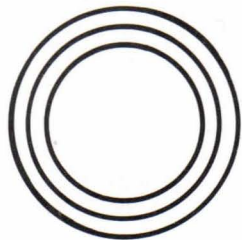


# VOICE KEY

SPOKEN COMMANDS ACT  
LIKE KEYBOARD INPUTS\*



FOR  
**COMMODORE 64**  
( & 128 IN 64 MODE)

**COVOX INC.**

\*Requires Voice Master hardware to operate

Do you have a favorite program that requires various keyboard letters as input? A game or educational or word processor? The object of VOICE KEY is to let you *command by voice* instead of pressing keys! VOICE KEY will not interfere with most programs (provided they do not use random-access memory "behind" read-only memory). Load VOICE KEY. Attach VOICE MASTER to the rear joy stick port. Then train the computer to the particular command words you want to use and define the keystrokes that you want these words to enter into the computer. Then load in your favorite program and proceed to run it using either keystrokes or command words. Up to 18 words. Each word can create up to 18 keystrokes including control characters. Complete instructions in manual. Disk is not copy protected. Special instructions for the popular "Speedscript" word processor.

Copyright 1987 by Covox, Inc., 675-D Conger St., Eugene, OR 97402.  
Tel: (503) 342-1271. All rights reserved.

Commodore is a trademark of Commodore Business Machines, Inc.  
"Speedscript" is a trademark of Compute! Publications, Inc. VOICE KEY and VOICE MASTER are trademarks of Covox, Inc.

## INSTRUCTION MANUAL

### VOICE KEY

For Commodore 64 and 128 (64 mode)

#### Introduction and General Considerations

April 15, 1987

Computer programs for business, education, and entertainment usually require the user to input data from the keyboard. For example, a game or teaching program might ask for a key press to represent up, down, left, faster, yes, no, etc. Or a word processing program can have review, center, underline, and other keyed in commands. Or you could speak a series of digits to enter numbers for a robot or an industrial control process.

VOICE KEY is a program that works with the VOICE MASTER so that voiced commands spoken into the VOICE MASTER microphone act like keystrokes. The program is designed to reside in computer main memory without interfering with the applications program. Furthermore, VOICE KEY does not require the user to do any programming, nor is knowledge of programming required. VOICE KEY does not disrupt original functions of the applications program--the original manual keystrokes can continue to be used. A single voiced command can replace a sequence of up to 16 keystrokes.

Note: VOICE KEY will not function properly with some applications programs because of conflicts in memory usage. Such matters are discussed later.

Although you don't have to be a programmer to use VOICE KEY, you do have to make "recognition templates" for the particular words you wish to use and also type in the sequence of characters that you want each recognized word to produce. The act of training puts templates for your own words into computer main memory. When you say a word to be recognized, it will be converted to a template and compared to templates in memory. A best-match criterion is employed to achieve recognition. If the computer can't find a reasonably convincing match, it won't accept your word, and it asks for a repeat. If your word might be confused with another word in memory, an error message will again be given with a repeat request. Outright recognition mistakes can be minimized by selecting distinctive words. (Errors can be evaluated to help in the selection process as will be described.)

Without writing your own programs, you can have up to 18 word templates at one time. A template is combined with corresponding typed equivalents and stored in memory as a unit. Each template set has its own file name. Any named template set can be loaded in. This makes it possible for one given program to utilize more than 18 commands. It also allows different people to use the same program with each person having a template set trained to his or her own voice.

#### General Operating Procedures

VOICE KEY in an applications program offers three recognition modes. The first has the computer always listening for a command. The speech input line is constantly scanned at a rate of 60 times per second in the same way that the Commodore keyboard is scanned. Background noise could cause false inputs. In a second mode, recognition cannot be attempted unless a shift key is pressed. (Don't press both shift keys because this has a different function to be described.) In the third mode, no recognition is possible and only actual keyboard inputs are accepted.

Installing and utilizing VOICE KEY will be discussed in detail in the balance of this manual. But we first give a brief overview of the steps you take to create a recognition file, and then how to use VOICE KEY in an applications program.

1. Make a backup of the VOICE KEY disk. Use the backup and put the original away in a safe place. The VOICE KEY disk is not copy protected. See the VOICE MASTER User Manual (or the Commodore User Manual) for instructions on making backups.
2. Decide on an applications program.
3. Determine which keystrokes in this program you want to be implemented in response to a voiced command.
4. Choose the words to be used for commands. Pair these with a descriptive word or two (for your reference only).
5. Make the necessary VOICE KEY templates, check accuracy, etc. and save to disk. You will in the process save the reference words as well as the sequence of keystrokes to be put into the program in response to recognition.

With a template set for recognition made as described above, perhaps on some previous day, proceed as follows to install VOICE KEY with the applications program:

1. Load in VOICE KEY from the backup disk.
2. Select the previously made template file.
3. Load in the applications program and start it as required in the user manual for that program.

Note that step 3 above can begin immediately after step 5 of the template making sequence. You need not reload VOICE KEY.

The VOICE KEY program has a short BASIC loader program. Applications programs in BASIC, or machine language applications programs with short loader segments written in BASIC, will overwrite the VOICE KEY loader program. But once it has acted, there is no further need for the loader program. It would be wrong to load VOICE KEY after an applications program with some BASIC program segment because VOICE KEY could disrupt the applications program. However, if the applications program has no segments written in BASIC, it is possible to reverse the loading procedure. This may in fact be necessary if special changes in the applications program are needed in order to make VOICE KEY work with it. Such is the case, for example, with the popular word processing program called "Speedscript", available from Compute Publications. (See Appendix.)

With VOICE KEY operating by itself, or if running with an applications program, you can return to an option menu at any time by pressing both shift keys simultaneously (and use care to not press one before the other). This will permit you to change recognition options, load in a different template set, check gain, or perform other tasks. You will be able to return to the applications program from this menu. Example: The operator tires and is replaced with a different person. But the program remains functioning. The new operator should load in his/her own template set. Another example has a complex multi-part program requiring different sets of commands for different

segments of the program. Or perhaps the program is an educational one having different sets of test questions requiring different answers.

Not all applications programs will work with VOICE KEY because certain memory locations may sometimes conflict. VOICE KEY accomplishes word recognition by interrupting the keyboard scan. The interrupt program can be located anywhere in useable memory. In many cases, getting an applications program to work with VOICE KEY requires only that the interrupt program be relocated to an unused part of memory. VOICE KEY also uses 8K of random access memory (RAM) "behind" the BASIC read-only memory (ROM) as well as some RAM "behind" the kernal ROM. These parts are not re-locatable and will prevent use of VOICE KEY with programs that also occupy these areas of memory. Applications programs written in BASIC, if they do not utilize calls or other special places in memory, should never conflict with VOICE KEY (except possibly for the interrupt program which can be re-located to avoid conflicts).

#### Loading VOICE KEY

Turn on the computer. Insert the back-up copy of the VOICE KEY disk. Load and run the BASIC loader program as follows:

```
LOAD"*",8    (RETURN)
RUN          (RETURN)
```

You will then be asked for the place to locate the interrupt routine. If there is no reason to change from the default value, press RETURN. (Otherwise, specify the desired location as discussed later.) You are advised of the default location 40704 (decimal) which lies just below BASIC. When the message disappears, the READY prompt re-appears.

To access VOICE KEY, hold down both shift keys at the same time and a display will appear on the screen. The top line of the display is the "option menu" which lists the various program options. Beneath this are six more numbered lines which specify the various options available. At the bottom of the display is the "status line" which shows important program information.

There are eight selections available from the option menu. To select one, hold down RUN/STOP and press the key that corresponds to the first letter of the option. For example, to select END, hold RUN/STOP and press E. A brief description of the options follows:

**VIEW:** This is the default mode which displays the 18 commands and responses. The listing can be scrolled with the standard Commodore cursor keys. The VIEW mode is not manually selectable--it is the default mode which appears after any task is completed in another mode (except for END).

**TRAIN:** This mode is used to make voice templates for recognition. Pressing any key while in this mode will cause an exit to the VIEW mode.

**CHECK:** Allows for testing of recognition accuracy. After each recognition check, the results are displayed and a beep or buzz is sounded. The high pitch beep indicates that a matching template was found and the low pitch buzz that it was not. Sensitivity can be changed from this mode. (See below.)

**SAVE:** For saving all templates and text data to disk.

**LOAD:** Loads previously saved templates and commands.

**MODE:** Select from one of three recognition modes.

1. Recognition during 1/60 second interrupt scan.
2. Recognition only when one shift key is pressed.
3. No recognition--keyboard action only.

GAIN: Used to check calibration and gain setting of the VOICE MASTER speech processor.

END: Exit the VOICE KEY display and return to the applications program, or BASIC.

### Entering Commands

Within the "command window" on line number one, a flashing cursor appears. The cursor can be moved about on the screen with the cursor control keys. The up/down key moves the cursor up when pressed alone, or down when pressed with the shift key. The right/left key moves the cursor right when pressed alone or left when pressed with the shift key. Use the up/down key to scroll through the 18 command lines.

Each numbered line within the command window is divided by a comma into two text fields. The field on the left of the comma is for typing in the nature of the command to be recognized; this text is for identification and is not used otherwise in the program. On the right of the comma is the field for entering the keyboard response to a recognized word. Text entered in this field represents the actual characters that will be placed in the computer keyboard buffer whenever the corresponding voiced command is successfully recognized. Up to 16 characters can be entered, including non-printing characters and spaces.

Up to 18 commands and their corresponding responses can be entered in the command window. In order to illustrate the procedure, we give an example with only two commands.

First, position the cursor within the left text field for line number one and type the reminder command "DIRECTORY". Do not press Return. If you should make a typing mistake, position the cursor back over the character(s) and retype. Next, use the cursor control keys to move the cursor to the response field just past the comma. Type

LOAD"\$",8 (RETURN)

When the RETURN key is pressed, a left arrow character will be printed. This is used as a symbolic representation for the non-printing RETURN character (ASCII #13). Some commands employ a RETURN as part of the command, which altogether could involve two or more RETURNS. Putting in a return does not tell the program that all of the command has been entered. Completion of the response text requires a defined terminating character. Without such a terminating character, the remaining space characters following the text in the response field (for 18 total characters including spaces) will also be put into the keyboard buffer when the command is recognized. To enter the terminating character, hold down the shift key and then press the key for " ". (This particular key also shows a large up arrow on the top.) When you have finished entering the command and response text, line number one should appear on the screen as

1) DIRECTORY , LOAD"\$",8 -

where the space following the comma is required. Now move the cursor down to line number two and enter the list command and response so that this line

appears as

2) LIST           , LIST -

This example sequence will require two voiced commands, one to load the disk directory into main memory, and the other to list it. You can load and list with a single command by typing a single command line having two commands separated with a colon as follows:

1)DIRECTORY       , LOAD"\$",8:LIST -

which uses up all 16 of the allowed letters. You can shorten the command line with the short form for commands as

LOAD   L and SHIFT O  
LIST   L and SHIFT I

Sometimes you may want to enter special editing control characters in the response field, such as "delete" (ASCII #20) or "insert" (ASCII #148). You can insert control characters by holding down the CONTROL key while pressing the appropriate editing key. For example, to enter the delete character, hold down CONTROL and press the DEL key, followed by . The delete character is shown as a reverse "T".

#### Gain Check

Before attempting to make templates or do a recognition, you should make certain that the VOICE MASTER has been properly calibrated and that the gain control is set correctly. From the menu, press RUN/STOP and G to get to the GAIN mode. A horizontal amplitude bar is displayed which lengthens in direct proportion to the loudness at the microphone. This mode allows gain and calibration adjustments to be checked or changed. Complete instructions for setting calibration and gain are given in the VOICE MASTER User Manual. Brief instructions are given here.

Although you may wish to calibrate as described at the end of this paragraph before setting gain, we will discuss the gain setting first. As you speak into the microphone, adjust gain on the VOICE MASTER so that speech peaks extend moderately beyond the point on the bar where the marks used to represent the bar change from asterisks to a plus signs. This is the nominal saturation level and some exceeding of this level is normal and actually desirable. If you don't make any sounds to the microphone, no part of the bar should show. A soft sound should cause the bar to barely show and a nasal sound ("n" or "m") should reveal two or three increments of amplitude. If necessary, adjust through the calibration hole on VOICE MASTER so that the bar barely disappears. It is suggested that the microphone be unplugged when making this adjustment. A large change in the calibration adjustment may require a slight change in the gain setting.

#### Making Templates

The next step is to make voice templates for each command. To make a template for command number one, position the cursor anywhere on line number one, then hold down RUN/STOP and press the T key. When both keys are released, a question mark, "?", will appear in the upper right corner of the screen. Also, the first letter of the word TRAIN in the option menu will be highlighted. At this point the program is waiting for microphone input and is ready to make a template when you speak the command. Say the word, "directory", and a template will be made. After training this one word, the program will return to the

VIEW mode and the status line will indicate that the template is enabled. Also, the number at the beginning of the command field will no longer be highlighted. Each command should be trained twice (or more) for best recognition accuracy (by simply repeating the same procedure in exactly the same way).

Next TRAIN the word "list" (twice) by putting the cursor on line number 2. Then proceed as above.

If you should press a key while in the TRAIN mode, a tone will sound, indicating an error, and the program will return to the VIEW mode. You might wish to do this on purpose in order to escape from the TRAIN mode. The tone will also be heard if a second or subsequent training of a particular template is unsuccessful. If this happens, then you should blank the template (i.e., erase it from memory) and retrain it again (twice).

To blank a template, position the cursor over the number at the beginning of the command field, press the space bar, and the status line will indicate that the template is BLANK. The blanking operation does not actually erase the template from memory unless you retrain. If you have blanked by pressing the space bar when the cursor is over the word number, and have not retrained, then simply press the space bar again and unblank the template. The space bar acts as a blank-unblank "toggle". But training after blanking will erase the template permanently. If you want to change a template from one word to another, be sure to blank before retraining or else the new template will be a confused average of both words. Blanking templates can be very useful in applications where all of the words represented by templates are not required, but where it is desired to retain the word templates themselves for future use.

If a template is blanked, the computer does not have to scan it in an attempt to find a match. With fewer templates to scan, the speed of recognition will be increased and fewer recognition errors will be made.

When waiting for speech input while in the TRAIN mode, the question mark in the upper right corner of the screen should be motionless. At most, only an occasional flicker should occur. When speaking, it will jitter about, which signifies that speech is being accepted. If the question mark jitters about when no speech is input, there may be too much background noise, or calibration may be in error. In this event, and if calibration is correct, try reducing the gain on the VOICE MASTER and talk louder and/or bring the microphone closer to the lips.

Note: You cannot make templates using standard VOICE MASTER programs and then use these with VOICE KEY. This is because VOICE KEY recognition templates also include extra information such as command names and the actual keystrokes to be placed in the keyboard buffer.

#### Checking Recognition Accuracy

Trained commands can be checked for accuracy and reliability. Select the CHECK option from the menu and speak the command. If the word you speak matches one of the trained templates, you will hear a beep and the number of the word recognized will be displayed on the screen. If a match is not found, then you will hear a short buzz and an error number will be displayed. The possible errors are:

248: All words are a blank and/or were never trained.

251: Recognition terminated prematurely by a key press (other than space bar).



253: Speech input buffer capacity was exceeded.

254: Minimum error. No recognition because two or more templates matched too closely to make a good decision.

255: Maximum error. No recognition because no acceptable match was found.

In addition to displaying the word recognized or an error, the first and second choice of template matches will also be indicated. The first choice is the number of the template that was determined to be the closest match to the word being recognized. The second choice is the number of the template which was determined to be the second best match to the word being recognized. This information can be quite useful for choosing which words work best in a command vocabulary. If problems arise, one of the words can be changed in order to increase template differences.

Note: Memory location 827 (\$33B) is used to store the index number (1-18) of the word recognized, or otherwise one of the error codes (248-255). This location can be PEEKed from a BASIC program (an example being described later).

When in the CHECK mode, you can adjust the recognition sensitivity level (which is essentially the same as "ACCEPT" in the VOICE MASTER program). The lower the sensitivity value, the less restrictive will be the recognition algorithm in determining a template match. See the VOICE MASTER User Manual for further details. Experimenting with the sensitivity level can help to improve recognition accuracy and reliability.

#### Recognition Modes

The MODE option permits selection of the manner in which word recognition is handled. In MODE 1, the VOICE KEY program continuously scans the microphone for input, at a rate of 60 times per second. If a signal of sufficient magnitude and duration is sensed, the program will attempt a recognition. In this mode, response is by verbal cue alone and the keyboard need not be touched. However, MODE 1 is not very useful in a noisy environment where background noise can interfere with recognition (although talking loudly with the microphone close to the lips helps). When in MODE 1, the speed with which an applications program runs will be slowed because VOICE KEY must constantly scan to see if a sound has entered the microphone.

In MODE 2, input from the microphone is only accepted when one (only) shift key is pressed. This does somewhat limit the operator, but functioning in a noisy environment is improved.

MODE 3 effectively turns off recognition so that input is possible only from the keyboard. But VOICE KEY does remain in memory and recognition can be re-enabled at any time.

#### Saving and Loading

The SAVE option allows the entire command file to be saved to disk. All text, templates, and sensitivity settings will be saved. When prompted for a filename, enter the name you have given to the particular file desired. The program assumes drive No. 0 and device No. 8. If you have previously created and saved a command file, then you can use the LOAD option to retrieve it. Everything except the recognition MODE setting will be restored to its previous state. You must manually reset the desired MODE.

#### Exiting VOICE KEY

After creating or loading in a command file and setting the recognition mode, you will want to leave the VOICE KEY display and return to BASIC or some other resident program. Simply select the END option (RUN/STOP and E) from the menu and answer "Y" to the verifying prompt. The original screen display will be restored and VOICE KEY will be active in the recognition MODE that was selected.

Recall the example involving the directory where two words were trained and corresponding keystrokes for outputting to the keyboard buffer were produced. Select MODE 2 and exit VOICE KEY via the END option. The Ready prompt will appear and BASIC will be in the same state as it was before entering VOICE KEY, except that you can now use VOICE KEY for voice commands. Press the shift key and the question mark appears on the screen to indicate readiness to accept a verbal command. (A small "click" will also be heard which helps to verify operation without having to closely examine the display.) Say "directory" and the response LOAD"\$",8 will appear on the screen just as if you had typed it in and BASIC will immediately begin loading the disk directory. When the Ready prompt appears again, press the shift key and say "list" and the directory is displayed on the screen. If you want to go back to VOICE KEY, press both shift keys at the same time.

In MODE 2, speech input for word recognition begins when the shift key is released--not when it is first pressed or while it is held down. If any other key is being pressed when the shift key is released, the recognition mode is terminated and the keyboard operates in a normal manner. You can get back to the recognition mode by pressing and releasing the shift key by itself.

#### VOICE KEY and BASIC

Many available programs, especially those in the public domain, are written in BASIC. Most Commodore owners who do programming do their work almost entirely in BASIC. Inputs desired from the keyboard are quite obvious when you run the program without VOICE KEY. Thus you can create a template set appropriate to a particular program rather easily so as to represent keystrokes as commands. Load in VOICE KEY and make your template set before loading in the BASIC applications program. Or load in VOICE KEY and then load a prepared template set from disk and select the recognition MODE (1 or 2). Finally, load in the applications program and RUN. You can now use either the keyboard or voice commands.

Keyboard input within a BASIC program can be accomplished in only two ways, with an INPUT statement, or with a GET statement. (Actually there is at least one other way. A PEEK can be put into a program in order to evaluate the last keystroke put into the keyboard buffer. This method can implement modified kinds of decision jumps.) In either case, the keyboard expects a string of one or more letters or numbers and this is what you prepare as a command, to be implemented with a spoken word. If you know something about programming in BASIC, you will be able to make changes to some programs if you so desire, perhaps to do things by voice that you would not normally want to do with keyboard input. If writing an original program is your objective, then perhaps you should use the more fundamental programs provided with VOICE MASTER where more versatility is available and also speech output can be used along with recognition.

Recall that the index of the recognized word is placed in memory location 827 (\$33B). If you PEEK this location and use an IF-THEN statement to branch to some condition with a GOTO, then you can cause a program to halt, branch, or do other such things. If you cause a STOP, you can look at the directory by voice

command, or even load in a different BASIC program and then RUN it.

Recall when first loading in VOICE KEY that you are asked for a location to install the interrupt routine. You were advised to select the default case. But you could put this routine anywhere in BASIC. In some cases, the default location may interfere with a program. Trial and error in relocating this routine can sometimes solve a problem. For "Speedscript" (see Appendix), you are asked to locate the program in the "cassette buffer" by selecting a menu option.

#### Some Technical Information

The VOICE KEY program was designed to be as "transparent" as possible so that it would work with a variety of other programs. However, it does require memory for itself as well as for recognition templates. An area of memory that is seldom used by other programs is the 8K of RAM "behind" the BASIC ROM. This is where the main code for VOICE KEY is located and it will not function with any other program that also uses this area. Another major area of memory, also seldom used, is the 8K of RAM located behind the Kernal ROM. Of this area, VOICE KEY requires the portion from \$F200 to \$FFFF as a speech input buffer and for temporary data storage. It is possible for VOICE KEY to share this area of RAM with some other programs that also use it for temporary (not permanent) storage. Finally, VOICE KEY uses 5 bytes for the two interrupt vectors and a recognition word index pointer, from \$337 to \$33B.

In order for VOICE KEY to perform without keyboard action (recognition MODE 1), it must periodically check for microphone input. This is accomplished by using the 1/60 second interrupt function of the Commodore. Unfortunately, many other programs also use this interrupt for their own purposes and this can be a source of conflict. (Programs written entirely in BASIC which do not have DATA statements that load in a machine language program with POKES, or which do not use POKES to put data in arbitrary places, will generally be compatible with MODE 1.) Two things have been done to minimize compatibility problems. First, the machine code for the interrupt routine used by VOICE KEY is less than 170 bytes long and is entirely relocatable. The BASIC program "VOICEKEY" contains the code for the interrupt routine as a series of DATA statements which are READ and then POKED into the chosen area of RAM. Second, the entry and exit vectors of the VOICE KEY interrupt routine are re-definable. The two byte starting address of the routine is placed into an entry vector at locations 823 (\$337) and 824 (\$338) (in the usual low-byte, high-byte order). The starting address of the interrupt routine in the applications program could be put here and this will cause the 1/60 second interrupt to vector to this routine first. However, the code must terminate with a jump (JMP) to the starting address of the VOICE KEY interrupt. The two byte exit vector is at locations 825 (\$339) and 826 (\$33A). Normally, this vector will be set to point at \$EA31. This effectively gives you the option of attaching your own interrupt code to the end of the VOICE KEY code. Once the entry and exit vectors have been defined, the VOICE KEY interrupt routine may be activated. See the REM statements in the "VOICEKEY" BASIC program for details.

#### APPENDIX

##### Use with "Speedscript"

VOICE KEY can be used with word processing programs to handle special functions such as "paragraph", "restore text", "delete", "underline", or long words (up to 16 characters) for text insertion. One of the more popular programs is "Speedscript" by Compute Publications. It works well with VOICE KEY, but some modifications are required with a program on the VOICE KEY disk called

"SPEEDFIX" (only applicable to "Speedscript"). Other word processing programs might work with VOICE KEY without changes. Only experimentation can reveal which will work and which will not.

The following instructions will get "Speedscript" up and running with VOICE KEY.

1. Remove cartridges from the computer and turn on power.
2. Insert the VOICE KEY disk and type  
LOAD"SPEEDFIX",8 (RETURN)  
RUN (RETURN). Follow directions. (Put in "Speedscript" disk and press any key. Then insert VOICE KEY disk and press any key.)
3. Locate the interrupt vector in the "cassette buffer" by selecting option number 2 when prompted.
4. After the Ready prompt, type SYS 2061 (RETURN).

Speedscript will now be running and you can access VOICE KEY at any time by pressing down both shift keys. You can now add in customized voice commands. When done, select recognition MODE 2 and then exit via the END menu option and control will return to "Speedscript".

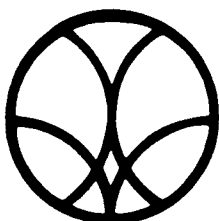
Note: Because "Speedscript" uses the 1/60 second interrupt for its screen display, recognition MODE 1 cannot be used effectively.

A template file example SPEED.TEMP is supplied on the VOICE KEY disk. Load as previously described. Refer to the "Speedscript" manual if changes in printer responses are needed for your particular printer.

#### VOICE MASTER Connections

If you have a Commodore 128, it must be operated in the "64" mode. Hold down the Commodore key and turn on the computer. Or if the computer is running in the 128 mode, type GO 64 (RETURN). If you have a monitor with a 40/80 column switch, it must be set to the 40 column position.

A special version of VOICE MASTER software is on the VOICE KEY disk. Speech recording and playback are not available with VOICE KEY. The VOICE MASTER plugs into the rear joy stick port and obtains power from the computer. For alternative microphones and use of the headphone on the headset, refer to the VOICE MASTER User Manual.



**COVOX INC.**

675-D Conger Street • Eugene, Oregon 97402 • U.S.A.  
Area Code (503) 342-1271 • Telex 706017 (Av Alarm UD)