# EARS

Speech Systems

38 W. 255 DEERPATH ROAD  BATAVIA, ILLINOIS 60510
(312) 879-6880

# Electronic Audio Recognition System

# EARS
# AN ELECTRONIC AUDIO
# RECOGNITION SYSTEM

WRITTEN BY LESTER HANDS

## COPYRIGHT NOTICE

## DISCLAIMER

Speech Systems makes no warranties or guarantees, express, statutory, or implied, of any kind whatsoever with respect to the product purchased, and any other implied warranty of merchantability or fitness For a particular purpose is hereby disclaimed by Speech Systems and excluded from any agreement made by speech systems. The product is sold on an “as is” basis.

Speech Systems will not be Responsible for any damage of any kind not covered by the exclusive remedies set forth in this limited warranty. Speech Systems will not be responsible for any special, indirect, or consequential damage caused by its products.

Speech Systems reserves the right to make changes to any products or specifications described in this manual without notification.

## WARRANTY

Speech Systems warrants EARS against defects in material and workmanship for a period of Ninety Days from the date of purchase to the original purchaser. The obligation of Speech Systems is limited to the repair or replacement of the product, free of all charges, which proves defective during this period. This warranty does not cover damage due to accident, negligence, abuse, or tampering.

## REPAIRS

Call the factory for information on the repair charges should the product become defective after the ninety day warranty has expired.

## UNPACKING AND INSTALLATION

Carefully remove EARS from the box and inspect it for possible damage that may have occurred during shipment. If there is any damage, save all packing material and notify the carrier immediately.

Your EARS uses circuitry which is sensitive to static charges. Do not handle the unit more than necessary. It is imperative that it never be removed or installed while the computer is on.

## SYSTEM REQUIREMENTS

EARS requires a color computer with a minimum of 32K of memory. If you wish to use EARS with our VOICE, or SUPER VOICE speech synthesizer simultaneously, then 64K is required. If you have the Radio Shack Speech and Sound Cartridge and you wish to use EARS with it simultaneously, you will need 64K and some additional software, our Radio Shack Speech and Sound Cartridge TRANSLATOR. This is available from speech systems for $24.95.

EARS will work in both a disk or tape system. However, it will not at this time work with JDOS, the operating system used by the J&M disk system controller.

## PROGRAMS INCLUDED WITH EARS

The tape and disk programs provided with EARS are identical. We supply the software on tape as well as disk at the same price. We do this as a convenience for you. The software is not copy protected and we encourage you to make backups. If you wish, we will supply you with the files on disk for $7. The files supplied are:

EARS
NUMBERS
REACTION
SIMON
QUIKDEMO
EARSEDIT
COLORS
EARSECHO
VOICECHO

All programs are BASIC except for the first two, EARS and NUMBERS.

## AUDIO OUTPUT CONNECTOR

You will notice upon examination of the EARS, that there are two connectors. The smaller connector is for the microphone headset and the other is for an auxiliary audio output connector to be used should you wish to connect the speech output of EARS to a hi-fi or stereo system. Note YOU DO NOT HAVE TO USE THIS OUTPUT CONNECTOR. The speech will automatically come through the speaker of your TV or monitor. We have supplied this additional output as a means for you to easily interface the speech reproduction capabilities of EARS with a high quality speaker rather than the typically low fidelity speaker of a TV or monitor. Note the output is not intended to drive a speaker directly. You must connect it to some type of amplifier.

## DIGITIZED SPEECH

We realize that you probably purchased EARS for its speech recognition capabilities. We thought we would save the best till last and first acquaint you with the speech reproduction capabilities. EARS is capable or reproducing extremely high quality speech as you are about to find out. We have included two BASIC programs REACTION and SIMON. Both of these programs use a file called NUMBERS which contains the information necessary for the reproduction of high quality speech.

## REACTION

Assuming you have your EARS unit plugged into your computer and turned on.

```
(C)LOADM “EARS”
EXEC
(C)LOADM “NUMBERS”
FLIP
(C)LOAD “REACTION”
RUN
```

The object of the game is to respond to the numbers pronounced by EARS. There really is not much more to say. We think EARS speaks for itself. We are planning to develop additional files consisting of hundreds of “high quality” words. These will be sold separately and we suggest you call Speech Systems for additional information.

## SIMON

Simon is another example of a program that speaks that uses the high quality speech NUMBERS file. Assuming that you have already loaded the program EARS and NUMBERS, just:

```
(C)LOAD “SIMON”
RUN
```

## THE MICROPHONE HEADSET

The headset supplied is an integral part of EARS. It has been specially designed for speech recognition. For example, the black foam shield covering the microphone is actually a wind screen to reduce “pops” and mouth wind that occurs when one speaks. In addition, the headset style of microphone assures that the distance between your mouth and the microphone is relatively constant. A distance of approximately 1 to 1 1/2 inches should give good results. Note also, that the microphone headset actually includes inside a tiny amplifier to amplify and condition your voice before it enters the EARS. This means that you cannot conveniently use another microphone.

## RECOGNITION vs. UNDERSTANDING

EARS is a speech recognition device. This means it is capable of recognizing words that you speak. Understanding on the other hand means that the computer through some extremely sophisticated artificial intelligence program, not only recognizes a particular word such as “hello”, but understands that hello is used as a method by which humans greet each other.

## SPEECH RECOGNITION

Speech recognition is the use of a machine to identify words spoken by a human being. It should not be confused with Voice Verification which is the use of a machine to identify a particular speaker. We are concerned with “What was said” and not with “Who said it”.

When the transistor was invented, many people thought that speech recognition was just around the corner. As investigators began to attack the problem, they learned how little they really knew about human speech. Now, thirty years later, when hundreds of thousands of transistors can be put on a single chip a quarter of an inch square, some types of speech recognition are becoming practical for some applications.

EARS brings you what we feel is the most viable implementation of speech recognition for the present and the near future. It combines a low price with high performance and a flexible design philosophy which will enable a graceful upgrade process as speech recognition techniques improve.

## SPEAKER DEPENDENT

EARS is a speaker dependent recognition system. This means that it must be trained to recognize specific words from a specific person's voice. At first glance, this may seem to be a more primitive system than a “speaker independent” system but note the following:

1) Current available speaker independent systems can barely recognize accurately with a vocabulary of ten words.

2) Speaker independent systems can be trained only by the manufacturer and the addition of a single word to a vocabulary usually necessitates restructuring the entire vocabulary.

3) Speaker dependent systems such as EARS offer a second level of input flexibility in that an individual user can train the system to recognize commands that make sense to that user. The most obvious example of this power occurs with different languages. The commands to run programs can be trained by each user in his/her own language. Even within a single language, you choose the verbal commands which make sense to you. This is not currently possible with a speaker independent system.

4) Since EARS performs its recognition task primarily in software, it can be upgraded or even converted to a speaker independent system as algorithms improve and become available.

## DISCRETE UTTERANCE SYSTEMS

The algorithm provided with EARS is a “discrete utterance” or “isolated word” system. This means that it requires a pause of silence before and after each word it recognizes. The pause required is only about 200 milliseconds long. During the pause the algorithm adjusts itself to the current level of background noise in the user's area. After some practice, you will become used to the operation but it is worth being aware of at first.

## BACKGROUND NOISE

All speech recognition systems are sensitive to background noise. EARS has three protections against the effects of background noise or aural pollution.

1) EARS comes with a headset microphone which insures that the microphone will be close to the user's mouth. Thus, when the user speaks she/he should be much louder and easier to hear than the background noise.

2) The constant adjustment to the level of noise heard between words also helps the system to successfully pick out the beginnings and ends of actual words.

3) EARS has a rejection threshold which can be programmed by the user. This feature is particularly useful in preventing EARS from reacting to typical room noise such as doors closing, telephones ringing, etc.

Even with these protectors, operating EARS in a quiet area will provide the best results. If this is not possible, observe the following precautions.

1) If EARS is to be used in an area with a “constant” level of background noise (such as a cooling fan running on equipment), then train EARS in that area.

2) if EARS will be used in an area with sharp intermittent noises (doors closing, telephone ringing), then train EARS in a “quiet” area and use the “Rejection Threshold” to protect against the intermittent noise.

## HOW TO SPEAK TO EARS

Changes in mood or emotional state can heavily “color” our speaking patterns. Differences in context or intent alter the vocalization of a spoken word so that it may convey both gross and subtle differences in meaning. These differences in inflection, pitch, and accentuation can make a word “unmatchable” for a discrete utterance voice system.

All speech recognition systems will perform better with a “consistent” tone of voice as their input. The best way to train and then command EARS is to adopt the firm tone of voice that you might use in training a dog to respond to your voice. Yes, seriously, when using EARS just picture your puppy sitting there waiting for your command and you will get fine results. It's not necessary to command EARS in a flat or artificial voice, just speak a command approximately the same way each time you would use it.

EARS accepts words or phrases up to approximately 2 seconds in length. For best results, use multi-syllable words as these are easier to differentiate than single syllable words. Any use of the alphabet will get best results using words like “Able, Baker, Charlie” To indicate the letters rather than the letter names themselves.

Another comment is worthwhile at this point. As you train EARS to your voice, you will also find that to some extent you will be training yourself to be cognizant of the way you speak. When EARS does make an error, it will many times be because you have not taught it correctly or you are not pronouncing the word or phrase the same way. For example, say 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. If you listen carefully, you will notice that when you said the last number 10, your voice probably dropped pitch slightly. This is normal for a person to speak the last word or words of a phrase differently than the words at the beginning. If you trained EARS with the lower pitched 10 and later said 10 in an isolated utterance, EARS will not accurately recognize it. However, be aware of the limitations of computers and the unique and complicated and perhaps wonderful manner in which we speak. EARS is by far one of the most complicated systems available on any computer today. The low price does not reflect the quality but the great strides that have been made in the last few years in the field of speech recognition.

**QUIKDEMO**

We have supplied you with a very brief and we hope “user friendly” program called “QUIKDEMO” to allow you to try speech recognition first hand. If you still have the ears program loaded just (C)LOAD “QUIKDEMO” and RUN, if not:

```
(C)LOADM “EARS”
EXEC
(C)LOAD “QUIKDEMO”
RUN
```

You will be asked to say the numbers 1 to 5. Just speak the number and each time you are requested to do so.You have taught EARS these 5 words, you will be requested to say them. Note that we have gone against our device of staying away from single syllable words. In other words, you will find that you get better results if you say “number 1”, “number 2” etc., rather than 1, 2 etc. We supplied this program as a quick example and to get you acquainted with EARS. Before moving on, spend some time with this program. You should find that with a little patience on your part, and a little practice training EARS, you should certainly get results with nearly a 100% recognition rate.

## EARSEDIT

EARSEDIT is the editor that you will use to train the system. The program is written in BASIC so it is loaded as you would any BASIC program (i.e. (C)LOAD “EARSEDIT”). We have assumed that you still have the machine language program EARS loaded from running SIMON and REACTION, if that the process is:

```
(C)LOADM “EARS”
EXEC
(C)LOAD “EARSEDIT”
RUN
```

The program is menu driven. Try option 1 to list the voice templates in memory and you will find there are none. We are going to train EARS how to say 9 colors and the word “end”. Our goal is two-fold. First to acquaint you with how the EARS EDITOR program works, since you will be using it a great deal to prepare vocabularies. And second, to prepare you for a program called COLORS which is a BASIC program that will allow you to change the colors displayed on the TV screen merely by speaking the 9 colors.

Select option 2 to input a word. The first word to teach EARS is BLACK. After hitting ENTER, you will be prompted to speak the word BLACK. You have just taught EARS how to pronounce BLACK. You will then be asked if you wish to average the word by saying it again. Found that each time you speak a word that you say it somewhat differently. Therefore by speaking the word twice and averaging it you will probably fine that the recognition rate is better. There are other ways to increase the recognition rate that are explained later in the manual. If you decide to speak the word again, you will be prompted to do so. Experience has shown that speaking or training EARS more than twice actually can reduce the recognition rate.

Having trained EARS how you say BLACK, you can now pick option 1 again to list the templates that you have in memory. You will now find the word BLACK. If you have not turned off your computer or cleared the voice templates(option 5), you will also find the words that you trained EARS in the QUIKDEMO program, namely the numbers 1 to 5. Let's continue teaching EARS the following words: GREEN, YELLOW, BLUE, RED, BUFF, CYAN, MAGENTA, ORANGE and END. The “end” was included since that is the way we will end the program when we are ready.

## TESTING and REJECTION THRESHOLD

Select option 1 and you will find that there are now 10 voice templates in memory. Before saving these templates for proceeding on to the program COLORS, you may wish to test these words to assure yourself that the words are recognized properly. To test the words, select option 6 and speak one of the words. You should find that the word you speak is printed. If not, you may go back and delete a word that is not being reliably recognized and train the word again. You may also find "NO MATCH" being printed. EARS gives you the ability to specify the precision of the match. That measure of precision is called the Rejection Coefficient. EARSEDIT defaults to a rejection coefficient of 4. Option 0 is used to modify the value to one that you find best. It is important to note that the program COLORS does not use the rejection coefficient so changing it in EARSEDIT does not in any way affect COLORS, we mention it here since we feel that when you write your own BASIC program, you will want to incorporate the REJ command. This command Will be discussed later in more detail. Note also, that if the REJ command is not used in a program, when you speak a word, by definition, EARS will always return a word which represents the "best match" even when that match is very poor. In most cases you will not want a response from EARS unless it is reasonably close. This is just another way of explaining why using a rejection coefficient is important.

Now ready to save your voice templates to disk or tape if you so wish. However, note that they will remain in memory until you either load in another set of templates (option 3), delete templates selectively (option 4), clear all templates (option 5), or of course turn off the computer. Should you wish to save the 10 templates choose option 7 and give a file name. Note that the system will automatically save the templates to disk if you have a disk system, or to tape if you have a tape system. If you have your disk controller plugged in, the system assumes you are running a disk system.

## COLORS

We are now ready to run our first speech recognition program. We assume that you still have 10 voice templates trained using EARSEDIT still in memory. Now just:

```
(C)LOAD "COLORS"
RUN
```

Now speak any of the 9 colors you have taught EARS and you will find that the screen of the TV changes to the color you describe. You will find that EARS has a recognition rate that is typically 95% or better. However, it is important that you make an effort to pronounce the words just as you did when you first trained EARS to that particular word. Consistency is the key. Merely speak the word the way you normally do and you should be fine. The program COLORS may be terminated merely by speaking END.

## EARSECHO

This BASIC program is supplied to show you how EARS can be used simultaneously as a speech recognition device and for speech reproduction. To run this program, you must have a 64K computer. The program assumes that you have already trained EARS how you say the numbers 0 to 9 and END. As we have previously done, we must (C)LOAD "EARSEDIT" so that we can train EARS for these 11 words. Assuming that you have EARS loaded, merely (C)LOAD "EARSEDIT". If you are starting from scratch:

```
(C)LOADM "EARS"
EXEC
(C)LOAD "EARSEDIT"
RUN
```

If you chose option 1, You will find the templates that you have loaded. If you have the COLORS templates still loaded you may wish to clear them out by selecting option 5. Since EARS is capable of storing 64 words you certainly have the option of just adding the numbers to the colors if you like. However, having templates in memory that you know are unnecessary is somewhat wasteful since it means the additional words will require more of the computers time to examine them and more importantly, additional errors may occur since one of the words may be similar to another.

As we did in the program COLORS where we used EARSEDIT to train the colors, we will train EARS how you say the numbers 0 to 9 and the word END. Merely choose option 2, type the number 0 and say the word when you are prompted by the computer. Continue until you have all 11 words entered. You made at this time wish to test the words by selecting option 6. If you feel that some of the words are not correctly identified, delete them individually (option 4) and train them again (option 2).

Because we will loose our voice templates that we just trained EARS for, we must save these templates by selecting option 7 and specifying a file name, assume "YOURNUMB". The reason we loose our templates is because the digitized speech (i.e. file NUMBERS) resides in the same memory area as the voice templates. It is only after would use the FLIP command that the digitized numbers are flipped to a second 32K memory page, which explains why a 64K computer is required when using speech recognition simultaneously with speech synthesis.

You are now ready to run the EARSECHO program. We assume that EARS is still loaded since you have not turned off your computer and now merely:

```
(C)LOADM "NUMBERS"
FLIP
TLOAD "YOURNUMB"
(C)LOAD "EARSECHO"
RUN
```

Speak the numbers 0 to 9 and you will find that EARS will echo what you say. Whether you realize it or not, this simple little program demonstrates technologies and sophistication that is in every sense of the phrase, “state of the art”.

## VOICECHO

The program VOICECHO is provided to allow you to actually carry on a conversation with your computer. This program requires a 64K computer. In addition, it also requires a speech synthesizer such as the VOICE, or SUPER VOICE. Since the computer cannot really understand what we speak, we must help computer computer by telling it what the conversation is going to be about. Here is the conversation that the program VOICECHO is expecting.

| ---- YOU ---- | ---- COMPUTER ---- |
|---|---|
| Computer | Yes sir. |
| How are you | Fine thank you and how are you? |
| Fine | Can I help you. |
| Yes please | What do you want to know. |
| What is 44*123 | 5412. |
| Capital of ND | Bismarck. |
| Thank you | You're welcome. |
| Goodbye | Have a nice day. |

The big advantage of a speech synthesizer like the VOICE for the SUPER VOICE is that it is has an unlimited vocabulary. For those that have elected to purchase the Radio Shack Speech and Sound cartridge speech synthesizer please be advised that we do have a software product that is sold separately that will allow you to run the VOICECHO program shown above. Call Speech Systems and order the Radio Shack Translator ($24.95). To our SUPER VOICE and VOICE customers, you have everything you need. SUPER VOICE users note that you may load either the TRNSLATE text-to-speech program or the SUPERTOK text-to-speech program.

By now you have realized that first you must train EARS foe the words that you are about to use in the conversation. To do this just (C)LOAD “EARSEDIT” assuming that “EARS” is loaded, if not then:

```
(C)LOADM “EARS”
EXEC
(C)LOAD “EARSEDIT”
RUN
```

By now, we hope that you are a little more comfortable with the EARSEDIT program. You will select option 2 to input the 8 phrases to be spoken by you. We should mention that whenever you input a phrase you cannot exceed 15 characters. Also note that in the question “Capital of ND”, you may speak North Dakota but not input it since it would be more than 15 characters. In addition, the program VOICECHO is expecting “Capital of ND”. Assuming you have trained EARS for the 8 phrases, we suggest that you quickly test them for reliability by selecting option 6. Should you find a phrase that is unreliable, delete it and train EARS again. You now must save this file (option 7) since what we are about to do will erase the voice templates that are currently in memory. For the sake of example, let's name the file “VOICECHO”. Even though the name is the same as the BASIC program on a disk system they carry two different extensions (i.e. .BAS and .TMP).

While you still have the EARS program loaded, let's assume you do not and suggest you perform the following steps:

```
(C)LOADM “EARS”
EXEC
(C)LOADM “TRNSLATE”
FLIP
TLOAD “VOICECHO”
(C)LOAD “VOICECHO”
RUN
```

Note that the TRNSLATE program is the program that comes with our speech synthesizers. The BASIC command “TLOAD” is a very special Template Loading command that will be discussed later.

You can now carry on a conversation with your computer. Try experimenting with it.

## HANDS OFF PROGRAMMING

EARS has the very unique capability of allowing you to literally program in BASIC without ever touching the keyboard. Since you are limited to 64 templates in memory and there are more than 64 BASIC commands (i.e. PRINT, CLS, GOTO etc.) you must train EARS to recognize the BASIC commands that you intend to use. We have found after much trial and error “hands-off programming” that the following list of words is sufficient for many applications. We included it here merely as a starting point, you should tailor it to your specific needs.

```
(13)          ,              )              EDIT
(08)          SCREEN         (              10
(03)          CLS            =              20
“             CHR$           +              30
0             FOR            -              40
1             NEXT           ELSE           *
2             IF             A              /
3             THEN           B              T
4             GOTO           X              M
5             GOSUB          C              D
6             >                             I
7             <              :              H
8             .              RUN            STEP
9             TO             LIST           SOUND
PMODE         CIRCLE         DIR            LOAD
```

Before we continue, you are probably excited about trying it out and not interested in training EARS in all 64 words. So let us quickly go through one example. (C)LOAD “EARSEDIT” as we have previously discussed. Select option 2 to input a word and type “LIST*”. You will be prompted to speak “list”. Now exit the program and go back to BASIC. You will see the familiar BASIC “OK” prompt. Now type “DRECT”. This is a new BASIC command that we have added and will discuss it in more detail later. Now speak “LIST”. You will find the program “EARSEDIT” is listed. Use the BREAK key to exit this direct input (hands-off) mode.

Note that many of the words used were picked for a very good reason. For example, rather than having 4 words: CLOAD, CLOADM, LOAD, LOADM, we used the single word LOAD and included C and M to give us the necessary 4 BASIC commands. Now we can use C and M for other purposes such as variables.

Note in column 3 of the above list, underneath the letter “C” is a blank, this is meant to be the word “space”. When you train EARS to recognize this character, you will merely type in a space followed by hitting the enter key and then speak SPACE.

The list shown above is the way you will see the commands when you list the templates. The word (13) represents the ENTER BASIC command. Therefore, assuming that you are in the EARSEDIT program, you would select option 2 to input a word and then type

*13 and speak the word ENTER.

In a similar manner, (08) would be pronounced BACKSPACE, and (03) would be the word BREAK.

Note also that the word “FOR” as well as the number “4” is listed. The computer cannot tell the difference so we suggest that when you train EARS for “FOR” of a FOR-NEXT loop that you speak the pseudo word “FOR-NEXT”.

In a similar manner, the word “TO” and the number “2” are both listed. Once again, you must take care not to use the same word. Think of a word to differentiate the two such as “number 2” when you wish the number 2.

Remember, using the least number of words for the application is the best general advice since the recognition rate is increased and the response is also better.

## EARS MEMORY MAP

EARS is a hardware device that is memory mapped just like your disk controller and other peripherals such as our VOICE, SUPER VOICE, STEREO PAK, and others. We have designed all our devices so as not to conflict with any of our products or the products of other vendors. EARS is completely decoded so it requires only 4 bytes of memory starting at $FF78. It is the extra work that we have put into our hardware devices that lets you use them with any expansion chassis such as the MULTI-PAK or a simple Y-CABLE.

## **A PERSONAL NOTE**

On behalf of myself, Rich Perry, the EARS project manager, and Lester Hands, the programmer; we both sincerely hope you enjoy the product and realize that EARS represents a product in many ways more sophisticated than the computer itself. We intend to support this product since applications exist that boggle the mind and we are sure that you the end user will find them. Should you develop something unique, we would like to hear from you.

## WRITING YOUR OWN PROGRAMS

Thus far we have not told you anything about how to program EARS yourself. We felt that we would let you first have a little fun and acquaint you with EARS by letting you use our programs. No doubt you are anxious to try writing your own programs. The remainder of this manual is for you.

# EARS: A SPEECH RECOGNITION SYSTEM
# FOR THE COLOR COMPUTER

Written by Lester Hands

## INTRODUCTION

EARS (Electronic Audio Recognition System) is a speech recognition system from Speech Systems designed to be used with the Radio Shack Color Computer.

The uses for speech recognition are endless; applications for the handicapped, east entry of programs, education, and games are just a few.

To use EARS, it is assumed that you have at least an acquaintance with programming in BASIC. EARS adds 14 new words to BASIC. With these words you can easily write programs that are capable of recognizing spoken speech. This program is compatible with Extended or Disk 1.0 and 1.1 BASIC.

If you have a speech synthesizer (such as "The Voice", or "Super Voice", or the Radio Shack Speech and Sound Cartridge, the Multi-Pak (or triple Y-cable), and 64K of memory, EARS will allow you to perform speech synthesis and speech recognition at the same time.

You are about to enter the leading edge of a rapidly developing technology. Speech recognition is still in its infancy. To mimc the human brain, with its capability to recognize a very large vocabulary (including more than one language), compensate for changes in pitch (such as male or female voices), concatentation (slurring of words together which normally occurs in conversational speech), and differences in accent, would take a computer much more sophisticated than available today.

EARS represents a breakthrough in terms of cost/performance. A few years ago such a system would have cost many times more and would have been much harder to use.

EARS uses a form of speech recognition that is called "speaker dependent". This means that the computer has to be taught by you how you say the words you want it to recognize. If someone else were to say the same words, chances are the computer would not recognize the same words correctly.

In contrast to this is "speaker independent" recognition.  As you would expect, this system is capable of recognizing anyone saying certain words. Such a system is much more complex and expensive than a speaker dependent one. Each word must be programmed by the manufacturer, and usually only a few words can be recognized.

## GETTING STARTED

Turn your computer off. Insert the EARS cartridge into the slot on the side of your computer or into any slot of a Multi-Pak extender. If you have a disk system but don't have a Multi-Pak, you can use the triple Y-cable available from Speech Systems. Plug the supplied microphone headset into the side of the EARS. Adjust the headset so that it is comfortable. Position the microphone so that its near your mouth but just above or to the side to avoid “wind” distortion. Please note that you must use the supplied microphone. This is because special voltage levels are needed to work with EARS. Now turn the computer on.

After loading the tape or disk containing the EARS program, load EARS into memory by typing

```
        CLOADM “EARS”:EXEC
```

or for a disk system,

```
        LOADM “EARS”:EXEC
```

and then press the “ENTER” key. A copyright message will appear. You are now ready to run a speech recognition program or to write your own program.

## GENERATING AND MATCHING TEMPLATES

To show you how easy EARS is to use, enter

```
        LISN “HELLO”
```

The cursor Will disappear and the computer will wait until you have said “hello”.

You have just taught the computer how you say the word “hello”. Enter three more words by entering

```
        LISN “HOW”
        LISN “ARE”
        LISN “YOU”
```

Now the computer has four words in memory. Enter

```
        LISN:MATCH
```

and say one of the words you have just entered. After a brief pause, the word you have just said will be printed on the screen. You could combine what you have typed into a small program that looks like this:

```
        10 PRINT”SAY HELLO: “
        20 LISN”HELLO”
        30 PRINT”SAY HOW: “
```

```
40 LISN”HOW”
50 PRINT”SAY ARE: “
60 LISN”ARE”
70 PRINT”SAY YOU: “
80 LISN”YOU”
90 PRINT:PRINT”NOW SAY ONE OF THE WORDS YOU JUST ENTERED”
100 LISN
110 MATCH
120 PRINT
130 GOTO 100
```

Now that was pretty easy, wasn't it?

You may have noticed that the command LISN Was used in two different ways in the program above. The first way was to teach the computer new words. The second way was to listen for a word so the computer could match what you had just said against the list of words it had in memory.

Each time, LISN generates a “voice print” or template of the word you say. If the command LISN is followed by a word, then the template is named by that word and saved for future reference. If LISN is not followed by a word, then the template is stored in a temporary holding area so it can be matched later against one of the named templates in memory.

You may be wondering if this word that follows LISN can be a string variable such as A$. Yes! any string variable is permissible, except for string arrays such as A$(4,5) and the specially used variable WO$.

## **LISTING TEMPLATES**

If you forget what words you have a memory, enter

```
TLIST
```

and you will get a list on the screen. If you have more than 15 words, the list will halt as soon as the screen fills up and wait for you to press any key.

## **USING THE WO$ VARIABLE**

WO$ is a variable used by the EARS program and should be used carefully by you. To see How it is used, type in the following program (it is assumed you have already entered some templates):

```
10 WO$=””
20 LISN
30 MATCH
```

```
        40 PRINT “THE WORD YOU SAID WAS :”;WO$
        50 GOTO 20
```

Notice this time that the command MATCH did not print the word on the screen; instead the word was put in the variable WO$ where the program could use it. The only restriction on the use of WO$ is that you must “declare” it before you use MATCH, as in line 10 of the program.

Now let's try something a little more ambitious: each time you say a color, the TV screen's color will turn to the color you said. First we will need to teach the computer the colors:

```
        10 WO$=””
        20 PRINT”SAY BLACK:”
        30 LISN “BLACK”
        40 PRINT “SAY GREEN:”
        50 LISN “GREEN”
```

Now tell the computer to listen for you to say a color, And then change the color of the screen accordingly:

```
        100 PRINT “NOW SAY A COLOR”
        110 LISN:MATCH
        120 IF WO$=”BLACK” THEN CLS0
        130 IF WO$=”GREEN” THEN CLS1
        140 GOTO 110
```

Of course, you could easily teach the computer all 9 colors.

## **SAVING AND LOADING TEMPLATES**

OK, now you have created a number of templates, and now it is time to save them. Get your tape recorder or disk drive ready and then enter

```
        TSAVE”MINE”
```

The templates will be saved to tape with the filename of “MINE” or to “MINE/TMP” if you have a disk system. If you have a disk drive, you cannot save the templates to tape.

The corresponding load command is TLOAD. Both TSAVE and TLOAD function like the CLOAD (or LOAD) and CSAVE (or SAVE) BASIC commands, with one important difference: you cannot specify an extension name for a disk system. This is always “/TMP”.

You can use more than one disk drive by specifying the drive number as you would with BASIC LOAD or SAVE. Here are some examples:

```
        TLOAD”1:MINE”
        TSAVE”MINE:1”
        A$=”FILE”+”:2”:TSAVE A$
```

## DELETING TEMPLATES

Should you goof and enter a bad template, you can delete it by entering TDEL. This deletes the last template in memory. Any template in the temporary holding area is unaffected. If you follow TDEL by a template name such as “HELLO” or A$ (where A$ contains the name you wish to delete), then all the templates with that name will be deleted. Try it out, then list the templates with TLIST to make sure it worked.

All the templates in memory can be deleted with TCLEAR.

## DIRECT ENTRY

Now you are ready for a real treat. How about programming your computer just by talking to it? First type in the following program:

```
        10 DIM A$(10),B$(10)
        15 TCLEAR
        20 DATA PRINT,PRINT,A,A
        25 DATA 1,ONE,2,TWO,3,THREE,4,FOUR
        27 DATA =,EQUALS,0,ZERO,+,PLUS,-,MINUS
        30 FOR N=0 TO 9
        40 READ A$(N),B$(N)
        50 A$=A$(N):B$=B$(N)
        60 PRINT”SAY: “;B$
        70 LISN A$
        80 NEXT N
        90 PRINT”SAY: ENTER”
        100 A$=CHR$(13)
        110 LISN A$
        120 PRINT”SAY: LIST”
        130 A$=”LIST”+CHR$(13)
        140 LISN A$
        150 CLS:PRINT”OK, NOW YOU ARE READY TO PROGRAM”
        160 PRINT”USING THE WORDS YOU HAVE ENTERED”
        170 PRINT
        180 DRECT
```

Noticed the new command word DRECT. This command has the effect of entering anything you say into BASIC as though you had typed it on the keyboard. If you need to, you can still type whatever you want on the keyboard, but simple programs using the words you just taught the computer can be entered without typing anything! Try this one:

```
        10 A=143
        20 PRINT A
```

To enter this, run the preceding program to teach the computer the right words, then say “one, zero, A, equals, one, four, three, enter, two, zero, print, A, enter”. Now verify that the program is there by saying “list”. Not bad, is it?

The “direct mode” which is entered by DRECT is exited by pressing the BREAK key.

You will notice that certain words were entered as “WORD”+CHR$(13). The CHR$(13) is the same thing as pressing the ENTER key. Any word that has this on the end of it will end a line. When you say “list”, the computer immediately lists the program because a CHR$(13) is on the end of “LIST” in the template. Try listing the templates with TLIST. Notice that all the words that had CHR$(13) added to them are printed with “[13]” on the end of the word. This makes it easy to tell exactly what “control codes” (codes that tell the computer to do something) have been added to a word.

## USING THE REJECTION COEFFICIENT

EARS has a feature that allows you to make speech recognition more or less precise. When you say a word, the computer compares what you said with what is stored in memory. Each word in memory is assigned a “comparison coefficient” during a match which is a measure of how closely the word matches what you just said. The smaller the number, the closer the fit. A number called a “rejection coefficient” can be used to reject words that don't fit closely enough.

The rejection coefficient is set by entering REJ followed by a number 0 to 9, such as REJ2. As you would expect, specifying a coefficient of 0 requires that any match made must be very close. A coefficient of 9 allows almost any match. The best general number to use is 2. If REJ is not followed by a number, the rejection coefficient is disabled.

The only way the rejection coefficient can be made use of is if the variable WO$ is used. A name is passed to the variable provided the matching coefficient is less than or equal to the rejection coefficient. If the matching coefficient is too big, the WO$ variable contains nothing.

Here is an example of the use of the rejection coefficient (it is assumed you have templates already entered in memory):

```
10 WO$=””
20 REJ2
30 LISN
40 MATCH
50 IF WO$=”” THEN PRINT”I DON'T RECOGNIZE THAT WORD”
 ELSE PRINT WO$
60 GOTO 30
```

If TLIST is followed by a number 0 through 9, then only those templates whose matching coefficient is less than or equal to the number will be listed.  This can be helpful in determining which rejection coefficient is best for your application.

## IMPROVING RECOGNITION ACCURACY

You have had enough experience with your speech recognition system now to realize that it is not perfect. Sometimes it just won't recognize a word. There are a number of tricks you can use to improve the systems performance.

First of all, be sure you speak clearly and consistently. Most problems with poor performance can be traced to improper speech entry. Another problem is noise in the background that can interfere with recognition.

Obviously, words that sound similar to each other will be hard for EARS to tell apart. For example, the words “go” and “no” are similar. Words that sound identical but are spelled differently are bound to cause problems such as “right” and “write”. When you make up a template file, be careful to avoid words such as these.

Once you are sure that entry problems are not the cause of poor recognition, you can retrain a word that is giving you problems.
First delete the word with TDEL, then re-enter it. For example, suppose that the computer has a hard time recognizing the word “hello”. Enter

```
        TDEL”HELLO”
```

and then re-enter it with

```
        LISN”HELLO”
```

Another simple trick is to train more than one template of a word. This has a definite advantage over deleting and re-entering a template; the reason is that everyone pronounces a word slightly differently from time to time. If you have more than one template for a given word, then it is more likely that these variations would be covered. The result is that the word is more frequently correctly recognized. Be careful when using this method because the more templates in memory, the longer it takes to find a match.

Averaging is another technique that can improve recognition. The command TAVG is used in conjunction with LISN to produce a template that is the average of two utterances of a word. Here is an example of how it is used:

```
        10 PRINT”SAY HELLO: “
        20 LISN”HELLO”
        30 PRINT”SAY HELLO ONCE MORE: “
        40 LISN
        50 TAVG
```

TAVG averages two templates together--the last one that is stored in permanent memory with the one that is in the temporary holding area. Now the last one in permanent memory is the average of the two templates. The averaging technique has the advantage of taking up less space.

EARS can easily achieve a recognition rate of 95 to 98 percent if you make good use of the above techniques.

## HIGH SPEED OPERATION

Your computer is probably capable of running twice as fast as usual. FAST and SLOW are two commands that are designed to take advantage of this. Be advised, however, that not every computer can run at the faster speed without locking up and erasing everything in memory. The first time you try these commands, be sure you have saved to tape or disk everything that you want to keep.

Enter FAST. Nothing will happen until you enter LISN or MATCH. During template generation or matching the program enters the high-speed mode. What you see is that the screen becomes a jumble of random colors and lines. As soon as the LISN or MATCH is done, the screen should return to normal and BASIC functions normally. If it doesn't, then you should never use FAST. The command SLOW cancels a FAST command.

## COMPATIBILITY PROBLEMS

There is a problem with compatibility between Extended BASIC, Disk BASIC 1.0, and Disk BASIC 1.1 that you should be aware of. The tokens (abbreviations for command words that BASIC uses) are different for the three systems. If you write a program to be used by other systems, save the program using the 'ASCII' format:

```
        (C)SAVE”FILENAME”,A
```

instead of:

```
        (C)SAVE”FILENAME”
```

A program saved in this format will load correctly into any system.

## HOW DOES IT WORK?

How does EARS work? “EARS” Contains a sophisticated chip capable of speech recognition and synthesis (SP-1000). This chip converts information from the microphone into digital data that is processed by EARS. The program listens for the start and end of a word, saving 450 bytes of data per second (up to 5 seconds) of speech. This raw data then is converted into a fixed length template of 112 bytes. If a name was assigned to the template, then the name with the template is stored in an area of memory called the template buffer.

During a match, the last template generated is compared with all the templates in the template buffer. The comparison process is a two dimensional match called “dynamic time warping”. This results in a number that is called the matching coefficient. The program then looks for the smallest matching coefficient and then returns the corresponding word to BASIC.

## A FEW ODDS AND ENDS

EARS holds up to 64 words in the template buffer at a time. You should find this to be more than enough space for most applications.

Be careful when using the BASIC CLEAR command. If you reset the start of protected memory, you could erase the EARS program.

If you wish to delete a template that has a control character in its name, you must first set up a string variable that contains the template name. For example the template name is “LIST” with CHR$(13) on the end (or LIST[13] when listed by TLIST). To delete this template, type in

```
        A$=”LIST”+CHR$(13)
        TDEL A$
```

## SPEECH SYNTHESIS

If you have the “Super Voice”, “The Voice”, or the Radio Shack speech and sound cartridge with the TRNSLATE program from Speech Systems and 64K memory available, you can use speech synthesis with EARS. First insert the speech synthesis cartridge into the Multi-Pak (or use a triple Y-cable, available from Speech Systems) according to the directions that come with it. Now load the translator program into memory:

```
        (C)LOADM”EARS”:EXEC
        (C)LOADM”TRNSLATE”
        FLIP
```

The FLIP Command relocates TRNSLATE into high memory so it won't cause memory conflicts with EARS. Be careful not to use the FLIP command again unless you have just loaded TRNSLATE. Also note that loading TRNSLATE into memory erases any templates you may have stored.

Now you can use the command SAY to say a word. Because of how TRNSLATE uses memory, the SAY command cannot be used in the direct mode. As part of a program, it will say a string variable or string constant (except string arrays) such as:

```
        10 SAY”HELLO THERE”
        20 A$=”THIS IS A TEST”
        30 SAY A$
```

EARS is also capable of high-speed synthesis. Unfortunately, it is not quite as versatile as “Super Voice”. To speak, you have to have word templates in memory. These templates are generated by specialized equipment not currently available for the color computer. Contact Speech Systems for information on the availability of word templates. Supplied with EARS is a program called “NUMBERS”. This program will allow you to say the numbers 0 through 9 without the addition of any hardware. This program works exactly like “TRNSLATE”. Follow these steps to use it:

```
        (C)LOADM”EARS:EXEC
        (C)LOADM”NUMBERS”
        FLIP
```

To say a number, use the SAY command followed by a string containing “0” through “9”:

```
        10 SAY”0”
        20 A$=”4”
        30 SAY A$
```

Please note that you cannot use the SAY command with the WO$ variable. This won't work:

```
        10 WO$=””
        20 LISN:MATCH
        30 SAY WO$           THIS WON'T WORK
```

To do the same thing, reassign the value of WO$ to some other variable, like this:

```
        10 WO$=””
        20 LISN:MATCH
        30 A$=WO$
        40 SAY A$      THIS WORKS FINE
```

## TECHNICAL DETAILS

EARS is located in high memory from BASE address to $7FFF. The BASE address may be determined by entering

```
        BASE=256*PEEK(&H23)+PEEK(&H24)+1
```

There are a number of pointers to important routines within EARS starting at BASE. These routines allow a machine language programmer to interface with EARS.

```
               ORG BASE
        MATCOE RMB 2        matching coefficient of last matched word
        MATWRD RMB 2        pointer to match template word
        REJCOE RMB 2        rejection coefficient
        REFT   RMB 2        pointer to start of template buffer
        REFTE  RMB 2        point to end of template buffer
        GENTEM RMB 2        pointer to template generation routine
        MATCH  RMB 2        pointer to template matching routine
        INIT   RMB 2        pointer to initialization routine
```

The last 3 pointers are relative pinters; that is, you have to add the BASE address to them to get the true address of the subroutine. Here is a brief description of each subroutine and how to use it:

GENTEM
Entry:  variables must have first been initialized by the INIT routine.
Exit:   a new template in the temporary holding area is generated. All
        processor registers are modified.

MATCH
Entry:  GENTEM must have been called already.
Exit:   the temporary template is matched against the reference
        templates and the MATCOE, MATWRD, and REJCOE pointers are
        updated. All registers are modifies.

INIT
Entry:  no entry conditions.
Exit:   All variables for the correct functioning of GENTEM and
        MATCH are set up; all registers modified.

The template buffer is located from $6000 to $7FFF. Each template occupies 128 bytes. This is the structure of each template:

```
               RMB 2        reserved for future use
        WRDLEN RMB 1        word length (0-15)
        WORD   RMB 15       template name
        TEMPLA RMB 118      template data
```

The SAY command causes memory from $6000 to $7FFF to be saved starting at $C000, transfers memory from $8000 to $A000 to $6000, then executes the transferred program at $6005. Upon return, the two blocks of memory are flipped. The end result is that the templates used by EARS are undisturbed and the program is run in its normal location.

Further details on the operation of the SP-1000 and explantion of the algorithm used in template generation and matching may be obtained from General Instruments, the manufacturer of the SP-1000.

## PROGRAMMING EXAMPLES

Here are several examples of speech recognition programs. Study them carefully for tips on how you can write your own programs. Each program assumes that you have already loaded and executed EARS.

The first program allows you to “draw” colors on the screen. In addition to entering the 8 available colors into templates, you also will enter the backspace and enter keys. The backspace will aid you in making changes should you say the wrong color.

```
        10 CLS
        15 TCLEAR
        17 PRINT”DRAW COLORS ON THE SCREEN”:PRINT
        20 FOR N=&H8F TO &HFF STEP 16
        30 A$=CHR$(N)
        40 PRINT”SAY THE COLOR: “A$
        50 LISN A$
        60 NEXT N
        70 A$=CHR$(8)
        80 PRINT”SAY BACKSPACE”
        90 LISN A$
        100 PRINT”SAY ENTER”
        110 A$=CHR$(13)
        120 LISN A$
        130 PRINT”NOW SAY ANY OF THE COLORS”
        140 LISN
        150 MATCH
        160 GOTO 140
```

Next follows a simple program that demonstrates how you can set up a template file to fit an application.

```
        10 FOR N=0 TO 9
        20 PRINT “SAY THE NUMBER: “N
        30 A$=CHR$(48+N)+CHR$(13)
        40 LISN A$
        50 NEXT N
        60 PRINT”PRESS ENTER WHEN YOUR DISK DRIVE”
        70 INPUT”OR TAPE RECORDER IS READY”;Z$
        80 TSAVE”NUMBERS”
```

If you have a speech synthesizer and 64K memory, using the templates generated by the preceding program, here is an example of how you can run a program without touching the keys or looking at the screen. Before running the program, load “SUPERTOK” into memory with (C)LOADM, then enter FLIP. Now RUN the program.

```
        21 REM FOLLOW INSTRUCTIONS IN THE MANUAL TO SET UP YOUR SPEECH
         SYNTHESIZER
        22 REM LOAD TEMPLATES GENERATED BY PREVIOUS PROGRAM
        23 TLOAD”NUMBERS”
```

```
24 DRECT
25 SAY”I AM THINKING OF A NUMBER FROM 1 TO 10”
26 FOR T=0 TO 100:NEXT T
27 N=0
28 SAY”CAN YOU GUESS WHAT IT IS”
29 NUMBER=RND(10)
30 SAY”SAY THE NUMBER”
31 INPUT GUESS
32 IF GUESS < NUMBER THEN SAY”NO THAT WAS TOO LOW”
33 IF GUESS > NUMBER THEN SAY”YOUR GUESS WAS TOO HIGH”
34 IF GUESS <> NUMBER THEN 30
35 SAY”THAT WAS CORRECT”
36 SAY”YOU TOOK”
37 A$=STR$(N+1)
38 SAY A$
39 SAY”ATTEMPTS”
```

## **COMMAND SUMMARY**

DRECT: Allows the user to have “voice keyboard” so that anything that is said is entered into BASIC as though it was being typed. This allows the input of programs, etc. without having to type a single letter.

FAST: Sets the program to perform template generation and matching in the high-speed mode. This is useful when there are a large number of templates in memory as it reduces search time by 50%. Unfortunately this results in loss of the TV display.

FLIP: Used only with 64K systems, this command allows a speech synthesis program to be saved in high memory (above $8000).

LISN: The computer waits for speech input and generates a template. If the command is followed by a string, the template is added to memory and the string is assigned to the template.

FLIP: Is used to allow the use of “Super Voice” with EARS if you have 64K of memory.

MATCH: The last template generated by a listen command is compared with the templates in memory and the word having the best match is printed. If the variable WO$ has already been declared, then WO$ will contain the matched word. If the matching coefficient exceeds the rejection coefficient, then WO$ will be an empty string (see the REJ command).

REJ: Sets the rejection coefficient. When followed by a number 0-9, this number becomes the rejection coefficient. If REJ is not followed by a number then the rejection coefficient is disabled (default condition).

SAY: When a speech synthesis program has been saved in high memory by the use of the FLIP command, that program can be called and the string following SAY will be spoken.

SLOW: Cancels the fast command.

TAVG: Averages the last template in memory with the last input from a LISN command (not followed by a string). This can be used to improve recognition rates.

TCLEAR: Clears the template buffer (deletes all templates).

TDEL: Deletes the last template in memory. If followed by a string, then all the templates with the string name are deleted.

TLIST: Lists the templates in memory. If followed by a number 0-9 then only those templates having matching coefficient less than or equal to the number will be listed.

TLOAD: On a tape system, a filename is not necessary (just as with

CLOAD). TLOAD loads the specified template file from tape or disk (depending on whether you have a disk system).

TSAVE: Must be followed by a filename. On a tape system, all the templates are saved to tape; on a disk system, they are saved to disk.

## **EARS FEATURES**

EARS is a complete speech recognition system to allow you to talk to your computer.

Works in any 32K or greater color computer.

May be used in both a tape or a disk system.

Easy to use in any BASIC program.

Recognition rates of 95% to 98% are typical.

EARS is trained by your voice. Up to 64 words or phrases may be in memory at one time. By making voice prints of your words, and saving them on tape or disk, Your recognition vocabulary is infinite.

EARS is also capable of extremely high quality natural sounding speech with a fixed vocabulary.

Hands Off programming is possible. You can actually speak a BASIC program without ever touching the keyboard.

EARS comes with everything you need including a specially designed noise canceling headset microphone.

When used with our VOICE or SUPER VOICE speech synthesizer, you can actually carry on a conversation with your computer.

Included are many example programs to allow you to both enjoy them as well as learn from them.

Speech that is reproduced comes from both your TV speaker as well as through an audio connector to allow you to connect it to a stereo system for increased intelligibility.

Disk system users may use EARS with a MULTI-PAK or our Y-cable. Those wishing to use a speech synthesizer along with EARS may use our Triple Y-cable.