sp synth CBM64

Commodore Magic Voice Speech Module and Speech64

updated 16 jan. 2000

Speech64, Digitized Speech Without Special Hardware

Dropbox

One of the most fascinating applications of microcomputers is speech synthesis. Unfortunately, giving your computer the gift of speech usually involves purchasing expensive hardware. Using Speech64, you can give your Commodore 64 a voice without any special equipment. You will need a cassette recorder and Datasette for digitizing your speech, but you need no extra hardware for playback. Your recorded voice is reproduced in the standard audio output of any Commodore 64.

Before we get started, here is a quick overview of what we are going to do. First you will record your voice onto a cassette tape using any cassette tape recorder. Next you will place this tape into the Datasette connected to your Commodore 64. Please note that it is a tape recording of a voice and not a computer program that is loaded into the Datasette. Finally, you will run the Speech64 program, which will record the speech from the Datasette. You will then be able to play back and save the speech data. The point here is that we are not using the Datasette to load or save programs (unless you do not have a disk drive), but only as a means to input the speech signal.

Let's get started. Type in the Speech64 program on page 103 and save it. Now, using a standard cassette recorder, record a short phrase (under 8 seconds). If you are using a recorder with a built-in microphone, use a plug-in mike instead of the built-in one. The reason is that the built-in microphones tend to pick up too much noise, and your voice will sound very thin on the computer. This is an interesting phenomenon, and you may wish to experiment with different microphones.

Anyway, once you have recorded a few seconds' worth of speech on a cassette tape, rewind the tape back to the beginning of your speech and place the tape in the Datasette (connected to the C-64). Run the Speech61 program and choose the RECORD option from the menu. You will be instructed to press play on the Datasette. When you press play, the screen will blank while your voice is read from the tape into the computer. After about 8 seconds, the menu will reappear, indicating that the recording is done. Press stop on the Datasette. Now choose option 2, PLAYBACK, from the menu. You should now hear your digitized voice emanating from the monitor's speaker. The reproduced speech will have a raspy, buzzing quality simllar to that of a CB radio, but it should be quite intelligible. You should speak slowly and articulate your words to get the best results when recording your voice on tape.

At this point you may save the digital speech data using option 4 from the menu. Note that this is the digitized speech data, and not the original voice signal you recorded onto cassette earlier. By saving this digital data, you can use it later in your BASIC programs. The default device is the disk drive, but you may save to cassette by entering a 1 instead of 8 for the device number. If you are saving to cassette, do not forget to replace the cassette with your voice on it with a program cassette.

Once you have digitized and saved your speech, you are ready to add a voice to your own BASIC program. All you need to do is put the following lines in your program:

```
10 IF A=0 THEN A=1: LOAD"YOURDATA",8,1
20 GOSUB 4000: REM POKE ML ROUTINES
30 REM
40 REM PUT YOUR PROGRAM HERE
50 REM
2999 STOP :REM END OF YOUR PROGRAM
3000 REM SAY THE PHRASE
3010 SRT=64 : A=49154:POKE 49358,5
3020 POKEA ,0:POKEA+1 , SRT:POKEA+3 ,SRT+24:POKEA+2, 0
3030 P0KE53265 PEEK( 53265)AND239 REM BLANK SCREEN
3040 SYS 49287 REM SPEECH PLAYBACK
3050 P0KE53265, PEEK( 53265)0R16 : REM ENABLE SCREEN
3060 RETURN
4000 REM PUT LINES 4000-5430 FROM SPEECH
64 PROGRAM HERE
```

Whenever you want your computer to speak, just use GOSUB 3000. Line 10 assumes you are using disk, but if you want to load from cassette, just change the 8 to a 1.

You may want to have some fun altering your digitized voice by varying the pitch. You may do this in line 35 of Speech64 which POKEs location 49358 with a 5. First, record the phrase using the initial value of 5. Then, during playback, replacing the number 5 with a smaller number will increase the pitch; a larger number will decrease the pitch.

One other option you have is whether or not you wish to blank the screen during playback. Speech quality is slightly degraded if you do not blank the Screen, because the computer must spend some time updating the display. If you choose to leave the screen visible during playback, you may omit line 3030 from the above program.

For those of you who are interested in the technical aspects of how Speech64 works, it is based on a method of signal processing called zerocrossing analysis. Essentially, this means that the analog speech signal is transformed into a square wave of 0's and 1's that the computer can understand. These 0's and 1's are used to click the speaker on and off at audio frequency, hence reproducing the original signal. The advantage of this technique is that it is a relatively simple process and requires very little memory. On the other hand, we have to pay a price in terms of speech quality - the sound is somewhat raspy but it is intelligible.

I have found experimenting with speech reproduction on the Commodore 64 to be rewarding, educational, and just plain fun. I hope that Speech64 proves to be just as exciting for you.

SEE PROGRAM liSTING ON PAGE 103

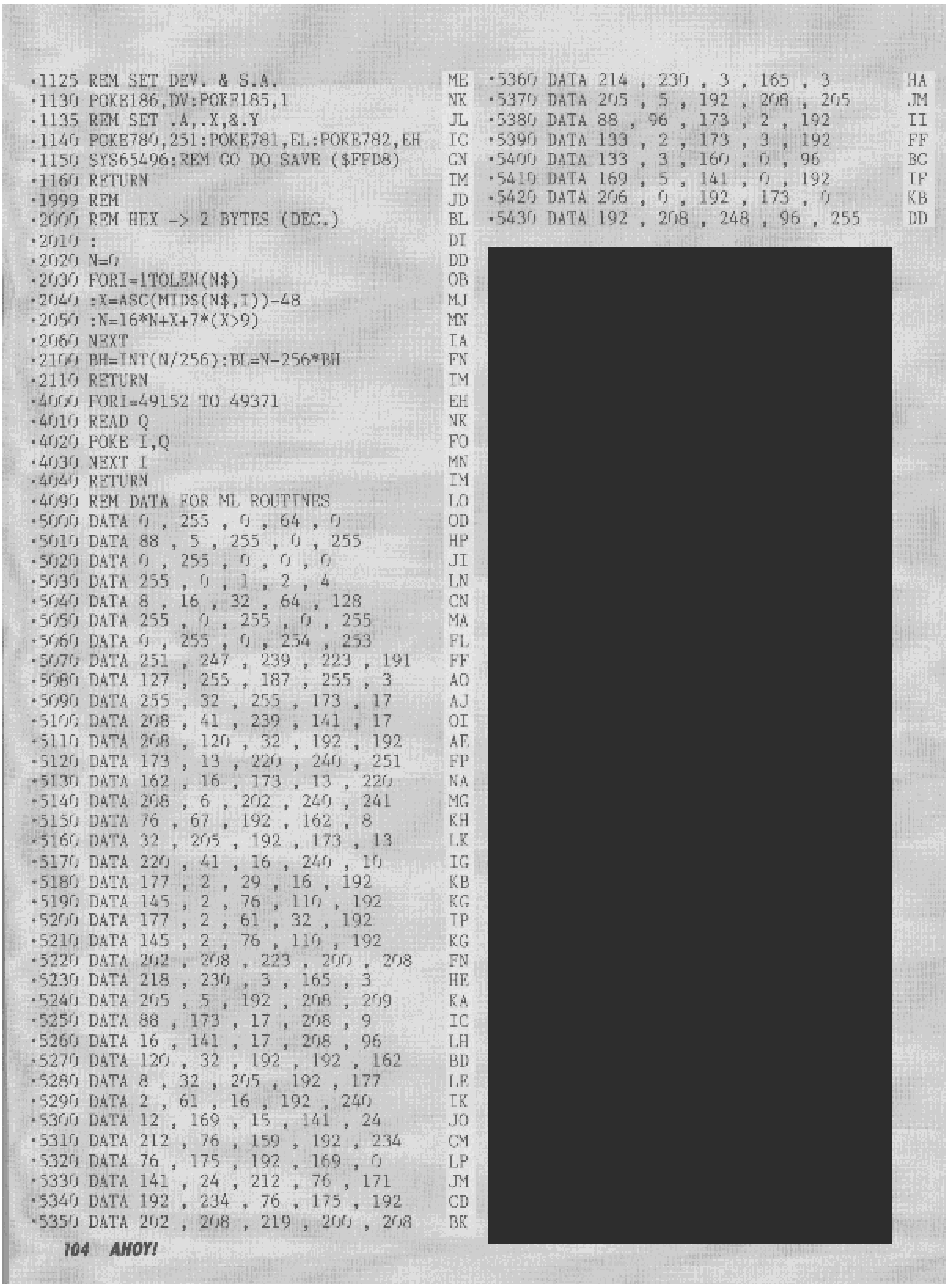
IMPORTANT. Letters on white background are BUG REPELLENT line codes. Do not enter them. Pages 87-89 explain these codes and provide other essential information on entering AHOY programs. Refer to thes pages before entering any programs.

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 87-89 explain these codes and provide other essential information on entering *Ahoy!* programs. Refer to these pages **before** entering any programs!

[redacted]

SPEECH64
FROM PAGE 39

```
•10 REM SPEECH64                            MN
•20 REM BY SCOTT C. BAGGS                   IH
•30 GOSUB 4000 :REM POKE ML ROUTINES        JJ
•35 POKE49358,5 : REM  SET PITCH            DO
•40 PRINT CHR$(147): REM CLEAR SCREEN       EG
•50 PRINT TAB(14);"* SPEECH64 *"            OM
•52 PRINT:PRINT:PRINT                       EF
•54 PRINT TAB(14);"1  RECORD WORD":PRINT
     TAB(14);"2  PLAYBACK WORD"                GE
•55 PRINT TAB(14);"3  LOAD DATA  ":PRINT
     TAB(14);"4  SAVE DATA[4" "]"              HP
•58 GETA$:IFA$=""THENGOTO   58                 NN
•59 IFA$>"4"THENGOTO   58                      KN
•60 ON VAL(A$) GOSUB 200,70,400,1000           BD
•65 GOTO40                                     PE
•70 REM  PLAYBACK PRAMETERS                    KP
•100 SRT=64 : GOSUB110                         IH
•102 POKE53265,PEEK(53265)AND239 :REM BLA
 NK SCREEN                                     FF
•105 SYS49287  : REM PLAYBACK                  LP
•106 POKE53265,PEEK(53265)OR16 :REM ENABL
 E SCREEN                                      IG
•108 RETURN                                    TM
•110 A=49154 : REM  $C002                      NA
•120 POKEA,0:POKEA+1,SRT :POKEA+3,SRT
     +24  :POKEA+2,0 : RETURN                  GG
•200 REM RECORD                                FH
•210 PRINT "RECORD - PRESS PLAY ON CASSET
 TE"                                           DA
•215 IF(PEEK(1)AND16)=16THEN 215 :REM WAI
 T                                             JH
•220 SRT=64: GOSUB110                          IH
•230 PRINT "START TAPE":
     SYS49200                                  MO
•240 RETURN                                    IM
•400 REM  LOAD DATA                            NB
•420 INPUT "FILENAME";FLN$                     CE
•425 INPUT"[DOWN][DOWN]DEVICE NUMBER #8[3
 "[LEFT]"]";DV                                 OC
•430 LOAD+FLNS,DV,1                            HF
•480 RETURN                                    IM
•1000 PRINT"[CLEAR][DOWN][DOWN]SAVE SPEEC
 H DATA"                                       OL
•1005 INPUT"[DOWN][DOWN]DEVICE NUMBER #8[
 3"[LEFT]"]";DV                                OC
•1010 INPUT"[DOWN][DOWN]FILE TO SAVE ";F$
 :IFF$=""THEN1010                              LL
•1015 REM LOC. OF F$                           KM
•1020 POKE187,PEEK(71):POKE188,PEEK(72)        GA
•1025 REM CSLC. POINTER TO F$                  CA
•1030 FA=PEEK(187)+256*PEEK(188)               IO
•1035 REM SET FILENAMR LENGTH                  JO
•1040 POKE183,PEEK(FA)                         PJ
•1055 REM SET FN. POINTER                      GE
•1060 POKE187,PEEK(FA+1):POKE188,PEEK(FA+
 2)                                            BA
•1070 REM                                      JD
•1080 SA$="4[3"0"]"                            HE
•1090 N$=SA$:GOSUB2000:SL=BL:SH=BH             IA
•1095 REM                                      JD
•1100 EA$="5800"                               GF
•1110 N$=EA$:GOSUB2000:EL=BL:EH=BH             JG
•1115 REM SET STRT ADD. PTR.                   OK
•1120 POKE251,SL:POKE252,SH                    HK
```

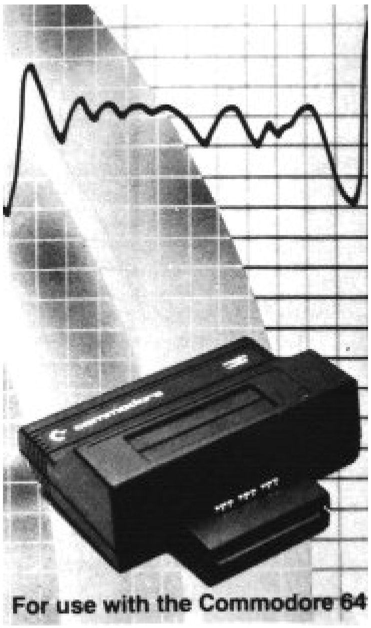


```
•1125 REM SET DEV. & S.A.                      ME
•1130 POKE186,DV:POKE185,1                     NK
•1135 REM SET .A,.X,&.Y                        JL
•1140 POKE780,251:POKE781,EL:POKE782,EH        IC
•1150 SYS65496:REM GO DO SAVE ($FFD8)          GN
•1160 RETURN                                   IM
•1999 REM                                      JD
•2000 REM HEX -> 2 BYTES (DEC.)                BL
•2010 :                                        DI
•2020 N=0                                      DD
•2030 FORI=1TOLEN(N$)                          OB
•2040 :X=ASC(MID$(N$,I))-48                    MJ
•2050 :N=16*N+X+7*(X>9)                        MN
•2060 NEXT                                     IA
•2100 BH=INT(N/256):BL=N-256*BH                FN
•2110 RETURN                                   IM
•4000 FORI=49152 TO 49371                      EH
•4010 READ Q                                   NK
•4020 POKE I,Q                                 FO
•4030 NEXT I                                   MN
•4040 RETURN                                   IM
•4090 REM DATA FOR ML ROUTINES                 LO
•5000 DATA 0 , 255 , 0 , 64 , 0               OD
•5010 DATA 88 , 5 , 255 , 0 , 255             HP
•5020 DATA 0 , 255 , 0 , 0 , 0                JI
•5030 DATA 255 , 0 , 1 , 2 , 4                LN
•5040 DATA 8 , 16 , 32 , 64 , 128             CN
•5050 DATA 255 , 0 , 255 , 0 , 255            MA
•5060 DATA 0 , 255 , 0 , 254 , 253            FL
•5070 DATA 251 , 247 , 239 , 223 , 191        FF
•5080 DATA 127 , 255 , 187 , 255 , 3          AO
•5090 DATA 255 , 32 , 255 , 173 , 17          AJ
•5100 DATA 208 , 41 , 239 , 141 , 17          OI
•5110 DATA 208 , 120 , 32 , 192 , 192         AE
•5120 DATA 173 , 13 , 220 , 240 , 251         FP
•5130 DATA 162 , 16 , 173 , 13 , 220          NA
•5140 DATA 208 , 6 , 202 , 240 , 241          MG
•5150 DATA 76 , 67 , 192 , 162 , 8            KH
•5160 DATA 32 , 205 , 192 , 173 , 13          LK
•5170 DATA 220 , 41 , 16 , 240 , 10           IG
•5180 DATA 177 , 2 , 29 , 16 , 192            KB
•5190 DATA 145 , 2 , 76 , 110 , 192           KG
•5200 DATA 177 , 2 , 61 , 32 , 192            TP
•5210 DATA 145 , 2 , 76 , 110 , 192           KG
•5220 DATA 202 , 208 , 223 , 200 , 208        FN
•5230 DATA 218 , 230 , 3 , 165 , 3            HE
•5240 DATA 205 , 5 , 192 , 208 , 209          KA
•5250 DATA 88 , 173 , 17 , 208 , 9            IC
•5260 DATA 16 , 141 , 17 , 208 , 96           LH
•5270 DATA 120 , 32 , 192 , 192 , 162         BD
•5280 DATA 8 , 32 , 205 , 192 , 177           LE
•5290 DATA 2 , 61 , 16 , 192 , 240            IK
•5300 DATA 12 , 169 , 15 , 141 , 24           JO
•5310 DATA 212 , 76 , 159 , 192 , 234         CM
•5320 DATA 76 , 175 , 192 , 169 , 0           LP
•5330 DATA 141 , 24 , 212 , 76 , 171          JM
•5340 DATA 192 , 234 , 76 , 175 , 192         CD
•5350 DATA 202 , 208 , 219 , 200 , 208        BK
```

```
•5360 DATA 214 , 230 , 3 , 165 , 3            HA
•5370 DATA 205 , 5 , 192 , 208 , 205          JM
•5380 DATA 88 , 96 , 173 , 2 , 192            II
•5390 DATA 133 , 2 , 173 , 3 , 192            FF
•5400 DATA 133 , 3 , 160 , 0 , 96             BC
•5410 DATA 169 , 5 , 141 , 0 , 192            IF
•5420 DATA 206 , 0 , 192 , 173 , 0            KB
•5430 DATA 192 , 208 , 248 , 96 , 255         DD
```

[redacted]

Commodore Magic Voice Speech Module, the most realistic voice synthesiser on the market



CONTENTS

YOUR COMMODORE COMPUTER TALKS!

Your COMMODORE 64 computer has a VOICE... in fact, lots of voices! Whether you're a student, teacher, parent, hobbyist, executive, novice programmer or expert... you're going to enjoy your Commodore Magic VOIceTM Speech Module.

Here's a short list of the Magic Voice's special features:

· High quality UNDERSTANDABLE speech output
· Includes pre-programmed vocabulary of 235 utterances
· Accepts talking software on cartridge, diskette or tape
· Easily programmed using Commodore BASIC
· Accepts additional vocabularies on diskettes
· Generates speech and music simultaneously
· Generates speech and graphics/animation simultaneously

Your Magic Voice plugs directly into the cartridge port of your COMMODORE 64 or SX-Portable Color Computer. You can use diskette, tape or cartridge software (cartridges plug into the Magic Voice's convenient top-loading cartridge port). The Magic Voice will give voice to programs that talk. Most programs that don't include speech operate normally when the Magic Voice Is plugged in. However, some older programs may not work. If you have difficulty using a non-speaking program while the module is plugged in, turn off the computer and then remove the module.

The Magic Voice speaks In a pleasant female voice, and has the ability to generate natural sounding human speech using male, female, children's and character voices. The module comes with its own built-in vocabulary of 235 utterances, and you can add thousands of other words using speech-compatible cartridges, diskettes and tapes.

Commodore's "talking" software Includes the innovative "MAGIC DESK" series, learning programs like "A Bee C's", and talking games like WIZARD OF WOR and GORF! We've only lust begun to explore the educational and recreational opportunities created by talking computers: foreign language training, elementary education, creative strategy games - the possibilities are endless.

If you're a computer programmer, you can use the Magic Voice in your own BASIC or ASSEMBLY LANGUAGE programs! If you develop a program you'd like us to consider as a software product, send a sample to our Software Acquisitions Manager in care of Commodore Software, 1200 Wilson Drive, West Chester, PA 19380.

To find out about new speech-compatible products as they're introduced, see your Commodore dealer. and read Commodore's two computer magazines (POWER/PLAY and COMMODORE).

1. INSTALLING YOUR MAGIC VOICE SPEECH MODULE

1. Using A Television

It's easy to set up the Magic Voice with your COMMODORE 64 or SX-64 computer. Just follow these simple steps:

STEP 1: SET UP YOUR COMPUTER for use with television as described in your User's Guide. Do NOT turn it on yet.

STEP 2: PLUG THE MAGIC VOICE INTO THE CARTRIDGE PORT of
your computer.
STEP 3: LOCATE THE AUDIO CABLE that comes packed with your speech module.

STEP 4: TAKE THE PHONO PLUG of the cable and PLUG IT INTO THE "AUDIO OUT" SOCKET on the speech module (there are two sockets on the side of the module. The audio in and out sockets are marked on the bottom of the module.

STEP 5: TAKE THE 5-pin DIN AUDIOIVIDEO CONNECTOR of your AUDIO CABLE and PLUG IT INTO THE AUDIOIVIDEO SOCKET on the computer.

STEP 6: TURN ON THE TELEVISION SET. TURN THE VOLUME UP.
(Note: if you're using a speech-compatible cartridge, PLUG IT
INTO THE CARTRIDGE SLOT on the top of the speech module
... DON'T EVER PLUG IN A CARTRIDGE UNLESS YOUR COMPUTER IS TURNED OFF!)

STEP 7: TURN ON THE COMPUTER.

Note: if you're using pre-recorded software on DISKETTE or
TAPE, LOAD the program at this time and RUN It.)

STEP 8: TEST THE MAGIC VOICE by typing the following exactly as shown (note that to obtain the quotation marks (") around the word Hi, you should hold down the SHIFT key and press the 2 key at the same time):

SAY "HI" (and press the RETURN key)

The computer should say "Hi", and you're ready to go. If It does not respond properly, adlust the volume and fine tuning adjustments on your television and try .... then consult the TROUBLESHOOTING CHART on Page 4.
* Note: The Commodore Magic Voice may be used with both the COMMODORE 64 and COMMODORE SX- computers, but we refer to COMMODORE 64 throughout the text, since the SX-64 portable computer is compatible with the COMMODORE 64 and works identically when used with the Speech Module.

2. Using A Monitor

Using the Magic Voice with a monitor may yield better sound quality than a television set. Commodore makes a low-priced color monitor which is specially engineered to provide maximum sound and picture quality on Commodore computers. The following steps describe how to use the speech module with Commodore Model 1701/1702 Color Monitor:

STEP 1: SET UP YOUR CQMPUTER for use with MONITOR, as described in your User's Manual. Do NOT turn it on yet.

STEP 2: PLUG THE MAGIC VOICE INTO THE CARTRIDGE PORT of your computer. The cartridge slot and COMMODORE label on the module must be facing UP.

STEP 3: Note: you will NOT USE THE AUDIO CABLE that comes packed with your Magic Voice module.

STEP 4: TAKE the long "TV CABLE" that comes with your COMMODORE 64 (it has identical PHONO PLUGS at each end) and PLUG ONE END INTO THE "AUDIO OUT" SOCKET on the speech module. PLUG THE OTHER END INTO THE "AUDIO" SOCKET on the front rlghthand corner of your Commodore Model 1701/1702 monitor.

STEP 5: TAKE THE "MONITOR CABLE" that comes with your Commodore monitor and PLUG THE LARGE AUDIO/VIDEO CONNECTOR INTO THE AUDIONIDEO SOCKET on your computer (the AUDIO/VIDEO connector is a round metal plug which has either 5 pins or 8 pins). Plug the VIDEO connector into the "VIDEO" SOCKET on your monitor. Connect the AUDIO PLUG on the monitor cable to the "AUDIO IN" SOCKET on the speech module.

STEP 6: Steps 6~ are the same as for a television set (above).

A SpeciaI Note for COMMODORE Color Monitor Owners:
The Commodore Model 1701/1702 Color Monitor has a special capability which can boost the quality even more on your COMMODORE 64... usIng 3 sockets on the BACK of the monitor labelled "Chroma," Luma" and "Audio." If you have a Commodore 64 with an 8-pin DIN audio/video connector, you can obtain a cable which has an 8-pin DIN connector on one end and 3 phono plugs on the other. You can improve the picture and sound quality by plugging one plug into the CHROMA socket, one plug into the LUMA socket, and the AUDIO plug into the AUDIO socket (you may have to experiment to determine which plug gets the CHROMA and LUMA sockets). When using the MAGIC VOICE with this arrangement, use the "TV Cable" from your COMMODORE 64 to connect the AUDIO OUT socket of the speech module to the AUDIO socket on the back of your Commodore monitor, and connect the AUDIO PLUG on the 8-pin DIN cable into the "AUDIO IN" socket on the speech module.

3. Using Your Own Audio System

You can channel the sound output from the Magic Voice through your own audio system, but if you do, remember that the SID CHIP which creates music synthesis and sound effects in the COMMODORE 64 operates SEPARATELY from the Magic Voice. This means if you use a cable to connect the AUDIO OUT socket on the module to the AUDIO IN socket on your audio system, you will need some special cables to connect the audio portion of the signal coming from the computer's music synthesizer. Do this by plugging the "AUDIO OUT" PLUG coming out of the computer into the "AUDIO IN" SOCKET on the speech module.

4. Troubleshooting Tips

if you've tried using your Magic Voice and your computer still isn't talking, try these troubieshoting steps:

1. Make sure your computer operates normally without the Magic Voice speech module.

2. Tune the fine tuning dial on your TV or monitor and go to Step 9.

3. Turn up the volume on your television set/monitor/system.

4. Turn off the power to your 64 and make sure the cartridge is firmly seated in the expansion slot on your computer.

5. Check the audio cable to make sure it's connected to AUDIO OUT and not AUDIO IN.

8. if using a TV, make sure the switchbox is set to "computer" and not "TV", and you're tuned to the right channel (Channel 3 or 4. See your Commodore 64 User's Guide).

7. Turn the computer OFF and ON and try again.

8. If using a Commodore monitor, make sure the "front/rear" selector switch on the back of the monitor is properly set.

9. Retype the SAY test command and press the RETURN key. The SAY test command is SAY "HI" <return> -

2. MAKING YOUR COMMODORE 64 TALK

1. Your First Words

Type these lines exactly as shown:
SAY"THE" (and press the RETURN key)
SAY"COMMODORE" (and press the RETURN key)
SAY"COMPUTER" (and press the RETURN key)
SAY" IS" (and press the RETURN key)
SAY"TERRIFIC" (and press the RETURN key)

Your computer has a female voice! Of course, special software can create any type of voice - man, child, cartoon character - even sound effects. One of the first voices created by Commodore's speech technology researchers was the robot voice in the best-selling game WIZARD OF WOR, available on cartridge for the Commodore 64.

As you've seen, you can SAY any of the words in the Magic Voice built-in vocabulary, ONE WORD AT A TIME, if you

1. Type the SAY command
2. Enclose the word you want to say in QUOTATION MARKS
3. Press the RETURN key to activate speech

2. Making Plural Words

Let's learn another secret of the Magic Voice - how to make plural words, words that end in "S". Try typing this example exactly as shown:
SAY"COMPUTER":SAY"SSSS" (press the RETURN key)
Did you hear "S" at the end of the word "COMPUTER"? There's a special utterance in the Magic Voice vocabulary which produces an "S" sound for pluralizing words... the special utterance is spelled with FOUR "S's".

Before we learn how to write simple computer programs that "talk", let's take a quick look at the Magic Voice built-In vocabulary...

3. Using the Built-in Vocabulary

There are 235 utterances built Into the Magic Voice vocabulary, which means you can mix and match a wide range of words and sounds to come up with all types of sentences. Colors, numbers, math words, computer words... they're all here. Most of the common words you'll want to use in your programs are included.

Take a look at the alphabetical listing on Page 22. Notice that each word also has a VOCABULARY NUMBER. If you like, you can use the number instead of the word. For example, look at the word "READY" which has the VOCABULARY NUMBER 144. Now

try these two examples:
SAY"READY" (press the RETURN key)
SAY144 (press the RETURN key)

Both examples have the same result.Remember... if you SAY an utterance be sure to put it in quotation marks. If you SAY the VOCABULARY NUMBER of the utterance, do not use quotation marks; just type SAY and the number.

4. Beginner's Programming Tips -How Basic Programs Work

So far, you've typed all commands DIRECTLY into the computer... but a much more efficient way is to type the same commands using a COMPUTER PROGRAM. Computer programs for your COMMODORE 64 are written in the computer language called BASIC, which is built into your Commodore computer.

If this is your first experience with BASIC PROGRAMMING, the following information may help you get started with the examples we're going to explore:

A BASIC program consists of one or more NUMBERED LINES with BASIC commands on each line. Every time you type a number at the far left margin followed by a valid BASIC command, your computer REMEMBERS what you typed and stores that line in its memory. BASIC programs are typically numbered by tens (10, 20, 30 and so on)... that's so you can go back later and insert additional lines (11,12,13, etc). A BASIC program using the Magic Voice looks like this... try typing it:
10 SAY"COMMODORE" (press RETURN)
20 SAY"COMPUTER" (press RETURN)
30 SAY"SSSS" (press RETURN)

Type the word RUN and press the RETURN key to hear the program, then type the word LIST and press RETURN to see the program listed. Here are a few "basic" rules for beginning programmers:

· Always start a program line at the far LEFT (first) column on your screen.

You must press the RETURN key at the end of each line to enter it into the computer's memory.

· A program actually doesn't start working until you type the word RUN and press the RETURN key. To STOP a program which keeps running, press the RUN/STOP key.

To view your program lines, type the word LIST and RETURN.
To erase an entire program, type the word NEW and RETURN.
· To erase one line in a program, type the LINE NUMBER BY ITSELF and press RETURN, or else RETYPE the entire line.

· To edit any line, LIST the program, then move the cursor to the line you want to change, change it and press RETURN.

Counting from 1 to 10

if you type the SAY command with a word, the word must be in quotation marks, but if you type the SAY command with the NUMBER of the word, you don't use quotation marks. This NUMBERING feature was included because using numbers allows BASIC programmers to conserve space in their programs. Here's a short example:
10 FORX= 1TO10 (press RETURN)

```
20 SAYX (press RETURN)
30 NEXT (press RETURN)
```

Type the word RUN and press the RETURN key. Note that the vocabulary numbers zero to ten are the same as the spoken numbers.

SAYing the Alphabet

Now type the following line exactly as shown, then type the word RUN and press the RETURN key:

```
10 FORX=21T046 (press RETURN)
```

This is the same as the counting program, except here we substituted vocabulary numbers 21 to 46, which are the numbers of the letters A through Z.

Well, your computer is starting to talk - but what if you want to create longer sentences? This requires some simple computer programming ... whether you're a beginner or expert, you'll be surprised how easily you can program speech using the Commodore Magic Voice.

5. Your First Sentences

Here's a little program which speaks a sentence. Type the word NEW and press RETURN to erase your previous program, and type the example as shown:

```
10 FORX= 1T05
20 READA$
30 SAYA$:NEXT
40 DATACOMMODORE,COMPUTER,SSSS,ARE,TERRIFIC
```

Type RUN and press RETURN. Is this a terrific sentence? Here's how the program works:
Line 10: A FOR...NEXT loop tells the computer to count from 1 to 5. The NEXT command farther down in the program tells the computer to perform all the actions between the FOR and NEXT commands... in this case, READ and SAY 5 words from the DATA statements.

Line 20: READA$ means look at the DATA in Line 40 and "read" it one item at a time (there are 5 items), all separated by commas.

Line 30: SAYA$ tells the speech module to speak each item, one item at a time The NEXT command is the other part of the FOR...NEXT loop. FORX= 1TO5 told the computer to perform the following actions 5 times (READA$ and SAYA$ which means READ one item from the DATA statements and SAY that item) and the NEXT command said go back and do it again until the LIMIT (5) is reached.

Line 40: The DATA contained here consists of 5 "words" which together make up the sentence you just programmed. The sentence the computer spoke was: "COMMODORE COMPUTERS ARE TERRIFIC".
Remember that a plural word requires that you add four S's as a separate word, immediately after the word you want to pluralize.

Now type the word NEW and press the RETURN key to erase your previous program.

Here's another example which includes 7 utterances... type this as shown:

```
10 FORX= 1TO7 (RETURN)
20 READA$ (RETURN)
30 SAYA$:NEXT (RETURN)
40 DATAYOUR,COMMODORE,SIX,TY,FOUR,CAN ,TALK (RETURN)
```

Type and RUN this program. (Note if you missed a comma or made another mistake, the computer will give you an ERROR message... LIST your program and retype the line which Is wrong.) Here, the sentence reads: "Your Commodore 64 can talk." Did you notice that in LINE 40 we split the word SIXTY into two parts - SIX and TY? This is because our researchers were able to save more space for extra words in our vocabulary by using the "TY" and combining it with FOUR to make FORTY, FIF to make FIFTY, SIX to make SIXTY, SEVEN to make
SEVENTY, and so on ... otherwise we would need separate words for all of these numbers and you would have fewer words for making sentences. The same applies to numbers in the "teens" - for example, THIRTEEN becomes two separate words: THIR and TEEN.

6. SAY and Spell

This little program will show you some general programming tips, as well as how to turn your computer into a SPELLING MACHINE. Type in this program:

```
10P0KE53280,6:P0KE53281 ,0:POKE646,7
20PRINTCHR$(147)TAB(240)
30PRINT"TYPE A WORD AND PRESS RETURN":INPUTX$
40FORY = 1TOLEN(X$)STEP1 :SAYMID$(X$,Y,1):NEXT
50GOTO10
```

Type the word RUN and press the RETURN key, then enter any word (no spaces or numbers) and press the RETURN key. Your COMMODORE 64 will automatically SPELL OUT LOUD the word you typed. The program keeps "looping back" to itself, which means it will keep asking you to type in a word ... to get Out of this "loop", hold down the RUN/STOP key and at the same time, press the RESTORE key. You can always stop a program which is running in this way. The computer will clear itself, BUT YOUR PROGRAM IS STILL IN THE COMPUTER so if you RUN or LIST it, it's still there. Let's see how this program works:

LINE 10 includes 3 POKE commands which change the screen and border colors. POKE 53280 followed by a comma and a number from 0 to 15 will change the BORDER COLOR. POKE 53281 changes the SCREEN COLOR and POKE 646 changes the CHARACTER COLOR. So the first POKE changes the border to blue, the second POKE changes the inside screen to black and last POKE changes the character color to yellow. You can type these POKEs directly into the computer without writing a program. Try experimenting yourself with different color combinations!

LINE 20 clears the screen (PRINTCHR$(147) is the same as PRINT"CLR/HOME") and TABs 240 spaces from the upper left corner, which just happens to be 6 lines down on your screen - this is used to position the message.

LINE 30 PRINTs the message on the screen. The INPUTX$ command tells the computer to wait until you type in a word (letters only, no numbers or spaces) and automatically prints a question mark.
LINE 40 uses a FOR...NEXT loop. Remember we said that a FOR...NEXT loop causes all the actions between the FOR and NEXT to be performed, up to the limit of the loop? Here, we start by saying that the loop limits are from 1 (the first letter) up to the LENgth

of the word you typed in. LEN(X$) means the length of the word you type in, which the computer has defined as "X$". So if you type a 9-Ietter word like "COMMODORE", the length of the word is 9 and the loop becomes:
FORY= 1T09. The next portion SAYMID$(X$,Y,1) tells the computer which letter to say. X$ is your word. Y is the position of the letter - It starts with position 1 and changes each time until the "loop" reaches the limit, which is 9, or the LENgth of our word. The number 1 can be considered as a "constant" for the purpose of this discussion.

LINE 50 uses a GOTO statement. This causes the computer to go back to the beginning of the program and start over again at LINE 10.

7. SAYing Numbers

The following example gives you a fairly simple way of telling the computer to SAY numbers generated In your program. it doesn't matter whether the numbers being generated come from a student entering the answer to a question, or are being used to repeat a number out loud for an accountant who doesn't have time to look up from a checklist during a calculation ... whatever the use, it's helpful if the computer is able to speak all numbers up to 999,999,999.

```
5 DIMB%(12)
10 FORL = 1T09:READA$(L):NEXT:FORL=2T09:READB$(L):NEXT:
FORL = 0T09:READC$(L):NEXT
15 FORL= 1T02:READD$(L):NEXT
20 POKE53280,6:P0KE53281,0:POKE646,7
25 PRINTCHR$(147)TAB(240)
30 PRINT"TYPE A NUMBER AND PRESS RETURN":INPUTZ
35 IFZ>999999999THEN:SAY"TOO":SAY"LARGE":G0T025
40 I FZ< -999999999THEN :SAY"TOO":SAY"SMALL"G0T025
45 IFZ<OTHEN:SAY"MINUS"
50 I FZ = 0THEN :SAY"ZERO":G0T025
55 D= 0:Z$ = STR$(Z): RATE(2)
60 FORL=LEN(Z$)TO2STEP -1: B % (D)=VAL(MID$(Z$, L,1)):
D= D+1:NEXT
65 FORL= INT(D/3)TO0STEP-1:A= B%(L*3+2):B= B%(L*3+1):
C= B%(L*3)
70 IFA >0THEN:SAYA$(A):SAY"HUNDRED"
75 IFB >1THEN:SAYB$(B):IFB$(B)< >"TWENTY"THEN:SAY"TY"
80 IFB =1THEN:SAYC$(C):IFC >2THEN:SAY"TEEN"GOTO90
85 IFC>0ANDB<>1THEN:SAYA$(C)
90 IFA+ B+ C>0ANDL>0THEN:SAYD$(L)
95 NEXT:RUN
100 DATAONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE
110 DATATWENTY,THIR,FOUR,FIF,SIX,SEVEN,EIGHT,NINE
120 DATATEN,ELEVEN,TWELVE,THIR,FOUR,FIF,SIX,SEVEN, EIGHT,NINE
130 DATATHOUSAND,MILLION
READY
```

8. The RATE Command - Changing the Talking Speed

You can change the "rate" or speed at which utterances are said, by adding a special command called the RATE command. Here's an example:

```
10 RATE1 :SAY"APOSTROPHE"
20 RATE4:SAY"APOSTROPHE"
```

```
30 RATE8:SAY"APOSTROPHE"
40 RATE10:SAY"APOSTROPHE"
```

Type RUN and press RETURN. Did you hear the difference In the rate each number was spoken? The computer talked slower each time. There are 10 RATE settings. RATE1 is the FASTEST speech and RATE 10 is the slowest. RATE4 is standard.

Here's a program that lets you vary the RATE of EACH WORD spoken in a sentence - the RATE settings in the DATA statements cause the utterances which FOLLOW to be spoken at that rate, until a different rate is specified. This is helpful in fine-tuning your programs to provide more natural sounding speech.

```
10 PRINT"CHANGING RATE IN SENTENCE"
20 GOSUB 5000
30 PRINT"PRESS ANY KEY FOR ANOTHER PHRASE"
40 GETA$:IFA$ = ""THENGOTO40
50 GOSUB5000
60 SAY"AGAIN":INPUT"AGAIN";A$
70 IFLEFT$(A$,1) <> "Y"THEN END
80 RESTORE:GOTO10
5000 READSP$ gets a word from DATA below
5010 IFSP$= "."THENRETURN It the word 18 a period then RETURNs above
5020 IF LEFT$(SP$,4)="RATE"THEN READSP: RATE (SP) :GOTO5OO if word Is
"RATE" -then READ the next
number to set RATE
5030 SAYSP$ say the word
5040 GOTO5000 do it again
5050 DATATHE,COMMODORE,RATE,1 ,SIX,TY, FOUR, RATE,4
5060 DATAIS,TERRIFIC,RATE,6, BECAUSE,RATE,3,NOW
5070 DATAIT,RATE,2,CAN,RATE,4,TALK,.
5080 DATARATE,5,THIS,RATE,3,lS,RATE,8,GOOD,.
```

9. Using The RDY Command

Because your COMMODORE 64 has the ability to generate speech simultaneously and independently of music and graphics, sometimes it becomes necessary to tell the computer to wait until speech is completed so that an animation, song or sound effect is properly timed in relation to speech.

Here's a quick example of how you might use the RDY command ... first, type the word NEW and press RETURN to erase any previous programs, then type In this program... notice that spoken words and printed words aren't synchronized properly:

```
10 PRINT"THIS":SAY"THIS"
20 PRINT"WORD":SAY"WORD"
30 PRINT"IS":SAY"IS"
40 PRINT"BLUE":SAY"BLUE"
```

Now type the word NEW and press RETURN, then enter this program:

```
10 IFNOTRDYTHENGOTO10
15 PRINT"THIS":SAY"THIS"
20 IFNOTRDYTHENGOTO20
25 PRINT"WORD":SAY"WORD"
30 IFNOTRDYTHENGOTO30
35 PRINT"IS":SAY"IS"
40 IFNOTRDYTHENGOTO40
```

```
45 PRINT"BLUE":SAY"BLUE"
```

The RDY command tells the computer that if NOT READY (in other words, if the computer isn't finished speaking the last word), THEN GOTO the same line and keep going back until the word is finished. Only then does the program drop through to the next command. The RDY command is often used as a NOTRDY condition as shown hem. It can be used in a similar way If you are programming simultaneous graphics and speech or music and speech bacause the computer can process speech1 graphics and music simultaneously, which means sometimes you will want the computer to wait until an utterance is completed before proceeding with a song, displaying a graphic symbol, or PRINTIng a message on the screen.

3. SUMMARY OF BASIC SPEECH COMMANDS

This section describes the Magic Voice program mode from the BASIC language. This feature gives users the ability to write their own games and applications using the Magic Voice preprogrammed vocabulary. Four new commands have been added to Commodore BASIC to make speech easy to use.

1. SAY Command

This command asks the Magic Voice to say a specific word from its preprogrammed vocabulary or additional vocabularies from diskettes, cassettes, and ROM cartridges. The Module accepts either the word number as a numeric expression with a value of 0,...,234; or a BASIC text string spelling out the word to be said. The distinction is made automatically by the module based on the type (string or numeric) of the argument it receives.

Examples:

```
SAY"HI" (Module says "HI")
B$= "HI"
SAY B$ (Module says "Hi")

SAY 21 (Module says word #21 in its vocabulary)
X=21
SAY X
SAY(3* 7)
```

2. RATE Command

This command sets the rate or speed at which words and phrases are spoken. Ten speeds are available ranging from 0.65 times slower to 1.4 times faster than the standard rate. The fastest rate is #1, the standard rate is #4, and the slowest rate is #10. The RATE command must be passed a numeric argument.

Example:

```
FOR 1= 1TOl0
RATE(l) REM SET SPEED
SAY "HI" REM NOW LISTEN TO IT
NEXT I
```

3. VOC Command

This command alerts the speech module to the presence of an extended vocabulary loaded into system memory. This vocabulary might come from optional disks or cassettes. This command is only needed when additional vocabulary data has been loaded into memory. Look for upcoming additional vocabulary from your Commodore dealer.

Example:

```
LOAD"FISH.VOCAB",8,1
FISH=32768 :REM ADDRESS OF VOCAB DATA
VOC(FISH) :REM TELL SPEECH MODULE
SAY"SHARK" :REM MODULE SAYS "SHARK"
```

4. RDY Command

This command is for use in many advanced applications. It is a logical function which retums the value true if the Magic Voice is ready to accept another SAY command and false if not.

Example:

```
10 IFNOTRDYTHEN10
20 SAY "HI"
30 IFNOTRDYTHEN30
40 SAY "BYE"
```

This command can also be used by BASIC programs to detect the presence of a speech module in a system. The variable named "RDY" will be zero when a program is LOADed and RUN if a speech module is not present. If the module is present the value of "RDY" will be - 1.

5. Error Conditions

The Magic Voice detects and informs the user of certain kinds of errors that it detects. The most common of these occurs when the user tries to "SAY" a word that is not in the module's vocabulary. When this happens, the module will flag an error to BASIC and the message "ILLEGAL QUANTITY ERROR" will appear on the screen.
The module's other functions (RATE & VOC) require numeric arguments. These functions will return the "? TYPE MISMATCH ERROR" message if called with a non-numerlc argument.

This simple command set gives the user the ability to program simultaneous speech output, music, and graphics actions. You may have noticed if you tried any of these examples that the BASIC READY message appears on the screen almost instantly - before the Magic Voice has finished saying what you asked it to say!

This is because the SAY command doesn't actually do the work of generating the speech output. it merely initiates the process by asking the module to start saying a word. Once the SAY command is finished "asking" the module to get started, the module operates independently and execution of our BASIC program continues while the speech is being produced! This is how combined speech, music, and graphics is possible.

The RDY command is included in order for your program to determine what the module is doing at any time. As an example, your program might start with a SAY"HI" and then play a three note tune. If you wanted to play the tune after the word "HI" was finished, you would use the following kind of program:

```
10 SAY "Hi"
```

```
20 IFNOTRDYTHEN20
30 GOSUB1000 :REMSUBROUTINETOPLAYTUNEAT1000
```

Several exciting examples of games that are created easily using these BASIC language commands and the pre-programmed vocabulary are included in this Commodore 64 Magic Voice User's Guide.

4. PROGRAMMING SPEECH IN ASSEMBLY LANGUAGE

The Commodore 64 Magic Voice includes a collection of linkage routines necessary to allow user assembly language programs to operate the speech synthesizer. User programs can "say" any of the built-in words and phrases as well as their own specially prepared speech data.

The synthesized signal is brought into the SID chip as an external input and mixed with music generated by the SID. It is important to note that the SID master volume must be turned on in order to hear the synthesized speech. The Magic Voice initializes the SID chip to a volume setting of 15 on power-up. User programs that manipulate the SID volume setting should restore It to 15 before attempting to generate speech output since the built-in talk command software does NOT Interact with the SID In anyway.

The design philosophy of the assembly language interface is that user access to the speech module occurs via service calls from the user program. Code to handle these service calls is copied into RAM at $C000-$C3FF at system power-on. In addition, the core of the speech driver routines and the native vocabulary of the Magic Voice are copied into RAM underneath the BASIC and KERNAL ROMS.

IT IS VERY IMPORTANT NOT TO DISTURB RAM MEMORY LOCATIONS $A000-$C3FF AND $E000-$FFFF for proper module operation. Programs which load their own speech data and do not rely on the built-in words and phrases of the module must only preserve $A000-$A800 and $C000-$3FF.

The assembly language Interface consists of subroutine calls with the arguments passed In the A and X registers.

There is a separate subroutine for each of the following functions:

Reset speech module
Get speech module status
Say a word

Detailed functional descriptions and calling specifications are provided below.

1. Assembly Language Calling Procedures

Upon power-up the speech module copies certain linkage code into memory locations $C000 to $C3FF and then enters a memory mapping mode appropriate to the demands of the cartridge. The cartrIdge program gains access to the module's programs by JSR'ing to the linkage programs which reside in that RAM area (which is active in all memory map modes).

Where possible, the calling procedures have been designed with ease of use in mind. Functions requiring a single byte except that byte in the processor's A-register. Where two bytes are needed the A-register and the X-register are both used. Only when a count is needed Is the Y-register used. Registers are preserved only as indicated in the detailed descriptions below. The conventional uses of the registers are as follows:

A-regIster: complete operand or low byte if an address is passed.
X-register: upper byte If an address is passed
Y-register: count (0 implies 256).

2. Memory Requlrements

Use of the speech module impacts the programmer's planning of memory utilization in several ways:

· RAM locations $0000 thru $C3FF are used for the linkage code and must NOT be modified by the application program.

· The following zero page locations are used during the voIce synthesis process (including NMI processing after speech is initiated).

$9B
$9C
$A7
$A8
$BD
$BE
$FB
$FC
$FD
· The NMI vector needed for voice synthesis is planted in locations $0318 and $0319 at the start of the synthesis operation. The original vector is relocated to locations $0338 and $0339. This original value Is restored when synthesis is complete.

3. Assembly Routines

SRESET - Reset The Speech System/Stop Talking

ENTRY: $COO3
ENTRY CONDITIONS: NONE

RETURNED INFORMATION: Accumulator is destroyed.

This command resets the synthesizer hardware. It should be included in applications programs for initialization purposes and may be used to stop speech output in mid-utterance.

SPSTAT - Get Status Of Speech Module

ENTRY: $C006
ENTRY CONDITIONS: NONE

RETURNED INFORMATION: ACC=0 if ready/- 1 if busy talking.
The N and Z bits of the processor status word are set accordingly. This command may be used to determine whether the speech module has completed previous "sayit" requests and whether a new "sayit" request can be handled immediately.

SAYIT - Say A Word

ENTRY: $C009
ENTRY CONDITIONS: X-reg = most significant portion of the word number. A-reg = least significant portion.

RETURNED INFORMATION: All registers destroyed.

This command initiates the synthesis of a word or phrase. Control is returned to the calling program almost immediately if the module is not already talking. If the module is talking, then control will not be returned until the current request can be initiated. Words number 0 through 255 are words which are in the speech module. Words number 256 through 4095 are interpreted as located in the user's program space.

CAUTION: Special care should be taken to ensure that no other process in the system that generates NMI's is active when the calls to the speech synthesizer are made. This includes the RS-232 channel and the serial bus devices. The RS-232 channel is the only channel which must actually be CLOSED. The serial bus channels may be open and may be left opened, but must not actively transmit data. It is recommended that programs which wish to perform series use I/O check, via SPSTAT to ensure that speech output is complete before initiating the operations.

SAYRAM - Say a Word From RAM

ENTRY: $C00C

ENTRY CONDITIONS: X-reg = most significant portion of the word number. A-reg = least significant portion.

RETURNED INFORMATION: All registers destroyed.
This command Is similar to the "SAYIT" except that the structure permitting access through the memory mapping system is bypassed. This entry point is intended to permit generation of speech from RAM based speech data. This may also be useful if RAM versions of words are modified under program control.

Speech is generated as in the SAYIT command above. Utterances 0...255 are taken from the on-board vocabulary and utterances 256...4095 are taken from memory as established by the SETTAB routine. No mapping Is performed. Speech data is taken directly from the program's address space.

NOTE: See caution under SAYIT above.

SPEED - Set Speaking Speed Of Synthesiser

ENTRY: $C00F

ENTRY CONDITIONS: A-reg = desired speed code from 1 ...10.

RETURNED IN FORMATION: No register affected.

The SPEED capability is used to speed up or slow down the speaking rate of the synthesizer. The range of speed variations Is from .65X decrease In the speed (speed code = 10) to a 1 .4X increase (speed code 1) Normal speed is speed code 4.

SETTAB - Set Utterance Look Up Table Address
ENTRY: $CO12

ENTRY CONDITIONS: A-reg = lower byte of address. X-reg = upper byte of address.

RETURNED INFORMATION: None.

The SETTAB function is used to tell the speech module the starting memory address of user supplied speech data. It is identical to the BASIC VOC command. This command MUST be employed prior to any attempts to generate speech with utterance numbers greater than 255.

SIGNAL - Enable Vectoring To Completion Code

ENTRY: $C015

ENTRY CONDITIONS: JMP instruction to completion code handling routine MUST be planted In memory locations $C018-$C01A

RETURNED INFORMATION: None.

This function is for support of advanced applications which implement asynchronous event driven programming techniques. The effect of the SIGNAL call is to increment a counting semaphore maintained by the talking process. This semaphore is waited on by the speech completion, which when signalled, will Jump to the completion code vector address ($C018). Users should plant a JMP instruction in the three bytes of memory starting at this address. ONLY $CO18-$C01A SHOULD BE CHANGED.

This JMP Instruction should point into user code which handles the completion of the speech output event. The user completion routine MUST terminate with an RTS Instruction. Completion code handlers may be used for a variety of purposes but, in general, should be kept short since IRQ's are disabled and NMI's are ignored during this time.

NOTE: This facility also provides a means for user programs to detect the presence or absence of a speech module: a power-up footprint. The completor code vector area ($C018-$C01A) Is Initialized on power-up to the instruction sequence:

```
$C018 NOP
$0019 NOP
$C01A RTS
```

These instructions correspond to the values $EA, $EA, $60. User programs may detect the presence of a module by verifying that these locations contain the given values.

5. APPENDIX

1. Resident Vocabulary

The Magic Voice contains a built-in vocabulary of 235 utterances In a female voice. The utterances are numbers 0 to 234. The following is a numerical listing of the utterances contained in the Commodore 64 Magic Voice.

0 ZERO
1 ONE
2 TWO
3 THREE

4 FOUR
5 FIVE
6 SIX
7 SEVEN
8 EIGHT
9 NINE
10 TEN
11 ELEVEN
12 TWELVE
13 THIR
14 FIF
15 TWENTY
16 HUNDRED
17 THOUSAND
18 MILLION
19 TEEN
20 TY
21 A
22 B
23 C
24 D
25 E
26 F
27 G
28 H
29 I
30 J
31 K
32 L
33 M
34 N
35 O
36 P
37 Q
38 R
39 S
40 T
41 U
42 V
43 W
44 X
45 Y
46 Z
47 (SILENCE)
48 ADD
49 SUBTRACT
50 DIVIDE
51 DIVIDED
52 EQUALS
53 LESS
54 MINUS
55 MORE
56 NUMBER
57 PLUS
58 REMAINDER

59 IT
60 AT
61 WHEN
62 HAS
63 TIMES
64 APOSTROPHE
65 WORD
66 ER
67 EST
68 ING
69 SSSS
70 TH
71 ALL
72 ME
73 MY
74 THAT
75 THEIR
76 THEY'RE
77 THIS
78 WHAT
79 WHICH
80 WHO
81 YOUR
82 YOU'RE
83 ANSWER
84 COMMODORE
85 END
86 POINT
87 THING
88 QUESTION
89 TURN
90 AM
91 ARE
92 BE
93 BUY
94 CAN
95 CORRECT
96 DID
97 DO
98 DOES
99 DONE
100 FIND
101 GET
102 HAVE
103 HEAR
104 HELP
105 IS
106 KNOW
107 LIKE
108 PRESENTS
109 READ
110 SAY
111 SEE
112 START
113 USE

114 WAlT
115 WANT
116 WAS
117 WON
118 WRITE
119 HERE
120 HIGH
121 HOW

122 THERE
123 TOO
124 WHERE
125 WHY
126 UH
12? AN
128 AND
129 AS
130 BUT
131 BY
132 FOR
133 FROM
134 IF
135 IN
136 LARGE
137 NEXT
138 NOT
139 OF
140 OFF
141 OH 179 TRY
142 ON 180 LETTER
143 OR 181 FIRST
144 READY
145 SAME
146 THAN
14? THE
148 THUH
149 THEN
150 TO 188 CYAN
151 WITH
152 VERY
153 BYE
154 HI 192 YELLOW
155 PLEASE
156 THANK
157 YES
158 BAD
159 GOOD
160 SORRY
161 WRONG
162 AFTER
163 MOST
164 BECAUSE
165 NO
166 OUT
167 SEND

168 TALK
169 ANY
170 EACH
171 ENTER
172 ERROR
173 MANY
174 EVERY
175 AGAIN
176 NOW
177 SMALL
178 TERRIFIC
179 TRY
180 LETTER
181 FIRST
182 SECOND
183 THIRD
184 COLOR
185 BLACK
186 WHITE
187 RED
188 CYAN
189 PURPLE
190 GREEN
191 BLUE
192 YELLOW
193 ORANGE
194 BROWN
195 GREY
196 LIGHT
197 HE
198 OUR
199 SHE
200 WE
201 YOU
202 COME
203 GIVE
204 PUT
205 SET
206 TAKE
207 WERE
208 UNTIL
209 MAY
210 OVER
211 DARK
212 COMMAND
213 COMPUTER
214 CONTROL
215 CURSOR
216 DISK
217 DOWN
218 GAME
219 KEY
220 LEFT
221 LOAD
222 NAME

223 PLAY
224 PRESS
225 PROGRAM
226 RIGHT
227 RUN
228 SAVE
229 STOP
230 SYMBOL
231 TAPE
232 UP
233 WEIGHT
234 HOUR

2. Alphabetical List of Words

A 21 E 25
ADD 48 EACH 170
AFTER 162 EIGHT 8
AGAIN 175 ELEVEN 11
ALL 71 END 85
AM 90 ENTER 171
AN 127 EQUALS 52
AND 128 ER 66
ANSWER 83 ERROR 172
ANY 169 EST 67
APOSTROPHE 64 EVERY 174
ARE 91 F 26
AS 129 FIF '14
AT 60 FIND 100
B 22 FIRST 181
BAD 158 FIVE 5
BE 92 FOR 132
BECAUSE 164 FOUR 4
BLACK 185 FROM 133
BLUE 191 G 27
BROWN 194 GAME 218
BUT 130 GET 101
BUY 93 GIVE 203
BY 131 GOOD 159
BYE 153 GREEN 190
C 23 GREY 195
CAN 94 H 28
COLOR 184 HAS 62
COME 202 HAVE 102
COMMAND 212 HE 197
COMMODORE 84 HEAR 103
COMPUTER 213 HELP 104
CONTROL 214 HERE 119
CORRECT 95 HI 154
CURSOR 215 HIGH 120
CYAN 188 HOUR 234
D 24 HOW 121
DARK 211 HUNDRED 16

DID 96 I 29
DISK 216 IF 134
DIVIDE 50 IN 135
DIVIDED 51 ING 68
DO 97 IS 105
DOES 98 IT 59
DONE 99 J 30
DOWN 217
K 31 Q 37
KEY 219 QUESTION 88
KNOW 106 R 38
L 32 READ 109
LARGE 136 READY 144
LEFT 220 RED 187
LESS 53 REMAINDER 58
LEUER 180 RIGHT 226
LIGHT 196 RUN 227
LIKE 107 S 39
LOAD 221 SAME 145
M 33 SAVE 228
MANY 173 SAY 110
MAY 209 SECOND 182
ME 72 SEE 111
MILLION 18 SEND 167
MINUS 54 SET 205
MORE 55 SEVEN 7
MOST 163 SHE 199
MY 73 (SILENCE) 47
N 34 SIX 6
NAME 222 SMALL 177
NEXT 137 SORRY 160
NINE 9 SSSS 69
NO 165 START 112
NOT 138 STOP 229
NOW 176 SUBTRACT 49
NUMBER 56 SYMBOL 230
O 35 T 40
OF 139 TAKE 206
OFF 140 TALK 168
OH 141 TAPE 231
ON 142 TEEN 19
ONE 1 TEN 10
OR 143 TERRIFIC 178
ORANGE 193 TH 70
OUR 198 THAN 146
OUT 166 THANK 156
OVER 210 THAT 74
P 36 THE 147
PLAY 223 THEIR 75
PLEASE 155 THEN 149
PLUS 57 THERE 122
POINT 86 THEY'RE 76
PRESENTS 108 THING 87
PRESS 224 THIR 13
PROG RAM 225 THIRD 183

PURPLE	189
THIS	77
PUT	204
THOUSAND	17
THREE	3
THUH	146
TIMES	63
TO	150
TOO	123
TRY	179
TURN	89
TWELVE	12
TWENTY	15
TWO	2
TY	20
U	41
UH	126
UNTIL	208
UP	232
USE	113
V	42
VERY	152
W	43
WAIT	114
WANT	115
WAS	116
WE	200
WEIGHT	233
WERE	207
WHAT	78
WHEN	61
WHERE	124
WHICH	79
WHITE	186
WHO	80
WHY	125
WITH	151
WON	117
WORD	65
WRITE	118
WRONG	161
X	44
Y	45
YELLOW	192
YES	157
YOU	201
YOU'RE	82
YOUR	81
Z	46
ZERO	0

Commodore Computers:
Commodore Business Machines, Inc
1200 Wilson Drive West Chester, PA 19380
Commodore Business Machines, Limited
3370 Pharmacy Avenue Agincourt, Ontario, M1W 2K4

Printed in Hong Kong