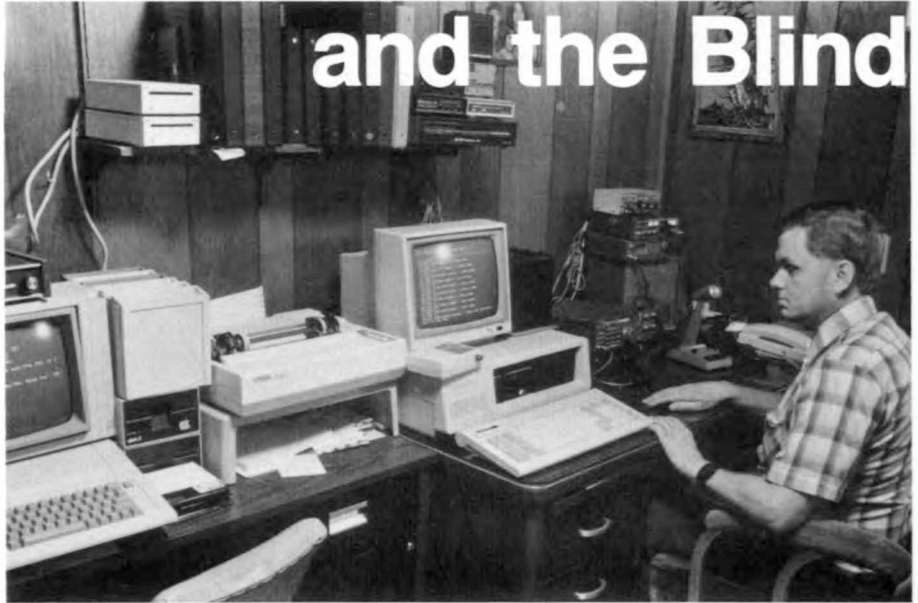# Amateur Radio and the Blind

Part 2: Computers, speech synthesizers, software and modems—what's involved in making them work together? Let's begin the learning process.[†]

By Butch Bussen, WA0VJR
Box 142,
Wallace, KS 67761

(Photos by Gary Mc Duffie, AG0N)

For the blind ham interested in digital communications, there are four important items that all must work together, not counting the radio and antenna: the software, computer, speech synthesizer and modem. Because all of these items are so closely linked, it's difficult to discuss them as entirely separate entities.

If you presently don't have a computer, speech synthesizer, software or modem, there are several things you need to think about before you get any or all of them. Is your system going to be dedicated exclusively to Amateur Radio use? If not, what else do you want to use it for? When people ask me which computer to buy, I tell them to first pick the software they want to use, *then* look for systems that will run it. Dealer service, assistance through computer clubs and help from fellow amateurs are all important.

EQUIPMENT AND SOFTWARE

In preparing this series of articles, I spent many hours on the air using four modems, three computers, several pieces of software and three speech synthesizers. Advanced Electronics Applications (AEA) and Kantronics helped with this project.[6,7] A PK-232 was provided by AEA, and Kantronics loaned me a KPC-2. Both modems work well and, as you'll see in a subsequent installment, there are certain parameters that become very important when using these boxes with a speech

†Part 1 appeared in the Oct 1987 issue of *QST*.
6Notes appear on page 32.

synthesizer. I also used the Kantronics UTU and AEA AMT-1 on AMTOR.

The computers I employed included an IBM® PC compatible, (I'll use the term PC to represent the IBM PC or PC-compatible computers), an Apple® //e and the Laser 128. Chances are, the first two computer types are probably familiar to many of you, but you may not have heard of the Laser, so a quick briefing is in order.

The Laser 128

The Laser 128 is an Apple //c clone that sells for about $400, is 99% Apple //c compatible and runs all of the talking software I have tried on it. The Laser is supplied with 128 kbytes of RAM and a single, built-in disk drive. Though the Apple //c is equipped with two I/O ports, the Laser has actually three I/O ports. On the Laser, I/O port 1 can be configured as a parallel or serial port simply by moving a switch. Parallel output data is then routed to a DB-15 connector. Port 1 can be used only with a printer, as it is not software programmable for use as a serial communications port.

A little wider than the //c, the Laser is equipped with an expansion port on the left-hand side; the //c has no such expansion port. An optional two-slot expansion chassis—equipped with its own power supply—can be plugged into this expansion port.

Other Tools

Among the terminal programs I've used are Talking Transend™ and PC-Talk (a shareware program available on many computer BBSs).[8] I also ran a couple of communications programs I wrote for the

Apple and PCs. COMM.BAS, a simple communications program included on the IBM DOS Supplemental Programs disk, is another that works successfully.

With the Apple and Laser computers, I used the Echo II and Echo Plus speech synthesizers; an Echo PC speech synthesizer was employed with the PC. All of these synthesizers are manufactured by Street Electronics.[9]

SPEAK TO ME...

Unless you can get the information you need *out* of the computer, it won't do you much good. We visually handicapped people are much more limited in our choice of equipment than our sighted friends. People with sight are used to looking at a video display whenever they need to gather information. Since I have never had vision, I cannot really understand what it's like to quickly scan a screen for the desired information. On the other hand, I'm sure sighted persons can't really know what it's like to have to depend *totally* on a speech synthesizer.

From the time I first learned about computers, I wanted one that would talk. Three years ago, I got one—and my life will never be the same! I'd always wanted to try my hand at RTTY, but the stumbling block I had to overcome was how to *read* what the other station was sending. I learned touch typing in the seventh grade, so using a keyboard to *transmit* was easy. I tried reading the perforated RTTY tape, but that wasn't possible for me. Also, I'm not much of a mechanic and wasn't sure I could keep one of the early "mechanical monsters" running, anyway.

I had a computer, a speech synthesizer

and some special talking software. I figured there should be a way to put everything together and get on the air with RTTY and AMTOR. Well, that turned out the way my repeater project did: The transmitter and receiver worked fine when operated separately, but when I tried to operate them as a pair, the "fun" started! I found that putting together all the hardware and software to run RTTY was the same kind of story. With the information presented in this series, however, your road to digital communications should be much smoother.

SPEECH SYNTHESIZERS

The choice of a speech synthesizer will depend to an extent on which computer or software you decide to use. There are, however, some general points about synthesizers that need a little explanation. The first thing to realize is that speech synthesizers were, in most cases, not designed for use by the blind. We are a very small part of a manufacturer's market. Speech synthesizers are usually considered a novelty or a toy, or maybe an educational tool. The term "talking computer" doesn't necessarily imply "usable by a blind person." Only by using specialized software and hardware am I able to put synthesizers to work for me.

Internal Synthesizers

There are two basic types of synthesizers: internal and external. The internally connected synthesizer plugs into a computer's expansion slot or game port. Such synthesizers are designed for use with a particular computer type, and software that will work with the synthesizer may or may not be available.

Prices for plug-in units start at about $150 for the Echo Plus (see Fig 3). As with most things, the sky is the limit. One speech synthesizer designed by Texas Instruments for use with IBM PC systems has fabulous speech quality. It sounds almost like a real person speaking and will even answer the phone for you! The synthesizer digitally records a message someone may want to leave. It responds to standard Touch Tone® frequencies to select menu choices or answer questions. With a little clever programming, you can call and get information from your computer using a standard Touch Tone phone. This Texas Instrument synthesizer sells for around $2000.

External Synthesizers

The second type of synthesizer is external to the computer. These synthesizers have their own CPUs as well as other firmware that do text-to-speech conversion and so on. You can think of this type of speech synthesizer as a "talking printer." Instead of printing what you send it, the synthesizer speaks it. As with printers, you need to select a parallel or serial data format. (Some of these synthesizers are equipped
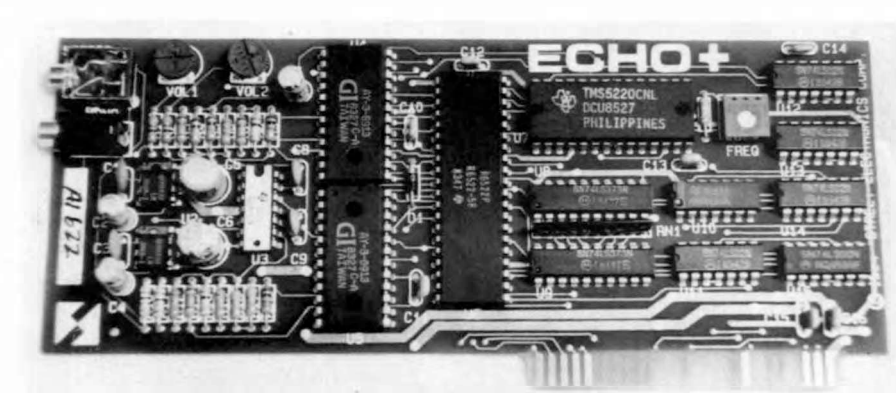

Fig 3—The Echo+ plug-in speech synthesizer for the Apple or Apple-compatible computer. This synthesizer requires text-to-speech software to be loaded into the computer's RAM prior to use.

with both parallel and serial input ports.) You send text strings to the synthesizer, and it voices them. As a rule, most synthesizers need to see a carriage-return character before they will speak the last string sent to them. Most synthesizers have a buffered input. Others have built-in sound-effects generators, clocks and filters you can shape through software. Prices for these "talkers" start at about $250 for the Echo GP (Fig 4).

Of the inexpensive models, the Echo GP is my favorite. I have used it on the IBM PC and PC compatibles as well as a C64. Among the better stand-alone synthesizers is the Dec-Talk, manufactured by Digital Equipment Corporation.[10] This one has many nice features and the speech is superb. The price is around $3000. No, I don't own one, but I'd love to!

The Echo synthesizer won't work when plugged directly into the Laser's expansion slot because there is no 5-V supply source at this connector. So, if you're going to use a Laser, you'll also need the optional expansion chassis. It costs $80 or less, depending on where you buy it. This box plugs into the expansion slot on the side of the computer and contains a built-in, ac-operated power supply that provides the voltage required by the Echo synthesizer. Because the expansion chassis is equipped with two slots, you can use the second one for memory expansion or some other feature.

The Apple //c has no expansion slot for the Echo card. Another synthesizer manufactured by Street Electronics (called the Cricket™) can make the Apple //c talk. The Cricket is designed to be connected to port 2 of the //c. You cannot, however, connect a Cricket to port 2 of a Laser because that port is required for serial communication with your RTTY/packet modem. A lot of existing software is not presently compatible with the Cricket, though that situation is changing.

There are some features to look for when

contemplating the purchase of a synthesizer. The Echo GP is small and designed for serial input only. This unit is supplied with a serial cable equipped with a *male* connector (DB-25P); it plugs into the serial port on my Apple //e. (It can be used with a PC, too, but requires a gender changer.) The Echo PC is basically the same as the Echo GP, but is designed specifically for use with IBM PCs or compatibles. An Echo PC is supplied with a cable and *female* (DB-25S) connector wired to fit the PC's male serial port connector.

Speech Quality

The inexpensive synthesizer models all have speech characteristics of their own and take a little getting used to. The first time I heard an Echo synthesizer, I wondered how I would ever understand it! It didn't take long, however. Like most things, it just takes time and practice. I now run the Echo as fast as it will go, and that isn't fast enough for me.

If you heard an Echo GP or Echo PC talk, you'd find its speech quality is not as clear as the synthesizers you attach to, or may have in, your radios. Why? Because the synthesizers used in radios (as well as those used in talking clocks and calculators) have a very limited vocabulary; they are designed to say a *few* words and numbers very clearly. The digital equivalent for each number or word is programmed into these devices. On the other hand, general-purpose synthesizers such as the Echo GP, must pronounce *thousands* of words and phrases. If you digitally represented each word, you'd use up a lot of memory quickly.

So how does a synthesizer like the Echo GP work? It looks at each text string, including spaces and punctuation. It then uses over 400 pronunciation rules to do its best to pronounce each word. The synthesizer even changes intonation for punctuation such as periods and question marks, and it pauses for commas. The

Fig 4—The Echo GP external speech synthesizer. This unit has a built-in text-to-speech converter and is driven by the computer through a serial communications port.

pronunciation isn't perfect, but it's certainly adequate. For the price, it does a good job.

I prefer the Echo GP because its voice is much like the first synthesizer I owned, and the command structure is much the same. An Echo synthesizer recognizes and voices punctuation characters. If you send it a ".", it will say "period." Some synthesizers totally ignore punctuation. If you want them to pronounce a ".", you have to spell it out for them by sending a "period."

SYNTHESIZER SOFTWARE

For the PC

Most of us who have IBM PCs or compatibles have a special program to make these machines talk. There are several such programs on the market ranging in price from $200 to $800 or so. The one I use is called Screen-Talk.Pro, produced by Computer Aids in Fort Wayne, Indiana.[11] This is a TSR (terminate, stay resident) program that hides itself in RAM and interfaces your machine to an external speech synthesizer. Screen-Talk.Pro provides you with capabilities similar to those you'd have using the Apple with the Echo Plus and Textalker software (that will be discussed in more detail later). Using Screen-Talk.Pro with ProKey® allows you to create some "hot keys" (macros) that really add power and flexibility.[12] You can also set up windows so that only characters printed within these windows are spoken by the synthesizer. This software gives you LINE REVIEW (the ability to reread a screen line),

punctuation control and a SILENCE KEY.

I used Screen-Talk.Pro with the three terminal programs I ran on my PC. Screen-Talk.Pro, or a similar program, will let you use much of the commercial software available for these machines, as well as some specially written software. Some of the standard commercial software talks poorly (in some cases, not at all) because it doesn't use standard DOS I/O routines to print to the screen.

Other Alternatives

If you don't have a talking program (like Screen-Talk.Pro) for your PC, there are other alternatives, such as the COMM.BAS program supplied on the IBM DOS Supplemental Programs disk, or a shareware BASIC communications program, or one you write yourself.

Of course, you'll need an external speech synthesizer. If your synthesizer requires parallel input, you can skip steps 2 and 3 that follow; the specific commands sent to the synthesizer will probably be different as well.

First, be sure you have your synthesizer hooked up properly. I have my synthesizer plugged into the computer's COM2 port. (I use COM1 to interface the computer to the modem.) Then set the synthesizer's switches appropriately. I set my Echo PC for 9600 bauds. You'll find four DIP switches on the bottom and near the back of the Echo PC. Push all four switches down toward the back of the box for 9600-baud operation. The Echo PC requires 8 data bits and one stop bit. When you turn the synthesizer on, you should

hear "ECHO READY." Now you're ready to make your computer talk. In the following steps, the command characters and keypresses you enter appear between the lesser-than (<) and greater-than (>) symbols. *Do not key in these symbols.*

1) Boot your system with your DOS disk. You'll probably be prompted for date and time, so press < Return > a couple times.

2) Type < MODE COM2:9600,N,8,1 >. Then press the < Return > key. This sets up serial port COM2 to talk to the Echo PC.

3) Next type < MODE LPT1: = COM2 > and press < Return >. This directs all output that would normally be routed to the printer to COM2 and on to the synthesizer.

4) To make sure everything is working so far, hold down one of the < Shift > keys and simultaneously press the < PrtSc > key. You should hear your synthesizer speak all the information that is on the screen.

5) Next, insert a disk that has BASICA and COMM.BAS on it. Now type < BASICA > followed by < Return >. Once BASICA is loaded, you can again do a screen dump by simultaneously pressing the < Shift > and < PrtSc > keys as before.

6) Press and hold down the < Ctrl > key, then press the < PrtSc > key. This sends characters as they are typed to the printer, which is usually connected to LPT1. In this case, we are using a synthesizer instead of a printer, and with the MODE command issued in step 3, we have rerouted everything to COM2 and on to the synthesizer.

7) Now we are going to send the Echo PC some specific commands. Be sure you turn < Caps Lock > on, or press the < Shift > key as the letters < C >, < D >, and < M > are typed. These letters must be *uppercase* to be recognized by the Echo PC. Type < PRINT CHR$(5);"1D" > and press < Return >. This tells the Echo PC to speak each character as it's sent. Next, type < PRINT CHR$(5);"C" > and press < Return >. This command makes the Echo speak faster. Type < PRINT CHR$(5);"M" > and press < Return >. This sets the Echo up for most punctuation—that is, most punctuation is pronounced. CHR$(5) is Control-E, which is a special command character for the Echo PC. These three commands are optional and will be different if you're using a different synthesizer. Some synthesizers don't have the ability to speak a single character without receiving a carriage return.

The < Ctrl-E"1D" > command is a bit confusing. The number can be anything from 0 through 9. Think of the < D > as a delay. This tells the speech synthesizer to speak the characters sent to it after a specified length of time even if it doesn't see a carriage return. A < 0 > cancels this command, and is the default. The Echo will

not say anything until it sees a carriage return. In other words, you'll not hear what you have typed until you press the <Return> key. If data is coming in a little slowly as it sometimes does on AMTOR, the synthesizer may speak parts of words because it says what is in its buffer without seeing a carriage return. In this case, you might want to lengthen the delay, or enter a <0> to turn it off altogether.

8) Now type <LOAD"COMM",R> and press <Return>. This BASIC program will be loaded and run. You'll hear it talk. Type a <6> from the menu and you'll be asked for the baud rate. Type <300> and press <Return>. To set parity, type <E> and <Return>. Set the number of bits per character by typing <7> and <Return>. To set the number of stop bits, type <1> and <Return>. Type <N> and then <Return> to omit echoing characters to the screen. If you've entered all the data correctly, type a <Y> and press <Return> when prompted. You can now talk and listen to messages from the KPC-2 or PK-232 (more on this later). Remember, you'll not have a SCREEN REVIEW as such, but you can dump the entire screen any time you want using the SHIFT and PRTSC keys. Neither do you have the use of a SILENCE key as the Control X isn't usually echoed back to the speech synthesizer.

For the Apple

My first talking computer system was an Apple //e with an Echo II synthesizer plugged into one of the internal expansion slots. (The Echo II is the predecessor of the Echo Plus mentioned earlier.) The Echo II becomes part of the operating system and requires special software (called Textalker) loaded from disk to make it talk. Textalker—a program for Apple computers only—is supplied with the Echo synthesizers. This software provides many special features for the blind. There are many general applications programs that make use of the abilities of the Echo II or Echo Plus.

With Textalker, you hear the character representation of each key as it's typed. You can select a slow or fast rate of speech. You've a choice of three levels of punctuation announcement. Your selection determines which punctuation marks the Echo will voice. If you select ALL, you hear spaces, line feeds, carriage returns; in short, everything. The MOST choice will give you the percent sign, question mark, plus sign, period and a few more. Two more commands included are the SILENCE key and LINE REVIEW. A Control X will momentarily silence the synthesizer until the next key stroke. If you are bored by hearing the same menu voiced over and over, just press CONTROL X; that tells your speech synthesizer to shut up. Not all synthesizers have this feature.

The LINE REVIEW command is one of the more useful. It allows you to enter an audio mode while running a program, or when the computer is in the immediate mode (no program running). LINE REVIEW gives you a sort of audio cursor that you can move around the screen. You can reread anything from a single line to the contents of the entire screen. You can switch from words to letters mode and use the arrow keys to check spelling of a word. A tap of the ESCAPE key will put you back at the point you were when you entered LINE REVIEW.

More Software

Other manufacturers are writing similar software to support their particular synthesizers. Some commercially written programs for the visually handicapped ask you which synthesizer you have and act accordingly. For Apple computers, I think there are more programs written for the Echo Plus or Echo II synthesizers than any other types. There are other synthesizers and more software just around the corner. Don't be afraid to ask questions and get second opinions. It is better to find out *before* you shell out your hard-earned dough what programs you can run with a particular synthesizer.

TALKING SOFTWARE PERFORMANCE

Though I find the PC's DOS powerful, I personally don't find the interaction between the computer and the synthesizer to be quite as smooth as I do the combination I use with my Apple. One big difference I notice between the two synthesizer and computer combinations is the way the screen and the synthesizer interact. Most of the external synthesizers are buffered so the computer is usually far ahead of the speech synthesizer. Some of these synthesizers have no provision to dump the buffer—no SILENCE command. If you load a BASIC program on the Apple and then LIST it and watch the screen, you see that the listing keeps pace with the synthesizer; it scrolls only as fast as the Echo is talking. This is a sort of real-time operation. If you press CONTROL X to silence the speech synthesizer, the lines will scroll at normal speed. There are advantages and disadvantages to using each method. In the real-time mode, you hear things exactly as they are occurring. This way, you get sound effects and prompts at the proper time. In this mode, the synthesizer may not talk nearly as fast as you can type. If you are using a word processor or typing a message to someone on the air, you would want to be able to type ahead of the voice, or not have the keys speak at all.

One example of why I mention all of this is the 300-baud versus 1200-baud rates used in data transfer. If you are just transferring files (without the need for speech output), 1200-baud operation is great. If, however, you are using a bulletin board and want to read all that is happening, while it's happening, there is no way the speech synthesizer can keep up with 1200-baud operation. It's like trying to have someone read to you what is on the screen with the data coming in at 1200 bauds and no way to buffer the data or temporarily stop it.

APPLICATIONS SOFTWARE PROBLEMS

Earlier I said the Echo II and Echo Plus required special software to be loaded from disk in order for them to make the Apple or Laser talk. Now, if the applications software you want to run (such as an RTTY, CW, word-processor or database program) has to be cold-booted, you blow away the speech synthesizer routines. That's exactly what happens when you use copy-protected software.

The first piece of equipment I bought to try on RTTY was the AEA CP-1 modem. The Apple software that came with it was produced by another company and was copy protected. I called the software producer in an effort to gain some insight on how to use the program with my speech synthesizer—they refused to give me any help at all. They said it wasn't possible for a blind person to operate these modes!

About this time, AEA released their own software for the Apple computer; it wasn't copy protected. AEA graciously offered any help they could give in getting the software to work with my speech synthesizer, so I thought I would give that a try. I called on some other hams for assistance, too, but we just couldn't make it talk properly.

The major problem I ran into is the use of a split-screen presentation. In such a display, the screen is divided into two, or more commonly, three sections: one section for received text, one for a transmit-text buffer and a third for display of outgoing transmitted text. That's a real neat thing for sighted people who can read the screen, but how does a speech synthesizer know what to do? Is it supposed to voice characters as you type them as well as tell you what is being printed in the received-text portion of the screen? You can see how that sort of output would get a little confusing! Actually, I never even got that far into solving the problem. Despite all my efforts, I never could find a way to make the program talk at all.

Terminal Programs

I was still working on that project when I discovered modems (such as the AEA PK-232, Kantronics KAM and others) that could be connected to any computer and used readily available terminal programs to drive them. I already had a special talking terminal program and had been active on some bulletin boards, so I was sure that was the answer. I'm happy to say—*it works*! It's not as simple to get things working together as you might at first think, but it

can be done, and it's a blast!

For my first such setup, I used my Apple, Talking Transend and an AEA AMT-1 modem. I operated AMTOR because of the inherent error-checking feature and because a good friend, Gary McDuffie, AGØN, was on AMTOR and was helping me along.

Gary had a C64 and a program called Chirptext,® which was produced by AEA. Because I was always using my Apple for things other than Amateur Radio, I thought it would be nice to have a computer totally dedicated to my radio. I already had an older Echo GP with a serial and a parallel port, so I got a C64 and printer card and switched my AMTOR operation from the Apple to the C64. I tried Chirptext, but there was no way to send everything to the speech synthesizer.

With the help of Gary and Chuck Woodman, KØKXR, I wrote a short terminal program in BASIC. It worked, but I encountered some problems. The first was that I had to send everything specifically to the printer port, to which the speech synthesizer is connected. The synthesizer did not speak with keyboard entry, so when I typed in program lines, I had no way to edit them. If I encountered an error, all files and devices were closed. So, I had a hard time figuring out where the program had crashed.

Tim Goodwin wrote a subroutine for me that dumped the entire screen to the speech synthesizer. This was a big help, but I had to listen to a whole screen of information whether I wanted to or not.

Another problem I had was related to the synthesizer's need for a carriage return before it will speak. Many RTTY stations don't send a carriage return character until 65 characters or more have been transmitted. Even though the incoming information was printing fine on the screen, *I* had to wait a long time before the entire line was spoken! That made the traffic flow seem uneven.

I solved this problem in a couple ways. I set program flags and looked for the + ? character combination (the AMTOR "over" signal, which normally is not preceded by a carriage return). When the program saw that, it immediately sent a carriage return to the speech synthesizer. This ensured that I would hear the last line of text sent by the other station. I also had the program look for a space after the string was 30 characters long, send a return, clear the string and start counting characters again. I checked each character coming from the AMT-1 and set appropriate counters and flags. I built these into a string of the desired length and then sent the entire string to the speech synthesizer. That procedure did work, but I was never really comfortable programming on the C64 without synthesizer feedback from the keyboard, so I went back to the Apple and started developing a program there.

Before leaving the subject of the C64, I want to mention that a special speech synthesizer is available for the C64. I've never tried it, but I'm sure it would have solved many of my problems. The price of this synthesizer is around $225. Joe Giovanelli, W2PVY, is using one of these systems on a 2-meter mailbox and is doing quite well with it. He is using a commercially available terminal program, but had to rewrite quite a bit of it.

There is much more to this speech synthesizer subject than just making a computer talk. I love the screens that are bordered with asterisks ("star" to a speech synthesizer) or equals signs! Imagine your speech synthesizer saying "star, star" or "equals, equals" 40 times! Sure, there are programs that talk fine and are quite usable, but most software was written to be viewed on a screen, not listened to.

There are two distinct opinions regarding software for the visually handicapped. One is that all you have to do is somehow make your computer talk, and you can run existing software. The other opinion is that the program should be written specifically to talk. I adhere to the latter line of thought.

OPTIONAL BOARDS

Many software vendors for Apple computers copy protect their product (overall, that trend is decreasing—Ed.), so there isn't a good way to make it talk. I do have a card in my Apple called a Print-It Card, manufactured by Textprint.[13] There are other cards of this type around, but I have not tried any of them. These cards allow you to interrupt a running program by pushing a button wired to the card. You can then dump the contents of the screen to your printer, after which the program resumes. Instead of sending the screen output to a printer, I dump it to my Echo GP. (As mentioned earlier, if I have to cold-boot, the Echo card inside my Apple will not talk as there is no longer any supporting program in memory.)

I use the Print-It card dump only occasionally as it isn't really a satisfactory solution. You have to listen to the synthesizer voice the entire screen each time. There's no way of telling for sure where the cursor is, and there's no LINE REVIEW feature. Use of the card does, however, give me access to utilities and other software that otherwise would not talk at all. Graphics characters sure do drive the speech synthesizer crazy, though!

There is unprotected Apple and PC software that can be made to talk, but there are still obstacles to overcome. Sometimes the screen is addressed differently and the program in the background that works with the synthesizer never sees the characters. The screen changes, but the speech synthesizer doesn't say anything. What does all this have to do with Amateur Radio? Software for Amateur Radio is no different than other software. If you don't enter what the program wants, nothing (or the wrong thing) happens. The cursor may not move and the speech synthesizer doesn't say anything. It is easy to lose track of where you are. Whether it be a logging program, a program for designing antennas, or an RTTY program, the software may not be usable by a blind person. At a minimum, the program will probably need some modification.

Stay tuned. There's more to come.

Notes

[6]AEA, PO Box C, 2160, Lynnwood, WA 98036-0918, tel 206-775-7373.

[7]Kantronics, 1202 E 23 St, Lawrence, KS 66046, tel 913-842-7745.

[8]PC-Talk III is also available from The Headlands Press, Inc, Freeware, PO Box 682, Tiburon, CA 94920. [PC-Talk 4, the most recent version of this program, is not available as shareware —Ed.] Talking Transend is available from Computer Aids, 124 West Washington Lower Arcade, Fort Wayne, IN 46802, tel 219-422-2424.

[9]Street Electronics, 1140 Mark Ave, Carpinteria, CA 93103, tel 805-684-4593. The Cricket speech synthesizer, also manufactured by this company, is designed to be connected to port 2 of the Apple //c computer.

[10]Digital Equipment Corp, 146 Main St, Maynard, MA 01754, tel 617-897-5111.

[11]124 West Washington Lower Arcade, Fort Wayne, IN 46802, tel 219-422-2424.

[12]ProKey is a keyboard-enhancement program produced by RoseSoft, PO Box 45880, Seattle, WA 98145, tel 206-282-0454. A similar product, SuperKey, is produced by Borland International, Inc, 4585 Scotts Valley Dr, Scotts Valley, CA 95066, tel 800-255-8008; in California, 800-742-1133.

[13]Textprint, 8 Blanchard Rd, Burlington, MA 01803, tel 800-255-151 or 617-273-3384.〔QST〕
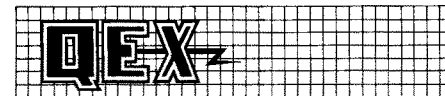
QEX: THE ARRL EXPERIMENTERS' EXCHANGE AND AMSAT SATELLITE JOURNAL

There are several ways in which to obtain a stable clock pulse for use in transmitting and receiving spread-spectrum signals. The circuit featured in October *QEX* is built from readily available parts and extracts jitter-free clock pulses by locking onto a different type of external reference signal source—an AM broadcast carrier. This circuit can also be used as a stable reference for frequency calibration purposes.

The October issue of *QEX* also includes articles on:
- "Extracting Stable Clock Signals From AM Broadcast Carriers for Amateur Spread-Spectrum Applications," by Andre Kesteloot, N4ICK
- "Midlatitude E_s at 220.1 MHz," by Michael R. Owen, W9IP/2
- "Power Dividers and Combiners," by Bill Olson, W3HQT

QEX is edited by Paul Rinaldo, W4RI, and Maureen Thompson, KA1DYZ, and is published monthly. The special subscription rate for ARRL/AMSAT members is $8 for 12 issues; for nonmembers, $16. There are additional postage surcharges for mailing outside the US; write to Headquarters for details.