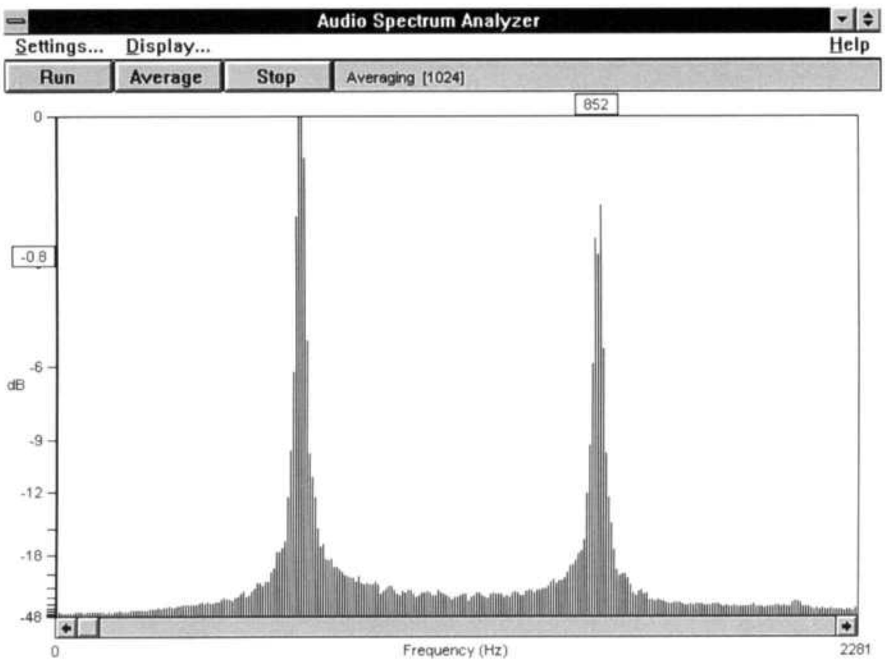
QEX

$1.75

ARRL Experimenter's Exchange

December 1993



Inexpensive FFT Tools

QEX: The ARRL
Experimenter's Exchange
American Radio Relay League
225 Main Street
Newington, CT USA 06111

Non-Profit Org.
US Postage
PAID
Hartford, CT
Permit No. 2929

QEX (ISSN: 0886-8093) is published monthly by the American Radio Relay League, Newington, CT USA.

Application to mail at second-class postage rates is pending at Hartford, CT.

Offices

225 Main St, Newington, CT 06111-1494 USA
Telephone: 203-666-1541
Telex: 650215-5052 MCI
FAX: 203-665-7531 (24 hour direct line)
Electronic Mail: MCIMAILID: 215-5052
Internet:qex@arrl.org

Subscription rate for 12 issues:

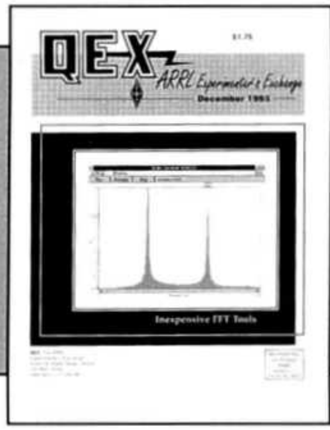
In the US by Third Class Mail: ARRL Member $12, nonmember $24;

US, Canada and Mexico by First Class Mail:
ARRLMember $25, nonmember $37;

Elsewhere by Airmail:
ARRLMember $48 nonmember $60.

QEX subscription orders, changes of address, and reports of missing or damaged copies may be marked: *QEX* Circulation. Postmaster: Form 3579 requested. Send change of address to: American Radio Relay League, 225 Main St, Newington, CT 06111-1494.

Members are asked to include their membership control number or a label from their *QST* wrapper when applying.



About the Cover:

Software tools for signal analysis, like this inexpensive FFT program, are part of the experimenter's "software junkbox," says NK6K.

142

Features

Columns

December 1993 QEX Advertising Index

THE AMERICAN RADIO RELAY LEAGUE

The American Radio Relay League, Inc, is a noncommercial association of radio amateurs, organized for the promotion of interests in Amateur Radio communication and experimentation, for the establishment of networks to provide communications in the event of disasters or other emergencies, for the advancement of radio art and of the public welfare, for the representation of the radio amateur in legislative matters, and for the maintenance of fraternalism and a high standard of conduct.

ARRL is an incorporated association without capital stock chartered under the laws of the state of Connecticut, and is an exempt organization under Section 501(c)(3) of the Internal Revenue Code of 1986. Its affairs are governed by a Board of Directors, whose voting members are elected every two years by the general membership. The officers are elected or appointed by the Directors. The League is noncommercial, and no one who could gain financially from the shaping of its affairs is eligible for membership on its Board.

"Of, by, and for the radio amateur, "ARRL numbers within its ranks the vast majority of active amateurs in the nation and has a proud history of achievement as the standard-bearer in amateur affairs.

A bona fide interest in Amateur Radio is the only essential qualification of membership; an Amateur Radio license is not a prerequisite, although full voting membership is granted only to licensed amateurs in the US.

Membership inquiries and general correspondence should be addressed to the administrative headquarters at 225 Main Street, Newington, CT 06111 USA.

Telephone: 203-666-1541 Telex: 650215-5052 MCI.
MCIMAIL (electronic mail system) ID: 215-5052
FAX: 203-665-7531 (24-hour direct line)

Officers

President: GEORGE S. WILSON III, W4OYI
1649 Griffith Ave, Owensboro, KY 42301

Executive Vice President: DAVID SUMNER, K1ZZ

Purpose of *QEX*:

1) provide a medium for the exchange of ideas and information between Amateur Radio experimenters
2) document advanced technical work in the Amateur Radio field
3) support efforts to advance the state of the Amateur Radio art

All correspondence concerning *QEX* should be addressed to the American Radio Relay League, 225 Main Street, Newington, CT 06111 USA. Envelopes containing manuscripts and correspondence for publication in *QEX* should be marked: Editor, *QEX*.

Both theoretical and practical technical articles are welcomed. Manuscripts should be typed and doubled spaced. Please use the standard ARRL abbreviations found in recent editions of *The ARRL Handbook*. Photos should be glossy, black and white positive prints of good definition and contrast, and should be the same size or larger than the size that is to appear in *QEX*.

Any opinions expressed in *QEX* are those of the authors, not necessarily those of the editor or the League. While we attempt to ensure that all articles are technically valid, authors are expected to defend their own material. Products mentioned in the text are included for your information; no endorsement is implied. The information is believed to be correct, but readers are cautioned to verify availability of the product before sending money to the vendor.

Empirically Speaking

What I Want for Christmas

Christmas approaches, and visions of sugar plums dance in children's heads. Here at *QEX*, we have visions, too: visions of articles. Lots of them. Starting in January, *QEX* will expand to 32 pages each and every month. To fill those pages, we'll need plenty of top-notch articles. Here's a brief list of what I'd like to see. (Are you listening, Santa?)

Antennas

One of the areas we have not covered as well as we'd like is that of antennas. With the marvelous antenna design tools available today, you would think our "in" baskets would be overflowing with manuscripts describing clever new antenna designs. Well, they're not. Oh, we still get some, but by the time the *QST* editors and the *Antenna Compendium* editor get done staking their claims, there's not much left for *QEX*! One reason is that most of the antenna submissions we do get are heavy on specific construction plans. Which is fine; hams want to read that kind of thing. But in *QEX*, we would like to cover antenna subjects of more interest to those readers who are analyzing and developing antenna models and designs.

Digital Voice

This is a hobby horse we climb onto regularly. With other voice communication services "going digital" at exponentially increasing rates, it's hard to see how amateur analog voice communications can fit into our future plans. Yet, relatively little progress in this area has been reported by the amateur community. Designs to send data at bit rates easily fast enough for digital voice are readily available. (Such as 56 kbit/s WA4DSY modems, or 2-Mbit/s microwave links as described in the ARRL *Handbook*.) And devices to digitize voice are no further away than your PC's sound card. So it's a wonder that we don't hear more about experiments in digital voice transmission. We sure would *like* to hear more!

RF Design

Sure, we have Zack Lau's bimonthly "RF" column, but there's plenty more ground that could be covered. With the field of RF extending from VLF up through microwaves, and encompassing techniques and devices that range from 811s to PHEMTs, no one column can possibly encompass it all. If you're working in RF, how about letting us know what you've developed?

Software

Probably one of the most fast-moving technologies, software for amateur experimenters ranges from electronic design tools to analysis tools (such as those described by NK6K this month) to software development tools themselves. There are lots of programmers writing interesting software for amateur applications. *QEX* readers would like to hear from those working in this area about new developments, be they algorithms, applications or simply "old favorites" dressed up in modern user interfaces.

The Other Stuff

There are plenty of other areas amateur experimenters find interesting: DSP, spread spectrum, new devices, and a plethora of others. We'd like to bring you all of that. Of course, the most exciting articles are the ones we don't solicit—because we have no idea that the breakthrough idea being described is upon us. Whether your latest project falls into that category, or is "just" a well-implemented solution to a problem, why not be Santa's helper, and write it up for *QEX*. And if you can't get it done for Christmas, hey, Groundhog Day is just around the corner!

This Month in *QEX*

In part 1 of two parts, Ulrich Rohde, KA2WEU, begins an in-depth look (not for the math phobic) at oscillator noise performance by covering the ways in which phase noise is described analytically.

In the final part of our series on automatic link establishment, Christopher Redding and Dennis Bodson, W4PWF, describe the part that link security plays—and will play—in the ALE process.

Following up on his 8085 microcomputer/EPROM programmer of last month, Sam Ulbing, N4UAU, describes a programmer for 87C51 single-chip microcontrollers that plugs into the 8085 unit.

This month in "Digital Communications," Harold Price, NK6K, describes some low-cost tools he dug out of his "software junkbox" to do some simple signal analysis. He also teases us with some preliminary results of an implementation of convolutional encoding for FEC.

If you searched in vain for the book review mentioned in this space last month, never fear. It's here this month. Really, it is. We promise.—*KE3Z, email: jbloom@arrl.org (Internet)*

All About Phase Noise In Oscillators

Part 1—How Oscillator Noise is Described and Analyzed

by Ulrich L. Rohde, KA2WEU

One of the key measurements of oscillator operation is the signal-to-noise ratio of the oscillator at certain offsets from the carrier. Ideally, the output of the oscillator would consist of only a single-frequency signal, but in reality there is energy—noise—at nearby frequencies as well. Although oscillator amplitude variations will result in sideband energy, most of this noise is phase noise, main sources of which are the flicker noise contribution of the (semiconductor) active device, the noise figure of the device under large-signal conditions, and the filtering effect of the resonator.

The actual phase noise of an oscillator is a composite of near- and far-carrier noise. This noise is most easily viewed in terms of its *spectral power density*, which is the power contained in a 1-Hz bandwidth at a particular frequency. Measuring the spectral power density of an oscillator results in a curve such as that of Fig 1. Rather than all of the power being concentrated at the oscillator frequency, some is distributed at frequencies around the oscillator frequency.

As noise is a form of instability, it is useful to characterize frequency stability in the time domain, too. Short-term stability extends between a very small fraction of a second to 1 second, occasionally up to 1 minute, although normally values for the stability over 1 second and 1 minute will be about the same. Long-term stability, or aging, is a frequency change typically expressed in terms of parts in 10^{-10} or 10^{-11} per day. Again, this information is in the time domain; in the frequency domain, we use terms like "random walk," "flicker," and "wide phase noise" to describe the slope of the spectral density curve.

52 Hillcrest Dr
Upper Saddle River, NJ 07458

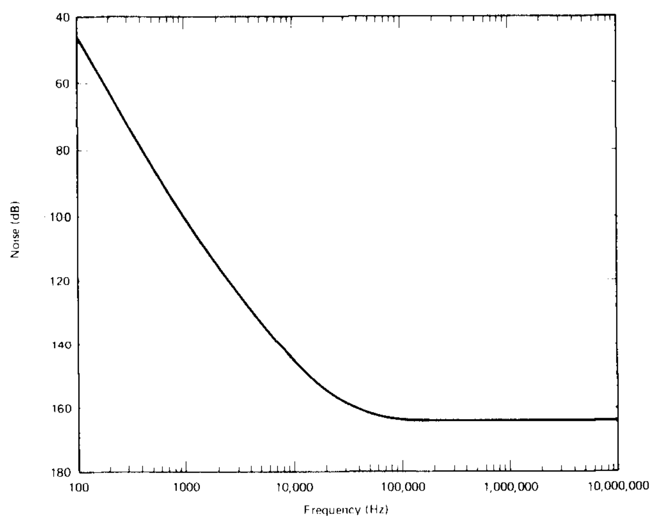


Fig 1—Typical noise sideband curve of a free-running oscillator (Rohde and Schwarz signal generator SMDU).

The Fourier frequency, usually labeled f_m, is at times called the sideband frequency, offset frequency, modulation frequency, or baseband frequency. In this article we will refer to it as the *offset frequency* and will describe the signal-to-noise ratio of an oscillator at a given offset from the center frequency. This way of describing oscillator noise--the spectral power density--is the most common way of characterizing phase noise. Probably this is because this is precisely what is seen on a spectrum analyzer when the AM noise contribution is insignificant. Note that the spectrum analyzer display--and the oscillator noise--is symmetrical around the center frequency. Curves like that of Fig 1 therefore only show one side of the noise spectrum.

Looking at sideband noise in a 1-Hz bandwidth leads to the definition of $L(f_m)$: the ratio of the single-sideband noise power in a 1-Hz bandwidth, f_m hertz away from the carrier frequency, to the total signal power. This is what is plotted in Fig 1.

These unwanted frequency components are referred to as *oscillator noise*. The oscillator output $S(t)$ can be expressed by the equation

$$S(t) = A(t)\cos[\omega_o t + \theta(t)] \qquad (1)$$

where $A(t)$ describes the amplitude variation as a function of time and $\theta(t)$ is the phase variation, or *phase noise*. A well-designed, high-quality oscillator is very amplitude stable, so $A(t)$ can be considered constant. Thus, for a constant-amplitude signal, all oscillator noise is due to $\theta(t)$. Leeson has developed a linear model that describes the origins of phase noise in oscillators, and since it closely fits experimental data, the model is widely used in describing the phase noise of oscillators and frequency synthesizers.[1] Leeson's linearized oscillator model for phase noise will be described in the second of these articles, but first we will develop the relation between the observed power spectral density function and $\theta(t)$.

A Model For SSB Phase Noise

A carrier signal of amplitude V which is frequency modulated by a sine wave of frequency f_m can be represented by the equation:

$$S(t) = V\cos\left(\omega_o t + \frac{\Delta f}{f_m}\sin\omega_m t\right) \qquad (2)$$

where Δf is the peak frequency deviation and $\theta_p = \Delta f / f_m$ is the peak phase deviation, often referred to as the *modulation index*, β. Equation 2 can be expanded as

$$S(t) = V[\cos(\omega_o t)\cos(\theta_p \sin\omega_m t) - \sin\omega_o t \sin(\theta_p \sin\omega_m t)] \qquad (3)$$

If the peak phase deviation is much less than 1 (that is, $\theta_p << 1$),

$$\cos(\theta_p \sin\omega_m t) \approx 1$$

and

$$\sin(\theta_p \sin\omega_m t) \approx \theta_p \sin\omega_m t$$

Thus for ($\theta_p << 1$), the signal $S(t)$ is approximately equal to

$$S(t) = V[\cos(\omega_o t) - \sin\omega_o t(\theta_p \sin\omega_m t)] = V\left\{\cos(\omega_o t) - \frac{\theta_p}{2}[\cos(\omega_o + \omega_m)t - \cos(\omega_o - \omega_m)t]\right\} \qquad (4)$$

[1]Notes appear on page 6.

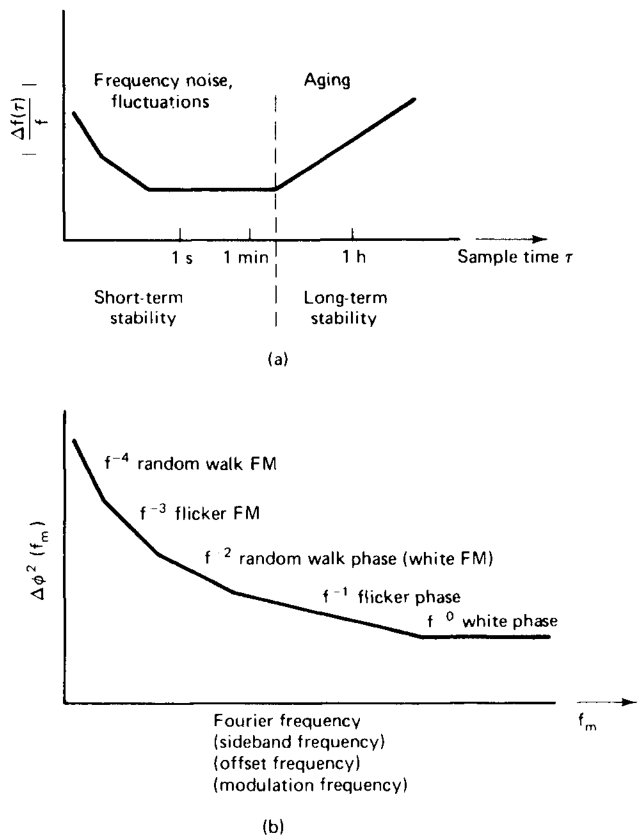


Fig 2 Characterization of the sources of noise sidebands in the (a) time and (b) frequency domains.

Table 1

£* at 1 kHz (dBc)	Slope of $£(f_m)$ Exponent	dB/oct	Residual FM Δf_{res} 50 Hz to 3 kHz	300 Hz to 3 kHz	20 Hz to 15 kHz
−100	0	0	1.34	1.34	15.0
−100	−1	−3	0.95	0.94	4.74
−100	−2	−6	0.77	0.73	1.73
−100	−3	−9	0.90	0.68	1.15

*For any £ at 1 kHz different to −100 dBc, multiply Δf_{res} of the table by

$$\text{antilog}\frac{100 - |£ \text{ at 1 kHz/dBc}|}{20}$$

The table does not take into account any microphonic or spurious sidebands.

Example: £ at 1 kHz = −88 dBc, slope −9 dB. For bandwidth 20 Hz to 15 kHz:

$$\Delta f_{res} = 1.15 \text{ Hz} \times \text{antilog}\frac{100 - 88}{20} = 4.6 \text{ Hz}$$

That is, when the peak phase deviation is small, the phase deviation results in frequency components on each side of the carrier of amplitude $\theta_p/2$. This frequency distribution of a narrowband FM signal is useful for interpreting an oscillator's power spectral density due to phase noise. The phase noise in a 1-Hz bandwidth has a noise power-to-carrier power ratio of

$$L(f_m) = \left(\frac{V_n}{V}\right)^2 = \frac{\theta_p^2}{4} = \frac{\theta_{rms}^2}{2} \qquad (5)$$

The total noise is the noise in both sidebands and will be denoted by S_θ. That is,

$$S_\theta = 2\frac{\theta^2_{\text{rms}}}{2} = \theta^2_{\text{rms}} = 2L(f_m) \quad (6)$$

With this interpretation of the noise power, the noise can now be described in terms of its origin; see Fig 2.

Oscillator noise can be expressed in a number of ways. Before delving into the oscillator itself, in part 2, we'll discuss the other ways of expressing oscillator phase noise.

Spectral Density of Frequency Fluctuations

Stability measurements using frequency comparators give the *spectral density of frequency fluctuations* ,

$$S_{\Delta f}(f_m) = \Delta f^2_{\text{rms}} \quad (7)$$

To relate the spectral density of frequency fluctuations to the spectral density of phase noise, recall that

$$\Delta f(t) = \frac{1}{2\pi}\frac{d\Delta\theta(t)}{dt} \quad (8)$$

Transformed into the frequency domain, this gives

$$\Delta f(f_m) = f_m \Delta\theta(f_m) \quad (9)$$

$$S_{\Delta f}(f_m) = \Delta f^2_{\text{rms}}(f_m) = f^2_m S_{\Delta\theta}(f_m) = 2f^2_m L(f_m) \quad (10)$$

The National Institute of Standards and Technology (NIST, formerly the National Bureau of Standards) proposes to standardize the definition of the spectral density of fractional frequency fluctuations. [3] The instantaneous frequency deviation is normalized to the carrier frequency f_o.

$$y(t) = \frac{\Delta f(t)}{f_o} \quad (11)$$

$$S_y(f_m) = \frac{1}{f_o^2}S_{\Delta f}(f_m) = \frac{f_m^2}{f_o^2}S_{\Delta\theta}(f_m) = \frac{2f_m^2}{f_o^2}L(f_m) \quad (12)$$

Characterizing fractional frequency fluctuations allows better comparison of sources with different carrier frequencies.

Residual FM Related to $L(f_m)$

Residual FM, the total RMS frequency deviation within a specified bandwidth, is another common way to specify the frequency stability of signal generators. Commonly used bandwidths are 50 Hz to 3 kHz, 300 Hz to 3 kHz and 20 Hz to 15 kHz.

$$\Delta f_{\text{res}} = \sqrt{2}\sqrt{\int_a^b L(f_m) f_m^2 df_m} \quad (13)$$

Table 1 correlates Δf_{res} and $L(f_m)$ for specific slopes of $L(f_m)$ and L at 1 kHz = -100 dBc.

Allan Variance Related to $L(f_m)$

For many applications, such as high-stability crystal oscillators or Doppler radar systems, it is more useful to describe frequency stability in the time domain. The time-domain characterization is based on the sample variance of fractional frequency fluctuations. Averaging differences of consecutive sample pairs with no dead time in between yields the *Allan variance*, $\sigma_y^2(\tau)$, which is the proposed standard measure of frequency stability.

$$\sigma_y^2(\tau) \sim \frac{1}{2(M-1)}\sum_{k=1}^{M-1}(\bar{y}_{k+1} - \bar{y}_k)^2 \quad (14)$$

$\bar{y}_k$ is the average fractional frequency difference of the kth sample measured over sample time .

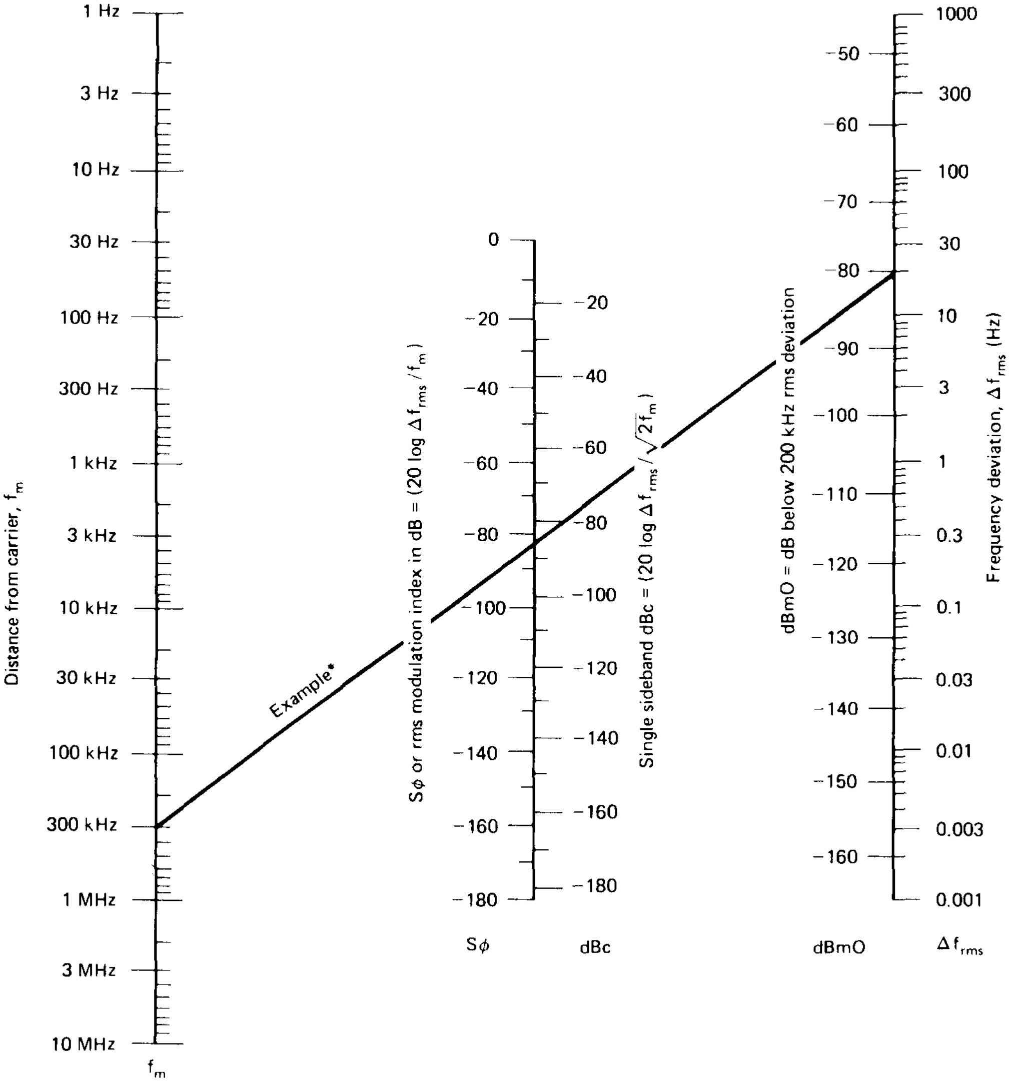
Conversions from frequency- to time-domain data and vice versa are possible but tedious. The power spectrum, $L(f_m)$, needs to be approximated by integer slopes of 0, −1, −2, −3, and −4. Then conversion formulas (see Table 2) can be applied. A good description of this procedure is given in Notes 2 and 3.

Table 2 CONVERSION TABLE*

	Slope of $\sigma_y^2(\tau)$	$\sigma_y(\tau) =$	$\mathcal{L}(f) =$	Slope of $\mathcal{L}(f)$
White phase	−2	$\frac{\sqrt{\mathcal{L}(f)f_h}}{2.565 f_o}\tau^{-1}$	$\frac{[\sigma_y(\tau)\tau f_o(2.565)]^2}{f_h} f^0$	0
Flicker phase	−1.9	$\frac{\sqrt{\mathcal{L}(f) f[2.184 + \ln(f_h\tau)]}}{2.565 f_o}\tau^{-1}$	$\frac{[\sigma_y(\tau)\tau f_o(2.565)]^2}{2.184 + \ln(f_h\tau)} f^{-1}$	−1
White frequency	−1	$\frac{\sqrt{\mathcal{L}(f)f^2}}{f_o}\tau^{-1/2}$	$[\sigma_y(\tau)\tau^{1/2} f_o]^2 f^{-2}$	−2
Flicker frequency	0	$\frac{1.665\sqrt{\mathcal{L}(f)f^3}}{f_o}\tau^0$	$0.361[\sigma_y(\tau)f_o]^2 f^{-3}$	−3
Random walk frequency	+1	$\frac{3.63\sqrt{\mathcal{L}(f)f^4}}{f_o}\tau^{1/2}$	$[(0.276)\sigma_y(\tau)\tau^{-1/2} f_o]^2 f^{-4}$	−4

*τ = measurement time, $y = \Delta f_o/f_o$, f_o = carrier, f = sideband frequency, f_h = measurement system bandwidth.

Table 3 NOISE-CONVERSION NOMOGRAPH: RELATIONSHIP AMONG MODULATING FREQUENCY (f_m), POWER SPECTRAL DENSITY OF PHASE (S_ϕ), MODULATION INDEX, SIDEBAND TO CARRIER RATIO (dBc), dBmO, AND FREQUENCY DEVIATION (Δf_{rms})[a]



[a]Use consistent measurement bandwidth. Example: 20-Hz deviation in a 1-kHz band at 300 kHz from carrier = single-sideband dBc of −87 dB in a 1-kHz band.

We have covered the most frequently used measures of phase noise and have interrelated them. Before we take a look at the generation of phase noise in oscillators in part 2, let us take a look at the noise-conversion nomograph in Table 3. The example given there is self explanatory.

As most of these relationships, for reasons of convenience, are expressed in decibels rather than absolute values, the following formulas are commonly used:

$$L(f_m) = 10 \log_{10} \left(\frac{\Delta f_{\text{peak}}}{2 f_m} \right)^2 \tag{15}$$

$$L(f_m) = 10 \log_{10} \left(\frac{\Delta f_{\text{rms}}}{\sqrt{2} f_m} \right)^2 \tag{16}$$

$$L(f_m) = 20 \log_{10} \frac{\Delta f_{rms}}{\sqrt{2} f_m} \tag{17}$$

$$L(f_m) = 20 \log_{10} \frac{\theta_d}{2} \tag{18}$$

Notes

[1]Leeson, D.B., "A Simple Model of Feedback of Oscillator Noise Spectrum," *Proceedings of the IEEE*, 1966, p 329-330.

[2]Fischer, M.C., "Frequency Stability Measurement Procedures," Eighth Annual Precise Time and Time Interval Applications and Planning Meeting, December 1976.

[3]Howe, D.A., "Frequency Domain Stability Measurements: A Tutorial Introduction," *NBS Technical Note 679*, March 1976.

The Growing Family of Federal Standards for HF Radio Automatic Link Establishment (ALE)

Part VI: Federal Standard 1049—The Future of ALE Operation in Stressed Environments

Link establishment can include linking protection—authentication—with the addition of standard protocols.

Christopher Redding and Dennis Bodson, W4PWF

As part of an ongoing standards development effort within the federal government, a proposed standard has been developed to provide protection to the signaling in high frequency (HF) radios that implement Federal Standard 1045 (FED-STD-1045) for automatic link establishment (ALE).[1] FED-STD-1045 is the baseline standard in a series of federal standards which specifies interoperability and performance requirements for ALE radios. Included in the series is proposed Federal Standard 1049 (FED-STD-1049) entitled *HF Radio Automatic Operation in Stressed Environments*. Section 1 of FED-STD-1049, entitled "Linking Protection (LP)," specifies requirements for the mechanism employed to provide transparent protection to ALE signaling.

ALE allows stations to automatically establish links, determine the best available channel, and transfer digital orderwire messages. While ALE technology automates and expedites the linking process, it creates a potential vulnerability among stations to other linking transmissions. These transmissions can be in the form of an unintended caller imitating a legitimate ALE station or a simple playback of a previous transmission. In both scenarios, the true identity of the caller may not be known; therefore, some sort of authentication is needed. The linking protection mechanism that has been developed counters these unwanted intrusions, as well as provides a measure of confidentiality to the ALE addressing and orderwire message transmission. If the Amateur Radio community operates ALE-capable systems in the future, linking protection will be a feature that should be given serious consideration.

[1]Notes appear on page 9.

Christopher Redding
12540 McKenzie Ct
Broomfield, CO 80020

Dennis Bodson, W4PWF
233 N Columbus St
Arlington, VA 22203

ALE Review

ALE technology enables radio stations to automatically initiate and establish bilateral connectivity. In the process of establishing links, a link quality analysis (LQA) is performed which allows the ALE radio to select the best available (frequency) channel. The ALE protocol also has the capability to exchange short digital text messages, even during the linking process.

The basic link-establishment process is accomplished via a three-way handshake between two or more stations. The three-way handshake consists of the call, response and acknowledgment, as shown in Fig 1. The calling station initiates the call by transmitting a series of 24-bit words containing a "To" preamble and the called station's address, and concludes with a word containing a "This Is" preamble and its own address. The called radio (or radios), which typically is scanning a number of channels, stops on the channel on which it hears the call and decodes the ALE words to determine if the call is intended for itself. The called radio answers with a short response beginning with two words containing the "To" preamble and the address of the calling station, and concludes by transmitting two words with a "This Is" preamble and its own address. When the original calling station receives this response, it is assured of bilateral connectivity and sends an acknowledgment to the called station, thus completing the three-way handshake for establishing the link. The time required for establishing a link is approximately 14 seconds (for stations scanning 10 channels at 2 channels per second). Also, note that orderwire messages can be inserted in any one of the three message sections of the handshake as indicated by the (▼) symbol in Fig 1. After a link has been established, the operator is signaled and voice or data communication can be initiated. Termination of a link is accomplished by the transmission of a "return to scan" signal or by the use of an internal timer, which automatically returns the radio to scanning or available status after a preset period of inactivity.

In block diagram terms, the ALE controller, shown in Fig 2, consists of three distinct modules: the ALE protocol module, the forward error correction (FEC) module and the ALE modem. The ALE protocol module incorporates standard protocols for link establishment, link quality analysis (LQA) and digital orderwire message transmission. The FEC module incorporates three error detection and correction techniques: Golay encoding/decoding, interleaving and deinter leaving, and triple redundancy/majority voting. The ALE modem employs 8-ary frequency-shift keying (FSK) modulation, with each of the 8 tones representing 3 bits of data. The resultant over-the-air data rate is 375 bits per second.

The protection module and its associated LP control module is added to the ALE system for protected operation. It is located before the FEC module so that the error-correcting power of the FEC module can be used to its full advantage.

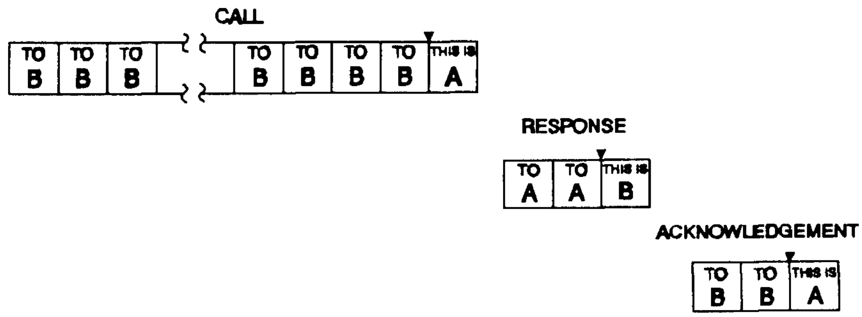


Fig 1—ALE three-way handshake.

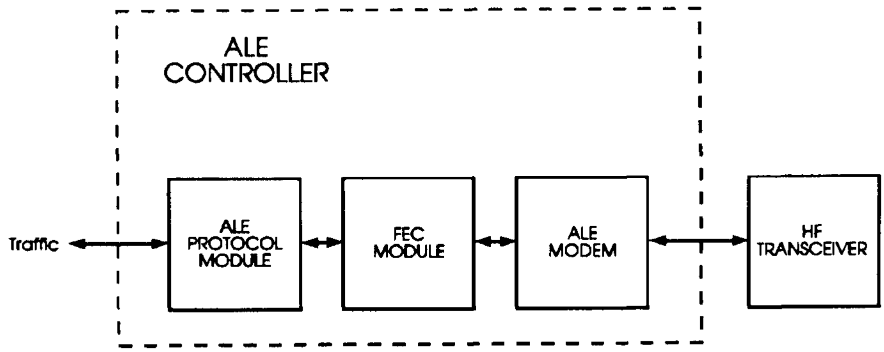


Fig 2—ALE system block diagram.

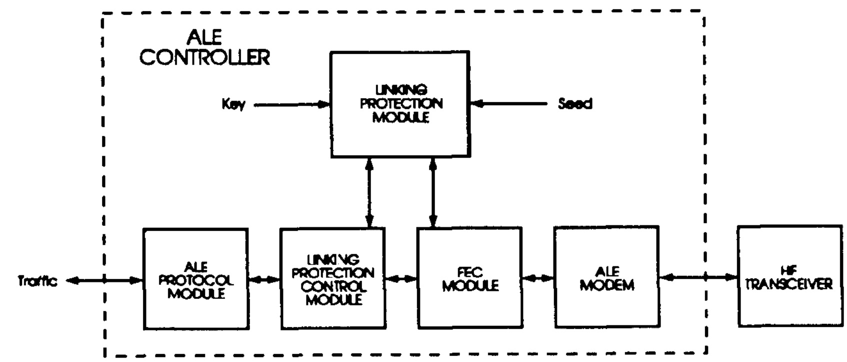


Fig 3—Protected ALE system block diagram.

Linking Protection

Linking protection is a mechanism for preventing other stations from establishing unintended links or interfering with the establishment of legitimate links. It is achieved through an automatic authentication process. Authentication is provided by processing the linking data through an appropriate algorithm before transmission, and processing the received linking data through the same algorithm. Verifying the identity of the sender is the primary objective of LP, but can also provide address and digital orderwire message protection. The digital order-wire message capability, embedded within the ALE signaling, can also be protected by the LP mechanism. Linking protection makes it extremely difficult for another station to play back previously recorded valid messages or generate new messages that the receiver will accept as valid. Additional information on this topic is given by Redding and Johnson.[2]

Protected ALE transmissions are protected by the use of an appropriate algorithm, a key variable and seed information. A block diagram of a protected ALE system is shown in Fig 3. Protection is performed on each individual 24-bit word. The seed, consisting of a known 64-bit time-of-day (TOD) code and the frequency-of-transmission information, is used to vary the protection function on a short-term basis. The minimum incremental change in the time-of-day used in the seed is referred to as the protection interval (PI). Because of the time-related protection interval, previously recorded messages that are played back will appear as unintelligible information to the receiver, and will be treated as such. Use of time and frequency varying information as an input to the protection function means that no extra synchronization bits or preambles are required in protected ALE transmissions. Although this protection method requires no overhead bits, it does require stations to keep accurate time and periodically transfer the timing information for synchronization maintenance.

A method for stations to obtain synchronization is required since protected radios operate on time-based protection intervals. The method developed is based upon an incremental two-step process. The first step, termed coarse synchronization, relies on stations synchronizing their clocks to within one minute of each other. After coarse synchronization is obtained, a time synchronization protocol is utilized to obtain fine synchronization by distributing timing-related information. The fine synchronization times range from 60 seconds to 2 seconds or less, with the lowest protection level employing a single PI of 60 seconds.

Several levels, or strengths, of LP have been specified so that the majority of users are provided with sufficient protection. Protection levels are distinguished by the strength of the algorithm and the length of the PI (ie, the protection increases as the PI decreases). Because of the unique requirements of LP, special algorithms have been developed for unclassified users. Linking protection can be implemented in hardware or software depending on the particular requirements of the procuring organization. The lowest level of LP can be easily implemented in software; therefore, its costs are kept to a minimum for the user who requires only this minimum level of protection.

The LP function only protects the linking process and any digital information transferred during that process; after a successful link is accomplished between two or more stations, no protection exists. Separate auxiliary higher-speed modems can be utilized after the initial ALE process if desired.

Although not applicable to Amateur Radio, data encryption capabilities for inclusion within the ALE radios are being developed in proposed Federal Standard 1049 Section 3. Details of FS-1049/3 will be presented at a later time when the technology is developed, implemented and tested.

Summary

Protection of the ALE signaling results in protection of the linking function, as well as providing a degree of privacy to the ALE addresses. The transparency of LP provides various benefits such as establishing a link in the same amount of time as that of nonprotected systems, and allowing a receiver to acquire synchronization at any point in a transmission. These features require additional cost and overhead due to the addition of accurate clocks and over-the-air time synchronization protocols, but the user is given a choice between cost and privacy by the multiple levels of linking protection.

Acknowledgments

This work was supported by the National Communications System (NCS) and the National Telecommunications and Information Administration/Institute for Telecommunication Sciences (NTIA/ITS).

Notes

1 *Federal Standard (FS) 1045, Telecommunications: HF Radio Automatic Link Establishment* (1990), GSA, Office of Information Resources Management.

2 Redding, C., and Johnson, E. E., "Linking Protection for HF Radio Automatic Link Establishment," *MILCOM '91*, pp 9.1.1-49.1.5 (Oct 1991).

A Programmer for 87C51 Microcontrollers

Single-chip microcomputers are great building blocks—if you can program them.

Sam Ulbing, N4UAU

The 8751 is a member of a family of microprocessors that are extremely popular these days and are readily available to the amateur builder. The devices in this family, the 8051 family, are 8-bit microprocessors that have been optimized for control applications. Variations of the basic 8051 microprocessor are found in a wide range of applications from medical instrumentation to automobile control systems—and ham applications. The main feature that makes 8051-based microprocessors so good for control applications is the ability to process only one bit of an 8-bit word (called boolean processing). An 8-bit processor normally processes data 8 bits at a time, so programmers must consider the state of all 8 bits even though they only want to control one of those bits. With the 8051 family, a single bit can be controlled and the other bits ignored.

Some projects are more easily implemented on a "regular" microcomputer, such as the 8085 system I described in a previous article.[1] Others are better suited to a single-chip solution like the 8051. I recently wanted to build a project making use of the features of the 8051 family. Rather than build an 8051 development system like I did with the 8085 computer, I took the lazy approach and bought one for about $100. It was similar to my 8085 system in layout, but it had a very poor operating system in its EPROM. Using the existing 8085 computer I was able to program a 2764 EPROM with a better and more flexible operating system for the 8051. This system was then fine for development and learning the 8051 system, but my real goal was to build a project with a specific member of the family, namely the 8751 microcontroller.

The 8751 offers many advantages to the designer over conventional microprocessors. The 8751 has all the components needed for a complete micro-controller system in a single 40-pin package. It has 128 bytes of RAM, 4 kbytes (or more) of ROM, up to 32 ports for I/O, 2 timers, 5 interrupt sources, a serial I/O port, a built-in UART and, of course, the CPU. For many projects, these features mean greatly simplified design and construction, as well as smaller size. I chose the 87C51 CMOS version for two reasons. First, it programs at 12.75 V instead of at 21 V. More important, it draws very little current and is adaptable to battery operation. In normal operation the 87C51 only draws around 12 mA. In addition, it has 2 power-down modes that can be implemented if desired. The idle mode reduces current draw to around 1 mA during times of inactivity but still maintains the clock and interrupt functions. The power-down mode reduces current draw to about 3 µA while preserving the contents of RAM. The 87C51 also has the efficiency typical of the entire 8051 family: one version can be operated with a 33-MHz crystal resulting in a machine cycle speed of almost 3 MHz. Instructions are normally only 2 bytes long due to a unique architecture that separates program and data addresses, and most instructions operate in 2 machine cycles.

A Circuit for Programming the 8751

To develop a project using this chip, I faced the same problem I had earlier with the 8085 system. I needed to program ROM. In this case, the ROM was quite different than the basic 27XX ROM since it was embedded in the 8751 40-pin chip. Fortunately, the modular design of the 8085 general-purpose computer makes it easy to build an 8751 programmer.

The 87C51 programming module is built on a Radio Shack 276-154 plug-in board. The schematic is shown in Fig 1. You will notice many similarities to the 2764 programming module described

[1] Ulbing, Sam, "An 8085-Based Computer System," November 1993 *QEX*, p 3, ARRL.

5305 NW 57th Lane
Gainesville, FL 32606

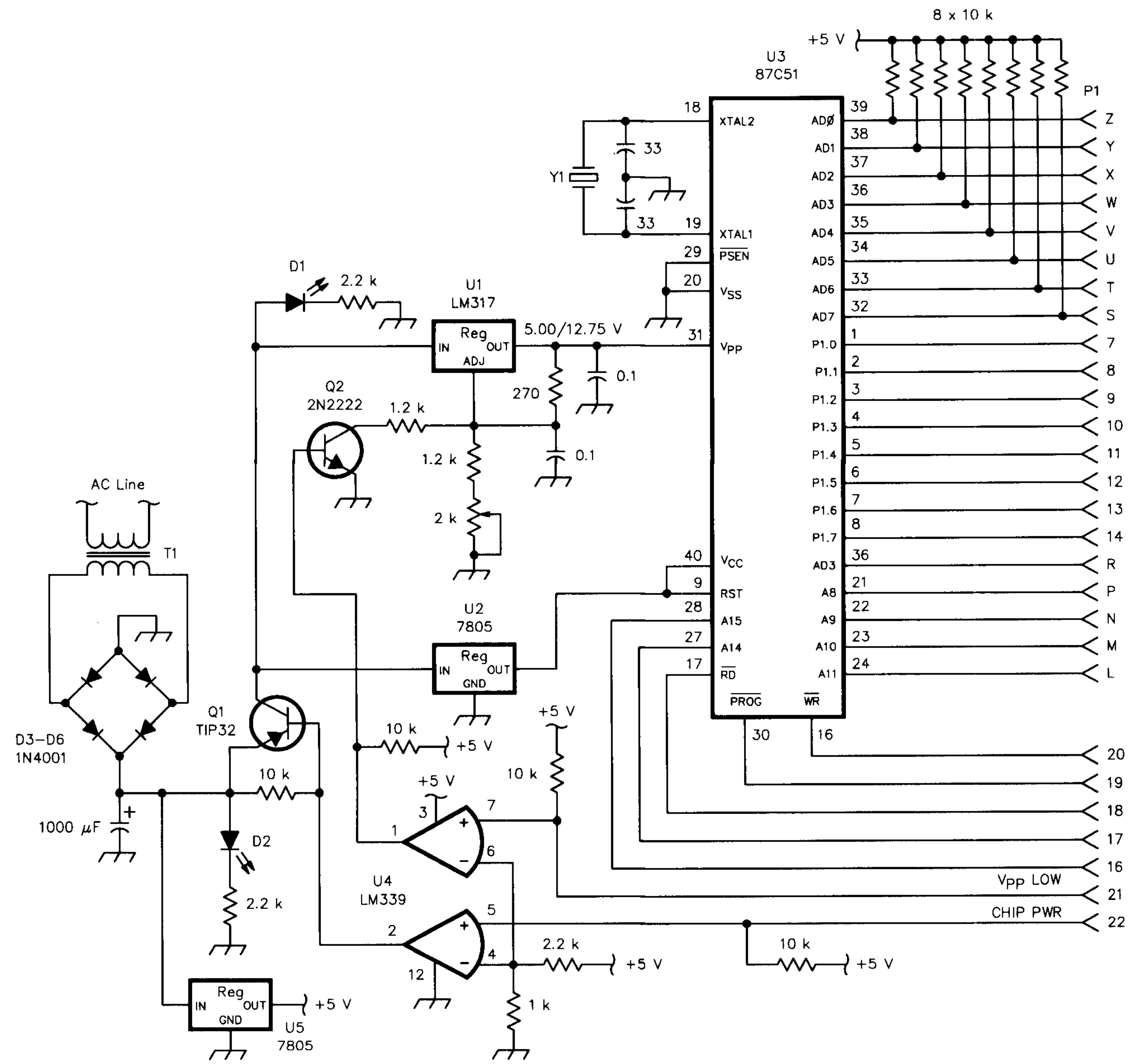


Fig 1—Schematic of the 87C51 programmer. Use a 10-turn potentiometer. Y1 is a 4- to 6-MHz crystal (see text).

in the previous article. There are also some significant differences; the most obvious difference is the fact that the 87C51 being programmed has a crystal connected, Y1. In order to program the 87C51, it must be running, and the oscillator (crystal) must operate at between 4 and 6 MHz. The reason that the micro's clock must be running is that it is actually executing internal address and program data transfers during the programming process. The value of the capacitors on the oscillator pins is not critical. I have used both 33 and 27 pF successfully.

20 pins of the 87C51 are used to enter the address and data to be programmed. You will notice that pins 32 to 39 have pull-up resistors. These pins do not have internal pull-up resistors. The V_{pp} level is controlled by changing the resistance in the feedback circuit of an LM317 adjustable voltage regulator. Two 7805s are used to control the 5-V power. One supplies power to the control circuitry and is on as soon as the power is applied to the module. The other applies 5 V to the 87C51. This is turned on only after the microcontroller has been socketed. Programming current demand is 50 mA or less.

The Software

The steps the program uses to program the 87C51 are:

1. Turn on power to chip.
2. Verify that the 87C51 ROM is blank.
3. Set the data, address and control levels.
4. Raise V_{pp} to 12.75 V.
5. Lower $\overline{PROG}$ to 0 V, delay 100 (plus or minus 10) µsec and raise it back again to 5 V.

6. Delay at least 10 µsec, then repeat steps 5 and 6, 24 times.
7. Lower V_{pp} to 5 V.
8. Increment the address and repeat steps 3 through 8 until all ROM locations are programmed.
8. Verify the entire ROM contents.
9. If desired, program the two security bits: set pins P2.6, P3.6, P3.7 to the appropriate level and repeat steps 4 through 7 for each security bit.
10. Verify that the data is secure.

The 8085 will program the data at addresses 2000 to 2FFF of the 8085 RAM into the 8751 ROM.

While this is a quick-pulse programming method, it is different from the one used in the 2764 programmer. For the 2764, the ROM was read after each programming pulse to determine if that byte of data was yet programmed. If it was, the program went on to the next byte of data. In the 87C51 quick-pulse method, no verification is made until all bytes are programmed, so all 25 programming pulses are used. The time to program the 87C51 is significantly longer than for the 2764, about 25 seconds in total. I have used the Signetics version of the 87C51 because I have found the company much more willing to talk to the "little experimenter" than Intel has been. The method above is the method described in the Signetics data book. The Intel data book describes a similar method using only 5 pulses and comments that 25 pulses can be used for compatibility. I have not tried the 5-pulse approach to see if it would work for the Signetics product. You will notice from the last two programming steps that the 87C51 offers the capability of security programming the chip. There are 2 bits to program. The first one will prevent writing to the chip; the second one will block any attempts to read the chip and will return an "FF" as the data for all addresses. Be aware that this is also the data for a blank chip. If you try to program this chip, the initial verify program will show that the chip is blank, but when you try to program it, the programming will fail. Be sure to erase the chip if you want to program it.

Where to Go from Here

If you find the 87C51 as exciting a product as I did and want to learn more, I recommend you contact the Philips/Signetics Literature Distribution Department and request a copy of their data book entitled *80C51-Based 8-Bit Microcontrollers*. This book describes their entire family of 80C51 micro controller. Their phone number is 800 447-1500 ext 3009.

I will be glad to provide a copy of my source code containing the above and all the programs described in the 8085 article on a 3.5" or 5.25" disk for $5. I can also provide a pre-programmed EPROM for $12. The charges are only to cover my costs. Or, you can download the file 8085SYS.ZIP from the ARRL BBS (203 666-0578) or via Internet from the ftp.cs.buffalo.edu system in the /pub/hamradio directory. □□

Digital Communications

Harold E. Price, NK6K

Audio Spectrum Analyzers - Cheap!

This month's column is about making do with what you have at hand. One of the things Amateur Radio can teach, or at least give you practice at, is cobbling up something quickly out of the "junk box," be it a hardware junk box or a software junk box. This month, I'll give an example, and point you toward a very good software junk box.

I needed to do a quick check on the center frequency of an audio filter recently. The filter in question was the NIR-10 from JPS Communications, Inc. This is a DSP-based unit, with, among other things, a variable-size bandpass filter with a variable center frequency. I was experimenting with this filter in conjunction with a PSK modem. The NIR-10's adjustment knob is not calibrated though, and I needed to set the filter center to 1600 Hz. As usual, the project was late and I needed to do it quickly.

The NK6K lab contains many things, but it does not contain an audio spectrum analyzer. What to do? I had recently upgraded one of my computers with a *Sound Blaster* card and CD-ROM. When my two-and-a-half year old isn't playing *Putt-Putt Joins the Parade*, I get to use them. I remembered that the Sound Blaster has an input port and an analog-to-digital converter (ADC), because my next door neighbor's kid had used one to digitize and play all seven minutes of "Stairway to Heaven" backwards looking for the infamous "power is Satan." (It's there, sort of.)

I could now digitize the audio output of the NIR-10. I also knew that with DSP and a Fast Fourier Transform (FFT) algorithm, I could plot the amount of energy present at various frequencies. You don't need to know much more than that about the FFT to use it; it is simply a technique that transforms signals in the time domain (amplitude vs time, or what you see on an oscilloscope) to the frequency domain (amplitude vs frequency). While coding up a quick FFT program isn't something I can do off the top of my head, it is a wheel that has already been invented. Here is where having a well-stocked software junk box comes in handy. My software junk box contains a CD-ROM of the Simtel20 MSDOS archive. This is a collection of 9000 DOS programs, many with source, that do various functions. ("Simtel20" was the name of the computer system on the Internet that first housed the MSDOS archive.) A quick scan showed a file with the likely looking name of FFT142.ZIP. Sure enough, it contained a program written by James R. Van Zandt that reads in a set of values from a file and outputs frequency buckets.

Fifteen minutes later, I had written a program to convert the .WAV audio capture file that the *Windows* sound recorder applet uses into an ASCII file that FFT142 wanted. An eight-bit .WAV file turns out, by inspection, to contain a header that I could ignore, and then the sampled audio, one byte per sample. I set the sound recorder options to 11-kHz sample rate, 8-bit mode, and mono input. Using *Excel* to plot FFT142 output resulted in a graph like the one in Fig 1. Job complete. Using an $80 Sound Blaster card, and the $39 Simtel Archives CD-ROM, I'd avoided cluttering up the shack with a swap meet dinosaur spectrum analyzer, or emptying my wallet with the top of the line $20,000 model.

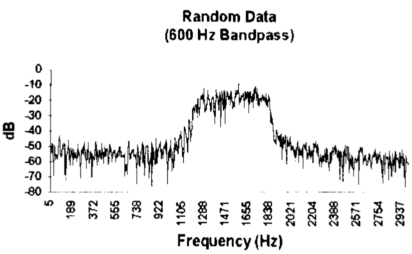


Fig 1—Manual plotting of an FFT output with *Excel*.

5949 Pudding Stone Lane
Bethel Park, PA 15102
email: nk6k@amsat.org (Internet)
71635,1174 (CompuServe)

The Simtel programs are available directly from the Internet, of course. The latest versions of programs are always found there. The CD-ROM format is handy for people without Internet access, or who have to pay for access, long distance phone charges, or are stuck with 2400-baud dialup lines. The CD-ROM is available from Walnut Creek CDROM (510-674-0783). Many of these programs are shareware, some are freeware.

Stepping Up From "Junk"

My first solution required storing the data in a file, converting it, and massaging the output data with *Excel*. It was obvious that one could write a program to do a real-time spectrum analyzer. Surely someone already had. After a bit of looking, I came up with three real-time programs, one shareware, one payware but cheap, and one payware for a little more, but of knock-your-socks-off quality. All of these programs will allow you to sample at rates up to 22 kHz, which means frequencies up to 11 kHz can be displayed. The Pioneer Hill software described below will sample at rates up to 44 kHz, for displays of 22 kHz.

Good

The simplest program is *micFFT* by Craig M. Walsh, and ferreted out for me by Skip Hansen, WB6YMH. It is available on Compuserve (GO MEDIAVISION, DL 12). This is a DOS program that works with Covox, ProAudio, and Sound Blaster cards. It provides a bouncing bar display as in Fig 2. The program works with real-time audio, and does not store the audio or the data. It has the fastest real-time display, and samples up to 22 kHz.

Better

I found two programs that work with

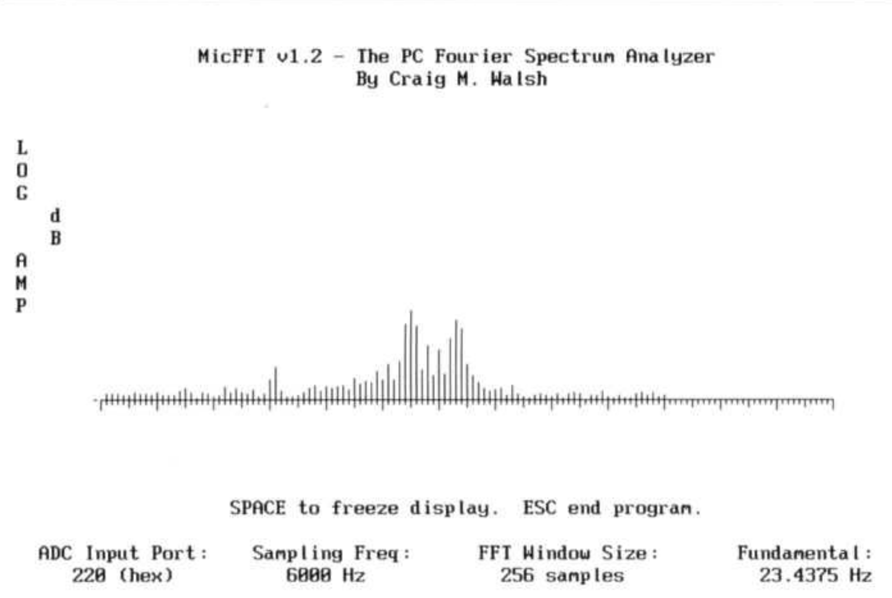


Fig 2—Real-time display from *micFFT.*

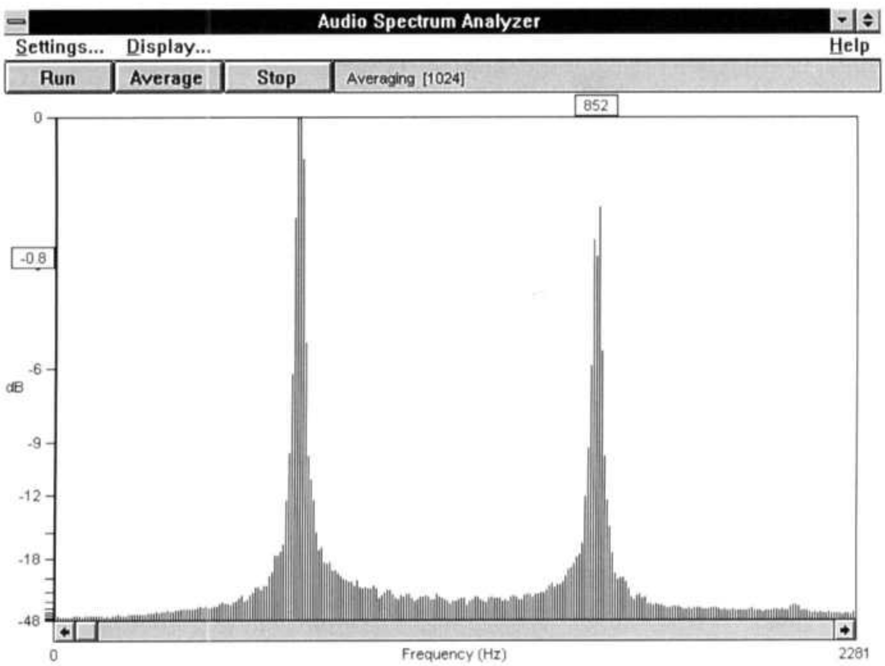


Fig 3—Real-time display from *Spectrum.*

Windows. The advantage of *Windows* programs is that they work with a virtual device interface for the sound card, making it easier for the program to support many different kinds of audio cards. It is also easier to support graphics displays and mice. The simplest program I found is the $39.00 Audio Spectrum Analyzer program (*Spectrum*) from Pioneer Hill Software (206-697-3472). The program displays a bouncing bar graph as shown in Fig 3. Up to three markers can be placed on the display at user specified frequencies. It only works with real-time audio, as does *micFTT*, but it has an easier to use interface to adjust the sampling rate, the FFT size (resolution), averaging, and display size. The mouse cursor can be used to display the frequency represented by a bar, or it can show the frequency difference between two points. Fig 3 shows an 850-Hz wide RTTY signal.

Best

On the high end of capability is *Spectra Plus*, also from Pioneer Hill Software. In addition to a spectrum display similar to that in *Spectrum*, *Spectra Plus* will display a time-domain view, a spectrogram view, and a 3-D surface view. It also allows you to record the audio in a .WAV file and post-process it using any of the above display types. This allows a high-resolution display of 44.4-kHz data with an FFT block size of 2048. Each display is in its own window pane, which can be individually sized and moved within the main *Spectrum Plus* window.

The time-domain view looks like a digital oscilloscope. It provides a cheap storage scope for audio bandwidths. It allowed me to easily measure the keyup and unkey times on the recently resurrected DOVE spacecraft's two-meter transmitter, simply by capturing the audio and using the cursor to measure the time between two points.

The spectrogram view is perhaps the most useful. It draws one line across the screen for each time sample, from low to high frequency, using color to show amplitude. This display shows the frequency domain over time, allowing you to look more easily at changes in the signal or filters as you make adjustments. As shown in Fig 4, it also allows you to see frequency drift. Fig 4 shows two CW stations in QSO on the OSCAR-13 spacecraft. One signal is increasing in frequency, the other is decreasing, caused by Doppler shift as the spacecraft approaches one user and recedes from the other. Overlaid on the Doppler seen by the spacecraft receiver is the Doppler on the spacecraft transmitter as seen by my receiver on the ground. This display shows visually the type of information that is used by the SARSAT satellite system to locate ELT beacons in downed aircraft.

The 3-D surface views give the same information as the spectrogram view, except that the display is tilted 45 degrees in 3-D. Time marches off into the distance, and frequency amplitude is shown by the height of little virtual mountains rising from the noise floor.

The package has many options, including marker displays, averaging, scaling, and smoothing. All of this capa-

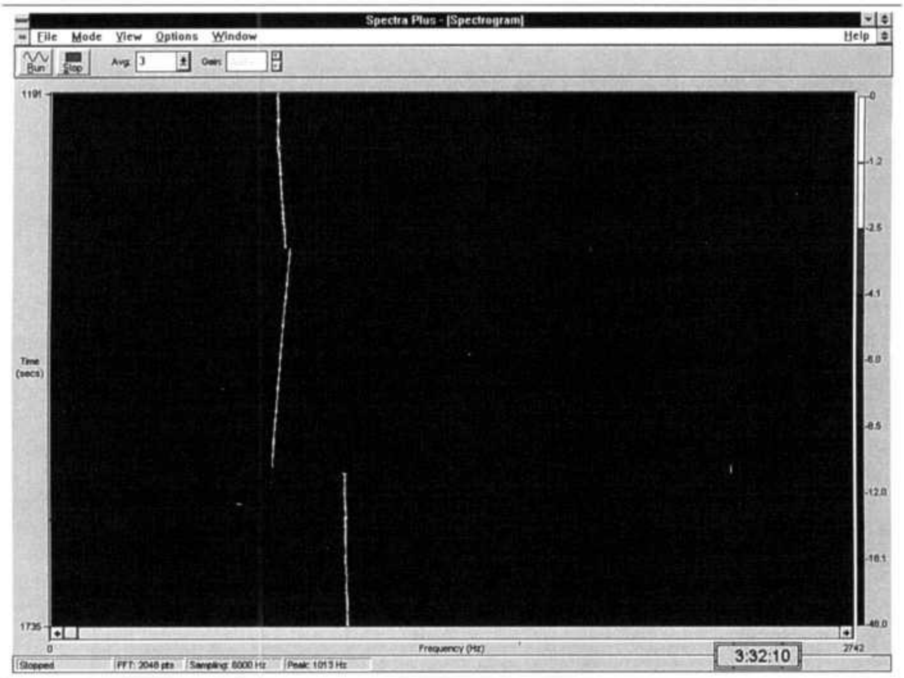


Fig 4—Doppler shifts with *Spectra Plus.*

bility comes with a higher price tag. *Spectra Plus* is $129 until the end of December 1993, $179 after that.

Why?

There are many other uses for an audio spectrum analyzer in the ham shack. The South Texas Balloon Launch Team's BLT-8 telemetry package uses audio tones to send telemetry for inside temperature, outside temperature, and atmospheric pressure (from which altitude can be derived). Any of the spectrum analyzers discussed above can be used to determine the frequency of these tones. I'm sure many of you have needed an audio spectrum analyzer from time to time, but the need was never enough to cause you to buy one. Assuming you already have a PC and an audio card, I've shown several solutions ranging from no money to $200 that will add this tool to your shack.

Forward Error Correction II - The Preview

In the "Digital Communications" column for June 1993, Phil Karn and I discussed forward error correction. Since then, I've implemented an actual system based on the Phil's design. The software uses a DRSI card at 1200 baud to send and receive frames with convolutional coding. Tests made by Jon Bloom at the ARRL lab show that coding adds at least 4 dB to the performance of the system in a white noise environment. The test was informal, and measured the difference in the lowest SNR that resulted in 10 frames received in a row with no errors (no FEC) vs the threshold at which no frames were received (with FEC). This compares the SNR required to receive a raw HDLC frame to the SNR required to received the coded frame.

At the low SNR end of the test, 20 to 40 errors out of 1280 bits were being corrected in the time it took to receive the frame. This shows that the modem with coding was performing well at a very high bit error rate. The test was run using a 12-MHz 80286 processor. Since the sequential decoder described by Phil is sensitive to the processor speed, more errors can be corrected on a faster processor. On a 486, frames with 50 to 70 bits are routinely corrected. The data throughput performance showed an infinite improvement, of course, since the modem with coding was receiving 100 percent of the frames while a modem without coding would have received 0 percent. In the next column, if all goes as planned, I hope to discuss the implementation in detail, with source code. With Jon's help, I'll also present more formal performance specs.

Correction for GRAPES Address

Bob Merritt wrote to say that the GRAPES address for the WA4DSY 56kb modem kits has changed. The current information is: GRAPES, Inc, PO Box 636, Griffin, GA 30224, email: ka4byp@kd4nc.atl.ga.us □□

Book Review

McGraw-Hill Circuit Encyclopedia and Troubleshooting Guide, Volume I, ISBN 0-07-037603-4, by John D. Lenk. Copyright 1993, McGraw-Hill, Inc., 1221 Avenue of the Americas, New York, NY 10020. Tel: 800 262-4729.

Reviewed by Ed Hare, KA1CV, ARRL Laboratory Supervisor

When I was first asked to do a review of this book, I thought I had an easy task on my hands. Although I keep a collection of circuit books at hand, they are all quite alike: reprints of sometimes incomplete schematics that leave me needing just a little bit more information. My 1980 copy of the *Modern Electronic Circuits Reference Manual*, by John Markus, (McGraw-Hill, ISBN 0-07-040446-1) is getting a bit dog-eared, so when I saw this new book by the same publisher, I was glad to be getting a more modern replacement.

Unlike other examples of the genre, this book is more than just a collection of circuits. The first sentences in the introduction tell you that: "When you have finished this encyclopedia, you should be able to recognize well over 700 circuits that are commonly used in all phases of electronics. You will also understand how the circuits operate and where they fit into electronic equipment and systems." While the book does not always meet these goals, it does contain more descriptions of circuit operation than any similar book I've seen.

The book starts off with a useful tool: the company addresses for all of the semiconductor manufacturers who contributed circuits from their data books and applications notes. (I would have liked to see telephone numbers, too, but I am an impatient fellow.) It appears that *all* of the circuits have been taken from data books and application notes. I am used to seeing *QST* and the *ARRL Handbook* throughout circuit books; this one is a bit less "hammy" than most.

Following the addresses are several pages of reference data. A section on "Substitution and Cross-Reference Tables" is useful, listing direct replacements for operational amplifiers, some data-conversion ICs and voltage regulators. On the other hand, the section on "IC Packages and Pin Connections" occupies only two pages. There's not a wealth of information there!

Starting with Chapter 1, "Audio, Ultrasonic and Direct-Current Circuits," you can see why the title includes the phrase "Troubleshooting Guide." Each chapter starts out with a theoretical section, presented as a testing and troubleshooting tutorial. The theoretical information is far from complete, though; you will not find the formula for Ohm's law anywhere! The circuits section of the chapter contains some useful circuits, but has too many phonograph preamplifiers and simple stereo circuits. More diversity in circuit selection would be welcome. If you are expecting to see a lot of modern devices, you sure won't find them in this chapter. Almost all of the circuits are made from either discrete devices or simple op-amp ICs. And most of the audio power amplifiers use discrete transistors.

The Circuit Encyclopedia contains more description of circuit operation than any similar book I've seen.

The RF and IF chapter proves more interesting, containing 86 pages of theory and circuits. Again, the theory section of the chapter is presented from a testing and troubleshooting point of view. This is the first time I have ever seen the concept of L-C resonance introduced as part of a discussion of circuit testing! The six pages devoted to RF troubleshooting only scratch the surface, but you will find a few solid tips.

Almost half of the RF circuits are derived from GEC Plessey Semiconductors data books. These GEC Plessey circuits demonstrate the use of modern ICs in design applications, but GEC Plessey devices are difficult to buy in small quantities. This makes these circuits of limited use to the experimenter. Many of the other circuits are from the *Motorola RF Device Data* book, with only a few circuits from other manufacturers. The reproduction of the Motorola data includes parts lists, printed-circuit board artwork and component-placement diagrams.

I'll cover the remaining chapters briefly. They each are organized in the same way: a troubleshooting section followed by a circuits section.

"Power-Supply and Regulator Circuits": The troubleshooting section gives a pretty good description of the testing of power supplies and the factors that affect those tests. The circuits show some of the common "cookbook" circuits, although the three-terminal regulator is conspicuously missing. (That's not necessarily a bad thing—these three-terminal devices are pretty well known.) More interesting are some of the dc-to-dc converter circuits and switching regulators.

"Oscillator and Generator Circuits": While there is a test and troubleshooting section, five pages is only enough to scratch the surface. Most types of "cookbook" audio and clock circuits are covered, primarily using op amps. It is disappointing not to see any "hammy" RF variable-frequency oscillators or phase-locked loop circuits, although I did find two simple versions of the latter in a subsequent chapter.

"Digital Circuits": This chapter contains more than the simple cookbook designs that might be expected.

There were lots of analog-to-digital and digital-to-analog circuits, a simple circuit for part of an EPROM programmer (sadly, not a complete project) and even a bit of DSP interface. And there were a few digital specialty circuits I have not seen elsewhere, such as a digital/analog circuit for taking the square root of an input voltage.

"Filter Circuits": Almost all of the circuits describe op-amp audio active filters. While it was nice to see all of these circuit permutations in one place, along with some useful descriptions of the associated design theory, more RF circuitry would have been nice. The only RF filter was a 4.5-MHz notch filter.

"Op-Amp Circuits": With so many op-amp circuits peppered throughout the book, it at first seems odd to have a chapter like this. It makes sense, though, as the troubleshooting section is tailored toward the testing of op amps, with the theory of operation and key parameters such as input-bias current, offset voltage, slew rate and common-mode rejection covered pretty well. And there are some interesting circuits, too, such as a microvolt comparator, precision current source and sink, and a linear thermometer. The more common "cookbook" op-amp circuits are found here as well.

There are other chapters on video circuits, UJT circuits, Norton amplifier circuits and the usual "Special-Purpose Circuits" chapter. The latter is a collection of circuits that don't fit anywhere else, ranging from the popular "fire siren" and thermometer circuits to a few music circuits. And, of course, the "Morse-Code Set."

As a circuits book, this volume deserves only a qualified endorsement. It is refreshing to see some discussion of each of the circuits, and some of the testing and troubleshooting information is quite good. On the minus side, at least from an amateur radio point of view, limiting the sources of circuits to semiconductor manufacturer data books and application notes limits the number of RF circuits included. So, I don't think I will retire my old standby, the Markus circuits book, just yet. □□

1993 Index

March 1993-December 1993 QEX (Volumes 133-142)

Correspondence

Digital Communications (Price)

Editorials (Bloom)

Miscellaneous

Miscellaneous Technical

RF (Lau)

Specialized Communications Techniques

Upcoming Technical Conferences

Want to reap the rewards of teaching ham radio? Ask ARRL Educational Activities Department for *your* free package on teaching youth or adults.