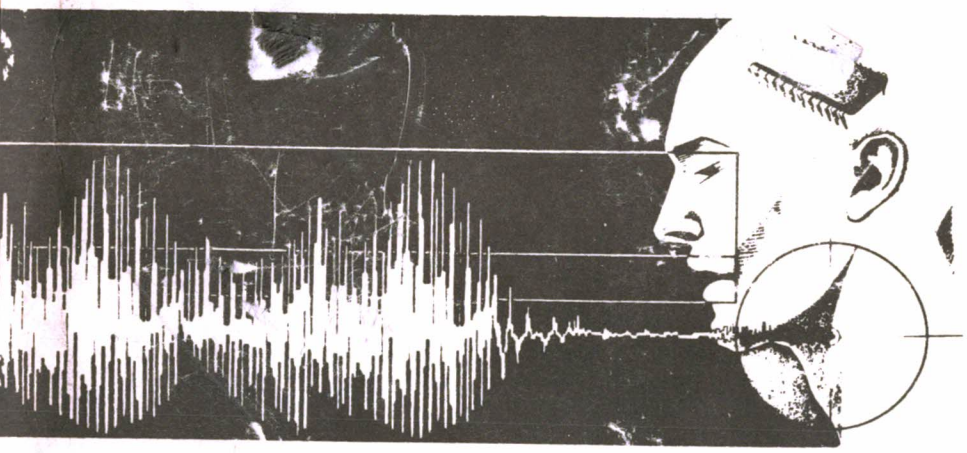
HEAR THE POWER...



TEXT TO SPEECH SYNTHESIZER

INSTRUCTION MANUAL

COMMODORE 64
VIC 20

Manufactured By:

R.esearch I.n S.peech T.echnology

Com TALKER 20/64 SPEECH SYNTHESIZER

Dear Customer:

Enclosed is the ComTALKER 20/64 Text-to-Speech synthesizer for your VIC 20 or Commodore 64. Also enclosed is the speech synthesis instruction manual and the Exclusive Phrase Finder Program resident on cassette tape or disk. This manual contains the operating instructions for your ComTALKER together with instructions and program listings for the Exclusive Phrase Finder Program. To load the Exclusive Phrase Finder Program, type:

LOAD"VIC" for the VIC 20 program on cassette tape.
LOAD"VIC",8 for the VIC 20 program on disk.
LOAD"64" for the C64 program on cassette tape.
LOAD"64",8 for the C64 program on disk.

This manual also includes a Sample Program explaining how you can add speech to your existing programs. A few chapters on speech synthesis theory are included for your reading pleasure. They are Theory of Speech Synthesis, Speech Reproduction Techniques, The Phoneme/Allophone Approach and an Introduction to Text-to-Speech. The last chapter of this manual includes the directions for the VIC 20, C64 Text-to-Speech program. This program has a number of features, as explained in the chapter, and may be purchased with this unit or at a later date. Please note, however, that the Text-to-Speech program requires the ComTALKER 20/64 text to speech synthesizer to operate.

Yours truly,

R.I.S.T. Inc.

TABLE OF CONTENTS

PREFACE

TALKING COMPUTERS. Why is there a growing interest in adapting speech to the home and personal computer?

TALKING. Defined in Webster's dictionary as the power to communicate to an individual or group of individuals. This ability to put ideas into words, has become the most natural form of human communication in the world today.

COMPUTERS. Defined in Webster's dictionary as an electronic machine that performs rapid calculations and compiles or correlates information.

TALKING COMPUTERS. Combining the two, therefore, offers the advantage of communicating information to an individual or group of individuals via the spoken word.

WHY AN INTEREST IN TALKING COMPUTERS?

The computer has just taken the next step closer to becoming a "friendly" computer. A computer that can communicate with its owner more naturally, has a certain personality and is easier to use. Talking computers make this all possible.

Until recently talking computers existed only in the fantasy of science fiction, or at a great cost in industrial and university laboratories. Since then speech synthesis units have appeared commercially at a price range of $250 to $350. This manual explains how you can adapt your own

speech synthesis unit, the ComTALKER 20/64, to your Commodore 64 or Vic 20.

In addition to the tremendous cost savings, it will give you an introduction to speech synthesis theory and help you understand the basic speech concepts. The applications described will show you how speech synthesis can be a useful tool for your programming needs.

Many texts have been written regarding speech synthesis theory, however, this is the first manual of its kind to bring a complex technology down to a level that any computer owner can understand. It is written in clear, concise layman's terms. It not only offers a step by step procedure in operating a speech unit for your home computer but also educates you in this exciting new technology. It's true, you can purchase other synthesis products and other texts but neither offer both advantages just stated.

As stated above, this manual is written to be both educational and informative. It describes how the human speech mechanism works, and the simple physical model that can be used to represent it. It is interesting to see the similarities between the two, which is, of course the reason why computers can now talk. The manual also provides complete detailed step by step instructions in operating a speech synthesis unit for the Commodore line of computers. Because this manual is intended as an introduction to speech

synthesis using home computers it includes software examples that illustrate the fundamental principles and techniques. It also provides sample programs that allow you to add speech to your own programs. In this way, you the reader can explore and invent with this exciting new aspect of computing on your own, given good tools to start with. A dictionary of over 250 most commonly used words is supplied to allow you to build your phrases easily.

The manual is formatted in such a way to be useful for the hobbyist, designer, student, or any computer owner interested in giving his computer the power of speech. It can be regarded as an introduction to speech synthesis for the beginner as well as technical reference for the advanced hobbyist. No previous experience is required in speech synthesis theory.

However, if further technical information is required, sections labeled "Let's Get Technical. . ." are included at the end of each chapter. These sections are intended for readers with some background in speech synthesis theory.

As stated, the manual gives you the tools you need to create speech applications on your own. You are limited only by your imagination. The power of speech is now within your reach.

INTRODUCTION

This manual is divided into seven chapters that present speech synthesis concepts followed by fully tested hardware and software programs.

Chapter One describes the speech hardware interface to the Commodore computers. Sample programs are given to both test the hardware and illustrate the fundamental computer generated speech principles. Directions that explain how to add speech to your own software programs are included. Program listings and operating instructions are all included.

Chapter Two discusses the speech synthesis technique chosen to give your computer an unlimited vocabulary. It explains in detail how sounds are classified and how these classifications are used when synthesizing speech. This chapter also explains, using examples, how your synthesizer can make your computer talk.

Chapter Three discusses how the human vocal tract constructs sounds that we interpret as speech. It explains how speech begins at the vocal cords as sounds and how these sounds are transformed into speech. By understanding the basic theory of speech production one can understand how computers can talk.

Chapter Four describes speech synthesis techniques in common use today. It also explains some of the advantages and disadvantages of each technique as related to the home computer.

Chapter Five introduces you to an area known as Text-to-Speech. Text-to-Speech is an "add on" software program that requires the speech synthesizer, ComTALKER 20/64. This chapter explains what Text-to-Speech is, why it is useful in certain types of speech applications and explains its advantages and disadvantages.

Chapter Six contains a dictionary of over 250 most commonly used words. The dictionary lists each word and its sounds used to create that word. The dictionary is an extremely useful tool when creating words and phrases for your speech synthesizer.

Chapter Seven describes the features of the Text-to-Speech program designed by R.I.S.T. Inc. It also includes all the operating instructions for use of the Text-to-Speech system resident on either cassette tape or disk. Please note ComTALKER 20/64 is required for this program to operate.

Glossary of Terms. Before each chapter a list of technical terms used in that chapter, with their definitions is explained. Written in this fashion, each

chapter becomes easier to understand because you are sure to be familiar with key words encountered while reading the chapter.

After reading the manual, you will have taken the first step in understanding the basic concepts of speech synthesis and its use with the home computer. It gives you the ability to make your computer into a "Talking Computer". By utilizing the information explained in the manual, you can now experiment with speech applications on your own. A few suggestions to experiment with are:

GAME APPLICATION PROGRAMS. Fierce competition between you and your computer can now be guided and accentuated with speech. Have your space games tell you when your running low on fuel, keep your eyes on the game instead of the clock, have the synthesizer count down your time. Improving your score on your game programs is inevitable.

EDUCATIONAL PROGRAMS. Generate a program for your child, bringing the overwhelming simplicity of listening and learning. Separating the sounds in large and small words for your children to hear will prove to be a vital asset in their learning experience. Adding speech in this ever demanding world of visual aids, introduces the most important of your senses to the educational applications of your computer.

HOUSEHOLD APPLICATIONS. Relieve the pressure of "one eye on the clock". Program your computer to give you a call when your on a tight schedule and your time in the shower is running out; call you when your roast is ready; tell you when your favorite TV show is about to come on. The possibilities are endless in solving everyday problems. For your home security system; have the synthesizer announce the location of the disturbance and the safest exit from the house. Use your computer as a telephone answering machine and have the synthesizer inform the caller that you are unable to answer the phone and to please leave a message.

HANDICAPPED APPLICATIONS. Your computer can also give speech to the handicapped. This in my opinion is one of the major assets of this new technology.

CHAPTER ONE: HARDWARE/SOFTWARE INSTRUCTIONS

NOTE: Refer to Figure 7-3 for ComTALKER 20/64 Descriptions

PORT AND POWER REQUIREMENTS

Vic 20, Commodore 64. This board plugs directly into the user port of the computer. The user port is the port on the left hand side of the computer; the port opposite the game cartridge port. The Vic 20 always requires external power. The Commodore 64 supplies the bus with 250mA. This is sufficient to drive the speech synthesizer. However, if another peripheral is used internal power must be disconnected and external power applied.

1. Reset Button - Depressing this button readies the synthesizer for operation. It will also cause the board to stop talking.

2. Volume Control - Turning this control to the right or left will increase or decrease the volume of the synthesizer.

3. Speaker Jack - This connection will drive any 4 or 8 ohm speaker or act as an auxiliary input to any receiver.

4. Power Jack - Optional power connection to be used for expanded system operation only (see Power Requirements in the beginning of this chapter). This allows you to supply the additional power required if the other modules are being used. The power supply required is a 9v, 300 mA supply, mini jack, the tip is positive. CAUTION: When using external power the INTERNAL/EXTERNAL switch (5) should be in the EXTERNAL position. This disables the internal power circuit.

5. INTERNAL/EXTERNAL switch - While in the INTERNAL position, the synthesizer board is powered internally from the Commodore 64 computer. This is disconnected in the EXTERNAL position.

6. 24 Pin Dual Edge Connector - This connector is the interface between the synthesizer and the user port of the Vic 20/Commodore 64.

Operating Procedure

A. Using the synthesizer with internal power (external power supply not required).

Step 1. Switch the INTERNAL/EXTERNAL switch to the EXTERNAL position. This disables the internal power circuit.

Step 2. Plug the speaker into the speaker jack (3).

CAUTION: THE COMPUTER MUST BE OFF

Step 3. Plug the synthesizer into the 24 pin bus (User Port) of the Vic 20/Commodore 64.

CAUTION: When plugging in the 24 pin connector into the USER PORT insure that the side of the connector with ONE wire is on TOP and the 9 wires are on the BOTTOM.

Step 4. Turn your computer on.

Step 5. Turn the synthesizer switch to the INTERNAL position.

Step 6. Depress the reset button (1).

NOTE: A faint "click" should be heard in the speaker when depressing the button. If this sound is not heard, turn up the volume and try it again.

Step 7. Proceed to the Software Test Procedures.

B. Using the synthesizer with external power.

Step 1. Switch the INTERNAL/EXTERNAL switch to the EXTERNAL position.

Step 2. Plug the speaker into the speaker jack (3).

Step 3. Plug the external power supply into the power jack (4) of the synthesizer board (the tip is positive).

NOTE: DO NOT PLUG INTO WALL OUTLET YET

CAUTION: THE COMPUTER MUST BE OFF.

Step 4. Plug the synthesizer into the 24 pin bus (User Port) of the Vic 20/Commodore 64.

CAUTION: When plugging in the 24 pin connector into the USER PORT insure that the side of the connector with ONE wire is on TOP and the 9 wires are on the BOTTOM.

Step 5. Plug in the 9v power adapter to the synthesizer into the wall outlet.

Step 6. Turn your computer on.

Step 7. Depress the reset button (1).

NOTE: A faint "click" should be heard in the speaker when depressing the button. If this sound is not heard, turn up the volume and try it again.

Step 8. Proceed to the Software Test Procedures.

SOFTWARE TEST PROCEDURES

After carefully wiring up your speech synthesizer board, the following commands can be used to test your circuit before loading in the respective programs. Before power is applied, visually inspect your hardware to ensure that the proper connections have been made

and all the grounds are secure. On power up, a hardware reset is required by simply closing the switch momentarily. A click or pop should be heard in the speaker. If this occurs, proceed to Software Test Procedures.

Software Test Procedures

Vic 20. When testing this circuit you must first set the User Port to accept data. To do so, enter the statement POKE 37136,255. Then enter the statement POKE 37138,127. This brings all the data lines including the ALD pulse high. The next statement to be entered is POKE 37136,5. This will output address 5 ("OY") on the data bus of the speech chip and bring $\overline{\text{ALD}}$ low. POKE 37136,69 will bring $\overline{\text{ALD}}$ high again causing the synthesizer to talk. If everything is correct, the board will continue to speak until a pause is entered. To enter a pause and silence the board, enter POKE 37136,0 and 37136,64. This is a pause at location 000. You are now ready to load in your exclusive program (Table 7-1) in the usual manner as specified in your computer manual and create your own phrases.

Commodore 64. When testing this circuit you must first set the User Port to accept data. To do so, enter the statement POKE 56577,255. Then enter the statement POKE 56579,127. This brings all the data lines including the $\overline{ALD}$ pulse high. The next statement to be entered is POKE 56577,5. This will output address 5 ("OY") on the data bus of the speech chip and bring $\overline{ALD}$ low. POKE 56577,69 will bring $\overline{ALD}$ high again causing the synthesizer to talk. If everything is correct the board will continue to speak until a pause is entered. To enter a pause and silence the board, enter POKE 56577,0 and 56577,64. This is a pause at location 000. You are now ready to load in your exclusive program (Table 7-1) in the usual manner specified in your computer manual and create your own phrases.

SOFTWARE DESCRIPTION

The program listed in Table 7-1 of this section allow you to build words and phrases from their constituent allophones. The phrase can be edited by moving a pointer left or right to the desired position. Inserting, deleting, or replacing allophones can then be accomplished as desired. When the phrase is prepared to your satisfaction, a simple ENTER, NEW LINE or RETURN will cause the synthesizer to talk.

The available commands are:

"phoneme strings" causes named allophone to be added to the phrase at the current position, either replacing the existing allophone or inserting one before it (see Insert section).

"L" Moves the position pointer left one allophone.

"R" Moves the position pointer right one allophone.

"D" Deletes allophone at the current position pointer one at a time.

"XL" Moves the position pointer "X" number of spaces to the left.

"XR" Moves the position pointer "X" number of spaces to the right.

"I" Turns on "Insert Mode". The next allophone entered will be inserted into the phrase at the current position. Additional allophones will be inserted until "I" is entered again. The second "I" command will turn "Insert Mode" off. When "Insert mode" is off, an entered allophone will replace the

one at the current position. This is the default at system startup.

NEW LINE, ENTER or RETURN — Causes the system to output to the hardware the commands necessary to pronounce the phrase.

"E" — Exits the program.

Program Description

At system start up a brief message will be spoken and the following commands are performed.[1] The screen is cleared and all variables are initialized. The allophone symbol array is itialized with the 64 two or three character symbols that represent each allophone. These are strings the user will enter in order to add an allophone to the phrase (see Table 3-1 or Table 9-1). The position pointer is at position one, and the user is prompted for input with a "→". At this prompt, an allophone or any of the above commands may be entered. An invalid allophone will be flagged as an error, as will attempting to move the position pointer left or

[1]Note: This message will be spoken each time the program is RUN. To delete this message from the system the following commands must be performed:

Insert line 125 GOTO 190 into the program.

right beyond the boundaries of the phrase. After each command, the updated phrase is displayed with the current position indicated by "→".

For example: At system startup, the screen will look like this:

Command	Screen	Comments
RUN	→ ?	Pointer at position one.
HH1	→ ?HH1	Desired Allophone.
ENTER	HH1→ ?	The first allophone has been entered; the pointer is at position two; the system is ready for the next input.
EH1	HH1→ ?EH1	Next allophone.
ENTER	HH1→ ?EH1 ***Invalid Entry***	User entered invalid data.
EH	HH1→ ?EH	
ENTER	HH1 EH→	The second allophone has been entered; the pointer is at position three; the system is ready for the next input.

Editing Features

Upon entering a string of allophones and noticing that a few corrections are in order, the following edit commands are useful.

First, we must position the pointer at the location where an editing command is to be performed. Let's

take the word "hello" for example. The screen should now look like this:

HH1 EH LL UW1→

Realizing that the UW1 allophone is incorrect, we would like to REPLACE it with "OW". To do so, the following commands are required:

To Replace

Command	Screen	Comments
"L"	HH1 EH LL UW1→ L	We have to move the pointer left one space to replace UW1.
"ENTER"	HH1 EH LL→ UW1	The system is now ready to replace the allophone "UW1".
"OW"	HH1 EH LL→ UW1 OW	The desired replacement allophone is typed.
"ENTER"	HH1 EH LL OW→	After pressing ENTER, the allophone has been replaced.

Moving the position pointer right works in the same manner as moving to the left, using the left command. The only exception is that we use an "R" instead of an "L". These commands move the pointer one space at a time.

In replace mode (the default at program start up) the new allophone will replace the allophone at the current position in the phrase.

Note: If you attempt to move the position pointer left or right beyond its boundaries, the message ***INVALID ENTRY*** will appear.

Command	Screen	Comments
"R" ENTER	HH1 EH LL OW→ R ***Invalid Entry***	To clear the invalid entry, either press ENTER or type in a valid command.
"L" ENTER	→HH1 EH LL OW L ***Invalid Entry***	

The XL and XR commands work in the same manner as the L and R commands. The only difference is that these commands move the position pointer "X" number of spaces to the left or right. Once again exceeding the boundaries will prompt an "***INVALID ENTRY***" message.

Example:

Command	Screen	Comments
"2L"	HH1 EH LL OW1→	
ENTER	HH1 EH→ LL OW	The position pointer has moved two spaces to the left.

Deleting an allophone

We'll use the same examples as above. Remember, we must first position the pointer to the specified allophone to be deleted. Once this is accomplished, the following commands are required:

Command	Screen	Comments
"L" ENTER (3 times)	HH1 EH EH LL OW→	We need to delete an "EH" here, so first we must move the pointer 3 spaces to the left.
"D"	HH1 EH→ EH LL OW D	The pointer is now positioned to the allophone to be deleted.
"ENTER"	HH1 EH→ LL OW	The "EH" allophone has been deleted.

Note: The delete command "D" deletes one allophone at a time.

Inserting an allophone

After creating your allophone phrases, and realizing that a few pauses (or other allophones) need to be inserted, the following command sequence must be performed. Let's use the word "CHATTER" as an example.

Command	Screen	Comments
"L" ENTER (3 times)	CH AE TT2 ER1 PA3→ ?	We must position the pointer at the location where the inserted allophone will go.
"I"	CH AE→ TT2 ER1 PA3 ?I	We are ready to turn on the "insert mode".
"ENTER"	CH AE→ TT2 ER1 PA3 ?	Insert mode has been turned on.
"PA3"	CH AE→ TT2 ER1 PA3 ? PA3	The desired allophone to be inserted is typed.
"ENTER"	CH AE PA3→ TT2 ER2 PA3 ?	The allophone has been inserted. Additional allophones may be entered at this time if required.

Command	Screen	Comments
"I"	CH AE PA3→ TT2 ER2 PA3 ?I	We are ready to turn off the "insert mode".
"ENTER"	CH AE PA3→ TT2 ER2 PA3 ?	Insert mode has been turned off.

Making Your Computer Talk

A simple "ENTER", "RETURN" OR "NEW LINE" is all that's required to make the system talk. If a new allophone string is desired, you must first EXIT (E) the program then RUN it again. This will clear all the previous allophone codes stored. Unfortunately at the time of writing this book, a "SAVE" routine was not supplied with the exclusive phrase finder program. You may want to try writing your own subroutine to save the allophone strings you've created with the exclusive phrase finder program, in separate files. An alternative method is to save the exclusive phrase finder program on cassette tape or disk, with the phrase or word you've just created. When reloading the program, the following commands are necessary for it to speak the prestored phrase.

Vic 20 GOTO 4999

Commodore 64 GOTO 4999

Upon typing this command, the computer will speak the prestored allophone string and the allophone codes will

be printed on the screen. The system is now ready to edit the existing string. To enter a new string RUN the program.

Table 7-1 - Exclusive Phrase Finder Program Listing
for the Vic 20/Commodore 64

```
  1     FIRST = 1
  5     LET LE=128
 10     DIM A$(64)
 20     DIM E(LE)
 30     LET P=0
 40     LET LN=0
100     FOR X=1 TO 64
110     READ A$(X)
120     NEXT X
130     LN=52
140     FOR X = 1 TO LN
150     READ E(X)
160     NEXT X
170     GOTO 5000
190     REM *UPDATE SREEN*
200     PRINT CHR$(147)
205     IF FIRST THEN FIRST = 0 : LN = 0
210     IF P=0 THEN 250
220     FOR X=1 TO P
230*    PRINT A$(E(X)) ; "b";
240     NEXT X
250*    PRINT "→"; "b";
260     IF P=LN THEN PRINT: GOTO 292
270     FOR X=P+1 TO LN
280*    PRINT A$(E(X)) ; "b";
290     NEXT X
292     PRINT
295     IF M=1 THEN PRINT "***INSERT MODE***"
297     LET J=1
299     REM *INPUT COMMAND OR ALLOPHONE*
300     LET I$="": INPUT I$
310     IF I$=""  THEN 5000
320     IF I$="I" THEN  500
330     IF I$="D" THEN  600
340     IF I$="L" THEN  700
350     IF I$="R" THEN  800
360     IF I$="E" THEN  STOP
365     IF ASC(I$)<58 THEN 1000
367     REM *DECODE ALLOPHONE*
370     FOR X=1 TO 64
380     IF I$=A$(X) THEN 900
390     NEXT X
400     PRINT "***INVALID ENTRY***"
410     GOTO 300
```

*b represents a space

```
 490     REM *TOGGLE INSERT MODE*
 500     LET M=ABS(M-1)
 510     GOTO 200
 590     REM *DELETE ALLOPHONE*
 600     IF P=LN THEN 400
 610     FOR X=P+1 TO LN
 620     IF X=LE THEN 640
 630     LET E(X)=E(X+1)
 640     NEXT X
 650     LET LN=LN-1
 660     IF P>LN THEN P=LN
 670     GOTO 200
 690     REM *MOVE LEFT*
 700     IF P-J<0 THEN 400
 710     LET P=P-J
 720     GOTO 200
 790     REM *MOVE RIGHT*
 800     IF (P+J>LN) OR (P+J>LE) THEN 400
 810     LET P=P+J
 820     GOTO 200
 890     REM *ADD ALLOPHONE TO STRING*
 900     IF LN=LE THEN 400
 910     IF M=0 THEN 960
 920     FOR Y=LN TO P+1 STEP -1
 930     LET E(Y+1) = E(Y)
 940     NEXT Y
 950     LET LN=LN+1
 960     LET P=P+1
 970     LET E(P)=X
 980     IF P>LN THEN LN=P
 990     GOTO 200
 999     REM *GET NUMBER OF SPACES TO MOVE LEFT OR RIGHT*
1000     LET J=VAL(I$)
1010     IF RIGHT$(I$,1) = "R" THEN 800
1020     IF RIGHT$(I$,1) = "L" THEN 700
1030     GOTO 400
4999     REM *SPEAK* FOR THE VIC 20
5000     POKE 37136,255
5010     POKE 37138,127
5020     FOR X=1 TO LN
5030     WAIT 37136,128,128
5040     POKE 37136, E(X)-1
5050     POKE 37136, E(X)+63
5060     NEXT X
5070     WAIT 37136,128,128
5080     POKE 37136,0
5090     POKE 37136,64
5100     GOTO 200
6999     REM *DATA*
```

```
7000      DATA PA1,PA2,PA3,PA4,PA5,OY,AY,EH,KK3,PP,
          JH,NN1,IH,TT2,RR1,AX,MM,TT1,DH1,IY,EY
7010      DATA DD1,UW1,AO,AA,YY2,AE,HH1,BB1,TH,UH,
          UW2,AW,DD2,GG3,VV,GG1,SH,ZH,RR2,FF,KK2,KK1
7020      DATA ZZ,NG,LL,WW,XR,WH,YY1,CH,ER1,ER2,
          OW,DH2,SS,NN2,HH2,OR,AR,YR,GG2,EL,BB2
9000      DATA 28,8,46,54,5,5,5,7,3,43,27,12,3,37,13,
               36,3,30,20,3,10,33,52,3
9100      DATA 16,16,36,3,56,1,10,20,1,51,3,14,32,3,
               26,59,3 ,43,16,17,1
9200      DATA 10,50,23,1,14,52,5
```

For Commodore 64 replace Lines 5000-5100 with the following:

```
4999      REM *SPEAK* FOR THE COMMODORE 64
5000      POKE 56577,255
5010      POKE 56579,127
5020      FOR X=1 TO LN
5030      WAIT 56577,128,128
5040      POKE 56577, E(X)-1
5050      POKE 56577, E(X)+63
5060      NEXT X
5070      WAIT 56577,128,128
5080      POKE 56577,0
5090      POKE 56577,64
5100      GOTO 200
```

Sample Program

The following program describes how to add "N" phrases to your existing programs.

Data Statements

These statements must appear in the program before the lines that enable the synthesizer to speak.

```
100       LET S$(1) = CHR$(*)+....+CHR$(64)
110       LET S$(2) = CHR$(*)+....+CHR$(64)
120       LET S$(3) = CHR$(*)+....+CHR$(64)
130       LET S$(N) = CHR$(*)+....+CHR$(64)
```

*NOTE: The decimal codes for each particular allophone as shown in Table 9-1 must be inserted in these places.

Subroutine

```
10000*          B = 37136
10010           POKE B,255
10020           POKE B+2,127
10030           N = 1
10040           S=ASC(MID$(S$(X),N,1))
10050           POKE B, MOD(S,64)
10060           POKE B, MOD(S,64) + 64
10070           N = N + 1
10080           IF S<64 THEN 10040
```

*For Commodore 64 B = 56577

First Phrase to be Spoken

These lines must appear in the program directly after the line when you want the synthesizer to speak the first phrase.

```
1000        LET X=1
1010        GOSUB 10000
```

Second Phrase to be Spoken

These lines must appear in the program directly after the line when you want the synthesizer to speak the second phrase.

```
2000        LET X=2
2010        GOSUB 10000
```

Third Phrase to be Spoken

These lines must appear in the program directly after the line when you want the synthesizer to speak the third phrase.

```
3000        LET X=3
3010        GOSUB 10000
```

Nth Phrase to be Spoken

These lines must appear in the program directly after the line when you want the synthesizer to speak the Nth phrase.

```
N000      LET X=N
N010      GOSUB 10000
```

LET'S GET TECHNICAL. . .

Driver Subroutines

This next section describes the driver, or talk subroutines, for the Commodore 64 and Vic 20. These drivers subroutines can be used directly in your own application programs, or studied as examples.

Vic 20/Commodore 64

Vic 20

```
5000      POKE 37136,255
5010      POKE 37138,127
5020      FOR X = 1 TO LN
5030      WAIT 37136,128,128
5040      POKE 37136,E(X)-1
5050      POKE 37136,E(X)+63
5060      NEXT X
```

Commodore 64

```
5000      POKE 56577,255
5010      POKE 56579,127
5020      FOR X = 1 TO LN
5030      WAIT 56577,128,128
5040      POKE 56577,E(X)-1
5050      POKE 56577,E(X)+63
5060      NEXT X
```

*Explanation of Program: Vic 20

*For Commodore 64 the POKE statements should be as listed above.

Line 5000 POKE 37136,255

This sets all output ports to a Logic 1.

Line 5010 POKE 37138,127

This sets the higher bit (ALD) to a Logic 1 and the lower bits (Data lines and LRQ) to a Logic 0.

Line 5020 FOR X = 1 TO LN

Selects first allophone.

Line 5030 WAIT 37136,128,128

The system waits for LRQ to go low in order to speak the first allophone. Other allophones cannot be entered at this time.

Line 5040 POKE 37136,E(X)-1

Pokes the first allophone onto data lines and brings ALD low.

Line 5050 POKE 37136,E(X)+63

Brings ALD high, completing the pulse and causing the system to speak first allophone.

Line 5060 NEXT X

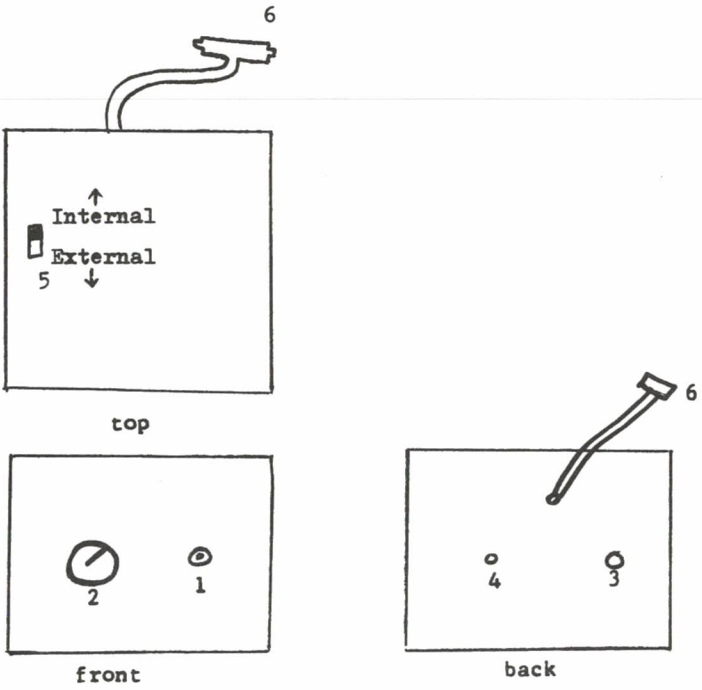
Next allophone to be spoken.

TABLE 9-1
ALLOPHONE ADDRESS TABLE

DECIMAL CODES	ALLOPHONE	SAMPLE WORD	DURATION
0	PA1	PAUSE	10 ms
1	PA2	PAUSE	30 ms
2	PA3	PAUSE	50 ms
3	PA4	PAUSE	100 ms
4	PA5	PAUSE	200 ms
5	/OH/	Boy	420 ms
6	/AY/	Sky	260 ms
7	/EH/	End	70 ms
8	/KK3/	Comb	120 ms
9	/PP/	Pow	210 ms
10	/JH/	Dodge	140 ms
11	/NN1/	Thin	140 ms
12	/IH/	Sit	70 ms
13	/TT2/	To	140 ms
14	/RR1/	Rural	170 ms
15	/AX/	Succeed	70 ms
16	/MM/	Milk	180 ms
17	/TT1/	Part	100 ms
18	/DH1/	They	290 ms
19	/IY/	See	250 ms
20	/EY/	Beige	280 ms

DECIMAL CODES	ALLOPHONE	SAMPLE WORD	DURATION
21	/DD1/	Could	70 ms
22	/UW1/	To	100 ms
23	/AO/	Aught	100 ms
24	/AA/	Hot	100 ms
25	/YY2/	Yes	180 ms
26	/AE/	Hat	120 ms
27	/HH1/	He	130 ms
28	/BB1/	Business	80 ms
29	/TH/	Thin	180 ms
30	/UH/	Book	100 ms
31	/UW2/	Food	260 ms
32	/AW/	Out	370 ms
33	/DD2/	Do	160 ms
34	/GG3/	Wig	140 ms
35	/VV/	Vest	190 ms
36	/GG1/	Guest	80 ms
37	/SH/	Ship	160 ms
38	/ZH/	Azure	190 ms
39	/RR2/	Brain	120 ms
40	/FF/	Food	150 ms
41	/KK2/	Sky	190 ms
42	/KK1/	Can't	160 ms
43	/ZZ/	Zoo	210 ms
44	/NG/	Anchor	220 ms

DECIMAL CODES	ALLOPHONE	SAMPLE WORD	DURATION
45	/LL/	Lake	110 ms
46	/WW/	Wool	180 ms
47	/XR/	Repair	360 ms
48	/WH/	Whig	200 ms
49	/YY1/	Yes	130 ms
50	/CH/	Church	190 ms
51	/ER1/	Fir	160 ms
52	/ER2/	Fir	300 ms
53	/OW/	Beau	240 ms
54	/DH2/	They	240 ms
55	/SS/	Vest	90 ms
56	/NN2/	No	190 ms
57	/HH2/	Hoe	180 ms
58	/OR/	Store	330 ms
59	/AR/	Alarm	290 ms
60	/YR/	Clear	350 ms
61	/GG2/	Got	40 ms
62	/EL/	Saddle	190 ms
63	/BB2/	Business	50 ms



CAUTION: When plugging in the 24 pin connector into the USER PORT insure that the side of the connector with ONE wire is on TOP and the 9 wires are on the BOTTOM.

Figure 7-3

CHAPTER TWO: THE PHONEME/ALLOPHONE APPROACH

GLOSSARY OF TERMS USED IN THIS CHAPTER

Phoneme

The smallest unit of speech that is used to distinguish meanings between words.

Allophone

Variations of particular phoneme sounds that depend on its position in a word.

Articulatory phonetics

The process of making speech sounds or phonemes.

Acoustic phonetics

The physical characteristics of a speech signal.

Coarticulation

The blending together of two allophones.

Although PCM and LPC methods of encoding speech sounds discussed in Chapter Two give very high-quality speech, the memory requirements to store a number of words is very large. Use of these processes to store synthesized speech as whole units, (i.e., words, sentences), make the storage of a large vocabulary very expensive. Using these processes to store sub-word units (i.e., syllables, half syllables, etc.), which can be combined to make whole words, large vocabularies can be stored in a small amount of memory. The words formed by combining these subword units will typically

sound somewhat less understandable than words which are stored as whole units. However, if the set of sub-word units is designed properly, any word in a given language can be constructed from this set.

The smallest units of speech are called PHONEMES. These are actually smaller than syllables and even half syllables.

For Example - the word "bee" is a monosyllabic (one syllable) word. However, the syllable "bee" is actually composed of two phonemes. The first phoneme is the sound of the letter "B" and has been named "/b/". The second phoneme in the word "bee" is the sound of the letters "EE" and is what we refer to as the long "E" (e) sound. This sound is equivalent to the "EA" sound in "eat" and is named "/e/".

Although a phoneme rarely appears as an entire word (e.g., "a" as in a lamp) a change in a single phoneme can completely change one word into another.

For Example:

Word	Phonemes
bee	/b/ /e/
pea	/p/ /e/

The phonetic symbols /p/ and /b/ are sufficiently different to signal a difference in meaning between the words "pea" and "bee", however, the phonemes /p/ and /b/ mean nothing by themselves. Therefore, a phoneme

does not typically have meaning but is used to distinguish meanings between words. This process of making speech sounds is known as articulatory phonetics.

To compare with the above example, another word which uses the phonemes /b/ and /e/ is "bleed". In theory, the sounds /b/ and /e/ should sound exactly the same as the /b/ and /e/ sounds in "bee". However, the sounds we actually make when speaking are affected by the sounds that precede and follow the particular sound. In this case, the /b/ as well as the /e/ is affected by the sound of the letter "L". It should also be noted that the /b/ and the /e/ affect the sound of the letter "L". The phonemes can be thought of as "blending together" at their "edges". This blending together does not create the addition of new phonemes. It actually creates a variation of the phonemes. These variations of a particular phoneme are known as ALLOPHONES.

Let's use our same example:

Word	Phonemes
bee	/b/ /e/
bleed	/b/- /e/-

Since the sounds are slightly different from each other when pronouncing such individual word, the allophones may be represented as follows:

sounds). For example, if you try to extract the "b" sound from the word brain by taking larger portions of the acoustic signal from the beginning of the word, one would encounter a non-speechlike noise and then the sound "br". There is no point at which the "b" sound can be heard in isolation. Due to this "coarticulation effect", information is lost when extracting these sounds and the speech quality is reduced somewhat. Methods of improving this overlapping of sounds are discussed in the Let's Get Technical section at the end of this chapter.

When designing a set of sub-word units, as mentioned above, taking this "coarticulation effect" into account will result in more natural sounding speech. Each language has a set of sub-word units which is slightly different from that of other languages. There are approximately 42 phonemes in the English language. The speech synthesizer described in Chapters Four through Eight uses a set of 59 allophones derived from these 42 phonemes. From this set we can produce any word in the English language.

HOW TO CREATE WORDS FROM BASIC SOUNDS

We have just discussed a speech synthesis method known as allophone synthesis which can "Make Your Computer Talk". Utilizing these basic sounds of the

English language an unlimited vocabulary can be created for your personal computer. However, due to the "coarticulation" effects of allophone synthesis described above, the unlimited vocabulary as been obtained at the expense of speech that is not as natural or smooth as compared to LPC or PCM. Allophone synthesis also requires familiarity with the speech sounds of English which form the words, which are quite different from the letters that are used to represent the words. The symbols used to represent the allophones must also be studied. Table 3-1 gives a detailed set of guidelines for using the allophone set with your speech synthesizer described in the hardware/software section of this manual.

For a further explanation of their classifications (i.e., vowels, resonants, etc.) see the Let's Get Technical section entitled General Phoneme Classifications.

The first column of Table 3-1 represents the allophone names (symbols) which will be typed into your computer to generate that particular sound.

The second column gives sample words and shows how

the allophone sounds are used in context. Here you can understand the actual sound the symbol represents.

The third column represents the duration of each sound or allophone. For some phonemes there are two allophones to account for the initial and final position. In final position, stop consonants are usually unreleased. For example, when pronouncing words such as rib, played and peg, the final stop consonants (b,d,g) are shortened or not fully pronounced (unreleased) because they are not followed by other phonemes. For this reason, when using a stop consonant in the final position of a word, an allophone with a shorter duration is required. As a result, an allophone designed for an initial position may sound too loud or strong in the final position and vice-versa. Notice that the initial version of some allophones are longer than the final version (see Table 3-1).

Example:

	Allophone	Duration
Initial	DD1	160 ms
Final	DD2	70 ms
Initial	DH1	290 ms
Final	DH2	120 ms

The allophones marked with a single asterisk (*) can be doubled or tripled. This means that these allophones may be joined together in succession. This feature is only incorporated in selected allophones. Therefore, to create an initial "s" you can use "ss,ss" as opposed to one "ss" at the end of a word. This can also be accomplished with the "th", "ff", and the short vowels. Other phonemes may appear as three different allophones. These allophones are also used for different vowel contexts.

Studies have shown that stressed syllables are higher in amplitude and pitch and longer in duration than unstressed. Duration is the more prominent cue to stress, when compared with amplitude. Due to this fact, a syllable will sound stressed if its vowel is lengthened. For this reason, it is useful to double short vowels when stress of a particular sound is required. For example, in the word "is" (IH-ZZ), you may want to double the "IH" allophone for increased stress, IH - IH - ZZ. You can also create differences between words like the noun "subject", which is stressed on the first syllable, and the verb "subject" which is stressed on the second syllable. This can be accomplished by using two "AX"s in the first syllable

of the noun and two "EH"s in the second syllable of the verb.

For Example:

Noun - subject	SS-SS-AX-AX-PA2-BB1-PA2-JH-EH-PA3-KK2-PA3-TT2
Verb - subject	SS-SS-AX-AX-PA2-BB1-PA2-JH EH-EH-PA3-KK2-PA3-TT2

Long vowels cannot be doubled but the "UW" allophone appears with two durations. The short one, "UW1" sounds good in words with many syllables after "YY", as in "computer". The long version, "UW2", is used in monosyllabic words as in two and food.

The column labelled R-Colored vowels contains allophones created from the vowel sound plus "R". The "ER" in particular contains two versions. The short one "ER1" is useful in words that end in "er" (letter, better). The long one, "ER2" is useful for monosyllabic words (fur, bird).

Several phonemes have allophones that were specifically designed to concatenate or join together with other phonemes. For example, TT1 was designed to be used in final clusters before "SS", as in "its" or "tests". Because of the coarticulatory effects of vowels on some consonants, different allophone consonants are required depending on the vowel context. GG1 is needed before allophone such as IY, YR, IH, or

EH, as in guest. NN2 is needed before allophones such as UH, OY, OR, or OW as in no.

A Word on Pauses

Some sounds, labeled as voiced stops, voiceless stops and affricates in Table 3-1, require a brief duration of silence before them. It has been shown that shortening the silent duration before a voiceless stop results in the perception of a voiced stop and the converse also holds true. Therefore, voiceless stops require a longer duration of silence or pause before them than voiced stops. So a PA1 or PA2 may be used before BB, DD, GG and JH while a PA3 be used before PP, TT, KK and CH. This will cause the following allophone to appear to be stressed somewhat. The allophones that require pauses before them appear in Table 3-1 and are denoted by a (†). You may need to change the duration of the pause a few times to make it correct, but don't get discouraged, because it will soon become an automatic process to you!

Take a look at the following sample words. This will give you an insight into allophone synthesis.

EXAMPLE WORDS CREATED FROM ALLOPHONES

HELLO	HH1-EH-LL-OW
I	AY
CAN	KK1-EH-PA1-NN1
GIVE	GG1-IH-VV
THE	TH-IY
POWER	PP-AW-ER1
OF	AX-VV
SPEECH	SS-PP-IY-CH
TO	TT2-UW2
ALL	AO-AO-PA1-LL
COMPUTERS	KK1-AX-MM-PA3-PP-YY1-UW1-PA3-TT2-ER1-ZZ
YOUR	YY2-OR
WISH	WW-IH-IH-SH
IS	IH-IH-ZZ
MY	MM-AY
COMMAND	KK2-AX-MM-AE-NN1-DD1

To create the word "computers", think of how it sounds, not the way it's spelled. Using Table 3-1, pick out the first sound, which is the "KK1" allophone. "KK1" was chosen because the next sound, when spoken slowly, sounds like the "a" sound in lapel. The allophone used to represent this sound is "AX" and "KK1" is used before "AX". The following sound is an /m/ sound. We will use the "MM" allophone. Now we

have "KK1-AX-MM" which represents the "com" in computers. Next we must find a /p/ sound. Look under voiceless stops, and you will find a "PP" sound as in trip. Because it is best to use a pause before voiceless stops, try adding a "PA3" before "PP". The following sound is a little tricky. One may think the next sound is a /u/ sound. Well that's only half correct. If you look under resonants, you will find a "YY1" sound and a "YY2" sound. Since the "YY1" sound is used in clusters as in "cute" and "beauty", this becomes the next allophone, because the word computer contains the cluster "pute". Now to continue the cluster we need the /u/ sound. You will find two choices under the long vowels. Notice UW1 is used after clusters with YY. Now, to continue the word we need a /t/ and /er/ sound. First insert another "PA3"; remember pause before a voiceless stop. Here, we have two possibilities, "TT1" or "TT2". Because the /t/ sound is not in a cluster with "SS", the "TT2" allophone will be used. The next sound is a vowel sound followed by an /r/ sound. Remember there were special allophones designed for specific use in this case. They are called the R-Colored vowels. Here we have two choices, once again. "Computers" is surely not a monosyllabic word, so the only choice left is

"ER1". The next sound is also a little tricky. Once again think of how the word sounds not the way its spelled. If you try an "SS" sound, you will notice this is incorrect. What the sound really is, is a /z/ sound. So let's end the word with the allophone /ZZ/. The final word should look like this:

KK1-AX-MM-PA3-PP-YY1-UW1-PA3-TT2-ER1-ZZ

Now let's try to create the word "wish". Here one may get confused with the "WW" sound or the "WH" sound. The correct sound is the "WW" sound. To fully explain why this is true would take a lengthy explanation. The position of the lips, and air being expelled, are two major reasons. For this situation it would be easier to try both of them. Whichever one sounds better to you is the one to use. The next sound is the short /i/ sound as in "sit". Under the short vowels there is an "IH" allophone. These short vowels are unique in that they can be stressed. This particular word requires a little stress, so double the "IH" allophone. (To hear the differences, try it both ways.) The final sound is the /sh/ sound as in ship. Under voiceless fricatives, you will notice the "SH" allophone. The completed word is:

WW-IH-IH-SH

For more examples of how to construct words from allophones, see Chapter SIX.

When constructing words from allophones, always remember, to think about how a word sounds, not how it is spelled. Although in some cases it may be obvious, in other cases it may not. It's obvious that an "NG" allophone belongs at the end of the words "song" and "long". It is not so obvious that it is used to represent the /n/ sound in "uncle". Furthermore, as you have already noticed, some sounds may not even be represented in words by any letters, like the "YY" in computers.

Please note that these are only suggestions, not rules. You may want to play with different sounds or different pauses to create a sound that is pleasing to your ear. (Don't be surprised if your synthesizer has your regional accent!) Speech synthesis is so subjective that what one person likes, another may not. Because allophone synthesis gives the user the ability to change sounds at will, it provides a rich environment for experimentation. So, go ahead and change a few sounds, the intent of this book is to teach you the fundamentals and basic concepts of allophone synthesis.

TABLE 3-1 ALLOPHONE GUIDELINES

ALLOPHONE	SAMPLE WORDS	DURATION
Silence		
PA1	- before BB, DD, GG, and JH	10 ms
PA2	- before BB, DD, GG, and JH	30 ms
PA3	- before PP, TT, KK, and CH and between words	50 ms
PA4	- between clauses and sentences	100 ms
PA5	- between clauses and sentences	200 ms
Short Vowels		
*/IH/	- sitting, stranded	70 ms
*/EH/	- extent, gentlemen, end	70 ms
*/AE/	- extract, acting, hat	120 ms
*/UH/	- cookie, full, book	100 ms
*/AO/	- talking, song, ought	100 ms
*/AX/	- lapel, instruct, succeed	70 ms
*/AA/	- pottery, cotton, hot	10 ms
Long Vowels		
/IY/	- treat, people, penny, see	250 ms
/EY/	- great, statement, tray, beige	280 ms
/AY/	- kite, sky, mighty	260 ms
/OY/	- noise, toy, voice, boy	420 ms
/UW1/	- after clusters with YY: computer	100 ms

TABLE 3-1 (Continued)

ALLOPHONE	SAMPLE WORDS	DURATION
/UW2/	- in monosyllablic words:	
	two, food	260 ms
/OW/	- zone, close, snow	240 ms
/AW/	- sound, mouse, down	370 ms
/EL/	- little, angle, gentlemen	190 ms
R-Colored Vowels		
/ER1/	- letter, furniture, interrupt	160 ms
/ER2/	- monosyllables: bird, fern, burn	300 ms
/OR/	- fortune, adorn, store	330 ms
/AR/	- farm, alarm, garment	290 ms
/YR/	- hear, earring, irresponsible	350 ms
/XR/	- hair, declare, stare	360 ms
Resonants		
/WW/	- we, warrant, linquist	180 ms
/RR1/	- initial position:	
	read, write, x-ray	170 ms
/RR2/	- initial cluster:	
	brown, crane, grease	120 ms
/LL/	- like, hello, steel	110 ms
/YY1/	- clusters: cute, beauty	
	computer	130 ms
/YY2/	- initial position:	
	yes, yarn, yo-yo	180 ms

TABLE 3-1 (Continued)

ALLOPHONE	SAMPLE WORDS	DURATION
Voiced Fricatives		
/VV/	- vest, prove, even	190 ms
/DH1/	- word-initial position: this, then, they	290 ms
/DH2/	- word-final and between vowels: bathe, bathing	120 ms
/ZZ/	- zoo, phase	210 ms
/ZH/	- beige, pleasure	190 ms
Voiceless Fricatives		
*/FF/	- Food	150 ms
*/TH/	- Thin	180 ms
*/SS/	- sit	90 ms
/SH/	- shirt, leash, nation	160 ms
/HH1/	- before front vowels: YR, IY, IH, EY EH, XR, AE - he, hen, hit, hear, heat, hay, hair	130 ms
/HH2/	- before back vowels: UW, UH, OW, OY, AO, OR, AR - hue, hook, hoe, hoist, hawk	180 ms
/WH/	- white, whim, twenty	200 ms

TABLE 3-1 (Continued)

ALLOPHONE	SAMPLE WORDS	DURATION
†Voiced Stops		
/BB1/	- final position: rib	
	between vowels: fibber	50 ms
	bleed, brown	
/BB2/	- initial position before	
	a vowel; beast	50 ms
/DD1/	- final position: played, end	70 ms
/DD2/	- initial position: down;	
	clusters: drain	160 ms
/GG1/	- before high front vowels:	
	YR, IY, IH, EY, EH, XR: guest	80 ms
/GG2/	- before high back vowels:	
	UW, UH, OW, OY, AX: and	
	clusters: green, glue	30 ms
/GG3/	- before low vowels: AE, AW,	
	AY, AR, AA, AO, OR, ER; and	
	medial clusters: anger; and	
	final position: peg	160 ms
† Voiceless Stops		
/PP/	- pleasure, ample, trip	210 ms
/TT1/	- final clusters before SS:	
	tests, its	100 ms

TABLE 3-1 (Continued)

SAMPLE WORDS	DURATION
- all other positions:	
test, street	140 ms
- before front vowels:	
YR, IY, IH, EY, EH,	
XR, AY, AE, ER, AX	
initial clusters: cute	
clown, scream	160 ms
- final position: speak;	
final clusters: task	190 ms
- before back vowels: UW, UH,	
OW, OY, OR, AR, AO;	
initial clusters; crane,	
quick, clown, scream	120 ms
- church, feature	190 ms
- judge, injure	140 ms
- milk, alarm, ample	180 ms
- before front and central vowels:	
YR, IY, IH, EY, EH, XR	
AE, ER, AX, AW, AY, UW;	
final clusters: earn	140 ms

TABLE 3-1 (Continued)

ALLOPHONE	SAMPLE WORDS	DURATION
/NN2/	- before back vowels;	
	UH, OW, OY, OR, AR, AA: no	190 ms
/NG/	- string, anger, anchor	220 ms

*These allophones may be doubled for initial position and used singly in final position.

+ Require a pause before allophone.

NOTE: Underlined letters indicate allophone sound.

LET'S GET TECHNICAL. . .

In linguistics, there are many levels of analysis in the study of how words are created and how they interact with each other when joined in sentences. We have just reviewed briefly allophone synthesis, which occupies the lowest level. Let us take a look at other levels of analysis.

There are basically four levels of analysis when dealing with the structure, construction and meaning of a word. They are semantics (meaning), syntax

(arrangement of words), morphology (form and structure of words), and phonology (sounds).

Semantics is the study of the development and changes in meanings of words. These changes occur depending on how the word is used in a phrase or sentence. For example, the word dress has two meanings. As a noun "the dress is blue" and as a verb "I will dress the baby".

Syntax is the relationship of arrangements of words used in phrases or sentences, of all degrees of length and complexity. For example, if one says "the man bit the dog" it conveys a significantly different meaning than "the dog bit the man" even though the individual words used have exactly the same meaning in both cases.

Noun - subject	SS-SS-AX-AX-PA2-BB1-PA2-JH-EH-PA3-KK2-PA3-TT2
Verb - subject	SS-SS-AX-PA2-BB1-PA2-JH-EH-EH-PA3-KK2-PA3-TT2

Morphology concerns itself with the construction of different types or classes of words. How nouns are made plural or how verbs are made past are two examples.

Phonology is the study of speech sounds. It includes areas concerning the distribution of sounds and how sounds in combination affect each other.

Although each level of analysis has its own distinct meaning, in a given situation they all interact with each other. The types of analysis just discussed make a text-to-speech system possible. (This system is discussed in detail in Chapter FIVE. A text string (composed of normal English words) is entered and the computer systems speaks it. For this type of system two sets of rules are required. The first set converts the text entered into the computer to the appropriate allophone sounds. An ideal system of this type would require a complex program incorporating rules for all four levels of analysis. The second set of rules converts the allophone symbols to sounds used to pronounce the desired word or words. In this chapter we have just discussed the second set of rules using the Phoneme/Allophone approach.

The major advantage of the Phoneme/Allophone approach is that it can create an unlimited vocabulary for your computer from a limited inventory of sounds. However, as stated previously the speech quality is reduced due to the overlapping of certain sounds (coarticulation effect) when compared to natural speech.

One method of improving coarticulation is with the use of diphones. Diphones are sounds that encompass

the transition from one sound to the next. They extend from the center of a phoneme to the center of the next phoneme.

Another method to increase speech quality is with the use of morphs. Morphs are the smallest units of sound that can convey meaning. They consist of root words, prefixes and suffixes.

For Example:

Word	Prefix	Root	Suffix
Unrelated	Un	relate	ed
Previewed	Pre	view	ed

Still another method is the demisyllable approach. Demisyllables consist of intial and final half syllables and phonetic affixes.

For Example:

Word	Demisyllable
box	bo ox

Although diphones, morphs and demisyllables increase the quality of allophone synthesis, sets of these tend to contain more units so, the memory required to do so becomes prohibitive as with LPC and PCM techniques. Even without the use of these extensions, allophone speech synthesis still offers a good compromise among many factors such as versatility, flexability, cost, hardware complexity, size of vocabulary, memory storage, and quality of speech.

GENERAL PHONEME CLASSIFICATIONS

The following phoneme classifications describe how phoneme sounds are grouped or classified. Their classification depends on how the sounds were produced, which articulators were used to produce the sounds, where the sound was produced in the vocal cavity, and, at the time of producing the sound, were the vocal cords vibrating or not.

Vowels:

Vowels are produced with a relatively unconstricted vocal tract. The energy source is the vibration of the vocal cords, which is periodic in nature. Vowels are classified according to whether the front or back of the tongue is high or low, whether they are long or short, and whether the lips are rounded or not.

Consonants:

Consonants are produced by creating a constriction in the vocal tract where the source can be in place of or in conjunction with the vocal cords. For stops, fricatives and affricates (see Table 3-1) the energy source is aperiodic (pseudo-random). For others it may be periodic or a combination of

both. Consonants are classified by the place and manner of articulation and by the articulatory features of voicing.

The place of articulation refers to the point in the vocal tract where the sound is made or where two articulators make contact.

The manner of articulation refers to how the consonants are made. It describes the way in which the articulators make contact to produce the speech sound.

Voicing refers to whether the vocal cords were vibrating or not at the time the sound was produced.

Voiced Phonemes:

Phonemes that are produced with the energy source being the vocal cords. (Includes all vowels, resonants, voiced fricatives, and voiced stops. See Table 3-1).

Unvoiced Phonemes:

Phonemes that are produced when the vocal cords are not vibrating and the energy source is at the lips or teeth. (Includes voiceless fricatives, stops, and affricates. See Table 3-1).

Fricatives:

A fricative is produced by using a narrow constriction as the energy source and allowing a rush of air to flow through it.

Stops:

These sounds are produced by completely blocking the flow of air through the oral cavity. This causes the air pressure to build up. When this pressure is released a short burst of noise is generated.

Affricates:

Affricates are produced by first blocking the vocal tract entirely and then allowing air to flow through a narrow constriction. Affricates can be referred to as a combination of a stop consonant followed by a fricative.

Nasals:

Produced by completely blocking the oral cavity and allowing the air to pass through the nasal cavity.

Resonants:

Formed by continuous movement of the tongue from an initial target to one for a following vowel.

CHAPTER THREE: THEORY OF SPEECH SYNTHESIS

GLOSSARY OF TERMS USED IN THIS CHAPTER

Pitch

The highness or lowness of a sound due to vibrations of sound waves.

Sound frequency

Any particular sound is measured by the number of times the sound wave oscillates in a given period. A sound frequency is measured in cycles per second which is known as hertz. For example, the sound frequency of human speech ranges from 0-5 thousand cycles per second or 0-5 thousand hertz.

Resonant frequencies

A sound frequency that is intensified or allowed to pass through a filter. A filter is a device used to screen out or pass through certain frequencies.

Articulators

In the human vocal tract the articulators are the lips, tongue and teeth. These articulators are used to shape the human oral cavity.

Vocal tract

The human vocal tract essentially consists of the articulators, oral cavity, nasal cavity and throat.

Trachea

Throat or windpipe.

Articulation

The process of positioning or connecting the articulators and charging the shape of the vocal tract when producing speech.

Voiced sounds

Sounds that are produced in the vocal tract when the vocal cords are vibrating.

Unvoiced sounds

Sounds that are produced in the vocal tract when the vocal cords are not vibrating.

Formants

The resonant frequencies of the vocal tract.

In order to understand how computers could speak it is important to understand how humans speak. Human beings produce speech in much the same way that flutes and horns produce their sounds. In the case of the horn, the player blows into one end of the horn with a kind of "humming" sound. The horn vibrates as a result of the fact that the player's lips are vibrating. The physical characteristics of the horn (i.e., size, shape, length, etc.) determine the way the horn vibrates. This gives every particular horn its own particular tone or sound. It is important to realize that when different types of horns play the same musical note (pitch) they do not sound alike. The reason is that the player is blowing in a way that causes many sound frequencies to enter the horn in addition to the sound frequencies of the desired musical note. The horn absorbs some of these frequencies and allows others to pass through. It is essentially the shape and size of the horn that determines which frequencies are absorbed and which are passed through. The frequencies that are absorbed are not heard. The frequencies that pass through are heard. This gives a particular horn its own characteristic sound. The frequencies that are passed through are known as resonant frequencies.

Human speech sounds are produced in much the same manner. The human vocal cords vibrate, like the lips of the horn player, and a variety of sound frequencies enter the throat. They then begin to pass up and out through the mouth. Just as in the horn, the size and shape of the mouth determine which of these frequencies will not be heard (absorbed) and which ones will be heard (passed through). The miracle of speech is accomplished because the human throat and mouth can be made to change shape at will. Thus, giving the speaker the ability to determine which frequencies will be heard and which one will not at any given point in time. It is the sequence of these frequency selections which we interpret as speech.

see Figure 1-1

The human brain positions the lips, teeth, tongue (articulators) and throat muscles, all of which essentially comprise the human vocal tract, in order to create the desired changes in size and shape. In the English language we associate any given word with a particular series of selected frequencies, and thus, with a particular series of positions of the human vocal tract. Even something as powerful as the human brain takes several years to master this control.

see Figure 1-2

The speech system then can be considered as consisting of a series of tubes and cavities connecting the lungs to the mouth or nose. These tubes and cavities are approximately 17.4 centimeters long. The vocal cords, located at the opposite end of the trachea from the lungs, control the flow of air from the lungs into the vocal tract. Under muscular control, the cavities that make up the vocal tract can significantly change shape at a rate of 10 times per second and the vocal cords can open and close at a rate of approximately 100-300 times per second. The changing in the shape of the vocal tract, and the shape and positioning of the articulators (teeth, tongue, palate and lips) is known as articulation.

When whispering, human beings produce recognizable speech without moving their vocal cords. This should be further evidence to the reader that it is the movement of the vocal tract that is important and not the movement of the vocal cords. In fact, during normal speech human beings produce sounds with and without their vocal cords vibrating. These two types of sounds are classified as "VOICED", when the vocal cords are vibrating, and "UNVOICED", when they are not. Examples of voiced sounds are the vowels (i.e., "a" as in at, "e" as in eat, "i" as in it, "o" as in oat, and

"u" as in flute) and some consonants such as m and n (i.e., "m" as in man and "n" as in nine). To demonstrate the vocal cord vibration, place your hand on your throat while pronouncing these sounds. You can actually feel your vocal cords vibrating.

see Figure 1-3

Examples of unvoiced sounds are the "s" as in stop and the "sh" as in sheet. If you place your hand on your throat once again and pronounce only the sound (i.e., just the "s", not the top), you will notice that you will not feel any vibration. The sound is actually being produced up near the teeth and lips by forcing air through a small opening between the tongue, teeth and lips. This is the same effect that produces a "hiss" when air rushes out of a tire under pressure.

see Figure 1-4

From the above discussion, we see that the three key features used in the human vocal tract to produce speech are:

1. Vibration of the vocal cords for voiced sounds (see Figure 1-3).
2. Forcing air through a small opening for unvoiced sounds (see Figure 1-4).
3. Change in size and shape of the vocal tract for frequency selection (see Figure 1-1).

Some of the methods of artificial speech production described in the following chapter reproduce only the effects of the human vocal tract. Other methods actually imitate the operation of the three major features listed above.

LET'S GET TECHNICAL. . .

We have just learned that human speech consists of two classifications of sound frequencies. Voiced sounds, which are produced by vocal cord vibration and unvoiced sounds, which are produced by forcing air through small constrictions in the vocal cavity, when the vocal cords are not vibrating. These sound frequencies enter the vocal tract and are passed through the vocal tract or absorbed by the vocal tract. The vocal tract is actually filtering out the undesirable sound frequencies. Therefore, the vocal cavity can be considered to be an adaptive filter network.

The varying shapes of the vocal cavities alter the resonances and frequency response of this network. At any given time, the placement of the articulators and the shape of the cavities determine the frequency response in the vocal tract network. The resonances of this filter network are known as formants. These are the peaks in the frequency plot of the speech signal as

seen in Figure 1-5. In mathematical terms these are the poles of the transfer function at any instant of time of the filter network. So when the vocal tract network changes, the frequency response changes, and the poles of the transfer function change (i.e., adaptive response). These poles are analyzed, and correlating values are applied in an electronic model of the vocal tract when speech is synthesized.

see Figure 1-5

In summary, speech is produced by vocal cord vibration (voiced sounds) or constrictions in the vocal tract (unvoiced sounds). These sounds pass through the vocal tract, which constantly changes shape, thereby, changing the frequency response of the signal. The output signal after being filtered in the vocal cavity is known as speech. The speech signal can be regarded as a waveform consisting of many complex frequencies at any instant of time (see Figure 1-6). The vocal tract can be regarded as a complex time variant filter network used to produce these complex speech frequencies.

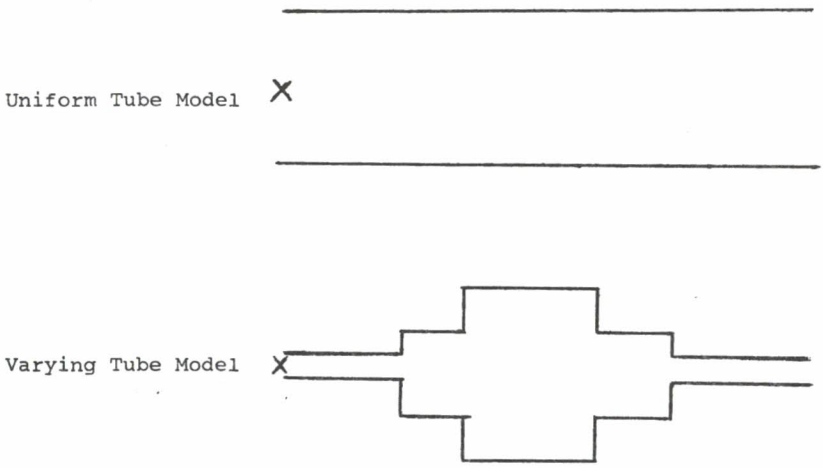


Figure 1-1

1. Lips
2. Teeth
3. Hard Palate
4. Soft Palate (Velum)
5. Tongue
6. Pharynx
7. Epiglottis
8. Position of Vocal Chords
9. Glottis

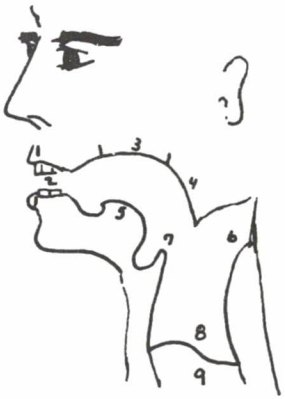


Figure 1-2

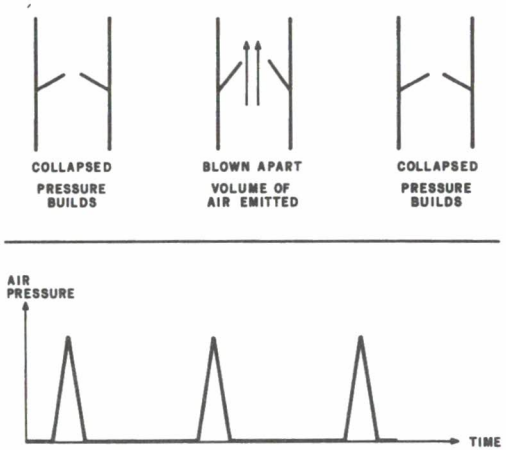

COLLAPSED
PRESSURE BUILDS
BLOWN APART
VOLUME OF AIR EMITTED
COLLAPSED
PRESSURE BUILDS
AIR PRESSURE
TIME

Figure 1-3

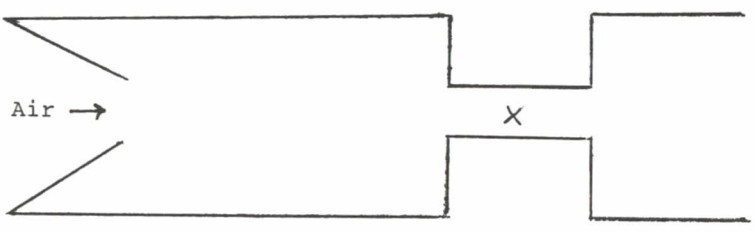


Figure 1-4

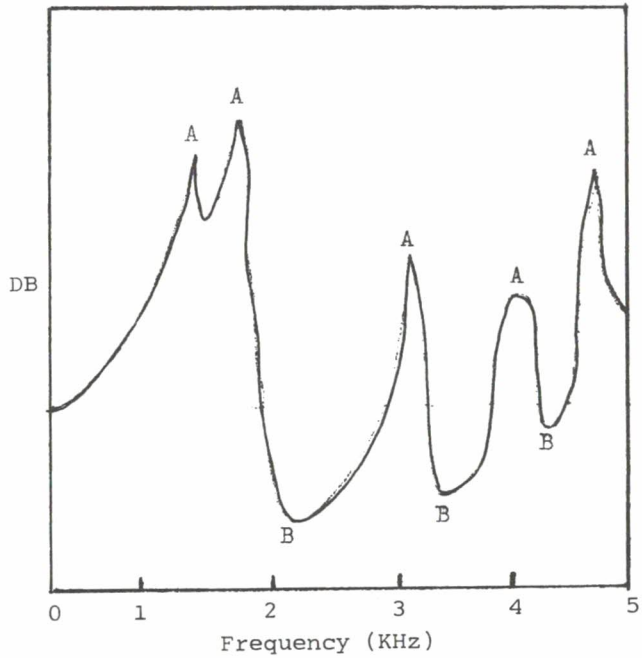


Figure 1-5

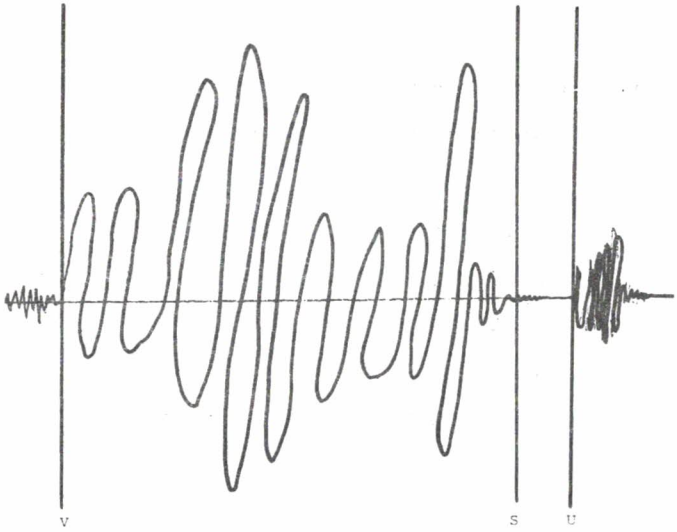


Figure 1-6

CHAPTER FOUR: SPEECH REPRODUCTION TECHNIQUES

GLOSSARY OF TERMS USED IN THIS CHAPTER

Intonation

The manner of producing tones with regard to accurate pitch.

Dynamics

In human speech the process of varying pitch in any utterance.

Analog waveform

A waveform that is continuous.

Digital waveform

A waveform that is used to represent an analog waveform using digital values (numerical codes in base two).

Analog filter

A device that is used to screen out or pass through analog waveforms.

Sampling frequency

The rate at which an analog waveform is "looked at" when converting to a digital waveform.

Analog-to-digital conversion

The process of converting an analog waveform to a digital waveform.

Amplitude

In waveforms, the range from the zero axis (mean) to the maximum value (extreme). In sound waves the amplitude can be regarded as the volume of the signal.

Discrete logic

Transistor Transistor Logic (TTL) which comprise electronic gates (e.g, NOR, AND etc.).

Energy source

A device used to produce electronic pulses.

Periodic source

An energy source that produces electronic pulses at regular intervals or periods.

Pseudo-random noise source

An energy source that produces electronic pulses at random intervals.

Three methods of human speech reproduction and synthesis are discussed in this chapter. The first is pulse code modulation, or PCM, used to digitally record, and playback speech waveforms. The second is continuously variable shape delta modulation (CVSD) which is a "cousin" of PCM. The third is linear predictive coding, or LPC, which uses a speech synthesis model to imitate the operation of the human vocal tract.

All three of the methods begin with the same basic steps. First let's review analog-to-digital conversion.

Analog-to-digital conversion operates as follows. Each point of the original waveform is represented by a number (digital code) which corresponds to the amplitude of the signal. The so called "sampling rate" determines how often the waveform is measured. For example, the temperature outside your house is constantly changing. If we decided to do an analog-to-digital conversion of the temperature, we might agree to measure the temperature at 12:00 noon every day and round it off to the nearest degree. If we continued this for 30 days we would have a digital representation of the termperature outside your house. The sampling rate would be equal to one sample per day. The fact that we rounded the measurement off to the nearest degree means that although the temperature might have changed very slowly at a continuous rate, each of our measurements would indicate that it changed in steps of one degree. If we wish to get more information about temperature movements, we may sample once per hour instead of once per day (see Figure 2-1). So analog-to-digital conversion is characterized by representing

an analog waveform by a series of numerical values spaced at equal intervals of time.

When synthesizing human speech, the first step is to make a high quality recording of the speech to be synthesized, spoken by a professional speaker. Professional speakers (orators) produce the best overall results because their voices have pleasing intonation and dynamics. The recorded signal is passed through an analog filter to remove frequencies above one half the sampling frequency. The signal is then passed through an analog-to-digital converter at a desired sample rate. The sampling rate should be at least 2 times the highest frequency found in the analog signal for the frequency information to be accurately preserved. Reproducing frequencies in a speech signal up to 5 kHz will give a good reproduction of human voice quality. In order to accomplish this reproduction, we must filter the voice waveform to eliminate frequencies above 5 kHz and then digitally sample it at 10 kHz. Sampling this signal means looking at it 10 thousand times in one second and recording the data. It is this data that is used for each of the three methods described.

PCM offers the highest quality speech reproduction, but unfortunately it also requires larger amounts of data storage than home computers possess, even for small vocabularies. Simple hardware such as a digital-to-analog convertor and a small amount of discrete logic will adequately perform PCM. Fairly high quality speech can be reproduced at a memory requirement of 70 thousand bits of memory storage per one second of speech. An average word will require about 35 thousand bits of storage in this form, assuming a 0.5 second duration in an average spoken word. Since we are talking about adding speech to computers that contain approximately 64 thousand bits of memory, you could only store approximately two words in the entire computer's memory.

In PCM, each different amplitude of the speech signal is represented by a different digital code. No assumptions are made about the nature of the signal or its relationship with the speech mechanism.

see Figure 2-2

A recorded or live speech waveform is passed through an analog-to-digital converter as previously described and is stored in memory. The quality of the speech, after passing it back through a digital-to-analog converter, depends on several factors. The

"sampling frequency," or how many times each second the waveform is "looked at" and its value digitized, is the most crucial factor. The higher the sampling rate, the closer it will be to the original recording. The minimum value of the sampling frequency must be twice as high as the highest frequency in the original waveform being sampled (i.e., if the highest frequency in the original waveform is 5 kHz the minimum sampling frequency should be 10 kHz).

Continuously variable slope delta modulation (CVSD) is another method which has often been used to reduce the amounts of memory required for speech storage. Unlike PCM, which assigns a digital code to each amplitude, CVSD assigns a digital code to represent only the change in amplitudes of adjacent samples. CVSD will also vary the amount of change represented by a given code when it has been preceded by certain other sequences of codes. CVSD speech synthesis requires less memory storage than the PCM technique discussed so far; generally 16 thousand bits per second, or 8 thousand bits per word. Using our same example as above a 64K computer would be able to store eight words.

see Figure 2-3

Using LPC, speech waveforms can be coded and stored using only one or two thousand bits per second. Although the hardware used in LPC synthesis is fairly complex, the savings in memory storage more than offsets any additional cost in synthesis hardware. At this rate, we can store approximately 64 words on our 64K computer.

LPC capitalizes more specifically on how the speech waveform is produced rather than merely reproducing the waveform, as previously discussed with the other two techniques. Its name, Linear Predictive Coding (LPC), was derived from the technique it uses to synthesize speech. It predicts the next speech sample by using a linear combination of the preceding speech samples. It actually imitates the features reviewed in Chapter One.

In linear predictions, an energy source is fed into a model of the vocal tract. In general two energy sources are used, a pseudo-random noise source and a periodic source (see Figure 2-4). The source which will be used to send sound frequencies into the filter is determined by whether the sound is voiced or unvoiced. For a voiced sound the voicing selector will feed in a periodic source (similar to the periodic vibration of the vocal cords). For an unvoiced sound (noiselike speech sounds) the selector will supply the

filter with a pseudo-random noise source. This signal will then be multiplied by the amplitude factor (gain) to increase or decrease the volume of the signal. The signal then passes through the model of the human vocal tract and is filtered by it. This corresponds to the human vocal cavity filtering the signal generated by the vocal cords. Finally the signal is passed through a digital-to-analog converter and fed into an audio amplifier and then a speaker. This signal is what is referred to as synthetic speech.

see Figure 2-4

To understand the filter network we must first understand how the human vocal tract mechanism filters speech signals. The varying shape of the vocal tract can correspond to a series of pipes having different diameters. When waveforms pass from one pipe to the next, they generate waveforms in the opposite direction, known as reflected waveforms. These reflected waveforms have associated with them parameters known as reflection coefficients. The reflection coefficients, which correspond to the formants in the speech signal, are analyzed and can be used in different representations or models of the vocal tract. Two examples are a lattice filter model and a cascade filter model. Both methods simulate the

characteristics of the human vocal tract electronically. Now we have one set of numbers that represent one position of the human vocal tract at any instant of time. When the vocal tract changes shape, the set of numbers used in the filter model also change. By varying this set of numbers to correspond to the vocal tract's changing shape and feeding in the appropriate energy source to correspond to a voiced or unvoiced portion of speech, synthetic speech can be created.

see Figure 2-5

A typical analysis sequence of the LPC process contains four main operations.

1) After the speech signal has been passed through an analog-to-digital converter, it is broken into intervals, 15 ms in duration, known as analysis frames. The short time energy of each frame is calculated and used to determine the amplitude or volume of the signal.

2) A voicing decision is made to determine whether the analysis frame is voiced or unvoiced. If the frame is classified as being voiced, the pitch period is determined.

3) An LPC analysis of each frame is performed. This produces a pre-selected (usually 10 or 12) number

of reflection coefficients which best match the spectral characteristics of the sample analysis frame.

4) Coding the data for the appropriate vocal tract model used is next. The generalized reflection coefficients are used to generate the actual filter coefficients that are applied to the vocal tract model. Adding the parameters of amplitude and pitch forms a complete set of data known as a synthesis frame. This data is supplied to the filter and updated every 15-20 ms. If the frame of data includes a pitch parameter, the filter is excited with the periodic source. If the data does not include a pitch parameter, the pseudo-random noise source is used.

The final speech data has now been reduced to approximately two thousand bits per second, a level that allows an inexpensive means of storing synthetic speech data.

LET'S GET TECHNICAL. . .

LPC synthesizers incorporate all pole digital filters where the number of poles represents the number of formants in the speech signal. In this way at any instant of time the digital filter can have approximately the same frequency response as the human vocal tract. By substituting different poles or values for the digital filter representation, in relation to

time, the model can now represent the varying shape of the human vocal tract (see Figure 2-6).

A lattice filter model uses the reflection coefficients, also known as the K parameters, of the speech signal. It can be described by the equation:

$$H_z = \frac{G}{1 - \sum_{k=1}^{10} a_k z^{-k}}$$

For a 10 stage filter, the a_K terms are the 10 reflection coefficients, G is the gain, and the z^{-k} terms represent time delays. The lattice structure includes multiplication, summation, and delay blocks. This technique requires about 400,000 multiplications and additions each second (see Figure 2-7).

A cascade filter model can consist of a series of second order sections. Each section is a digital resonator capable of modelling a single vocal tract resonance or formant. To model six formants, a 6-stage cascade filter is used forming a 12-pole digital filter (see Figure 2-8 bottom). This system requires only one multiplication per pole whereas the lattice filter

requires two multiplications per pole. This model uses frequency and bandwidth parameters which are derived from the reflection coefficients. Each pole of this system can be described by the equation:

$$H_N(Z) = \frac{G}{1 - 2FTZ^{-1} - B_1 Z^{-2}}$$

G represents the gain, F represents the first formant center frequency and B represents the first formant bandwidth (see Figure 2-8 top and Figure 2-9).

The cascade filter is very advantageous in that it can also be used for formant coding techniques. Formant coding is similar to LPC in that it models speech signals in the frequency domain. It differs from LPC in that it uses only the center frequency values of the formants. The bandwidth values are either set to some constant value or are applied algorithmically. It further makes use of the fact that the intelligibility of human speech signals is found in the first three formants. For this reason, only the first three formant frequencies are used. Since this representation of the speech signal is less complete, less memory storage is required to store the speech data. Typically 600-800 bit per second rates can be

obtained. However, since the last three formants are not used, and it is this information that gives the speech emotion, quality, intonation and emphasis, high quality speech recognizable as being that of a particular speaker cannot be obtained.

see Figure 2-10

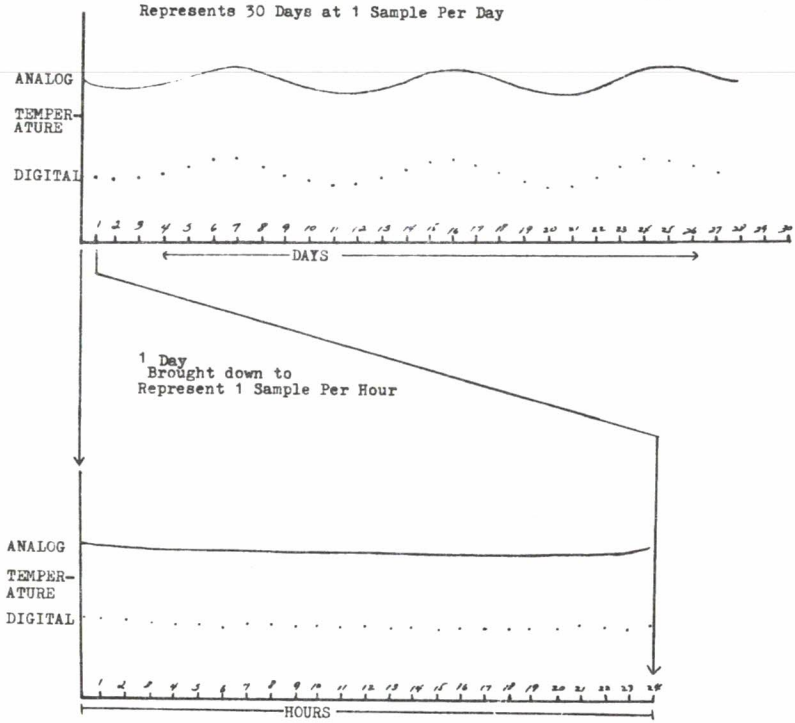
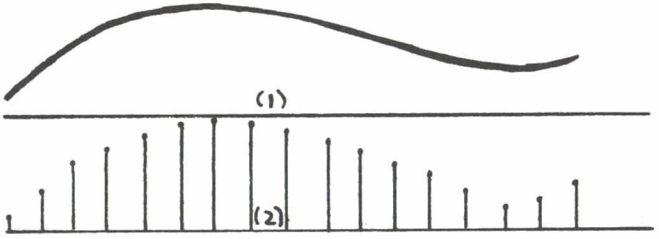
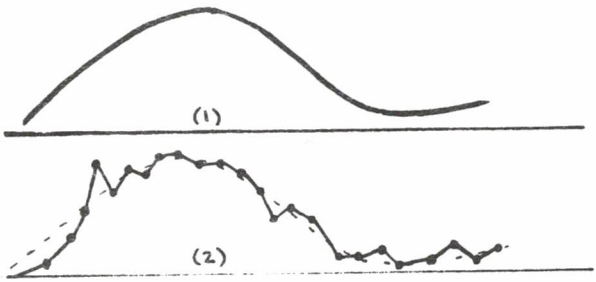

Represents 30 Days at 1 Sample Per Day
ANALOG
TEMPER-
ATURE
DIGITAL
DAYS
1 Day
Brought down to
Represent 1 Sample Per Hour
ANALOG
TEMPER-
ATURE
DIGITAL
HOURS

Figure 2-1



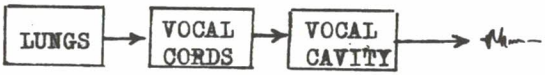
1) Original Waveform - 2) PCM Waveform

Figure 2-2

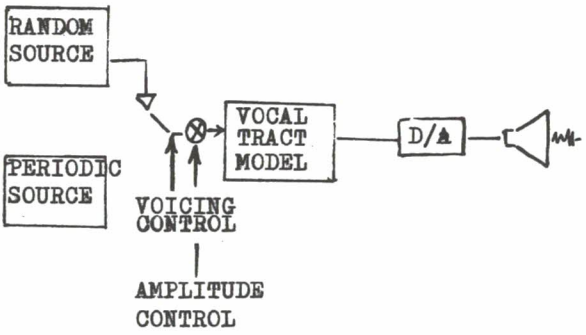


1) Original Waveform - 2) CVSD Waveform

Figure 2-3

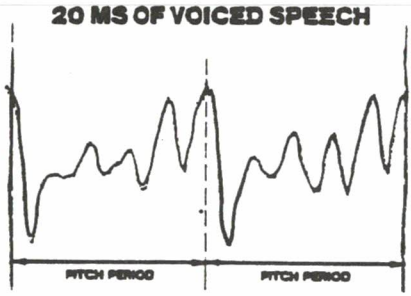


Human Voice Path



Electronic Voice Path (LPC Synthesis Model)

Figure 2-4


20 MS OF VOICED SPEECH
PITCH PERIOD
PITCH PERIOD

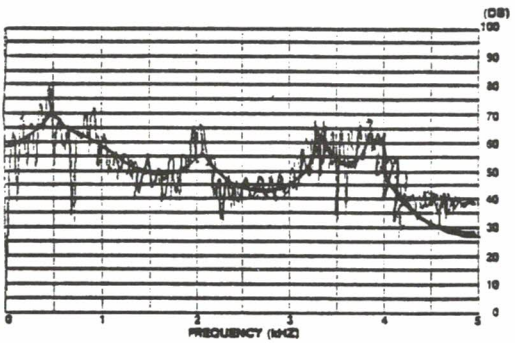

(DB)
100
90
80
70
60
50
40
30
20
10
0
0
1
2
3
4
5
FREQUENCY (KHZ)

Figure 2-5

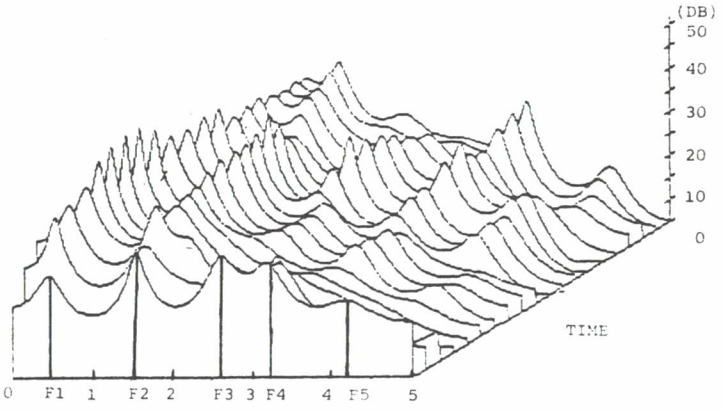


Figure 2-6

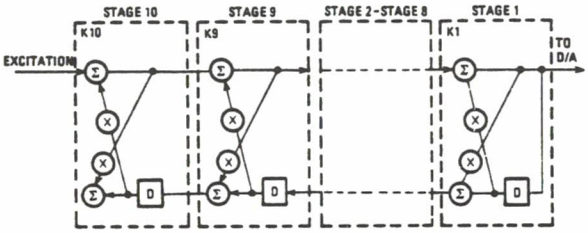
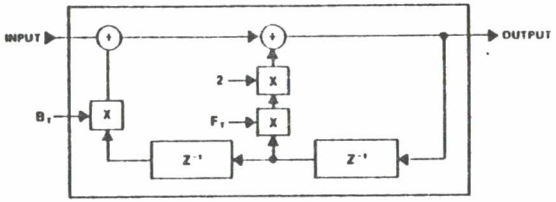
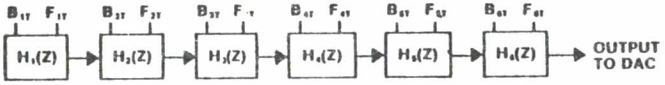

STAGE 10
K10
STAGE 9
K9
STAGE 2-STAGE 8
STAGE 1
K1
EXCITATION
TO
D/A
Σ
X
D

Figure 2-7



$$H_N(Z) = \frac{1}{1 - 2FTZ^{-1} - B_1 Z^{-2}}$$

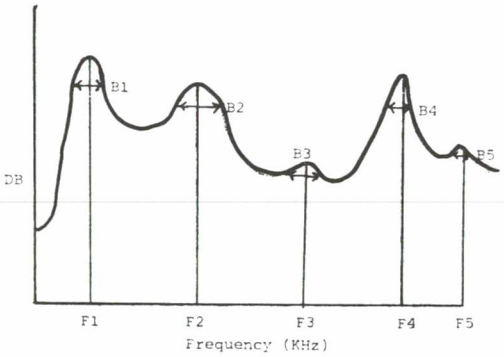
Single Cascade Stage



$$V(Z) = [H_1(Z)] \cdot [H_2(Z)] \cdot [H_3(Z)] \cdot [H_4(Z)] \cdot [H_5(Z)] \cdot [H_6(Z)]$$

$$OUTPUT = A \cdot V(Z)$$

Six Cascade Stages



PARAMETER	DESCRIPTION
A	AMPLITUDE
P	PITCH
R	REPEAT
B_1	FILTER COEFFICIENTS
F_1	
B_2	
F_2	
.	
.	
.	
.	
B_6	
F_6	

SPEECH PARAMETERS

Figure 2-9

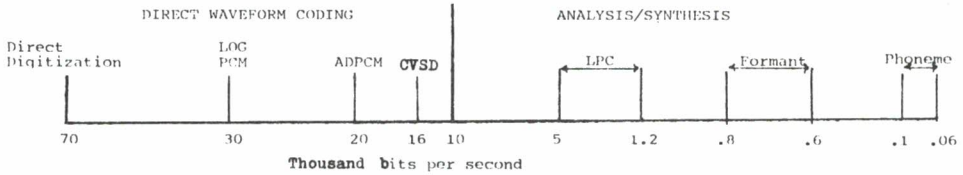

DIRECT WAVEFORM CODING
ANALYSIS/SYNTHESIS
Direct Digitization
LOG PCM
ADPCM
CVSD
LPC
Formant
Phoneme
70
30
20
16
10
5
1.2
.8
.6
.1
.06
Thousand bits per second

Figure 2-10

CHAPTER FIVE: INTRODUCTION TO TEXT TO SPEECH

This manual explored the area of adapting speech synthesis to your home computer through a technique called allophone synthesis. We discussed how words are made up of small "subunits," and how by joining these "subunits" together, words and phrases can be created. In the programs listed, you must make the decision as to which allophones are to be used in creating a desired phrase.

Using this technique, your programs can be made to speak any word which you have assembled the correct allophones. For example, if you want your program to speak the word "hello" you will choose the allophones from the list provided in the book (see Chapter Three) which when strung together form the word "HELLO". You will do this for every word that you wanted your program to speak. It would be nice if you could simply type the desired word the way it is spelled, into the computer, and have it assemble the allophone automatically. Such a process is commonly referred to as text-to-speech conversion. Returning to the example above, in order to get your computer to speak the word "HELLO", one might simply type "H-E-L-L-O" and press return or enter. The computer would then sound out the word using its own internal program. It may also

present you with a list of allophones necessary to say that word.

With this type of system, the computer can actually read aloud messages typed into the computer keyboard or read messages displayed on the computer screen. This system offers many advantages. It is much easier and more convenient to use. It is especially useful when developing vocabularies for your own programs. It simplifies the task of creating words and phrases making the speech synthesizer more enjoyable and fun to use.

The text-to-speech conversion itself is accomplished by what is known as a letter-to-sound rule set. The letter-to-sound rule set is a simple list of instructions which tell what allophone to use for a given letter depending on the surrounding letters. Let's use the word "pale" as an example. Remember, we have just typed "P-A-L-E" into the computer and pressed RETURN. It now has to determine how that word sounds or is spoken. The program would take the first letter "P" and look into its set of statements describing how to use allophones for the letter "P". In this particular case there is only one allophone that is ever used for "P", so when it got to that statement, it would use the allophone, "PP". The next letter "A", is

a little more complicated. The "A" may be a long "A" as in "pale" or a short "A" as in "pat". There is a rule for the letter "A" which says, when "A" is followed by a consonant which is followed by the vowel "E" at the end of the word, the allophone which represents the long "A" sound "EY" is used for "A". This is the simple rule that we have all learned in grade school for sounding out words. This process continues until all the letters of the word have been converted to sounds.

Let's take another example and see how the letter-to-sound program pronounces the word "HELLO". First it looks at the first letter which is an "H". It then goes to the "H" rules. If the "H" is followed by an "E", and then a consonant, and the word does not end with an "E", the HH1 allophone is used. It then goes to the "E" rules. All the rules for "E" are special cases when using ER or EW. Since none of these rules apply and the word does not end with an "E", the "EH" allophone is used. Next it goes to the "L" rules. Since there is only one allophone to pronounce an L, the "LL" allophone is used. Next it goes to the "L" rules again. There is a special rule that says if an "L" has another "L" before it, the second L is silent. Finally it goes to the "O" rules. An "O" followed by a

space is pronounced with the OW allophone. So the final allophone codes that the letter-to-sound rules chose is HH1-EH-LL-OW, which in this case is correct.

A system of this type might contained about 500 such rules. Since the English language has many exceptions to its own pronounciation rules, as any of us know if we try to sound out a word we have never seen before, this type of text-to-speech system alone will not get the words correct 100% of the time. The rule set itself can be expanded to begin to include many special cases, but even then there is always the exception.

The system can be augmented further by the addition of a lexicon (dictionary). These are really specialized rules which apply only to given groups of letters, usually a specific word. For example, we might use a rule which consists of "when you find the letter B-U-I-L-D you will say the allophones BB2-IH-LL-PA2-DD1, which then result in correct pronounciation of the word "BUILD". This is an augmentation to text-to-speech since it helps cover cases which true text-to-speech (letter-to-sound rule set) simply will not handle. It is interesting to note that even if we formed all the words in the English language as specific rules, we will still not get correct

pronounciation 100% of the time. An example of a failure would be the word "READ". In order to determine whether that is pronounced "red" or "reed", the system must understand the letters involved as well as the words that surround the word "READ". This becomes an artificial intelligence problem and involves an entirely new level of complexity.

Although with the use of an extensive dictionary, the text-to-speech becomes very accurate, the memory and expense to store it would be phenomenal. However, minimum dictionary storage of certain words that are hard to pronounce can increase the accuracy of the text-to-speech system for minimal additional cost. For this reason, incorporating dictionary storage in text-to-speech systems becomes a very useful tool.

The addition of emotional content and intonation to machine generated speech also involves artificial intelligence or true understanding of what is being said on the part of the machine. If a text-to-speech system randomly selects stressed syllables, the speech tends to sound very unnatural and unintelligible. Both stress and pitch must be precisely applied to be effective in a text-to-speech system. For this reason, text-to-speech conversion systems typically use a "robot-like" voice with little or no emotion.

Because the English language is so complex, more and more speech rules are required to increase the accuracy of the text-to-speech system. This in turn increases the cost of the system. Therefore using a combination of the above rules, a reasonable solution to the problem can be achieved. Of course, the optimum system attainable would sound like a human being and be 100% accurate. However, there is only one system that I know of that comes close to this level of accuracy and quality. This text to speech system operates on a main frame computer and costs approximately $100,000. This figure is an unrealistic figure for any small business to pay, never mind the personal computer user. Reducing the cost is possible, and some text-to-speech systems for personal computers cost as low as $29.95. However, with speech synthesis, there is always a trade-off. To reduce the cost one must reduce the amount of memory required. Reducing the memory storage reduces the complexity and size of the software algorithm. Reducing the features of the software algorithm leads to an affordable text-to-speech system with "acceptable" speech quality and accuracy.

With all these factors in mind, the choice of what rule sets to use and if pitch should be applied or not is left up to the manufacturer of the text-to-speech

system. It is his or her job to choose the rule set that occupies the least amount of memory, is inexpensive, and offers the maximum accuracy and flexibility. In this way, the consumer can purchase a quality text-to-speech system at an affordable price.

R.I.S.T., Inc. has satisfied all these requirements. Our text to speech system described in Chapter Seven is unique in the marketplace, in that it offers the capability of interfacing to most software cartridges and it gives them the power of speech. You can also create your own dictionary file for each particular program making the system 100% accurate. See Chapter Seven for the other features our system offers. After reading Chapter Seven, if you haven't already purchased the text to speech program; you will!

CHAPTER ELEVEN: [1]ALLOPHONE DICTIONARY OF COMMONLY USED WORDS

The following list of words was added to this book to help you when building phrases for your programs. The allophone sounds corresponding to each word are the symbols that should be entered into the Exclusive Phrase Finder Program.

If you would like to enter the speecn in your own programs after testing it in the Exclusive Program, you may do so by converting the allophone sounds to decimal codes. After converting to the decimal codes, the codes should be stored in the Sample Program.

Keep in mind that the following allophone codes are suggestions only. As described in Chapter TWO what is pleasing to you may not be to someone else. So use this dictionary as a guideline only. If you want to use other sounds to pronounce the words, go right ahead!

NUMBERS

ZERO	ZZ YR OW
ONE, WON	WW SX AX NN1
TWO, TO, TOO	TT2 UW2
THREE	TH RR1 IY
FOUR, FOR, FORE	FF FF OR
FIVE	FF FF AY VV
SIX	SS SS IH IH PA3 KK2 SS
SEVEN	SS SS EH EH VV IH NN1
EIGHT, ATE	EY PA3 TT2

NINE	NN1 AA AY NN1
TEN	TT2 EH EH NN1
ELEVEN	IH LL EH EH VV IH NN1
TWELVE	TT2 WH EH EH LL VV
THIRTEEN	TH ER1 PA2 PA3 TT2 IY NN1
FOURTEEN	FF OR PA2 PA3 TT2 IY NN1
FIFTEEN	FF IH FF PA2 PA3 TT2 IY NN1
SIXTEEN	SS SS IH PA3 KK2 SS PA2 PA3 TT2 IY NN1
SEVENTEEN	SS SS EH VV TH NN1 PA2 PA3 TT2 IY NN1
EIGHTEEN	EY PA2 PA3 TT2 IY NN1
NINETEEN	NN1 AY NN1 PA2 PA3 TT2 IY NN1
TWENTY	TT2 WH EH EH NN1 PA2 PA3 TT2 IY
THIRTY	TH ER2 PA2 PA3 TT2 IY
FORTY	FF OR PA3 TT2 IY
FIFTY	FF FF IH FF FF PA2 PA3 TT2 IY
SIXTY	SS SS IH PA3 KK2 SS PA2 PA3 TT2 IY
SEVENTY	SS SS EH VV IH NN1 PA2 PA3 TT2 IY
EIGHT	EY PA3 TT2 IY
NINETY	NN1 AY NN1 PA3 TT2 IY
HUNDRED	HH2 AX AX NN1 PA2 DD2 RR2 IH IH PA1 DD1
THOUSAND	TH AA AW ZZ TH PA1 PA1 NN1 DD1
MILLION	MM IH IH LL YY1 AX NN1

DAY OF THE WEEK

SUNDAY	SS SS AX AX NN1 PA2 DD2 EY
MONDAY	MM AX AX NN1 PA2 DD2 EY
TUESDAY	TT2 UW2 ZZ PA2 DD2 EY
WEDNESDAY	WW EH EH NN1 ZZ PA2 DD2 EY
THURSDAY	TH ER2 ZZ PA2 DD2 EY
FRIDAY	FF RR2 AY PA2 DD2 EY
SATURDAY	SS SS AE PA3 TT2 PA2 DD2 EY

MONTHS

JANUARY	JH AE AE NN1 YY2 XR 1Y
FEBRUARY	FF EH EH PA1 BR RR2 UW2 XR IY
MARCH	MM AR PA3 CH
APRIL	EY PA3 PP RR2 IH IH LL
MAY	MM EY
JUNE	JH UW2 NN1
JULY	JH UW1 LL AY
AUGUST	AO AO PA2 GG2 AX SS PA3 TT1
SEPTEMBER	SS SS EH PA3 PP PA3 TT2 EH EH PA1 BB2 ER1

OCTOBER	AA PA2 KK2 PA3 TT2 OW PA1 BB2 ER1
NOVEMBER	NN2 OW VV EH EH MM PA1 BB2 ER1
DECEMBER	DD2 IY SS SS EH EH MM PA1 BB2 ER1

LETTERS

A	EY
B	BB2 IY
C	SS SS IY
D	DD2 IY
E	IY
F	EH EH FF FF
G	JH IY
H	EY PA2 PA3 CH
I	AA AY
J	JH EH EY
K	KK1 EH EY
L	EH EH EL
M	EH EH MM
N	EH EH NN1
O	OW
P	PP IY
Q	KK1 YY1 UW2
R	AR
S	EH EH SS SS
T	TT2 IY
U	YY1 UW2
V	VV IY
W	DD2 AX PA2 BB2 EL YY1 UW2
X	EH EH PA3 KK2 SS SS
Y	WW AY
Z	ZZ IY

DICTIONARY

ALARM	AX LL AR MM
BATHE	BB2 EY DH2
BATHER	BB2 EY DH2 ER1
BATHING	BB2 EY DH2 IH NG
BEER	BB2 YR
BREAD	BB1 RR2 EH EH PA1 DD1
BY	BB2 AA AY
CALENDAR	KK1 AE AE LL EH NN1 PA2 DD2 ER1
CLOCK	KK1 LL AA AA PA3 KK2
CLOWN	KK1 LL AW NN1
CHECK	CH EH EH PA3 KK2
CHECKED	CH EH EH PA3 KK2 PA2 TT2
CHECKER	CH EH EH PA3 KK1 ER1

CHECKERS	CH EH EH PA3 KK1 ER1 ZZ
CHECKING	CH EH EH PA3 KK1 IH NG
CHECKS	CH EH EH PA3 KK1 SS
COGNITIVE	KK3 AA AA GG3 NN1 IH PA3 TT2 IH VV
COLLIDE	KK3 AX LL AY DD1
COMPUTER	KK1 AX MM PP1 YY1 UW1 TT2 ER
COOKIE	KK3 UH KK1 IY
COOP	KK3 UW2 PA3 PP
CORRECT	KK1 ER2 EH EH PA2 KK2 PA2 TT1
CORRECTED	KK1 ER2 EH EH PA2 KK2 PA2 TT2 PA2 DD1
CORRECTING	KK1 ER2 EH EH PA2 KK2 PA2 TT2 NG
CORRECTS	KK1 ER2 EH EH PA2 KK2 PA2 TT1 SS
CROWN	KK1 RR2 AW NN1
DATE	DD2 EY PA3 TT2
DAUGHTER	DD2 AO TT2 ER1
DAY	DD2 EH EY
DIVIDED	DD2 IH VV AY PA2 DD2 IH PA2 DD1
EMOTIONAL	IY MM OW SH AX NN1 AX EL
ENGAGE	EH EH PA1 NN1 GG1 EY PA2 JH
ENGAGEMENT	EH EH PA1 NN1 GG1 EY PA2 JH MM EH EH NN1 PA2 PA3 TT2
ENGAGES	EH EH PA1 NN1 GG1 EY PA2 JH IH ZZ
ENGAGING	EH EH PA1 NN1 GG1 EY PA2 JH IH NG
ENRAGE	EH NN1 RR1 EY PA2 JH
ENRAGED	EH NN1 RR1 EY PA2 JH PA2 DD1
ENRAGES	EH NN1 RR1 EY PA2 JH IH ZZ
ENRAGING	EH NN1 RR1 EY PA2 JH IH NG
ESCAPE	EH SS SS PA3 KK1 PA2 PA3 PP
ESCAPED	EH SS SS PA3 KK1 PA2 PA3 PP PA2 TT2
ESCAPES	EH SS SS PA3 KK1 PA2 PA3 PP SS
ESCAPE	EH SS SS PA3 KK1 PA2 PA3 PP IH NG
EQUAL	IY PA2 PA3 KK3 WH AX EL
EQUALS	IY PA2 PA3 KK3 WH AX EL ZZ
ERROR	EH XR OR
EXTENT	EH KK1SS TT2 EH EH NN1 TT2
FIR	FF ER2
FREEZE	FF FF RR1 IY ZZ
FREEZER	FF FF RR1 IY ZZ ER1
FREEZERS	FF FF RR1 IY ZZ ER1 ZZ
FREEZING	FF FF RR1 IY ZZ IH NG
FROZEN	FF FF RR1 OW ZZ EH NN1
GAUGE	GG1 EY PA2 JH
GAUGED	GG1EY PA2 JH PA2 DD1
GAUGES	GG1 EY PA2 JH IH ZZ
GAUGING	GG1 EY PA2 JH IH NG
HELLO	HH EH LL AX OW
HOUR	AW ER1

INFINITIVE	IH NN1 FF FF IH IH NN1 IH PA2 PA3 TT2 IH VV
INTRIGUE	IH NN1 PA3 TT2 RR2 IY PA1 GG3
INTRIGUED	IH NN1 PA3 TT2 RR2 IY PA1 GG3 PA2 DD1
INTRIGUES	IH NN1 PA3 TT2 RR2 IY PA1 GG3 ZZ
INTRIGUING	IH NN1 PA3 TT2 RR2 IY PA1 GG3 IH NG
INVESTIGATE	IH IH NN1 VV EH EH SS PA2 PA3 TT2 IH GG1 EY PA2 TT2
INVESTIGATED	IH IH NN1 VV EH EH SS PA2 PA3 TT2 IH GG1 EY PA2 TT2 IH PA2 DD1
INVESTIGATOR	IH IH NN1 VV EH EH SS PA2 PA3 TT2 IH GG1 EY PA2 TT2 ER1
INVESTIGATORS	IH IH NN1 VV EH EH SS PA2 PA3 TT2 IH GG1 EY PA2 TT2 ER1 ZZ
INVESTIGATES	IH IH NN1 VV EH EH SS PA2 PA3 TT2 IH GG1 EY PA2 TT1 SS
INVESTIGATING	IH IH NN1 VV EH EH SS PA2 PA3 TT2 IH GG1 EY PA2 TT2 IH NG
KEY	KK1 IY
LEGISLATE	LL EH EH PA2 JH JH SS SS LL EY PA2 PA3 TT2
LEGISLATED	LL EH EH PA2 JH JH SS SS LL EY PA2 PA3 TT2 IH DD1
LEGISLATES	LL EH EH PA2 JH JH SS SS LL EY PA2 PA3 TT1 SS
LEGISLATING	LL EH EH PA2 JH JH SS SS LL EY PA2 PA3 TT2 IH NG
LEGISLATURE	LL EH EH PA2 JH JH SS SS LL EY PA2 PA3 CH ER1
LETTER	LL EH EH PA3 TT2 ER1
LITTER	LL IH IH PA3 TT2 ER1
LITTLE	LL IH IH PA3 TT2 EL
MEMORY	MM EH EH MM ER2 IY
MEMORIES	MM EH EH MM ER2 IY ZZ
MINUTE	MM IH NN1 IH PA3 TT2
MONTH	MM AX NN1 TH
NIP	NN1 IH IH PA2 PA3 PP
NIPPED	NN1 IH IH PA2 PA3 PP PA3 TT2
NIPPING	NN1 IH IH PA2 PA3 PP IH NG
NIPS	NN1 IH IH PA2 PA3 PP SS
NO	NN2 AX OW
PHYSICAL	FF FF IH ZZ IH PA3 KK1 AX EL
PIN	PP IH IH NN1
PINNED	PP IH IH NN1 PA2 DD1
PINNING	PP IH IH NN1 IH NG1
PINS	PP IH IH NN1 ZZ
PLEDGE	PP LL EH EH PA3 JH
PLEDGED	PP LL EH EH PA3 JH PA2 DD1

PLEDGES	PP LL EH EH PA3 JH IH ZZ
PLEDGING	PP LL EH EH PA3 JH IH NG
PLUS	PP LL AX AX SS SS
RAY	RR1 EH EY
RAYS	RR1 EH EY ZZ
READY	RR1 EH EH PA1 DD2 IY
RED	RR1 EH EH PA1 DD1
ROBOT	RR1 OW PA2 BB2 AA PA3 TT2
ROBOTS	RR1 OW PA2 BB2 AA PA3 TT1 SS
SCORE	SS SS PA3 KK3 OR
SECOND	SS SS EH PA3 KK1 IH NN1 PA2 DD1
SENSITIVE	SS SS EH EH NN1 SS SS IH PA2 PA3 TT2 IH VV
SENSITIVITY	SS SS EH EH NN1 SS SS IH PA2 PA3 TT2 IH VV IH PA2 PA3 TT2 IY
SINCERE	SS SS IH IH NN1 SS SS YR
SINCERELY	SS SS IH IH NN1 SS SS YR LL IY
SINCERITY	SS SS IH IH NN1 SS SS EH EH RR1 IH PA2 PA3 TT2 IY
SISTER	SS SS IH IH SS PA3 TT2 ER1
SPEAK	SS SS PA3 IY PA3 KK2
SPELL	SS SS PA3 PP EH EH EL
SPELLED	SS SS PA3 PP EH EH EL PA3 DD1
SPELLER	SS SS PA3 PP EH EH EL ER2
SPELLERS	SS SS PA3 PP EH EH EL ER2 ZZ
SPELLING	SS SS PA3 PP EH EH EL IH NG
SPELLS	SS SS PA3 PP EH EH EL ZZ
START	SS SS PA3 TT2 AR PA3 TT2
STARTED	SS SS PA3 TT2 AR PA3 TT2 IH PA1 DD2
STARTER	SS SS PA3 TT2 AR PA3 TT2 ER1
STARTING	SS SS PA3 TT2 AR PA3 TT2 IH NG
STARTS	SS SS PA3 TT2 AR PA3 TT1 SS
STOP	SS SS PA3 TT1 AA AA PA3 PP
STOPPED	SS SS PA3 TT1 AA AA PA3 PP PA3 TT2
STOPPER	SS SS PA3 TT1 AA AA PA3 PP ER1
STOPPING	SS SS PA3 TT1 AA AA PA3 PP IH NG
STOPS	SS SS PA3 TT1 AA AA PA3 PP SS
SUBJECT (NOUN)	SS SS AX AX PA2 BB1 PA2 JH EH PA3 KK2 PA3 TT2
SUBJECT (VERB)	SS SS AX PA2 BB1 PA2 JH EH EH PA3 KK2 PA3 TT2
SWEAT	SS SS WW EH EH PA3 TT2
SWEATED	SS SS WW EH EH PA3 TT2 IH PA3 DD1
SWEATER	SS SS WW EH EH PA3 TT2 ER1
SWEATERS	SS SS WW EH EH PA3 TT2 ER1 ZZ
SWEAT	SS SS WW EH EH PA3 TT2 IH NG
SWEATS	SS SS WW EH EH PA3 TT2 SS
SWITCH	SS SS WH IH IH PA3 CH

```
SWITCHED            SS SS WH IH IH PA3 CH PA3 TT2
SWITCHES            SS SS WH IH IH PA3 CH IH ZZ2
SWITCHING           SS SS WH IH IH PA3 CH IH NG2
SYSTEM              SS SS IH IH SS SS PA3 TT2 EH MM
SYSTEMS             SS SS IH IH SS SS PA3 TT2 EH MM
                    ZZ
TALK                TT2 AO AO PA2 KK2
TALKED              TT2 AO AO PA2 KK2 PA3 TT2
TALKER              TT2 AO AO PA2 KK1 ER1
TALKERS             TT2 AO AO PA2 KK1 ER1 ZZ
TALKING             TT2 AO AO PA2 KK1 IH NG
TALKS               TT2 AO AO PA2 KK2 SS
THREAD              TH RR1 EH EH PA2 DD1
THREADED            TH RR1 EH EH PA2 DD2 IH PA2 DD1
THREADER            TH RR1 EH EH PA2 DD2 ER1
THREADERS           TH RR1 EH EH PA2 DD2 ER1 ZZ
THREADING           TH RR1 EH EH PA2 DD2 IH NG
THREADS             TH RR1 EH EH PA2 DD2 ZZ
THEN                DH1 EH EH NN1
TIME                TT2 AA AY MM
TIMES               TT2 AA AY MM ZZ
UNCLE               AX NG PA3 KK3 EL
WHALE               WW EY EL
WHALER              WW EY LL ER1
WHALERS             WW EY LL ER1 ZZ
WHALES              WW EY EL ZZ
WHALING             WW EY LL TH NG
YEAR                YY2 YR
YES                 YY2 EH EH SS SS
```

CHAPTER SEVEN: VIC 20/Commodore 64 Text-to-Speech Directions

The text-to-speech cassette cartridge incorporates many features allowing your computer to speak any desired English text sentence entered from any of the 255 device ports.

The applications and uses for this type of system are endless. Text entered from the keyboard, displayed on the screen, incorporated in your own programs, resident in game cartridges, now has the ability to be spoken automatically!

Features

I. Text-to-Speech Vocabulary Development System

This system has two main features. One was designed to allow you to build dictionaries for any program of your choice. It has the ability to EDIT various phrases and words and then SAVE them in a dictionary file. The maximum capability of each dictionary file is a combination of 4000 text characters and/or allophone sounds. So using an average of 5 text characters and 5 allophone sounds per word, each dictionary file can store up to 400 words. An unlimited number of dictionaries can be stored depending on how much disk or tape space you want to utilize. However only one dictionary may be stored in memory at any one time.

For example, you want to play an adventure cartridge and you want to have the text sent to the screen spoken. While playing the game you notice that the text-to-speech cartridge is pronouncing a few words incorrectly. (This tends to happen with any text-to-speech system. Even the more expensive ($1,000.00 systems) are only 95-99% accurate). After making a list of which words you want stored in a dictionary file, you would reload the Text-to-Speech Development System. After entering the text "T" command the desired word may be entered. Depressing the RETURN key converts the text to the actual allophone sounds used. Now you may EDIT that word in any way that you deem necessary. After the word sounds pleasing to your ear it may be added and saved in the dictionary file. When you reload the Text-to-Speech Operating System and play your adventure game again it will always speak the corrected words in the way you have just specified. You can create any number of dictionaries for each individual cartridge of your choice. You have just made this particular text-to-speech system 100% accurate.

The other feature that the Text-to-Speech Development System offers is the ability to convert the allophone sounds to the decimal codes that the speech synthesizer actually utilizes. This is extremely helpful when adding speech to your basic programs as specified in the Sample Program in your VIC 20/C64 Complete Module Instruction Section.

For example, if you wanted to have your basic program speak without utilizing the text-to-speech cartridge you would first develop your vocabulary using the Text-to-Speech Development System. After entering the desired text and converting to allophone sounds, editing may begin. After the desired phrase sounds pleasing to your ear, you may convert those sounds to the decimal codes with either the "LIST" or "N" commands. These decimal codes are the codes to be used with the Sample Program located in the Complete Module Instruction Section. Once the decimal codes and Sample Program are added to your basic program you need not enter the text-to-speech cartridge to have your program speak. Just load in your basic program. In some cases this method may be more efficient and faster. What is advantageous is that the choice is yours!

II. Text-to-Speech Input/Output Device Interface

The VIC 20/Commodore 64 has the ability to be expanded by adding on peripherals or devices. These devices may be input or output and are located at any port location between 0 and 255. The Text-to-Speech system was designed to speak any text sent to it through any port via any input or output device.

For example, the VIC 20/Commodore 64 utilize the following ports:

Device Number	Device	Input/Output
4-255	Serial bus	Input/Output
0	Keyboard	Input
1	Tape Recorder	Output/Input
2	RS232	Output/Input
2	Modem	Output/Input
3	Screen	Output
4	Printer 1	Output
5	Printer 2	Output
8	Disk Drive	Input/Output

If you want your computer to speak anything that is typed into it you would have to specify device number 0 (zero),

input. For the RS232, if you are sending text from the computer to the RS232 port you would have to specify device number 2, output. If you want text entered via a remote terminal and modem you would have to specify 2, input.

III. Scott Adams Adventure Series Interface (VIC 20 only)

Utilizing device number 2, output the text-to-speech cartridge directly interfaces to all the Scott Adams Adventure Game series. While playing the game you have the option for the text printed on the screen to be spoken or not. This feature enhances the game and makes it more enjoyable to play.

IV. Infocom/Commodore Adventure Series Interface (C64 only)

Utilizing device number 3, output the text-to-speech cartridge interfaces directly to all the Infocom/Commodore Adventure Game Series (e.g. ZORK I). While playing the game, the text printed to the screen will be spoken. This feature enhances the game and makes it more enjoyable to play.

V. Other Talking Cartridges

If you want any of your cartridges to speak, you may do so by utilizing device number 3, output. This mode will speak any text written to the screen.

Note: VIC 20: If the cartridge utilizes the same MEMORY space as the text-to-speech cartridge, the game cartridge will not speak. The MEMORY space for the text-to-speech cartridge is located at BLOCK 5.
C64: If the cartridge gives you control over the computer before the game play, the text-to-speech system may be loaded. If not, the cartridge will not speak.

VI. Print "Speak" Mode

If you want any of your basic programs to speak, it can be accomplished by simply printing the text to be spoken with the use of "PRINT" statements. The system must be utilizing device number 3, output.

For example, at the point in your program when you want the synthesizer to speak, just print the desired phrase in a PRINT statement.

```
10   GOTO 200
20   For X = 1 TO L
30   I = 1
40   PRINT "HELLO"
```

At line 40 the synthesizer will say Hello. It's as simple as that to incorporate speech in any program!

Operating Instructions

Note: The VIC 20 requires a 3 or 6 slot expansion interface and an 8K RAM card for the text-to-speech system. The RAM card should be set at BLOCK 5. While in the text-to-speech development mode only, it also requires an additional 3K, 8K or 16K RAM card. Once you have completed developing a vocabulary with the text-to-speech development system the additional 3K, 8K or 16K memory may be removed.

Step 1. After inserting the Text-to-Speech cassette or disk cartridge type: For cassette, LOAD "RETURN" for disk VIC 20, LOAD "INSTALL VIC-20",8. C64: LOAD "INSTALL COM-64",8. The program takes approximately 2 1/2 minutes to load.

Step 2. For cassette: press the play button on the tape recorder. For disk: automatically loads install program.

Step 3. After the program is loaded Type RUN, "RETURN".

Note: Keep the play button depressed.

Step 4. The system will then ask "Do you want to load in the Text-to-Speech Development System?"

If Yes: see Text-to-Speech Development System instructions. The program takes approximately five minutes to load.

Note: The speech module must be in the User Port to operate. For future expansion a modem adapter will allow you to plug in the speech module to the expansion port. If you are utilizing this adapter

the speech synthesizer must be disconnected and plugged into the User Port.

Step 5. For the C64: The system will then ask if you want the text-to-speech program loaded under the BASIC ROM (starting at location 40960) or the KERNAL ROM (starting at location 57344).

Note: For most applications like the InfoCom/Commodore Adventure series (e.g. ZORK I), the text-to-speech program should be loaded under the KERNAL ROM. If your application or game programs uses interupts (timing programs, timers in the C64, modem) then the text-to-speech system should be loaded under the BASIC ROM.

For the VIC 20: proceed to Step 6.

Step 6. The system will then ask "Do you want to load a dictionary file?"

If Yes, do the following:

1) Type YES, RETURN

2) Type in the dictionary file name.

3) For cassette:

   a) Stop tape recorder.

   b) Insert cassette tape with desired dictionary file.

   For Disk:

   a) Insert disk with desired dictionary file.

4) Enter whether the dictionary will be loaded from tape or disk. Once dictionary file is loaded, the program will automatically go to Step 7.

Note: Any number of dictionaries may be saved on cassette tape or disk. However, only one dictionary may be loaded into memory at any one time.

If No, go to Step 7.

Step 7. For the C64: The system will ask "Where should the driver routines go?" In the UPPER RAM Block or the STACK AREA.

Note: The driver routines handle the memory management and will install the text-to-speech program. These routines are about 70 bytes long and are normally put in the UPPER RAM Block starting at location 49152-49222. If your application or game program utilizes this memory space, put the driver routines in the STACK AREA.

For Example: The Infocom/Commodore Adventure Series (e.g. ZORK I) requires this area of memory, so place the driver routines in the STACK AREA.

For the VIC 20: proceed to Step 8.

Step 8. The system will ask "What device number do you want to capture text from"?

Enter device number (0-255 possibilities)

Is it an input or output device?

Enter Input or Output

Step 9. The system will then ask "Where is the speech module located?"

User Port or the Expansion Port

Note: The User Port should always be used. The Expansion Port is to be used with a future expansion interface card which captures text from a modem or terminal.

Step 10. The text-to-speech system is now installed into the operating system. To deactivate and reactivate it, follow instructions on the screen at this point.

Note: Keep a record of the "SYS" command to reactivate the system.

Step 11. Turn off tape recorder or remove disk.

For Scott Adams Adventure Game Cartridges (VIC 20 only)

Step 1. For Step 8, enter device number 2. Enter Output.

Step 2. Insert game cartridge into expansion slot.

Step 3. Type SYS 32592 - loads game cartridge.

The system will now speak any text written on the screen. A "V" RETURN command will turn the speech on and off.

For InfoCom/Commodore Adventure Game Cartridges (C64 only)

Step 1. For Step 8, enter device number 3. Enter Output.

Step 2. Insert Infocom Disk.

Step 3. Type LOAD "GAME",8

Step 4. TYPE RUN.

Step 5. Follow the instructions in the InfoCom/Commodore instruction manual.

For Any Adventure/Software Program Cartridge or Disk

Note: Vic 20: The text-to-speech system will not work if the software cartridge utilizes BLOCK 5 of the MEMORY card. C64: The text-to-speech system will not work if the software cartridge takes control of the computer before game play.

Step 1. For Step 8. enter device number 3. Enter Output.

Step 2. Load your software cartridge as specifed in the manufacturer's instructions.

The system will now speak any text written on the screen.

For Your own Basic Programs to Speak

Step 1. For Step 8, enter device number 3. Enter output.

Step 2. Load your basic program.

The system will now speak any statement stored in your basic program as a "PRINT" statement.

Text-to-Speech Development System Instructions

Note: VIC 20 only: When using the text-to-speech development system an additional 3K, 8K or 16K RAM card is required.

Step 1. Enter Yes when the text-to-speech system asks if you want to load in the Text-to-Speech Development System. The program is automatically loaded and RUN.

Step 2. Turn off tape recorder if your using a cassette recorder.

You may now enter the allophones as explained in this Speech Synthesis Instruction manual (Chapter 1). In addition to the commands explained in the Speech Synthesis Instruction Manual (phoneme strings , REPLACE, INSERT, DELETE, RETURN, XL, XR, L, R) you have available the following:

COMMAND	DESCRIPTION
"T" RETURN	Allows you to enter in a a text string. The maximum number of text characters that may be entered at any one time for the VIC 20 is 88. For the C64: A maximum of 67 characters (2 lines of text only) may be entered at any one time. If you exceed this boundary, clear it and start over. A "RETURN" command must be depressed before new text can be entered. The new text string will be added to the previously stored words or phrases. To clear the text use the "H" and "C" commands.

Note: In the "T" text mode, the following punctuation marks represent different pauses.

Punctuation	Pause
-	PA1
Space	PA2
:;,	PA4
()+*/	PA5
!?.	PA5 PA5

COMMAND	DESCRIPTION
"RETURN" (directly after text has been entered)	Converts the text to the allophone symbols used.
"RETURN" (after text has been converted)	Speaks the allophone string.
"N"	Converts allophone symbols to the decimal codes that the synthesizer utilizes. Another "N" command will convert the codes back to the allophone symbols.
"LIST"	Lists the allophones and decimal codes in two columns for easy access and look up.
"RETURN" (directly after the "LIST" command)	Converts back to the allophone string once all allophones have been displayed. Note: The allophones are displayed 20 at a time on the screen.
"H"	Home. Positions the arrow cursor back to the beginning of the screen.
"C"	Clear. Deletes all the allophones after the arrow cursor. After clearing the screen new phrases and words may be entered.
"ADD"	Adds a word or phrase to the end of the dictionary file. You cannot add a word to the dictionary that has already been stored. You must first delete the old word then add the new one.

COMMAND	DESCRIPTION
"SAVE"	Saves the new words in the dictionary file on tape or disk. To SAVE proceed with the following: 1. Type SAVE, "RETURN". 2. Type the file name of the dictionary file being saved. 3. Insert disk into disk drive or place cassette in tape recorder. 4. Type TAPE or DISK Note: for cassette press Play and Record on tape recorder. 5. After dictionary is saved STOP tape recorder or remove disk.
"DEL"	Deletes one word at a time from the dictionary file.
"DIR"	Lists a directory of words stored in the dictionary file. The words will appear on the screen 20 at a time. To continue the list just press "RETURN". To get back into the program you must go to the end of the list or press the RUN/STOP key. Then "RUN" the program again.
"ERA"	Clears everything in the dictionary file.

COMMAND	DESCRIPTION
"LOAD"	Loads the desired dictionary file from tape into memory. To operate proceed with the following: 1. Insert cassette tape or disk with the desired dictionary file. 2. Type LOAD, "RETURN". 3. Enter the file name of the desired dictionary file to be loaded. 4. Enter TAPE OR DISK. Note: For cassette press play on the tape recorder. 5. After loading STOP the tape.

NOTE: Dictionaries previously stored in memory can be written over by loading in a new dictionary file.

To load the Text-to-Speech Development System while in another mode (i.e. Scott Adams, Keyboard etc.).

Note: The speech module must be located in the User Port. For cassette tape:

If the cassette tape already loaded the Text-to-Speech Development system, the tape must be rewound to the beginning of the dictionary file. For the C64: The text-to-speech system must be under the KERNAL ROM.

Step 1. Press the RUN/STOP and RESTORE keys at the same time. This removes text to speech from the operating system.

Step 2. For the VIC 20: Type LOAD "VIC-20 DEVELOPMENT". For the C64: Type LOAD "COM-64 DEVELOPMENT".

Step 3. For cassette, after loading turn tape recorder off.

To load the Text-to-Speech Operating System while in the Text-to-Speech Development mode.

Step 1. Press the RUN/STOP and RESTORE keys at the same time.

Step 2. Reactivate text-to-speech with "SYS" command from screen (Step 10, operating instructions)

Step 3. The system is now set up for the device number and input/output previously entered. If nothing was specified, the system will be set for device number 3, output.

Step 4. Enter your own game or application program.

To change the device number while in another mode.

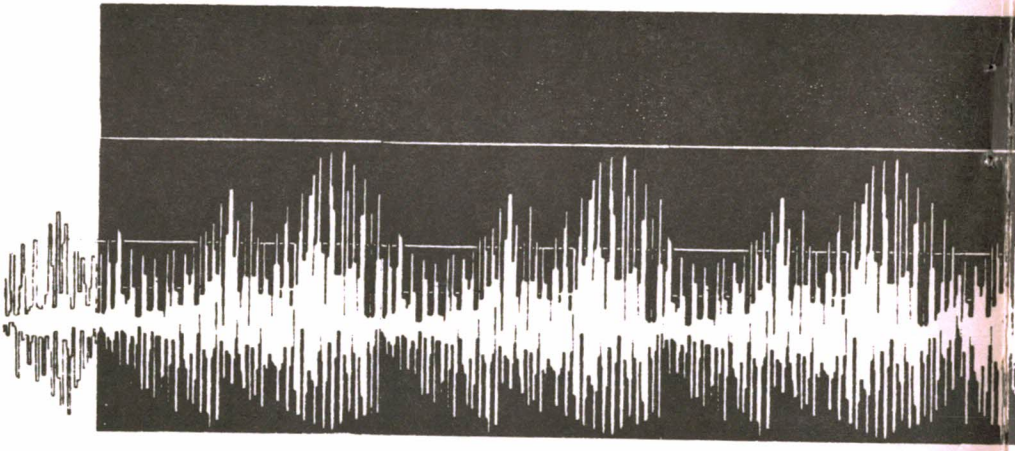
Step 1. Press the RUN/STOP and RESTORE keys at the same time.

Step 2. For VIC 20: type POKE 41521, (device number). For C64 under the KERNAL ROM, type POKE 57905, (device number). For C64 under the BASIC ROM, type POKE 41521, (device number).

To change from input to output or vice versa.

Step 1. Rerun entire text-to-speech program from the beginning.

"THE SPOKEN WORD"



Man's Most Powerful Means of Communication Comes to Your Computer

R.I.S.T. Inc.

P.O. Box 499
Ft. Hamilton Station
Brooklyn, New York 11209
(212) 259-4934