further choice, the bridge with lowest locants will remain uncombined.

(b) The expanded base cycle may have as ring units only monocyclic rings, or where they have trivial names allowed by Organic Rules A-21.3, A-23.1, B-2, or D-Table IV in footnote 1, polycyclic rings are permitted.

(c) When a bridge is combined with another part of the structure, the ring with the least number of atoms will be the choice as a unit in the expanded base cycle unless the larger unit is part of a polycyclic ring having an allowed trivial name. In a choice between the rings of the same size, the more senior ring will be taken as part of the expanded base cycle.

(d) Bridges will be combined with other parts of the structure in the following order: (1) bridges between atoms in the base cycle; (2) bridges between atoms within one bridge; (3) bridges between atoms in the base cycle and a ring unit of the expanded base cycle; (4) bridges between atoms in different ring units. The minimum number of bridges remaining after the above combinations have been carried out will be treated in the same way as substituents in forming the name of the macrocycle. A zero-atom bridge will be cited in the form $O^{x,y}$, where x and y are locants for the bridge.

In the examples of cyclic structures containing bridges shown in Chart II, the base cycle is shown in heavy lines; numbering is that of the system after bridge combinations have been made.

ACKNOWLEDGMENT

I express my appreciation to Joy E. Merritt for her many valuable criticisms and suggestions; she contributed in a major way to the improvement of the nomenclature system proposed in this paper.

REFERENCES AND NOTES

(1) International Union of Pure and Applied Chemistry "Nomenclature of Organic Chemistry"; Pergamon Press: Oxford, 1979; Sections A–F and H.
(2) "Nomenclature of Regular Single-Strand Organic Polymers". *Pure Appl. Chem.* **1976**, *48*, 373–385.

Voice-Operated Microcomputer-Based Laboratory Data Acquisition System To Aid Handicapped Students in Chemistry Laboratories

ROBERT C. MORRISON,* DAVID LUNNEY, RONALD J. TERRY, JOHN HASSELL,† and GARY BOSWOOD

Department of Chemistry, East Carolina University, Greenville, North Carolina 27834

Received April 24, 1984

A voice-entry microcomputer system is described that consists of a Voice-Operated Isolated Command Entry (VOICE) terminal connected to a previously developed talking microcomputer-based data acquisition system. The system has been used to carry out a titration experiment with voice commands, and it can be used to aid a student in performing a number of undergraduate chemistry experiments as well as function as a voice-entry scientific calculator. This system will enable students with upper limb disabilities to participate in chemistry laboratory experiments that use standard chemical instruments.

INTRODUCTION

Students with disabilities of the upper limbs face high occupational barriers if they plan to enter technical fields where manipulative skills are required. We have attempted to lower some of the barriers by applying microcomputer technology to the problems of giving them access to genuinely instructive laboratory experiences in chemistry. Since chemistry courses are required as cognates for degrees in all the natural sciences, in every branch of engineering, and in most of the health professions, lowering the barriers that keep persons with upper limb disabilities from taking chemistry courses will improve their access to many other fields. Because of the great and ever-increasing importance of instrumental measurements in chemistry and in other fields in which chemistry is a useful tool, we have chosen to concentrate on the objective of facilitating the use of chemical instruments by students with upper limb disabilities in order to increase their independence in chemistry laboratories. Microcomputers and voice-recognition technology provide effective means of realizing this objective. Using commercially available technology, we have developed a voice-command recognition system that will enable persons with upper limb disabilities to acquire laboratory data and control chemistry experiments by giving voice commands to an existing laboratory microcomputer.

We have previously described a microcomputer-based Universal Laboratory Training and Research Aid (ULTRA) that we developed to aid blind students in undergraduate chemistry laboratories.[1,2] This system consists of a portable microcomputer equipped with voice output, keyboard input, and a variety of analog and digital inputs and outputs. It comes with an extensive software package designed to perform most of the instrumental measurements now done in freshman and sophomore chemistry laboratories.

The ULTRA system can be interfaced with a wide variety of instruments and sensors and is programmed to assist the student during experiments and during subsequent data analysis tasks. The ULTRA software includes a program that functions as a talking scientific calculator so handicapped students can perform calculations on the data they collect during a laboratory experiment. The blind student controls the computer and enters data through a standard keyboard; the computer's principal means of communication with blind students is spoken words generated by a speech synthesizer. The ULTRA system is also capable of functioning as a talking terminal to a large central computer, so students can use it to obtain access to other educational materials.

In this paper, we discuss extensions we have made to the ULTRA system by adapting it to accept voice commands, and we discuss a voice-operated titration we carried out by using

† Permanent address: Battelle Columbus Laboratories, Columbus, OH 43201.

this system. We can use the voice-controlled system to acquire data from chemical instruments such as electronic balances, pH electrodes, temperature sensors, titrators, gas and liquid chromatographs, spectrophotometers, etc. In addition, this system can function as a voice-controlled scientific calculator so that a student can perform calculations on the data that are collected.

The recognition of continuous speech was not possible with voice-recognition systems that were commercially available when we started this project; they were only capable of recognizing isolated words or commands separated by about 0.1 s. (Recently, Texas Instruments has introduced a product that will recognize connected words.[3]) Using the available technology, we developed a Voice-Operated Isolated Command Entry (VOICE) terminal so students with upper limb disabilities can use the ULTRA system to perform limited experiments in undergraduate chemistry laboratories by using voice-entry commands.

A VOICE-OPERATED ISOLATED COMMAND ENTRY TERMINAL

General Description. The VOICE terminal consists of a word-recognition module capable of recognizing words in a 100-word vocabulary, a Z80 microcomputer with 64K of memory and disk controller, and an 8-in. floppy disk drive. The video display can be either a video terminal or a liquid-crystal flat panel display for portability. The voice-entry module is not contained in the same enclosure as the portable ULTRA but exists as a separate unit that can be connected to the ULTRA computer through a serial port.

The word-recognition module we have used is the Interstate Electronics Voice Recognition Module (VRM).[4] Its circuitry is contained on a Multibus board and is housed in the same enclosure as the computer, which is also on a Multibus board.

The VRM recognizes spoken words only after it has been "trained" by the user. It can be trained to recognize up to 100 different words at a time. If a user needs more than 100 command words, the VOICE terminal's Z80 microcomputer can swap subsets of vocabulary words and voice patterns between a disk file and the VRM's memory.

A user begins using the VRM by training the system to recognize his or her voice patterns for a selected vocabulary. The words in a vocabulary represent commands for a specific application package such as titration, infrared spectrophotometry, calculator, etc. When the user utters a command, the VRM extracts certain features from the voice pattern and forms a digital representation of the pattern. The user will utter each command word in a vocabulary several times (usually 8 or 9 times) so the VRM can find an average digital representation of the voice pattern for each command in the vocabulary. These averaged digital representations of voice patterns are stored in the VRM's memory and compared with the patterns of future utterances. Because two different people will produce different voice patterns for the same word, each user of the VRM must train it to recognize his or her voice patterns for every word he or she intends to use.

After the user trains the VRM to recognize the vocabulary words he or she wishes to use, the VRM is switched to recognition mode. The digital representations of uttered words are compared with patterns that are stored in the VRM's memory to see whether there is a match. If the pattern of an utterance matches a stored pattern, the VRM sends a two-digit code for the word to the VOICE terminal's main processor. If there is not a match, the VOICE terminal asks the user to repeat the word.

The VOICE terminal uses a Z80 microprocessor to provide communication between the VRM and the ULTRA or other host computer. It can load the selected vocabulary into the VRM, interpret codes received from the VRM, and send the appropriate ASCII characters to the ULTRA host computer. The VOICE terminal can be connected to the ULTRA computer or any other computer through a standard serial link. However, the command vocabularies we have developed are primarily for use with the ULTRA calculator and for performing specific chemistry experiments with the ULTRA system.

Software for the VOICE Terminal. We used the C language for writing the terminal software because it is portable and was designed for writing system software; the Unix operating system is written in C. The software for the VOICE terminal is contained in two programs: CREATE and VOTERM. The CREATE program is used only to set up a master and application files for a user. It allows the user to define application vocabularies to be used with various software packages run on the host computer. The user can create application vocabularies and set up appropriate disk files to be used by the VOTERM program.

The VOTERM program is normally running when the VOICE terminal is in use. It reads the MASTER.VRM file to load the terminal command words, the user's name, and applications available to the user on disk. The user can then specify which application vocabularies he or she wants to load into the terminal's memory. If voice-pattern files are available on disk, they are also loaded into the VRM's memory.

The VOTERM program uses the CP/M files MASTER.VRM, ⟨application⟩.VOC,[5] and ⟨application⟩.PAT. The MASTER.VRM file is the master file and it contains the user's name, the names of the application vocabularies, and the number of words in each application vocabulary. The ⟨application⟩.VOC file contains training words (words to be spoken by the user) and command words (the actual ASCII character strings used in the vocabulary); it is a vocabulary file for a specific application. The name ⟨application⟩ is unique for each application. The ⟨application⟩.PAT file contains the digital representations of the user's voice patterns for words used in the application.

The video display is split into two parts. The top part displays command and vocabulary words available to the user. The bottom part is used to display information sent to or received from the host computer. A 24-line screen of dialogue with the host is stored in the terminal's memory. An eight-line window of this screen can be displayed on the video display. The user may display any part of the 24-line screen by scrolling forward or backward or by saying the TOP, BOTTOM, or MIDDLE commands. The screen-manipulating commands are listed in Table I with the other terminal commands. Spoken words for the alphabetic characters are listed in Table II.

ULTRA CALCULATOR PROGRAM

The ULTRA system has a calculator program residing in ROM as an integral part of the system. The ROM-based calculator is written in FORTRAN, and it, or its subroutines and functions, can also be used by the FORTRAN programs running experiments. The original version of the calculator program has been described previously.[6] We have rewritten the calculator so its subroutines can provide general utility functions for the programs used for experiments and so these programs can easily be linked to the calculator program residing in ROM. A more detailed description of the implementation of the calculator will follow in a subsequent paper.[7]

The ROM-based calculator provides an important tool for the handicapped student taking undergraduate chemistry courses. It is used by entering calculator commands from the terminal. The vocabulary words for the calculator program are listed in Table III. For example, the two numbers 5.1

Table I. Voice Terminal Command Vocabulary

Yes, No	response to terminal inquiries
Attention	clears the screen and rewrites the vocabulary words
Refrain	causes the system to ignore the users conversation until he or she says ATTENTION
Up Date	invokes a training pass for one of the application vocabularies
Remove	removes an active vocabulary from the terminals memory
Additional	adds an application vocabulary to the active list and loads it into memory
Enter	(1) leaves the terminal command mode and enters the terminal communication mode or (2) sends a carriage to the host if in the communication mode
Beginning	displays the first eight lines of the screen buffer
Middle	displays the middle eight lines of a 24-line screen buffer
Down	scrolls the display window down one line
Before	scrolls the display window up one line
Erase	back-spaces one character

Table II. Spoken Alphabetic Characters

spoken word	character	spoken word	character
Alpha	A	November	N
Bravo	B	Oscar	O
Charlie	C	Papa	P
Delta	D	Quebec	Q
Echo	E	Romeo	R
Foxtrot	F	Sierra	S
Golf	G	Tango	T
Hotel	H	Uniform	U
India	I	Victor	V
Juliet	J	Whiskey	W
Kilo	K	X-ray	X
Lima	L	Yankee	Y
Mike	M	Zulu	Z

Table III. Table of Calculator Commands

spoken word	command	result
Plus	+	add
Times	*	multiply
Divide	/	divide
Minus	-	subtract
Square Root	SQR	take the square root
Cosine	COS	cosine of an expression
Sine	SIN	sine of an expression
Exponential	EXP	exponential of an expression
Natural log	LN	natural log of an expression
Logarithm	LOG	logarithm of an expression
Tangent	TAN	tangent of an expression
Power	∧	raise to a power
Equals	=	store in a variable
Plot	PLOT	least-squares fit
Degrees	DEGREES	sets angular unit to degrees
Radians	RADIANS	sets angular unit to radians
Quit	QUIT	exits the calculator
Echo	ECHO	toggle to echo to the terminal
Say	SAY	lists values of a variable
Clear	CLEAR	deletes variables from the calculator
Fix	FIX	sets the number of the digits to be displayed in numeric values
Format	FORMAT	toggles between free format and scientific format
Save	SAVE	saves calculator data on disk
Load	LOAD	loads calculator data from disk

and 6.8 could be added by entering "5.1 + 6.8 RETURN". The RETURN command causes a carriage return character to be sent to the host. The calculator interprets this as the end of the statement, and it then performs the calculation. It then responds with "11.9" and is ready for the next calculator command. The calculator has all of the usual functions you would find on a scientific calculator that students use in freshman and sophomore chemistry labs. In addition, it contains 50 memory locations used for storing experimental data that are collected during a laboratory experiment. Hence, a student could collect data and perform the calculations needed for a laboratory report. Of course, it could also be used for working homework problems that require calculations.

The calculator also provides useful tools for the programmer writing programs used for collecting data for experiments. It contains a powerful set of command interpreting and I/O formatting subroutines that can perform useful functions for other programs running on the system. The software used for collecting data is written in FORTRAN. All of the data collection programs interact with the user through a package of I/O routines written for the calculator and stored in the calculator ROM area. Every FORTRAN data collection program is linked to the ROM-based calculator through a calculator interface routine that switches ROM/RAM memory banks and calls the appropriate calculator routine. When the calculator routine is finished, control is then passed back to the calculator interface routine that switches RAM back in and returns control and data back to the FORTRAN calling routine.

Because data collection programs are linked to the calculator, the user can enter a calculator command line at any time during an experiment. This feature allows students to have access to the calculator while they are in the process of collecting data. Many of the data collection programs actually store the data in the calculator's memory; this allows the user to do calculations on the data when they are in the laboratory performing the experiment. This way, students can quickly tell whether their experiments are successful or whether they should repeat certain portions of an experiment. For example, in an experiment involving pH measurement, a student could read the pH and calculate the approximate concentration by entering

```
READ PH1
CONC = 10∧(-PH1)
```

The value of the pH is read and stored in the calculator's memory at a location referenced by PH1. The value in PH1 is then used to calculate the approximate hydrogen ion concentration, and the result is stored in the calculator's memory at a location referenced by CONC. This ability to intermix calculator commands and experiment commands gives handicapped students the same flexibility that other students have to perform calculations while doing the experiment.

Calculator commands and experiment commands are entered with the VOICE terminal by uttering the appropriate command sequence. Each experiment has its own command vocabulary.

APPLICATIONS IN CHEMISTRY

The applications programs that a student would use to perform an experiment are not designed to do a particular experiment but are designed to use a type of instrument. For example, a student could use the titration program to do many different titrations, not just one type of titration, and the pH measurement program could be used to do a number of different experiments, just as a pH meter could be used to collect data for a number of different experiments.

A list of the programs for collecting data from various instruments is given in Table IV. All of these programs are written in FORTRAN and are linked to the ROM-based calculator. A student would run one of these programs by entering the appropriate program name from the terminal. If the program name is not part of an application vocabulary, then it could be spelled by using the alphabetic words Alpha, Bravo, etc. All of these programs can be used either by visually handicapped students or by students with upper limb disabilities. Visually handicapped students can use these programs

Table IV. List of Programs Used with Various Chemical Instruments

program name	program function
IR	collect and save data from an infrared spectrophotometer; locate and save peaks
GC	collect and save data from a gas chromatograph; find retention times and peak areas
TITRATE	measure volume delivered from a piston buret; find the end point
PH	measure the pH of one or more unknown solutions
TEMPER	make temperature measurements
SPECTRA	set up and collect data from a Spectronic-20
BALANCE	collect data from an analytical balance with BCD output
TITRATEV	control and collect data from a Fisher automatic buret/dispenser; used with voice commands

with a great deal of independence from a sighted assistant. A student with upper limb disabilities would still need an assistant to manipulate the glassware and instruments.

Data Collection and Review. The programs for collecting and processing data are written in FORTRAN except for the interrupt driven routines. The data can be collected, stored in the calculator memory area, and/or saved on disk. Previously collected data may be read from disk for review or additional processing.

Data from a scanning spectrophotometer or chromatography instrument are collected at discrete, evenly spaced time intervals. This is accomplished by using a Zilog Counter Timer Circuit (CTC) that contains four channels that can be used for counting or timing. We have linked three of the four channels together to form a 24-bit timer. The data acquisition programs for obtaining spectra use this timer to generate evenly spaced interrupts so data can be collected at evenly spaced time intervals. These time intervals are adjusted so that the rate of data collection is a multiple of the 60-Hz line frequency; this reduces any 60-Hz noise that might be present in the incoming signal.

The routines that set up the CTC channels and handle the interrupts are written in Z80 Assembler. These routines and the FORTRAN programs for doing the experiments are linked together to run as an integrated unit.

During a typical experiment, data are collected from an analog port, averaged, and passed to the FORTRAN program that is controlling the data collection and interacting with the user. A total of 1200 averaged data points would be collected in an IR spectrum and up to 2000 could be collected in a gas chromatogram. The user has the option to display the data as an audible tone whose pitch is proportional to the peak intensity.

After the data have been collected, the user can choose to process them immediately, or the data may saved in a disk file to be retrieved for interpretation later. Data from disk files can also be displayed as audible tones by using the Data Review routines, which allow the user to scan spectral data. The user can control the speed and direction (forward or reverse), stop the scan at any point, and obtain the numerical values of both axes at any point in the scan. These routines can be used to obtain retention times and peak areas from gas chromatograms, or a blind user may wish to just scan a spectrum to get an idea of the relative positions and heights of the peaks.

A VOICE-Operated Titration Experiment. We chose to develop the titration experiment as a demonstration project because titration is one of the fundamental manipulative experiments that students do in undergraduate chemistry laboratories. It is a fundamental skill required by undergraduate students in both freshman and sophomore laboratory courses.

In the voice-operated titration experiment, we use a Fisher Model 395 buret/dispenser, which has input/output lines that

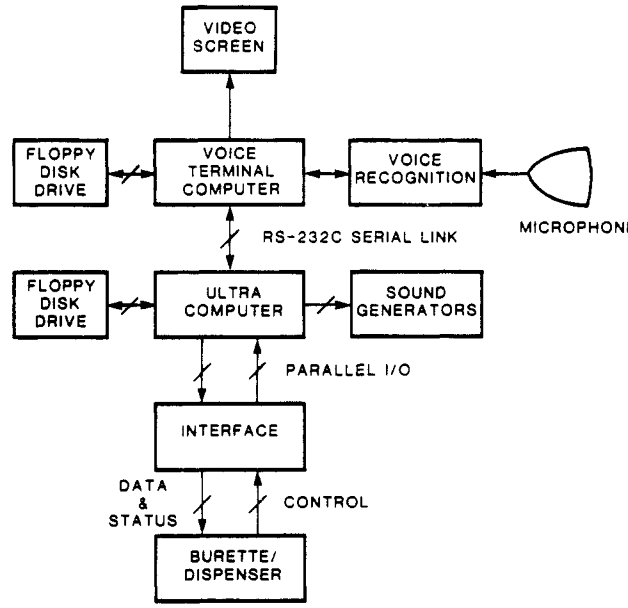

Figure 1. Block diagram of the VOICE system interfaced to a titration experiment.

Table V. Subroutines Used in Voice Titration Program

MAIN	handles the logic for carrying out a titration with the Fisher 395 buret/dispenser
BLOCK DATA PORTS	contains port initialization parameters for the parallel I/O ports used with the buret/dispenser
FILLIT	puts the buret/dispenser in the fill mode to begin filling it with titrant
STOPIT	puts the buret/dispenser in a temporary hold condition
STARTT	starts the buret/dispenser to begin a titration
DELAY(TC)	turns the titration pulse off after a time specified by TC
ZRODPY	resets internal parameters for starting a new titration
CLSCR	clears the video display
TINIT	sets program parameters for a specified buret size
TITLE	identifies the program by displaying "VOICE TITRATION PROGRAM"
VDPLY	reads and displays the volume from the buret/dispenser
DCODE1	converts ASCII digits to a floating point number
FILCHK	checks to see if the buret/dispenser is finished filling
PULS2	sends demand pulses to the buret/dispenser to do the titration; DELAY is called to turn the pulse off after a specified time

can be connected to a computer such as the ULTRA. The diagram of the system with the buret, interface, ULTRA, and the VOICE terminal is shown in Figure 1. We used the VOICE terminal to control the experiment, including filling the buret, dispensing titrant, controlling the rate of the delivery of titrant, and reading the volume. Volumes are read and stored in the calculator program's memory space to be used in subsequent calculations.

A laboratory assistant must set up the experiment by connecting the ULTRA to the automatic buret/dispenser. The titration inlet tube must be connected to a reservoir containing the titrant, and the outlet tube must be inserted into the solution to be titrated. The assistant must be present to change solutions between titrations. After the assistant has set up the experiment, the data collection can be done completely under voice control by a handicapped student.

The TITRATEV program is used to control the buret/dispenser and to collect data from it. The subroutines used in the TITRATEV program are listed in Table V. The voice commands used to interact with this program are listed in Table VI.

Table VI. Table of Titration Commands

spoken word	command	result
Titrate	TITRATE	start the titration
Stop	STOP	stop the titration
Fill	FILL	fill the buret
Set	SET	set the size
Reset	RESET	reset internal parameters
Size	SIZE	used to set the buret size
Read	READ	read the volume
Continue	CONTINUE	continue the titration after a temporary stop
Exit	EXIT	exit the titration program
End	END	end of the titration

A student begins by specifying the size of the buret that is being used. If the buret size used is not 25 mL, the buret size must be entered by using the "SET SIZE= " command. The syntax for this command is

SET SIZE=⟨number⟩

where ⟨number⟩ is the buret size. Allowable buret sizes are 10, 25, and 50 mL.

The buret can be filled with titrant by saying

Fill

No other parameters are necessary. The buret begins to fill automatically and is ready to begin a titration when the filling is completed.

The titration is started by saying

Titrate

This causes the buret to start delivering titrant. The speed can be controlled by saying a digit between one and nine; A one gives the slowest rate, and nine gives the fastest rate. A fast rate would normally be used during the first part of the titration. As the end point is approached, the rate of delivery can be slowed gradually by saying the appropriate number command. The titration can be interrupted or stopped by saying any command other than a number (which is for speed control). If the buret empties before the student stops the titration, the buret will be refilled automatically, and the program will query the student to see if the titration should be continued. Normally, the student will stop the titration at the end point.

The volume of the buret is read by saying

Read ⟨name⟩

where ⟨name⟩ is a variable name used for storing data in the calculator program's memory. The variable ⟨name⟩ can be used in subsequent calculations.

The buret can be reset for another titration by saying

Reset

This merely resets internal parameters in preparation for the next titration.

Occasionally, a user may want to dispense titrant continuously without speed control—cleaning the buret would be an example. The user can say

Dispense

Table VII. Results of a Base Standardization Titration

wt KHP (g)	vol of base (mL)	intermediate result	normality of base
0.9426	46.04	0.10025	0.1002
0.9803	47.88	0.10025	0.1002
0.8876	43.35	0.10026	0.1003

Table VIII. Results of an Acid–Base Titration[a]

vol of NaOH (mL)	vol of HCl (mL)	vol ratio (base/acid)
24.17	25.00	0.9668
23.88	24.70	0.9668
24.17	25.00	0.9668

[a] Normality of the acid is 0.1002 × 0.9668 = 0.09687.

to start continuous delivery of titrant. Continuous delivery is stopped by saying

Off

One of the questions that we wanted to answer was whether students could perform titrations under voice control with the precision necessary to pass sophomore quantitative analysis courses. The system was tested by an abled-bodied student using voice entry. We found that the voice-controlled titration was very precise and that the quality of the data obtained during the voice-controlled experiments was as good or better than that obtained by students using conventional titration techniques in the quantitative analysis laboratory. The reason for this is simple: with the computer we were able to deliver titrant very slowly (and precisely) when we approached the end point. The student can control the speed of the titration, so when he/she observes the approach of the end point, the delivery rate can be slowed to the extent that only a small fraction of a drop is delivered at one time. This system offers better control of titrant delivery than an able-bodied student would have using a manual buret. Sample data for acid–base titrations are shown in Tables VII and VIII. All end points were visual with phenolphthalein indicator.

Handicapped students need maximum independence in order for them to obtain the meaningful laboratory experiences that are a necessary part of the eduction of a scientist. Modern technology can be used to give them these experiences.

ACKNOWLEDGMENT

This work was supported in part under U.S. Department of Education, Special Education Program Grant G008004987.

REFERENCES AND NOTES

(1) Lunney, D.; Morrison, R. C. "High Technology Aids for Visually Handicapped Chemistry Students". *J. Chem. Educ.* **1981**, *58*, 228–231.

(2) Lunney, D.; et al. "A Microcomputer-Based Laboratory Aid for Visually Impaired Science Students". *IEEE Micro.* **1983**, *3* (4), 19–31.

(3) Bucy, J. Fred; et al. "Ease-of-Use Features in the Texas Instruments Computer". *Proc. IEEE* **1984**, *72* (3), 269–282.

(4) The Voice Products Operations of Interstate Electronics Corporation is now a separate company called Interstate Voice Products, is a subsidiary of FIGGIE International, and is located at 1849 West Sequoia, Orange, CA 92668.

(5) Items enclosed in angle brackets, "⟨" and "⟩", refer to names that can change. For example, ⟨application⟩ is part of a file name. The name ⟨application⟩ would change from one application to another.

(6) Morrison, R. C.; et al. In "Personal Computers in Chemistry"; Lykos, P., Ed.; Wiley-Interscience: New York, 1981; pp 164–176.

(7) Morrison, R. C.; Lunney, D.; Boswood, G., unpublished results.