Development of a low-cost personal computer speech recognition system

by

Gary Lee Ayers

A Thesis Submitted to the

Graduate Faculty in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Interdepartmental Program: Biomedical Engineering
Major: Biomedical Engineering

Signatures have been redacted for privacy

Iowa State University
Ames, Iowa

1993

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Computer speech recognition has been a popular area for study, research, and applications development. Although there are more difficulties associated with developing a useful recognition system than there are for applications involving the related technology of speech synthesis, major computer manufacturing companies as well as smaller non-computer manufacturers have delved into the speech recognition market throughout the 1980s and into the 1990s.

Costs and the present level of recognition technology appear to be holding back the large scale integration of voice recognizers into our society. The ultimate achievement in voice recognition would be the ability of any person to converse with a computer or other machine with recognition capability as if the person was talking to another human. The enormity of this task can be realized when one thinks of how hard it can be for human beings to understand each other's spoken messages. The ultimate speaker independent, connected speech recognition system would require that the recognizer overcome a host of difficulties, such as dialects, accents, distinguishing between homonyms (to, too, and two, for instance), and understanding both grammatical and semantic content of received input. For a voice recognition system to intuitively begin to extract meanings from spoken phrases and sentences, it must have some knowledge and capabilities in natural language processing. This complex task not only involves the science of computers, but also philosophy, linguistics, and semantics. What the computer must do as it solves the problem of understanding speech is to determine the intended motive behind the message, the emotive content that was intended and, from that, the true meaning of what was spoken (Cater, 1984).

If the problems for the ultimate speaker independent, connected speech recognition system are still being addressed and will not be perfected until some time in the future, what does the current level of technology hold for us today? To answer this question, we must first seek an answer to the question of how one can go about identifying those areas where speech recognition might be advantageous. Then a determination must be made as to whether the voice technology used to gain a certain level of efficiency for a task, at a feasible cost, provides enough advantages to warrant its use over the current method of performing the task. Certain indicators can be useful when judging applications to be potential beneficiaries of speech recognition technology (Vaissiere, 1985). One indicator would be the amount of close hand-eye coordination required by a task. If a computer operator has very little time to be distracted by a computer keyboard, then his task might be a candidate for speech input. A second indicator is the degree of mobility required by the computer operator. An example of a task that would require mobility is the job of taking inventory in a large warehouse. Again, it might be more convenient to have alternative methods of input to the computer, other than a keyboard. Another indicator of potential use of speech input is when the computer operator is unskilled and possibly doesn't know how to type, and the reliability of entered data must be high. If any one or more of the above conditions exist in a situation, then speech recognition might be given strong consideration for the task (Vaissiere, 1985). The key is whether the task is made simpler by speech input to the computer. If it is not, then success is not likely.

Another important factor is the commitment made to the use of recognition in a particular application. If a hobbyist is involved, the commitment is made based on curiosity and a desire to have or create something unique and there is normally sufficient motivation to carry out the implementation. For a corporation, this commitment would be extensive because of the man-hours necessary for implementation and the interruption of the current process that is being replaced. Ample motivation and capital would be needed prior to the integration of speech recognition into the process to insure that the implementation succeeds and that the system is accepted (Vaissiere, 1985).

A final consideration is the noise environment into which the recognition system is being implemented. If speech input driven tasks are to occur in environments where the noise level is above 85 to 95 dB, then speech recognition becomes a more difficult task (just as it would for humans) and the chances of successful recognition are minimal (Vaissiere, 1985). NASA has researched the relationship of noisy environments and recognition accuracy. It was generally thought that of the four possible combinations of training and usage: (1) quiet training and quiet recognition, (2) quiet training and noisy recognition, (3) noisy training and quiet recognition, (4) noisy training and noisy recognition, the best combination was the first, followed by 4 and 3, in that order, with condition number 2 providing the least desirable situation for accuracy (Vaissiere, 1985). It was found that total recognition accuracy variance from combination 1 to combination 2, the best and the worst combinations, was on the order of 18%. Later research with voice recognition systems has indicated very little difference in the way these systems respond to the four environments (Vaissiere, 1985).

Voice input typewriters, vocal data entry systems, and speech controlled systems are some application areas to which development efforts have been extended (Vaissiere, 1985). Voice-activated typewriters that would be able to recognize speech from a multitude of different speakers and have a big enough vocabulary to make them practical will most likely not be found until the late 1990's or possibly even the next century. One of the goals of the Fifth Generation Computer project of Japan was a voice-activated typewriter with a vocabulary of ten thousand words and the ability to recognize the voice patterns of hundreds of users without retraining (Vaissiere, 1985).

Another area which will probably see a shift toward voice input is the automated business office (Vaissiere, 1985). Accountants and certain types of office workers who view column after column of numbers and then input these numbers into a machine would certainly benefit by being able to enter the data into a computer by speech. Other current and potential uses are the performance of quality control in a production assembly line, the reciting of written dimensions into a computer by a draftsman while visually reviewing blueprints, and systems in the home for balancing checkbooks or taking inventory of household goods (Vaissiere, 1985).

Finally, speech recognition could be used in speech controlled systems in aircraft, spacecraft, automobiles, elevators, hospital rooms, and in aids for the physically handicapped. Voice input has been tried and used in a variety of situations for the disabled who have or retain some vocal ability. This use should continue and expand as speech technology progresses. Speech recognition could allow control of devices that were either very difficult or impossible to control otherwise. Numerous projects and studies have been reported in which speech recognition has been applied to help those who are disabled. Thus, this expanding area of technology could be a great boost to a number of the approximately 500,000 Americans and the many more throughout the world who must adjust to physical disabilities (Vaissiere, 1985).

LITERATURE REVIEW

Voice Recognition Technology

History

Interest in automatic speech recognition began about 35 years ago with the increased availability of electronic hardware to perform spectral analysis of signals. There were two primary reasons for the early interest in speech recognition: 1) a desire for automatic transcription of the input speech into stenographic-like symbols of phonemes; and 2) direct identification of words or phrases to command or control machines by voice. The first purpose would result in the so-called phonetic typewriter and the second would bring into being an isolated word speech recognizer. This first phonetic typewriter, also called the "phonotograph" was meant to translate the voice input directly into short-hand-like symbols, to eventually be changed into words by a reader. Subsequent phonetic typewriters were supposed to transcribe speech input directly into phoneme-like segments (Vaissiere, 1985).

Most early word recognition systems involved the concept of template matching, in which the frequency spectrum of the incoming signal was matched against a set of standard spectrum patterns for each phoneme. The systems were based on matching whole input patterns against expected word templates received from prior training samples. These devices recognized digits spoken by one individual at a normal rate with between 95 and 97 percent accuracy (Vaissiere, 1985). Other approaches were proposed at this time, including one based on the principle of distinctive features (features-based method). This approach involved making several binary classifications based on different features of the input speech signal.

Interest in the speech field was renewed around 25 years ago when the influence of digital computers on business and industry was in the initial stages of proliferation (Vaissiere, 1985). The prospects of entering data, giving commands, or retrieving information by voice input attracted attention once again. Computers also opened up the means for applying a wide variety of digital signal processing techniques and for trying expanded, complex recognition algorithms. The first isolated word recognizer using a computer appeared in the early 1960s, along with proposals for continuous speech recognizers. Improved algorithms were continually being proposed. New technology continued to be introduced, such as dynamic time warping techniques which compensate for the differences in the duration of input words. The ARPA (Advanced Research Project Agency) project from 1971 to 1976 sparked much activity in the field of continuous speech systems, although many scientists feel this project did not contribute as much to state-of-the-art speech sciences as it did to artificial intelligence (Vaissiere, 1985). Technology is at a level such that successful speaker-dependent isolated word recognition systems are in use today in industry and offices. Some systems have been expanded to handle the recognition of up to three connected words, to recognize sentences composed of words separated by pauses, or to recognize a very limited number of words spoken by those unknown to the system. Recognizers are now available that allow for continuous or fully connected speech recognition without the associated pauses usually necessary in discrete word recognition systems (Vaissiere, 1985).

Speech characteristics

Circumstances which prevent human understanding of spoken words are the same factors that affect a listening computer's ability to understand speech (Cater, 1984). Therefore, understanding voice recognition technology requires knowledge of the properties of speech.

The features of speech communication and their associated semantic content are extremely complex. As a spoken message is generated, a number of events must occur almost simultaneously. Figure 1 (Cater, 1984) displays a simplified diagram to show the speech process.

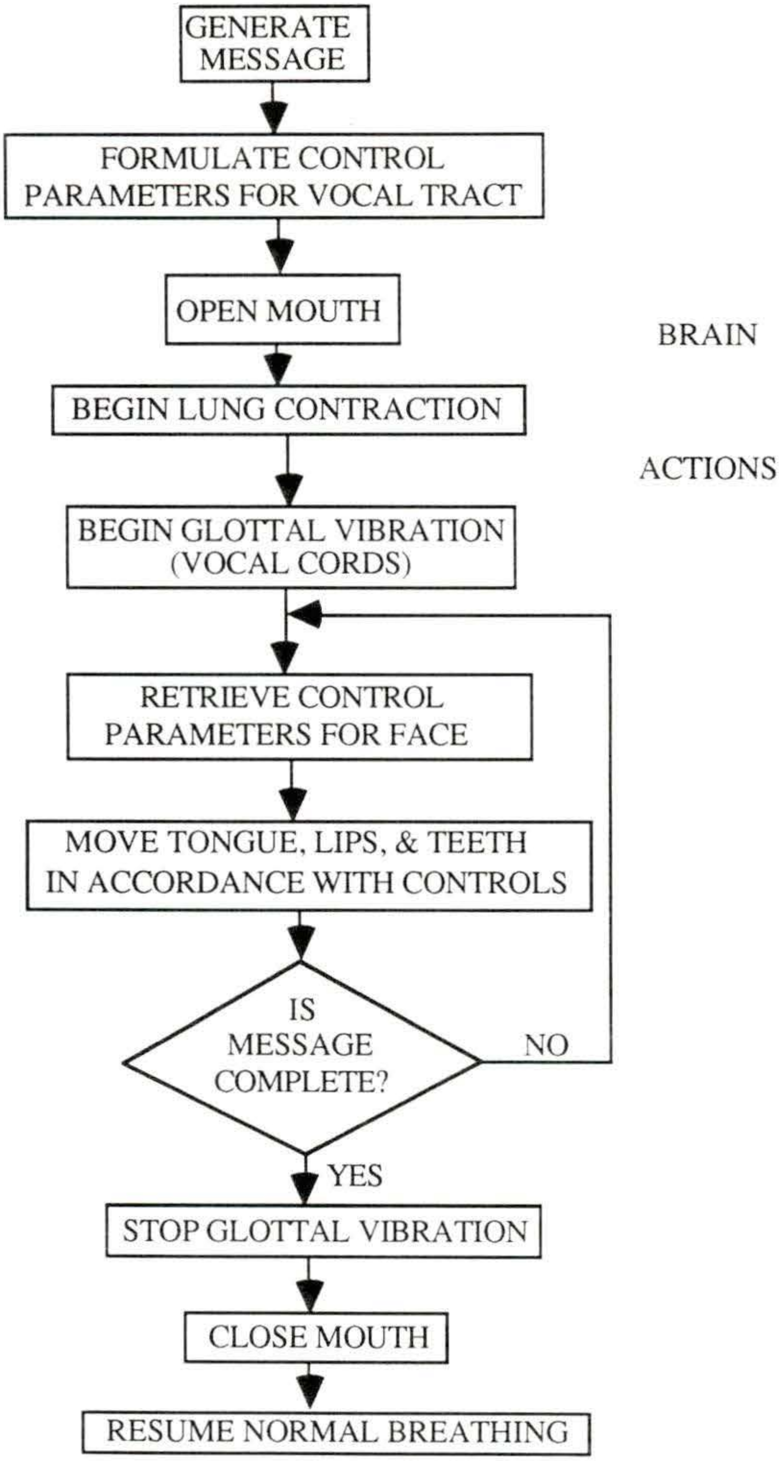

Figure 1. Flow chart of the speech process (Cater, 1984)

Although the order in which some of these steps occurs may be debated, these components will occur sometime during the speech event. The true event order may depend on the particular word being spoken. Certain words have a nasal beginning; in this case the mouth is not opened until after the lungs have contracted and glottal vibrations have begun with the first sounds coming through the nose (Cater, 1984).

Linguists refer to the close to 40 distinct sounds that make up our language as phonemes. These comprise a set of distinguishable, mutually exclusive, speech sounds that are present in almost any spoken language, shown in Table 1 (Cater, 1984). The range of frequencies generated during vocalization distinguishes between the independent phonemes of the General American dialect. Phonemes can be categorized into special groups, such as the continuants, based upon spectral characteristics. Because of lesser vocal tract motion, the continuants have a more stable and sustained frequency spectrum throughout their production. Vowels, semivowels, nasals, and fricatives are included in this group. The plosives and the glides, the other classes of phonemes, are more dynamic sounds that normally couple to surrounding phonemes in a manner resembling diphthongs (Cater, 1984).

Diphthongs are speech sounds characterized by much vocal tract motion when coupling two phonemes together, and they occur as one goes from phoneme to phoneme during speech. If we spoke phonemes without any attempt to join the sounds, there would be no diphthong, but normal talking produces many of these diphthongs (Cater, 1984).

A great difference exists between the frequency spectrum produced from a glottal vibration and that produced outside of the vocal tract. The glottal vibration is a vibration in the vocal cords that initiates speech generation. The glottal pulse spectrum is composed of the pitch period harmonics, the fundamental range of frequencies produced by the vocal cords.

Table 1. General American phonemes (Cater, 1984)

Minimum Phoneme Set From General American Dialect		
Phoneme Type	**Phoneme**	**"as in"**
Vowels*	ah	father
	ae	tap
	aw	talk
	ā	bay
	eh	step
	uh	run
	ee	beep
	i	lift
	oh	tone
	$\overline{oo}$	moon
	oo	book
	er	stir
Consonants		
Fricatives (Voiced)*	v	very
	th	there
	z	zebra
	zh	beige
Fricatives (Unvoiced)*	f	fast
	th	thing
	s	seek
	sh	show
	h	hit
Plosives (Voiced)	g	get
	d	dither
	b	base
Plosives (Unvoiced)	k	cat
	t	two
	p	poke
Nasals*	n	no
	m	me
	ng	ring
Glides	y	you
	w	will
Semivowels*	l	last
	r	real

*Continuants

This glottal frequency spectrum, by itself, is not very useful as a speech signal. When the frequency spectrum is taken from speech recorded outside the vocal tract, near the lips, it gives a different appearance and is said to carry true speech information. This occurs because of the physical effect of resonance in the nose, throat, and mouth (Cater, 1984). The speech spectrum at this point shows three primary frequencies of resonance, called formant frequencies. Normal human speech has been shown to have more than three main formant frequencies, but those frequencies above the third formant contain relatively little energy and may be effectively disregarded when considering the total spectrum. Of the three primary formants, the lowest frequency is formed by the throat resonance, the next formant is produced by the nasal resonator, and the highest frequency formant is identified with the mouth (Cater, 1984).

The keys to reliable speech recognition are the identifiable categories in which the relationship and ratio of the formants remain. This is true even though the precise positioning of the formants can vary from person to person. The position of the formants in the speech spectrum can be directly related to individual phonemes. This is an important link between the spectrum and any actual information contained within the spoken message that might be used for speech recognition (Cater, 1984).

Analysis of frequency spectra for the speech of a large number of randomly chosen speakers has been done numerous times by speech researchers. One such compilation is given in Table 2 (Cater, 1984). This correlates the spoken phonemes and the related formant frequencies, and indicates resonant frequency ranges specific to each semivowel and nasal. Speech recognition depends on the fact that none of the sounds in the table shares exactly overlapping resonances. Table 3 (Cater, 1984) has the major vowel sounds with the accompanying formant frequency resonances. The vowel parameters, like the previous consonants, have been shown to be mutually exclusive sets.

Table 2. Results of a frequency spectrum analysis of speech (Cater, 1984)

Characteristic Resonances of Nasals and Semivowels			
Sound	**Throat Resonance (Hz)**	**Nasal Resonance (Hz)**	**Mouth Resonance (Hz)**
l	250-400	600	2000-3000
r	500-700	100-1600	1800-2400
n	200-250	600	1400-2000
ng	200-250	600	2300-2600
m	250-300	600	900-1700

Table 3. Major vowel sounds with formant frequency resonances (Cater, 1984)

Formant Resonances of Vowel Sounds			
Vowel	**F1 Resonance (Hz)**	**F2 Resonance (Hz)**	**F3 Resonance (Hz)**
ee (eat)	210-330	2230-2350	2950-3070
i (bit)	330-450	1930-2050	2490-2610
eh (bet)	470-590	1790-1900	2420-2540
aae (bat)	600-720	1660-1780	2350-2470
ah (top)	670-790	1030-1150	2380-2500
aw (ball)	510-630	780-900	2350-2470
oo (book)	380-500	960-1080	2180-2300
$\overline{oo}$ (moon)	240-360	810-930	2180-2300
uh (tug)	580-700	1130-1250	2330-2450
er (nerve)	430-550	1290-1410	1630-1750

Another important feature of the spectral speech formants is their amplitudes. As the glottal speech spectrum decays almost exponentially, the corresponding formants produced by the resonances of the vocal tract decay in a similar manner. Therefore the amplitude of each of the three formant resonances is not equal to the other two. Generally, the first formant will be the strongest of the three, the second formant resonance will be one-half to one-tenth of the first resonance, and the third formant might be from one-half to one-thirtieth the amplitude of the second resonance (Cater, 1984).

Two classes of phonemes, fricatives and plosives, are produced by vocal tract occurrences other than glottal vibrations. The major characteristic of these phonemes is a rapid burst of air through a constriction, such as the lips, teeth, or tongue. The fricatives sustain a high frequency noise characteristic that spreads over the spectrum from 3 kHz to as high as 30 or 40 kHz. The fricatives do not usually display formant frequency resonance since they are initiated toward the front of the vocal tract, at the lips and teeth (Cater, 1984).

The plosives differ from the fricatives as they are noncontinuant but still characterized by high frequency noises generated from a constricted air passage. One of the main qualities of these phonemes is the absence of speech right before their utterance. This is obvious when pronouncing plosives such as "p", "t", or "d." If a spectrogram of a whole spoken sentence was viewed, the plosive phonemes would be easy to identify. The preceding zero energy characteristic is the identifier as that characteristic does not usually occur between words in natural connected speech (Cater, 1984).

Speech is a dynamic phenomenon and this must be considered when attempting speech recognition. Speech analysis by frequency discrimination must take into account this dynamic nature, as normal speech does not consist of single continuant phoneme words. If a recognition system is to accept more than a few single phoneme words, it must be programmed to accept dynamic spectral input (Cater, 1984). If such a system is to work reliably, it must not only have the capacity to recognize the main phonemes of the word, but also have the ability to recognize the coupling diphthongs which bind these sounds together.

Other variables that enter into the analysis of dynamic speech are pitch frequency, the 50 millisecond pause produced by the plosives or stop consonants, and the stresses or accents that occur on certain syllables of words. Although with pitch variation there is little change in the formant position in the frequency spectrum (because the shape and volume of the vocal tract are not changing), it does carry contextual information. The pauses that are produced in the pronunciation of the plosives make it impossible for a recognition system to separate words in continuous speech by looking for these pauses (Cater, 1984).

Variations in normal speech are also a problem for recognition. A person's speech changes when the speaker has a head cold or allergy reactions. These problems primarily affect the nasal passage, making it difficult to pronounce the nasal phonemes (Cater, 1984). The vocal cords can also be affected with ailments such as laryngitis, which mainly affect the pitch frequency of the afflicted person's speech (Cater, 1984).

There are also external factors that cause difficulties for speech recognition. These include room acoustics, ambient noise, and microphone quality. Room resonances may interfere with the speech signal and create a condition known as "room presence." External noise, including bursts of air from the mouth itself, creates the possibility of interference with the speech waveform. The best solution to these problems presently is the use of a high quality noise cancelling microphone and locating it near the speaker's mouth. To deal with the small bursts of wind from the spoken plosives, the microphone should be covered with an acoustically transparent material which dissipates the wind velocity (Cater, 1984). The microphone that is marketed with the Covox speech recognizer used for this project is covered with such a material. Covox recommends positioning the microphone about one inch from the lips and slightly to one side (Covox, Inc., 1985).

Technology of the past decade

International Business Machines Corporation (IBM) has done much work in the area of voice recognition. Among their products is a system designed to conserve memory while recognizing human speech and placing the words spoken onto a screen. This speech system was demonstrated in April 1986 at the Thomas J. Watson Research Center in Yorktown

Heights and again at the national Computer Conference in Las Vegas, Nevada (IBM Information Systems Group, 1986).

This experimental system was developed at the Watson Research Center and used a specially built IBM Personal Computer AT. The system was purported to have a 95% transcription accuracy and base vocabulary of 5000 words (IBM Information Systems Group, 1986). It had undergone some experimentation in office settings to determine its adaptability to different speakers and varying background noise.

The main challenges that faced IBM researchers in improving this system included the following (IBM Information Systems Group, 1986):

1) eliminating the need to pause between words

2) increasing the vocabulary size

3) improving the system's resistance to background noise

An example of an applied computerized speech recognition system is EARS (Entry to the Anesthesia Record by Speech), a system developed to simplify data entry for automated anesthesia record keeping at the Department of Anesthesiology, University of California, San Diego (Smith, 1986; Sarnat, 1983). Experimental speech recognition was used as a means of input to the automated anesthetic record (AAR) because of the ever-increasing demands for record keeping and the need to record some of the data at inopportune times, such as when the patient needed attention. EARS was developed to overcome some of the disadvantages in entering these data to the computer by keyboard or other manual means (Sarnat, 1983).

EARS had a vocabulary of about 350 words, with about 120 syntax rules. The syntax created some flexibility. For instance, "monitor precordial stethoscope" and "monitor stethoscope precordial" were equally valid. Another example is "drug morphine four" and "drug morphine four milligrams IV;" omitting the unit and route of dosage implied defaults and simplified most drug entries (Sarnat, 1983).

Accuracy of the speech recognition process was important because the error correction procedure made errors very costly in terms of over-all throughput. In a preliminary study of computerized speech recognition for application to anesthesia record keeping, recognition accuracy was measured on a few large subsets of a 240-word vocabulary. Users trained five times and then recognition was attempted several times. Overall accuracy in these studies was 95.9% ranging from 90% to 99% for individuals (Sarnat, 1983). However, the results were not obtained from the EARS system and therefore cannot be applied to it. The performance of EARS in the operating room seemed satisfactory, but its accuracy was not measured. It was intended that the redesign of the EARS system would include the capability to measure accuracy and other performance statistics during use (Sarnat, 1983).

Applications for the Physically Disabled

There have been numerous attempts to apply voice recognition technology to the needs of the physically disabled in recent years.

Cohen and Graupe (1980) developed a prototype, microprocessor-based, speech recognition and control system for rehabilitation applications. It was designed to be an interface between the disabled person and the different devices that enabled more mobility and functionality. The system recognized isolated words which easily accommodated the short commands required by the application. It responded automatically by activating controls for the various assist devices.

The following design criteria were imposed on the system (Cohen and Graupe, 1980):

1. Small size, low weight for attachment to a conventional wheel chair.
2. Low power consumption so as not to drain wheel chair's batteries.

3. Vocabulary size of 100 commands to accommodate the different tasks.

4. Recognition time fast enough to facilitate a one second interval between commands.

5. Recognition accuracy requiring a maximum substitution (false recognition) error of one percent and maximum rejection (no recognition) of ten percent. The restriction on substitution errors was due to the high accuracy necessary for wheel chair control.

6. Flexibility in additions to and changes in the vocabulary in order to accommodate special assist devices and functions for specific users.

An initial session was required to train the system for the particular individual. The system was designed to provide control over four tasks, running an electric wheel chair, operation of a touch-tone phone, activation of an electric typewriter, and operation of an Environmental Control Unit (E.C.U.). This unit permitted activation of a variety of devices, including a television, radio, light, and alarm. The system used the MIPROC-16 microcomputer, a 16-bit machine. Recognition accuracy was somewhat less than what has been demonstrated by more elaborate systems, but those systems didn't meet the constraints imposed by the rehabilitation application (Cohen and Graupe, 1980).

Devaney and Rastgar (1983) also attempted speech recognition for wheel chair control. Their system was tested with a vocabulary consisting of the words NO, YES, STOP, START, LEFT, RIGHT, NORTH, SOUTH, EAST, and WEST. Three different protocols were used. Within the first protocol, each word was entered just once during training. Each word was then tested ten times with the result of an average recognition rate of 49%.

Within the second protocol each word was entered ten times in training and also tested ten times. This resulted in an average recognition rate of 58%, a 9% improvement over the first test (Devaney and Rastgar, 1983).

In the third protocol, the data from the second were maintained to test the system in a multi-user environment. Testing was done by two male and two female subjects with a resultant average recognition rate of 20% for both cases. This low rate could be explained by the diversity in human speech pattern between the trainers and the actual users. The fact that accuracy rates were the same for both male and female indicates the success of the pitch normalization process (Devaney and Rastgar, 1983).

The project used a time averaged model for recognition of isolated words. The results suggested that the model would be suitable in a single user/trainer situation, which would meet the needs of the wheelchair user. Another set of commands was suggested, including FORWARD, REVERSE, LEFT, RIGHT, ON, OFF, HALF, FULL, (SPEED), START, and STOP, to provide for wheel chair control (Devaney and Rastgar, 1983).

Grooms (1986) developed applications for a speech input system that was used as a vocational training tool. For this project, individuals were chosen to participate on the basis of being developmentally disabled and having severe hand dysfunction. The participants were then required to understand their work and perform in a manner that would likely be acceptable to a prospective employer.

Grooms made use of a speech recognition product manufactured by Texas Instruments (TI) Incorporated called the Texas Instruments Speech Command System (TISCS), which included the hardware and software necessary to allow voice recognition using linear predictive coding (LPC). The hardware chip was not a specialized LPC chip; instead it was the more general-purpose digital processing chip made by TI. Since the speech recognition algorithm is changeable by software, this hardware should not become obsolete as new and better speech algorithms appear (Grooms, 1986).

The TISCS vocabulary can be defined in the following manner. When a system operator speaks, the computer responds with an action if recognition occurs. With the TI speech system, the action can be a single keyboard stroke or a series of keystrokes. These keystrokes are defined by the user prior to beginning the application and are accomplished through a controlling program called Vocabulary Manager which creates and then manages the vocabulary of utterances and actions.

A single vocabulary for the system can contain no more than 60 utterances. When the system is in use, the vocabularies that have been formed are loaded into random access memory (RAM). However, the speech hardware only acts on one vocabulary, the active vocabulary, at a time. When this vocabulary is in use, the computer responds to a spoken word or phrase by attempting a statistical best-fit to the utterances that are stored in memory. If a poor fit results, no recognition occurs, and the computer does nothing. If the statistical best-fit exceeds a certain threshold value, the action that was paired with the spoken phrase is carried out. Important concepts investigated were vocabulary definitions and switching patterns (Grooms, 1986). If an application required utterances and actions from more than one vocabulary, a means of switching to that other vocabulary was accomplished without hanging up or terminating the application program.

The previous efforts represented attempts to respond to needs of the physically disabled user by development of speech recognition systems or applications that met a particular need or combination of needs. Each system was also subject to design constraints based on the demands of the application and the constraints of the particular technology used.

This present project was also attempted to respond to the needs of certain users, those who desire access to a personal computer but also desire or need an alternative to the traditional methods of input such as the keyboard. The major constraint applied in this case was the use of lower cost technology. The requirement was to supply the same basic functioning as the TISCS but at reduced cost. This would result in a system useful at home and preferably in some areas of employment.

Because of the nature of the attempted application in this project, it was not expected that the system meet a small size or low weight requirement such as the criteria imposed on a wheelchair application. However, vocabulary size or number of vocabularies available, recognition time, and recognition accuracy were expected to be sufficient so as to not create excessive hindrances in operating a word processing program or other attempted programs.

PURPOSE OF PROJECT

Speech recognition technology may be still in its growing stages but, as previously discussed, there has been no lack of initiative in applying what is presently available to the needs of the physically disabled. A number of computer manufacturers have developed recognition systems for their own computers and compatibles. Small microcomputer and microprocessor based systems have also been developed when greater portability was a necessity.

The overall purpose of this project was to gain knowledge about some specific products and accompanying concepts in order to provide a useful computer system for those desiring access to the computer while bypassing the computer keyboard. This knowledge was then applied to the development of specific programs to enhance the computer's value as an immediately useful assistive device. With this overall purpose in mind, the following were more specific objectives:

1. To develop a personal computer based system utilizing speech as the means of input to the extent that it would be of real benefit to the user and not create overwhelming hindrances.

2. To develop a system that, for what it could offer, would be cost advantageous over other recognition systems.

3. To implement the above as a Commodore 64 based system which would afford the possibility of using hundreds of software products written for this computer. There would also exist the potential for conversion to the Commodore 128, an enhanced version with more power and memory capacity, which can also operate as a C64 machine. The Commodore 64 and 128 are relatively inexpensive personal computers that are still widely used.

4. To gain a greater understanding of the Commodore operating system and memory configurations so that the machine could eventually be fully exploited in its capabilities for speech input applications.

5. To develop a voice-operated system dedicated to running two particular applications, word processing and environmental control, which would serve as initial demonstrators of the potential for Commodore programs to respond to speech input. One application, environmental control, would demonstrate the ease of creating and running a user-written BASIC application program. The other, word processing, would demonstrate the interface necessary to allow off-the-shelf user-purchased software to also be operated by voice.

There are more sophisticated voice units (hardware and software) available which give the programmer more options and greater capacity for converting applications to voice input. But the cost can be prohibitive. It is important to note also that the applications chosen for this project are of immediate practical benefit to the disabled, and word processing capability could be attractive to a potential employer of a disabled person.

SYSTEM COMPONENTS

System Overview

The system used was composed of a Commodore 128 (C128) personal computer (used in C64 mode), a Commodore 1571 disk drive, a Commodore 1750 512K RAM Expansion Unit (REU) or Module (REM), and a VOICE MASTER speech and music processor manufactured by Covox, Inc. of Eugene, Oregon. The BSR Controller, a feature of the ADC-1 Data Acquisition System made by Remote Measurements Systems, Inc., was attached to the C128 user port to enable environmental control. The ADC-1 system was not a requisite item; another compatible controller system could have been substituted, depending on availability and the features desired by the user.

The Computer

One of the advantages of the Commodore 64 (C64) and peripherals is the low cost relative to other personal computer systems. The Central Processing Unit for the C64 is the 8-bit 6510 microprocessor which has an internal architecture and instruction set identical to the MOS Technology 6502. The C64 has been the target of hundreds and possibly thousands of existing software products; with the availability now of expansion RAM products for the Commodore 128 (64 mode as well as 128 mode) and the C64, these same C64 programs are candidates for speech input on either machine. This requires no adaptation to the software interface to accommodate the particular expansion RAM (1764 for the C64) in use. The C128's natural 128 mode can also become a host for voice input applications; Covox's voice recognition software can potentially be ported to the C128 without a change in the speech processing hardware. Machine language monitors or assemblers are available for straightforward entry of assembly language programs and viewing of RAM locations or ROM routines.

The 1571 Disk Drive

The 1571 disk drive provides double-sided, double density recording for a 339K storage capacity per disk (169K per side). The 1571 also offers special high-speed burst commands, used for machine language programs, which transfer data several times faster than the standard or fast serial rates. However, for the C64, the 1571 disk drive functions at standard 1541 speed only, that which meets Commodore 64 compatibility requirements.

The VOICE MASTER

Hardware

The VOICE MASTER produced by Covox, Inc. of Eugene, Oregon actually has three different functions - speech recognition, speech recording and playback, and voice production of music, in which the computer either plays or composes music that an individual hums or whistles. For this project, only the speech recognition function was used.

The VOICE MASTER comes as a complete hardware and software system ready for accomplishing the functions previously mentioned. The hardware includes the digitizing/processing system enclosed in a hard case, a headset with microphone and earphone, and all necessary cables. The VOICE MASTER was designed to plug into the rear control port (port number 2) on the Commodore 64 or 128. Instructions are given for adjusting the gain control and properly calibrating the instrument to the individual computer. The VOICE MASTER also has an audio input jack which, when connected by cable to one of the audio outputs from the computer, allows for listening with the headset earphone to any of the computer's sound outputs. The earphone connects to an earphone jack located immediately next to the microphone jack on the VOICE MASTER.

Recognition concepts

A template for a spoken phrase is formed when the spoken phrase is reduced to a set of characteristics and each characteristic is transformed to a graphical variation of time. Several different phrases formed into templates make up a vocabulary or catalog which can be matched, one by one, against the template of the unknown word. Recognition results from the best fit of the unknown template with one in the vocabulary. To accomplish this, the unknown must be compared with each template in the catalog. Recognition refusal occurs when no comparison yields a match defined as acceptable by the user through the VOICE/RECOG software. If the comparisons of two or more words with the unknown phrase each produces an acceptable match, then a decision is not made and this condition is made known to the user.

These facts apply to almost all types of pattern recognition, including speech and vision. Differences in different pattern recognition algorithms arise in the nature of the characteristics used to form the templates and in the error criteria used to measure closeness of a match. If some clues exist early in the recognition processing which narrow down the choices, it won't be necessary to complete the processing with every member of the vocabulary. This process that narrows choices sequentially is often called a tree pattern search. The VOICE MASTER's recognition algorithm involves a limited tree search in that the process for a particular template may not be completed due to a poor match being indicated before the process is finished. When this happens, the process jumps to the template. Another form of tree search is applied with the use of sub-vocabularies of words. The spoken phrase in this case is only searched for within the sub-group, even if it happens to be elsewhere in the larger vocabulary.

Voice pattern shapes are measured at 20 millisecond intervals and the individual pattern is assigned a set of 8 numbers. Phrase length determines the total number of 8-number sets designated to a pattern. A second operation involves time normalization in which all the patterns are reduced to 12 of these 8-number sets. Templates for the candidate set and templates for the unknown phrase are normalized in the same manner. The quantity of numbers in each template is 96 (12 x 8) with four additional bytes of RAM used for each to store the memory location data.

Taking the difference between corresponding numbers of the unknown phrase and each template of the vocabulary or group begins the process of matching. The sum of the differences in root-mean-square magnitudes forms a computed closeness score. If various parts of the pattern are relatively more important than others, then weightings are applied to those parts. The lowest score then points to the best match for the unknown. A large lowest score might indicate no match - a threshold value must be decided upon to determine if that score is a probable match or if it is a probable mismatch. Two or more low scores indicate uncertainty and a difference value between the candidates must be established so that one sufficiently lower than the others is considered the probable match.

A variation of dynamic time warping is used in the VOICE MASTER algorithm, to account for minor differences in the way the word is pronounced. The characteristics or cues as functions of time can be moved slightly; thus the template could be compared to a rubber sheet. Words should continue to give a good match even if a syllable is stretched in comparison to the way the original trained pattern was made, as discussed earlier.

Software

The software marketed with the VOICE MASTER includes a main DEMO program which demonstrates its different features. There are other application programs written in Commodore BASIC that are designed to show the user some of the VOICE MASTER's capabilities and to aid the user in exploiting them.

The most important programs are the ones written in 6502 machine language to facilitate the recognition and recording/playback functions. These programs have characteristics which place them into one or more of the following categories:

* Enable recording/playback - VM3.0, VM3.0N, VM3.0R, PLAY/FIND

* Contains FFIND (FFIND is a software routine which loads in a saved speech vocabulary from disk faster than the standard method) - VM3.0

* Enable speech recognition - VR3.0, VM3.0R, VOICE/RECOG

* Allow both recognition and recording/playback - VM3.0R

* Contain wedges (a wedge is a new programmer defined BASIC command which supplements the Commodore BASIC command set) - VM3.0, VM3.0N, VR3.0, VM3.0R

VM3.0R enables both the recording/playback and recognition features.

Covox created wedged BASIC commands in most of its software to simplify use of the VOICE MASTER for users who are not machine language programmers. These new programmer defined commands allow a user to write a speech input application in simpler BASIC without having to learn how to access machine language programs from a BASIC program. A technique to create wedges and also minimize extra execution time can be executed by specifying a character (@ for example) that will occur before any new wedged-in commands. The Kernal (operating system) routine CHRGET, used by BASIC to get each program character or token, is preempted by a new programmer substituted routine which will search for the special character. If the special character is not present, control is passed to the normal BASIC CHRGET routine. If the character is present, the new command is accepted and executed by a machine language program written by the programmer. Covox created wedges, but did not utilize this technique of specifying a special character for each new command.

Therefore the Covox programs with wedges use much more memory than the VOICE/RECOG (enables speech recognition) and PLAY/FIND (enables speech playback) programs which do not. For this reason, VOICE/RECOG was the only Covox program used to enable voice input for this project. It is easily accessed from either machine language or BASIC.

The wedged in commands allow statements that enable speech input to be easily written into BASIC programs. A summary of these commands implemented in VR3.0 and VM3.0R, and a description of the functions of the VOICE MASTER software follow:

1. TRAIN N. This command constructs a template using 96 bytes of memory for the word N. N can be computed as well as specified directly. Recognition programs reserve space for a vocabulary of 32 words; N is in the range of 0 to 31. Retraining the same phrase N creates an average template for that N. TRAIN can be terminated by the RUN/STOP key.

2. BLANK N. This clears the template memory space for word N in the same range, 0 to 31. BLANK 255 clears all 32 templates.

3. RECOG n. With RECOG execution, a program waits for voice input. It then compares the input word to the templates stored in memory. If n is 0, then all 32 templates are scanned and compared to the unknown input. Otherwise, n can be specified as 1 to 4, which designates the scanning of one of four sub-vocabularies as follows:

   * RECOG 1 - scans template numbers 0-7

   * RECOG 2 - scans template numbers 8-15

   * RECOG 3 - scans template numbers 16-23

   * RECOG 4 - scans template numbers 24-31

Any combination of 2 or 3 sub-vocabularies can be designated in the command by RECOG n1,n2 or RECOG n1,n2,n3. For example RECOG 2,4 scans templates 8-15 and 24-31 and compares them to the input word. The number corresponding to the best fit word is placed in memory location 151 ($97 Hex). The recognition mode can be exited at any time before an attempt to recognize an input word with a push of the computer keyboard "Q" key by the user. If this occurs, VOICE/RECOG places the value 253 in location 151. A voice input application program can then detect that the recognition mode was stopped by the user by checking location 151 for that value.

4. MAX y. The MAX command sets the maximum template to input word matching score or threshold allowed for a recognition match to occur. No recognition occurs if all templates' scores exceed this value y. If this type of recognition refusal happens, VOICE/RECOG places the value 255 in location 151. A programmer can use this information to determine the next action a voice input application will take. If the value y is not exceeded by one or more templates, the template having the lowest score "wins" and becomes the match with the input phrase. The range for y is 0-700 with a default value of 450.

5. MIN x. This sets the minimum template score difference so that recognition refusal takes place if the difference between two template scores is less than this minimum value x. When this type of refusal occurs, the software puts the value 254 in location 151 which, again, can be used to determine application program flow. The range for x is 0 to 100 with the default value being 50.

6. TPUT"filename",8. TPUT saves a set of 32 templates from main memory to floppy disk.

7. TFIND"filename" ,8. TFIND brings a previously saved vocabulary of 32 templates into main memory from the disk. Main memory location 151 ($97) has been mentioned

as an important location used by the recognition programs. Other addresses in page 0 of main memory are also utilized but 151 is the most important to the user or programmer. There are three areas of memory which contain various VOICE MASTER programs. The major part of each of the machine language programs is contained in a 4K free RAM block from $C000 to $CFFF which is between the BASIC and Kernal ROM. The two programs which activate only speech recognition are exclusively within this area. A general work area for speech recording is the 8K RAM bank behind the Kernal ROM. Some numerical data for recognition and the template storage area are behind the BASIC ROM.

VOICE/RECOG, the recognition program without wedged-in BASIC commands, was designed by Covox to be a program of minimum length for embedding in a user's machine language program to provide voice recognition capability. It was the only Covox speech program used to complete this project. VOICE/RECOG can be accessed from BASIC with the Commodore POKE and SYS commands rather than the Covox wedged-in commands mentioned previously. The manner in which POKE and SYS can be used to invoke the recognition subroutine highlights important memory locations used in this recognition program. POKE and SYS were utilized as follows:

TRAIN N was replaced by:	POKE 151,N : SYS 49920	(49920 is $C300)
BLANK N was replaced by:	POKE 782,N : SYS 49932	(49932 is $C30C)
TPUT was replaced by:	SYS 49926"filename",8	(49926 is $C306)
TFIND was replaced by:	SYS 49929"filename",8	(49929 is $C309)

Replacing RECOG was slightly more complex. The ruling location is 49935 ($C30F) and initially contains a null. When this is the case, the entire 32 templates are scanned four times. Two to four lines of code will modify this, depending on the choice to scan 1, 2, or 3 sub-vocabularies. The number 255 must be poked in on the last line so that the scanning stops. The following are examples:

RECOG n can be replaced by:

POKE 49935,n	($C30F)
POKE 49936,255	($C310)
SYS 49923	($C303)

RECOG n1,n2 can be replaced by:

POKE 49935,n1	($C30F)
POKE 49936,n2	($C310)
POKE 49937,255	($C311)
SYS 49923	($C303)

RECOG n1,n2,n3 can be replaced by:

POKE 49935,n1	($C30F)
POKE 49936,n2	($C310)
POKE 49937,n3	($C311)
POKE 49938,255	($C312)
SYS 49923	($C303)

If n is equal to 0 or if no value is poked into that location, all 32 templates are scanned once.

The maximum and minimum threshold values require POKE statements:

MIN x:	Low byte to location	45351	($B127)
	High byte to location	45352	($B128)

MAX y: Low byte to location 45353 ($B129)

High byte to location 45354 ($B12A)

For example, if y in MAX y is 450, then the command would look like this:

POKE 45353,194

POKE 45354,1

since 450 in binary is:

0 0 0 0 0 0 0 1	1 1 0 0 0 0 1 0	= 256 + 128 + 64 + 2 = 450
High Byte ($01)	Low Byte ($C2)	

Similarly, if x in MIN x is 50, you would poke 0 ($00) into the high byte at 45352 and 50 ($32) into the low byte at 45351.

ADC-1 BSR Controller

The environmental control aspect of this project was implemented with the BSR control feature of the ADC-1 data acquisition and control system manufactured by Remote Measurements Systems, Inc. The ADC-1 is inexpensive, reliable, and easy to operate. The system can be interfaced to a host of different personal computers and gives a computer the ability to sense and act on information received from its surrounding environment. Its main features are as follows (Remote Measurements, Inc., 1983):

* 16 analog input channels coupled with a 12 bit analog-to-digital converter (A/D) permitting the ADC-1 to perform precision measurements.

* 4 digital inputs that can be used to monitor or count on/off signals.

* 6 controlled TTL outputs allow the operation of external electronic devices.

* Remote control of appliances and lamps via signals transmitted over AC wiring to BSR modules.

* Optional instrumentation amplifier providing microvolt resolution.

* RS-232 communications for easy interconnection to most computers or modems.

The ADC-1 hardware is enclosed in two boxes. The larger aluminum box contains the printed circuit board that performs the data acquisition and controlled output functions. On the top are two sets of labeled terminals to which sensor or control cables may be attached.

The smaller box plugs into an AC receptacle and houses the transmitter for modulating signals on the AC wiring which control remotely located modules. This transmitter was placed in a separate enclosure to isolate the AC voltages from the sensitive A/D circuitry. A flat 3-wire cable runs from this transmitter to a DB-25 male connector which mates with the female DB-25 on the right side of the panel of the larger box. This male connector serves as the ADC-1 end of the cable which then connects to a computer.

Each BSR module has two dials or switches used to assign a house code (from "A" to "P") and a unit code (1-16). Setting of the dip switches on the bottom of the ADC-1 determines which house code pair will be enabled at any one time. These switch settings are shown in Table 4. A different setting of the switches will activate a different pair of house codes which provide for 32 separate unit codes. A list of all the possible unit and command codes is given in Table 5.

After a unit code has been transmitted, that designated unit can be turned on or off repeatedly by command codes without repetition of the unit code. When a different unit is desired, the new unit code can then be transmitted.

Table 4. Switch settings for house codes (Remote Measurements, Inc., 1983)

House Code	Switch Number					
Pair	1	2	3	4	5	6
A,I	on	off	off	on	off	on
B,J	off	on	off	on	off	on
C,K	on	off	on	off	off	on
D,L	off	on	on	off	off	on
M,E	on	off	on	off	on	off
N,F	off	on	on	off	on	off
O,G	on	off	off	on	on	off
P,H	off	on	off	on	on	off

Table 5. Unit and control codes for BSR modules (Remote Measurement Systems, Inc., 1983)

	Unit Command / Number	Command Code		Unit Command / Number	Command Code
Commands and	CLEAR	193	Commands and	CLEAR	225
addresses for	ALL	195	addresses for	ALL	227
first house of	ON	197	second house of	ON	229
house code pair.	OFF	199	house code pair.	OFF	231
(A,B,C,D,	DIM	201	(I,J,K,L,	DIM	233
M,N,O,P)	BRIGHT	203	E,F,G,H)	BRIGHT	235
	1	204		1	236
	2	220		2	252
	3	196		3	228
	4	212		4	244
	5	194		5	226
	6	210		6	242
	7	202		7	234
	8	218		8	250
	9	206		9	238
	10	222		10	254
	11	198		11	230
	12	214		12	246
	13	192		13	224
	14	208		14	240
	15	200		15	232
	16	216		16	248

1750 RAM Expansion Unit

The 1750 RAM Expansion Unit increases the amount of memory of the C128 (either mode) by 512K, which is arranged in eight 64K banks. The memory is not directly accessible to the computer; programs or data must first be transferred to main memory in order to be used. The speed and convenience of this RAM, though, gives the appearance of direct access and this allows execution of larger programs or storage of large amounts of data which would not otherwise be possible. This memory transfer (Direct Memory Access or DMA) is accomplished through the RAM Expansion Controller (REC). The REC internal registers are shown in Figure 2.

The REC has four operating modes:

1) Transfer a block of data from main to expansion memory.

2) Transfer a block of data from expansion to main memory.

3) Exchange or swap a block of main with a block of expansion memory.

4) Verify a block of main versus a block of expansion memory.

The programmer uses several REC internal registers to set up a particular operating mode and the mode is chosen by setting the appropriate bits in the command register. Starting C64 address, expansion RAM address, expansion RAM bank number, and number of bytes to transfer are all programmable values.

The REM was used in this project to store vocabularies for the program VOICEMANAGER, a user interface which includes the environmental control function, and for the software interface that enabled the use of Commodore application programs. It also stored portions of memory critical to an application or to the software interface, whenever one software package was preempted by the other for CPU time.

ADDRESS	BITS	FUNCTION	
$00	7-0	Status Register -- Read Only	
		7 - Interrupt Pending	1 = Interrupt waiting to be serviced
		6 - End of Block	1 = Transfer complete
		5 - Fault	1 = Block verify error
		4 - Size	1 = 256 K
		3-0 - Version	
		Note: Bits 7-5 are cleared when this register is read	
$01	7-0	Command Register -- Read/Write	
		7 - Execute	1 = Transfer per current configuration
		6 - Reserved	
		5 - Load	1 = Enable AUTOLOAD option
		4 - FF00	1 = Disable FF00 decode
		3 - Reserved	
		2 - Reserved	
		1,0 - Transfer Type	00 = Transfer C64 > RAM module
			01 = Transfer C64 < RAM module
			10 = Swap C64 < > RAM module
			11 = Verify C64 -- RAM module
$02	7-0	C64 Base Address, LSB -- Read/Write Lower 8 bits of base address, C64	
$03	7-0	C64 Base Address, MSB -- Read/Write Upper 8 bits of base address, C64	
$04	7-0	Expansion RAM address, LSB -- Read/Write Lower 8 bits of base address expansion RAM	
$05	7-0	Expansion RAM address, MSB -- Read/Write Upper 8 bits of base address, expansion RAM	
$06	2-0	Expansion RAM bank -- Read/Write Expansion RAM bank pointer Bits 2 (MSB) to 0 (LSB) are significant	
$07	7-0	Transfer Length, LSB -- Read/Write Lower 8 bits of the byte counter	
$08	7-0	Transfer Length, MSB -- Read/Write Upper 8 bits of the byte counter	
$09	7-5	Interrupt Mask Register -- Read/Write	
		7 - Interrupt Enable	1 = Interrupts enabled
		6 - End of Block mask	1 = Interrupt on end of block
		5 - Verify error	1 = Interrupt on verify error
$0A	7-6	Address Control Register -- Read/Write 0,0 = Increment both addresses (default) 0,1 = Fix expansion address 1,0 = Fix C64 address 1,1 = Fix both addresses	

Figure 2. REC internal registers (Commodore Electronics Limited, 1985c)

RESULTS AND DISCUSSION

Project Results

The result of this project was a voice-controlled computer system dedicated to running compatible applications (demonstrating system use of off-the-shelf software) and environmental control (demonstrating use of a programmer-developed application). These two functions were brought together to form an integrated product which, with a minimum of keyboard maneuvering, benefits those who desire to bypass the traditional method of input, the keyboard, which can be a barrier to effective use of the computer. The computer's detailed memory map and operating system were of necessity under close study throughout the project, resulting in acquired knowledge useful for exploiting the computer's resources for further development of voice input capabilities.

Product

The chief product of this project is a software interface between a speech recognition system developed for the Commodore 64 or 128 Personal Computers and C64 application programs. This result eliminates the need for the developer to write new software for any application that might be desired for implementation in the speech input system. This product is referred to as the application interface.

This interface, composed of three machine language routines called MEMORYSWAP1, MEMORYSWAP2, and VOICEINTERPRETER, was designed to enable voice input to any C64 application that normally requests keyboard input from the user. Particular applications have to be researched, examined, or tested for compatibility with the interface. The only requirement for compatibility is that the application not use a relatively small section of page 2 memory, 80 bytes from $02A7 to $02F6 ($02A7 to $02FF is the only part of page 2 not used by Commodore and left for programmer use), where the main routine of this interface, MEMORYSWAP1, resides when loaded. MEMORYSWAP1 controls the swapping of application related memory locations with voice input related locations and is the only routine required to reside in computer memory at the same time as the application. If an application is not compatible with the interface, it might be possible to shift MEMORYSWAP1 to another unused location to accommodate the application the user needs or desires. The current demonstration system also incorporates a user interface, the program VOICEMANAGER which is written in Commodore BASIC, that allows only the selection of an application program or the environmental control function programmed into the user interface by the developer.

The main project goal was to design a system that was user friendly to those users who have limited access to the keyboard. Although the recognition software enables the user to stop recognition by pressing the "up arrow" key, the user and application interfaces have built into the software stop recognition functions which do not necessitate a keystroke. This feature allows the user to pause to speak with someone or to perform a task other than computing and still does not require access to the system through the keyboard. When it is executed, this function causes the system to choose from a vocabulary or group consisting of just one phrase, "restore speech" and to respond only to that phrase. This design prevents the system from attempting to interpret random noise or speech as a command at times when the user does not intend that speech be recognized. It also makes the system and its dedicated applications virtually independent of hand use since the user interface is autobooted upon power-up eliminating the need for a typed command to start it. The result is a system where computer power switch engagement and disengagement are the only requirements not activated by voice.

Procedures and Design Features

Two features of the voice input system were particularly strategic in accomplishing this implementation. The first feature involved the software of the application interface. MEMORYSWAP1, the chief program written for this interface, MEMORYSWAP2, and VOICEINTERPRETER, were developed to enhance the Kernal routine responsible for getting a character from the keyboard. When an application program attempts to call a Kernal routine called GETIN that processes keyboard input, the application instead is directed to the starting location of MEMORYSWAP1. MEMORYSWAP1 readies the system for speech input and then calls the program VOICE/RECOG to perform the actual recognition function. The program MEMORYSWAP1 and its associated machine language routines are the link to getting voice input to the application program. They therefore produce speech input applications from those designed for keyboard input.

Many of the operating system routines can be replaced or augmented by a programmer via a jump table of addresses for these routines located in Kernal ROM. Commodore has maintained this jump table through all of its computer design updates from the earliest PET through the 128 and will maintain it throughout any later versions of the operating system. This may allow a user's written programs to be portable from a computer to its descendant and will protect programs from changes in successive versions of the Kernal. For example, the GETIN (get a character) routine is accessed by a jump-to-subroutine (JSR) instruction to location $FFE4. This is not the actual address of the routine; what it does contain is a three-byte entry consisting of the indirect jump (JMP) operation code (4C) and the bytes 2A and 03 which represent an indirect address. This three-byte entry assembles into JMP ($032A) which in 6510 assembly language is an indirect jump to an address which is stored at $032A (low byte) and $032B (high byte), the address of GETIN in ROM. The programmer can write code to change this vector by placing different values in locations $032A and $032B, replacing 3E and F1, respectively. $F13E is the actual address of the GETIN routine. Changing the vector allows the programmer to either totally replace the Kernal routine or to add to it; this was done to the GETIN vector for this system implementation. The code that is pointed to by the $032A-$0320 vector must end with the instruction return-from-subroutine (RTS) so that program execution continues at the instruction following the original JSR that called the Kernal routine. The other main design feature was use of the Direct Memory Access (DMA) operation of the RAM Expansion Controller (REC), the processor in the 1750 which controls I/O between itself and the computer. The computer processor is temporarily halted during DMA so that the REC may utilize computer memory. The expansion memory contains indirectly accessible RAM so that programs cannot be executed directly and data cannot be directly accessed by the computer.

The use of DMA operations allows the system to use multiple vocabularies, which are essential to executing the numerous functions that have been designed for this system. The voice recognition software allows for the use of only one 32-phrase vocabulary in main memory at one time and accessing the disk drive for a substitute vocabulary would consume too much time to allow for the proper functioning of the tasks that the system must accomplish.

The other need for DMA operations is due to the fact that the voice recognition unit does not possess its own microprocessor. The VOICE/RECOG software is therefore executed by the computer and not within the voice unit. Because application programs written for the C64 will typically exhaust the computer's memory resources, the application and VOICE/RECOG cannot share these resources at the same time. In order for the application to perform with voice input it must be able to access VOICE/RECOG when attempting to call the routine to get a keyboard input character. Again, DMA transfers allow for sharing of computer resources and CPU time by the two programs. Memory and register contents critical to one program are stored in the REM while the other program employs the CPU and main memory. They are then restored to main memory when it is time for the two to trade locations.

Software Developed

VOICEMANAGER

VOICEMANAGER is a program written in Commodore BASIC that functions as the main program and user interface for this demonstration. The environmental control function was developed in BASIC as an option in VOICEMANAGER and does not require that an external program be loaded. VOICE/RECOG was the recognition program used in conjunction with the user interface. Its various subroutines necessary for implementing VOICEMANAGER functions were accessed with Commodore BASIC POKE and SYS statements. POKE allows the programmer to specify a decimal number value to place in a specified memory location and SYS is the BASIC equivalent of the assembly language instruction JSR, allowing the programmer to jump to an assembly language routine from BASIC. The BASIC program regains control when assembly language RTS is executed. The program also has a section that prepares the system for the application interface by loading its program files.

VOICEMANAGER begins by loading VOICE/RECOG, the speech recognition program, and ACSTASHORFETCH, a short assembly language program written to allow the STASHORFETCHVOCABULARY subroutine to function properly. As the name implies, STASHORFETCHVOCABULARY, a part of VOICEMANAGER, is used by the main program to both transfer a main memory located vocabulary to the 1750 and to fetch a 1750-based vocabulary back to main memory. Since vocabulary space was allocated to a RAM memory bank that resides behind the BASIC ROM space, ACSTASHORFETCH performs bank switching between the BASIC ROM and the RAM behind it that holds the vocabulary. Because ACSTASHORFETCH banks in this RAM before a vocabulary stash or fetch is attempted, then initiates the stash or fetch operation, and banks in the ROM following that procedure, it is essential to the proper performance of STASHORFETCHVOCABULARY. After an opening message is displayed for the user and the program's arrays are dimensioned, FOR/NEXT loops are used to read in to the program all of the phrase names for all vocabularies used, as well as the BSR codes used to control appliances in the environmental control option. After all main program variables are initialized and maximum and minimum recognition thresholds set, a check is made to determine whether the main vocabulary used, COMMANDER, is already on the program disk. If it is, training of COMMANDER is bypassed and the vocabulary is instead loaded from disk before the main program's main menu is displayed to the user. If COMMANDER is not on disk, then the program treats this run as the user's first run of VOICEMANAGER and informs him of the need for training this vocabulary. COMMANDER is trained and stored in expansion memory.

The user is given an option as to the number of times he would like to train the vocabulary, from one to five times. The limit of five is imposed since to do more training would not improve recognition results. Two trainings are recommended to the user before he trains as this corresponds to recognition unit instructions.

After each training of this vocabulary, the user is shown a menu asking whether he would like to continue training it. He is asked to respond to the menu with "affirm," "negative," or "stop recognition." The "stop recognition" response is a choice that is given at every user input opportunity throughout VOICEMANAGER except one. Its purpose is to provide the ability to stop the program when the user wants to suspend activity while preventing the system from responding to sound, either accidental or intentional. The recognition of "stop recognition" causes the program to jump to a subroutine called STOPRECOGNITION. This subroutine designates group 4 of the COMMANDER vocabulary, which has just one trained phrase, "restore speech," as the only group eligible for a match. Therefore, if the sensitivity of the system is appropriate, the program will only respond to that one phrase spoken by the user. This will enable him to return to that point in the program from which he left. When the phrase "restore speech" is spoken and recognized, the program will exit the STOPRECOGNITI0N subroutine and return to the menu from which "stop recognition" was selected so that the user can attempt another response.

A response of "affirm" will lead the user back through the program loop that does the training. A "negative" response will take the program out of the training loop and to the point in the program where the vocabulary is stashed in expansion memory just before the main menu is displayed.

The main menu provided in VOICEMANAGER is the principal user interface with the program. From this menu, the user is asked to select from the options "application program," "environmental controller," "program escape," "retrain commander," or "stop recognition." The "retrain commander" response will cause the current template set for the COMMANDER vocabulary to be blanked and direct the user through the training loop to create a new COMMANDER. This new vocabulary can later be saved to disk if the user chooses that option.

A response of "program escape" will direct the program through a section of code which closes VOICEMANAGER while checking the status of all vocabularies used during the current program run, whether they have been trained, retrained, or neither. The software will monitor whether any of the vocabularies used during the run have been trained for the first time (indicated by the vocabulary's absence on disk when VOICEMANAGER was loaded). If this is the case, the vocabulary will automatically be saved to disk without requiring the user to make a choice. The user will be notified of the save assuming there is disk space available which will be the case if the program disk is dedicated to VOICEMANAGER. If the program does not find enough disk space in which to place the vocabulary, it will not be saved and the user will also be notified of this result. If the program finds that a vocabulary has been retrained during this run, the user will be given the choice of saving the updated vocabulary to replace the one on disk or not saving it. The user is queried regarding this choice and must respond with "affirm," "negative," or "stop recognition." If "negative" is recognized, a message is displayed saying that the vocabulary will not be saved to disk. The previous vocabulary file on disk is deleted if the program detects a match with "affirm." The updated vocabulary is then placed on disk and the program proceeds to check the status of another vocabulary. The process of testing these vocabulary statuses continues until all have been evaluated. The Commodore user port is then closed and an exit message is printed before the program ends and the user is returned to Commodore interactive mode.

Environmental controller option

The response of "environmental controller" to the main menu will cause the program to jump to a lengthy section of code which will execute the environmental control function.

The first action is the display informing the user that he has entered the environmental control section of the program. Variables exclusive to this part of the program are initialized before the disk is checked for the ECCOMMANDS vocabulary, which contains the phrases necessary for this function. If it is already on the program disk, ECCOMMANDS training is bypassed and the vocabulary is loaded and stashed in expansion memory before the first environmental control menu is displayed to the user. If ECCOMMANDS is not on disk, then the program will instead see this as the first time the control function has been used and inform the user of the need for training this vocabulary. ECCOMMANDS is trained and stored in expansion memory with the same general procedure that the program uses for all vocabulary training.

When the environmental control user interface is reached, whether training has been finished or bypassed, the user is asked for the name of an appliance to be controlled by the software. He is given a default menu with sixteen choices. This is the only menu in VOICEMANAGER without an option given to stop recognition. The sixteen options fill all the templates for two groups (eight templates per group), and it was determined to be inefficient use of the vocabulary to use another group of eight templates for the one phrase, "stop recognition." If the user decides to use "stop recognition," that option will be available at the next menu.

Next, the chosen device name is echoed to the user, and he is asked if this is the device he intended to choose. The responses allowed are "affirm," "negative," and "stop recognition." A "negative" response will cause the program to query the user once more regarding what piece of equipment he desires to control. An "affirm" response directs the software to obtain the appliance code, stored in an array of codes, which corresponds to this device and output the code through the RS-232 port to the BSR Controller. This is accomplished by converting the decimal appliance code to a character code and using the PRINT# command to send it through channel 2. The program performs this task through a subroutine called RS232OUTPUT. Following the output of the appliance code, the user is asked whether he desires to power up or power down the device. He is asked to respond with "turn on," "shut off," or "stop recognition." When either "turn on" or "shut off" is recognized, RS232OUTPUT will send the appropriate corresponding command code to the controller.

The software then reaches a main user menu within the environmental control function which gives the user the final options available to him in this part of the program. At this menu, the user encounters five options; they are "continue control," "goto main menu," "exit the program," "retrain E/C," and "stop recognition." If "retrain E/C" is recognized, the user is led back through the program to replace the current training of this vocabulary with a new training. Recognition of "exit the program" will direct the software to save all updated vocabularies to disk that the user requests before END is executed and the user is returned to the Commodore READY prompt. The response "continue control" forces the program back to the beginning of the environmental control user interface loop where the user once again encounters "What appliance do you want to control?". This environmental control function will continue as long as the user desires.

Application Interface

The application program option is chosen by a "program application" response to the VOICEMANAGER main menu. This prompts the program to begin executing a section of code that prepares the system for the switch from running VOICEMANAGER to executing the application. When this section of the program is entered, a message is displayed to inform the user that the system is preparing to switch to the application. The MEMORYSWAP1, MEMORYSWAP2, and VOICEINTERPRETER routines that form the application interface are loaded into internal memory before variables used only for this part of the program are initialized. MEMORYSWAP2 and VOICEINTERPRETER are also stashed in expansion memory; they will soon be overrun when the application is loaded into computer RAM. The disk is checked for each of the vocabularies used for the interface, vocabularies that contain phrases corresponding to the keystrokes a user could initiate from the keyboard. If any or all of them are on the disk already, training of these vocabularies is bypassed; they are instead loaded into memory. If not, VOICEMANAGER will respond as if this is the first time these vocabularies have been needed and inform the user that they will be trained. Whether training is performed or not, the vocabularies used for the application interface are stashed in expansion memory in the same manner as other vocabularies.

The next action of VOICEMANAGER is to place default numbers in the memory locations that are set aside for the recognition threshold values. The important task of changing the vector to GETIN in locations \$032A and \$032B from \$F13E to \$02D8, the entry point to MEMORYSWAP1, is then performed. VOICEMANAGER is itself stored in expansion memory along with VOICE/RECOG before it executes a SYS to an assembly language routine which loads the application and initiates execution by branching to the application's starting address.

When the application calls GETIN to obtain user keyboard input, the changed vector will direct it instead to $02D8 in MEMORYSWAP1. This routine will load its companion routines, MEMORYSWAP2 and VOICEINTERPRETER, from expansion memory as it swaps the two for RAM locations that would otherwise be overwritten. The routine MEMORYSWAP2 is then called and continues the same process by swapping other RAM contents for VOICEINTERPRETER, VOICE/RECOG, and their associated critical memory areas.

VOICE/RECOG is then called to handle user voice input and puts the number corresponding to the recognized phrase in page 0 memory location 151 ($97). The VOICEINTERPRETER routine is called to interpret the value and put a corresponding character into the keyboard queue or no character if there is no match. GETIN is then called to perform its normal function of getting the character from the queue and placing it in the accumulator register. MEMORYSWAP1, along with MEMORYSWAP2, will repeat the swapping processes so that the application gets reloaded into main memory. The final action of the application interface, in MEMORYSWAP1, is execution of the RTS assembly instruction which causes operation to continue at the instruction following the JSR in the application program which originally attempted to call GETIN. The application will continue until the next attempt to get keyboard input results in the functions of the application interface being repeated again.

Summary

The system developed for this project met the stated objective of being usable without creating an overwhelming obstacle that would discourage its use. Despite the lowering costs of

IBM Personal Computer compatibles, the Commodore system used is still cost advantageous because of comparable price reductions for these computers and their peripheral devices. The application interface that was developed opens a door to several established software products, many of great practical use to the system user.

The main disadvantage to this Commodore based speech recognition system is the probable limit on applications that can be purchased or developed for it. This is due to the exhaustion of resources by the system application developed to provide the interface to user applications. The voice recognition unit used has no internal microprocessor and put too much demand on Commodore CPU resources to allow for the greater flexibility provided by more advanced and costly recognition systems. Like many other recognition devices, this unit is limited to recognizing isolated, trained phrases. This means the user must in advance of using applications invest a period of time to prepare the system for practical use by training and storing vocabularies. This technology has been surpassed by some voice recognition units which are continuous speech recognizers or speaker independent or are both.

This project's chief objective to access information about and gain experience with the Commodore 64 and 128 personal computers and the Covox speech recognition system was met. This experience could form the basis for further development of speech input applications for the C128 and possibly for PC compatible computers connected to PC-compatible Covox recognition instruments. This effort also resulted in the production of demonstration software which provides a usable and affordable tool for those desiring access to C64 applications without using the keyboard.

REFERENCES

Cater, John P. 1984. Electronically Hearing: Computer Speech Recognition. 1st ed. Howard W. Sams and Co., Inc., Indianapolis, Indiana. 263 pp.

Cohen, Arnon, and Daniel Graupe. 1980. Speech recognition and control system for the severely disabled. J. Biomed. Engng. 2(2):97-107.

Commodore Business Machines, Inc. 1982. Commodore 64 Programmer's Reference Guide. Commodore Business Machines, Inc., West Chester, Pennsylvania. 486 pp.

Commodore Electronics Limited. 1985a. Commodore 128 Personal Computer System Guide. Commodore Electronics Limited., Agincourt, Ontario. 404 pp.

Commodore Electronics Limited. 1985b. Commodore 1571 Disk Drive User's Guide. Commodore Electronics Limited., Agincourt, Ontario. 118 pp.

Commodore Electronics Limited. 1985c. Commodore 1700/1750 Ram Expansion Module User's Guide. Commodore Electronics Limited., Agincourt, Ontario. 24 pp.

Covox, Inc. 1985. VOICE MASTER Owner's Manual Versions 2.19 and 3.0. Covox, Inc., Eugene, Oregon. 51 pp.

Covox, Inc. 1986. VOICE MASTER User Manual Version 4.0. Covox, Inc., Eugene, Oregon. 42 pp.

Devaney, Michael J., and Farid G. Rastgar. 1983. Speech recognition for wheelchair control. Pages 109-110 in David Carlson, ed. Proceedings of the Twentieth Annual Rocky Mountain Bioengineering Symposium. Instrument Society of America, Research Triangle Park, North Carolina.

Fried-Oken, Melanie. 1985. Voice recognition device as a computer interface for motor and speech impaired people. Arch. Phys. Med. Rehabil. 66(10):678-681.

Grooms, Ronald G. 1985. Vocational Training Using a Speech-Input Computer. In Conference Proceedings of the Preparation for Life Conference. Mountain Plains Regional Resource Center and Rehabilitation Services Administration, Des Moines, Iowa.

Grooms, Ronald G. 1986. Computer speech recognition for vocational training: Strategies and observation. Pages 47-49 in Christopher Smith, ed. Discovery III: Training and Technology for the Disabled Conference Papers. Materials Development Center, School of Education and Human Services, University of Wisconsin-Stout, Menomonie, Wisconsin.

IBM Information Systems Group, 1986. Beyond simple dictation. Pages 14-15 in Robert Hawkins, ed. Innovation Digest. International Business Machines Corporation Information Systems Group, White Plains, New York.

Lam, P. C., and P. T. Lopez. 1989. Computer interaction through voice recognition: application to aid the handicapped. Pages 657-661 in D. R. Riley and T. J. Cokonis, eds. Computers in Engineering 1989: Proceedings of the 1989 ASME International Computers in Engineering Conference and Exposition. The American Society of Mechanical Engineers, New York, New York.

Leemon, Sheldon. 1984. Mapping the Commodore 64. 1st ed. Compute! Publications, Inc., Greensboro, North Carolina. 268 pp.

Remote Measurement Systems, Inc. 1983. ADC-1 Owner's Manual. Remote Measurements Systems, Inc., Seattle, Washington.

Sarnat, Andrew J. 1983. Computerized speech recognition for anesthesia recordkeeping. Medical Instrumentation 17(1):25-27.

Smith, N. Ty. 1986. Automated record keeping and data management in the operating room. Pages 79-80 in Charles J. Robinson and George V. Kondraske, eds. Proceedings of a Special Symposium on Critical Emerging Issues in Biomedical Engineering. The Institute of Electrical and Electronics Engineers, Inc., New York, New York.

Vaissiere, J. 1985. Speech recognition: a tutorial. Pages 191-242 in Frank Fallside and William A. Woods, eds. Computer speech processing. Prentice-Hall International (UK) LTD, London, England.

ACKNOWLEDGEMENTS

Initially, I would like to express my appreciation to Dr. Patrick Patterson for serving as my major professor. He has allowed me to do most of the work on my own while we have been separated by a distance of about 1800 miles. He also deserves credit for originally contributing the basic ideas that led to this project. Ronald Grooms, who served on my committee, helped refine this assignment, especially in its early stages. He was helpful, available, and encouraging. I enjoyed our many phone conversations and discussions. I also thank Dr. Richard Horton and Dr. Curran Swift for serving on my committee and for teaching classes that were the foundation of my training in programming microprocessors and microcomputers.

I also express appreciation to Commodore Business Machines, Inc., Covox, Inc. (especially Brad Stewart), and Remote Measurements Systems, Inc. for graciously giving technical help.

A note of appreciation goes to the Biomedical Engineering Department for the financial support that I received while at Iowa State. I also thank the department for allowing me to complete the work (and giving me extra time to do that) while involved in a full-time job in California 1800 miles away from Ames, Iowa.

Dexter Iishi encouraged me and helped me when I was in Ames. I thank him as well as the "birthday club" - those four students (and one wife) who came into the Biomedical Engineering program the same semester as I. Thanks to Jim and Joan Graf, Tou-Chung (and later Shu-Ching) Hou, Raveesh Talanki, and Je-phil Ryoo for their friendship.

Lastly, I thank my wife, Corrie, who has struggled with me to bring the work to completion. I also give thanks to God for continually dealing patiently and lovingly with my weaknesses, especially the areas of procrastination and choosing to do those things which are priority as a way of life.

APPENDIX A

MEMORYSWAP1

```
02A7 A0 EE     LDY #$EE      * SETS OPERATING MODE TO SWAP FOR REU
02A9 4C AE 02 JMP $02AE      * JUMP UNCONDITIONAL SINCE MODE IS SET
02AC A0 EC     LDY #$EC      * SETS OPERATION MODE TO STASH
02AE A2 08     LDX #$08      * "#OF REC REGISTER" PUT IN X REG
02B0 BD CF 02 LDA $02CF,X * LOAD PARAMETER
02B3 9D 02 DF STA $DF02,X * STORE IN REC REGISTER
02B6 CA        DEX           * DECREMENT X REG VALUE
02B7 10 F7     BPL $02B0     * LOOP UNTIL "#OF REC REGISTER"=0
02B9 8C 01 DF STY $DF01      * STORE OPERATING MODE IN REC REGISTER 1
02BC 78        SEI           * DISABLE INTERRUPTS
02BD A5 01     LDA $01       * LOAD CONTENTS OF LOCATION $01
02BF 48        PHA           * PUSH THE $01 CONTENTS ONTO STACK
02C0 29 FC     AND #$FC      * INSURE THAT BITS 0 AND 1 = 0
02C2 85 01     STA $01       * INSURE THAT RAM IS BANKED IN
02C4 AD 00 FF LDA $FF00      * LOAD WHATEVER VALUE THAT $FF00 CONTAINS
02C7 8D 00 FF STA $FF00      * INITIATE EXECUTION OF THE STASH OR SWAP
02CA 68        PLA           * PULL ACCUMULATOR FROM STACK
02CB 85 01     STA $01       * RESET LOCATION $01 TO PREVIOUS VALUE
02CD 58        CLI           * CLEAR INTERRUPT DISABLE
02CE 60        RTS           * RETURN FROM THIS SUBROUTINE
02CF 00                      * THESE 9 LOCATIONS FOR REC PARAMETER STORAGE
02D0 00
02D1 00
02D2 00
02D3 00
02D4 00
02D5 00
02D6 00
02D7 00
02D8 AD F6 02 LDA $02F6      * LOAD FLAG WHICH INDICATES IF CHAR OBTAINED
02DB F0 06     BEQ $02E3     * FLAG = 0 SO BRANCH TO $02E3
02DD 20 A7 02 JSR $02A7      * CALL SUBROUTINE TO SWAP MEMORY
02E0 20 F0 C1 JSR $C1F0      * CALL "MEMORYSWAP2" ROUTINE
02E3 20 3E F1 JSR $CB7B      * CALL "VOICEINTERPRETER" ROUTINE
02E6 8D F6 02 STA $02F6      * STORE VALUE OBTAINED IN LOCATION $02F6
02E9 D0 01     BNE $02EC     * BRANCH IF A CHARACTER OBTAINED
02EB 60        RTS           * RETURN FROM THIS SUBROUTINE
02EC 20 F0 C1 JSR $C1F0      * CALL "MEMORYSWAP2" ROUTINE
02EF 20 A7 02 JSR $02A7      * CALL SUBROUTINE TO SWAP MEMORY
02F2 AD F6 02 LDA $02F6      * LOAD CHARACTER OBTAINED IN ACCUMULATOR
02F5 60        RTS           * RETURN FROM THIS SUBROUTINE
02F6 01                      * FLAG INDICATING IF CHAR OBTAINED INITIALLY=1
```

MEMORYSWAP2

```
0C1F0 A9 00    LDA #$00
0C1F2 8D D0 02 STA $02D0
0C1F5 8D D2 02 STA $02D2
0C1F8 8D D5 02 STA $02D5
0C1FB A9 02    LDA #$02
0C1FD 8D CF 02 STA $02CF
0C200 8D D1 02 STA $02D1
0C203 A9 FD    LDA #$FD
0C205 8D D4 02 STA $02D4
0C208 20 A7 02 JSR $02A7  * SET UP REGISTERS & SWAP $02 - $FE
0C20B A9 04    LDA #$04
0C20D 8D CF 02 STA $02CF
0C210 8D D1 02 STA $02D1
0C213 A9 DD    LDA #$DD
0C215 8D D0 02 STA $02D0
0C218 A9 30    LDA #$30
0C21A 8D D2 02 STA $02D2
0C21D A9 0C    LDA #$0C
0C21F 8D D4 02 STA $02D4
0C222 20 A7 02 JSR $02A7  * SET UP REGISTER & SWAP $DD04 - $DD0F
0C225 A9 00    LDA #$00
0C227 8D CF 02 STA $02CF
0C22A 8D D1 02 STA $02D1
0C22D 8D D4 02 STA $02D4
0C230 A9 E0    LDA #$E0
0C232 8D D0 02 STA $02D0
0C235 A9 31    LDA #$31
0C237 8D D2 02 STA $02D2
0C23A A9 10    LDA #$10
0C23C 8D D5 02 STA $02D5
0C23F 20 A7 02 JSR $02A7  * SET UP REGISTERS & SWAP $E000 - $EFFF
0C242 A9 B1    LDA #$B1
0C244 8D D0 02 STA $02D0
0C247 A9 11    LDA #$11
0C249 8D D2 02 STA $02D2
0C24C A9 2C    LDA #$2C
0C24E 8D D4 02 STA $02D4
0C251 A9 0C    LDA #$0C
0C253 8D D5 02 STA $02D5
0C256 20 A7 02 JSR $02A7  * SET UP REGISTERS & SWAP $B100 - $BD2B
0C259 A9 00    LDA #$00
0C25B 8D D5 02 STA $02D5
0C25E A9 DC    LDA #$DC
0C260 8D D0 02 STA $02D0
0C263 A9 30    LDA #$30
0C265 8D D2 02 STA $02D2
0C268 A9 03    LDA #$03
0C26A 8D D4 02 STA $02D4
0C26D 20 A7 02 JSR $02A7  * SET UP REGISTERS & SWAP $DC00 - $DC02
0C270 A9 86    LDA #$86
0C272 8D CF 02 STA $02CF
0C275 8D D1 02 STA $02D1
0C278 A9 02    LDA #$02
```

```
0C27A 8D D0 02 STA  $02D0
0C27D 8D D2 02 STA  $02D2
0C280 A9 04    LDA  #$04
0C282 8D D4 02 STA  $02D4
0C285 20 A7 02 JSR  $02A7  * SET UP REGISTERS & SWAP $0286 - $0289
0C288 A9 3C    LDA  #$3C
0C28A 8D CF 02 STA  $02CF
0C28D 8D D1 02 STA  $02D1
0C290 A9 03    LDA  #$03
0C292 8D D0 02 STA  $02D0
0C295 8D D2 02 STA  $02D2
0C298 A9 02    LDA  $#02
0C29A 8D D4 02 STA  $02D4
0C29D 20 A7 02 JSR  $02A7  * SET UP REGISTERS & SWAP $033C - $033D
0C2A0 A9 03    LDA  #$0E
0C2A2 8D CF 02 STA  $02CF
0C2A5 8D D1 02 STA  $02D1
0C2A8 A9 01    LDA  #$01
0C2AA 8D D4 02 STA  $02D4
0C2AD 20 A7 02 JSR  $02A7  * SET UP REGISTERS & SWAP $030E
0C2B0 A9 27    LDA  #$27
0C2B2 8D CF 02 STA  $02CF
0C2B5 8D D1 02 STA  $02D1
0C2B8 A9 04    LDA  #$04
0C2BA 8D D0 02 STA  $02D0
0C2BD 8D D2 02 STA  $02D2
0C2C0 20 A7 02 JSR  $02A7  * SET UP REGISTERS & SWAP $0427
0C2C3 A9 D8    LDA  #$D8
0C2C5 8D D0 02 STA  $02D0
0C2C8 A9 2F    LDA  #$2F
0C2CA 8D D2 02 STA  $02D2
0C2CD 20 A7 02 JSR  $02A7  * SET UP REGISTERS & SWAP $D827
0C2D0 A9 01    LDA  #$01
0C2D2 8D CF 02 STA  $02CF
0C2D5 8D D1 02 STA  $02D1
0C2D8 A9 00    LDA  #$00
0C2DA 8D D0 02 STA  $02D0
0C2DD 8D D2 02 STA  $02D2
0C2E0 20 A7 02 JSR  $02A7  * SET UP REGISTERS & SWAP $0001
0C2E3 A9 F0    LDA  #$F0
0C2E5 8D CF 02 STA  $02CF
0C2E8 8D D1 02 STA  $02D1
0C2EB A9 C1    LDA  #$C1
0C2ED 8D D0 02 STA  $02D0
0C2F0 A9 21    LDA  #$21
0C2F2 8D D2 02 STA  $02D2
0C2F5 A9 66    LDA  #$66
0C2F7 8D D4 02 STA  $02D4
0C2FA A9 0A    LDA  #$0A
0C2FC 8D D5 02 STA  $02D5  * SET UP REGISTERS TO SWAP $C1F0-$CC55 NEXT TIME
0C2FF 60       RTS         * RETURN TO "MEMORYSWAP1"
```

VOICEINTERPRETER

```
0CB7B 20 14 C3 JSR  $C314   * CALL SUBROUTINE "RECOG" IN "VOICE/RECOG"
0CB7E A5 97    LDA  $97     * LOAD RECOGNIZED PHRASE #
0CB80 C9 1A    CMP  #$1A    * COMPARE WITH ASCII 26
0CB82 10 03    BPL  #CB87   * BRANCH IF > 26
0CB84 69 41    ADC  #$41    * CREATE ASCII CHAR FOR PHRASE RECOGNIZED
0CB86 60       RTS          * RETURN TO "MEMORYSWAP1"
0CB87 C9 1A    CMP  #$1A    * COMPARE WITH 26
0CB89 D0 03    BNE  $CB8E   * BRANCH IF NOT EQUAL TO 26
0CB8B A9 93    LDA  #$93    * CREATE ASCII CHAR FOR PHRASE 26
0CB8D 60       RTS          * RETURN TO "MEMORYSWAP1"
0CB8E C9 1B    CMP  #$1B    * COMPARE WITH 27
0CB90 D0 03    BNE  $CB95   * BRANCH IF NOT EQUAL TO 27
0CB92 A9 8E    LDA  #$8E    * CREATE ASCII CHAR FOR PHRASE 27
0CB94 60       RTS          * RETURN TO "MEMORYSWAP1"
0CB95 C9 1C    CMP  #$1C    * COMPARE WITH 28
0CB97 D0 03    BNE  $CB9C   * BRANCH IF NOT EQUAL TO 28
0CB99 A9 0E    LDA  #$0E    * CREATE ASCII CHAR FOR PHRASE 28
0CB9B 60       RTS          * RETURN TO "MEMORYSWAP1"
0CB9C C9 1D    CMP  #$1D    * COMPARE WITH 29
0CB9E D0 03    BNE  $CBA3   * BRANCH IF NOT EQUAL TO 29
0CBA0 A9 20    LDA  #$20    * CREATE ASCII CHAR FOR PHRASE 29
0CBA2 60       RTS          * RETURN TO "MEMORYSWAP1"
0CBA3 4C 7B CB JMP  $CB7B   * JUMP TO $CB7B SINCE NO RECOGNITION
```

APPENDIX B

The following programs comprise the user interface (main demonstration program). They are the BASIC program VOICEMANAGER and the assembly language routine ACSTASHORFETCH. ACSTASHORFETCH is included here because it is called and used primarily by VOICEMANAGER.

VOICEMANAGER

```
1000 REM
1001 REM
1002 REM:     MAIN PROGRAM VOICEMANAGER
1003 REM
1004 REM      PURPOSE:  TO LOAD OR TRAIN THE MAIN VOCABULARY, 'COMMANDER',
1005 REM      -------   AND GUIDE THE USER TO FIVE MAIN PROGRAM OPTIONS.
1006 REM
1007 REM
1008 REM      DATA DICTIONARY
1009 REM
1010 REM AC  = APPLIANCE CODE SENT TO BSR UNIT TO IDENTIFY DEVICE
1011 REM AL$ = ARRAY HOLDING PHRASE NAMES FOR 'ALPHABET' VOCABULARY
1012 REM AP  = DISCRETE: APPLICATION PROGRAM FUNCTION USED THIS RUN (1)
1013 REM       OR NOT (0)
1014 REM AS$ = 'ALPHABET' STATUS ON DISK VARIABLE ("00"='ALPHABET'
1015 REM       ON DISK AND "62" = 'ALPHABET' NOT ON DISK)
1016 REM AT  = COUNTER OF # TIMES 'ALPHABET' VOCABULARY HAS TRAINED
1017 REM AU  = DISCRETE: 'ALPHABET' VOCABULARY UPDATED (1) OR NOT (0)
1018 REM A1  = DISCRETE: 'ALPHABET' TRAINED 1ST TIME (1) OR NOT (0)
1019 REM BA  = BASE ADDRESS, RAM EXPANSION CONTROLLER REGISTERS IN I/O
1020 REM CR$ = ARRAY HOLDING PHRASE NAMES FOR 'COMMANDER' VOCABULARY
1021 REM CS$ = 'COMMANDER' STATUS ON DISK VARIABLE ("00" = 'COMMANDER'
1022 REM       ON DISK AND "62" = 'COMMANDER' NOT ON DISK)
1023 REM CT    COUNTER OF # TIMES 'COMMANDER' VOCABULARY HAS TRAINED
1024 REM CU  = DISCRETE: 'COMMANDER' VOCABULARY UPDATED (1) OR NOT (0)
1025 REM C1  = DISCRETE: 'COMMANDER' TRAINED 1ST TIME (1) OR NOT (0)
1026 REM DE  = INDEX FOR A DELAY FOR/NEXT LOOP
1027 REM DM  = DECISION MAKER FOR A BASIC 'ON' STATEMENT - IT IS THE
1028 REM       RESULT OF A FORMULA CONVERTING RECOGNIZED PHRASE #'S
1029 REM       INTO ACCEPTABLE OPERANDS FOR THE 'ON' - [1,2,3,…]
1030 REM D%  = ARRAY HOLDING THE APPLIANCE CODES ASSIGNED TO DEVICES
1031 REM EC  = DISCRETE: ENVIRONMENTAL CONTROLLER FUNCTION USED THIS
1032 REM       RUN (1) OR NOT (0)
1033 REM EC$ = ARRAY HOLDING PHRASE NAMES FOR 'ECCOMMANDS' VOCABULARY
1035 REM ES$ = 'ECCOMMANDS' STATUS ON DISK VARIABLE ("00"='ECCOMMANDS'
1036 REM       ON DISK AND "62" = 'ECCOMMANDS' NOT ON DISK)
1037 REM ET  = COUNTER OF # TIMES 'ECCOMMANDS' VOCABULARY HAS TRAINED
1039 REM EU  = DISCRETE: 'ECCOMMANDS' VOCABULARY UPDATED (1) OR NOT (0)
1040 REM E1  = DISCRETE: 'ECCOMMANDS' TRAINED 1ST TIME (1) OR NOT (0)
1041 REM FR  = FAILED RECOGNITION ATTEMPTS SUM
1042 REM G2  = DISCRETE: 'ECCOMMANDS' 1ST 2 GROUPS SET FOR RECOGNITION (1)
1043 REM       OR ANOTHER ONE VOCABULARY GROUP SET FOR RECOGNITION (0)
```

```
1044 REM MN  = APPLIANCE MODULE NUMBER FOR ENVIRONMENTAL CONTROLLER
1045 REM NP$ = ARRAY HOLDING PHRASE NAMES FOR 'NUMBERSPLUS' VOCABULARY
1046 REM NS$ = 'NUMBERSPLUS' STATUS ON DISK VARIABLE ("00"='NUMBERSPLUS'
1047 REM       ON DISK AND "62" = 'NUMBERSPLUS' NOT ON DISK)
1048 REM NT  = COUNTER OF # TIMES 'NUMBERSPLUS' VOCABULARY HAS TRAINED
1049 REM NU  = DISCRETE: 'NUMBERSPLUS' VOCABULARY UPDATED (1) OR NOT (0)
1050 REM N1  = DISCRETE: 'NUMBERSPLUS' TRAINED 1ST TIME (1) OR NOT (0)
1051 REM PH  = INDEX FOR FOR/NEXT LOOP, MEANS 'PHRASE'
1052 REM PS$ = 'PUNCTUATION' STATUS ON DISK VARIABLE ("00"='PUNCTUATION'
1053 REM       ON DISK AND "62" = 'PUNCTUATION' VOCABULARY NOT ON DISK)
1054 REM PT  = COUNTER OF # TIMES 'PUNCTUATION' VOCABULARY HAS TRAINED
1055 REM PU  = DISCRETE: 'PUNCTUATION' VOCABULARY UPDATED (1) OR NOT (0)
1056 REM PU$ = ARRAY HOLDING PHRASE NAMES FOR 'PUNCTUATION' VOCABULARY
1057 REM P1  = DISCRETE: 'PUNCTUATION' TRAINED 1ST TIME (1) OR NOT (0)
1058 REM
1059 REM
1060 REM
1061 REM      THE 'COMMANDER' VOCABULARY: THESE PHRASE NAMES ARE STORED
1062 REM      IN THE ARRAY CR$
1063 REM
1064 REM    GROUP 1
1065 REM      PHRASE    0 = 'APPLICATION PROGRAM'
1066 REM      PHRASE    1 = 'ENVIRONMENTAL CONTROLLER'
1067 REM      PHRASE    2 = 'PROGRAM ESCAPE'
1068 REM      PHRASE    3 = 'RETRAIN COMMANDER'
1069 REM      PHRASE    4 = 'STOP RECOGNITION'
1070 REM
1071 REM    GROUP 2
1072 REM      PHRASE    8 = 'AFFIRM'
1073 REM      PHRASE    9 = 'NEGATIVE'
1074 REM      PHRASE   10 = 'STOP RECOGNITION'
1075 REM
1076 REM    GROUP 3
1077 REM      PHRASE   16 = 'BEGIN TRAINING'
1078 REM      PHRASE   17 = 'STOP RECOGNITION'
1079 REM
1080 REM    GROUP 4
1081 REM      PHRASE   24 = 'RESTORE SPEECH'
1082 REM
1083 REM      THE 'ECCOMMANDS' VOCABULARY: THESE PHRASE NAMES ARE STORED
1084 REM      IN THE ARRAY EC$
1085 REM
1086 REM    GROUP 1
1087 REM      PHRASE    0 = 'LAMP-1'
1088 REM      PHRASE    1 = 'RADIO-1'
1089 REM      PHRASE    2 = 'TELEVISION'
1090 REM      PHRASE    3 = 'STEREO'
1091 REM      PHRASE    4 = 'MICROWAVE'
1092 REM      PHRASE    5 = 'FAN'
1093 REM      PHRASE    6 = 'LAMP-2'
1094 REM      PHRASE    7 = 'RADIO-2'
1095 REM
1096 REM    GROUP 2
1097 REM      PHRASE    8 = 'LAMP-3'
1098 REM      PHRASE    9 = 'PORCHLIGHT'
```

```
1099 REM      PHRASE   10 = 'LAMP-4'
1100 REM      PHRASE   11 = 'PATIOLIGHT'
1101 REM      PHRASE   12 = 'TOASTER'
1102 REM      PHRASE   13 = 'BLENDER'
1103 REM      PHRASE   14 = 'GARAGELIGHT'
1104 REM      PHRASE   15 = 'BATTERYCHARGER'
1105 REM
1106 REM    GROUP 3
1107 REM      PHRASE   16 = 'TURN ON'
1108 REM      PHRASE   17 = 'SHUT OFF'
1109 REM      PHRASE   18 = 'STOP RECOGNITION'
1110 REM
1111 REM    GROUP 4
1112 REM      PHRASE   24 = 'CONTINUE CONTROL'
1113 REM      PHRASE   25 = 'GOTO MAIN MENU'
1114 REM      PHRASE   26 = 'EXIT THE PROGRAM'
1115 REM      PHRASE   27 = 'RETRAIN E/C'
1116 REM      PHRASE   28 = 'STOP RECOGNITION'
1117 REM
1118 REM      THE 'ALPHABET' VOCABULARY: THESE PHRASE NAMES ARE STORED
1119 REM      IN THE ARRAY AL$
1120 REM
1121 REM    GROUP 1
1122 REM      PHRASE    0 = 'ALPHA A'
1123 REM      PHRASE    1 = 'BRAVO B'
1124 REM      PHRASE    2 = 'CHARLIE C'
1125 REM      PHRASE    3 = 'DELTA D'
1126 REM      PHRASE    4 = 'ECHO E'
1127 REM      PHRASE    5 = 'FOXTROT F'
1128 REM      PHRASE    6 = 'GURU G'
1129 REM      PHRASE    7 = 'HOTEL H'
1130 REM      PHRASE    8 = 'INDIA I'
1131 REM
1132 REM    GROUP 2
1133 REM      PHRASE    9 = 'JULIET J'
1134 REM      PHRASE   10 = 'KILO K'
1135 REM      PHRASE   11 = 'LIMA L'
1136 REM      PHRASE   12 = 'MICHAEL M'
1137 REM      PHRASE   13 = 'NOVEMBER N'
1138 REM      PHRASE   14 = 'OSCAR O'
1139 REM      PHRASE   15 = 'PAPA P'
1140 REM      PHRASE   16 = 'QUEBEC Q'
1141 REM
1142 REM    GROUP 3
1143 REM      PHRASE   17 = 'ROMEO R'
1144 REM      PHRASE   18 = 'SIERRA S'
1145 REM      PHRASE   19 = 'TANGO T'
1146 REM      PHRASE   20 = 'UNIFORM U'
1147 REM      PHRASE   21 = 'VICTOR V'
1148 REM      PHRASE   22 = 'WHISKEY W'
1149 REM      PHRASE   23 = 'X-RAY X'
1150 REM
1151 REM    GROUP 4
1152 REM      PHRASE   24 = 'YANKEE Y'
1153 REM      PHRASE   25 = 'ZULU Z'
```

```
1154 REM     PHRASE  26 = 'CLEAR SCREEN'
1155 REM     PHRASE  27 = 'CAPITALS'
1156 REM     PHRASE  28 = 'LOWERCASE'
1157 REM     PHRASE  29 = 'SPACE BAR'
1158 REM     PHRASE  30 = 'SWITCH PUNCTUATION'
1159 REM     PHRASE  31 = 'SWITCH NUMBERSPLUS'
1160 REM
1161 REM    THE 'PUNCTUATION' VOCABULARY: THESE PHRASE NAMES ARE STORED
1162 REM    IN THE ARRAY PU$
1163 REM
1164 REM    GROUP 1
1165 REM     PHRASE   0 = 'EXCLAMATION POINT'
1166 REM     PHRASE   1 = 'QUOTATION MARKS'
1167 REM     PHRASE   2 = 'NUMBER SIGN'
1168 REM     PHRASE   3 = 'DOLLAR SIGN'
1169 REM     PHRASE   4 = 'PERCENTAGE'
1170 REM     PHRASE   5 = 'AMPERSAND'
1171 REM     PHRASE   6 - 'APOSTROPHE'
1172 REM     PHRASE   7 = 'OPEN PAREN'
1173 REM
1174 REM    GROUP 2
1175 REM     PHRASE   8 = 'CLOSE PAREN'
1176 REM     PHRASE   9 = 'ASTERISK'
1177 REM     PHRASE  10 = 'POSITIVE SIGN'
1178 REM     PHRASE  11 = 'COMMA'
1179 REM     PHRASE  12 = 'MINUS SIGN'
1180 REM     PHRASE  13 = 'PERIOD'
1181 REM     PHRASE  14 = 'SLASH MARK'
1182 REM     PHRASE  15 = 'COLON'
1183 REM
1184 REM    GROUP 3
1185 REM     PHRASE  16 = 'SEMICOLON'
1186 REM     PHRASE  17 = 'LESSER THAN'
1187 REM     PHRASE  18 = 'EQUAL TO'
1188 REM     PHRASE  19 = 'GREATER THAN'
1189 REM     PHRASE  20 = 'QUESTION MARK'
1190 REM     PHRASE  21 = 'CIRCLE A'
1191 REM     PHRASE  22 = 'LEFT BRACKET'
1192 REM     PHRASE  23 = 'BRITISH POUND'
1193 REM
1194 REM    GROUP 4
1195 REM     PHRASE  24 = 'RIGHT BRACKET'
1196 REM     PHRASE  25 = 'UP ARROW'
1197 REM     PHRASE  26 = 'LEFT ARROW'
1198 REM     PHRASE  27 = 'CAPITALS'
1199 REM     PHRASE  28 = 'LOWERCASE'
1200 REM     PHRASE  29 = 'SPACE BAR'
1201 REM     PHRASE  30 = 'SWITCH ALPHABET'
1202 REM     PHRASE  31 = 'SWITCH NUMBERSPLUS'
1203 REM
1204 REM    THE 'NUMBERSPLUS' VOCABULARY: THESE PHRASE NAMES ARE STORED
1205 REM    IN THE ARRAY NP$
1206 REM
1207 REM    GROUP 1
1208 REM     PHRASE   0 = 'ZERO Z'
```

```
1209 REM       PHRASE    1 = 'ONE O'
1210 REM       PHRASE    2 = 'TWO T'
1211 REM       PHRASE    3 = 'THREE TH'
1212 REM       PHRASE    4 = 'FOUR R'
1213 REM       PHRASE    5 = 'FIVE F'
1214 REM       PHRASE    6 = 'SIX X'
1215 REM       PHRASE    7 = 'SEVEN V'
1216 REM
1217 REM     GROUP 2
1218 REM       PHRASE    8 = 'EIGHT E'
1219 REM       PHRASE    9 = 'NINE N'
1220 REM       PHRASE   10 = 'ONE O FUNCTION'
1221 REM       PHRASE   11 = 'THREE TH FUNCTION'
1222 REM       PHRASE   12 = 'FIVE F FUNCTION'
1223 REM       PHRASE   13 = 'SEVEN V FUNCTION'
1224 REM       PHRASE   14 = 'TWO T FUNCTION'
1225 REM       PHRASE   15 = 'FOUR R FUNCTION'
1226 REM
1227 REM     GROUP 3
1228 REM       PHRASE   16 = 'SIX X FUNCTION'
1229 REM       PHRASE   17 = 'EIGHT E FUNCTION'
1230 REM       PHRASE   18 = 'CURSOR UP'
1231 REM       PHRASE   19 = 'INSERT'
1232 REM       PHRASE   20 = 'CURSOR LEFT'
1233 REM       PHRASE   21 = 'RETURN'
1234 REM       PHRASE   22 = 'CURSOR DOWN'
1235 REM       PHRASE   23 = 'GO HOME'
1236 REM
1237 REM     GROUP 4
1238 REM       PHRASE   24 = 'DELETE A STROKE'
1239 REM       PHRASE   25 = 'CURSOR RIGHT'
1240 REM       PHRASE   26 = 'QUIT APPLICATION'
1241 REM       PHRASE   27 = 'CAPITALS'
1242 REM       PHRASE   28 = 'LOWERCASE'
1243 REM       PHRASE   29 = 'SPACE BAR'
1244 REM       PHRASE   30 = 'SWITCH ALPHABET'
1245 REM       PHRASE   31 = 'SWITCH PUNCTUATION'
1246 REM
1247 REM     OTHER NOTES
1248 REM
1249 REM     *  THREE-DIGIT LABELS ARE DISPERSED THROUGHOUT THE PROGRAM
1250 REM        IN ORDER TO AID TRACING PROGRAM FLOW. GOTO STATEMENTS ARE
1251 REM        USUALLY FOLLOWED BY A COMMENT WITH ONE OF THESE LABELS.
1252 REM        FINDING THE LABEL SHOWS WHERE THE GOTO GOES TO. ALL
1253 REM        SUBROUTINE CALLS (GOSUB'S) ALSO HAVE THESE COMMENTED
1254 REM        LABELS WHICH MATCH WITH A LABEL IN THE HEADER OF THE
1255 REM        SUBROUTINE CALLED.
1256 REM
1257 REM     *  THE PROGRAM OPENS WITH FOUR STATEMENTS THAT BEGIN WITH
1258 REM        EITHER "IF A = 0" OR "IF B = 0". THAT IS BECAUSE OF A
1259 REM        QUIRK IN COMMODORE BASIC, WHICH EXECUTES AT THE BEGINNING
1260 REM        OF THE PROGRAM FOLLOWING A "LOAD" THAT IS EXECUTED WITHIN
1261 REM        A BASIC PROGRAM. THE "IF-THEN" STATEMENTS AND
1262 REM        CORRESPONDING ASSIGNMENT OF 1 TO B OR C PREVENT THE
1263 REM        PROGRAM FROM CONTINUALLY EXECUTING THE BEGINNING "LOAD"
```

```
1264 REM        STATEMENTS OVER AND OVER.
1265 REM
1510 REM    BEGIN VOICEMANAGER
1512 REM    ------------------
1515 REM
1520 IF A = 0 THEN OPEN 2,2,0, CHR$(136) + CHR$(0)
1525 IF A = 0 THEN PRINT "                    - LOADING VOICE/RECOG -"
1530 IF A = 0 THEN A = 1 : LOAD "VOICE/RECOG",8,1
1535 IF B = 0 THEN PRINT "                    - LOADING ACSTASHORFETCH -"
1540 IF B = 0 THEN B = 1 : LOAD "ACSTASHORFETCH",8,1
1545 IF C = 1 THEN GOTO 3674
1547 IF D = 1 THEN GOTO 3678
1550 REM
1560 REM: PRINT OUT OPENING MESSAGE TO USER AND DIMENSION ARRAYS
1565 PRINT "      * * * * * * * * * * * * * * * * * * * *"
1570 PRINT
1575 PRINT "      *    VOICE INPUT APPLICATION         *"
1580 PRINT "      *              WITH                  *"
1585 PRINT "      *     ENVIRONMENTAL CONTROL          *"
1590 PRINT "      * * * * * * * * * * * * * * * * * * * *"
1595 PRINT "               DEVELOPED BY"
1600 PRINT "                GARY AYERS"
1605 PRINT "      * * * * * * * * * * * * * * * * * * * *"
1610 DIM  CR$(24), D%(15)            :REM:  DIMENSION
1612 DIM  EC$(28), VO$(31)           :REM:  THE
1613 DIM  AL$(31), PU$(31), NP$(31) :REM:  ARRAYS
1615 REM -------------------------------------------------------------
1620 REM:  READ IN THE PHRASE NAMES FOR THE 'COMMANDER' VOCABULARY
1625 REM
1630 FOR PH = 0 TO 4                     :REM:  LOOP TO READ NAMES
1635 READ CR%(PH)                        :REM:  READ NAMES FOR #'S 0-4
1640 IF PH < 3 THEN READ CR$(PH + 8)     :REM:  READ NAMES FOR #'S 8-10
1645 IF PH < 2 THEN READ CR$(PH + 16)    :REM:  READ NAMES FOR #'S 16-17
1650 IF PH > 1 THEN READ CR$(PH + 24)    :REM:  READ NAMES FOR #24
1655 NEXT PH                             :REM:  END FOR/NEXT LOOP
1660 REM -------------------------------------------------------------
1665 REM:  READ ENVIRONMENTAL CONTROLLER CODES & 'ECCOMMANDS' NAMES
1670 REM
1675 FOR PH = 0 TO 7                     :REM:  LOOP TO READ NAMES
1680 READ EC$(PH)                        :REM:  READ NAMES FOR #'S 0-7
1685 READ D%(PH)                         :REM:  READ CODES FOR #'S 0-7
1690 READ EC$(PH + 8)                    :REM:  READ NAMES FOR #'S 8-15
1700 READ D%(PH + 8)                     :REM:  READ CODES FOR #'S 8-15
1702 IF PH < 3 THEN READ EC$(PH + 16)    :REM:  READ NAMES FOR #'S 16-18
1705 IF PH < 5 THEN READ EC$(PH + 24)    :REM:  READ NAMES FOR #'S 24-28
1710 REM -------------------------------------------------------------
1720 REM:  INITIALIZE VARIABLES & THRESHOLDS THEN DELAY TO VIEW A DISPLAY
1725 REM
1727 AP = 0            :REM:  INDICATES APPL PROGRAM NOT USED THIS RUN
1728 EC = 0            :REM:  INDICATES ENV CONTROLLER NOT USED THIS RUN
1730 BA = 57088        :REM:  $DF00=BASE ADDR FOR REC REG'S INTO I/O SPACE
1733 AS$ = " "         :REM:  NULL THE AS$ STRING FOR LATER USE
1734 CS$ = " "         :REM:  NULL THE CS$ STRING FOR LATER USE
1735 ES$ = " "         :REM:  NULL THE ES$ STRING FOR LATER USE
1736 NS$ = " "         :REM:  NULL THE NS$ STRING FOR LATER USE
```

```
1737 PS$ = " "        :REM:  NULL THE PS$ STRING FOR LATER USE
1738 AU = 0           :REM:  INDICATES NO UPDATE TO 'ALPHABET' YET
1739 CU = 0           :REM:  INDICATES NO UPDATE TO 'COMMANDER' YET
1740 EU = 0           :REM:  INDICATES NO UPDATE TO 'ECCOMMANDS' YET
1741 NU = 0           :REM:  INDICATES NO UPDATE TO 'NUMBERSPLUS' YET
1742 PU = 0           :REM:  INDICATES NO UPDATE TO 'PUNCTUATION' YET
1743 A1 = 0           :REM:  = 1 ONLY IF 'ALPHABET' NOT ON DISK AT START
1744 C1 = 0           :REM:  = 1 ONLY IF 'COMMANDER' NOT ON DISK AT START
1745 E1 = 0           :REM:  = 1 ONLY IF 'ECCOMMANDS' NOT ON DISK
1746 N1 = 0           :REM:  = 1 ONLY IF 'NUMBERSPLUS' NOT ON DISK
1747 P1 = 0           :REM:  = 1 ONLY IF 'PUNCTUATION' NOT ON DISK
1750 G2 = 0           :REM:  = ZERO THE '2 GROUPS' VARIABLE
1752 POKE 45351, 35   :REM:  SET MIN RECOGNITION THRESHOLD LOW BYTE
1754 POKE 45352, 0    :REM:  SET MIN RECOGNITION THRESHOLD HIGH BYTE
1756 POKE 45353, 204 :REM:  SET MAX RECOGNITION THRESHOLD LOW BYTE
1758 POKE 45354, 1    :REM:  SET MAX RECOGNITION THRESHOLD HIGH BYTE
1760 REM ------------------------------------------------------------
1765 FOR DE = 1 TO 3250     :REM:  SET UP DELAY LOOP IN ORDER TO ALLOW
1770 NEXT DE                :REM:  USER TO VIEW OPENING SCREEN DISPLAY
1775 REM-------------------------------------------------------------
1780 REM*************************************************************
1785 REM: CHECK IF 'COMMANDER' VOCABULARY IS ON DISK & DO ACCORDINGLY
1790 REM
1795 OPEN 15,8,15                 :REM:  OPEN CHANNEL TO DISK DRIVE
1800 OPEN 3,8,3:COMMANDER,S,R     :REM:  OPEN CHANNEL FOR 'COMMANDER'
1805 CLOSE 3                      :REM:  CLOSE CHANNEL
1810 INPUT#15,CS$                 :REM:  INPUT THE FILE STATUS
1815 CLOSE 15                     :REM:  CLOSE CHANNEL
1820 REM ------------------------------------------------------------
1825 REM - IN NEXT STATEMENT, IF CS$ <> "62" THEN 'COMMANDER' VOCABULARY
1830 REM - HAS BEEN TRAINED BEFORE AND WAS PUT ON DISK SO LOAD AND STASH
1835 REM ------------------------------------------------------------
1840 REM     --100--
1850 IF CS$ <> "62" GOTO 2430     :REM:  GOTO LOAD 'COMMANDER'     (120)
1855 REM*************************************************************
1860 REM:  CS$="62" & 'COMMANDER' IS NOT ON DISK SO TELL USER THAT IT HAS
1865 REM:  NOT BEEN TRAINED BEFORE & PREPARE FOR THE 'COMMANDER' TRAINING
1870 REM
1875 PRINT " * * * * * * * * * * * * * * * * * * * * * *"
1880 PRINT
1885 PRINT " * THE COMMANDER FILE HAS NOT BEEN      *"
1890 PRINT
1895 PRINT " * TRAINED BY YOU BEFORE. YOU WILL      *"
1900 PRINT
1905 PRINT " * NOW BE GUIDED THROUGH THE TRAIN-     *"
1910 PRINT
1915 PRINT " * ING PROCESS.  IN THE FUTURE, YOU     *"
1920 PRINT
1925 PRINT " * WILL HAVE THE OPTION OF RETRAIN-     *"
1930 PRINT
1935 PRINT " * ING THIS PARTICULAR VOCABULARY OF    *"
1940 PRINT
1945 PRINT " * PHRASES OR USING THIS VOCABULARY     *"
1950 PRINT
1955 PRINT " * NOW TRAINED & LOADED IN FROM DISK    *"
```

```
1960 PRINT
1965 PRINT " * * * * * * * * * * * * * * * * * * * * * * *"
1970 REM                   ------------------------------------------
1975 FOR DE = 1 TO 9000 :REM:  DO A DELAY LOOP SO THAT USER CAN
1980 NEXT DE            :REM:  VIEW THE MESSAGE ABT 'COMMANDER'
1985 REM                ------------------------------------------
1990 C1 = 1             :REM:  DOING FIRST TIME TRAINING SO SET C1 TRUE
1995 REM***************************************************************
2000 REM:  DO PREREQUISITES AND THEN DO 'COMMANDER' VOCABULARY TRAINING
2005 REM
2010 REM   --120--
2020 CT = 0             :REM:  ZERO OUT CT BEFORE TRAINING A VOCABULARY
2025 GOSUB 6690         :REM:  CALL BLANKTEMPLATEAREA                (580)
2030 IF C1=0 THEN CU=1  :REM:  IF C1=0 THIS IS UPDATE OR RETRAINING
2031 PRINT "PUT MICROPHONE IN POSITION TO SPEAK INTO"
2032 PRINT "  SPEAK ANY PHRASE INTO THE MICROPHONE"
2033 PRINT "   WHEN READY TO BEGIN THE TRAINING"
2034 POKE 49935, 3      :REM:  SET RECOGNITION FOR GROUP 3 (16-23) ONLY
2035 GOSUB 5880         :REM:  CALL GET-VOICE-INPUT                  (530)
2039 REM ------------------------------------------------------------
2040 REM - THE NEXT INSTRUCTION IS PLACE WHERE TRAINING LOOP BEGINS & IS
2045 REM - THE LINE TO WHICH LABEL 117 BRANCHES WHEN THE PHRASE 'AFFIRM'
2050 REM - IS RECOGNIZED.  THIS MEANS THAT THE USER WANTS TO DO ANOTHER
2052 REM - TRAINING & 'BEGIN TRAINING' IS RECOGNIZED FROM USER AFTER IT'S
2053 REM - ESTABLISHED CT IS LESS THAN 5 MEANING THE USER HASN'T ALREADY
2055 REM - REACHED THE LIMIT OF FIVE TRAINS IMPOSED BY THE PROGRAM.
2060 REM ------------------------------------------------------------
2065 REM   --105--
2075 FOR PH = 0 TO 4     :REM:  LOOP TO TRAIN 'COMMANDER' PHRASES 0-4
2080 VO$(PH) = CR$(PH)   :REM:  PASS VO$ TO TRAIN
2085 GOSUB 5935          :REM:  CALL TRAIN                            (535)
2090 NEXT PH             :REM:  END FOR/NEXT LOOP
2095 FOR PH = 8 TO 10    :REM:  LOOP TO TRAIN 'COMMANDER' PHRASES 8-10
2098 VO$(PH) = CR$(PH)   :REM:  PASS VO$ TO TRAIN
2100 GOSUB 5935          :REM:  CALL TRAIN                            (535)
2105 NEXT PH             :REM:  END FOR/NEXT LOOP
2110 FOR PH = 16 TO 17   :REM:  LOOP TO TRAIN 'COMMANDER' PHRASES 16-17
2112 VO$(PH) = CR$(PH)   :REM:  PASS VO$ TO TRAIN
2115 GOSUB 5935          :REM:  CALL TRAIN                            (535)
2120 NEXT PH             :REM:  END FOR/NEXT LOOP
2125 PH = 24             :REM:  LOOP TO TRAIN 'COMMANDER' PHRASE 24
2128 VO$ (PH) = CR$(PH)  :REM:  PASS VO$ TO TRAIN
2130 GOSUB 5935          :REM:  CALL TRAIN                            (535)
2135 CT = CT + 1         :REM:  COUNTER OF # TIMES TRAINING HAS OCCURRED
2140 REM ------------------------------------------------------------
2145 REM - THE NEXT LINE IS THE PLACE THAT 112 BRANCHES TO WHEN
2150 REM - RECOGNITION OF PHRASES 'AFFIRM', 'NEGATE', OR 'STOP
2155 REM - RECOGNITION' HAS NOT OCCURRED FOR 3 CONSECUTIVE TIMES
2160 REM - THROUGH THIS LOOP.  THE FIRST 2 TIMES, 112 WILL LOOP
2161 REM - TO 110 TO QUERY USER ABOUT TRAINS BUT NOT TO ZERO CT.
2165 REM ------------------------------------------------------------
2170 REM   --107--
2180 FR = 0             :REM:  FR SUM SET = 0 BEFORE RECOGNITION TRIES
2185 REM   --110--
2195 GOSUB 6585         :REM:  CALL ASKUSER-ABOUT-MORETRAINS         (570)
```

```
2200 GOSUB 6645        :REM:  CALL MATCHCOMMANDERGROUP2               (575)
2205 DM = PEEK(151)-7  :REM:  CONVERT RECOGNIZED 8-10 TO 1-3 FOR 'ON'
2210 REM-----------------------------------------------------------------
2215 REM - NEXT LINE, GOTO DO MORE TRAINING OF 'COMMANDER'     (117) OR
2220 REM -            GOTO EXIT TRAINING OF 'COMMANDER'        (122) OR
2225 REM -            GOTO DO STOP RECOGNITION FUNCTION        (115)
2230 REM-----------------------------------------------------------------
2235 REM   --112--
2245 ON DM GOTO 2370,  2480,  2335
2250 REM*****************************************************************
2255 REM: NO MATCH HAS OCCURRED SO DO THE  FOLLOWING
2260 REM
2265 FR = FR + 1              :REM:  NO MATCH OCCURRED SO ADD 1 TO FR SUM
2266 IF FR<3 THEN PRINT "...NO RECOGNITION, TRY AGAIN...(";PEEK(151);")"
2267 IF FR<3 THEN FOR DE = 1 TO 900 : NEXT DE
2270 IF FR<3 THEN GOTO 2195  :REM:  LOOP BACK TWICE IF NO MATCH    (110)
2275 FP = 8                   :REM:  PASS  8 TO RETRAINGROUP
2280 LP = 10                  :REM:  PASS 10 TO RETRAINGROUP
2285 V$ = "COMMANDER"         :REM:  PASS "COMMANDER" TO RETRAINGROUP
2286 TT = CT                  :REM:  PASS CT TO RETRAINGROUP
2287 FOR PH = FP TO LP        :REM:  LOOP TO SET VO$ PHRASES = TO CR$
2288 VO$(PH) = CR$(PH)        :REM:  PASS CR$(PH) TO RETRAINGROUP
2289 NEXT PH                  :REM:  END FOR/NEXT LOOP
2290 GOSUB 6020               :REM:  CALL RETRAINGROUP               (540)
2295 GOSUB 6100               :REM:  CALL STASHCOMMANDER             (545)
2300 IF C1 = 0 THEN CU = 1    :REM:  'COMMANDER' UPDATED SO SET CU = 1
2305 GOTO 2180                :REM:  LOOP BACK AFTER RETRAINING    (107)
2310 REM*****************************************************************
2315 REM:  'STOP RECOGNITION' RECOGNIZED FROM GRP 2 SO CALL ITS SUBROUTINE
2320 REM
2325 REM   --115--
2335 GOSUB 5575       :REM:  CALL STOPRECOGNITION                    (500)
2340 GOTO 2180        :REM:  LOOP BACK AFTER RECOGNITION RESTORED  (107)
2345 REM*****************************************************************
2350 REM:  'AFFIRM' RECOGNIZED IN GRP 2, DO MORE TRAINING OR LIMIT REACHED
2355 REM
2360 REM   --117--
2370 IF CT < 5 THEN GOSUB 6385   :REM:  CALL MATCHCOMMANDERGROUP3 (565)
2372 IF CT < 5 THEN GOTO 2075    :REM:  GO TO DO MORE TRAINING    (105)
2375 GOSUB 5635                  :REM:  CALL ENOUGH-TRAINING-DONE (505)
2380 GOTO 2480                   :REM:  GOTO CALL STASHCOMMANDER  (122)
2385 REM*****************************************************************
2390 REM:  LOAD 'COMMANDER' AND/OR STASH IT IN REU MEMORY
2400 REM-----------------------------------------------------------------
2405 REM - NEXT LINE IS BRANCHED TO FROM 100 AFTER IT'S DETERMINED THAT
2410 REM - 'COMMANDER' HAS BEEN TRAINED BEFORE & IS ALREADY ON THE DISK
2415 REM-----------------------------------------------------------------
2420 REM   --120--
2430 PRINT "-LOADING 'COMMANDER' FILE-"
2435 SYS 49929"COMMANDER",8
2440 REM-----------------------------------------------------------------
2445 REM - THE NEXT INSTRUCTION IS BRANCHED TO FROM LABEL 117 AFTER IT
2450 REM - IS DETERMINED THAT ENOUGH TRAINS HAVE BEEN PERFORMED OR AF-
2455 REM - TER 'NEGATE' IS RECOGNIZED IN A RESPONSE TO LABEL 112 WHICH
2460 REM - CHECKS FOR RECOGNITION IN 'COMMANDER' VOCABULARY GROUP #2
```

```
2470 REM   --122--
2480 GOSUB 6100                   :REM:  CALL STASHCOMMANDER          (545)
2482 CT = 2                       :REM:  IF RETRAIN OCCURS WILL DO 2 TIMES
2485 REM*************************************************************
2490 REM:  MAIN MENU USER INTERFACE
2495 REM
2510 REM-------------------------------------------------------------
2515 REN - THE NEXT INSTRUCTION IS THE PLACE THAT LABEL 130 BRANCHES TO
2520 REM - WHEN RECOGNITION OF PHRASES 'APPLICATION PROGRAM', 'ENVIRON-
2525 REM - MENTAL CONTROLLER', 'PROGRAM ESCAPE', 'RETRAIN COMMANDER',
2530 REM - OR 'STOP RECOGNITION' HAS NOT OCCURRED FOR THREE CONSECUTIVE
2535 REM - TIMES THROUGH THIS LOOP.  THE FIRST TWO TIMES THIS HAPPENS,
2540 REM - LABEL 130 WILL LOOP BACK TO LABEL 127 TO ASK FOR USER INPUT.
2545 REM
2550 REM   --125--
2560 FR = 0
2565 REM   --127--
2575 PRINT  "   WOULD YOU LIKE TO (USE)"
2580 PRINT  "             -'APPLICATION  PROGRAM'?"
2585 PRINT  "   OR        -'ENVIRONMENTAL CONTROLLER'?"
2590 PRINT  "   OR        -'PROGRAM  ESCAPE'?"
2595 PRINT  "   OR        -'RETRAIN COMMANDER'?"
2600 PRINT  "   OR        -'STOP RECOGNITION'?"
2605 POKE 49935, 1             :REM:  SET RECOGNITION FOR GROUP 1  (0-7)
2610 GOSUB 5880          :REM:  CALL GET-VOICE-INPUT               (530)
2615 DM = PEEK(151) + 1 :REM:  SET DM RECOGNIZED PHRASE # = 1
2620 REM -----------------------------------------------------------
2625 REM - NEXT LINE, GOTO USE A USER-CHOSEN APPLICATION PROGRAM  (200)
2630 REM - OR GOTO USE ENVIRONMENTAL CONTROL APPLICATION          (300)
2635 REM - OR GOTO EXIT THE PROGRAM 'VOICEMANAGER'                (135)
2640 REM - OR GOTO RETRAIN THE 'COMMANDER' VOCABULARY             (102)
2645 REM - OR GOTO DO STOP RECOGNITION FUNCTION                   (132)
2650 REM------------------------------------------------------------
2655 REM   --130--
2665 ON DM GOTO 3600, 3970, 2795, 2020, 2755
2670 REM*************************************************************
2675 REM:   NO MATCH WITH ANY GROUP 1 PHRASE SO DO THE FOLLOWING
2680 REM
2685 FR = FR + 1               :REM:  NO MATCH OCCURRED SO ADD 1 TO FR SUM
2686 IF FR<3 THEN PRINT "...NO RECOGNITION, TRY AGAIN...(";PEEK(151);")"
2687 IF FR<3 THEN FOR DE = 1 TO 900 : NEXT DE
2690 IF FR<3 THEN GOTO 2575  :REM:  LOOP BACK TWICE IF NO MATCH   (127)
2695 FP = 0                  :REM:  PASS 0 TO RETRAINGROUP
2700 LP = 4                  :REM:  PASS 4 TO RETRAINGROUP
2705 V$ ="COMMANDER"         :REM:  PASS "COMMANDER" TO RETRAINGROUP
2706 FOR PH = FP TO LP       :REM:  LOOP TO SET VO$ PHRASES = TO CR$
2707 VO$(PH) = CR$(PH)       :REM:  PASS CR$(PH) TO RETRAINGROUP
2708 NEXT PH                 :REM:  END FOR/NEXT LOOP
2710 GOSUB 6020              :REM:  CALL RETRAINGROUP               (540)
2715 GOSUB 6100              :REM:  CALL STASHCOMMANDER             (545)
2720 IF C1 = 0 THEN CU = 1   :REM:  'COMMANDER' UPDATED SO SET CU = 1
2725 GOTO 2560               :REM:  LOOP BACK AFTER RETRAINING     (125)
2730 REM*************************************************************
2735 REM: 'STOP RECOGNITION' RECOGNIZED FROM GRP 1 SO CALL ITS SUBROUTINE
2740 REM
```

```
2745 REM   --132--
2755 GOSUB 5575               :REM:  CALL STOPRECOGNITION          (500)
2760 GOTO 2560                :REM:  LOOP BACK, RECOG RESTORED     (125)
2765 REM***************************************************************
2770 REM:  CLOSE OUT MAIN PROGRAM VOICEMANAGER
2775 REM
2780 REM   --135--
2795 IF CU = 0 THEN GOTO 3110 :REM:  IF CU = 0 THEN GOTO CHECK C1 (152)
2800 REM ---------------------------------------------------------------
2805 REM - CU = 1 SO QUERY USER AND DO WHAT THE USER SAYS TO DO
2810 REM ---------------------------------------------------------------
2815 REM   --137--
2825 FR = 0                  :REM:  FR SET BEFORE RECOGNITION TRIES
2830 REM   --140--
2840 PRINT "   THE 'COMMANDER' VOCABULARY HAS BEEN"
2845 PRINT "   UPDATED DURING THIS RUN OF 'VOICE -"
2850 PRINT "   MANAGER'. WOULD YOU LIKE TO SAVE"
2855 PRINT "   THIS UPDATED 'COMMANDER' VOCABU-"
2860 PRINT "   LARY TO DISK REPLACING THE CURRENT"
2862 PRINT "   VOCABULARY THAT IS THERE NOW?"
2865 GOSUB 6645              :REM:  CALL MATCHCOMMANDERGROUP2       (575)
2870 DM = PEEK(151) - 7      :REM:  CONVERT MATCHED PHRASES 8-10 TO 1-3
2875 REM ---------------------------------------------------------------
2880 REM - NEXT LINE, GOTO SAVE THE UPDATED 'COMMANDER'         (150) OR
2885 REM              GOTO PRINT MESSAGE & ON TO CHECK EU       (141) OR
2890 REM              GOTO DO 'STOP RECOGNITION' FUNCTION       (145)
2895 REM ---------------------------------------------------------------
2900 REM   --142--
2910 ON DM GOTO 3070, 3035, 3000
2915 REM***************************************************************
2920 REM:  NO MATCH HAS OCCURRED SO DO THE FOLLOWING
2925 REM
2930 FR = FR + 1                :REM:  NO RECOGNITION SO ADD 1 TO FR SUM
2932 IF FR<3 THEN PRINT"...NO RECOGNITION, TRY AGAIN...(";PEEK(151);")"
2933 IF FR<3 THEN FOR DE = 1 T0 900 : NEXT DE
2935 IF FR<3 THEN GOTO 2840   :REM:  LOOP BACK TWICE IF NO MATCH  (140)
2940 FP = 8                     :REM:  PASS  8 TO RETRAINGROUP
2945 LP = 10                    :REM:  PASS 10 TO RETRAINGROUP
2950 V$ = "COMMANDER"           :REM:  PASS 'COMMANDER' TO RETRAINGROUP
2951 TT = CT                    :REM:  PASS CT TO RETRAINGROUP
2952 FOR PH = FP TO LP          :REM:  LOOP TO SET VO$ PHRASES = TO CR$
2953 VO$(PH) = CR$(PH)          :REM:  PASS CR$(PH) TO RETRAINGROUP
2954 NEXT PH                    :REM:  END FOR/NEXT LOOP
2955 GOSUB 6020                 :REM:  CALL RETRAINGROUP             (540)
2960 GOSUB 6100                 :REM:  CALL STASHCOMMANDER           (545)
2965 IF C1 = 0 THEN CU = 1      :REM:  'COMMANDER' UPDATED SO SET CU = 1
2970 GOTO 2825                  :REM:  LOOP BACK AFTER RETRAINING   (137)
2975 REM***************************************************************
2980 REM: 'STOP RECOGNITION' RECOGNIZED FROM GRP 2 SO CALL ITS SUBROUTINE
2985 REM
2990 REM   --145--
3000 GOSUB 5575  :REM:  CALL STOPRECOGNITION                        (500)
3005 GOTO 2825   :REM:  LOOP BACK AFTER RECOGNITION RESTORED        (137)
3010 REM***************************************************************
3015 REM:  NEGATIVE RECOGNIZED IN GRP 2 SO PRINT MESSAGE & SAVE TO DISK
```

```
3020 REM
3025 REM   --147--
3035 PRINT "...UPDATED 'COMMANDER' WON'T BE SAVED..."
3037 FOR DE = 1 T0 1000         :REM:  DELAY SO USER CAN
3038 NEXT DE                    :REM:  VIEW THE MESSAGE
3040 GOTO 3165                  :REM:  GOTO CHECK EU                  (155)
3045 REM***********************************************************************
3050 REM:  'AFFIRM' RECOGNIZED FROM GRP 2 SO PRINT MESSAGE & SAVE TO DISK
3055 REM
3060 REM   --150--
3070 PRINT "'COMMANDER' VOCABULARY IS BEING SAVED..."
3075 OPEN 15,8,15               :REM:  OPEN CHANNEL TO DISK DRIVE
3080 PRINT#15,"S0:COMMANDER"    :REM:  DELETE OLD 'COMMANDER' ON DISK
3085 CLOSE 15                   :REM:  CLOSE CHANNEL TO DISK DRIVE
3090 SYS 49926"COMMANDER",8     :REM:  CALL 'TPUT' IN VOICE/RECOG TO SAVE
3095 REM - C1 WILL = 0 WHEN CU = 1, THIS MEANS PREVIOUS
3100 REM - LINE IS FOLLOWED BY AN AUTOMATIC GOTO
3105 REM -----------------------------------------------------------------
3110 IF C1 = 0 THEN GOTO 3165     :REM:  IF C1 = 0 GOTO CHECK EU   (155)
3115 REM***********************************************************************
3120 REM:  C1 = 1 SO SAVE 'COMMANDER' TO DISK
3125 REM
3130 PRINT "'COMMANDER' VOCABULARY IS BEING SAVED..."
3135 SYS 49926"COMMANDER",8     :REM:  CALL 'TPUT' IN VOICE/RECOG TO SAVE
3140 REM***********************************************************************
3145 REM:  NOW CHECK EU FOR 'ECCOMMANDS' VOCABULARY UPDATE
3150 REM
3155 REM   --155--
3165 IF EU = 0 THEN GOTO 3485     :REM:  IF EU = 0 GOTO CHECK E1   (172)
3170 REM
3175 REM - EU = 1 SO QUERY USER AND DO WHAT THE USER SAYS TO DO
3180 REM------------------------------------------------------------------
3185 REM   --157--
3195 FR = 0                     :REM:  FR SET BEFORE RECOGNITION TRIES
3200 REM   --160--
3210 PRINT "      THE 'ECCOMMANDS' VOCABULARY HAS BEEN"
3215 PRINT "      UPDATED DURING THIS RUN OF 'VOICE-"
3220 PRINT "      MANAGER'.  WOULD YOU LIKE TO SAVE"
3225 PRINT "      THIS UPDATED 'ECCOMMANDS' VOCABU-"
3230 PRINT "      LARY TO DISK REPLACING THE CURRENT"
3232 PRINT "      VOCABULARY THAT IS THERE NOW?"
3235 GOSUB 6645                 :REM:  CALL MATCHCOMMANDERGROUP2     (575)
3240 DM = PEEK(151) - 7         :REM:  CONVERT MATCHED PHRASES 8-10 TO 1-3
3245 REM------------------------------------------------------------------
3250 REM - IN NEXT LINE, GOTO SAVE THE UPDATED 'ECCOMMANDS'    (170) OR
3255 REM                 GOTO PRINT MESSAGE & THEN TO EXIT     (1AO) OR
3260 REM                 GOTO DO 'STOP RECOGNITION' FUNCTION   (1AP)
3265 REM------------------------------------------------------------------
3270 REM   --162--
3280 ON DM GOTO 3440, 3405, 3370
3285 REM***********************************************************************
3290 REM:  NO MATCH HAS OCCURRED SO DO THE FOLLOWING
3295 REM
3300 FR = FR + 1                :REM:  NO MATCH OCCURRED SO ADD 1 TO FR SUM
3302 IF FR<3 THEN PRINT "...NO RECOGNITION, TRY AGAIN...("; PEEK (151);")"
```

```
3303 IF FR<3 THEN FOR DE = 1 TO 900 : NEXT DE
3305 IF FR<3 THEN GOTO 3210    :REM:  LOOP BACK TWICE IF NO MATCH
3310 FP = 8                    :REM:  PASS  8 TO RETRAINGROUP
3315 LP = 10                   :REM:  PASS 10 TO RETRAINGROUP
3320 V$ = "COMMANDER"          :REM:  PASS "COMMANDER" TO RETRAINGROUP
3321 TT = CT                   :REM:  PASS CT TO RETRAINGROUP
3322 FOR PH = FP TO LP         :REM:  LOOP TO SET VO$ PHRASES = TO CR$
3323 VO$(PH) = CR$(PH)         :REM:  PASS CR$(PH) TO RETRAINGROUP
3324 NEXT PH                   :REM:  END FOR/NEXT LOOP
3325 GOSUB 6020                :REM:  CALL RETRAINGROUP            (540)
3330 GOSUB 6100                :REM:  CALL STASHCOMMANDER          (545)
3335 IF C1 = 0 THEN CU = 1     :REM:  'COMMANDER' UPDATED SO SET CU = 1
3340 GOTO 3195                 :REM:  LOOP BACK AFTER RETRAINING   (157)
3345 REM**************************************************************
3350 REM: 'STOP RECOGNITION' RECOGNIZED FROM GRP 2 SO CALL ITS SUBROUTINE
3355 REM
3360 REM   --165--
3370 GOSUB 5575                :REM:  CALL STOPRECOGNITION         (500)
3375 GOTO 3195                 :REM:  LOOP BACK AFTER RECOGNITION RESTORED
3380 REM**************************************************************
3385 REM: 'NEGATIVE'  RECOGNIZED FROM GRP 2 SO PRINT MESSAGE & GOTO EXIT
3390 REM
3395 REM   --167--
3405 PRINT "UPDATED 'ECCOMMANDS' WON'T BE SAVED..."
3406 FOR DE = 1 TO 1000        :REM:  DELAY SO USER CAN
3407 NEXT DE                   :REM:  VIEW THE MESSAGE
3410 GOTO 3550                 :REM:  GOTO EXIT THE PROGRAM        (175)
3415 REM**************************************************************
3420 REM:  'AFFIRM' RECOGNIZED FROM GRP 2 SO PRINT MESSAGE & SAVE TO DISK
3425 REM
3430 REM   --170--
3440 PRINT "'ECCOMMANDS' VOCABULARY IS BEING SAVED..."
3445 GOSUB 6225                :REM:  CALL FETCH-ECCOMMANDS        (560)
3450 OPEN 15,8,15              :REM:  OPEN CHANNEL TO DISK DRIVE
3455 PRINT#15,"S0:ECCOMMANDS" :REM:  DELETE OLD 'ECCOMMANDS' ON DISK
3460 CLOSE 5                   :REM:  CLOSE CHANNEL TO DISK DRIVE
3465 SYS 49926"ECCOMMANDS",8   :REM:  CALL 'TPUT' IN VOICE/RECOG TO SAVE
3470 REM - E1 WILL = 0 WHEN EU = 1, THIS MEANS PREVIOUS
3475 REM   --172--                  LINE IS FOLLOWED BY AN AUTOMATIC GOTO
3480 REM--------------------------------------------------------------
3485 IF E1 = 0 THEN GOTO 3550 :REM:  IF E1 = 0 GOTO MESSAGE, EXIT (175)
3490 REM**************************************************************
3495 REM: E1 = 1 SO SAVE 'ECCOMMANDS' TO DISK
3500 REM
3505 PRINT "'ECCOMMANDS' VOCABULARY IS BEING SAVED..."
3510 GOSUB 6225                :REM:  CALL FETCH-ECCOMMANDS        (560)
3515 SYS 49926"ECCOMMANDS",8   :REM:  CALL 'TPUT' IN VOICE/RECOG TO
3520 REM*********************************
3525 REM--------------------------------------------------------------
3530 REM- NEXT LINE BRANCHED TO FROM 167 AND 172 - THIS IS THE END
3535 REM--------------------------------------------------------------
3540 REM   --175--
3550 PRINT "  'VOICEMANAGER' SAYS GOOD-BYE!"
3555 FOR DE = 1 TO 1250        :REM:  DELAY MOMENTARILY FOR USER
3560 NEXT DE                   :REM:  TO SEE A 'GOOD-BYE' MESSAGE
```

```
3565 PRINT ""                    :REM:  CLEAR THE SCREEN
3567 CLOSE 2                     :REM:  CLOSE THE USER PORT
3570 END
3575 REM:      END VOICEMANAGER
3580 REM-------------------------------------------------------------
3585 REM*************************************************************
3590 REM:      APPLICATION PROGRAM
3595 REM
3597 REM  --200--
3600 PRINT "    APPLICATION PROGRAM BY SPEECH INPUT"
3602 FOR DE = 1 TO 1250           :REM:  DELAY TO ALLOW USER
3603 NEXT DE                      :REM:  TO SEE THE MESSAGE
3604 IF AP = 1 THEN GOTO 3668     :REM:  GOTO START THE PROGRAM
3606 AP = 1                       :REM:  APPLICATION PROGRAM USED THIS RUN
3608 OPEN 15,8,15                 :REM:  OPEN CHANNEL TO DISK DRIVE
3610 OPEN 3,8,3,"ALPHABET,S,R"    :REM:  OPEN CHANNEL FOR 'ALPHABET'
3612 CLOSE 3                      :REM:  INPUT THE FILE STATUS
3614 INPUT#15,AS$                 :REM:  CLOSE THE CHANNEL
3618 IF AS$ <> "62" GOTO 2560     :REM:  GOTO MAIN MENU USER INTERFACE
3620 OPEN 15,8,15                 :REM:  OPEN CHANNEL TO DISK DRIVE
3622 OPEN 3,8,3,"PUNCTUATION,S,R":REM:  OPEN CHANNEL FOR 'PUNCTUATION'
3624 CLOSE 3                      :REM:  CLOSE THE CHANNEL
3626 INPUT#15,PS$                 :REM:  INPUT THE FILE STATUS
3628 CLOSE 15                     :REM:  CLOSE THE CHANNEL
3630 IF PS$ <> "62" GOTO 2560     :REM:  GOTO MAIN MENU USER INTERFACE
3632 OPEN 15,8,15                 :REM:  OPEN CHANNEL TO DISK DRIVE
3634 OPEN 3,8,3,"NUMBERSPLUS,S,R":REM:  OPEN CHANNEL FOR 'NUMBERSPLUS"'
3636 CLOSE 3                      :REM:  CLOSE THE CHANNEL
3638 INPUT#15,NS$                 :REM:  INPUT THE FILE STATUS
3640 CLOSE 15                     :REM:  CLOSE THE CHANNEL
3642 IF PS$ <> "62" GOTO 2560     :REM:  GOTO MAIN MENU USER INTERFACE
3644 PRINT "-LOADING 'ALPHABET' FILE-"    :REM:  PRINT MESSAGE &
3646 SYS 49929"ALPHABET",8        :REM:  LOAD THE VOCABULARY
3648 GOSUB 6704                   :REM:  CALL STASHALPHABET        (585)
3650 AT = 2                       :REM:  SET TIMES TRAINED TO 2
3652 PRINT "-LOADING 'PUNCTUATION' FILE -":REM:  PRINT MESSAGE &
3654 SYS 44929"PUNCTUATION",8     :REM:  LOAD THE VOCABULARY
3656 GOSUB 6711                   :REM:  CALL STASHALPHABET        (590)
3658 PT = 2                       :REM:  SET TIMES TRAINED TO 2
3660 PRINT "-LOADING 'NUMBERSPLUS' FILE -":REM:  PRINT MESSAGE &
3662 SYS 44929"NUMBERSPLUS",8     :REM:  LOAD THE VOCABULARY
3664 GOSUB 6718                   :REM:  CALL STASHNUMBERSPLUS     (595)
3666 NT = 2                       :REM:  SET TIMES TRAINED TO 2
3668 GOSUB 6725                   :REM:  CALL FETCHALPHABET        (600)
3670 PRINT "-LOADING 'VOICEINTERPRETER' FILE -"
3672 IF C = 0 THEN C = 1 : LOAD"VOICEINTERPRETER",8,1
3674 PRINT "-LOADING 'TEST' FILE -"
3676 IF D = 0 THEN D = 1 : LOAD"TEST",8,1
3678 POKE 810, 216             :REM:  SET  LOW BYTE GETIN VECTOR ($D8)
3679 POKE 811, 2               :REM:  SET HIGH BYTE GETIN VECTOR ($02)
3680 SYS 49152
3682 GOSUB 6185                :REM:  CALL FETCHCOMMANDER          (555)
3684 GOTO 2560                 :REM:  GO TO MAIN MENU USER INTERFACE
3685 REM:                END APPLICATION
3690 REM                 ---------------
```

```
3945 REM***************************************************************
3950 REM:  ENVIRONMENTAL CONTROL
3955 REM
3960 REM   --300--
3970 PRINT "ENVIRONMENTAL CONTROL BY SPEECH INPUT"
4005 FOR DE = 1 TO 1250          :REM:  DO A DELAY LOOP FOR
4010 NEXT DE                     :REM:  VIEWING THE MESSAGE
4012 IF EC = 1 THEN GOSUB 4595  :REM:  ENV CONTROLLER USED BEFORE THIS RUN
4013 EC = 1                      :REM:  INDICATES ENV CONTROL USED THIS RUN
4015 REM***************************************************************
4020 REM:  CHECK IF 'ECCOMMANDS' VOCABULARY IS ON DISK & DO ACCORDINGLY
4025 REM
4030 OPEN 15,8,15                    :REM:  OPEN CHANNEL TO DISK DRIVE
4035 OPEN 3,8,3,"ECCOMMANDS,S,R"     :REM:  OPEN CHANNEL FOR 'ECCOMMANDS'
4040 CLOSE 3                         :REM:  CLOSE THE CHANNEL
4045 INPUT#15,ES$                    :REM:  INPUT THE FILE STATUS
4050 CLOSE 15                        :REM:  CLOSE THE CHANNEL
4055 REM --------------------------------------------------------------
4060 REM - IN NEXT INSTRUCTION, IF ES$ <> 62 THEN 'ECCOMMANDS' VOCABULARY
4065 REM - HAS BEEN TRAINED BEFORE & IS ON THE DISK SO GO LOAD & STASH IT
4070 REM --------------------------------------------------------------
4075 REM   --302--
4085 IF ES$ <> "62" GOTO 4650        :REM:  GOTO LOAD 'ECCOMMANDS'
4090 REM***************************************************************
4095 REM: ES$ ="62" & 'ECCOMMANDS' IS NOT ON DISK SO TELL USER THAT IT
4100 REM: NOT BEEN TRAINED BEFORE & PREPARE FOR THE 'ECCOMMANDS' TRAINING
4105 REM
4110 PRINT  "    * * * * * * * * * * * * * * * * * * * * * * *"
4115 PRINT
4120 PRINT  "    *   THE ECCOMMANDS FILE HAS NOT BEEN     *"
4125 PRINT
4130 PRINT  "    *   TRAINED BY YOU BEFORE.  YOU WILL     *"
4135 PRINT
4140 PRINT  "    *   NOW BE GUIDED THRU THE TRAIN-        *"
4145 PRINT
4150 PRINT  "    *   ING PROCESS.  IN THE FUTURE, YOU     *"
4155 PRINT
4160 PRINT  "    *   WILL HAVE THE OPTION OF RETRAIN-     *"
4165 PRINT
4170 PRINT  "    *   ING THIS PARTICULAR VOCABULARY OF    *"
4175 PRINT
4180 PRINT  "    *   PHRASES OR USING THIS VOCABULARY     *"
4185 PRINT
4190 PRINT  "    *   NOW TRAINED & LOADED IN FROM DISK.  *"
4195 PRINT
4200 PRINT  "    * * * * * * * * * * * * * * * * * * * * * * *"
4205 REM --------------------------------------------------------------
4210 FOR DE = 1 TO 9000 :REM:  DO A DELAY LOOP SO THAT USER CAN
4215 NEXT DE            :REM:  VIEW MESSAGE ABOUT 'ECCOMMANDS'
4220 REM --------------------------------------------------------------
4225 E1 = 1             :REM:  DOING FIRST TIME TRAINING SO SET E1 TRUE
4230 REM***************************************************************
4235 REM: DO PREREQUISITES & THEN DO THE 'ECCOMMANDS' VOCABULARY TRAINING
4240 REM
4245 REM   --305--
```

```
4255 ET = 0                :REM:  ZERO ET BEFORE TRAINING
4265 IF E1 = 0 THEN EU = 1 :REM:  IF E1=0 THIS IS UPDATE/TRAINING
4270 REM -----------------------------------------------------------------
4275 REM - THE NEXT INSTRUCTION IS PLACE WHERE TRAINING LOOP BEGINS & IT
4280 REM - IS THE LINE TO WHICH 320 BRANCHES WHEN THE PHRASE 'AFFIRM' IS
4285 REM - RECOGNIZED.  THIS MEANS THE USER WANTS TO DO ANOTHER TRAINING
4290 REM - AND ET<5 MEANING THE USER HASN'T REACHED THE FIVE TRAIN LIMIT
4295 REM -----------------------------------------------------------------
4300 REM  --307--
4310 GOSUB 6385                   :REM:  CALL MATCHCOMMANDERGROUP3   (565)
4312 GOSUB 6690                   :REM:  CALL BLANKTEMPLATEAREA      (580)
4313 IF ET > 0 THEN GOSUB 6225    :REM:  CALL FETCHECCOMMAND         (560)
4317 FOR PH = 0 TO 18             :REM:  LOOP TO TRAIN PHRASES 0-18
4318 VO$(PH) = EC$(PH)            :REM:  EC$ IS PASSED TO SUBROUTINE TRAIN
4320 GOSUB 5935                   :REM:  CALL TRAIN                  (535)
4325 NEXT PH                      :REM:  END FOR/NEXT LOOP
4330 FOR PH = 24 TO 28            :REM:  LOOP TO TRAIN PHRASES 24-28
4332 VO$(PH) = EC$(PH)            :REM:  EC$ IS PASSED TO SUBROUTINE TRAIN
4335 GOSUB 5935                   :REM:  CALL TRAIN                  (535)
4340 NEXT PH                      :REM:  END FOR/NEXT LOOP
4345 ET = ET + 1                  :REM:  INCREMENT # TIMES TRAINING
4350 GOSUB 6140                   :REM:  CALL STASH-ECCOMMANDS       (550)
4355 GOSUB 6185                   :REM:  CALL FETCHCOMMANDER         (555)
4360 REM -----------------------------------------------------------------
4365 REM - THE NEXT INSTRUCTION IS THE PLACE THAT 315 BRANCHES TO WHEN
4370 REM - RECOGNITION OF PHRASES 'AFFIRM', 'NEGATIVE', OR 'STOPRECOG-
4375 REM - NITION' HASN'T OCCURRED FOR 3 CONSECUTIVE TIMES THROUGH THE
4380 REM - LOOP.  THE FIRST TWO TIMES THIS HAPPENS, 315 WILL LOOP BACK
4385 REM - TO 312 TO QUERY THE USER ABOUT TRAINING BUT NOT ZERO OUT FR
4390 REM -----------------------------------------------------------------
4395 REM  --310--
4405 FR = 0               :REM:  FR SET BEFORE RECOGNITION TRIES
4410 REM  --312--
4420 GOSUB 6585           :REM:  CALL ASKUSER-ABOUT-MORETRAINS   (570)
4425 GOSUB 6645           :REM:  CALL MATCHCOMMANDERGROUP2       (575)
4430 DM = PEEK(151) - 7   :REM:  CONVERT MATCHED PHRASES 8-10 TO 1-3
4435 REM -----------------------------------------------------------------
4440 REM - NEXT LINE, GOTO DO MORE TRAINING OF 'ECCOMMANDS'  (320)  OR
4445 REM -            GOTO EXIT TRAINING OF 'ECCOMMANDS'     (322)  OR
4450 REM -            GOTO DO STOP RECOGNITION FUNCTION      (317)  OR
4455 REM -----------------------------------------------------------------
4460 REM  --315--
4470 ON DM GOTO 4570, 4595, 4535
4475 REM*****************************************************************
4480 REM:  NO MATCH HAS OCCURRED SO DO THE FOLLOWING
4485 REM
4490 FR = FR + 1                :REM:  NO MATCH OCCURRED SO ADD 1 TO FR SUM
4492 IF FR<3 THEN PRINT "...NO RECOGNITION, TRY AGAIN...(";PEEK(151);")"
4493 IF FR<3 THEN FOR DE = 1 TO 900 : NEXT DE
4495 IF FR<3 THEN GOTO 4420     :REM:  LOOP BACK TWICE IF NO MATCH  (312)
4496 FP = 8                     :REM:  PASS  8 TO RETRAINGROUP
4497 LP = 10                    :REM:  PASS 10 TO RETRAINGROUP
4498 V$ = "COMMANDER"           :REM:  PASS "COMMANDER" TO RETRAINGROUP
4499 TT = CT                    :REM:  PASS CT TO RETRAINGROUP
4500 FOR PH = FP TO LP          :REM:  LOOP TO SET VO$ PHRASES = TO CR$
```

```
4501 VO$(PH) = CR$(PH)          :REM:  PASS CR$(PH) TO RETRAINGROUP
4502 NEXT PH                    :REM:  END FOR/NEXT LOOP
4503 GOSUB 6020                 :REM:  CALL RETRAINGROUP             (540)
4504 GOSUB 6100                 :REM:  CALL STASHCOMMANDER           (545)
4505 IF C1 = 0 THEN CU = 1      :REM:  'COMMANDER' UPDATED SO SET CU = 1
4506 GOTO 4405                  :REM:  LOOP BACK AFTER RETRAINING    (310)
4510 REM***************************************************************
4515 REM: 'STOP RECOGNITION; RECOGNIZED FROM 'COMMANDER' GROUP 2
4520 REM
4525 REM   --317--
4535 GOSUB 5575             :REM:  CALL STOPRECOGNITION              (500)
4540 GOTO 4405              :REM:  LOOP BACKRECOGNITION RESTORED     (310)
4545 REM***************************************************************
4550 REM:  'AFFIRM' RECOGNIZED FROM 'COMMANDER' GROUP 2
4555 REM
4560 REM        --320--
4570 IF ET < 5 THEN GOTO 4310 :REM:  ET<5 SO GO DO MORE TRAINING  (307)
4575 GOSUB 5635               :REM:  CALL ENOUGH-TRAINING-DONE    (505)
4577 REM***************************************************************
4579 REM: 'NEGATIVE' RECOGNIZED FROM 'COMMANDER' GROUP 2
4580 REM
4585 REM   --322--
4595 GOSUB 6225               :REM:  CALL FETCH-ECCOMMANDS          (560)
4600 GOTO 4775                :REM:  GO TO ENVIRONMENTAL CONTROL  (330)
4605 REM***************************************************************
4610 REM:  LOAD 'ECCOMMANDS' AND/OR STASH IT IN REU MEMORY
4615 REM
4620 REM --------------------------------------------------------------
4625 REM - NEXT LINE IS BRANCHED TO FROM LABEL 302 AFTER IT IS DETERMINED
4630 REM - THAT 'ECCOMMANDS' HAS BEEN TRAINED BEFORE & IS ALREADY ON DISK
4635 REM --------------------------------------------------------------
4640 REM   --325--
4650 PRINT "  - LOADING 'ECCOMMANDS' FILE -"
4655 SYS 49929 "ECCOMMANDS",8
4660 REM
4690 REM   --327--
4700 GOSUB 6140             :REM:  CALL STASH-ECCOMMANDS             (550)
4702 ET = 2                 :REM:  SET # TRAINS = 2 SINCE NO TRAINS
4705 REM***************************************************************
4710 REM:  ENVIRONMENTAL CONTROL USER INTERFACE
4730 REM --------------------------------------------------------------
4735 REM - THE NEXT INSTRUCTION IS THE ONE THAT IS BRANCHED TO WHEN
4740 REM - RECOGNITION OF NONE OF THE 'EQUIPMENT PIECE' PHRASES HAS
4745 REM - OCCURRED FOR 3 CONSECUTIVE TIMES THROUGH THE LOOP.  THE
4750 REM - 1ST TWO TIMES THIS HAPPENS, LABEL 332 IS BRANCHED TO; ON
4755 REM - THE THIRD ATTEMPT, 330 IS BRANCHED TO SO AS TO SET FR=0.
4760 REM --------------------------------------------------------------
4765 REM   --330--
4775 FR = 0                     :REM:  ZERO THE # OF FAILED RECOGNITIONS
4780 REM   --332--
4790 PRINT "WHAT APPLIANCE TO YOU WANT TO CONTROL?"
4795 FOR PH = 0 TO 14 STEP 2    :REM:  LOOP TO PRINT NAMES
4800 PRINT EC$(PH); EC$(PH + 1) :REM:  PRINT THE NAMES OF DEVICES
4805 NEXT PH                    :REM:  END FOR/NEXT LOOP
4810 G2 = 1                     :REM:  CHECK 2 GROUPS SO SET G2 = 1
```

```
4815 POKE 49935, 1                :REM:  SET RECOGNITION FOR GROUP 1 (0-7)
4817 POKE 45351, 15               :REM:  SET LOW BYTE OF MIN THRESHOLD
4820 GOSUB 5880                   :REM:  CALL GET-VOICE-INPUT (530)
4822 POKE 45351, 35               :REM:  RESET LB OF MIN TO PREVIOUS VALUE
4825 G2 = 0                       :REM:  INPUT OBTAINED SO G2 = 0 AGAIN
4830 MN = PEEK(151)               :REM:  SET MODULE # = PHRASE #
4835 IF MN <= 15 THEN GOTO 4990 :REM:  PHRASE WAS RECOGNIZED GO CONFIRM
4840 REM***************************************************************
4845 REM:  NO MATCH WITH ANY 'ECCOMMANDS' GROUP 1 OR GROUP 2 PHRASE
4850 REM
4855 FR = FR + 1                :REM:  NO MATCH OCCURRED SO ADD 1 TO FR SUM
4856 IF FR<3 THEN PRINT "...NO RECOGNITION, TRY AGAIN ...(";PEEK(151);")"
4857 IF FR<3 THEN FOR DE = 1 TO 900 : NEXT DE
4859 IF FR<3 THEN GOTO 4790     :REM:  LOOP BACK TWICE IF NO MATCH  (332)
4860 FP = 0                     :REM:  PASS  0 TO RETRAINGROUP
4861 LP = 15                    :REM:  PASS 15 TO RETRAINGROUP
4862 V$ = "ECCOMMANDS"          :REM:  PASS "ECCOMMANDS" TO RETRAINGROUP
4863 TT = ET                    :REM:  PASS ET TO RETRAINGROUP
4864 FOR PH = FP TO LP          :REM:  LOOP TO SET VO$ PHRASES = TO EC$
4865 VO$(PH) = EC$(PH)          :REM:  PASS EC$(PH) TO RETRAINGROUP
4866 NEXT PH                    :REM:  END FOR/NEXT LOOP
4867 GOSUB 6020                 :REM:  CALL RETRAINGROUP             (540)
4868 GOSUB 6140                 :REM:  CALL STASH-ECCOMMANDS         (550)
4869 IF E1 = 0 THEN EU = 1      :REM:  'ECCOMMANDS' UPDATED SO SET EU = 1
4870 GOTO 4775                  :REM:  LOOP BACK AFTER RETRAINING   (330)
4875 REM***************************************************************
4880 REM:  ONE OF THE EQUIPMENT PIECE PHRASES WAS RECOGNIZED
4885 REM
4890 REM   --335--
4900 GOSUB 6185                 :REM:  CALL FETCHCOMMANDER           (555)
4905 FR = 0                     :REM:  FR SUM SET BEFORE RECOGNITION TRIES
4910 PRINT"  OKAY YOU JUST CHOSE";
4915 PRINT EC$(MN)              :REM:  PRINT DEVICE NAME
4920 REM   --337--
4930 PRINT "  IS THIS THE DEVICE YOU WANTED?"
4935 GOSUB 6645                 :REM:  CALL MATCHCOMMANDERGROUP2    (575)
4950 DM = PEEK(151) - 7         :REM:  CONVERT MATCHED PHRASES 8-10 TO 1-2
4955 ON DM GOTO 5090, 5055, 5020
4960 REM***************************************************************
4965 REM:  NO MATCH WITH ANY 'COMMANDER' GROUP 2 PHRASE
4970 REM
4975 FR = FR + 1                :REM:  NO MATCH OCCURRED SO ADD 1 TO FR SUM
4977 IF FR<3 THEN PRINT "...NO RECOGNITION, TRY AGAIN...(";PEEK(151);")"
4978 IF FR<3 THEN FOR DE = 1 TO 900 : NEXT DE
4979 IF FR<3 THEN GOTO 4930     :REM:  LOOP BACK TWICE IF NO MATCH  (337)
4980 FP = 8                     :REM:  PASS  8 TO RETRAINGROUP
4981 LP = 10                    :REM:  PASS 10 TO RETRAINGROUP
4982 V$ = "COMMANDER"           :REM:  PASS "COMMANDER" TO RETRAINGROUP
4983 TT = CT                    :REM:  PASS CT TO RETRAINGROUP
4984 FOR PH = FP TO LP          :REM:  LOOP TO SET VO$ PHRASES = TO CR$
4985 VO$(PH) = CR$(PH)          :REM:  PASS CR$(PH) TO RETRAINGROUP
4986 NEXT PH                    :REM:  END FOR/NEXT LOOP
4987 GOSUB 6020                 :REM:  CALL RETRAINGROUP             (540)
4988 GOSUB 6100                 :REM:  CALL STASHCOMMANDER           (545)
4989 IF C1 = 0 THEN CU = 1      :REM:  'COMMANDER' UPDATED SO SET CU = 1
```

```
4990 GOTO 4905                  :REM:  LOOP BACK AFTER RETRAINING   (335)
4995 REM***************************************************************
5000 REM:  'STOP RECOGNITION' RECOGNIZED FROM 'COMMANDER' GROUP 2
5005 REM
5010 REM   --340--
5020 GOSUB 5575   :REM:  CALL STOPRECOGNITION                      (500)
5025 GOTO 4905    :REM:  LOOP BACK AFTER RECOGNITION IS RESTORED  (335)
5030 REM***************************************************************
5035 REM:  'NEGATIVE' MATCHED IN 'COMMANDER' GRP 2 SO GET DIFFERENT DEVICE
5040 REM
5045 REM   --342--
5055 GOSUB 6225     :REM:  CALL FETCH-ECCOMMANDS                   (560)
5060 GOTO 4775      :REM:  GO BACK TO ASK USER AGAIN FOR DEVICE    (330)
5065 REM***************************************************************
5070 REM:  'AFFIRM' RECOGNIZED IN 'COMMANDER' GRP 2 SO RIGHT DEVICE CHOSEN
5075 REM
5080 REM   --345--
5090 AC = D%(MN)          :REM:  SELECT APPLIANCE CODE FOR THIS DEVICE
5095 GOSUB 5755           :REM:  CALL RS232OUTPUT
5100 GOSUB 6225           :REM:  CALL FETCH-ECCOMMANDS              (560)
5105 REM   --347--
5115 FR = 0               :REM:  FR SUM SET BEFORE RECOGNITION TRIES
5120 REM   --350--
5130 PRINT "  'TURN ON' OR 'SHUT OFF' OR 'STOP RECOGNITION'?"
5135 POKE 49935, 3        :REM:  SET RECOGNITION 'ECCOMMANDS' GRP 3 (16-23)
5137 POKE 45351, 15       :REM:  LOWER MIN THRESHOLD TO 15
5140 GOSUB 5880           :REM:  CALL GET-VOICE-INPUT               (530)
5150 POKE 45351, 35       :REM:  RESET MIN THRESHOLD TO 35
5160 DM = PEEK(151)-15  :REM:  CONVERT RECOGNIZED PHRASES 16-18 TO 1-3
5165 ON DM GOTO 5305, 5265, 5230
5170 REM***************************************************************
5175 REM:  NO MATCH WITH ANY GROUP 3 PHRASE SO DO THE FOLLOWING
5180 REM
5185 FR = FR + 1              :REM:  NO MATCH OCCURRED SO ADD 1 TO FR SUM
5187 IF FR<3 THEN PRINT "...NO RECOGNITION, TRY AGAIN...(";PEEK(151);")"
5188 IF FR<3 THEN FOR DE = 1 TO 900 : NEXT DE
5190 IF FR<3 THEN GOTO 5130 :REM:  LOOP BACK TWICE IF NO MATCH    (350)
5192 FP = 16                  :REM:  PASS 16 TO RETRAINGROUP
5193 LP = 18                  :REM:  PASS 18 TO RETRAINGROUP
5194 V$ = "COMMANDER"         :REM:  PASS "COMMANDER" TO RETRAINGROUP
5195 TT = ET                  :REM:  PASS ET TO RETRAINGROUP
5196 FOR PH = FP TO LP        :REM:  LOOP TO SET VO$ PHRASES = TO EC$
5197 VO$(PH) = EC$(PH)        :REM:  PASS EC$(PH) TO RETRAINGROUP
5198 NEXT PH                  :REM:  END FOR/NEXT LOOP
5199 GOSUB 6020               :REM:  CALL RETRAINGROUP              (540)
5200 GOSUB 6140               :REM:  CALL STASHCOMMANDER            (545)
5201 IF E1 = 0 THEN EU = 1   :REM:  'COMMANDER' UPDATED SO SET EU = 1
5202 GOTO 5115                :REM:  LOOP BACK AFTER RETRAINING     (337)
5205 REM***************************************************************
5210 REM:  'STOP RECOGNITION' RECOGNIZED IN GRP 3 SO CALL THAT SUBROUTINE
5215 REM
5220 REM   --352--
5230 GOSUB 6185      :REM:  CALL FETCHCOMMANDER                      (555)
5232 GOSUB 5575      :REM:  CALL STOPRECOGNITION                     (500)
5233 GOSUB 6225      :REM:  CALL FETCH-ECCOMMANDS                    (560)
```

```
5235 GOTO 5115       :REM:  LOOP BACK AFTER RECOGNITION RESTORED   (347)
5240 REM***************************************************************
5245 REM: 'TURN ON' RECOGNIZED IN GRP 3 SO SEND PROPER CODE TO CONTROLLER
5290 REM
5295 REM   --357--
5305 AC = 197                :REM:  SET APPLIANCE CODE = 'TURN ON'
5310 GOSUB 5755              :REM:  CALL RS232OUTPUT                (515)
5315 REM***************************************************************
5320 REM:  MAIN USER MENU FOR ENVIRONMENTAL CONTROLLER
5325 REM
5330 REM   --360--
5340 FR = 0
5345 REM   --362--
5355 PRINT "   DO YOU WANT TO:"
5360 PRINT "                 1)CONTINUE WITH CONTROL ?"
5365 PRINT "                    (SAY 'CONTINUE CONTROL')"
5370 PRINT "                 2)JUMP BACK TO MAIN MENU ?"
5375 PRINT "                    (SAY 'GOTO MAIN MENU')"
5380 PRINT "                 3)EXIT THE PROGRAM ?"
5385 PRINT "                    (SAY 'EXIT THE PROGRAM')"
5390 PRINT "                 4)RETRAIN 'ECCOMMANDS' ?"
5395 PRINT "                    (SAY 'RETRAIN E/C')"
5400 PRINT "                 5)STOP RECOGNITION ?"
5405 PRINT "                    (SAY 'STOP RECOGNITION')"
5410 POKE 49985, 4     :REM:  SET RECOGNITION 'ECCOMMANDS' GRP 4 (24-31)
5415 GOSUB 5880              :REM:  CALL GET-VOICE-INPUT            (530)
5420 DM = PEEK (151) - 23  :REM:  CONVERT PHRASE #'S 24-28 TO 1-5
5425 ON DM GOTO 4775, 5525, 5464, 5490
5430 REM***************************************************************
5435 REM:  NO MATCH WITH ANY GROUP 4 PHRASE SO DO THE FOLLOWING
5440 REM
5445 FR = FR + 1             :REM:  NO MATCH OCCURRED SO ADD 1 TO FR SUM
5447 IF FR<3 THEN PRINT "...NO RECOGNITION, TRY AGAIN...(";PEEK(151);")"
5448 IF FR<3 THEN FOR DE = 1 TO 900 : NEXT DE
5450 IF FR<3 THEN GOTO 5355:REM:  LOOP BACK TWICE IF NO MATCH     (362)
5452 FP = 24                 :REM:  PASS 24 TO RETRAINGROUP
5453 LP = 28                 :REM:  PASS 28 TO RETRAINGROUP
5454 V$ = "ECCOMMANDS"       :REM:  PASS "ECCOMMANDS" TO RETRAINGROUP
5455 TT = ET                 :REM:  PASS ET TO RETRAINGROUP
5456 FOR PH = FP TO LP       :REM:  LOOP TO SET VO$ PHRASES = TO EC$
5457 VO$(PH) = EC$(PH)       :REM:  PASS EC$(PH) TO RETRAINGROUP
5458 NEXT PH                 :REM:  END FOR/NEXT LOOP
5459 GOSUB 6020              :REM:  CALL RETRAINGROUP               (540)
5460 GOSUB 6140              :REM:  CALL STASHCOMMANDER             (550)
5461 IF E1 = 0 THEN EU = 1 :REM:  'ECCOMMANDS' UPDATED SO SET EU = 1
5463 GOTO 5115               :REM:  LOOP BACK AFTER RETRAINING      (360)
5464 REM***************************************************************
5465 REM:  'RETRAIN E/C' RECOGNIZED FROM GROUP 4 SO LOOP BACK TO RETRAIN
5466 REM
5467 GOSUB 6185              :REM:  CALL FETCHCOMMANDER             (555)
5468 GOTO 4255               :REM:  LOOP BACK TO RETRAIN            (305)
5469 REM***************************************************************
5470 REM:  'STOP RECOGNITION' RECOGNIZED IN GRP 4 SO CALL THAT SUBROUTINE
5475 REM
5480 REM   --365--
```

```
5490 GOSUB 6185        :REM:  CALL FETCHCOMMANDER                          (555)
5492 GOSUB 5575        :REM:  CALL STOPRECOGNITION                         (500)
5494 GOSUB 6225        :REM:  CALL FETCH-ECCOMMANDS                        (560)
5495 GOTO 5340         :REM:  LOOP BACK AFTER RECOGNITION RESTORED  (360)
5500 REM****************************************************************
5505 REM: 'GOTO MAIN MENU' OR 'EXIT THE PROGRAM' RECOGNIZED FROM GROUP 4
5510 REM
5515 REM      --367--
5525 GOSUB 6185                  :REM:  CALL FETCHCOMMANDER            (555)
5530 IF DM = 2 THEN GOTO 2560 :REM:  'GOTO MAIN MENU' SO GO TO IT
5535 GOTO 2795              :REM:  'EXIT THE PROGRAM' SO END 'VOICEMANAGER'
5540 REM****************************************************************
5545 REM  SUBROUTINES:  THE FOLLOWING ARE SUBROUTINES TO 'VOICEMANAGER'
5550 REM ---------------------------------------------------------------
5555 REM*****--500--***************************************************
5560 REM  SUBROUTINE STOPRECOGNITION
5565 REM
5575 POKE 45353, 44      :REM:  SET MAX RECOGNITION THRESHOLD LOW BYTE
5580 POKE 49935, 4       :REM:  SET RECOGNITION 'COMMANDER' GRP 4 (24-31)
5590 PRINT "  SAY 'RESTORE SPEECH' WHEN READY...."
5591 GOSUB 5880          :REM:  CALL GET-VOICE-INPUT                    (530)
5592 REM ---------------------------------------------------------------
5593 REM - LOOP UNTIL 'RESTORE SPEECH' RECOGNIZED
5595 REM - PEEK(151) EQUALS 24 IS 'RESTORE SPEECH'
5600 REM ---------------------------------------------------------------
5602 IF PEEK(151) <> 24 THEN PRINT "...NO RESTORE RECOG YET...(";
5603 IF PEEK(151) <> 24 THEN PRINT PEEK(151);")"
5604 IF PEEK(151) <> 24 THEN FOR DE = 1 TO 800 : NEXT DE
5605 IF PEEK(151) <> 24 THEN GOTO 5590
5607 POKE 45353, 204     :REM:  RESET MAX RECOGNITION THRESHOLD LOW BYTE
5610 RETURN
5615 REM*****--505--***************************************************
5620 REM SUBROUTINE ENOUGH-TRAINING-DONE
5625 REM
5635 PRINT " ...NO NEED FOR MORE TRAINS, LET'S MOVE ON"
5640 FOR DE = 1 TO 2000
5645 NEXT DE
5650 RETURN
5655 REM*****--510--***************************************************
5660 REM SUBROUTINE STASHORFETCHVOCABULARY
5665 REM
5675 POKE BA + 2,0   :REM:  CPU ADDRESS LSB (BA SET TO $DF00 [57088])
5680 POKE BA + 3,177 :REM:  CPU ADDRESS MSB
5685 POKE BA + 4,0   :REM:  REU ADDRESS LSB
5690 POKE BA + 5,RA  :REM:  REU ADDRESS MSB
5691 REM  177=COMMANDER  193=ECCOMMANDS  209=ALPHABET
5692 REM  225=PUNCTUATION   241=NUMBERSPLUS
5695 POKE BA + 6,8   :REM:  REU BANK #
5700 POKE BA + 7,44  :REM:  # BYTES VOCAB TEMPLATE AREA TO TRANSFER(LSB)
5705 POKE BA + 8,12  :REM:  # BYTES VOCAB TEMPLATE AREA TO TRANSFER(MSB)
5710 POKE BA + 9,0   :REM:  RESET INTERRUPT CONTROL BITS
5715 POKE BA + 10,0  :REM:  INCREMENT BOTH SOURCE, DESTINATION ADDRESSES
5720 POKE 49632, OM  :REM:  OPERATING MODE (220 = STASH  221 = FETCH)
5725 SYS 49633       :REM:  CALL ACSTASHORFETCH (SWAP RAM-BASIC, EXECUTE)
5730 RETURN
```

```
5735 REM*****--515--*****************************************************
5740 REM  SUBROUTINE RS232OUTPUT
5745 REM
5755 RS$ = CHR$(AC)
5760 PRINT#2,RS$;
5765 RETURN
5770 REM*****--520--****************************************************
5775 REM   SUBROUTINE SAVERS232BUFFER
5780 REM
5782 REM:  DATA DICTIONARY
5783 REM-----------------------------------------------------------------
5784 REM * S1 = TEMP STORAGE VARIABLE FOR RS232  INPUT BUFFER LOC $F7 *
5785 REM * S2 = TEMP STORAGE VARIABLE FOR RS232  INPUT BUFFER LOC $F8 *
5786 REM * S3 = TEMP STORAGE VARIABLE FOR RS232 OUTPUT BUFFER LOC $F9 *
5787 REM * S4 = TEMP STORAGE VARIABLE FOR RS232 OUTPUT BUFFER LOC $FA *
5788 REM-----------------------------------------------------------------
5789 REM
5790 S1 = PEEK(247)
5795 S2 = PEEK(248)
5800 S3 = PEEK(249)
5805 S4 = PEEK(250)
5810 RETURN
5815 REM*****--525--****************************************************
5820 REM   SUBROUTINE RESTORERS232BUFFER
5825 REM
5827 REM:   DATA DICTIONARY
5828 REM-----------------------------------------------------------------
5829 REM * S1 = TEMP STORAGE VARIABLE FOR RS232  INPUT BUFFER LOC $F7 *
5830 REM * S2 = TEMP STORAGE VARIABLE FOR RS232  INPUT BUFFER LOC $F8 *
5831 REM * S3 = TEMP STORAGE VARIABLE FOR RS232 OUTPUT BUFFER LOC $F9 *
5832 REM * S4 = TEMP STORAGE VARIABLE FOR RS232 OUTPUT BUFFER LOC $FA *
5833 REM-----------------------------------------------------------------
5834 REM
5835 POKE 247, S1
5840 POKE 248, S2
5845 POKE 249, S3
5850 POKE 250, S4
5855 RETURN
5860 REM*****--530--*****************************************************
5865 REM   SUBROUTINE GET-VOICE-INPUT
5870 REM
5880 IF G2 = 0 THEN POKE 49936,255 :REM:  END SELECTION OF GRPS SCANNED
5885 IF G2 = 1 THEN POKE 49936,2   :REM:  SET RECOG FOR GRP 2 (8-15) ALSO
5890 IF G2 = 1 THEN POKE 49937,255 :REM:  END SELECTION OF GRPS SCANNED
5895 GOSUB 5790                    :REM:  CALL SAVERS232BUFFER
5900 SYS 49923                     :REM:  CALL RECOG IN VOICE/RECOG
5905 GOSUB 5835                    :REM:  CALL RESTORERS232BUFFER
5910 RETURN
5915 REM*****--530--*****************************************************
5920 REM   SUBROUTINE TRAIN
5925 REM
5926 REM:  DATA DICTIONARY
5927 REM ----------------------------------------------------------------
5928 REM * VO$=ARRAY TO WHICH ARRAY OF PHRASES BEING TRAINED IS PASSED *
5929 REM ----------------------------------------------------------------
```

```
5930 REM
5935 PRINT "SAY"; VO$(PH); :REM:  PROMPT USER TO SAY THE PHRASE
5940 POKE 151, PH          :REM:  PUT PHRASE # TO TRAIN IN LOCATION 151
5945 GOSUB 5790            :REM:  CALL SAVERS232BUFFER
5950 SYS 49920             :REM:  CALL 'TRAIN' SUBROUTINE IN VOICE/RECOG
5955 GOSUB 5835            :REM:  CALL RESTORERS232BUFFER
5960 PRINT "...TRAINED"    :REM:  PRINT WORD INDICATING PHRASE IS TRAINED
5962 FOR DE = 1 TO 400     :REM:  DELAY MOMENTARILY FOR DISPLAY
5963 NEXT DE               :REM:  END FOR/NEXT FOR DELAY
5965 RETURN
5970 REM*****--530--*****************************************************
5975 REM   SUBROUTINE RETRAINGROUP
5977 REM
5978 REM:  DATA DICTIONARY
5980 REM ---------------------------------------------------------------
5985 REM  *  FP = FIRST PHRASE # FOR THE VOCABULARY GROUP BEING TRAINED *
5990 REM  *  LP =  LAST PHRASE # FOR THE VOCABULARY GROUP BEING TRAINED *
5995 REM  *  RT = INDEX FOR FOR/NEXT LOOP, MEANS 'RETRAINS'             *
5997 REM  *  TT = TIMES TRAINED, # TIMES THIS GROUP PREVIOUSLY TRAINED  *
6000 REM  *  V$ = NAME OF VOCABULARY TO WHICH RETRAINED GROUP BELONGS   *
6005 REM ---------------------------------------------------------------
6010 REM
6020 PRINT "   YOU HAVE HAD 3 NO RECOGNITION"
6025 PRINT "   RESULTS IN SUCCESSION FOR THIS GROUP"
6030 PRINT "   OF PHRASES IN THE VOCABULARY '";
6032 PRINT V$; "'."
6035 PRINT "      LET'S RETRAIN THOSE PHRASES. . . "
6037 FOR DE = 1 TO 2000          :REM:  DELAY TO SEE
6038 NEXT DE                     :REM:  USER PROMPT
6040 FOR RT = 1 TO TT            :REM:  RETRAIN TT # TIMES
6045 FOR PH = FP TO LP           :REM:  START LOOP TO RETRAIN PHRASES
6050 IF RT = 1 THEN POKE 782,PH :REM:  PHRASE # TO BLANK IN TEMPLATE AREA
6055 IF RT = 1 THEN SYS 49932   :REM:  CALL 'BLANK' SR IN VOICE/RECOG
6060 GOSUB 5935                  :REM:  CALL TRAIN   (535)
6065 NEXT PH                     :REM:  END FOR/NEXT PH INNER LOOP
6070 NEXT RT                     :REM:  END FOR/NEXT RT OUTER LOOP
6075 RETURN
6080 REM*****--545--*****************************************************
6085 REM   SUBROUTINE STASHCOMMANDER
6090 REM
6100 RA = 177              :REM:  SET UP REU ADDRESS MSB FOR 'COMMANDER'
6105 OM = 220              :REM:  SET UP OPERATING MODE FOR STASH
6110 GOSUB 5675            :REM:  CALL STASHORFETCHVOCABULARY       (510)
6115 RETURN
6120 REM*****--550--*****************************************************
6125 REM   SUBROUTINE STASH-ECCOMMANDS
6130 REM
6140 RA = 193              :REM:  SET UP REU ADDRESS MSB FOR 'ECCOMMANDS'
6145 OM = 220              :REM:  SET UP OPERATING MODE FOR STASH
6150 GOSUB 5675            :REM:  CALL STASHORFETCHVOCABULARY       (510)
6160 RETURN
6165 REM*****--555--*****************************************************
6170 REM   SUBROUTINE FETCHCOMMANDER
6175 REM
6185 RA = 177              :REM:  SET UP REU ADDRESS MSB FOR 'COMMANDER'
```

```
6190 OM = 221              :REM:  SET UP OPERATING MODE FOR FETCH
6195 GOSUB 5675            :REM:  CALL STASHORFETCHVOCABULARY       (510)
6200 RETURN
6205 REM*****--560--**************************************************
6210 REM   SUBROUTINE FETCH-ECCOMMANDS
6215 REM
6225 RA = 193              :REM:  SET UP REU ADDRESS MSB FOR 'ECCOMMANDS'
6230 OM = 221              :REM:  SET UP OPERATING MODE FOR FETCH
6235 GOSUB 5675            :REM:  CALL STASHORFETCHVOCABULARY       (510)
6236 RETURN
6249 REM*****--565--**************************************************
6250 REM   SUBROUTINE MATCHCOMMANDERGROUP3
6255 REM
6260 REM:       PURPOSE:    TO TAKE VOICE INPUT FOR PHRASE CHOICES IN
6265 REM        -------     'COMMANDER' GROUP 3, LOOK FOR A MATCH, AND
6270 REM                    TAKE THE APPROPRIATE ACTION
6290 REM
6325 REM:       BEGIN MATCHCOMMANDERGROUP3
6385 FR = 0              :REM:  FR SUM SET = 0 BEFORE RECOGNITION TRIES
6390 PRINT  "SAY 'BEGIN TRAINING' WHEN READY OR"
6392 PRINT  "    'STOP RECOGNITION' TO PAUSE"
6400 POKE 49935, 3       :REM:  SET RECOG 'COMMANDER' GRP 3 (16-23) ONLY
6405 GOSUB 5880          :REM:  CALL GET-VOICE-INPUT
6415 DM = PEEK(151)-15 :REM:  PHRASE #'S 16,17 CONVERTED TO 1,2 FOR 'ON'
6445 ON DM GOTO 6550, 6525
6450 REM**************************************************************
6455 REM:  NO MATCH HAS OCCURRED SO DO THE FOLLOWING
6460 REM
6465 FR = FR + 1               :REM:  NO MATCH OCCURRED SO ADD 1 TO FR SUM
6467 IF FR<3 THEN PRINT "...NO RECOGNITION, TRY AGAIN...(";PEEK(151);")"
6468 IF FR<3 THEN FOR DE = 1 TO 900 : NEXT DE
6470 IF FR<3 THEN GOTO 6390  :REM:  LOOP BACK TWICE IF NO MATCH
6475 FP = 16                   :REM:  PASS 16 TO RETRAINGROUP
6480 LP = 17                   :REM:  PASS 17 TO RETRAINGROUP
6485 V$ = "COMMANDER"          :REM:  PASS "COMMANDER" TO RETRAINGROUP
6486 TT = CT                   :REM:  PASS CT TO RETRAINGROUP
6487 FOR PH = FP TO LP         :REM:  LOOP TO SET VO$ PHRASES = TO CR$
6488 VO$(PH) = CR$(PH)         :REM:  PASS CR$(PH) TO RETRAINGROUP
6489 NEXT PH                   :REM:  END FOR/NEXT LOOP
6490 GOSUB 6020                :REM:  CALL RETRAINGROUP              (540)
6495 GOSUB 6100                :REM:  CALL STASHCOMMANDER            (545)
6500 IF C1 = 0 THEN CU = 1    :REM:  'COMMANDER' JUST UPDATED SET CU = 1
6505 GOTO 6385                 :REM:  LOOP BACK AFTER RETRAINING
6510 REM**************************************************************
6515 REM:  'STOP RECOGNITION' RECOGNIZED GRP 3 SO CALL THAT SUBROUTINE
6520 REM
6525 GOSUB 5575                :REM:  CALL STOPRECOGNITION
6530 GOTO 6385                 :REM:  LOOP BACK AFTER RECOGNITION RESTORED
6535 REM**************************************************************
6540 REM:  'BEGIN TRAINING' RECOGNIZED FROM GRP 3 SO RETURN TO TRAIN
6545 REM
6550 RETURN
6555 REM:       END MATCHCOMMANDERGROUP3
6560 REM:       ----------------------
6565 REM*****--570--**************************************************
```

```
6570 REM   SUBROUTINE ASKUSER-ABOUT-MORETRAINS
6575 REM
6585 PRINT "   WOULD YOU LIKE TO DO ANOTHER TRAINING?"
6590 PRINT "   THIS TRAINING WOULD ACCUMULATE"
6595 PRINT "   WITH PREVIOUS TRAININGS TO FORM"
6600 PRINT "   THE VOICEPRINT, NOT ERASE THEM TO"
6605 PRINT "   FORM A COMPLETELY NEW VOICEPRINT."
6610 PRINT "   (TWO TRAININGS RECOMMENDED BUT"
6615 PRINT "    NOT TO EXCEED FIVE TRAININGS)"
6620 RETURN
6625 REM*****--575--****************************************************
6630 REM   SUBROUTINE MATCHCOMMANDERGROUP2
6635 REM
6645 PRINT  "   SAY 'AFFIRM', 'NEGATIVE' OR"
6650 PRINT  "        'STOP RECOGNITION'"
6655 POKE 49935, 2   :REM:  SET RECOG FOR 'COMMANDER' GRP 2 (8-15) ONLY
6660 GOSUB 5880      :REM:  CALL GET-VOICE-INPUT                  (530)
6665 RETURN
6670 REM*****--580--****************************************************
6675 REM   SUBROUTINE BLANKTEMPLATEAREA
6680 REM
6690 POKE 782, 255         :REM:  SET UP FOR BLANK OF WHOLE TEMPLATE AREA
6695 SYS 49932             :REM:  CALL 'BLANK' IN VOICE/RECOG
6700 RETURN
6701 REM****************************************************************
6702 REM   SUBROUTINE STASHALPHABET
6703 REM
6704 RA = 209              :REM:  SET UP REU ADDRESS MSB FOR 'ALPHABET'
6705 OM = 220              :REM:  SET UP OPERATING MODE FOR STASH
6706 GOSUB 5675            :REM:  CALL STASHORFETCHVOCABULARY      (510)
6707 RETURN
6708 REM*****--590--****************************************************
6709 REM   SUBROUTINE STASHPUNCTUATION
6710 REM
6711 RA = 225              :REM:  SET UP REU ADDRESS MSB FOR 'PUNCTUATION'
6712 OM = 220              :REM:  SET UP OPERATING MODE FOR STASH
6713 GOSUB 5675            :REM:  CALL STASHORFETCHVOCABULARY      (510)
6714 RETURN
6715 REM*****--595--****************************************************
6716 REM   SUBROUTINE STASHNUMBERSPLUS
6717 REM
6718 RA = 241              :REM:  SET UP REU ADDRESS MSB FOR 'NUMBERSPLUS'
6719 OM = 220              :REM:  SET UP OPERATING MODE FOR STASH
6720 GOSUB 5675            :REM:  CALL STASHORFETCHVOCABULARY      (510)
6721 RETURN
6722 REM*****--600--****************************************************
6723 REM   SUBROUTINE FETCHALPHABET
6724 REM
6725 RA = 209              :REM:  SET UP REU ADDRESS MSB FOR 'ALPHABET'
6726 OM = 221              :REM:  SET UP OPERATING MODE FOR STASH
6727 GOSUB 5675            :REM:  CALL STASHORFETCHVOCABULARY      (510)
6728 RETURN
6729 REM*****--605--****************************************************
6730 REM   SUBROUTINE FETCHPUNCTUATION
6731 REM
```

```
6732 RA = 225               :REM:  SET UP REU ADDRESS MSB FOR 'PUNCTUATION'
6733 OM = 221               :REM:  SET UP OPERATING MODE FOR STASH
6734 GOSUB 5675             :REM:  CALL STASHORFETCHVOCABULARY          (510)
6735 RETURN
6736 REM*****--610--*****************************************************
6737 REM   SUBROUTINE FETCHNUMBERSPLUS
6738 REM
6739 RA = 241               :REM:  SET UP REU ADDRESS MSB FOR 'NUMBERSPLUS'
6740 OM = 221               :REM:  SET UP OPERATING MODE FOR STASH
6741 GOSUB 5675             :REM:  CALL STASHORFETCHVOCABULARY          (510)
6742 RETURN
6760 REM*****************************************************************
6762 REM   END OF SUBROUTINES
6763 REM   ------------------
6764 REM*****************************************************************
6765 REM
6766 REM*****************************************************************
6767 REM   DATA:  FOR READ STATEMENTS IN PROGRAM TO SET UP VOCABULARIES
6768 REM   ----
6769 REM*****************************************************************
6670 REM
6771 DATA      "'APPLICATION PROGRAM'     "
6772 DATA      "'AFFIRM'                  "
6773 DATA      "'BEGIN TRAINING'          "
6774 DATA      "'RESTORE SPEECH'          "
6775 DATA      "'ENVIRONMENTAL CONTROLLER'"
6780 DATA      "'NEGATIVE'                "
6785 DATA      "'STOP RECOGNITION'        "
6790 DATA      "'PROGRAM ESCAPE'          "
6795 DATA      "'STOP RECOGNITION'        "
6800 DATA      "'RETRAIN COMMANDER'       "
6805 DATA      "'STOP RECOGNITION'        "
6810 DATA      "'LAMP-ONE'                ", 204
6815 DATA      "'LAMP-THREE'              ", 206
6820 DATA      "'TURN ON'                 "
6825 DATA      "'CONTINUE CONTROL'        "
6830 DATA      "'RADIO-ONE'               ", 220
6835 DATA      "'PORCHLIGHT'              ", 222
6840 DATA      "'SHUT OFF'                "
6845 DATA      "'GOTO MAIN MENU'          "
6850 DATA      "'TELEVISION'              ", 196
6855 DATA      "'LAMP-FOUR'               ", 198
6860 DATA      "'STOP RECOGNITION'        "
6865 DATA      "'EXIT THE PROGRAM'        "
6870 DATA      "'STEREO'                  ", 212
6875 DATA      "'PATIOLIGHT'              ", 214
6880 DATA      "'RETRAIN E/C'             "
6885 DATA      "'MICROWAVE'               ", 194
6890 DATA      "'TOASTER'                 ", 192
6895 DATA   "'STOP RECOGNITION'           "
6900 DATA   "'FAN"                        ", 210
6905 DATA   "'BLENDER'                    ", 208
6910 DATA   "'LAMP-TWO'                   ", 202
6915 DATA   "'GARAGELIGHT'                ", 200
6920 DATA   "'RADIO-TWO'                  ", 218
```

```
6925 DATA   "'BATTERYCHARGER'                ", 216
6930 DATA   "'ALPHA A'                       "
6935 DATA   "'BRAVO B'                       "
6940 DATA   "'CHARLIE C'                     "
6945 DATA   "'DELTA D'                       "
6950 DATA   "'ECHO E'                        "
6955 DATA   "'FOXTROT F'                     "
6960 DATA   "'GURU G'                        "
6965 DATA   "'HOTEL H'                       "
6970 DATA   "'INDIA I'                       "
6975 DATA   "'JULIET J'                      "
6980 DATA   "'KILO K'                        "
6985 DATA   "'LIMA L'                        "
6990 DATA   "'MICHAEL M'                     "
6995 DATA   "'NOVEMBER N'                    "
7000 DATA   "'OSCAR O'                       "
7005 DATA   "'PAPA P'                        "
7010 DATA   "'QUEBEC Q'                      "
7015 DATA   "'ROMEO R'                       "
7020 DATA   "'SIERRA S'                      "
7025 DATA   "'TANGO T'                       "
7030 DATA   "'UNIFORM U'                     "
7035 DATA   "'VICTOR V'                      "
7040 DATA   "'WHISKEY W'                     "
7045 DATA   "'X-RAY X'                       "
7050 DATA   "'YANKEE Y'                      "
7055 DATA   "'ZULU Z'                        "
7060 DATA   "'CLEAR SCREEN'                  "
7065 DATA   "'CAPITALS'                      "
7070 DATA   "'LOWERCASE'                     "
7075 DATA   "'SPACE BAR'                     "
7080 DATA   "'SWITCH PUNCTUATION'            "
7085 DATA   "'SWITCH NUMBERSPLUS'            "
7090 DATA   "'EXCLAMATION POINT'             "
7095 DATA   "'QUOTATION MARKS'               "
7100 DATA   "'NUMBER SIGN'                   "
7105 DATA   "'DOLLAR SIGN'                   "
7110 DATA   "'PERCENTAGE'                    "
7115 DATA   "'AMPERSAND'                     "
7120 DATA   "'APOSTROPHE'                    "
7125 DATA   "'OPEN PAREN'                    "
7130 DATA   "'CLOSE PAREN'                   "
7135 DATA   "'ASTERISK'                      "
7140 DATA   "'POSITIVE SIGN'                 "
7145 DATA   "'COMMA'                         "
7150 DATA   "'MINUS SIGN'                    "
7155 DATA   "'PERIOD'                        "
7160 DATA   "'SLASH MARK'                    "
7165 DATA   "'COLON'                         "
7170 DATA   "'SEMICOLON'                     "
7175 DATA   "'LESSER THAN'                   "
7180 DATA   "'EQUAL TO'                      "
7185 DATA   "'GREATER THAN'                  "
7190 DATA   "'QUESTION MARK'                 "
7195 DATA   "'CIRCLE A'                      "
```

```
7200 DATA   "'LEFT BRACKET'                  "
7205 DATA   "'BRITISH POUND'                 "
7210 DATA   "'RIGHT BRACKET'                 "
7215 DATA   "'UP ARROW'                      "
7220 DATA   "'LEFT ARROW'                    "
7225 DATA   "'CAPITALS'                      "
7230 DATA   "'LOWERCASE'                     "
7235 DATA   "'SPACE BAR'                     "
7240 DATA   "'SWITCH ALPHABET'               "
7245 DATA   "'SWITCH NUMBERSPLUS'            "
7250 DATA   "'ZERO Z'                        "
7255 DATA   "'ONE O'                         "
7260 DATA   "'TWO T'                         "
7265 DATA   "'THREE TH'                      "
7270 DATA   "'FOUR R'                        "
7275 DATA   "'FIVE F'                        "
7280 DATA   "'SIX X'                         "
7285 DATA   "'SEVEN V'                       "
7290 DATA   "'EIGHT E'                       "
7295 DATA   "'NINE N'                        "
7300 DATA   "'ONE O FUNCTION'                "
7305 DATA   "'TWO T FUNCTION'                "
7310 DATA   "'THREE TH FUNCTION'             "
7315 DATA   "'FOUR R FUNCTION'               "
7320 DATA   "'FIVE F FUNCTION'               "
7325 DATA   "'SIX X FUNCTION'                "
7330 DATA   "'SEVEN V FUNCTION'              "
7335 DATA   "'EIGHT E FUNCTION'              "
7340 DATA   "'CURSOR UP'                     "
7345 DATA   "'INSERT'                        "
7350 DATA   "'CURSOR LEFT'                   "
7355 DATA   "'RETURN'                        "
7360 DATA   "'CURSOR DOWN'                   "
7365 DATA   "'GO HOME'                       "
7370 DATA   "'DELETE A STROKE'               "
7375 DATA   "'CURSOR RIGHT'                  "
7380 DATA   "'QUIT APPLICATION'              "
7385 DATA   "'CAPITALS'                      "
7390 DATA   "'LOWERCASE'                     "
7395 DATA   "'SPACE BAR'                     "
7400 DATA   "'SWITCH ALPHABET'               "
7405 DATA   "'SWITCH PUNCTUATION'            "
7410 REM
7415 REM************************************************************
7420 REM    END OF DATA
7425 REM    -----------
7430 REM************************************************************
```

ACSTASHORFETCH

```
0C1E0  00        BRK
0C1E1  78        SEI
0C1E2  20 4C CB  JSR $CB4C
0C1E5  AD E0 C1  LDA $C1E0
0C1E8  8D 01 DF  STA $DF01
0C1EB  20 45 CB  JSR $CB45
0C1EE  58        CLI
0C1EF  60        RTS
```