it using a standard Ethernet cable. If the PC is also connected to the internet, then it may also perform the function of a *gateway*.

The WRT54GL is a Linux-based model that supports firmware upgrades. HSMM-active amateurs have been creating their own WRT54GL firmware to support special applications (such as mesh networking). They are effectively "modifying" the WRT54GL in ways Linksys could not have imagined! You do not have the use a WRT54GL to explore HSMM, however. Any wireless router will do. The advantage of the WRT54GL is only that it is widely available and easily modifiable.

The first step in configuring your router for HSMM is to disconnect both flexible antennas that came with the unit and replace them with a high-gain directional antenna system. To connect a gain antenna, you are going to be become familiar with RP (reverse polarity) connectors. These connectors appear to be male connectors on the outside, but they have a socket rather than a pin for the center conductor. RP connectors are used by the manufacturers to discourage Part 15 owners from using the equipment in ways for which it was not WiFi certified.

As licensed radio amateurs, we can modify the system to accomplish our specific requirements within the amateur bands. To connect coaxial cable for an external antenna, you can use an adapter or short jumper cable assembly with appropriate connectors. The adapter shown in **Figure 13** has an RP connector on one end and a standard SMA connector on the other.

There are often two antenna ports on wireless devices. These are used for *receive space diversity*. The wireless device will normally automatically select whichever antenna is receiving the best signal at any specific moment. Which do you connect to?

The transmitted signal from the wireless router always goes out the same antenna port. It does not switch. In other words, most wireless devices use space diversity on the receive side and not on transmit. Some access points/wireless routers will allow you to manually (via software) select the antenna port that is used for transmission. When there is no obvious choice, you will need to find some means of detecting which antenna is the transmit antenna port with RF output power present. That is the port to connect to your gain antenna.

Be sure to use low-loss coaxial cable, and keep the run as short as possible. Coaxial cable loss at these frequencies can be quite high, so you are often better off mounting the router directly at the antenna, connecting to it through a very short length of cable. From there you can run the Ethernet cable back to your computer.

SOFTWARE CONFIGURATION

You will need to use either the network configuration software from the router manufacturer or the configuration tool that is part of your operating system. Some routers offer user-friendly access through internet browsers, which is particularly convenient. Regardless of the approach, the goal is to access and change your router's settings.

SSID: The AP/wireless router host software is provided with an *SSID* (Service Set Identifier) that many Part 15 stations turned off for somewhat higher security. But radio amateurs should leave it **ON**. Enter your call sign as the SSID and use it for the station identification. It constantly broadcasts your call thus, providing automatic and constant station identification. There are 32 characters available for use in this field so more information such as your group's name can be entered too, including spaces and punctuation. If the router asks if you want to enable the broadcast, click **YES**. (Note that SSID in the HSMM world has a somewhat different meaning than SSID in the packet radio community. Among packet users, SSID is defined as Secondary Station Identifier.)

ESSID: Some manufacturers use this term in place of SSID to put emphasis on the fact that the SSID is the name for your *network,* not for a specific wireless AP/router.

Access Point Name: When this field is made available (by default is it blank) it is for you to enter a description. This may be handy if you have deployed more than one AP in your network all with your call sign as the SSID. It would allow you to tell them apart. Otherwise, just leave it blank.

Channel: To avoid interfering with other services as much as possible, we need to also look at channel selection. Most HSMM activity is concentrated on 2.4 GHz, using channels 1 through 6, which are within the 2.4 GHz ham allocation.

It is probably best to avoid channel 6. It is the most common manufactures default channel setting and 80% or more of your neighbors will be using it for their household wireless local area network (WLAN). Channel 1 is used by most of the remaining manufacturers as their default channel, so avoid that too. The result is most radio amateurs use channels 3 or 4 depending whether there is a WISP (Wireless Internet Service Provider) operating in their area. Often a WISP will use one of these intermediate channels with a highly directive antenna for back-haul or other purposes. If so, you may wish to coordinate with the WISP and arrange to use some other channel rather than the one specifically used by the WISP. It is not a perfect solution because of all the overlap, but it is a good faith effort to keep most of your stronger signal out of anyone's home, business or governmental WLAN traffic.

WEP: This stands for Wired Equivalent Privacy. Despite all the horror stories you may have read in the press, this encryption method provides more than adequate means to economically achieve authentication and thus keep most freeloaders off your network. If you live in the country, you may not need to enable this capability. In an urban environment, it is probably a good thing to do so that you need not constantly monitor every bit of traffic coming over the network to ensure that it originates from an amateur radio station. Mixing traffic with another service that shares the same frequency band is not a generally accepted practice except in times of emergency. Therefore, it is often necessary for HSMM radio stations to encrypt their transmissions. This is not to obscure the meaning of the transmission or hide the information. In amateur HSMM applications, the purpose of using WEP is only to block access by non-hams. Amateur HSMM networks openly publish their encryption keys for other hams to use.

Most wireless routers will allow for the use of multiple WEP keys, typically up to four. This will allow you to configure the device so that different client stations have different access authority. For example, club members may have one level of access, while a visiting radio amateur may be given a lesser access. Most HSMM radio groups have just one WEP key and everybody gets that one.

Remember that when it comes to the length of the WEP or other key used, our main purpose is to provide a simple and economical means of authentication already available on the wireless devices. In other words, it is to ensure that only Part 97 stations and not Part 15 stations auto-associate or auto-connect with our HSMM radio node. The shorter the WEP key, the better. This makes it easy to use and remember. During early HSMM radio experimenting around the year 2000, the shortest possible key (5 characters) was used: HSMM-

Authentication Type: Some routers will ask for the type of authentication you want to use such as *shared key*, *open system*, and *both*. Click on **SHARED KEY** because you will be sharing the WEP/WPA key with all radio amateurs who wish to access your HSMM radio node.

*DHCP***:** Some routers will ask if you want Dynamic Host Configuration Protocol enabled. This is the function that assigns IP addresses. Unless you have another source of the DHCP function on your network, you will want to **ENABLE** this function.

Antenna Selection: A few wireless routers with dual antennas will ask you select an antenna. The default is normally receive space diversity. Because we are going to connect an outside gain antenna, you want to make a selection. Otherwise, you will need to identify which antenna is the actual transmit antenna and connect the feed line to that port, as discussed previously.

Mac Address Filter: Some wireless routers will allow for this security measure, but it is troublesome to administer it, so it is recommended that you not bother enabling this function. Use WEP or some other method of encryption using the guidelines discussed previously.

Output Power: Some wireless routers will allow you to set this power level, often up to 100 mW. As with all other radio amateur operations use only the minimum power needed to accomplish your mission.

THE FAR END OF THE LINK

The computer at the client end of the HSMM link need only have a wireless networking adapter, not a separate router. These transceivers/wireless adapter cards usually come in three forms:

1) One form is called a PC card. Earlier these were called PCMCIA cards, but more recent terminology is to simply call them laptop PC cards.

2) Another type of transceiver/adapter comes with a USB interface, such as the one shown in **Figure 14**. This is often considered a superior interface for most HSMM stations. The reason for this has nothing to do with the quality of the transceiver, but rather the fragile nature of the tiny connectors found on PC cards. They are not really designed for frequent plugging and unplugging. Without extreme care, they can be easily torn out.

3) Linksys and other manufactures also produce similar cards for the expansion slot on the rear of your desktop PC too.

Figure 14 — At the client end of the HSMM link, a simple wireless transceiver/adapter will do.

Regardless of the device you choose, it must have an antenna that is removable or has an external antenna port of some type.

To connect to the AP/wireless router in an HSMM radio network, the wireless computer user(s) at the far end must exit ad-hoc mode and enter what is called the *infrastructure mode*, in their operating software. Infrastructure mode requires that you specify the radio network your computer station is intended to connect to (the host station's call sign), so set your computer to recognize the SSID you assigned to the AP/wireless router to which you wish to connect.

As described previously, you may need to use an adapter to connect the card or interface antenna connector to the coaxial cable running to the antenna. At the other end, most 2.4 GHz antennas come with a standard female N-series connector.

Team up with a nearby radio amateur to test. Do your initial testing in the same room together making sure the link-up is working. Then as you increase distances going toward your separate station locations, you can coordinate using a suitable local FM simplex frequency. You will increasingly need this communication to assist with directional antenna orientation as you get farther apart.

5.2 Running High Power

It is tempting for some radio amateurs to think that if they run higher power, they will get better range out of their HSMM radio station. This is not always the case. There are many factors involved in range determination when operating at 2.4 or 5.8 GHz. The first and most significant of these is the lay of the land (topology) and path obstructions. Running additional power is unlikely to correct for either of these impediments.

Second, running higher power to improve signal link margins often requires that this be done at both ends of the link to obtain meaningful results.

Third, most RF amplifiers for use with 802.11 are of the BDA (bidirectional amplifier) type, such as the one shown in **Figure 15**. They amplify both the incoming signal and the outgoing signal, and to get maximum effectiveness out of such devices they must be mounted as close as possible to the antenna.

Fourth, 802.11 signals from such inexpensive broadband devices often come with significant sidebands. This is not prime RF suitable for amplification. A tuned RF channel filter should be added to the system to reduce these sidebands and to avoid splatter.

Also, if your HSMM radio station is next door to an OSCAR satellite ground station or other licensed user of the band, you may need to take extra steps to avoid interfering with them, such as moving to channel 4 or even

Figure 15 — A 2.4 GHz bidirectional amplifier.

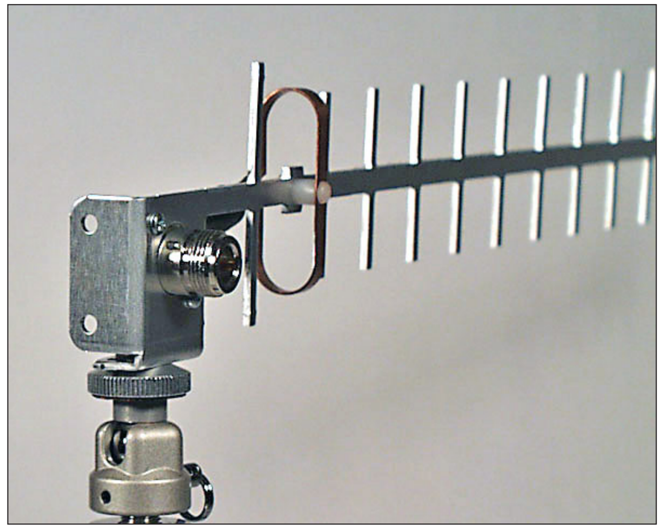


Figure 16 — Directional antennas, such as this MFJ 2.4 GHz Yagi, are best for HSMM links.

channel 5. In this case a tuned output filter may be necessary to avoid not only causing QRM, but also to prevent some of your now amplified sidebands from going outside the amateur band, which stops at 2450 MHz.

Do not use higher power as a substitute for higher antenna gain at both ends of the link. Add power only after all reasonable efforts have been taken to get the highest possible antenna system gain and directivity.

5.3 HSMM Antenna Options

There are several factors that determine the best antenna design for a specific HSMM radio application. Most commonly, HSMM stations use horizontal instead of vertical polarization because it seems to work better. In addition, most Part 15 stations are vertically polarized, so this is sometimes thought to provide another small barrier between the two different services sharing the band. With multipath propagation is it doubtful how much real isolation this polarization change provides.

More importantly, most HSMM radio stations use highly directional antennas (**Figure 16**) instead of omnidirectional antennas. Directional antennas provide significantly more gain and thus better signal-to-noise ratios, which in the case of 802.11 modulations means higher rate data throughput and greater range. Higher data throughput, in turn, translates into more multimedia radio capability. Highly directional antennas also help amateurs avoid interference from users in other directions.

6 Automatic Link Establishment (ALE)

Automatic link establishment (*ALE*) is a "system" that incorporates digital techniques to establish a successful communications link. ALE is not a digital mode *per se*, but the primary software is sound device based.

To understand ALE, consider how HF propagation changes. A band can open over a path one hour, but close the next. A signal may be fully readable on one band, but inaudible on another. With that in mind, imagine two stations, one in New York City and one in Los Angeles, that wish to communicate on the HF bands (see **Figure 17**). Basic propagation guidelines give a general idea of which bands might be best, but operators at both ends of the path will have to experiment, calling on several bands until they find a frequency that supports a transcontinental path.

ALE automates this process, allowing the stations under computer control to automatically call each other on different bands until contact is established. In the example, the operator in New York City initiates the ALE call by entering the call sign of the receiving station into his ALE software, along with some likely frequencies. The call sign becomes the *Selective Call*, or SELCAL. The ALE software switches the transceiver to the first frequency in the queue and transmits the chosen SELCAL. If there is no response, the software will step the transceiver to another frequency and try again.

At the same time, the receiving station in Los Angeles has been automatically scanning specific ALE frequencies, listening for its call sign. When it finally decodes something that resembles its call sign, it stops scanning and listens since the ALE data burst contains several repetitions of the call. When it finally decodes its full call sign, the Los Angeles station transmits a "handshake" signal to the

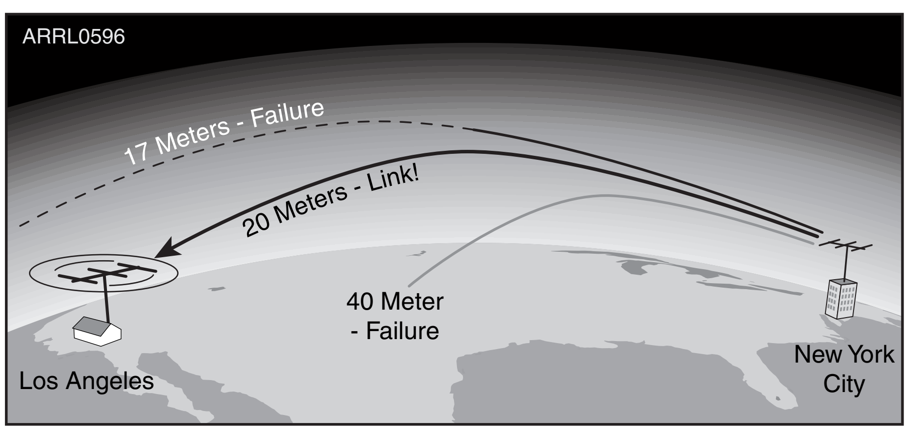


Figure 17 — The ALE calling station typically scans through a list of frequencies, calling on each one until a path to the receiving station is found.

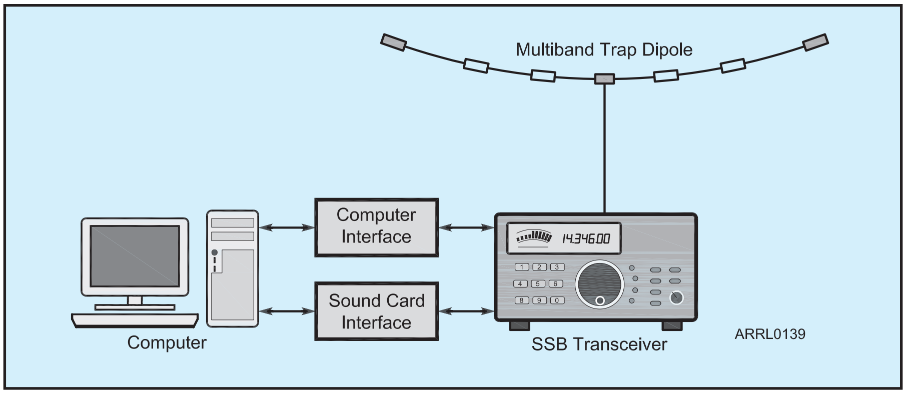


Figure 18 — An optimum ALE station has a computer-controlled transceiver and a multiband antenna system. In this example, the station is using a multiband trap dipole antenna. Note that a separate interface is needed between the computer and transceiver for frequency control.

New York City station to confirm that a link has been established. All scanning stops and the ALE software sounds an alarm to call the operators to their stations.

ALE is a good tool for HF net operation in both routine and emergency applications where many stations may be standing by for calls. Think of ALE as being analogous to a VHF FM scanning transceiver with a digitally coded squelch that automatically scans several repeaters. Unless it hears the correct code, the radio remains silent. With amateur ALE you can call individual stations or entire groups of stations.

In addition to its application as a selective calling system for HF voice, ALE can also be used as a pure digital mode for exchanging text and images.

AMATEUR ALE SOFTWARE

ALE has been used by the military and government for years, and hams have generally adopted the government ALE standards: FED-STD-1045 or MIL-STD 188-141. In 2001, ALE captured the attention of the amateur radio world with the release of *PCALE* software by Charles Brain, G4GOU. *PCALE* not only controls your transceiver, it uses your computer sound device to decode ALE signals and generate them for transmission. From the standpoint of assembling a station for ALE, it works best when two conditions are satisfied: 1) you have a multiband HF antenna system; the more bands the better; and 2) your HF transceiver can be controlled by computer.

Although you can use ALE on just one frequency, the real power of ALE comes into play when your station is able to operate on multiple bands under computer control. A typical setup is shown in **Figure 18** and includes a radio control interface as well as a sound device interface. With *PCALE* controlling your radio, it will automatically change frequencies, monitoring as many bands and frequencies as you have programmed into the software. For the latest on *PCALE* and ALE for amateur radio, do an internet search. Note that all ALE transmissions use USB. ALE must be used in accordance with the rules for stations operating automatic control.

7 D-STAR

The D-STAR digital protocol began in 2001 as a project funded by the Japanese Ministry of Posts and Telecommunications to investigate digital technologies for amateur radio. The research committee included representatives of the Japanese amateur radio manufacturers and the Japan Amateur Radio League (JARL). JARL is the publisher of the D-STAR protocol.

All radio manufacturers are free to develop D-STAR equipment, but at the time of this writing Icom and Kenwood are the only companies that have done so for the US amateur market. Neither company "owns" the original D-STAR protocol.

7.1 What is D-STAR?

The primary application of D-STAR is digital voice, but the system can handle any sort of data — including text, voice or images.

As shown in **Figure 19**, a D-STAR network

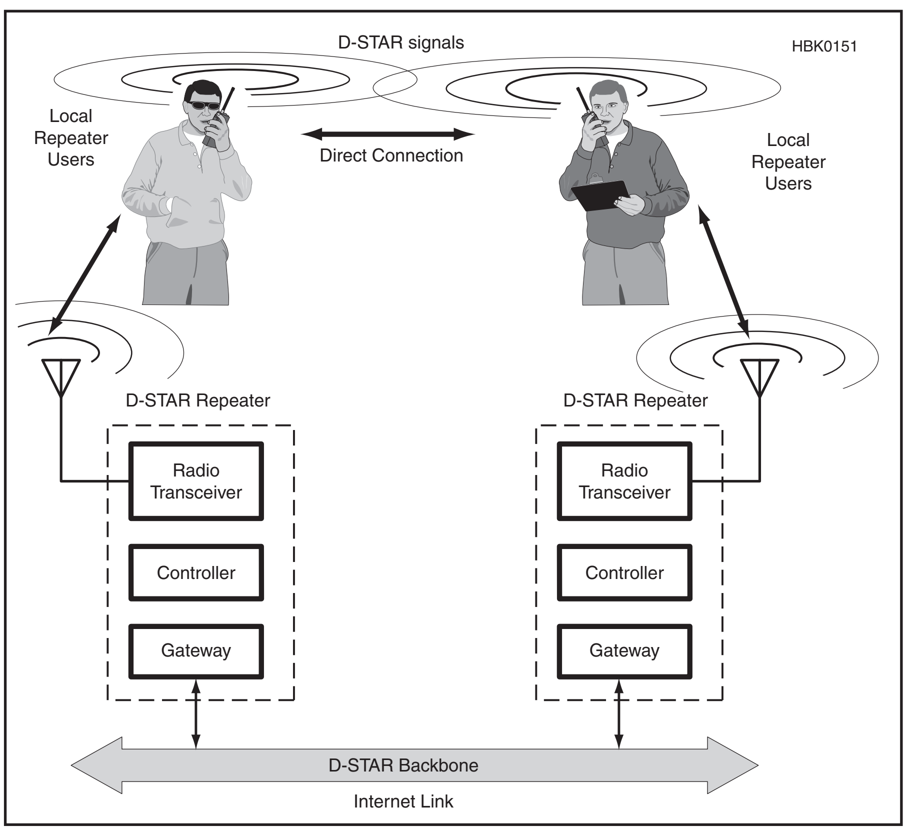


Figure 19 — A D-STAR network can take several forms. D-STAR compatible transceivers can communicate directly (simplex) or through a D-STAR repeater for wide coverage.

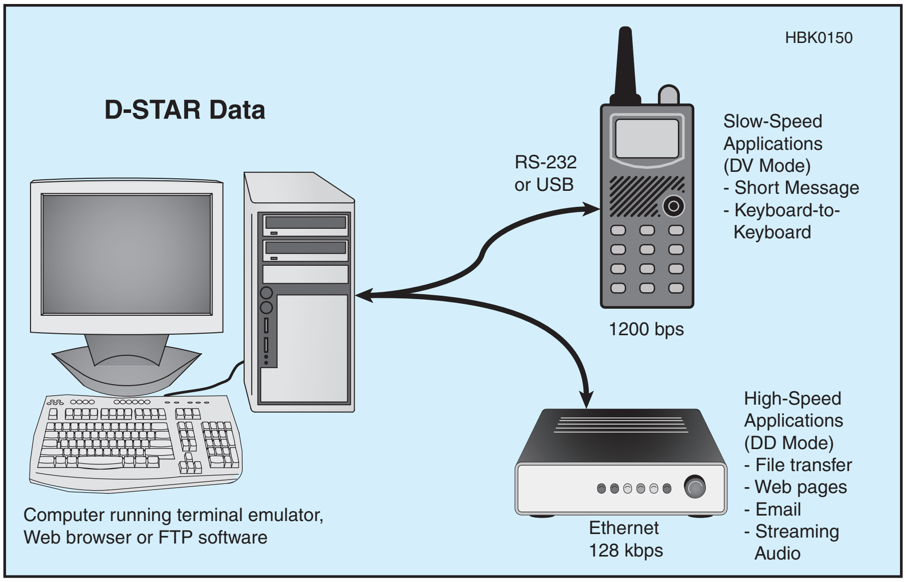


Figure 20 — Radios that support DV voice and data present an RS-232 or USB 2.0 interface to the user.

can take several forms. D-STAR compatible transceivers can communicate directly (simplex), or through a D-STAR repeater for wide coverage. D-STAR signals cannot be repeated through traditional analog repeaters without modification.

The D-STAR system carries digitized voice and digital data, but does the job in two different ways. There is a combined voice-and-data mode (DV) and a high-speed data-only stream (DD). From the perspective of the D-STAR user, data and voice are carried at different rates and managed in different ways, but over the air they are transported as bit streams.

7.2 Digital Voice and Low-Speed Data (DV)

The D-STAR codec digitizes analog voice by using the AMBE 2020 codec. AMBE stands for Advanced Multiple Band Encoding and 2020 designates the variation used by D-STAR.

AMBE can digitize voice at several different rates. The D-STAR system uses a 2.4 kbit/s rate that offers a good compromise between intelligibility and the speed at which data must be transmitted via the radio link. In addition, AMBE adds information to the voice data that allows the receiving codec to correct errors introduced during transmission. The net result is that the digitized voice stream carries data at a rate of 3.6 kbit/s.

Interleaved with the digitized voice information, D-STAR's DV mode also carries 8-bit digital data at 1200 bit/s. This data is used for synchronization in the D-STAR protocol with the remaining bandwidth available for use by the radio manufacturer. Icom uses the remaining bandwidth to carry repeats of the D-STAR RF header, the 20-character front-panel message, and serial data as described below.

Radios that support DV voice and data provide an RS-232 or USB 2.0 interface to the user as shown in **Figure 20**. (The RS-232 interface is restricted to RxD, TxD and ground — "three-wire" connection.) Any computer terminal or program that can exchange data over those types of interfaces can use D-STAR's DV mode capabilities as a "radio cable."

Because D-STAR's DV mode handles the data stream in an unmodified "raw" format, it is up to the equipment or programs that are exchanging data to manage its flow.

7.3 High-Speed Data (DD)

D-STAR's high-speed data mode is called *D-STAR DD*. This mode does not digitize the analog voice signal. The data-only packets sent over the RF link at a raw data rate of 128 kbit/s half-duplex, but since that includes the packet header and the delay between packets, the *net data rate* is somewhat lower. As with the DV mode, data is transmitted without modification, so flow control is left to the applications on each end. Radios supporting DD mode communications may also support DV mode.

Users connect to a radio supporting DD mode with an Ethernet interface via the usual RJ-45 modular jack found on computer networking equipment. The DD mode interface looks to computer equipment just like a customary IP network connection. Specifically, the DD mode interface is an Ethernet "bridge." This allows web browsers and other internet software to run normally, as if they were connected to standard computer network.

The net data rate of DD mode is comparable to or better than a high-speed dial-up internet connection. Any streaming media mode that will run over dial-up internet will likely perform well over D-STAR.

With its high signaling rate and 130-kHz bandwidth, FCC regulations restrict D-STAR DD operating to the 902 MHz and higher bands.

7.4 D-STAR Radios

At the present time, Kenwood offers a single D-STAR-compatible handheld transceiver, the model TH-D74A, for 2 meters and 70 centimeters.

Icom D-STAR radios are available in several models as mobile/base transceivers and as handheld transceivers. Except for the ID-1 base/mobile transceiver, which operates at 1.2 GHz, Icom D-STAR transceivers are designed to operate at 2 m, 70 cm or both. Icom D-STAR transceivers also support both digital and analog operation, which means that they can also be used for traditional analog FM communication as needed. An example is shown in **Figure 21**.

To determine the most appropriate D-STAR radio for an application, the first step is to understand the requirements:

- Is DD mode operation required (high-speed data)? If so, the only radio supporting DD mode is the ID-1.
- What bands are required? That will depend on activity in your area. If data is to be transmitted while in motion, higher frequencies will result in fewer transmission errors, improving the net data exchange rate. (Note: DD operation is only permitted on 902 MHz and higher.)

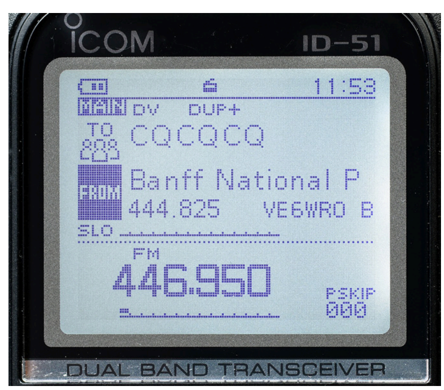


Figure 21 — The Icom ID-51A is a 2 meter/70-cm handheld with D-STAR functionality.

- Is high-power required? Using higher power base/mobile radios will result in stronger signal strengths and fewer data transmission errors.
- If data is to be transmitted, what data interface does the computer have?
- Error correction for low-speed data (DV mode) is the responsibility of the data communications programs used to exchange data. Make sure to carefully evaluate the software you intend to use.

More information about D-STAR may be found in the **Digital Protocols and Modes** and **Repeaters** chapters of the *ARRL Handbook*.

8 DMR: Digital Mobile Radio

Digital Mobile Radio (DMR) is not new technology as such, but what is relatively new is its adoption by the amateur radio community. DMR's origins are in commercial and public service applications such as police, fire, and so on. Amateurs began re-purposing 2 meter and 70-centimeter DMR transceivers by simply reprogramming them for ham frequencies and they did the same with commercial DMR repeaters. Many DMR repeaters are linked to the internet, creating a global network. The result has been substantial growth in DMR activity during the last several years. In addition, at the time of this writing, several manufacturers were marketing DMR transceivers to the amateur radio market in recognition of its increasing popularity.

More information about DMR may be found in the **Digital Modes and Protocols** and **Repeaters** chapters of the *ARRL Handbook*. A comprehensive introduction by W2XAB to the DMR protocol, operation using DMR, and the equipment needed for DMR is available at **www.raqi.ca/~ve2rae/dmr/Amateur_Radio_Guide_to_DMR.pdf**.

8.1 DMR Advantages

Digital Mobile Radio is spectrally efficient compared to other VHF/UHF digital modes. A DMR repeater can support two simultaneous digital conversations within a 12.5 kHz bandwidth. The repeater achieves this by using TDMA – Time Domain Multiple Access – to rapidly switch between data channels. A DMR repeater has two available time slots, which means that two individuals can effectively use the repeater at the same time and on the same frequency. The switching is so rapid that the signals are repeated seamlessly and with little interruption. If both timeslots happen to be in use, users hear a "busy" beep when they attempt to transmit.

Another advantage is a DMR repeater's ability to support analog FM transmissions in addition to digital. This factor has given an additional boost to DMR adoption.

8.2 DMR TalkGroups

A DMR repeater will typically offer several TalkGroups and these are simply "chat rooms" that cover a state or region. Thanks to internet linking, TalkGroups can also span the globe. Different DMR repeaters link to different TalkGroups, so it helps to obtain this information from the hams who use your local repeaters.

8.3 CodePlugs and Programming

Unlike an analog FM repeater that just requires the frequency pair and possibly a CTCSS tone, a DMR repeater requires the user's transceiver to transmit the call sign and "Color Code" of the desired repeater. DMR transceivers must be programmed with this information using a data file known as a CodePlug. Fortunately, CodePlugs and programming software are easily obtainable online, often free of charge. The best approach is to contact local DMR users for assistance. They'll know where to find CodePlugs not only for DMR repeaters in your area, but also throughout the region in which you reside.

9 System Fusion

System Fusion is a proprietary C4FM-based digital system developed by Yaesu. The use of the word "fusion" is a reference to the fact that Fusion hardware is capable of detecting analog or C4FM digital signals and automatically switching from one mode to the other as necessary, effectively "fusing" analog and digital functionality. The analog/digital detection and switching is completely transparent to the operator and does not require operator action of any kind.

Thanks to the automatic switching function, a System Fusion repeater can replace an existing analog FM repeater with little impact on analog FM users. A Fusion repeater will repeat all analog signals, regardless of the type of transceiver that generates them. In fact, FM users will not notice a difference except when the repeater is relaying a digital signal from a System Fusion transceiver. In that instance, analog users will hear a buzzing sound.

Unlike DMR or D-STAR transceivers, Yaesu System Fusion transceivers do not require elaborate programming beyond entering one's call sign when the transceiver is powered on for the first time. Using a Fusion transceiver with a Fusion repeater is as simple as squeezing the Push-To-Talk button. The "analog side" of a System Fusion repeater may require CTCSS tones in some instances and, of course, these would have to be programmed in the transceiver for analog FM use.

There are a variety of Yaesu System Fusion transceivers and their capabilities differ from model to model, but all can switch from analog to digital automatically. A basic Fusion radio supports digital and analog FM, but usually includes higher and lower quality digital voice modes. (The higher quality mode consumes somewhat more bandwidth and may be more vulnerable to "dropouts" under margin signal conditions.) More expensive transceivers add the ability to send and receive images, among other data. They also provide access to the global Yaesu WIRES-X network through repeaters that linked to the internet.

More information about System Fusion may be found in the **Digital Modes and Protocols** and **Repeaters** chapters of the *ARRL Handbook.*

10 HF Digital Voice

While D-STAR is the dominant digital voice/data system on the VHF+ bands, two very different systems can be found in operation on the HF bands. One uses dedicated hardware while the other two rely on sound devices and software. All require HF SSB transceivers, although they could just as easily be used on VHF SSB (or even FM) as well.

10.1 AOR and AMBE

The AOR Corporation was the first ham manufacturer to arrive on the scene with an HF digital voice and data "modem" in 2004. The fundamental design is based on a Vocoder protocol created by Charles Brain, G4GUO. His protocol involves the use of Advanced Multi-Band Excitation, better known as *AMBE,* a proprietary speech coding standard developed by Digital Voice Systems. Brain's protocol operates at 2,400 bit/s, with Forward Error Correction added to effectively produce a 3,600 bit/s data stream. This data stream is then transmitted on 36 carriers, spaced 62.5 Hz apart, at 2 bits/symbol, 50 symbols/s using QPSK. This gives the protocol a total RF bandwidth of approximately 2250 Hz (compared to 2700 – 3000 Hz for an analog single sideband transmission).

The AOR units, such as the ARD9800 (**Figure 22**), are designed to be as "plug and play" as possible. You simply plug your microphone into the front-panel jack and then plug the modem into the microphone input of your transceiver. A front panel switch selects digital encoding or analog transmission, so the unit can be kept in the line when not used for digital conversations. On the receive side, the modem automatically detects the sync signal of an AMBE transmission and switches to digital mode automatically. In addition to voice, the AOR modems can send digital data (typically images).

AOR ON THE AIR

AOR digital voice is clear and quiet, an unusual thing to hear on an HF frequency. Tune around 14.236 MHz and you're likely to hear AOR AMBE signals. They sound like rough hissing noises.

During *QST* Product Review testing in 2004, the ARRL Lab discovered that decoding was solid down to about 10 dB S/N. Digital voice is an all-or-nothing proposition, however, and below that level the signals begin to break up. The result is absolute silence during those periods. Interference to the received signal produced a similar outcome.

The common approach on the air is to begin the conversation in analog SSB, then switch to digital. Each transmission starts with a one-second sync burst after the push-to-talk button is pressed. If the other station misses the sync signal, the audio doesn't decode and the transmission sounds like analog white noise. This means that for successful communication you must be on the correct frequency and ready to receive at the beginning of the transmission. However, if there is fading or interference during the sync transmission, you may be unable to decode the signal that follows.

A few additional steps will improve odds of success with AOR modems.

- Make sure that both stations are on the same frequency, within about 100 Hz.
- Set IF receive filters to 3 kHz or wider.
- Don't overdrive the modem audio input.
- Turn off transceiver speech compression.
- Don't overdrive the radio. If the ALC meter shows any activity, turn down the output from the modem.

Transceiver duty cycle will be substantially higher when you're operating digital voice compared to normal SSB. To avoid damaging your radio, it is a good idea to reduce your output by 25 to 50%.

10.2 FreeDV

FreeDV is an open-source software-based HF digital voice system for Windows, Linux, or OS X (**Figure 23**). It is designed to use a computer sound device or on-board sound chipset to send and receive amateur digital voice. Activity is primarily on HF at 14.236 MHz. In addition to the software, you need a

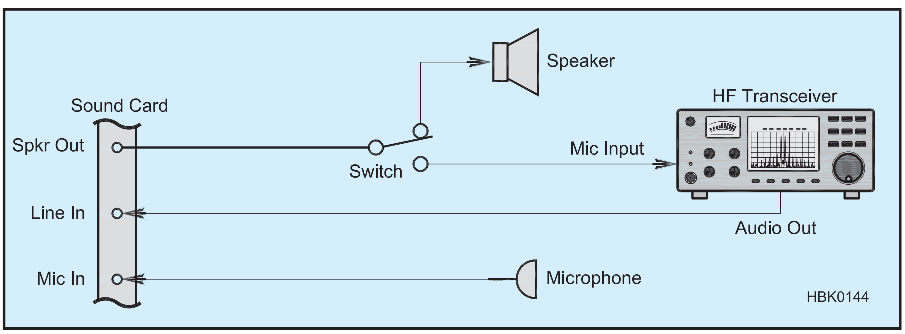


Figure 24 — The single sound device approach to using *FreeDV* requires the means to switch the audio stream when switching from transmit to receive.

Figure 22 — The AOR ARD-9800 digital voice modem.

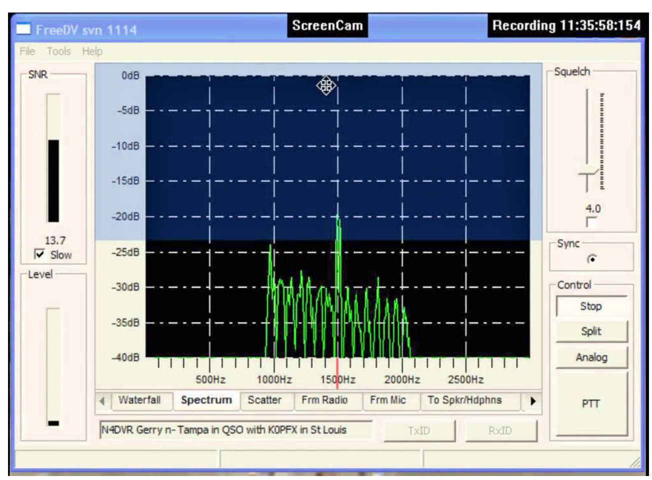


Figure 23 — *FreeDV* digital voice software. [N9SJA photo]

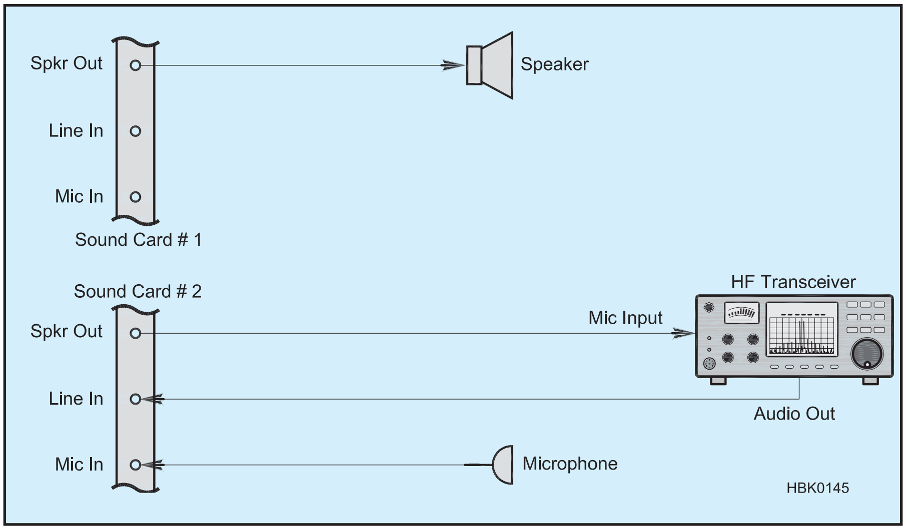


Figure 25 — The most elegant, easy-to-operate solution for *FreeDV* is to use two sound devices — one to process the received signal and the other to generate the transmit signal.

radio, computer and one of the sound-device interfaces discussed earlier in this article. Start by downloading and installing the *FreeDV* software at **freedv.org**. On this same website you will find documentation that explains installation and operation.

To receive, all you need is a cable between the audio output of your radio and the LINE INPUT of your sound device. You may need to get into your sound device control software and boost the LINE INPUT gain. If the signal in the waterfall display is too bright, reduce the gain control.

Transmitting with *FreeDV* is somewhat more complicated in terms of your station setup, especially if you have only one sound device. (**Figure 24**) The audio output from your sound device must be applied to your headset or PC speakers for receiving, but the *same* output must also feed your transceiver for transmitting, so a switch for the audio output is needed. The easier, more elegant approach is to use two separate sound devices. (**Figure 25**) The second sound device doesn't need to be high-end; you can use an inexpensive USB sound device for this application.

An alternative to the software and computer approach is available with the Rowetel SM1000 FreeDV adapter. The SM1000 integrates the entire FreeDV system within a handheld enclosure, allowing you to enjoy digital voice without a computer. At the time of this writing, the SM1000 was available for $201 US. See **www.rowetel.com** for more information.

11 EchoLink, IRLP, and WIRES-X

Worldwide communication on HF requires reliable propagation and more substantial radio and antenna requirements than a basic VHF FM setup. By using internet links instead of HF radio links, a ham with a modest radio (even a handheld transceiver) can reliably communicate with stations hundreds or even thousands of miles away at any time of the day or night (**Figure 26**).

The most common form of amateur internet linking involves the exchange of audio using *VoIP* — Voice over Internet Protocol — technology. This is the same technology used by internet telephone services, and by online voice "chat" applications such as *Skype*, *TeamSpeak* and others. Three versions of amateur radio voice linking have become popular in the US: EchoLink, IRLP and WIRES-X.

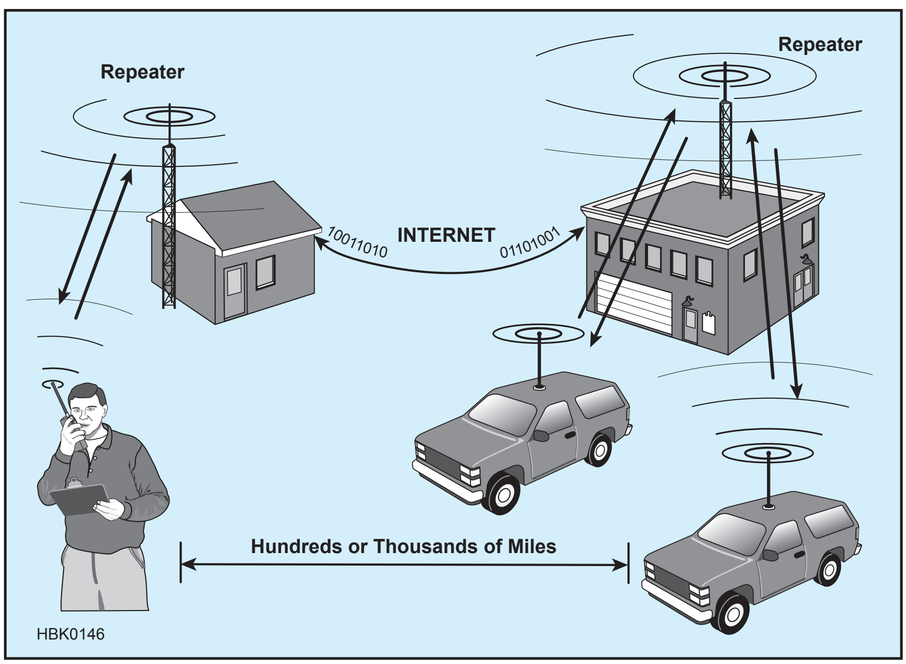


Figure 26 — The internet VoIP link allows repeater users to speak with other amateurs who may be transmitting through other repeaters.

11.1 EchoLink

EchoLink software, developed by Jonathan Taylor, K1RFD, is designed for Windows PCs, but there is a Mac version as well. With the software installed on a sound-device-equipped computer, any ham can create an EchoLink *node* that others can access by radio. Hams can also join the network directly without using radios. They simply plug microphone/headsets into their computers.

Each EchoLink node is assigned a number that can be up to six digits in length. RF users access the EchoLink network by first sending the DTMF code required to activate the link on a repeater or simplex node. (**Figure 27**) Of course, this requires a transceiver equipped with a DTMF keypad. Once the link is up, they use the keypad again to transmit the number of the EchoLink repeater, link or user they wish to contact.

Hams who access the network directly (without a radio) select the person or system they wish to contact by simply clicking on its name in the software listing (**Figure 28**). The EchoLink network maintains continuously updated databases on several servers that indicate node and user locations and operational status. The network servers also support *conferencing* where many users can "meet" and speak together, either by direct access or via radio.

EchoLink supports communication among three different source groups:

• *Repeaters*: This is typically a VHF/UHF FM repeater with a computer at the site and a VoIP connection to the internet via EchoLink. When the EchoLink function is switched on, audio from the repeater is available on the EchoLink network, and vice versa. The internet VoIP link allows repeater users to speak with other amateurs who may be transmitting through other repeaters, or who may be linking directly.

• *Links*: Similar to repeater nodes, but operating on simplex frequencies without full repeater functionality. These are typically FM transceivers connected to computers. Depending on the station setup, their coverage can be confined to a neighborhood, or it may encompass an entire county.

• *Users*: This is strictly a computer-to-computer VoIP connection with no RF involved. In other words, the operator is sitting in front of his computer wearing a microphone/headset.

All the interfacing on the "RF side" of EchoLink is handled by connections to the computer sound device and serial port with a sound-device interface used to switch the radio between transmit and receive — the same setup used by sound device based digital communications such as PSK

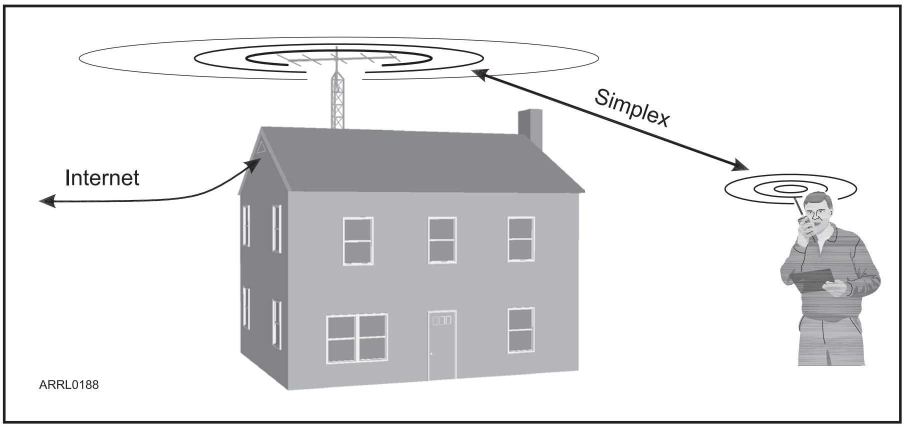


Figure 27 — A typical EchoLink simplex node.

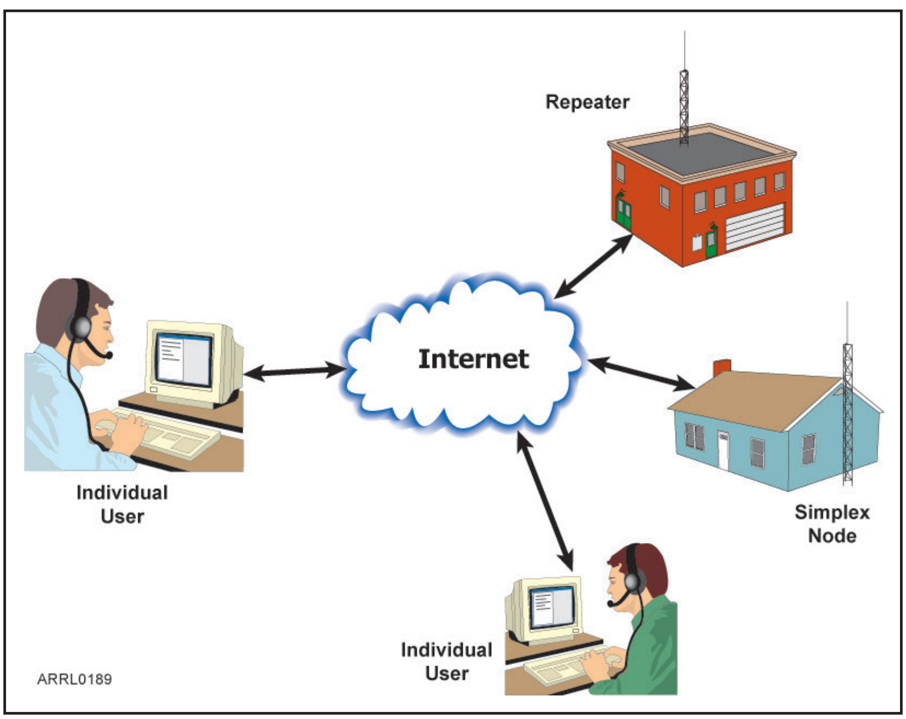


Figure 28 — For some EchoLink users, there is no radio involved (at least at their end of the conversation).

ECHOLINK SECURITY

Before being granted access, every ham who establishes an EchoLink node, or who connects to the network directly, must provide positive proof of identity and license during the EchoLink registration process. After having been validated, each EchoLink node or user must provide a password, along with his or her call sign, to log in. Each time a connection is made for a QSO, the EchoLink servers verify both the sender and the receiver before communication can begin. Hams who use their radios to connect to the EchoLink network through a repeater or a simplex node do *not* have to be validated or provide a password. This is only necessary for the repeater or link stations, or for those who connect directly.

It is possible to configure EchoLink to accept connections only from certain types of stations: repeaters, links, users or all three. You can also set up a list of any number of "banned" call signs, which will not be allowed access. In addition, you can block or accept connections according to their international call sign prefix, to comply with reciprocal control-operator privileges or third-party traffic restrictions.

In Sysop mode, by default, EchoLink announces each station by call sign when the station connects. (The user can program a voice or CW ID that is generated automatically when needed.) The EchoLink software automatically generates detailed logs and (optionally) digital recordings of all activity on the link.

Unlike software such as e-mail programs, file-sharing programs and web browsers, EchoLink does not have any way to pass files or "attachments" that might harm your computer. There are no known cases of EchoLink accepting or spreading a computer virus. Of course, any PC connected to the internet should always be protected by some sort of internet security hardware or software.

11.2 IRLP

IRLP stands for Internet Radio Linking Project. Like EchoLink, IRLP uses the internet to establish VoIP links. The difference is that IRLP only permits access via RF nodes at repeaters or on simplex frequencies (all IRLP nodes are interlinked via a central internet server). You must use a radio to access the IRLP network.

A typical IRLP node consists of a transceiver that is connected to the internet through dedicated IRLP hardware and software. This requires a Linux computer running IRLP software and an IRLP interface board, or a turnkey "embedded" IRLP unit that combines the Linux computer, IRLP software and interface in a single package. The IRLP software controls the VoIP audio stream using carrier operated squelch (COS) or continuous tone coded subaudible squelch signals (CTCSS) from the transceiver. When COS is present, the computer detects it through the IRLP interface board.

The operator connects to the IRLP network (at a repeater, for instance) using DTMF signals sent over the radio. As with EchoLink, this means that the operator must have a radio with a DTMF keypad. The actual access code is determined by the owner of the node. The user sends the correct DTMF sequence to bring up the IRLP connection, then sends the sequence of the distant node he wishes to contact. That might be a repeater in another state or a distant simplex node. You can view a list of current IRLP nodes at **status.irlp.net/**.

Here is an example of a typical IRLP connection:

- Press the transceiver PTT and identify your station. For example, "This is WB8IMY, Steve in Wallingford, Connecticut, USA connecting to node 8880."
- Next, press *#8880* (node 8880 in this example) on the DTMF keypad, release PTT and wait for the node connect announcement. The announcement indicates that your audio is now reaching the distant node. You can call a specific person, or just ask if anyone is around to chat.
- When finished, key PTT, make an announcement such as "WB8IMY releasing node 8880" and then press *#73* on the DTMF keypad to disconnect. The IRLP node will announce the disconnect and the node may now connect to a different node, reflector, or it will stand by awaiting another user.

It is interesting to note that there is a project underway (known as *EchoIRLP*) to bridge the IRLP and EchoLink networks.

11.3 WIRES-X

WIRES-X — Wide-Coverage Internet Repeater Enhancement System — is a VoIP network created by Vertex Standard (Yaesu). It is functionally like IRLP (including management via a central internet server), but nodes and repeaters are connected via a Vertex Standard HRI-100 interface and a computer running under the Windows operating system. Vertex Standard transceivers are *not* required for WIRES-X communication, but as with IRLP, all access must be via RF. WIRES-X also supports the Yaesu System Fusion network of repeaters for automatic routing and many other features.

There are two WIRES-X operational configurations using home stations (simplex nodes), repeaters or a combination of the two:

SRG — Sister Radio Group. You operate within WIRES-X in a small (10 node maximum) network that is ideal for closed-group operations. Within the network, all nodes operate using the same repeater node list, so you can link only to stations within this 10-node network. Because there are only ten nodes maximum, access to any of these nodes is possible using a single DTMF tone when calling. At the beginning of each transmission, this single DTMF tone locks communication between the calling node and the called node, but local (non-linked) transmissions are also possible, simply by omitting the DTMF tone at the beginning of the transmission.

FRG — Friends Radio Group. You may call any repeater registered with the WIRES-X FRG server. In the case of FRG operation, a six-digit DTMF code is required for access, and once the link is established this code need not be sent again (this is called the LOCK mode), unless the operator wants the ability to make non-linked transmissions (UNLOCK mode), in which case the six-digit code must be sent at the beginning of each transmission (using the DTMF Autodial feature of the transceiver, for example). Group calling to preset 10-repeater B, C, and D lists is also possible.

SRG is similar in philosophy to the local FM voice repeater where you tend to talk to the same group on a regular basis. FRG is similar in concept to IRLP, linking together a worldwide group of repeaters and base stations.

12 FT8 and FT4

When you hear someone referring to the "*WSJT* modes," they are talking about the digital operating modes included in the *WSJT* software suite created by Dr. Joe Taylor, K1JT. *WSJT* — now known as *WSJT-X* and supported by a team of protocol designers and programmers — offers several different modes, each with its own strengths for a given application.

WSJT had its beginnings as a collection of software for moonbounce communication. As a Nobel Prize-winning scientist who studies pulsars and other distant astronomical objects, Joe has a keen interest in weak signals, the kind hams encounter when reflecting signals from the lunar surface. Joe's software exploits the power of modern desktop and laptop computers to separate weak signals from noise and decode the information they contain. With just a sound card or sound chipset and a transceiver interface, *WSJT* made it possible for hams with modest stations to enjoy VHF meteor scatter communication and even moonbounce. Prior to Joe's invention, hams needed to make large investments in antennas, transceivers, and amplifiers to pursue these activities. The advent of *WSJT* opened an exciting world of operating for many who could not otherwise afford it.

But it wasn't long before someone wondered what would happen if one of the *WSJT* modes, known as JT65, was used on the HF bands. Digital communication on HF isn't nearly as challenging as getting a signal to the moon and back, so it stood to reason that there would be plenty of "performance margin" to provide fascinating results. To no one's surprise, this turned out to be true. Using a variant of JT65 known as JT65A, even a few watts of JT65-modulated RF to a wire dipole antenna resulted in transcontinental and even global communication.

JT65 became a popular HF digital operating mode in short order and, by 2016, it even eclipsed PSK31. Then, in 2017, a version of *WSJT-X* was released that included a new mode: FT8.

FT8 took off like a proverbial rocket. It offered many of the benefits of JT65 but was a much faster and somewhat semiautomated mode. By the end of 2017, FT8 had become the most widely used digital mode on the HF bands.

Like its cousin, JT65, FT8 is not a "conversational" mode. In fact, there isn't a conversation taking place at all. Instead, all you are doing is making contact and exchanging call

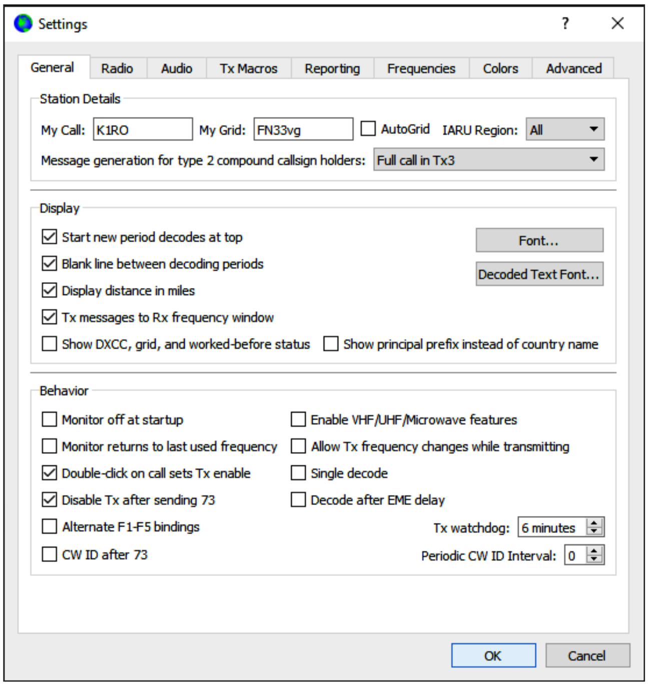


Figure 29 — The *WSJT-X* settings screen. The tabs along the top allow you to set up your operating preferences, customize the screen appearance, and interface the software with your transceiver.

signs, signal reports, and grid square locations with acknowledgements when the information is received. So, what is the attraction of modes like FT8?

- They provide "honest," objective signal reports. The software automatically makes a signal-to-noise (SNR) measurement and generates a report for the other operator. Signal strength is expressed in dB (decibels). For example, a –25 dB signal is quite weak while a +4 dB signal is very strong. These reports give you a true sense of how well your signal is being received. This is critical when testing antennas or studying propagation.
- The performance of FT8 is such that even a minimal station, such as someone transmitting 5 W to an attic antenna, can make large numbers of contacts, including contacts with stations in other countries.
- Their exchanges provide all the information needed for valid contacts that can be applied to awards such as the ARRL DX Century Club (DXCC).

And for FT8, another attraction is speed. It is possible to complete an FT8 contact in little more than 90 seconds.

12.1 Getting Started with FT8

To get started with FT8, you must first download and install *WSJT-X*. While most hams use the Microsoft Windows operating system, you will also find versions of *WSJT-X* for macOS and Linux at **physics.princeton.edu/pulsar/k1jt/wsjtx.html**. Be sure to use Version 2.0 or higher —earlier 1.x versions are not compatible with the new version, and you won't be able to work anyone.

When you start *WSJT-X*, you'll see two windows: a main window with menus for the settings, options, and the information being decoded, and a separate window for a combined waterfall and spectrum display. Each window can be moved on your monitor and resized to suit your preference and the monitor's resolution. More on these two windows later.

First, before you can use *WSJT-X*, you need to set up the software. At the upper left, click on the **FILE** menu option and choose **SETTINGS** (or just press the F2 keyboard shortcut) to bring up the screen shown in **Figure 29**.

In the **STATION DETAILS** section, you will need to enter **MY CALL:** and **MY GRID:** for your station. If you don't know your grid square, Dave Levine, K2DSL, has a handy calculator at **www.levinecentral.com/ham/grid_square.php**.

Under **DISPLAY**, check the first two options to help make it easier to follow decoded text when you are on the air. Check the third box to show distance in miles (unchecked for kilometers). Check the fourth box to show your transmitted messages in the receive frequency window. (That way, when you are in a QSO, you can see alternating lines showing what you sent and received. More on this later.) An important feature in the **DISPLAY** section is the **FONT** and **DECODED TEXT FONT** buttons, which you can click on to match the size and type of the font you prefer.

Under **BEHAVIOR** check the third and fourth options. Check **DOUBLE CLICK ON CALL SETS TX ENABLE** so you do not have to constantly hit the **ENABLE TX** button whenever you want to transmit. The instructions in the rest of this section will be written under the assumption that you've checked the **DOUBLE CLICK ON CALL SETS TX ENABLE** box. Also check **DISABLE TX AFTER SENDING 73**, which is a safety valve of sorts, preventing your continual transmissions even after your contact has completed with a 73 message. You can read about the other options in *WSJT-X* documentation, but these are the most important ones.

Under the **RADIO** tab, you can pick the method of switching your radio to the transmit mode. You'll find it in the **PTT METHOD** box. If you pick **CAT** control, you must choose the correct communication protocol for your transceiver (by using the drop-down **RIG** menu at the top), the serial (COM) port number, the COM port data rate, and the parity values for your radio in the **CAT CONTROL** box at the immediate left. Note that if you activate CAT without your radio turned on, or if *WSJT-X* is unable to communicate with your radio because of an incorrect COM port or speed setting, the program will generate an error message and you won't be able to continue.

Choose VOX (voice operated switch) in the **PTT METHOD** box if you are using an interface with VOX switching that responds to audio from your computer, or if you intend to allow your transceiver to switch from receive to transmit with its own VOX circuit.

Some transceivers will do transmit/receive switching (PTT) through the CAT connection. If your radio is one of these, select **CAT** as your PTT option. Otherwise, select either **RTS** or **DTR** and a COM port number for your keying device. In most cases this will be one of the interfaces we discussed earlier in this book. There is a **TEST PTT** button and **TEST CAT** button that you can use to be sure your PTT and/or CAT circuit is working.

CAT control is not necessary to operate *WSJT-X* in any mode, but it is awfully convenient. For example, when you use the drop-down menu in the main window to select your operating frequency, your transceiver will switch to that frequency automatically.

The next step is to select the **AUDIO** tab and choose the **SOUND CARD** device, both **INPUT** (receive) and **OUTPUT** (transmit). You may want to select **MONO** for receive audio and **BOTH** (left and right channels) for transmit. You can pick whatever works for your radio and computer, but be careful; some of the earlier Kenwood HF SSB radios, for example, have the left and right channel audio reversed on the USB pins, so you may have to pick **BOTH** for the output audio.

You also have the option to select your

LOGGING QSO choices and to report the stations you copy to the PSKReporter web page (**www.pskreporter.info/pskmap**). This is a valuable tool that keeps track of digital mode signals received on all bands, in a variety of modes. If you have an internet connection at your station computer, be sure to check the **ENABLE PSKREPORTER SPOTTING** box.

When you exit the setup mentions and return to the main window, pick **FT8** in the **MODE** menu and pick the **DECODE** speed. Selecting **FAST** decode will work well for most computers. If you are blessed with a very fast machine, you can select **NORMAL** or **DEEP**, which will allow you to decode weaker signals. However, this takes more time and could delay the speed at which the results appear on your monitor.

12.2 Navigating the Windows

When you start *WSJT-X*, you'll see two windows: a main window with menus for the settings, options, and the information being decoded, and a separate window for the combined waterfall and spectrum display. Each window can be moved on your monitor and resized to suit your preference and the monitor's resolution.

THE WATERFALL/SPECTRUM DISPLAY WINDOW

Figure 30 shows the *WSJT-X* waterfall/spectrum window as recorded with the VFO set for 14.074 MHz (upper sideband). The top 80% of the display is a waterfall that shows received signals, while the bottom 20% is called the spectrum window and shows the strength of signals in the passband.

The numbers at the top of the waterfall represent the audio frequencies (pitch) from 200 Hz to 2500 Hz, with the receiver set for an SSB-width bandpass filter. (This is just a portion of the passband. It can be set to 3000 Hz or even wider depending on your transceiver filtering.) The vertical axis in the waterfall represents time in 15 second intervals. (FT8 contacts follow a carefully times sequence of 15 seconds transmitting, 15 seconds listening.)

The vertical axis in the spectrum window represents the signal strength (dB). There are options to adjust the waterfall speed, frequency span, start frequency, averaging pattern and other parameters, but you can leave these at their default settings for now.

The signal strength of each FT8 signal can be estimated by its peak amplitude relative to the average noise floor. In the waterfall, the strongest signals show up as a solid orange. Two FT8 signals will often overlap somewhat in frequency, but the software does an excellent job separating the individual messages.

MAIN WINDOW OVERVIEW

Figure 31 shows the activity in the afternoon on 20 meters (14.074 MHz) at K1RO. The **BAND ACTIVITY** window on the left shows stations copied. Note that the band was busy, with a mix of stations in North America, Europe, and Africa represented.

The gray bar with the **20M** label signals the beginning of the most recent 15-second listening period. Below that are lines showing each station copied. From left to right, each line shows time in UTC (hour, minute, seconds — for example, 204300). Next is received signal strength (SNR in **DB**), and then **DT**, which is any difference (in seconds) in time between the station you are receiving and your own computer's clock. The **FREQ** column is the received station's audio frequency within the passband in Hz. The **MESSAGE** column shows what was sent in that sequence. In this case, it's a mix of CQs, stations calling other stations, and stations sending signal reports or confirmations to each other. Check the **CQ ONLY** box just below the window to show only received stations who are calling CQ.

The **RX FREQUENCY** window shows activity on or very near the frequency selected by your receive cursor. The top five lines show three transmissions from Z81D, and two transmissions from EA5BFA who is calling Z81D on that frequency. The last line, which is highlighted, is K1RO calling Z81D on a frequency higher in the passband. It shows up here because **TX MESSAGES TO RX FREQUENCY WINDOW** is checked in the **SETTINGS**.

Each line represents a 15 second time interval, and the line are color-coded to help keep track of what is happening. Color coding is adjustable in the settings. For example, in this case green represents another station calling CQ, yellow represents your transmissions, and red represents other stations' transmissions to you.

In the row of gray buttons below the **BAND ACTIVITY** and **RX FREQUENCY** windows, the *WSJT-X* **MONITOR** button must be engaged to receive and the **ENABLE TX** button must be engaged to transmit. **HALT TX** instantly stops transmission, and **TUNE** sends a solid carrier to check the transmitter and set power levels. **DECODE** forces a decode of the most recent time segment, **ERASE** clears one or both windows, and **LOG QSO** brings up a window

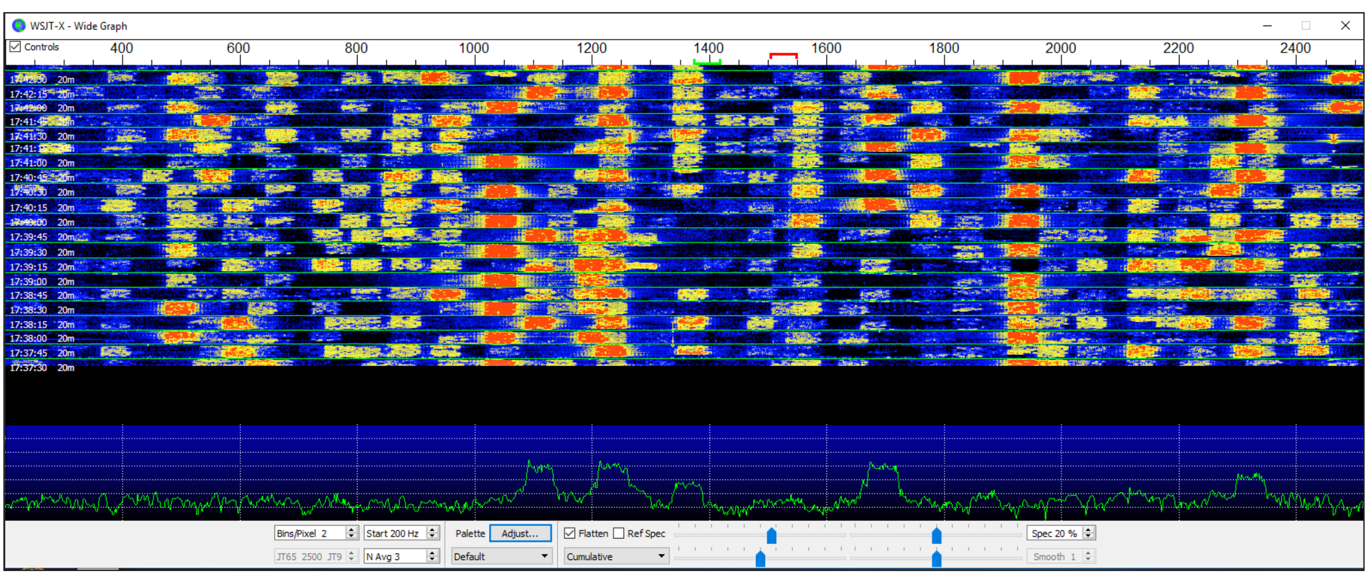


Figure 30 — The *WSJT-X* waterfall/spectrum window as recorded with the VFO set for 14.074 MHz (upper sideband). The numbers above the waterfall represent the audio frequencies (pitch) from 200 Hz to 2500 Hz. The horizontal lines represent 15 second intervals. Each fuzzy "square" represents an FT8 transmission (the stronger the signal, the brighter the "square").

to log the contact.

The messages that K1RO will send during a contact are shown in the **GENERATE STD MSGS** section in the lower right and are generated automatically. In this case, when K1RO clicked on a Z81D decode in the **BAND ACTIVITY** window, the software generated the messages shown. The messages can be sent manually by clicking on the button next to the desired one. If you check the **AUTO SEQ** box to the left, messages are sent automatically in the correct sequence once a contact is initiated and the correct response is received from the station you are working. We'll have more detail on making a contact later in this section.

To the far lower left in this window a vertical scale, 0 to 90, showing the total strength of *all* received audio signals (like an S-meter on your radio), which at this moment reads 50 dB. You don't want too much, or too little received audio. The **PWR** slider at the far right is for adjusting your transmit power.

Above the received signal indicator is a drop-down menu for selecting the band manually, and to the right of that is a window showing the radio's current VFO frequency if you have set up *WSJT-X* to read your transceiver's frequency directly. Having the correct frequency in this window is very important for logging contacts or if you send your data to online databases.

Below the frequency window is information about the station you are calling, and below that the current date and time. To the right of the date and time are options to set your transmit and receive audio frequencies within the passband and to change the signal report from the one generated automatically. There's also a checkbox to toggle your transmissions between the odd and even segments (for example, in a given minute, starting at 00 seconds is even, starting at 15 seconds is odd).

The checkbox for **HOLD TX FREQ** is just that — if you call someone, or someone calls you, your receive frequency moves to the spot where the other station is transmitting but your transmit frequency stays where it is. Check the **CALL 1ST** box, and if multiple stations are calling you, the software will automatically respond to the first one decoded.

Remember that the radio's VFO is left set to the FT8 operating "watering hole" frequency (in this example, 14.074 MHz). The radio's displayed carrier frequency is not changed or tuned.

12.3 The Importance of Time

In every *WSJT-X* operating mode, a critical component is time. The software depends on tight time synchronization to "know" when to expect data signals to be present so they can be decoded. Your computer clock doesn't need to match the clocks of other stations exactly, but it must be synced within less than 2 seconds – the closer the better. If your computer clock is too far out of sync, you will not be able to decode signals.

When you look at decoded signals, pay attention to the numbers in the DT column. If you notice that most of the numbers are greater than 1, that's a clue that your clock computer is drifting out of sync with the other stations.

Fortunately, Windows has built-in time calibration that synchronizes automatically with time sources via the Internet. In my experience, however, the synchronization doesn't take place as often as it should and, as a result, your PC's time may drift by a second or so after just a few days.

Fortunately, Windows has built-in time calibration that synchronizes automatically with time sources via the internet. In my experience, however, the synchronization doesn't take place as often as it should and, as a result, your PC's time may drift by a second or so after just a few days. My recommendation is to periodically "force" resynchronizing. In *Windows 10*, for example, open **CONTROL PANEL** and click on **TIME & LANGUAGE**. In the **DATE & TIME** screen, you will see a section labeled **SYNCHRONIZE YOUR CLOCK**. You can see when the time was last synchronized, and force synchronization by clicking on **SYNC NOW**. You don't necessarily need to do this before every operating session, but it doesn't hurt.

A more convenient alternative is the free time-synchronizing application *Dimension 4*, which you can download at **www.thinkman.com/dimension4/**. Install the application and set it up so that it loads and runs constantly in the background whenever you start your

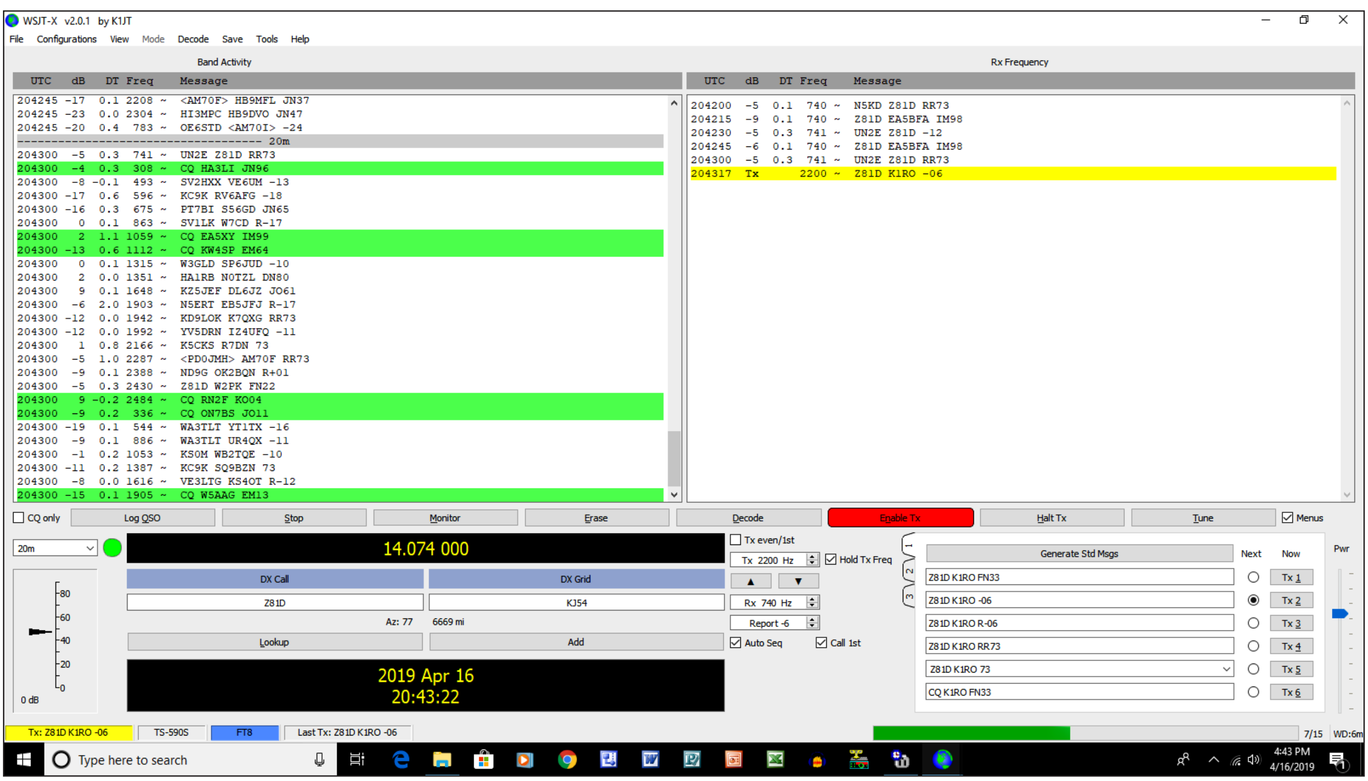


Figure 31 — The *WSJT-X* main screen in the FT8 mode, decoding signals on 20 meters.

computer. Another popular alternative is *Meinberg NTP* (**www.meinbergglobal.com/english/sw/ntp.htm**)

12.4 On the Air with FT8

With the menu items set and saved, the next thing to do is to pick a frequency and listen for activity. You can't miss the sweet slow sounds of FT8 tones once you hear them. They are a much shorter tone duration than the even slower JT65 tones, which are often present 2 kHz higher in the band.

Do *not* choose a narrow passband 500 Hz filter if your radio offers one. Keep your IF passband set for a normal phone conversation, 2 to 3 kHz. This will allow you to see every signal that is scattered throughout the receive passband.

Having said that, there are times when you may want to use a narrower bandwidth. For example, if there is a station in the passband that is extremely strong, that signal may cause your AGC to reduce your receiver gain dramatically, causing weaker signals to disappear. Switching in a filter may help block the strong signal.

Never use speech processing, and never turn on noise blankers, or noise-reduction software. They will distort your transmit signal and receive signals respectively. Also, if your transceiver has an AGC adjustment, set it to "fast."

Set your transceiver for maximum RF output (typically 100 W for many radios) and place it in the USB or USB-Data mode. Start by clicking your mouse on an empty area of the waterfall display. You'll notice that the red/green brackets above the waterfall will move to that position. Next, click on the **TUNE** button in the *WSJT-X* main window. If you've set up everything correctly, *WSJT-X* should key your transceiver and transmit a continuous tone.

Quickly click your mouse on the **PWR** slide control on the right side of the main window and drag it down. Watch your transceiver's RF power output and ALC metering. Keep dragging the **PWR** control down until you see no ALC indication, or until the ALC meter indicates that it is within the minimal zone.

Continue dragging the **PWR** control until your RF output drops to where you feel it should be. You definitely don't need 100 W, but how much is too much? Many FT8 enthusiasts say you should never transmit more than 30 W; others suggest even less. The answer really depends on your antenna system. If you are using a gain antenna such as a Yagi, 30 W or less would be appropriate. On the other hand, if you are using a poor antenna such as an HF mobile whip sitting in your back yard with a couple of radial wires in the grass, you may need more than 50 W to be effective. My recommendation is to start very low and see what sorts of signal reports you receive. If you consistently receive weak reports from most stations, that's a strong hint that you should probably increase power.

After tuning up, you are ready to operate, but before you make your first contact, start by just listening and observing. The first thing you'll notice is that transmissions take only 13 seconds, with a two-second pause between transmissions. This means the length of each transmit or receive cycle is a total of 15 seconds.

As you observe the exchanges, you'll probably see a pattern emerging. FT8 exchanges usually, though not always, follow a strict sequence. It goes like this, starting at 21:02:00 UTC. Note that when there is an exchange between two stations, the transmitting station's call sign is the one on the *right*.

21:02:00 — CQ WB8IMY FN31
WB8IMY sends CQ from grid square FN31.

21:02:15 — WB8IMY N1NAS EN72
N1NAS replies and tells WB8IMY that he is in grid square EN72.

21:02:30 — N1NAS WB8IMY -11
WB8IMY replies with a signal report of -11 dB.

21:02:45 — WB8IMY N1NAS R-15
N1NAS acknowledges the signal report from WB8IMY with an "R" followed by a signal report (-15).

21:03:00 — N1NAS WB8IMY RRR
WB8IMY sends "RRR," which means "Roger, roger, roger." Everything has been received and the exchange is complete.

21:03:15 — WB8IMY N1NAS 73
N1NAS sends 73 – best wishes.

21:03:30 — N1NAS WB8IMY 73
WB8IMY sends his 73 as well. The contact has ended.

The entire contact is complete within just 90 seconds, start to finish. Of course, this assumes that all goes well. If there is noise, fading, or interference that corrupts reception, an exchange may need to be repeated, and that will take longer. Under decent signal conditions, however, most contacts are completed in less than two minutes. **Figure 32** shows a contact between K1RO and K9IG, followed by K1RO working WVØI.

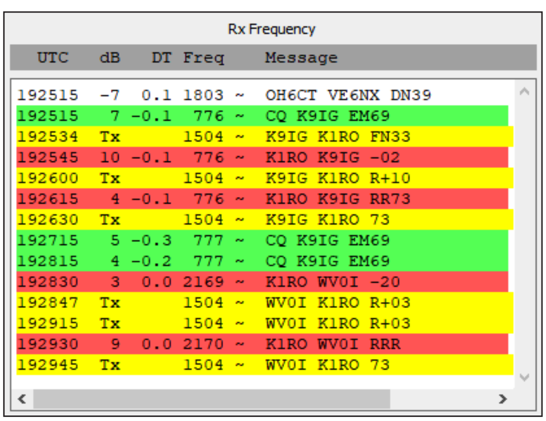
MAKING CONTACT

If you're comfortable monitoring FT8, it is time to try a contact. Watch for someone calling CQ. Depending on how you've set up your colors in *WSJT-X*, a CQ may appear as green-shaded text. Make sure you've checked the auto sequence (**AUTO SEQ**) box.

Keep in mind that FT8 is a bit different from traditional CW or SSB operation. You can call another station anywhere in the audio passband, and split-frequency contacts (different transmit and receive audio frequencies) are very common. It's a good idea to set your transmit frequency to a clear spot and enable **HOLDTX FREQ** to keep your transmissions in that clear spot, rather than trying to call another station where they are transmitting.

When you see a CQ, quickly double-click your mouse on the text. *WSJT-X* should respond by keying your transceiver and sending the other station's call sign, your call sign, and your grid square. Now just sit back and watch; *WSJT-X* will handle the rest.

Automatic operating depends on *WSJT-X* receiving the correct text at the correct times. If it does, it will select the correct response


Rx Frequency

UTC	dB	DT	Freq		Message
192515	-7	0.1	1803	~	OH6CT VE6NX DN39
192515	7	-0.1	776	~	CQ K9IG EM69
192534	Tx		1504	~	K9IG K1RO FN33
192545	10	-0.1	776	~	K1RO K9IG -02
192600	Tx		1504	~	K9IG K1RO R+10
192615	4	-0.1	776	~	K1RO K9IG RR73
192630	Tx		1504	~	K9IG K1RO 73
192715	5	-0.3	777	~	CQ K9IG EM69
192815	4	-0.2	777	~	CQ K9IG EM69
192830	3	0.0	2169	~	K1RO WV0I -20
192847	Tx		1504	~	WV0I K1RO R+03
192915	Tx		1504	~	WV0I K1RO R+03
192930	9	0.0	2170	~	K1RO WV0I RRR
192945	Tx		1504	~	WV0I K1RO 73

Figure 32 — The *WSJT-X* Rx Frequency window showing activity at station K1RO in grid FN33. Starting with the second line: K9IG in grid EM69 calls CQ on 776 Hz. K1RO answers, transmitting on 1504 Hz. K9IG sends K1RO a –02 dB report. K1RO responds with an R (roger) and sends a +10 dB report. K9IG acknowledges that and sends 73 (RR73). K1RO sends the final 73. K9IG calls CQ twice more. Note that K1RO transmitted on a clear frequency, 1504 Hz, rather than switching to K9IG's CQ frequency.

Next, K1RO's receive frequency jumps to 2169 Hz because WVØI has called there and sent a –20 dB report. K1RO, still transmitting on 1504 Hz, acknowledges and sends a +03 report (R+03). K1RO did not hear a response from WVØI and so repeated that transmission. This time, a RRR acknowledgement report is received from WVØI, so K1RO sends 73 to complete the contact.

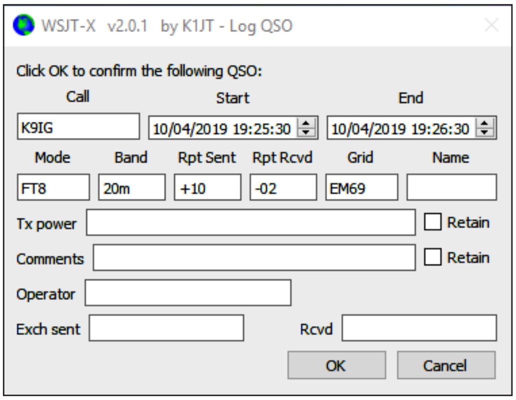


Figure 33 — When K1RO completed the contact with K9IG (ie, reports were sent in both directions and acknowledgements received), the *WSJT-X* logging window popped up to record the contact. *WSJT-X* stores the contact information in both text and ADIF format for transfer to other logging software or online databases such as ARRL's Logbook of the World.

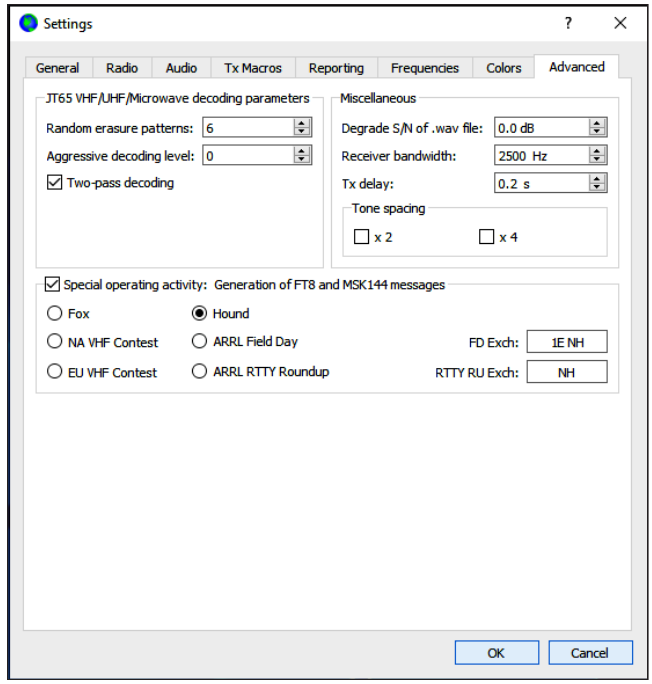


Figure 34 — The Advanced tab in the *WSJT-X* Settings menu is used to set up special operating activities such as Fox/ Hound DXpedition mode, ARRL Field Day, or contests such as ARRL RTTY Roundup.

and send it automatically. You won't need to lift a finger. When the station responds to you, *WSJT-X* should respond in turn.

Sometimes, however, things do not go as planned. The station may send a signal report, but will not receive your confirmation and your report. Perhaps there was interference, or maybe conditions suddenly changed. If so, you may see the station repeating its report of your signal, and *WSJT-X* responding by sending your report of the other station's signal again. The other station's *WSJT-X* is doing this automatically because it didn't decode your text the first time around. Just wait as *WSJT-X* tries again. With luck, the station will receive and decode properly on the next attempt.

Once the contact is complete, usually with a 73 message, you'll notice that the **ENABLE TX** button is no longer highlighted. You may also receive a prompt to add the contact to your *WSJT-X* log, if you've enabled that prompt in the settings menus (see **Figure 33**).

TRY CALLING CQ

After you've made a few contacts, try calling CQ yourself. Once again, check the **AUTO SEQ** box, but this time you need to also check the **CALL 1ST** box to its immediate right. The **CALL 1ST** box is important because it lets *WSJT-X* know that you want to respond automatically to the first decoded responder to your CQ. If you don't check that box, you will need to double-click on the text line from a responding station to start the contact sequence.

Pick a point in the waterfall window where no one seems to be transmitting. Double-click your mouse on that point and the red/green brackets will jump to that position.

Now look in the **GENERATE STD MSGS** window and click your mouse in the circle on the bottom line, just to the left of the **TX 6** button, which should be your CQ message. Click the **ENABLE TX** button and *WSJT-X* should begin transmitting. If you happen to begin at the middle or near the end of a transmit cycle, the first transmission may be short. *WSJT-X* will likely wait until the *next* cycle is complete before sending the CQ transmission again.

During receive cycles *WSJT-X* will be "watching" for text that contains your call sign, the call sign of another station, and a grid square — in other words, a reply! If no one replies, *WSJT-X* will send CQ again. When someone does reply, it will automatically go through the "dance" of completing the contact while you relax and watch.

ADDED FEATURES

WSJT-X software is constantly evolving, and it is nearly impossible to provide up-to-date information within the pages of a printed book. For instance, in the **ADVANCED** tab in **SETTINGS** (see **Figure 34**), later versions of *WSJT-X* added the "Fox/Hound" configuration to handle DX pileups. The DX station — the so-called "fox" — transmits low in the passband, and those in pursuit of contacts — the "hounds" — are spread among other frequencies above 1000 Hz. The fox can respond to multiple stations simultaneously, greatly increasing the contact rate. Fox/hound operations generally occur on frequencies outside the normal FT8 watering holes.

There is also a configuration specific to contest applications, one that allows sending the correct exchange for the event. Currently, VHF contests (grid squares), ARRL RTTY Roundup (RST and state), and ARRL Field Day (entry class and ARRL section) are supported as well as contests sponsored by other organizations that use one of these exchange formats. *WSJT-X* can communicate with some contest logging/station management software such as *N1MM+ Logger*.

12.5 FT4

WSJT-X Version 2.1 debuted with a new mode designed specifically for contesting: FT4 (see **Figure 35**). Compared with FT8 it is 3.5 dB less sensitive and requires 1.6 times the bandwidth, but it offers the potential for twice the QSO rate. Once you've set up *WSJT-X* as described in this section, you will be ready to try FT4. The exchanges are very much the same as FT8, only much faster. FT4 also supports the text exchanges required according to the rules of various contests.

At the time of this writing, FT4 was in use alongside FT8 for contest applications. Contests such as ARRL RTTY Roundup offer categories for FT8/FT4 operation as well as conventional RTTY. The Worldwide Digi DX Contest and FT Roundup are for the FT8/FT4 modes exclusively. Other contests for these modes are cropping up as well.

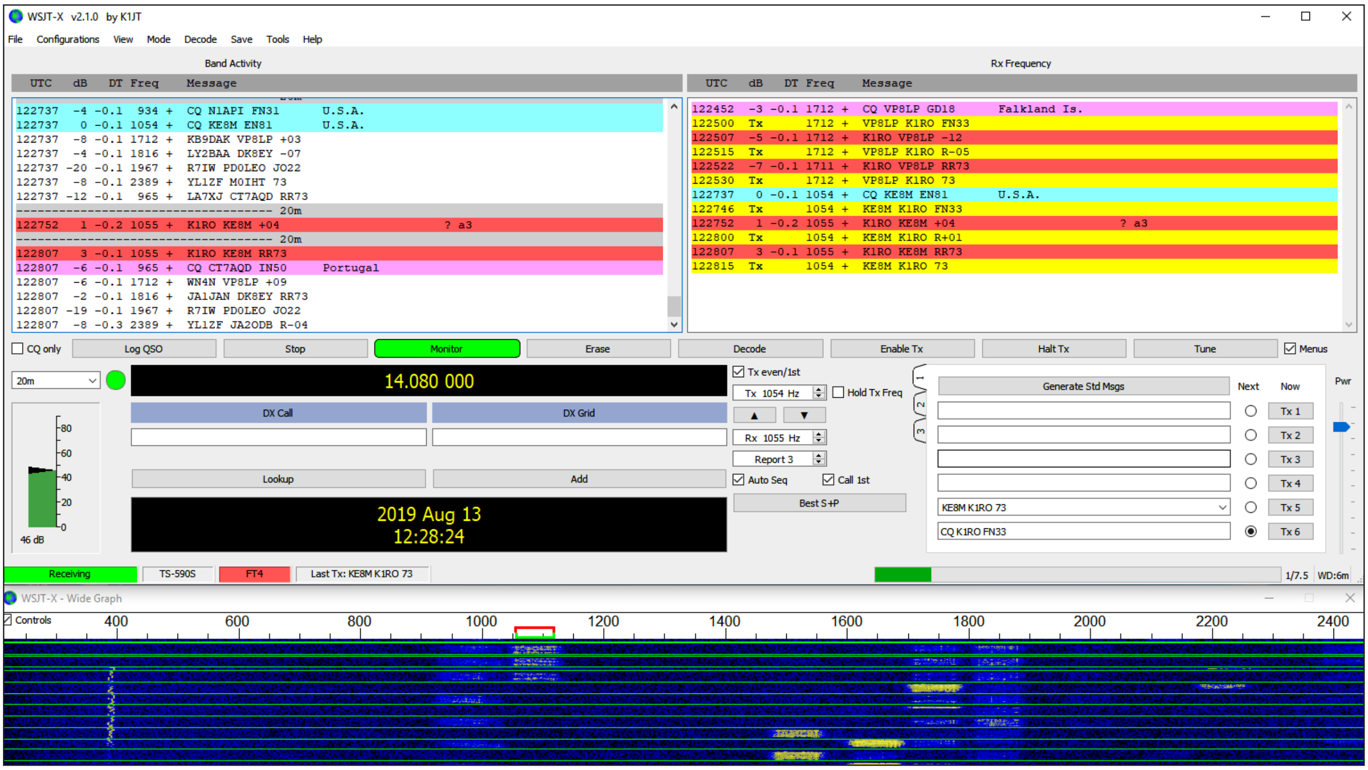


Figure 35 — FT4, optimized for contesting, uses 7.5 second sequences. Operation is similar to FT8.

12.6 Moving Stations from Band to Band

The initial structure of FT8/FT4 messages resulted in most VHF activity congregating on 50 and 144 MHz. Since moving stations between bands and modes is an important part of VHF+ contesting, users requested enhancements to the protocols. This would encourage frequency changes to the upper UHF and microwave bands where the contest contacts had higher point values. Multiband rover stations using the FT8 and FT4 modes would greatly improve their ability to make contacts on several more bands using this coordinating information, as well.

In response to a proposal by Bob Lear, W4ZST, and VHF clubs, recognizing the limitations of the basic set of pre-programmed macros, the *WSJT-X* development team added the ability to send band-change messages as one of the Tx# messages. This change was effective with version 2.5.3 in Dec 2021. The new capability is described in the *WSJT-X* User Guide at **www.physics.princeton.edu/pulsar/K1JT/wsjtx.html**. Read section 4.4, "Tx Macros," in part 4, "Settings," for complete information.

Summarizing, the messages now support the use of the long-used band abbreviations ABCD, etc. that represent 50 MHz, 144 MHz, 222 MHz, 432 MHz, and so forth. The band abbreviation is followed by a mode abbreviation: V (SSB Voice), W (CW), or 8 (FT8). For example, the message $DXCALL BV 228 means "send the call sign of the station being worked and request they QSY to 2-meter voice on 144.228 MHz." The station being queried would presumably send an acknowledgement message, although that is not required. Macros can be pre-programmed with a message that is selected from the **TX5** drop-down menu. Text can also be entered into the **TX5** message at any time.

For more information about using macros to move stations, read the article, "Moving on up with FT8" by Phil Miguelez, WA3NUF, with contributions by Joe Taylor, K1JT, in the November 2021 issue of the Mt. Airy V.H.F. Radio Club's newsletter, *Cheese Bits* at **www.packratvhf.com** in the **NEWSLETTER** section. The codes and techniques may change as operating practices and the *WSJT-X* software evolve.

13 Bibliography and References

BOOKS

Ford, S., WB8IMY, *Get on the Air with HF Digital*, 3rd Edition (ARRL, 2022).

Popiel, G., KW5GP, *High Speed Multimedia for Amateur Radio* (ARRL, 2016).

Taylor, J., K1RFD, *VoIP: Internet Linking for Radio Amateurs* (ARRL, 2009).

Burningham, J., W2XAB, *Amateur Radio Guide to Digital Mobile Radio* **www.trbo.org/docs/Amateur_Radio_Guide_to_DMR.pdf**.

ARTICLES

Feierman, B., K3EUI, "Join the FT8 Revolution," *QST*, Jan. 2018, pp. 41 – 44.

Ford, S., WB8IMY, "JT65 — The 'Musical' Mode," *QST*, Apr. 2011, pp. 45 – 46. Feedback: Jun 2011 *QST*, p. 64.

Franke, S., K9AN; Somerville, B., G4WJS; Taylor, J., K1JT, "The FT4 and FT8 Communication Protocols," *QEX*, Jul./Aug. 2020, pp. 7 – 17.

Kionka, M., KIØGO, "Project 25 for Amateur Radio," *QST*, Sep. 2008, pp. 36 – 38.

Loveall, P., AE5PL, "D-STAR Uncovered," *QEX*, Nov./Dec. 2008, pp. 50 – 56.

Maeda, M., JR1AQN, "Who's Using FT8?," *QST*, Feb. 2019, pp. 72 – 74.

Martinez, P., G3PLX, "PSK31: A New Radio-Teletype Mode," *QEX*, Jul./Aug. 1999, pp. 3 – 9.

Muns, E., WØYK, "FT4 and FT8 Contesting," Digital Contesting, *NCJ*, May/Jun. 2020, p. 29.

Ostrowski, J., KI5FJ, "Evaluating Receive Performance with FT8," Technical Correspondence, *QST*, May 2021, p. 52.

Schier, D., K1UHF, "The Ins and Outs of a Sound Card," *QST*, Oct 2003, pp. 33 – 36.

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

Taylor, J., K1JT, "FT8 in the ARRL RTTY Roundup," *QST*, Jan. 2019, pp. 72 – 73.

Taylor, J., K1JT; Franke, S., K9AN; Somerville, B., G4WJS, "Work the World with WSJT-X, Part 1: Operating Capabilities," *QST*, Oct 2017, pp. 30 – 36.

Taylor, J., K1JT; Franke, S., K9AN; Somerville, B., G4WJS, "Work the World with WSJT-X, Part 2: Codes, Modes, and Cooperative Software Development," *QST*, Nov. 2017, pp. 34 – 39.

Taylor, J., K1RFD, "Computer Sound Cards for Amateur Radio," Product Review, *QST*, May 2007, pp. 63 – 69.