COVOX iNC.
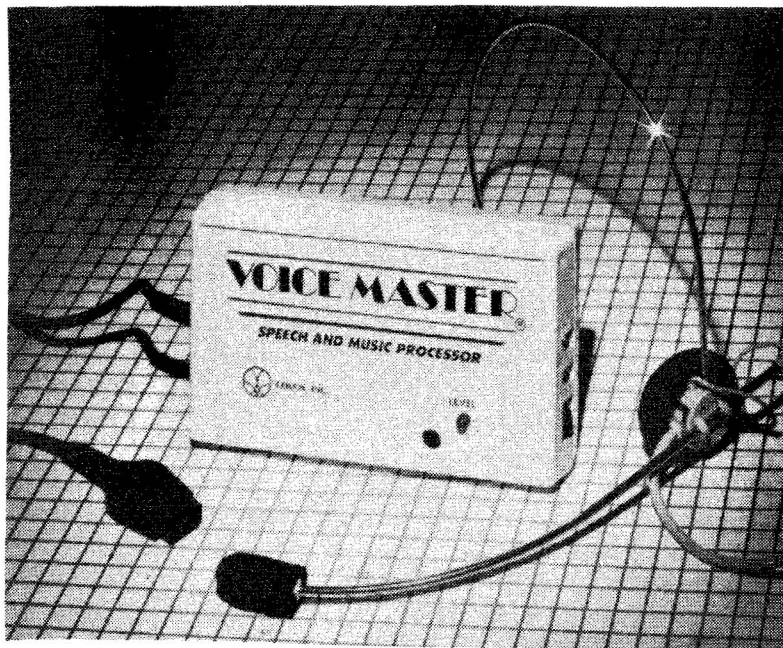
VOICE MASTER OWNER'S MANUAL

Version AT/S1.0

ATARI 800, 800XL, 130XE

Includes:
SPEECH RECORDING AND PLAYBACK
SPEECH WORD RECOGNITION

Programming and manual by

Kevin C. Gevatosky

December 1, 1985

LIMITED WARRANTY STATEMENT

COVOX, Inc. guarantees the VOICE MASTER to be free from defective materials and workmanship for a period of one year from the date of purchase. COVOX, Inc. will replace defective parts and make repairs under this warranty when the defect occurs under normal use, provided the unit is returned to the factory via prepaid transportation. The warranty provides that examination of the returned product must disclose a manufacturing defect to be judged by COVOX, Inc. The warranty does not extend to any product which has been subject to misuse, neglect, accident, improper installation, or where the panel legends or other markings have been removed or defaced, and is given in lieu of any other warranty implied or expressed, and will not cover any consequential damages.

Information in this manual and associated software are provided on an "as is" basis. No warranty, either expressed or implied, is made by COVOX, Inc. pertaining to suitability for any specific application or commercial use. It is the purchasers responsibility to make appropriate evaluations for such purposes. COVOX, Inc. disclaims liability for direct, indirect, or incidental damages arising from the use of this product, including but not being limited to interruption of service, loss of business or potential profits, legal actions, or other consequential damages.

PATENTS AND COPYRIGHTS

The software supplied with VOICE MASTER is copyrighted. It may not be copied, reproduced, translated, or reduced to any readable medium or code for other than personal use without prior written permission of COVOX, Inc.

The hardware/software system comprising the COVOX VOICE MASTER is subject to existing patent applications. Unauthorized duplication for commercial purposes or to otherwise avoid payment of appropriate royalties or license fees will be deemed to be a violation of proprietary rights under patent and trademark laws.

The names COVOX, VOICE MASTER, and VOICE HARP, and the COVOX "logo" are registered trademarks and are the property of COVOX, Inc.

THE VOICEMASTER USER MANUAL
by
Kevin C. Gevatosky

TABLE OF CONTENTS

```
**********************************
          PART I
        INTRODUCTION
**********************************
```

This manual is devided into five parts and accompanying appendices. Part I is concerned with interfacing the VOICEMASTER unit to your Atari and 'booting-up' the disk software. Parts II and III will explain how to use the VOICEMASTER for recording and playback of speech. Recognition of speech will be covered in Part IV and Part V will discuss loading and using the various programs that are on the VOICEMASTER disk. When reading this manual, it is important that you read it from beginning to end without skipping over anything. Each part contains pertinent imformation concerned with proper operation of the VOICEMASTER unit and software.

MAKING A COPY

Before proceeding any further, a copy of your VOICEMASTER disk should be made and the original kept in a safe place as a back-up only. If something happens to the the copy while using it, then you can always make another from the original. This is a good practice to follow whenever possible and could save you valuable time and money.

Making a copy is easy because we have supplied, for your convenience , ATARI DOS 2.5 on the VOICEMASTER disk. The VOICEMASTER disk has no notch ('write protected') and cannot be 'written' to, but the contents of the disk can still be copied. The software itself is protected under copyright, however you have our permission to make a back-up copy, but only for your own personal use. To make a copy, just follow these simple steps:

1. Turn off your computer and remove any cartridges. If you have an Atari 800, then insert the BASIC cartridge.

2. Be sure the disk drive is on and set for single density. Then insert the VOICEMASTER disk.

3. Turn on the computer and the VOICEMASTER disk will automatically 'boot-up'. When the loading process is finished, the VOICEMASTER MENU will be displayed.

4. Type the number for: 'EXIT TO BASIC' and when the READY prompt appears: type DOS then press RETURN.

The DOS 2.5 menu will appear on the screen.

    5. Remove the VOICEMASTER disk and insert a blank disk
in the drive.

    6. Select item: 'J. DUPLICATE disk' from the menu and
press RETURN.
when you see the prompt: 'DUP disk-SOURCE, DEST DRIVES?'
answer by typing: '1,1' and then press RETURN.

    7. Follow the prompts for inserting source and
destination disks as required. It will take several swaps of
the disks to complete the duplication process.

    You should now have a complete copy which will function
exactly like the original.   However, this disk is 'full' of
programs and there is no room to put your BASIC programs or
speech files on it. If you need more space on the same disk,
then refer to the section in PART V of the manual on, "HOW TO
MAKE A CUSTOMIZED VOICEMASTER DISK."

SETTING UP THE VOICEMASTER

    Interfacing the VOICEMASTER unit with your Atari
couldn't be simpler. Just follow the procedure below and the
VOICEMASTER will be 'up and running'.

    1. Turn off your computer and remove any cartridges. If
you have an Atari 800, then insert the BASIC cartridge.

    2. Find the port with the number '2' stamped beside it
and carefully plug the VOICEMASTER unit into it.

    3. Insert the BLACK plug of the headset into the jack
marked 'MIKE'. Ignore the red plug for now. Its use will be
explained shortly.

    4. Be sure the disk drive is on and set for single
density, then insert your copy of the VOICEMASTER disk (not
the original which should, by now, be a back-up copy).

    5. Turn on the computer and the VOICEMASTER disk will
automatically 'boot-up'.   When the loading process is
finished, the VOICEMASTER MENU will be displayed.

    6. Type the number for the 'BARGRAPH' option.   The
program will load into memory and await input from the
microphone.

7. Locate the 'GAIN' control knob on the VOICEMASTER unit and turn it 'counter-clockwise' until it stops. Now speak into the microphone and turn the knob 'clockwise' as you continue to speak. When the red light, marked 'LEVEL' on the unit, just begins to 'flicker' then stop turning the knob. The 'GAIN' control is now properly adjusted.

If the above steps were carefully followed, then you should now see a display of 'dancing' vertical bars on the screen when you speak into the microphone. If there are no bars on the screen, then the VOICEMASTER unit probably needs calibration. If so, follow the procedure given at the end of the chapter in the section called: "CALIBRATION OF THE VOICEMASTER".

The bar farthest to the right on the screen indicates the 'amplitude' or loudness of the speech. The number displayed at the base of this bar is an additional amplitude indicator. Both the bar and number display are equivalent. When amplitude exceeds 15 units then each bar segment above this value will change to an inverse 'X'. The bar immediately to the left of the amplitude bar indicates the 'fundamental pitch' of the speech. Further to the left, is a histogram composed of 12 'frequency' bars which will respond to the various frequency patterns of your voice. Try speaking sounds like 'eee', 'aahhh' and 'ooo', and watch the patterns of the bars. The frequency bars will plot sounds with the highest frequency toward the left and those with the lowest frequency toward the right.

Feel free to experiment with various sounds, such as humming and whistling, until you are comfortable with the bar-graph display. When you want to quit, exit the 'BARGRAPH' program by pressing the SPACEBAR, or any other key, and the main VOICEMASTER MENU will be displayed again.

Now select the 'DEMONSTRATION' option from the menu. After a few moments the program will load into memory and display a second menu. Choose any option as you desire. Just type the corresponding letter and follow the prompts. When you have finished experimenting with the demonstration program, select the: 'To main VOICEMASTER MENU' option from the menu.

USING THE EARPHONE JACK

The headset is equipped with an earphone for private listening of sound or speech output from the computer. For this purpose, two additional cables have been supplied with the VOICEMASTER unit. The cable with one large and one small end is for use with a telivision; and the cable with two small ends is for use with a monitor. To use the earphone capability, insert the RED plug of the headset into the jack marked 'EAR', which is adjacent to the 'MIKE' jack. Select the appropriate cable (T.V. or monitor) and plug the 'male' end into the jack marked 'EAR IN', on the VOICEMASTER unit. If you are using a T.V., then plug the large 'female' end into the jack marked 'MONITOR' on the rear of the Atari. If you are using a monitor, then attach the small 'female' end to the 'AUDIO' end of your monitor cable. The earphone of the headset will now work, even if your monitor does not have audio capability.

CALIBRATION OF THE VOICEMASTER

Your VOICEMASTER unit was carefully calibrated at the factory and should not require further adjustment. If calibrated correctly then when you are not speaking, and there are no other nearby sounds, the amplitude bar should not be seen on the screen and its value should read '00'. When you speak softly into the microphone the amplitude bar should begin to appear on the screen. If all is well, then skip the rest of this section on calibration, and begin reading PART II of the manual.

For calibration, the VOICEMASTER unit should be plugged into joystick port #2 and the black plug of the headset removed from the 'MIKE' jack. With the bar-graph program loaded and running, insert the 'screwdriver-like' tool (supplied with the VOICEMASTER) into the small slot of the potentiometer which can be seen through the hole marked 'CALIBRATE'. Turn the tool 'counter-clockwise' until the amplitude bar can be seen on the screen. Now turn the tool 'clockwise' ,very slowly, until the amplitude bar just disappears from the screen and the numbers at the base read '00'. The VOICEMASTER unit is now calibrated. Reinsert the black plug of the headset into the 'MIKE' jack. Speak into the microphone and readjust the 'GAIN' control as previously instructed (see step #7 above).

```
************************************
                PART II
    LEARNING TO USE THE VOICEMASTER
************************************
```

The VOICEMASTER unit's controlling software is a sophisticated machine language program which operates via 'wedged in' BASIC commands. This wedge adds new commands to the standard repertoire of Atari BASIC commands. All the added commands were designed to be compatible with Atari BASIC and follow the same general format as the standard commands. The wedge operates in both Direct Mode and Program Mode.

In Direct Mode, the command is executed immediately after being entered. Just type the command (no line number) with the appropriate parameter and press RETURN. Remember, as with any BASIC command in Direct Mode, you must press RETURN after the command in order for the computer to know that you have entered it.

In Program Mode, the command is entered within a BASIC line which has a line number, and hence, becomes part of the BASIC program. Multiple commands within a single program line are also supported by the wedge (as they also are in non-wedged Atari BASIC). See APPENDIX IV for more details on using the wedge with Atari BASIC.

There are 18 (19 for Atari 800) added BASIC commands available to you. With these added commands, even the novice who has little or no programming experience, can use the VOICEMASTER to record, playback, and recognize speech.

Most of the commands require that a parameter 'n' be specified, whereas with other commands the parameter is optional or unnecessary. A parameter can be entered as a numeric constant (0,45,3 etc.) or given a legal variable name (Z,WORD,X2 etc.). Any variable which is not an Atari BASIC or VOICEMASTER command and begins with an alphabetic character is legal. If a command is entered with an incorrect parameter then an ERROR-3 will result. If the command is within a BASIC program then the ERROR-3 will not occur until the program is executed, whereupon BASIC will STOP execution and display the line number where the ERROR-3 occurred.

GETTING FAMILIAR

Now it's time to get introduced to your new VOICEMASTER. We start here with some easy programming in Direct Mode. For now, just follow the given examples. Details about all the available commands will be thoroughly covered further on in this manual.

If you have not already booted the VOICEMASTER disk, then please do so now according to the directions given in the chapter on SETTING-UP.

From the VOICEMASTER MENU, select the 'EXIT TO BASIC' option. When the READY prompt appears, type NEW and press RETURN. Get the microphone ready for speech and type the command LEARN 1. When you are ready to speak, press RETURN. You will notice an inverse '+' sign in the prompt 'window' at the upper left corner of the screen. This indicates that VOICEMASTER is in the LEARNing mode and awaiting your input. (If neccessary, adjust the GAIN slightly to stabilize the inverse '+' sign.) Pause the LEARNing mode at this point by pressing the up arrow key '^'. Actual speech recording will now be suspended and an inverse up arrow, '^', will appear in the window. This feature comes in handy when you are recording in an enviroment with occasional loud background noise. When you are ready to resume recording your speech, just press the SPACEBAR, or any other key. Now speak into the microphone and notice how the window 'flickers' as you speak. Each 'flick' is a single bit of speech data being sampled. When you stop speaking the window will 'vanish' from the screen and the READY prompt will reappear. If the READY prompt reappears before you have finished speaking, then you have either paused too long between words and the program assumed you were finished speaking, or you exceeded the capacity of the speech input buffer (capacity is about 4 seconds for Atari 800's and about 7 sec. for XL's and XE's). So, don't pause too long between words and use phrases short enough to 'fit' within the capacity of the buffer. (Note: On Atari XE models, more time is needed to store speech in the extra 64K memory due to 'bank-switching' and an inverse 'W' will be seen in the window before the READY prompt reappears. This means, 'please wait'.) Type SPEAK 1 and press RETURN. The computer will now play back the recorded word. Be sure the volume on your T.V. or monitor is set high enough to hear the playback. Now, press the SPACEBAR (or any other key) during playback of the word and the playback routine will start over again.

If the SPACEBAR is rapidly and repeatedly 'tapped' during the playback, it will make an interesting 'reverb' sound effect. Because word #1 remains in memory, you can SPEAK 1 over and over again. Similarly, you can re-record word no.1 by entering the command LEARN 1 again, using either longer or shorter utterances.

The loudness of the recorded word can be varied by using the VOLUME command. First type VOLUME 7 and press RETURN. Then type SPEAK 1 and press RETURN. Notice the reduced loudness. Now restore the original volume by entering VOLUME 15.

SPEED is a command that is fun to work with. First, record a word or phrase with LEARN 1, and then change the playback speed by entering SPEED 1. Now enter SPEAK 1 and notice the slowed playback. Try entering SPEED 3 and then SPEAK 1. Finally, restore the normal default value by entering SPEED 2.

Saving a speech file is as simple as saving an Atari BASIC program. Use LEARN 1 to record a word or phrase to be saved, then SPEAK it back. Remove the VOICEMASTER disk from the drive and insert a blank formatted disk, to be used for storage of your speech files. Then type SSAVE"D:SPEECH" and press RETURN. Your recorded speech is now safely stored on disk. Type CLEAR and press RETURN to erase all speech stored in memory and then enter SPEAK 1. You will here a 'beep' signifying that there is now no recorded speech in memory. To load the speech file back into memory, just type SLOAD"D:SPEECH" and press RETURN. Enter SPEAK 1 and you will hear that the word you recorded and SSAVEd is back in memory where you can re-record it, if you like. The filename "D:SPEECH" was used above, but this could have been any filename of your choice, such as "D:MYWORD".

As previously stated, all Direct Mode commands can also be entered in the Program Mode as numbered program statements. This means that BASIC programs can be written which include speech recording and playback! And the variety of ways in which you can incorporate VOICEMASTER commands in your BASIC programs is limited only by your imagination.

USING THE CLEAR COMMAND

Before starting in on some actual examples in Program Mode, speech and program memory should be cleared. In order to erase your vocabulary from memory, simply type CLEAR.

On the Atari 800 and XL models, speech is stored in user-ram above 'ramtop'. The area below ramtop is available for the graphics display list, screen data, and your BASIC program. The amount of ram available for your BASIC program can be increased or decreased, according to your needs, by entering the 'CLEAR n' command. The variable 'n' can be any value between 48 and 160. This number specifies the 'page' in memory where speech data is stored. To maintain the graphics display, memory should be reserved in 4K increments. So, use values of 'n' that are multiples of 16, for example: CLEAR 64, CLEAR 96 etc. If the CLEAR statement without a value for 'n' is used, then the default value will be set at 48 for the Atari 800 and 64 for the 800XL; or whatever value was last assigned to 'n' with the 'CLEAR n' command. (Note: speech data on the Atari 130XE model is stored in the extra 64K bank of memory, and all of the ram that is normally available to BASIC is still available with the VOICEMASTER software loaded and running. Therefore, no value for 'n' is needed, nor permitted, with the CLEAR command for the 130XE.)

The CLEAR command erases speech data stored in memory, but does not affect a BASIC program. Likewise, the BASIC 'NEW' command erases BASIC program memory and variables, but does not erase recorded speech or affect the VOICEMASTER program in any way. Whenever you start a new recording session you may wish to CLEAR. This will ensure that leftover speech data from a previous recording session is erased. You may also want to NEW, so as to remove any statements left over from some previous BASIC program and also reset BASIC's pointers and variables.

VOICEMASTER COMMANDS IN PROGRAM MODE

The following program allows you to enter 12 words in Program Mode. Prompts to the screen are included. (If you can't think of some words, just count!)

```
10 FOR WORD=1 TO 12
20 PRINT "SAY WORD NO.";WORD
30 LEARN WORD
40 NEXT WORD
```

You can hear each word after recording it if you insert the statement:

```
35 SPEAK WORD
```

If you don't like any of the words learned, then you can replace them in direct mode by typing 'LEARN n' (with the appropriate value for 'n'). This will not affect any of the other recorded words.

Now erase the BASIC program with NEW, but do not CLEAR. The 12 words remain in memory. You can now use the same words in another program, in any order. Here is an example using a data statement to assign values to the 'SPEAK n' command.

```
10 RESTORE
20 READ WORD
30 SPEAK WORD
40 GOTO 20
50 DATA 5,4,10,1
```

Now RUN the program. An ERROR-6 (out of data error) will occur when the program has finished running. If words SPEAK back too close together in time, then insert this simple delay loop:

```
25 FOR T=1 TO 100:NEXT T.
```

Or use the 'PAUSE n' command, where 'n' is a number between 1 and 255 which specifies the number of 1/4 second increments in the pause (e.g. 25 PAUSE 2). Other commands like 'SPEED n', 'VOLUME n', and 'SLOAD "D:filename"' can also be included and the BASIC program saved in the same manner as a standard BASIC program.

```
*****************************************
                 PART III
        RECORDING AND PLAYBACK COMMANDS
*****************************************
```

This part of the manual describes those commands which
are used solely for speech reproduction. If you followed the
examples, given in the "GETTING FAMILIAR" section in PART II
of the manual, then you are already aquainted with most of
them.

Any sounds within the sensitivity range of the
VOICEMASTER unit can be recorded. This includes short
monosylabic utterances, words, phrases, whispers, bells,
whistles, humming, clapping, hammering, sawing and many more.
When speech (or any sound) is recorded with the 'LEARN n'
command, it is converted into digital data ('digitized') and
simultaneously stored in an area of memory designated as a
speech input 'buffer'. After the speech has been input, the
digitized data is then moved, byte by byte, into another part
of memory reserved for the storage of speech data. The
recorded speech is then ready to be played back at any time
with the 'SPEAK n' command.

                THE LEARN AND SPEAK COMMANDS

Words are numbered from 0 to 63 and up to 64 different
words or phrases can be in memory at one time. A word number
must be specified when using the LEARN or SPEAK commands.
'LEARN n' and 'SPEAK n' are the correct format, where 'n' can
be assigned any constant or variable that has a value within
the range indicated above. Each word number, 'n', functions
as an index to that particular word or phrase which was
recorded and stored in memory. After a word has been
LEARNed, it can be accessed again and again using the same
word number. If you don't like a particular word, then
simply re-record it using 'LEARN n'. Re-LEARNing a word or
phrase replaces the original one with the new one and the
memory space where speech data is stored is adjusted to
accomodate it.

The maximum length of a word that can be LEARNed is
limited by the capacity of the speech input buffer. If you
attempt to LEARN a word or phrase that is too long, or SPEAK
one that has not been previously recorded, then you will hear
a telltale 'beep'. You will also hear the beep if you have
used up all of the memory allocated for speech storage.

There are times when it is difficult to discern what caused the beep, especially when it occurs within a BASIC program. Therefore, VOICEMASTER uses memory LOCATION 209 as an error number register. This location is also used for other purposes. (see the table in APPENDIX II). A value of 253 indicates that, the speech input buffer capacity has been exceeded. A 249 means that a SPEAK command was given for a word number that had not been previously LEARNed. And a 252 means that there is no room left in the speech storage area to save any more speech data. Just enter: PRINT PEEK(209) to see the value returned here.

When recording a word with the LEARN command, the machine language recording routine takes control of the computer after you press RETURN. If you want to exit this routine prematurely, then press the ESCape key and this will return control to BASIC. If the ESCape key is pressed while in Program Mode, BASIC may not return control to the keyboard unless the BREAK key is also subsequently pressed. The ESCape key can also be used to terminate the SPEAK command. A 251 will be stored in LOCATION 209 if the ESCape key is pressed while in the LEARN mode and a 250 will be stored there if in the SPEAK mode.

THE VOLUME COMMAND

The loudness of a word or phrase can range from 0 to 15. The default value is set at 15, which is the maximum VOLUME setting. Use 'VOLUME n' to adjust the loudness of words or phrases that are played back with the 'SPEAK n' command. The 'VOLUME n' command can be used to produce some interesting sound effects, such as echoing a word or phrase during playback.

THE PAUSE COMMAND

This command acts exactly like a timing loop in BASIC (e.g. 10 FOR T=1 TO 10:NEXT T). However, 'PAUSE n' is easier to use. The PAUSE command produces a fixed timing delay of 1/4 second. The 'n' parameter specifies the number of quarter second increments to use. For example, PAUSE 20 will give a 5 second delay. For timing delays of less than 1/4 second, use a BASIC timing loop or a 'dummy' time wasting statement.

THE RATE COMMAND

When speech is recorded, the VOICEMASTER hardware samples both the frequency and amplitude of the incoming speech and converts this information into binary 'bits' of data. This process is known as digitization. After the speech has been digitized, it is stored in the computer's memory as a series of data bits. A single bit of data in memory represents a moment in time at which a sound pulse from the incoming speech was detected. If a pulse was detected, the bit will have a value of one, otherwise it will be zero. The number of times that the incoming speech is sampled, per given period of time, is called the sampling rate. The sampling rate at which the VOICEMASTER samples speech, in conjunction with the Atari, can range from under 4,000 bits per second to over 15,000 bits per second. Higher sampling rates reproduce speech with greater precision and somewhat better reproduction quality. But, this is at the cost of consuming greater amounts of memory to store the recorded speech. Lower sampling rates require less memory to store the speech data, but the reproduction quality is not as good as with the higher rates.

There are three sampling rates that can be used during the recording of speech. The 'RATE n' command can be used to select low, normal or high sampling rates. RATE 2 is the normal default setting and corresponds to an actual sampling rate of about 7,800 bits per second. This rate is adequate for most purposes. Use the RATE 1 command when memory space is critical. With RATE 1 you can store up to twice as much speech in memory as is possible with RATE 2. Using RATE 3 will give the best quality speech attainable with your Atari. However, this sampling rate is so fast that the 6502 microproccessor must record and playback speech without interference from the 'ANTIC' chip which 'steals' cycles from the 6502 to update the screen. For more information on ANTIC see: "De Re Atari" by Chris Crawford et al., Atari Program Exchange (Calif.:1982). To prevent ANTIC from stealing cycles, use the SCREEN command to turn off the screen during recording and playback of speech. (See below.)

The 'RATE n' command sets both recording and playback speed to the same value. If RATE 3 is used to LEARN a word or phrase, then SPEED 3 will be used to play it back. If a word or phrase is first LEARNed using RATE 2 and then another LEARNed at RATE 3, SPEED 3 will be selected by the RATE command as the default setting.

THE SPEED COMMAND

The command 'SPEED n', varies the playback rate of the recorded word or phrase, but has no effect on the recording rate. This allows you to SPEAK back words at rates slower or faster than the rate at which they were recorded. Five speed settings are available, ranging from 0 to 4, with 0 being the slowest and 4 the fastest. Playback SPEEDs 1,2 and 3 correspond to recording RATEs 1,2 and 3 and a word or phrase that is recorded at RATE 1 will sound normal when played back at SPEED 1. The default setting for the 'SPEED n' command is '2'.

THE SCREEN COMMAND

This command gives you the option of turning off the screen while LEARNing or SPEAKing. Use SCREEN 1 to turn off the screen during recording, SCREEN 2 to turn it off during playback, and SCREEN 3 to turn off it during word recognition (see PART IV). Once the screen has been turned off with the 'SCREEN n' command, it will remain off until restored to normal with SCREEN 0. Entering SCREEN 0, restores all screens to normal (on). As stated above, this command should be used with the the high sampling rate in order to get the best quality of speech reproduction.

THE PORT COMMAND

The VOICEMASTER unit can operate from either jostick port #1 or joystick port #2 by entering PORT 1 or PORT 2 as appropriate. This feature allows the VOICEMASTER unit to be used with other peripherals, which might only connect with port #2.

THE FREE COMMAND

There is not nearly as much memory available for speech data storage on the Atari 800 as there is with the 800XL and 130XE models. For this reason, a special command has been added exclusively for the Atari 800.

The FREE command allocates about 4K of additional memory space as an extension to the speech data storage area. Because this 4K of memory is normally used for storing recognition templates (more about this later), no speech recognition can be done after entering the FREE command unless system RESET is pressed first. There are no parameters needed for this command, just type FREE and press RETURN.

THE SSAVE AND SLOAD COMMANDS

A complete vocabulary can be saved on disk or tape in either Direct or Program Mode. When saved to disk, each speech vocabulary is saved with the particular filename that you assign to it. Thus, there is no theoretical limit to the number of vocabularies that can be saved to disk (or tape) and then loaded back for incorporation of speech within your BASIC program(s). The command to save a speech file to disk is, SSAVE"D:filename" and for loading use, SLOAD"D:filename". The 'filename' can be any name which is legal for the particular DOS that you are using. If using a cassette to store and load speech data, then use SSAVE"C:" and SLOAD"C:".

When speech data is SSAVEd, all pointers associated with the data are also saved. This means that when the speech is SLOADed back into memory, it will load back to the same area of memory that it was SSAVEd from and ramtop will be set accordingly. Also, all values for VOLUME, SCREEN, SPEED etc. will be restored to what they were when the speech file was SSAVEd.

```
*****************************
          PART IV
    RECOGNITION COMMANDS
*****************************
```

This part of the manual deals exclusively with speech
recognition and assumes that you have already familiarized
yourself with the previous part on speech reproduction. The
same principles of assimilation and digitization of data
during the recording process of speech reproduction are
equally applicable to the recording proccess used in speech
recognition. The main difference with speech recognition is
that the recorded speech data must be saved and then later
compared with new recorded data. When a word or phrase is
recorded for recognition purposes, only data which is
'characteristic' of that particular word or phrase is saved,
the rest is not. The part that is saved is called a
'template'. Because a template requires much less memory for
storage than does the actual recorded word or phrase, up to
32 templates can be stored in less than 4K of memory space.
Hence, there is room enough in just a 48K Atari for
utilization of both speech reproduction and recognition
within the same BASIC program.

                 THE TRAIN COMMAND

     Words or phrases, even short utterances and noises, can
be recorded and saved as templates for recognition. Due to
the time consuming calculatons that are needed to recognize
templates, it is impractical to make templates from words or
phrases that are too long in duration. For this reason, the
speech input buffer used for recognition has a limited data
capacity of two seconds which is the same for all Atari
models.

     To record data for recognition, the 'TRAIN n' command is
used. This command is analogous to the 'LEARN n' command and
is used in the same way. The parameter 'n' however, has a
range of only 0 to 31. To TRAIN a word or phrase just type
'TRAIN n', with the appropriate value for 'n' and press
RETURN. When you do this, an inverse question mark, '?',
will appear in the prompt window. As with the LEARN command,
you may pause the record routine at this point by pressing
the up arrow '^', or exit the routine with ESCape. Speak a
word or phrase into the microphone and it will be recorded
and saved as a template. For a word to be recognized, it
should be TRAINed twice -- once may not be sufficient.

                          -15-

When a word or phrase with the same template number, 'n', is re-TRAINed, its 'characteristics' are averaged with those of the previously saved template and a 'new', modified template is saved in the 'older' one's place. The method of 'dynamic time warping' is employed for this averaging procedure. An average can be acquired for any number of repetitions of the same word or phrase which has the same template number. However, TRAINing a word or phrase more than twice may not significantly improve its template and might actually degrade it.

THE BLANK COMMAND

There are times when you will need to erase a template from memory. The 'BLANK n' command will erase the specific template designated by its index number, 'n'. No other templates in memory will be erased if an index number is specified. If the BLANK command is entered alone, without a template number, then all templates stored in memory will be BLANKed. BLANK and CLEAR, without the index parameter, are analogous in their function.

THE RECOG COMMAND

Once the desired number of templates has been made with the 'TRAIN n' command, then words or phrases can be recognized. The RECOG command is used to record a word or phrase that is to be recognized. Simply type RECOG (without any parameters) and press RETURN. Again, as with the 'TRAIN n' command, an inverse question mark will appear in the prompt window. You can also pause or exit the routine. Now speak one of the words or phrases that was previously TRAINed. If the word or phrase is recognized, a value corresponding to the number of its template will be stored in memory LOCATION 209. Enter PRINT PEEK(209) to examine the contents of the location. If any number other than the correct template number for the word or phrase RECOGnized is found here, then the word or phrase was incorectly recognized or the computer refused to make an identification. See APPENDIX II for a complete table of values for this location.

The algorithm used to compare a template with a word or phrase to be recognized is quite complex. A considerable amount of microproccessor time is required to scan all the templates and find a comparison. Although it takes less than a second to scan all 32 templates, there are programming applications where a minimum amount of scanning time is desirable.

For this reason, the RECOG command can be used with a field of parameters that specifies which subgroup of the 32 templates is to be scanned.

The format for the command is 'RECOG a,b,c,d', where the variables a,b,c and d, are independently optional and can be assigned any value from 0 to 4. The 'group' of templates that will be scanned when a parameter is specified is as follows:

```
RECOG 0   (scans all 32 templates, same as RECOG)
RECOG 1   (scans templates 0-7 only)
RECOG 2   (scans templates 8-15 only)
RECOG 3   (scans templates 16-23 only)
RECOG 4   (scans templates 24-31 only)
```

One to four groups of templates can be scanned and in any order desirable. Some examples are:

RECOG 3,1 ;this command will scan templates 16-23 first and then scan templates 0-7.
RECOG 1,3,2 ;this scans templates 0-7, 16-23 and 8-15 respectively.
RECOG 4,3,2,1 ;this scans all groups of templates in reverse order.

Because a group consists of a minimum of 8 templates, all un-TRAINed templates in that group should be BLANK to prevent accidental false recognition. For example, if there are, say, 6 words which have TRAINed templates in memory, then the scan will cover two additional templates. These two templates should be BLANKed if they were previously used (TRAINed).

Many practical applications for recognition involve only a few words. If the chosen vocabulary has words that are quite different from one another, then the recognition accuracy will be better than if the words are similar. For example, the sounds for the letters 'D', 'E', 'P' are very similar and difficult to RECOGnize.

Accuracy can be further enhanced with a two step recognition process involving partitioning. In the first step, only a few distinctly different words are RECOGnized from a single group. In the second step, one of these RECOGnized words is used as a 'vector' to select yet another group of words to be RECOGnized.

In this manner, it is not neccessary to scan more than one group (8) of templates at a time. For example, assume we have 4 groups of templates of which one has 3 distinctly different words that function as vectors into the three other groups of templates. Now the first RECOGnition will select one of the other three groups for a second RECOGnition. Of the 24 total templates, we presume that some are TRAINed from similar sounding words and could possibly be confused. By not TRAINing more than one of these words within a single group of templates, we will have effectively reduced the RECOGnition error. One possible application for this sort of partitioning would be in designing programs that employ voice controlled 'pull-down' menus.

In addition to using template partitioning, the SCREEN 3 command can be used to gain a 30% increase in recognition speed. When the SCREEN 3 command has been entered, the screen will go blank while the templates are being compared. To restore the screen to normal, enter SCREEN 0.

THE MIN AND MAX COMMANDS

When a word or phrase is compared to a TRAINed template with the RECOG command, it is compared to all of the templates (within the selected template groups) and a numerical score is computed for it. This score is based on the number of dissimilar 'characteristics' found between the word and a given template. The smaller the computed value for the score, then the fewer are the dissimilarities and the closer the word or phrase matches the template. The score must lie below the maximum (nominal) threshold setting in order that the word be RECOGnized. The maximum threshold is preset at a default value of 320. This setting has a range of 0 to 700 and can be reset with the 'MAX n' command. If a word fails to be RECOGnized on the basis of the MAXimum threshold, then a value of 255 will be stored in LOCATION 209.

In addition, there is a minimun (differential) threshold that affects the matching of a template to a word or phrase. When a template is found which is the 'best match' within the MAXimum threshold limit, then its score is compared with the score of the template that is the 'next best' match. If the difference between these scores, for the two templates, is less than the value for the minimum threshold setting, then the RECOGnition will fail. For example, if two TRAINed words are very similar, like 'hat' and 'mat', then the RECOGed word 'at' would probably be minimally rejected.

The minimum threshold is preset at 30 and can **be reset** with the 'MIN n' command. If a word fails to be **recogniz**ed based on a MINimum threshold, then a value of 254 **will be** stored in LOCATION 209.

MAXimum and MINimum threshold settings can have a significant effect on recognition accuracy. Optimum values will depend on the particular speaker as well as the chosen vocabulary. Although recognition templates are generally restricted to use by the particular person who TRAINed them, some word sounds are more adaptable between different speakers than are others, and special 'multi-speaker' vocabularies can be constructed. However, multi-speaker speech recognition is still in the infancy stage and experimentation by the user is in order here.

THE TSAVE AND TLOAD COMMANDS

TRAINed templates can be saved to and loaded from disk or cassette tape with the TSAVE"D:filename" and TLOAD"D:filename" commands. As with speech files, there is no theoretical limit to the number of templates that can be TSAVEd for later retrieval and use by your BASIC program.

```
**************************************************
                     PART V
       LOADING AND USING THE VOICEMASTER PROGRAMS
**************************************************
```

In this part of the manual, each individual program on
the VOICEMASTER disk will be described in detail.
Additionally, instructions will be given on how to make a
'customized' disk containing only those programs that you
need for your particular application.

THE MOST IMPORTANT PROGRAM

The main VOICEMASTER program, hereafter referred to as
the 'MASTER' program, is the one that 'drives' the hardware,
analyzes and stores in memory speech and recognition data,
and performs all the supplementary data management chores.
It even manages the wedge which gives you easy access to its
functions via added BASIC commands. If this program is not
loaded into memory and running, then you will not be able to
record or recognize words and phrases. There are three
seperate versions of the MASTER program. These are:

```
            VM1.800   for the Atari 800
            VM1.XL    for the Atari 800XL
            VM1.XE    for the Atari 130XE
```

Each program performs all the same general functions as
the others, but differs from them in what memory locations it
uses to input and store speech and recognition data. For
example, the VM1.XE version stores speech data in the extra
64K bank of memory, the other two versions do not (see
APPENDIX I for more information on memory useage). This is
because the VM1.XE version was designed to take advantage of
the extra memory that is available on the Atari 130XE and
will not run on the Atari 800 or 800XL. Similarly, the
VM1.XL version will not run on the Atari 800.

However, the programs are 'upward' compatible and the
VM1.800 version of the program will run on the Atari 800XL
and Atari 130XE, as well as the Atari 800. Likewise, the
VM1.XL version will run on the Atari 130XE as well as the
Atari 800XL model.

Also, speech files created with the VM1.XE version and
SSAVEd cannot be SLOADed and played back with one of the
other two versions. But, like the MASTER program versions,
speech files are also upward compatible.

So, a speech file made with the VM1.800 version, for example, can be SLOADed and playback with the VM1.XL or VM1.XE versions.

On the other hand, recognition templates are not upward compatible and you cannot TLOAD and RECOG templates with the VM1.XE or VM1.XL versions that were TRAINed and TSAVEd with the VM1.800 version. Or vice-versa.

All three MASTER versions are binary files and require an associated 'loader' program to place them in memory. The loader programs are:

        VM800   for the VM1.800 version
        VMXL   for the VM1.XL version
        VMXE   for the VM1.XE version

These programs are written in BASIC and can be 'RUN' by entering at the READY prompt, one of the following appropriate commands:

        RUN"D:VM800"
        RUN"D:VMXL"
        RUN"D:VMXE"

When the loader program has finished doing its job, the corresponding MASTER program will be resident in memory, initialized, protected from system RESET, and ready for use. Since the program is 'reset proof', neither it or any other data, including a BASIC program, will be lost if you need to press RESET for some reason.

NOTE: Althogh pressing reset does not cause data to be lost, but, all of the MASTER program pointers, as well as BASIC's pointers (except ramtop), are restored to their default values.

Each version of the MASTER program reserves a specified amount of computer memory for BASIC programs. Enter FRE(0) to see how much memory is available for this purpose. If you need more memory, then use the 'CLEAR n' command as described in PART II of the manual. If an attempt is made to load a BASIC program that is too large to fit in the reserved memory, then an 'ERROR-19' will be returned. If this happens, it means that you must CLEAR more memory before the BASIC program will load.

IMPORTANT NOTE: If you are using the VM1.800 version, at least 64 pages must be CLEARed before accessing any DOS functions. Just enter CLEAR 64. (Note: the "MENU" program automatically does this for you when the 'EXIT TO BASIC' option is selected.)

THE "MENU" PROGRAM

This program is responsible for loading, running and allowing easy access to the various programs that are on the VOICEMASTER disk.

The "MENU" program automatically loads and runs (auto-boots) via an "AUTORUN.SYS" file. In turn, the "MENU" program loads in the MASTER program applicable to your model of Atari (800, XL or XE) and initializes it. Although convenient, use of the "MENU" program is not mandatory since all the programs, including "MENU" itself, can be loaded and 'RUN' independently. If you do not want the "MENU" program to auto-boot, then simply change the name of the "AUTORUN.SYS" file to "AUTORUN" according to the directions below:

1. Boot-up the copy you made of the VOICEMASTER disk according to the directions given in PART I of the manual.

2. From the VOICEMASTER MENU, select the 'EXIT TO BASIC' option and type DOS at the READY prompt, then press RETURN

3. When the DOS menu appears on screen:
select item 'E. RENAME FILE'.
when you see the prompt: 'RENAME - GIVE OLD NAME, NEW'.

answer by typing: 'AUTORUN.SYS,AUTORUN' and then press RETURN.

This will change the name of the file to "AUTORUN", which becomes inactive without the '.SYS' extender. When the computer boots-up into BASIC, control will be given over to you instead of the "MENU" program. The "MENU" program can still be used, at any time, by entering RUN"D:MENU". To restore the auto-boot function, just rename the file back to "AUTORUN.SYS".

NOTE: If the VM1.800 or VM1.XL version of the MASTER program is resident in memory, then be sure that at least 64 pages (CLEAR 64) have been CLEARed or you may get an 'ERROR-19' when you attempt to LOAD this program.

THE "BAR.COM" PROGRAM

This is a binary file that can be accessed from menu or loaded directly from DOS. The program is designed to work either independently from, or in conjunction with, the MASTER program.

Follow the steps below to load the "BAR.COM" program only when the MASTER program is not in memory:

1. Boot-up the copy you made of the VOICEMASTER disk according to the directions given in PART I and select the 'EXIT TO BASIC' option from the menu.

2. Type 'POKE 106,144:GRAPHICS 0' and press RETURN.

3. Type 'DOS' at the READY prompt, then press RETURN

4. When the DOS menu appears on screen:
select item 'L. BINARY LOAD'.
when you see the prompt: 'LOAD FROM WHAT FILE?'.
answer by typing: 'BAR.COM' and then press RETURN.

If you want to load "BAR.COM" with either the VM1.800 or VM1.XL MASTER program resident in memory, then follow the above procedure but skip step #2. If you have the VM1.XE version resident, then do not skip step #2 and follow the above steps as shown.

When loaded, the "BAR.COM" program occupies the area of memory from decimal address 39936 to address 40959, which happens to be the same area of memory occupied by the Atari's display list and screen display data. Step #2 above, lowers the value of ramtop (location #106) to 144 and moves the display list and screen data below the new value so that the area above 39864 will be clear for loading in the "BAR.COM" program.

A complete discription of the bar-display was given in PART I of the manual, please refer to it if neccessary.

To exit the "BAR.COM" program and return to DOS, just press the SPACEBAR or any other key. If you want to return to BASIC from DOS, then select item: 'B. RUN CARTRIDGE'. From BASIC you can restart the "BAR.COM" program again by entering BAR=USR(39936).
NOTE 1: Pressing system RESET will cause the 'BAR.COM' program to be 'wiped out'. If this happens, you will need to re-load it.
NOTE 2: Unlike the MASTER program which can be configured to use either joystick port #1 or port #2, the "BAR.COM" program will only work when the VOICEMASTER unit is plugged into port #2.

THE "LOADPLAY" PROGRAM

One of the many good reasons for owning a VOICEMASTER is that you now have the capabillity to add speech to any of your BASIC programs. And best of all, you do not need to have the VOICEMASTER hardware 'hooked-up' to playback the speech. All you need is this little (about 500 bytes) "LOADPLAY" program and the sound chip ('POKEY') that is already built into your Atari. The "LOADPLAY" program contains data that was aquired from a 'relocatable', binary file that could load and playback speech files created with the MASTER program. When the program is initialized, the binary data in the BASIC DATA statements is READ into a string variable called 'ML$', which is stored in BASIC program memory. The first memory location of 'ML$' is the beginning of the binary data for the Machine Language routine.

Because "LOADPLAY" is written in BASIC, it can easily be 'merged' with your BASIC program as follows:

1. Before merging the two programs, be sure that your BASIC program does not use line numbers above 31999.

2. Boot-up the disk containing the BASIC program that you want to add speech to. Then LOAD (or ENTER) your BASIC program and remove the disk.

3. Insert the disk with the "LOADPLAY" program on it.

4. Type: ENTER"D:LOADPLAY" and press RETURN. Then remove the disk.

5. Insert the disk on which you want to save the merged program and SAVE it with a filename of your choice.

To use the "LOADPLAY" program, you must first create a speech file with the MASTER program and SSAVE it to disk with the filename, "D:SPEECH". Of course, other filenames can be used and you may access more than one speech file with the program. Just LIST the "LOADPLAY" program to the screen and make the appropriate changes. When making the speech file, do not use the VM1.XE MASTER program or "LOADPLAY" will not be able to SLOAD or playback the speech file. This is because "LOADPLAY" cannot access the extra 64K bank on an Atari 130XE.

You must use either the VM1.800 or the VM1.XL version of the MASTER program to make the speech file. Therefore, if you are adding speech to a BASIC program that you want to work with any model of Atari (a commercial application perhaps), then you should create the speech file with the VM1.800 version.

For further instructions on using the "LOADPLAY" program, ENTER and LIST the program to the screen and read the REM statements.

NOTE: When the program is 'RUN', it takes a minute or so to initialize before it begins loading the "D:SPEECH" file.

WARNING: "LOADPLAY" is somewhat sophisticated for a BASIC program and you may have trouble understanding or using it if you are a neophyte to the BASIC language.

THE "DEMO" PROGRAM

The purpose of this program is to introduce a newcomer to the basic features of the VOICEMASTER. It includes a speech recording and playback demonstration with options to vary the speech playback rate and speech volume (echo effect). A simple speech recognition demonstration and a speech bar-graph ("BAR.COM") program are also included. The "DEMO" program can be ran by entering, RUN"D:DEMO", or selecting it from the main menu.

NOTE: The MASTER program must be resident in memory to run this program. See also the note for the "MENU" program.

THE "CALCULATOR" PROGRAM

This BASIC program turns your computer into a 'talking calculator' which recognizes voice input. It is a four function (+,-,*,/) calculator, accurate to the eighth decimal place. To load the program, enter:

RUN:"D:CALCULATOR".

After the program loads, you will be prompted to enter a vocabulary for the recognition templates and another one for the playback words used by the program. Happy calculating!

NOTE: See the note for the "DEMO" program.

THE "CLOCK" PROGRAM

With this program, you get a clock that actually talks the time, there is even an alarm mode so your Atari can wake you up in the morning.

Run the program by entering, 'RUN"D:CLOCK" and when the introduction screen appears, press any key to get to the CLOCK MENU. Select the 'Voice Creator' option from the menu and the program will prompt you to enter a speech vocabulary. After you have done this, the program will enter the EDIT MODE. From this mode, you can SPEAK back and/or re-LEARN any individual word that was previously entered. When you have finished editing, you can save the vocabulary with the 'Save Clock Voice' option. If you decide to save the vocabulary, then you will not have to create another clock voice the next time you run the program. When the program finally returns to the CLOCK MENU, select the 'Set Clock Time' option and follow the prompts.

The digital clock should now be displayed on the screen and will keep accurate time until you have the clock SPEAK. That's right, the "CLOCK" program does not keep track of time when it SPEAKs. Why not? Because the Atari uses a 'vertical blank' interupt to update its system timers and this interupt is disabled whenever the VOICEMASTER is recording or playing back speech. Now there is a solution to this 'problem', but it will require some experimentation and BASIC programming on your part. If you feel up to the challenge, then read the section in APPENDIX I on: "FIXING THE CLOCK".
NOTE: See the note for the "DEMO" program.

THE "BLACKJACK" PROGRAM

"Welcome folks, to the VOICEMASTER talking 'BLACKJACK' game! Just step right up and place your bets please...here's your oppurtunity to 'hit the big one' and the 'house gives you twenty bucks to get started with!" Want to play? The rules are simple:

1. With the MASTER program resident, enter: RUN"D:BLACKJACK" (See the note below). Or, select the program from the main VOICEMASTER MENU
2. Follow the prompts and speak, very clearly and distinctly, the words that the program requests you to say. When all the templates have been made, you can start playing the game.
3. The minimum bet allowed is $2. When asked to place your bet, enter it one number at a time. If you make a mistake, then say: 'ERASE' and enter the numbers again. When you have finished entering your bet, say: 'BET'. The cards will then be dealt out.

4. The object of the game is to get a higher value 'hand' then the dealer's without exceeding ('busting') a total value of 21. Face cards count 10 and Aces count 1 or 11, as you please. Other cards count their face value. A Blackjack is any face card plus an ace.

5. When both hands have been dealt, you will be prompted to make a choice. To get another card, say: 'HIT ME'. Keep taking 'hits' until you think that your 'hand' is higher then the dealer's 'hand' and then say: 'STAND'. Sometimes you will not need to take any 'hits' at all, like when you get dealt a Blackjack. When this happens then just 'STAND' on it.

6. Winning hands pay 2 to 1 and Blackjack pays 3 to 1. A 'push' (both 'hands' equal) pays 1 to 1 (which means that you don't lose your bet). Also, if you get more than five cards in your hand (a 'five-card Charlie'), without 'busting', then you win.

7. The dealer must play 'house rules'. That is, he (your Atari) must 'hit' a 'hand' which equals 16 or less. He must stand on a 'hand' which is 17 or more, unless it's a 'soft-seventeen' (a 'hand' with an ace that counts 11).

If at any time during the game you want to see exactly which cards have been played, then say: 'CARDS' and you will get a display of the played cards on screen. Say, 'CARDS', again to continue with the game. When you eventualy run out of money (the Atari always wins), then you can answer the 'Play Again?' prompt with a 'yes' to play again, or a 'no' to return to the main VOICEMASTER MENU.

NOTE: If the VM1.800 or VM1.XL version of the MASTER program is resident in memory, then be sure that at least 70 pages ('CLEAR 70') have been CLEARed or you will get an 'ERROR-19' when you attempt to LOAD this program.

THE "BJVOICE" PROGRAM

If you get tired of listening to the same old dealer's voice in the "BLACKJACK" game, then you can easily make another with this short BASIC program. With the MASTER program resident, enter: RUN"D:BJVOICE" and answer all the prompts. When you have finished entering the vocabulary, the program goes to the EDIT MODE where you can 'clean-up' any mis-LEARNed words. This edit mode is similar to the one used in the talking "CLOCK" program.

When you have finished editing the speech, the program
will SSAVE the file to disk with the filename: "DEALER.SPE".
Be sure that the file is SSAVEd to a disk which has a copy of
the "BLACKJACK" program on it.

THE "COMPOSER" PROGRAMS

There are two programs on the disk which will not be
described here. They are: "COMPOSER" and "COMP1.COM". These
files constitute the famous 'VOICEMASTER MUSIC COMPOSER'
programs and are thoroughly explained in the manual dedicated
to them.

THE BASIC PROGRAMS IN GENERAL

All of the BASIC programs described here can be LISTed
to the screen and studied, so that you can 'learn by example'
how to incorporate the VOICEMASTER commands in your own BASIC
programs. You may even want to adopt some of the more useful
subroutines like the vocabulary editor in "BJVOICE" or the
"BAR.COM" loader found in the "DEMO" or "MENU" programs.
Also, some of these programs could be edited and improved.
The "CLOCK" program for example. So, feel free to edit,
modify and customize them to suit your fancy.

ATARI DOS 2.5 FILES

There are three files (programs) which are part of
"ATARI DOS 2.5", these are:

        "DOS.SYS"   the Disk Operating System
        "DUP.SYS"   the Disk Utility Package
        "RAMDISK"   a program for the ATARI 130XE

There is a manual available that completly explains the
use of these, and other, DOS 2.5 files. The "ATARI DOS 2.5
OWNERS MANUAL" is available from:

                ATARI Customer Relations
                P.O. Box 61657
                Sunnyvale, CA 94008

The price is $10.00 plus $2.50 postage.

HOW TO MAKE A CUSTOMIZED VOICEMASTER disk

As shipped, the VOICEMASTER disk is 'stuffed' full, with no room left for saving your BASIC programs and speech files. Of course, you could SAVE or SSAVE everything to a seperate disk, but this tends to be inconvenient. Especially if you only have one disk drive. The solution is to make a customized version of the VOICEMASTER disk, which only has on it those programs that you need.

If you want to make a customized disk for developing BASIC programs which include speech recording, then you must make a copy of one of the versions of the MASTER program along with its affiliated loader program. For example, if you have an ATARI 800, you will want the following programs on your customized programming disk:

        "DOS.SYS" the Disk Operating System
        "DUP.SYS" the Disk Utility Package
        "VM1.800" the MASTER program
        "VM800"   the affilliated loader program.
To make this customized disk, follow these instructions:

1. Boot-up the copy you made of the VOICEMASTER disk according to the directions given in PART I of the manual.

2. From the 'VOICEMASTER MENU', select the 'EXIT TO BASIC' option and type DOS at the READY prompt, then press RETURN

3. Remove the VOICEMASTER disk and insert a blank disk.

4. From the DOS menu, select item: 'P. FORMAT SINGLE'. when you see the prompt: 'WHICH DRIVE TO FORMAT?' answer by typing: '1' and then press RETURN.

when you see the prompt: 'TYPE "Y" TO FORMAT DISK 1' answer by typing: 'Y' and then press RETURN.

5. From the DOS menu, select item: 'H. WRITE DOS FILES' when you see the prompt: 'DRIVE TO WRITE DOS FILES TO? answer by typing: '1' and then press RETURN

when you see the prompt: 'TYPE "Y" TO WRITE DOS TO DRIVE 1.' answer by typing: 'Y' and then press RETURN. This will write the DOS.SYS and DUP.SYS files to the disk.

6. Next, select from the DOS menu:
item 'O. DUPLICATE FILE'
when you see the prompt:'NAME OF FILE TO MOVE?'
answer by typing: the name of a VOICEMASTER file that you
want copied on to the new disk.

when you see the prompt: 'INSERT SOURCE DISK,TYPE RETURN'
answer by inserting the VOICEMASTER disk, then pressing
RETURN.

when you see the prompt:
'INSERT DESTINATION DISK,TYPE RETURN'
answer by removing the VOICEMASTER disk and inserting the
blank, formatted disk, then pressing RETURN. Repeat this
step for each file that you want to put on your customized
disk.

If you have an ATARI 130XE, then you have the option of
using the extra 64K bank of memory for either speech storage
or as a 'ramdisk'. To use the ramdisk, make a copy of the
VM1.XL version of the MASTER program instead of the VM1.XE
version and include a copy of the "RAMDISK" file. Rename the
"RAMDISK" file to, "RAMDISK.COM" in order to activate it. To
rename the "RAMDISK" file, just follow the same instructions
given for renaming the "AUTORUN.SYS" file.

*** APPENDIX I ***

MEMORY USEAGE OF
SPEECH AND RECOGNITION FILES

SPEECH DATA FILES

Speech data alone would be useless without an orderly method of storing and retrieving it. The MASTER program uses the first 330 memory locations to store 'pointers' for the actual speech data. These pointers, along with the speech data, constitute a 'speech file'. The very first of these 330 memory locations is known as the 'BASE' address. The BASE is set to decimal address 16384 ($4000) by default and can be re-set to another memory location by use of the 'CLEAR n' command. The method for determining the BASE address depends on which version of the MASTER program you are using.

If you are using the VM1.800 version, the BASE address is: BASE=(PEEK(106)+16)*256
In order to accomodate the 4K speech input buffer, which is below the speech storage area, 16 pages must be added to the value in location 106 (ramtop).

If you are using the VM1.XL version, the BASE address is: BASE=PEEK(106)*256
In this case, the speech storage area begins at the BASE address.

If you are using the VM1.XE version, the BASE address is: BASE=64*256
This is always the case because the speech file is stored in the extra 64K bank and must be accessed through the 'window' located from 16384 ($4000) to 32767 ($7FFF). See pp. 121-122 of your "Atari 130XE Owners Manual."

1. MEMORY LOCATIONS FROM BASE TO BASE+255
These first 256 bytes of memory define the starting and ending address of where a particular word or phrase is stored in memory. Two bytes are required to define the starting address of a word or phrase and two more for the ending address. So, the first two locations, BASE+0 and BASE+1, correspond to the 'low' and 'high' order bytes, respectively, for the starting address of the first word or phrase (word index #0); and the next two consecutive memory locations, BASE+2 and BASE+3, define the ending address. Therefore, for each recorded word or phrase, a total of four memory locations is needed to delimit the area of memory that is occupied by the actual speech data. To find the location of the address pointers for word 'n': multiply 'n' times 4 and add the product to the BASE address.

Then use the PEEK(X) statement to determine the location in memory of the actual speech data. The following BASIC program exemplifies this procedure:

```
10 PRINT"ENTER WORD (INDEX) NUMBER";
20 INPUT WORD
30 BASE=16384 :REM The default value is used here
40 POINTER=BASE+(WORD*4)
50 POKE 54017,225 :REM Switches 'on' extra 64K
60 START=PEEK(POINTER)+256*PEEK(POINTER+1)
70 END=PEEK(POINTER+2)+256*(POINTER+3)
80 POKE 54017,253 :REM Switches 'off' extra 64K
90 PRINT "START=";START,"END=";END
```

NOTE: This program, and others in this appendix, work the same for any model Atari. If you are not using a 130XE, then line numbers 40 and 70 (above) will have no effect. Also, it is assumed that the default BASE address has not been changed with the 'CLEAR n' command. If the BASE has been changed, then adjust line No. 20 appropriately.

2. MEMORY LOCATIONS BASE+256 AND BASE+257

These two memory locations define the address of the end (or top), of the entire speech file. The first location is the low order byte and the second location is the high order byte of the address.

To find out how much total memory the speech file occupies, use the following program:

```
10 POKE 54017,225
20 BASE=16384
30 TOPMEM=BASE+256
40 PRINT (PEEK(TOPMEM)+256*PEEK(TOPMEM+1)-BASE)
50 POKE 54017,253
```

NOTE: When a speech file is SSAVEd, all speech data, from BASE to the address specified by TOPMEM, is saved to disk or tape.

3. MEMORY LOCATION BASE+259

This location contains the count of the total number of words or phrases that have been LEARNed (recorded).

4. MEMORY LOCATION BASE+260
This location contains the number which determines the length
of the shortest phrase that can be recorded. The value in
this location depends on the sampling rate being used:

                10 at the normal rate.
                5 at the low rate.
                20 at the high rate.

Poking a smaller value to this location will allow very short
utterances, such as individual phonemes, to be recorded. Be
careful: do not set this value too low or breathing, lip
noises and clicks will start the recording process and
prevent it from terminating at the intended end of the speech
sound. If carefully done, very short speech intervals can be
utilized.

    5. MEMORY LOCATION BASE+264
This is the 'flag' location for the playback screen. A value
of 34 indicates the normal 'screen on' condition. A zero
indicates the screen will be off during playback.

    6. MEMORY LOCATION BASE+265
This is the 'flag' location for the recording rate. Three
recording rates are available and the values found in this
location correspond to the rates as follows:

            VALUE   RATE
              1   15,630 Khz
              2   7,813 Khz
              4   3,906 Khz

    7. MEMORY LOCATION BASE+266
This is the 'flag' location for the playback rate. Five
playback rates are available and the values found in this
location correspond to the rates as follows:

            VALUE   RATE
              0   30,000 Khz (Approx.)
              1   15,630 Khz
              2   7,813 Khz
              3   5,208 Khz
              4   3,906 Khz

8. MEMORY LOCATIONS BASE+267 to BASE+330

These 64 bytes are an index of the order which the words or phrases have been recorded in. For example, if you LEARNed phrases 3,8,2,0,40 and 23, in that order, then the first six memory locations, beginning at BASE+267, will contain 3,8,2,0,40 and 23, respectively. BASE+267 will be 3, BASE+268 will be 8 and so on. The following BASIC program checks these pointers to see if a particular word has been recorded:

```
10 POKE 54017,225
20 BASE=16384
30 PRINT"ENTER WORD INDEX NUMBER FOR SEARCH";
40 INPUT WORDNUM
50 WRORCNT=PEEK(BASE+259)
60 FOR X=0 TO WORDCNT-1
70 ORDERNUM=PEEK(BASE+267+X)
80 IF ORDERNUM=WORDNUM THEN PRINT"FOUND!":GOTO 110
90 NEXT X
100 PRINT"NOT FOUND."
110 END
```

9. MEMORY LOCATIONS BASE+331 TO END OF SPEECH FILE

Here is where the actual speech data bits are stored. The amount of memory available for storage of speech data depends on the version of the MASTER program in use.

If using the VM1.800 version, a maximum length speech file will occupy memory locations BASE+331 to 30463 ($76FF). If the FREE command was entered, then the end of the speech file is extended to 34559 ($86FF).

If using the VM1.XL version, a maximum length speech file will occupy memory locations BASE+331 to 39935 ($9BFF).

With the VM1.XE version, the speech file is actually stored in the extra 64K bank which has an address range of 0 ($0000) to 65,280 ($FFFF). However, the speech file can only be examined through the 16K 'window' within the address range 16384 ($4000) to 32767 ($7FFF). To examine the speech file pointers and the beginning of the speech data, first POKE location 54017 with 225. This will expose the first 16K block of the 64K extra bank. To expose the second, third and fourth 16K blocks, POKE 54017 with 229, 233 and 237 respectively. In this manner the entire speech file can be examined. Remember to POKE 54017 with 253 when you have finished examining the file. This will restore the 'window' to its default value.

RECOGNITION DATA FILES

A recognition data file consists of file pointers and 32 word templates. Unlike the speech file pointers, the template pointers are not user accessible and the MASTER program uses them for 'book-keeping' purposes only. Since each template is 96 bytes long, all 32 templates require a total of 3,072 bytes of memory for storage. A total of 4,096 bytes (4K) are required to store both pointers and templates. When a recognition file is TSAVEd to disk (or tape), all pointers and templates are saved, regardless of whether some of the templates were un-TRAINed.

If using the VM1.800 version of the MASTER program, the recognition file will occupy the area of memory from 30464 ($7700) to 34559 ($86FF).

If using the VM1.XL or VM1.XE version of the MASTER program, the recognition templates are stored behind the 'Operating System' ROM and occupy the area of memory from 49152 ($C000) to 53247 ($CFFF). Templates made from these two versions of the MASTER program are identical and can be used interchangeably with one another, but not with the VM1.800 version.

FIXING THE CLOCK

By using the address values of stored words, found at BASE to BASE+255, it is possible to program the 'talking clock' to keep track of time when it SPEAKs. To do so, you will need to measure, exactly, how much time it takes to entirely fill up the speech input buffer. Enter LEARN 0 and fill the buffer to capacity and record the time with a stopwatch. Do this several times and then average the results (TIME=calculated average). Next, examine the starting and ending addresses of word #0 and calculate the length in bytes (LENGTH=total bytes). Now calculate the RATIO: TIME/LENGTH.

Examine the "CLOCK" program and find the program lines where the 'SPEAK n' commands are. Add a subroutine which calculates the length of a vocabulary word from its corresponding starting and ending addresses. Once the length has been calculated, it can be multiplied by the RATIO to give the time in seconds. Add this time to the clock timer routine. As an exercise, you might add recognition so that you can ask for the time or other information by voice.

-35-

*** APPENDIX II ***

VALUES ASSOCIATED WITH MEMORY LOCATION #209

VALUE   EXPLANATION

0-31 The recognition template numbers for recognized words.

128   The default value when no error or recognition has occurred.

249   The 'SPEAK n' command was given for an unLEARNed word.

250   The playback routine was exited prematurely by pressing the ESCape key during a 'SPEAK n' command.

251   The record routine was exited prematurely by pressing the ESCape key during a 'LEARN n', 'TRAIN n', or 'RECOG' command.

252   The speech storage area is full. No more new words can be LEARNed.

253   The speech input buffer was exceeded during the previous 'LEARN n', 'TRAIN n', or 'RECOG' command.

254   The word was not RECOGnized due to MINimal rejection.

255   The word was not RECOGnized due to MAXimal rejection.

VOICEMASTER MEMORY USEAGE MAPS

MAP FOR VM1.800 MASTER PROGRAM

```
                                            65280 ($FFFF)
 _____
| 24K    OPERATING SYSTEM              |
|         AND BASIC ROM                |
|THIS AREA IS NOT USED BY VM1.800|     |
|                                      |  40960 ($A000)
|_____|
| 1K      FREE RAM SPACE               |
|    ('BARGRAPH' loads here.)          |
|                                      |  39936 ($9C00)
|_____|
| 5K+     MASTER PROGRAM               |
|         VM1.800                      |
|                                      |  34560 ($8700)
|_____|
| 4K    RECOGNITION DATA FILE          |
|                                      |  30464 ($7700)
|_____|
| 14K+   SPEECH DATA FILE              |
|  (size varies with CLEAR cmd.)       |
| - - - - - - - - - - - - - - -        | -16714 ($414B)
|    POINTERS FOR SPEECH FILE          |
|                                      |  16384 ($4000)
|_____|
| 4K    SPEECH INPUT BUFFER            |
|                                      |  12288 ($3000)
| DISPLAY LIST AND SCREEN DATA         |
|         960 BYTES                    |
|                                      |  11264 ($2C00)
|_____|
| 3.5K+   FREE RAM SPACE               |
|  FOR BASIC AND BASIC PROGRAMS        |
|                                      |  7420 ($1CFC)
|_____|
| 5.5K+   DOS IF PRESENT               |
|                                      |  1792 ($700)
|_____|
|     PROGRAM JUMP TABLE               |
|                                      |  1716 ($680)
|_____|
|      FREE RAM SPACE                  |
|                                      |  1536 ($600)
|_____|
|   STACK USE AT LOCATION 256          |
|                                      |  256 ($100)
|_____|
| PROGRAM ZP FROM 207 TO 255           |
| - - - - - - - - - - - - - - -        | -207 ($CF)
| FREE ZP FROM 203 THROUGH 206         |
| - - - - - - - - - - - - - - -        | -(203) ($CB)
|      OS ZERO PAGE                    |
|                                      |  0
|_____|
```

MAP FOR VM1.XL MASTER PROGRAM

```
                                          _65280 ($FFFF)
| 7K     SPEECH INPUT BUFFER          |
|                                     |
|                                     |_58368 ($E400)
| 1K     CHARACTER SET                |
|       (COPIED FROM ROM)             |
|                                     |_57344 ($E000)
| 2K       FREE RAM SPACE             |
|                                     |_55296 ($D800)
| 2K     HARDWARE REGISTERS           |
|                                     |_53248 ($D000)
| 4K     RECOGNITION DATA FILE        |
|                                     |_49152 ($C000)
| 8K     RAM BEHIND BASIC ROM         |
|       VM1.XL MASTER PROGRAM (7K)    |
|           RESIDES HERE              |
|                                     |_40960 ($A000)
| 1K     FREE RAM SPACE               |
|       ('BARGRAPH' loads here.)      |
|                                     |_39936 ($9C00)
| 23K+   SPEECH DATA FILE             |
|  (size varies with CLEAR cmd.)      |
| - - - - - - - - - - - - - - -       |-16714 ($414B)
|       POINTERS FOR SPEECH FILE      |
|                                     |_16384 ($4000)
|   DISPLAY LIST AND SCREEN DATA      |
|           960 BYTES                 |
|                                     |_11264 ($3C00
| 7.5K+  FREE RAM SPACE               |
|   FOR BASIC AND BASIC PROGRAMS      |
|                                     |_7420 ($1CFC)
| 5.5K+  DOS IF PRESENT               |
|                                     |_1792 ($700)
|       PROGRAM JUMP TABLE            |
|                                     |_1716 ($680)
|       FREE RAM SPACE                |
|                                     |_1536 ($600)
|   STACK USE FROM 256 TO 350         |
|                                     |_256 ($100)
|   PROGRAM ZP FROM 207 TO 255        |
| - - - - - - - - - - - - - - -       |-207 ($CF)
|   FREE ZP FROM 203 THROUGH 206      |
| - - - - - - - - - - - - - - -       |-(203) ($CB)
|           OS ZERO PAGE              |
|_____|_0
```

```
                                           65280 ($FFFF)
 _____
| 16K SEE THE VM1.XL MAP FOR         |
|      USEAGE OF THIS AREA           |
|                                    |  49152 ($C000)
|_8K____RAM_BEHIND_BASIC_ROM_____|
|    VM1.XE MASTER PROGRAM (7K)      |
|         RESIDES HERE               |
|                                    |  40960 ($A000)
|_DISPLAY_LIST_AND_SCREEN_DATA_____|
|          960 BYTES                 |
|     ('BARGRAPH' loads here)        |
|                                    |  39936 ($9C00)
|_32K+___FREE_RAM_SPACE_____|
|  FOR BASIC AND BASIC PROGRAMS      |
|                                    |  7420 ($1CFC)
|_5.5K+__DOS_IF_PRESENT_____|
|                                    |  1792 ($700)
|      PROGRAM JUMP TABLE            |
|                                    |  1716 ($680)
|_____FREE_RAM_SPACE_____|
|                                    |  1536 ($600)
|_STACK_USE_FROM_256_TO_350_____|
|                                    |  256 ($100)
|_PROGRAM_ZP_FROM_207_TO_255_____|
| - - - - - - - - - - - - - - - -    | -207 ($CF)
| FREE ZP FROM 203 THROUGH 206       |
| - - - - - - - - - - - - - - - -    | -(203) ($CB)
|        OS ZERO PAGE                |
|_____|  0
```

```
64K SPEECH DATA FILE IN EXTRA BANK
                                           65280 ($FFFF)
 _____
| BANK4 16K SPEECH DATA      (237)   |
|_____|  49152 ($C000)
| BANK3 16K SPEECH DATA      (233)   |
|_____|  32768 ($8000)
| BANK2 16K SPEECH DATA      (229)   |
|_____|  16384 ($4000)
| BANK1 15.5K SPEECH DATA    (225)   |
| - - - - - - - - - - - - - - - -    | -330 ($14B)
|     SPEECH FILE POINTERS           |
|_____|  0
```

EXPLANATION OF MAPS

ZERO PAGE: All versions use locations 207 through 209 and these locations are not available to the user. Locations 211 through 255 are 'shared' with BASIC and available for user application, but any data stored here will be 'wiped-out' when BASIC or the MASTER program accesses them. Locations 203 through 206 are not used by either BASIC or the MASTER program, and are 'free' for user application.

STACK: Location 256 is used to store the MASTER program version identification number. The value in this location, after program initialization, will be:

>   1   for VM1.800 version
>   2   for VM1.XL version
>   3   for VM1.XE version

Versions VM1.XL and VM1.XE use an additional 93 bytes (from 256 to 350) to store the ROM switching routine that allows access to the RAM behind the BASIC and O.S. ROMs. During normal operations, the stack should never go low enough to interfere with the data stored here. Do not mess with this area!

PAGE 6: The upper half of 'page 6' (128 bytes) is used to store a 'jump table' for the command routines. The table is only neccessary for operation of the wedge. If you absolutely need to use this area of RAM, then you can bypass the table with direct jumps to the command routines via the USR(X) statement. Careful examination of this area will reveal where the command routines are located in the MASTER program. The lower half of 'page 6' is free RAM for user application.

MEMORY FROM 7420 TO 40960: The 1K area of memory, from 39936 to 40960, is ideal for PM graphics and character set storage (provided that the 'BARGRAPH' program is not used). Other details about these areas of memory have been given elsewhere in the manual. See, particularly, APPENDIX I.

BEHIND BASIC and O.S. ROM: On the Atari 800XL and 800XE, this area (40960 to 65280) is available as free RAM via 'ROM switching' techniques. The VM1.XL and VM1.XE MASTER programs are located 'behind' the BASIC ROM and accessed with the wedged commands via the routines on the STACK and PAGE 6.

The complete Atari character set has been duplicated in the RAM from 57344 to 58368. This was neccessary to preserve the screen display during O.S. ROM switching. If you want to use a custom character set with the MASTER programs, then you will have to copy it to this area of memory, as well as to the main RAM area. The RAM area from 55296 to 57344 is free for user application.

NOTES ON THE VOICEMASTER WEDGE

The wedge used in the VOICEMASTER program converts a command word like LEARN or SPEAK into a BASIC 'USR(X)' statement. For instance, when the command LEARN 5 is entered, the wedge will return 'LEARN=USR(1670,5)'. BASIC interprets 'LEARN' as a variable which has a value equal to the decimal address (LEARN=1670) of the USR(X) routine. The addresses 1670 and 1670+1 on the VM1.800 version of the MASTER program (1672 and 1672+1 on XL and XE versions) contain the low and high bytes of the address where the actual 'LEARN n' routine begins. If you want to see how the wedge works, just type a few BASIC program lines containing VOICEMASTER commands and then LIST the program.

Because of the way that the wedge checks for correct syntax of wedged VOICEMASTER commands, there are some minor restrictions that must be observed. When entering VOICEMASTER commands in BASIC program lines:

1. The VOICEMASTER command must immediately follow the colon in multiple statement program lines. For example:
        10 FOR X=1 TO 10:LEARN X:NEXT X
This is acceptable to the wedge. However:
        10 FOR X=1 to 10: LEARN X:NEXT X
is not acceptable because of the space after the colon and the wedge will not convert the LEARN X command to the USR(X) form.

2. To use a VOICEMASTER command with an 'IF THEN' statement on the same program line, the command must be preceeded with an ':' and a '?' must follow the 'THEN'. Enter the line in this manner:
        10 IF X=10 THEN ?:SPEAK X
Then LIST the line and the SPEAK X will be converted to the USR(X) form:
        10 IF X=10 THEN ?:SPEAK=USR(1676,X)
Now the '?' and the ':' can be edited out and the line, when listed again, will appear as:
        10 IF X=10 THEN SPEAK=USR(1676,X)
which is acceptable to BASIC. (Note: the question mark, '?', is the abbreviated form of the PRINT statement in Atari BASIC.)

The alternative method of using a VOICEMASTER command with an 'IF THEN' statement is to put the VOICEMASTER command on a seperate program line and use the 'IF THEN' statement as a pointer to that line. For example:

```
10 IF X=10 THEN 40
20 REM These are other
30 REM program lines.
40 SPEAK X
```

3. Another 'peculiarity' is that: the wedge will always convert to the USR(X) form a VOICEMASTER command that immediately follows a colon. With this program line:

```
10 REM    This  is   an  example   of   a   VOICEMASTER
command:SPEAK 1
```

the wedge would convert the 'SPEAK 1' to the USR(X) form and the line, when listed to the screen, would appear as:

```
10 REM    This  is   an  example   of   a   VOICEMASTER
command:SPEAK=USR(1676,1)
```

This means that, if you do not want the wedge to convert the VOICEMASTER command to the USR(X) form, then you must be careful not to preceed the command with a colon in a BASIC REM, PRINT or DATA statement.

If you keep in mind the above limitations when writing your BASIC programs, then you will have no problem with the wedged commands.
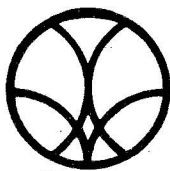
COVOX CO.
675-D Conger Street
Eugene, OR 97402