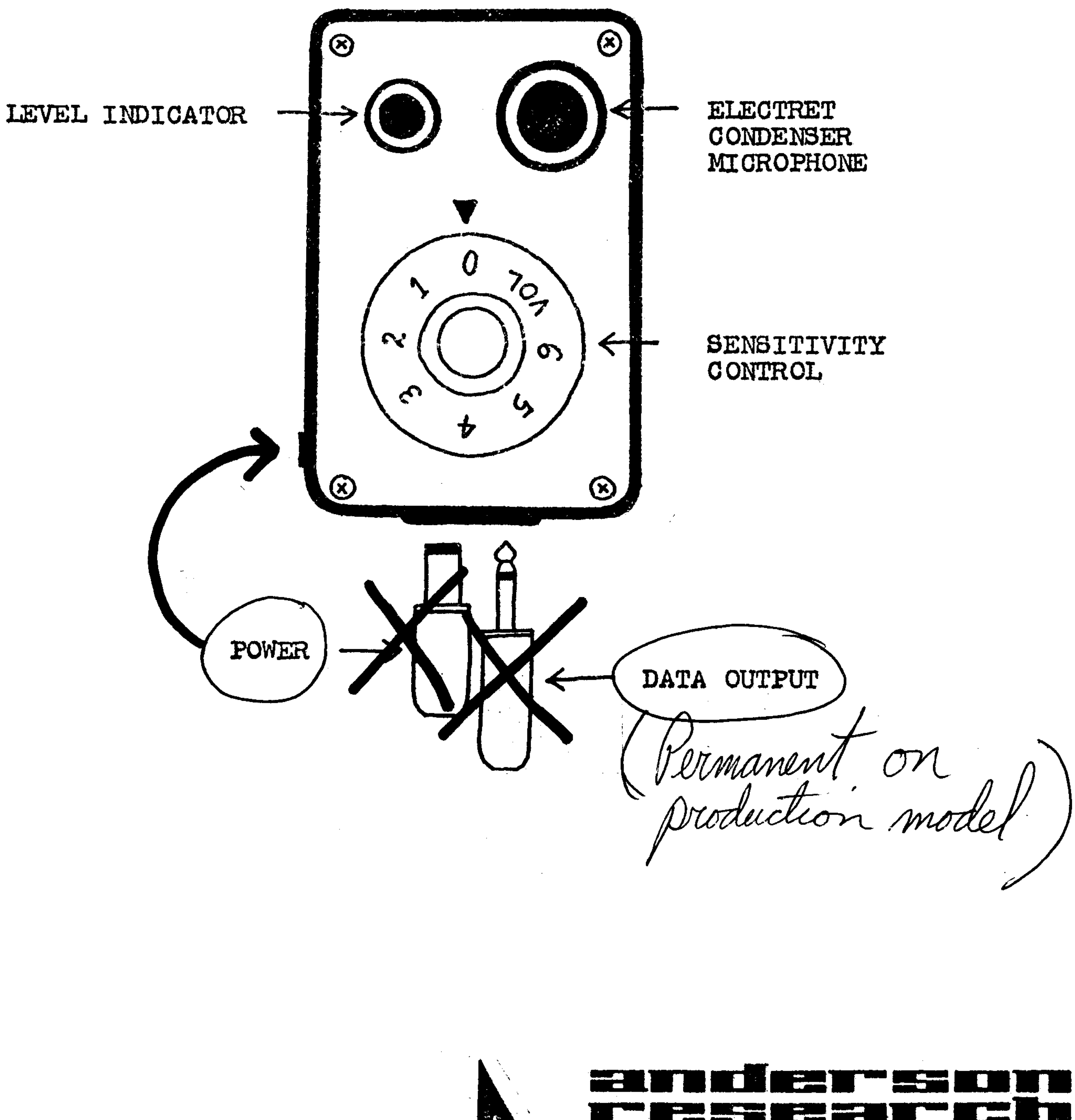
The COMPUTER EAR

User manual



anderson research and design

1611 Lacota Lane
Burnsville, MN 55337

Your COMPUTER EAR system includes the following:

Integral unit now

- Audio amplifier with optically isolated analog data output.
- Power cord with DC plug and battery clip.
- Computer input cable with RS-232 9-pin female connector.
- User manual.
- Program cassette: Side 1: Digital Oscilloscope
  Flash Math
  ~~Side 2:~~ Speech 1.8
  Speech 4.2

All on one side on your cassette.

The Digital Oscilloscope displays speech waveforms graphically on the screen. Flash Math is an example of a simple two-word speech recognition program using average amplitudes. Speech 1.8 is the basic speech recognition program with 1800 bytes of available RAM. Speech 4.2 is an advanced speech recognition program which "borrows" an additional 2400 bytes from the screen memory for an overall memory size of 4200 bytes!

INTRODUCTION

Anderson Research and Design was formed in 1972 by C. J. Anderson, a Minneapolis-based communications security consultant, to design and build custom communications security systems. We've been working with audio systems a long time, and when we became hooked on microcomputers in 1978 it was with the intent that eventually we would want to talk to them. The COMPUTER EAR is the result of our efforts to date.

Speech recognition by machine is not a well-developed field. It has not totally disentangled itself from the science-fiction world of Artificial Intelligence. After all, computers cannot "think," so how can they "talk" or "hear?" Do they merely mimic these human attributes, or is the rapid switching of electric current that allows them this pretense the same thing that is taking place in the human brain? Evidence suggests it may be.

There are no "turn-key" speech recognition systems on the market. The COMPUTER EAR is not an exact peripheral, like a printer, the results of which are exact and well-defined. It is highly experimental. We have achieved accuracy of upwards of 80% with the COMPUTER EAR, but only after patient practice and a solid understanding of the concept. We cannot possibly convey all that we have learned in this manual; we can only give you the basics and leave the rest to your own talent and imagination. We will be more than happy to answer any questions, give you further details, help you with any applications or correct any problems immediately. In addition, we will be sending out periodic updates as our system, and the field of speech recognition, becomes more sophisticated. To get on our mailing list, just send your name, address, date of purchase and make of your computer to:

Anderson Research and Design
1611 Lacota Lane
Burnsville, Minnesota 55337
Phone: (612) 894-2633

POWER SUPPLY

The COMPUTER EAR requires from 5 to 9 volts DC at a nominal 10 milliamperes. It will operate off of a standard 9 volt transistor radio battery or a 6 to 7.5 volt AC to DC adaptor. It may also be tied into a standard 5 volt regulated computer power supply buss using the DC power plug. Center pin is positive, sleeve negative. There is no "on-off" switch; the COMPUTER EAR is "on" as soon as power is applied. Be sure to disconnect power when the unit is not in use or is being transported.

LEVEL INDICATOR and SENSITIVITY CONTROL

With power connected, speak at a normal conversational voice level at arm's length from the built-in microphone. Slowly advance the VOL knob from "0" to "6", turning clockwise. The level indicator will begin to flicker at a setting of about "2" and increase in steadiness and intensity until the maximum of "6" is reached. (No indicator at all or a higher minimum setting is an indication of a weak battery or bad power connection.)

When you reach a maximum setting of "6", stop speaking and watch the indicator. If it continues to flicker, it is responding to ambient background noise of which you may be unaware. You have conditioned yourself to ignore everyday sounds once you have established their unimportance, but the COMPUTER EAR system will want to analyze everything it hears. You can filter out obvious sounds by decreasing the VOL knob setting until the indicator is dark when you are not speaking. If this adjusted setting is less than "4", you might consider finding a quieter location for your system. You will also be able to eliminate these false triggers by use of the threshold variable we will discuss shortly, but you will have a head start on achieving maximum results if your environment is relatively quiet.

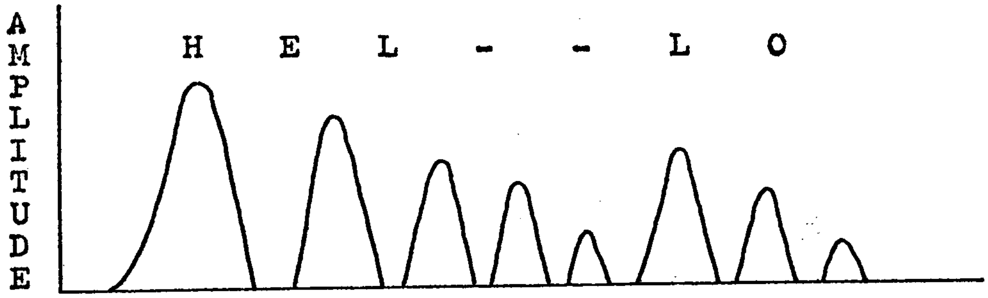
CONNECTION TO THE COMPUTER

Plug the nine-pin RS-232 plug into the number 4 port on the rear of your Bally Computer console. On most units this is the only connection necessary, ~~but some early models have a 1/8" jack to the right of the power jack to which the other end of the input cable must be connected.~~

Set the COMPUTER EAR on the front of your console to the left of the keypad (ideal) or on the desktop near the console. The COMPUTER EAR is designed to match the Bally cabinet.

AUDIO WAVEFORMS

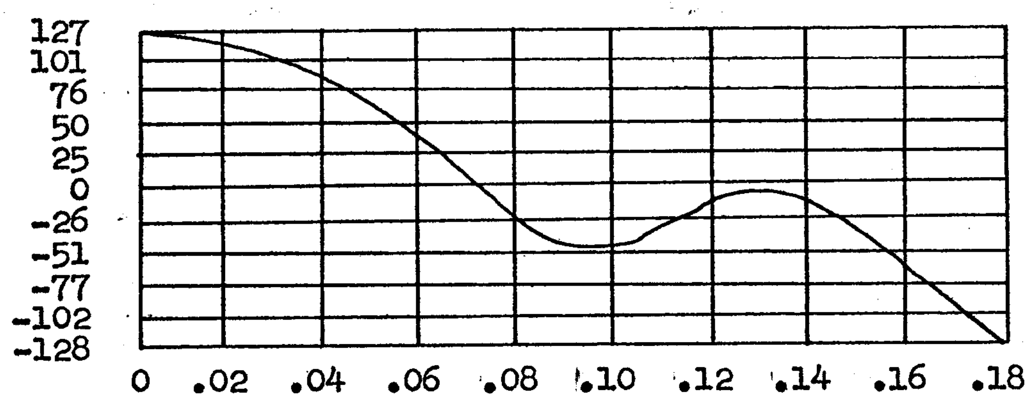
A sound wave is a wall of compressed air moving a little faster than 1000 feet per second and decreasing in intensity with the square of its distance from the source. Behind the compression wave is a partial vacuum travelling with it, called the rarefaction portion of the wave. The COMPUTER EAR disregards the rarefaction half of the wave, and measures the volume of the compression wave. We call this an amplitude-modulated half wave system. The word "hello" looks like this to the COMPUTER EAR:



This is a simplified graph. Actually, the COMPUTER EAR samples the waveform about 50 times per second, producing a lot more "wavecaps" than you see above. The more samples, the more distinct an individual word becomes. Each consonant and vowel sound has a distinctive pattern.

If we now assign a time constant to the horizontal line (X axis) and a numeric scale to the amplitude or vertical line (Y axis) we can read off a numeric value for each point in time. We'll use .02 second for our time increment since that's how fast the system cycles, and for our amplitude scale we'll use from Ø to the highest number the computer can store in one byte of memory: 255. But the Bally's analog input variable, KN, will shift this scale downward to -128 to 127. That's okay as long as you remember that the negative numbers do not represent the rarefaction half-wave. -128 is actually the Ø reference line.

Our word "hello" now looks like this:



Each time the program samples the waveform, at .02 second intervals, it takes the amplitude value at that point and stores it in memory. Our simplified "hello" example ends up as a "string" of numbers:

(127)(115)(83)(37)(-24)(-45)(-13)(-12)(-62)(-128)

In real time, the word "hello" will occupy from 25 to 30 string positions. If we tell the computer that that string represents the word "hello", it can compare that word against the next word it hears on a point-by-point basis and tell us whether the strings are a close enough match to assume that the second word is also "hello."

Since you will want a vocabulary of more than one word, the COMPUTER EAR <u>SPEECH</u> program compares the test word against all words in its vocabulary simultaneously and assigns each word a "score" based on how closely it matches the incoming word. If no word receives a passing score, the program simply falls through and asks you to repeat...sort of the computer's way of saying "Huh?" You may have spoken a word that is not in its vocabulary, or were not clear in your pronunciation. When the computer finds a word with a passing score, or the highest-scoring word of several passing scores, it branches to whatever job you want it to execute for that command. You may simply want it to print the word on the screen, or you may want it to "Turn on the lights" or "Walk the dog." If you've got a dog-walking peripheral, out goes Rover for a stroll around the block.

Word recognition is not an exact science, even in humans. Experienced programmers always insert a fail-safe subroutine that causes the computer to first tell you what it thinks you said before it executes it. A simple "yes" or "no" will then send the program off to execute the command or back to square one for another try. Without such a subroutine, Rover could be walking a long time before you ever get the computer to turn the lights on...

Now you understand the concept, and it's time to look at some variables including the biggest variable of all: human speech.

<u>THE DIGITAL OSCILLOSCOPE</u>

An oscilloscope displays waveforms graphically. The <u>Digital Oscilloscope</u> program on Side 1 of the COMPUTER EAR cassette allows you to freeze words on the screen and to compare waveforms against one another as an aid to vocabulary selection.

It is important to avoid including similar words in the computer's vocabulary. We are not ashamed to admit that it cannot distinguish between "to","too" and "two", or "where", "wear" and "ware". Neither can you except in the context of a sentence. But how about "stop" and "step", or "light" and "right"? The differences are slight, and you will need to examine those words visually before you can decide whether to include them in the computer's vocabulary.

Load and RUN the <u>Digital Oscilloscope</u> program. A listing is included at the end of this manual. The screen will begin by asking for variables.

1. THRESHOLD. The screen asks for the <u>threshold</u> variable, a number between -127 and 127. (Note: that is not a typographical error; the threshold variable freezes the program at any amplitude <u>less than</u> itself. Since there are no values <u>less than</u> -128, the lowest threshold you can set is -127.)

Since the COMPUTER EAR system is voice-actuated (it does not use a "push-to-talk" button like other units on the market...we wanted total "hands-off" control) you set the triggering amplitude by entering a value between -127 and 127, then pressing GO. -127 is maximum sensitivity: any sound will trigger the program. 127 is least sensitive. You will also use this variable, as you gain experience, to "clip" waveforms. Aberrations such as sibilance (an "s" or hissing sound) and "pop" (such as the sudden peak amplitude caused by pronouncing words beginning with the letter "p") can be reduced at the beginning of words by raising the threshold variable. For starters, enter -127.

2. SCAN. The screen asks for the scan variable. Since we know that the computer scans about 50 points per second it is obvious that the scan variable is time-related. There is no sense in having the computer burn up its limited memory by scanning beyond the end of the longest word you intend to pronounce. Set the scan variable to that value and press GO. Here are some relative scans:

25 equals .5 second: the word "hello"
50 equals 1 second: the word "rhinocerous"
150 equals 3 seconds: the word "antidisestablishmentarianism"

The Digital Oscilloscope can only graph 160 points, since that is the width of the Bally screen. In the Speech programs, you may enter words or sentences until you run out of memory. Each scan point requires 2 bytes of free memory. In Speech 1.8 the operating system (the COMPUTER EAR) uses 556 bytes out of 1800 available. That leaves you 1244 bytes freeboard. Now you need room for your executive subroutines, such as PRINT statements or LIST statements (Speech 1.8 uses the latter) or complicated instructions for walking the dog or turning on the lights. These must be subtracted from the 1244 bytes remaining. Each word or sentence requires another 2 bytes for "scoring overhead." Finally, the test word requires the same number of scan points as the rest of the vocabulary, so it must be subtracted. Whatever free memory is left is then divided by 2 (2 bytes per point, remember?) and the result is the number of scan points available. Speech 1.8 allows you about a ten one-second-word vocabulary if all you want is a printout of the word on the screen. Speech 4.2 is an exotic experimental program that gives you a lot more, but we will discuss that later.

Now you know the maximum number of points you may scan. How about a minimum? We have found that anything less than 20 points per word, even one-syllable words, is rock bottom. Discrimination accuracy drops sharply below 20 scan points. As you work with the Digital Oscilloscope you will notice that, due to the threshold clipping effect, the first syllable of a word tends to begin at maximum amplitude. You do not really get into the "meat" of a word until you are 10 to 15 points along. At a scan value of only 20, the computer is going to be "guessing" more than analyzing.

How deep must we scan? Only until the waveforms become distinct from each other. Two syllables will be enough to separate "rhinocerous" from "rhinitis" or "Rhine wine." Don't scan any deeper than you have to.

For a beginning value, enter 50 for "SCAN?:" and press GO.

Back to the Digital Oscilloscope. You've just entered a scan value of 50. The screen now asks for:

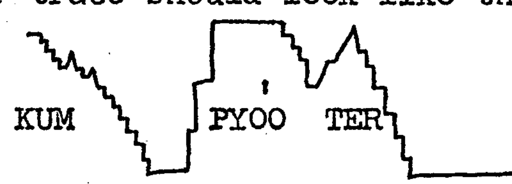
3. TRACE (S,M). Do you want a single (S) trace, so that each word is displayed individually, or do you want a multiple (M) trace in which each word is compared to the previous words? Enter S or M. You need not press GO. Try S for starters. Later, you will find M useful for selecting a full vocabulary, or for comparing word waveforms against each other or the same word against itself using different inflections, pronunciations or even speakers.

4. AVERAGE AMPLITUDE (Y,N). The screen asks whether you want the average amplitude feature (Yes or No.) Enter N for starters, but later you will use this feature to write "quickie" two-word programs such as true-false tests for educational use. True is easily distinguished from False, and Yes from No, just by comparing their average amplitudes. There is no need for a point by point comparison, so you will save time and memory when your vocabulary consists of only two distinct words.

The average amplitude is displayed as a horizontal line across the trace and a decimal value on the screen.

All the variables for the Digital Oscilloscope are now entered, and the screen will clear. The COMPUTER EAR begins listening for a word. Try the obvious: "Computer". The resulting graph will allow you to adjust one more variable: the VOL setting of the Sensitivity Control.

Assuming you use the midwestern, Walter Kronkite pronunciation "kum-pyoo-ter", your trace should look like this:



But suppose it looks like one of these two traces:

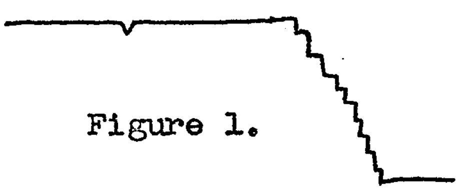


Figure 1.

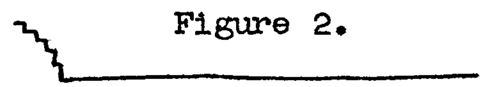


Figure 2.

In Figure 1 the VOL is set too high. You are "pinning" the amplitude at full sensitivity. Decrease the setting and try again until your trace resembles the upper example. Do not alter your voice! Always speak in a clear conversational tone.

In Figure 2 the VOL is set too low. Increase the setting and try again until you get that distinct, well-centered waveform.

What does the waveform represent? Notice the hard "K" sound, followed by a vowel warble on the "uh" sound. Amplitude decays until the hard "p" sends it back to maximum. "OO" is the accented vowel...notice the sustained amplitude. Another intersyllable decay, then a hard "t" peak and a gradual decay on the "er" ending. There it is, a picture of the word "Computer". No other word looks quite like it.

Okay...solo! You're on your own. Go crazy. Try every word you know. Play with the variables...use them all. See what they do. Try the different features. Use the Multiple trace to compare words against each other. Measure their average amplitudes. Talk loud. Whisper. Hold your nose. Watch the waveforms change as you walk around the room. But always try to understand what the waveform is illustrating. Pick out the individual syllables, the hard and soft consonants, the long and short vowels, the accents.

Come back in a couple of hours with your new insight into the world of human speech and we'll get into Word Recognition and some more variables.

NOTES

AVERAGE AMPLITUDE

You've observed from your work with the Average Amplitude feature that each word, pronounced the same way each time, has a distinctive average amplitude. One word may tend to average between -25 and -40 each time, while another may average between 90 and 110. You've also noted that average amplitude is related to scan; change the scan and you change the average amplitude. But providing the scan remains constant, as it does in a Speech program, the average amplitude can be used to distinguish between two vastly different words. Is average amplitude a good basis, then, for Speech Recognition? Not a chance! It can be used to distinguish between two vastly different words...period. Add a third word and you're in trouble. Fortunately, "yes" and "no" or "true" and "false" happen to be two pairs of vastly different words. If you're a parent or a teacher, or want a fast executive program, these words are of great value and worth including in a program.

Let's work with "yes" and "no." NO has a higher average amplitude than YES. Why? A hard consonant followed by a long vowel drives the amplitude higher than a soft vowel followed by a sibilant consonant every time. In our tests, "yes" usually resulted in a negative average amplitude, while "no" usually resulted in a positive value. (We say "usually" whenever we make solemn pronouncements about speech recognition programming.)

What good are they?

1. You can use them to write true-false tests.
2. You can use them in that "fail-safe" system we discussed earlier on page 4.
3. They require no resident vocabulary, therefore you can free most of your available memory for test questions, executive routines etc.
4. They are one-syllable words that need only a 20 point scan.

Load and RUN the second program on Side 1 of the COMPUTER EAR cassette: Flash Math. A listing is included at the end of this manual.

The operating program is short and simple. It scans 20 points, adds them together, divides by 20 and decides whether the answer is above or below a certain value. (Feel free to change that value.)

The Flash Math program is an example of a true-false test. The screen asks whether a multiplication problem is correct. You speak the answer "yes" or "no." The screen then tells you whether you are correct.

How reliable is an average amplitude based program? Try shouting "yes" and whispering "no." See? You've confused it already. We obviously need a more sophisticated system if we're going to recognize words and full sentences with 80% accuracy or better. Enter Speech 1.8.

SPEECH RECOGNITION

The concept of the COMPUTER EAR speech recognition program is explained on pages 2 through 4, and you have actually seen what the computer is hearing with the Digital Oscilloscope. You may have also made the following observations:

1. Monosyllabic (one-syllable) words tend to look alike unless they have some distinguishing characteristic. Words like "dog", "cat" and "box" resemble each other so closely that the computer will not achieve a great deal of accuracy in telling them apart. A word like "squash", however, with its sibilant beginning and end, is very distinctive. Even a word like "man" has a vowel warble in the middle that sets it apart. So, as a rule, don't include more than one monosyllabic word in the computer's vocabulary unless the other monosyllabic words have distinctive waveforms.

2. The same word looks different each time it is pronounced. This is a matter of practice. You must develop a steady, conversational tone and avoid changing your voice level, turning your head away, emphasizing different syllables or altering pronunciation. Keep practicing with the Digital Oscilloscope until you can match the waveform on the screen as closely as possible each time.

We will show you two programming aids to help the computer sort out variances in the same word: the discrimination variable and redundant vocabulary.

In our research and testing we have noticed that people tend to personify the computer, especially when it begins to mimic human attributes like hearing. When the computer misses a word, they carefully and slowly repeat the word as if the computer were a stupid or hard-of-hearing human. Naturally this tactic changes the waveform drastically, so the computer misses the word again. Some people actually end up shouting at the computer! Don't fall into this trap; always try to pronounce each word the same way it was originally entered into vocabulary. Practice until you master this ability.

SPEECH 1.8

Before loading the Speech 1.8 program from Side 2 of the COMPUTER EAR cassette, you must decide on a vocabulary. Your experiments with the Digital Oscilloscope will allow you to select words whose waveforms are distinct from each other. To begin with, select a small vocabulary such as four very distinctive words of about one-second length. As we mentioned earlier, Speech 1.8 will accept ten such words; fewer if they are longer, more if they are shorter.

Let's use, for our example, the words "rhinocerous", "hippopotamus", "giraffe" and "kangaroo". Program lines 1 through 19 are reserved for listing words as REM (remark) statements using Bally Basic's period symbol (.) to preface each word. Press RESET, then enter your vocabulary as shown on the next page.

1 .RHINOCEROUS
2 .HIPPOPOTAMUS
3 .GIRAFFE
4 .KANGAROO

Press GO after each word to enter it into text. After the last word has been entered, type:

:INPUT then press GO

and press PLAY on your cassette deck to load the Speech 1.8 program. The program will RUN automatically when it is fully loaded; the screen will clear and the letter V will appear at the upper left-hand corner of the screen. Stop the cassette and begin entering the variables the screen requests.

1. V (Vocabulary). How many words in vocabulary? Enter the number of words, 4 in this example, and press GO.

2. S (Scan). We hope you're familiar with this one, the time-based point scan variable. Enter 50 for starters; press GO.

3. T (Threshold). Another old friend. Enter your threshold, from -127 to 127, and press GO.

The screen will clear and then display:

SPEAK WORD 1 .RHINOCEROUS

Speak the word "rhinocerous" in your most normal manner. The oscilloscope trace will appear on the screen, then the question:

GO OR ERASE?

If you are satisfied with the waveform of the word "rhinocerous" press GO and it will become a part of the permanent vocabulary. If you are not satisfied, press ERASE and the trace will disappear. The screen will then ask you to SPEAK WORD 1 .RHINOCEROUS again. You may continue repeating this word, examining the trace and pressing ERASE until you are satisfied that you have a "keeper". (In this case, sort of a "zookeeper". But we digress.) Press GO and the screen will display:

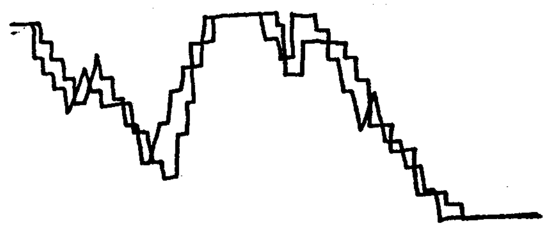
SPEAK WORD 2 .HIPPOPOTAMUS

Speak the word "hippopotamus". Compare its trace to the trace of the first word "rhinocerous". Are they satisfactorily distinct from one another? If not press ERASE and the "hippopotamus" trace will disappear. You will be asked to try again until you are satisfied with the distinction and press GO.

At this point it will be helpful to remember that the pronunciations you are entering into permanent vocabulary will have to be duplicated later on when the computer asks for a test word, so do not use any strange pronunciations to vary the waveforms. You will never remember them! If two waveforms match closely after several tries, change your vocabulary.

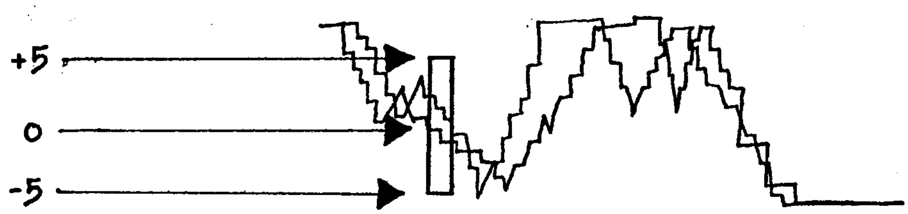
Enter the remaining two words in vocabulary in the same manner. The vocabulary is now "loaded", and the screen is displaying the waveforms of the four words. The computer will now ask for two more variables which we have not encountered yet.

1. D (DISCRIMINATION). Perhaps you noticed in your experiments with the Digital Oscilloscope that the same word could have a slightly lower or higher overall amplitude when spoken more than once. Not only that, but certain portions of the word could be higher than the first pronunciation and other portions lower. Still, the overall waveform had approximately the same shape. However, if we were to score only those points where the two waveforms coincided (a "hit") we would end up with a very low score indeed. For example:



Above, you see two word waveforms roughly paralleling each other. Sometimes one is higher, then it crosses over and becomes lower in amplitude. We know it is the same word, but if only we could tell the computer to look a little bit above and below the original waveform and score the second waveform as a series of "hits"..!

Well, we can. The DISCRIMINATION variable is a number that tells the computer how many points high or low the second waveform can be and still score as a "hit". We say that DISCRIMINATION "opens a vertical acceptance window". The value of D tells the computer how far to open the window. Don't open it too far, or pieces of other words will blow in. A DISCRIMINATION value of 1 requires a direct "hit", a perfect match, to score. A value of 5 opens a window that is 10 amplitude numbers high: 5 above the vocabulary word waveform and 5 below. The "window" can be visualized as follows:



The scale is exaggerated for emphasis; obviously you wouldn't want a window that high. But as you can see, any waveform that passes through the window at that point in time will score as a hit toward that word.

Enter a DISCRIMINATION value for D based on the waveforms you see before you on the screen. We suggest you begin with a value of 1, then increase it to improve accuracy. You will reach a value where accuracy is greatest, then begins to decrease again as the "window" begins letting in other words. The largest D value is 128, which is useless: it lets in everything!

DO NOT enter a D value of Ø. 1 is the lowest value of D.

DISCRIMINATION is one of the two programming aids mentioned on page 9 to help the computer sort out variances in the same word.

2. F (FALL-THROUGH or FAIL). F is a value between Ø and S (SCAN) which establishes a failing score for the vocabulary. Simply: If you have a 50 point scan (the computer samples the waveform 50 times), the best score any word can receive is 50. A score of 50 would mean an exact match, a "hit" at every point. A score of Ø would be a complete "flunk"; no hits at all. Somewhere in between is a score that tells the computer to quit wasting its time looking through its vocabulary. The word it is looking for just isn't there. Maybe you spoke a word that wasn't in the vocabulary. Maybe you were just clearing your throat and triggered the search routine. Maybe your pronunciation was so far off that the computer couldn't possibly recognize the word you spoke. Whatever the reason, you will speed up the search by telling the program when to quit looking. You pick the failing score; it's up to you. What is a good score? It depends on the value you picked for D. High D values give high overall scores (waveforms get points just for being close). As a rule of thumb: at a SCAN (S) of 50 and a DISCRIMINATION (D) of 1, 25 would be a pretty lousy score. Set F even higher if you wish.

When the computer begins passing out scores to each word in its vocabulary and sees that no word has a higher score than F, it simply falls through the rest of the program and displays REPEAT. We mentioned this earlier at the beginning of page 4.

When D and F have been entered, and GO has been pressed after the F entry, the screen will display the prompt:

SPEAK

That's it. You're on. The COMPUTER EAR is listening for a word. Speak one of the four vocabulary words, trying to pronounce it the same way you did when you entered it into vocabulary. The COMPUTER EAR will scan the word, then display:

SEARCHING

The computer is now looking through its vocabulary to see if it can find the word you just said. If it finds it, the screen will display the word. If it cannot find it or another word close to it, it will ask you to REPEAT. Finally, it may display the wrong word. Now what?

Try again. Don't force it...the computer is not stupid. It wants to hear the word you entered into vocabulary. We hate to say this, but the computer seldom makes mistakes. If it displayed the wrong word, your pronunciation has changed significantly. Keep trying until you get the word you want on the screen.

Suppose you don't. Now is the time to begin adjusting the variables. Start with D, remember? HALT the program, enter D=(new value), then type GOTO 110 and press GO. The screen will display SPEAK and you're ready to try again. You can change any variable (T,D or F) in the same manner. GOTO 110 puts you back to SPEAK again without disrupting the vocabulary or the program.

We mentioned two programming aids on page 9. We've covered discrimination. What about redundant vocabulary?

This may be obvious by now: you can give the computer a better chance at finding the word it is looking for by entering that word into vocabulary more than once. Let's say twice, or three times if you have enough memory and can wait a little longer for the computer to complete its search. HALT the program and change the REM lines to read:

```
1 .RHINOCEROUS
2 .RHINOCEROUS
3 .HIPPOPOTAMUS
4 .HIPPOPOTAMUS
5 .GIRAFFE
6 .GIRAFFE
7 .KANGAROO
8 .KANGAROO
```

Now RUN the program and enter those eight words into vocabulary. The minor differences in the pronunciation of the same words will "average out" and increase the computer's accuracy greatly.

Okay, solo again. Once more you are on your own with a full speech recognition program this time. Practice with the variables until you know what to expect from each. Try different vocabularies. Amaze your friends and confound your enemies. When you are satisfied that you've got the hang of it, come back again and we'll take you on a journey into the Twilight Zone.

N O T E S

THE UNCHARTED WILDERNESS

What? Back again? Good heavens, man, you've got what you came for! But you say you want more? Very well, then, come with me into the laboratory. Mind you, it's all very experimental and I make no guarantees. Some of these things are still in the daydreaming stage.

SPEECH 4.2

4200 bytes of RAM from an 1800 byte computer? Sure. We'll just steal 2400 bytes from the screen memory and use it to store our vocabulary. After all, vocabulary is merely data. It isn't part of the operating system; it's just being operated upon. We'll poke it into the screen memory, leaving the top three lines free for communication, then tell the operating system to peek at those locations and read what we've written there. That frees the rest of our text area, where Speech 1.8 was storing its vocabulary, for executive subroutines like walking the dog and turning on the lights. We can't play video games because the screen will be loaded with permanent vocabulary, but for real-world interfacing, this is the cat's pajamas.

With 2400 bytes available for vocabulary we can enter up to 1200 scan points (including the test word). That's over 20 one-second words (50 points each) or several sentences.

If you simply want the word to print on the screen, enter your vocabulary in lines 1 through 19 just as you did earlier with Speech 1.8.

If you want the computer to branch to an executive subroutine, you will have to change line 120 of Speech 4.2 after it has been loaded to read:

```
120  IF @(B) = A GOSUB 200+B; GOTO 70
```

then write the executive command for Word 1 at line 201, the executive command for Word 2 at line 202, etc. Don't forget to RETURN.

Load Speech 4.2. It will RUN automatically, and begin asking for the usual variables...all at once. There is no Digital Oscilloscope at the beginning of Speech 4.2 as there was with Speech 1.8. The screen is being used to store vocabulary, and nothing can be printed below the third line. So D and F, the discrimination and fall-through variables, appear right after V, S and T. How will you know if a waveform is satisfactory? You'll have to have worked it out ahead of time using the Digital Oscilloscope. There is no GO OR ERASE either; the words load one after the other. You will actually see the individual bytes graphically represented on the screen in a "crazy quilt" pattern as you load the vocabulary into screen memory.

Yes, you can turn off the TV at any time. The vocabulary is in memory, not actually on the TV screen. But isn't it a beautiful sight?

Do you really want to see something dramatic? How about a "horse race" between words, watching the score counters for each word print out across the top of the screen as the computer searches its vocabulary? And, better yet, let's blank and restore each byte as the computer polls it. We will then see the entire memory in action, with flashing points of light illustrating the search program's progress as it dashes through the vocabulary comparing points and adding up the scores. Add these lines to Speech 4.2:

```
105  Q = %(A); R = %(A+ (BxSx2)); %(A) = Ø; %(A+ (BxSx2)) = Ø;
     %(Ā) = Q; %(Ā+ (BxSx2)) = R

106  CY = 4Ø; PRINT @(1),#4,@(2),#4,@(3),#4,@(4)
```

There. RUN the program, load your vocabulary and speak the test word when the screen displays the INPUT prompt. With lines 105 and 106 in place the program will take a long time to find your word, but we have yet to demonstrate it to anyone who doesn't begin cheering for their word to move up in the pack and finish as the winner. This is truly a dramatic program, and all of the COMPUTER EAR's secrets are laid bare. The top line(s) are the test word. The light steps sequentially from the first byte to the last. At each step, the search light flashes to each corresponding point of each vocabulary word and the score counters advance for every "hit". At the end, the flashes stop, the counters freeze, and the computer displays the winning word. You have just watched a speech recognition system in action.

For practical applications, delete lines 105 and 106.

There are some "bugs" in this program; as we said, it is purely experimental. It appears that the stack either overloads or that certain values in certain screen memory locations cause the computer to CALL certain ROM (read-only memory) locations. Don't be surprised if the program suddenly vanishes and is replaced by Bally's GUNFIGHT or some other ROM game. Just reload the program and try again. We have no idea why this happens on occasion, but as soon as we find out we will mail an update to everyone on our mailing list.

Speech 4.2 is an experimental program. Feel free to experiment with it. Try storing data in other locations than screen memory. We will appreciate hearing from you on your findings and on the applications to which you have put the COMPUTER EAR.

OKAY, IT CAN HEAR. CAN IT TALK?

Strictly experimental. We have words stored as numeric values in memory. They are recorded, just as if they were on magnetic tape or a phonograph record. By working backwards, we should be able to reconstruct the original waveform and "play it back" through, let's say, the on-board Sound Synthesizer. You will need a thorough understanding of the operation of the Sound Synthesizer and its many variables, but our experiments to date have indicated that it is possible to make the Bally Computer System "talk" by playing back the recorded waveform. You will have to recreate the full sine wave, both compression and rarefaction portions, and you will not be able to modulate the frequency, only the amplitude. But we have been able to come up with a sort-of Forbin Project monotone "voice" from the Sound Synthesizer, although it is hardly speech quality. By tweaking the various synthesizer values, and perhaps adding your own frequency modulation, you should be able to come up with a recognizable "human-sounding" voice. Please let us know if you do. In turn, we'll send out an update.

SECURITY APPLICATIONS

Law enforcement agencies use the "voiceprint" to identify individuals through their speech patterns. While this area is not fully developed, it appears that voiceprints are as unique as fingerprints, and the banking community is currently experimenting with their use for financial transactions via telephone as the cashless society approaches. (We refer to the end of currency and coin as a trading medium, not to that embarrassing state that we occasionally find ourselves in.)

The COMPUTER EAR can distinguish between persons by the manner in which they speak a test phrase. For a fun party game (assuming it is a quiet intellectual gathering and not a disco party), type in the names of guests instead of words in the REM lines 1 through 19 in either Speech 1.8 or Speech 4.2. Then have each guest, in the order listed, speak a two-second phrase such as "Who am I?" (100 scan points) into vocabulary. The computer should now be able to identify the guests as they repeat the phrase. (This is a wonderful aid if you are a night-club impressionist.)

Practical applications are unlimited. The most obvious is a door lock with an electric strike that unlatches only when the owner speaks a key sentence.

REAL-WORLD CONTROL OF PERIPHERALS

See the article: "Breslin - the Home Computer that Runs my House" by Bill Hawkins in the January 1980 issue of Popular Science magazine. For more far-fetched ideas, read Ray Bradbury's short story: "The House that Wouldn't Die", and Robert Heinlein's novel: The Number of the Beast. Don't forget the movie "The Forbin Project (Colossus)" next time it's on TV, even if it does make you want to take an axe to your computer.

GAMES

If all you need is a one, two or three line printout to play, you can use Speech 4.2 with the entire text area (minus the operating system) at your disposal. On the other hand, if your game requires use of the video screen, you'll have to use Speech 1.8 and limit your vocabulary, or move the vocabulary to another section of memory that's not doing anything. You cannot use Speech 4.2 with any project that requires full screen graphics.

Okay...solo. Now you're really on your own. (Not to worry...we're only a postcard away and we do answer our mail.)

Drop us a line about your latest COMPUTER EAR project. Oh, yes... say "hello" to your computer from us!

Line #	Statement(s)	Comments
1Ø	.DIGITAL OSCILLOSCOPE	
2Ø	.(C)1979 A.R.D.	
3Ø	NT=Ø;BC=Ø;FC=7;CLEAR;INPUT	Input variables.
	"THRESHOLD?:"A	
4Ø	INPUT "SCAN?:"B;W=16Ø÷B	
5Ø	PRINT "TRACE?(S,M):";C=KP	
6Ø	PRINT "AMP. AVG.?(Y,N):";D	
	=KP;CLEAR	
7Ø	IF KN(4)<AGOTO 7Ø	Threshold.
8Ø	FOR E=1TO B;@(E)=KN(4);NEX	Input spoken word.
	T E	
9Ø	IF C=83CLEAR	Reset if single trace.
1ØØ	X=-8Ø;Y=@(1)÷3;LINE X,Y,4;	Graph waveform.
	FOR E=2TO B;X=X+W;Y=@(E)÷3	
	;LINE X,Y,1;NEXT E	
11Ø	IF D=78GOTO 7Ø	
12Ø	F=Ø;FOR G=1TO B;F=F+@(G);N	Compute and display
	EXT G;H=F÷B;LINE -8Ø,H÷3,4	average amplitude.
	;LINE 79,H÷3,1;PRINT H;GOT	
	O 7Ø	

USE OF SHADED AREA IS FOR 2ND OR MORE LINES OF MULTI-LINE STATEMENTS

DO NOT ENTER A SPACE BETWEEN LINE # AND STATEMENT, THIS IS DONE BY THE UNIT

PROGRAM NAME FLASH MATH

Line #	Statement(s)	Comments
1	.FLASH MATH	
2	.(C)1979 A.R.D.	
1Ø	CLEAR ;NT=Ø;A=RND (1Ø);B=RND (1Ø);C=AxB;D=RND (2)	Computer picks a random multiplication problem.
2Ø	E=RND (1ØØ);IF E=CGOTO 2Ø	Select a wrong answer.
3Ø	PRINT "DOES",#3,A,#2,"x",#3,B,#2;"=",;IF D=1PRINT #3,C,#2,"?";GOTO 5Ø	Display problem and either correct or wrong answer.
4Ø	PRINT #3,E,#2,"?"	
5Ø	IF KN(4)<-127GOTO 5Ø	Threshold.
6Ø	FOR F=1TO 2Ø;@(F)=KN(4);NEXT F	Input spoken word.
7Ø	G=Ø;FOR F=1TO 2Ø;G=G+@(F);NEXT F;F=G÷2Ø	Compute average amplitude.
8Ø	IF F<ØIFD=1PRINT "'YES' IS CORRECT.";GOTO 1Ø	Display.
9Ø	IF F<ØIFD=2PRINT "'YES' IS WRONG. TRY AGAIN.";GOTO 3Ø	
1ØØ	IF F>ØIF D=1PRINT "'NO' IS WRONG. TRY AGAIN.";GOTO 3Ø	
11Ø	IF F>ØIF D=2PRINT "'NO' IS CORRECT.";GOTO 1Ø	

USE OF SHADED AREA IS FOR 2ND OR MORE LINES OF MULTI-LINE STATEMENTS

DO NOT ENTER A SPACE BETWEEN LINE # AND STATEMENT, THIS IS DONE BY THE UNIT

USE OF SHADED AREA IS FOR 2ND OR MORE LINES OF MULTI-LINE STATEMENTS

DO NOT ENTER A SPACE BETWEEN LINE # AND STATEMENT, THIS IS DONE BY THE UNIT

Line #	Statement(s)	Comments
2Ø	.SPEECH 1.8	
3Ø	.(C)1979 A.R.D.	
4Ø	NT=Ø;CLEAR ;INPUT V;INPUT S;INPUT T;CLEAR ;FOR A=1TO V;B=AxS;C=B+S-1	Input variables; set addresses for vocabulary.
5Ø	GOSUB 2ØØ;PRINT "SPEAK WORD",#3,;LIST A,1	
6Ø	IF KN(4)<TGOTO 6Ø	Threshold.
7Ø	FOR E=BTO C;@(E)=KN(4);NEXT E;G=1	Load one word into vocabulary.
8Ø	X=-8Ø;Y=@(B)÷3;LINE X,Y,4;FOR E=B+1TO C;X=X+(16Ø÷S);Y=@(E)÷3;LINE X,Y,G;NEXT E;IF G=2GOTO 5Ø	Oscilloscope trace.
9Ø	GOSUB 2ØØ;PRINT "GO OR ERASE?";E=KP;IF E=31G=2;GOTO 8Ø	Save or erase word and oscilloscope trace.
1ØØ	NEXT A;GOSUB 2ØØ;INPUT D;INPUT F;CLEAR	Input variables.
11Ø	PRINT "SPEAK"	
12Ø	IF KN(4)<TGOTO 12Ø	Threshold.
13Ø	FOR E=ØTO S-1;@(E)=KN(4);NEXT E;PRINT "SEARCHING";FOR G=C+1TO C+V;@(G)=Ø;NEXT G	Load test word. Clear scoreboard.
14Ø	FOR E=ØTO S-1;G=C;FOR H=E+STO CSTEP S;G=G+1;IF ABS(@(H)-@(E))<D@(G)=@(G)+1	Compare test word against vocabulary; post scores if points within "window".
15Ø	NEXT H;NEXT E	
16Ø	FOR H=STO ØSTEP -1;FOR J=C+1TO C+V	Find highest score.
17Ø	IF H=FPRINT "REPEAT";GOTO 12Ø	Fall-through.
18Ø	IF @(J)=HLIST J-C,1;GOTO 11Ø	Post winning word.
19Ø	NEXT J;NEXT H	
2ØØ	H=Ø;J=C+V;CY=4Ø;PRINT " ";CY=4Ø;RETURN	Reset "stack"; print blank line (26 spaces).

USE OF SHADED AREA IS FOR 2ND OR MORE LINES OF MULTI-LINE STATEMENTS

DO NOT ENTER A SPACE BETWEEN LINE # AND STATEMENT, THIS IS DONE BY THE UNIT

Line #	Statement(s)	Comments
2Ø	.SPEECH 4.2	
3Ø	.(C)1980 A.R.D.	
4Ø	NT=Ø;CLEAR ;INPUT V;INPUT	Input variables.
	S;INPUT T;INPUT D;INPUT F;	
	CLEAR ;FOR A=1TO V;B=175Ø4	Set vocabulary addresses.
	+(AxSx2);C=B+(Sx2-2);GOSUB	
	14Ø;LIST A,1	
5Ø	IF KN(4)<TGOTO 5Ø	Threshold.
6Ø	FOR E=BTO CSTEP 2;%(E)=KN(	Load vocabulary.
	4);NEXT E;NEXT A	
7Ø	TV=1Ø7;TV=115;PRINT	
8Ø	IF KN(4)<TGOTO 8Ø	Threshold.
9Ø	FOR E=175Ø4TO 175Ø4+(Sx2-2	Load test word(s).
	)STEP 2;%(E)=KN(4);NEXT E;	
	GOSUB 14Ø;FOR B=1TO V;@(B)	Clear scoreboard.
	=Ø;NEXT B	
1ØØ	FOR A=175Ø4TO 175Ø4 (Sx2-2	Compare and score.
	)STEP 2;FOR B=1TO V;IF ABS	
	(%(A)-%(A+(BxSx2)))<D@(B)=	
	@(B)+1	
11Ø	NEXT B;NEXT A;FOR A=STO Ø	
	STEP -1;FOR B=1TO V;IF A=F	Fall-through.
	GOTO 7Ø	
12Ø	IF @(B)=ALIST B,1;GOTO 7Ø	List winning word(s).
13Ø	NEXT B;NEXT A	
14Ø	Z=SZ;BOX Ø,32,16Ø,24,2;CY=	Collect garbage; blank
	4Ø;RETURN	print lines; set cursor.