Speech Synthesis Utilizing Microcomputer Control

Fall 1978

Joseph N. Uzel
ju84123@aol.com

Find similar works at: http://stars.library.ucf.edu/rtd

University of Central Florida Libraries http://library.ucf.edu

Part of the Engineering Commons

STARS Citation


This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of STARS. For more information, please contact lee.dotson@ucf.edu.
SPEECH SYNTHESIS UTILIZING MICROCOMPUTER CONTROL

BY

JOSEPH N. UZEL
B.S.A.E., Virginia Polytechnic Institute, 1972
B.S.E., Florida Technological University, 1977

RESEARCH REPORT
Submitted in partial fulfillment of the requirements
for the degree of Master of Science: Engineering
in the graduate studies program of
the College of Engineering
of Florida Technological University at Orlando, Florida

Fall Quarter
1978
SPEECH SYNTHESIS UTILIZING MICROCOMPUTER CONTROL

JOSEPH N. UZEL

ABSTRACT

This report explores the subject of speech synthesis. Information given includes a brief explanation of speech production in man, an historical view of speech synthesis, and four types of electronic synthesizers in use today.

Also included is a brief presentation on phonetics, the study of speech sounds. An understanding of this subject is necessary to see how a synthesizer must produce certain sounds, and how these sounds are put together to create words.

Finally a description of a limited text speech synthesizer is presented. This system allows the user to enter English text via a keyboard and have it output in spoken form.

The future of speech synthesis appears to be very bright. This report also gives some possible applications of verbal computer communication.

Approved by: [Signature]
Director of Research Report
# TABLE OF CONTENTS

List of Tables ........................................ iv
List of Illustrations .................................... v
Introduction ........................................... 1
I. Human Speech Production .............................. 2
II. A History of Speech Synthesis ...................... 5
III. Phonetics ........................................... 9
IV. Electronic Speech Synthesis ......................... 14
   Adaptive Differential Pulse Code Modulation ........ 14
   Linear Predictive Coding ............................. 17
   Formant Synthesis ................................... 20
   Text Synthesis ....................................... 30
V. Microcomputer Controlled Text Synthesizer ........... 33
   Hardware ............................................. 33
   Software ............................................. 37
   Results .............................................. 43
VI. Conclusions ......................................... 47

Appendix A ..............................................
   Text Synthesizer Vocabulary Assembly Listing .......... 49

Appendix B ..............................................
   Text Synthesizer Program Assembly Listing ............ 56

References Cited ..................................... 61

iii
LIST OF TABLES

1. IPA Phonetic Symbols .................................. 10
2. Phoneme Classes ........................................ 13
3. Comparison of Data Rates for Storage of Speech Vocabularies .................................. 16
4. Steady State English Vowel Formants ....................... 23
5. Fricative Spectra .......................................... 30
6. Votrax Phonetic Codes .................................... 36
7. ASCII Code Conversion Chart .............................. 40
### LIST OF ILLUSTRATIONS

1. The Human Vocal System ........................................... 3
2. Block Diagram of the VODER ........................................ 7
3. Block Diagram of a Computer Voice Response System ........... 15
4. Block Diagram of Adaptive Differential Coding of Speech ..... 18
5. Waveforms Coded by 3-Bit (a) DPCM and (b) ADPCM .......... 19
6. Linear Predictive Coding Used in the TMC0280 Single Chip Synthesizer ......................................................... 21
7. Formant Synthesizer Block Diagrams (a) Series (b) Parallel .. 22
8. Relationships Between the First Two Formants of Steady State English Vowels ......................................................... 25
9. Formant Sweeps for the English Diphthongs ....................... 26
10. Stop Consonant Patterns ............................................. 27
11. Parallel Formant Circuit Used in the A1 Cybernetics Systems Model 1000 Speech Synthesizer ........................................... 29
12. Block Diagram of a Text Synthesizer .............................. 31
13. Synthesizer Circuit of the VOX-1 Board ........................... 35
14. Partial Results of the "L" Command ................................ 41
15. Illustration of Program Features .................................. 44
INTRODUCTION

For centuries man has been fascinated with the concept of artificially produced speech. Mechanical analogs of the vocal system produced speech sounds 200 years ago; however, it was the recent invention of the digital computer which made practical speech synthesis feasible.

The purpose of this report is to explore the subject of speech synthesis. Questions to be answered are: (1) How is speech produced in man? (2) What approaches have been taken in the past to synthesize speech? (3) What methods are presently used to electronically produce artificial speech?

In addition to answering the above questions this report also presents a description of a limited text speech synthesizer which utilizes a Votrax speech module and is controlled by a 6800 microcomputer. The control software allows the user to enter English text via a keyboard or to store coded text in a memory table and then have the text output in spoken form by the machine.

Some applications of speech synthesis include talking typewriters, verbal response learning systems, aircraft flight control, and telephone information retrieval systems.
CHAPTER 1

HUMAN SPEECH PRODUCTION

Human speech production is a physiological process which is understood fairly well. Figure 1 shows the human vocal tract and those components which make speech possible. The vocal tract is a non uniform acoustic tube, 16 to 18 centimeters in length, which extends from the glottis to the lips, and varies in shape as a function of time. The major anatomical components causing this time-varying change are the lips, tongue, jaw, and the velum. The velum is a flap which couples the nasal tract to the vocal tract through a trap-door type of action. The nasal cavity is about 12 centimeters long and has an approximate volume of 60 cubic centimeters (Flanagan 1972a).

The vocal system can produce three basic types of sound: voiced, fricative, and plosive. Voiced sounds, such as the vowels, are produced by the vibration of the vocal chords due to air released from the lungs. These vibrations interrupt the airflow and generate a series of sharp pulses that excite the vocal tract.

Fricative sounds (s, sh, f, th) are generated by forcing air through a constricted vocal tract at a high velocity which causes turbulence. Plosives (p, t, k) are produced by closing the vocal tract completely with the lips or tongue, allowing a pressure buildup, and then abruptly opening the closure.
1. The Human Vocal System
These speech sounds have a fairly broad spectrum of frequencies ranging from about 100 Hz to more than 3000 Hz. The vocal system acts as a time-varying filter to produce resonance characteristics.

For voiced sounds, the excitation source is typically pulsive and periodic, with a spectrum whose harmonics decrease in amplitude approximately 12 dB per octave. The vocal tract acts as a filter with poles corresponding to the acoustic resonances known as formants. The resonant frequencies roughly equal the odd quarter-wave resonances of a pipe 17 centimeters long. For a straight pipe of this length, the first three resonant frequencies are 500, 1500, and 2500 Hz. The vocal tract, however, is not of constant diameter, and the formants will not be located exactly at the 1000 Hz intervals described.

When the nasal tract is made part of the transmission system, another resonance pole and an antiresonance zero are introduced at around 1400 Hz. The various voiced sounds of speech are produced by changing the shape of the vocal tract and hence its resonances.

Unvoiced sounds are excited by a noise source that has a fairly broad, uniform spectrum. The source such as the tongue constricting flow in the back of the mouth, effectively alters the length of the vocal tract. Thus the resonances and antiresonances fall at different frequencies (Flanagan 1972b).
CHAPTER 2

A HISTORY OF SPEECH SYNTHESIS

Man has been fascinated with the production of speech for many centuries; however, the first successful artificial speech sounds were produced 200 years ago. In 1779 Christian Gottlieb Kratzenstein constructed a set of acoustic resonators which produced the vowel sounds a,e,i,o,u. The resonators were activated by a vibrating reed which, like the human vocal chords, interrupted a stream of air.

In 1791 Wolfgang Von Kempelen of Vienna, demonstrated an elaborate machine for generating connected speech. However, his machine was not taken seriously by the scientific community because of an earlier chess playing "machine;" a hoax, which involved a legless master chess player as the principle "mechanism." His speech machine was legitimate. A bellows was used to supply air to a reed. This in turn excited a hand controlled resonator that produced voiced sounds. Consonants and nasals were simulated by four constricted passages (Flanagan 1972a).

In 1820 a machine was constructed which could carry on a normal conversation when manipulated by a skilled operator. The machine, built by Joseph Faber, a Viennese professor, was demonstrated in London where it sang "God Save the Queen." Like the Von Kempelen synthesizer, Faber's also incorporated bellows, reeds, and resonant
cavities to simulate the human vocal tract (Atmar 1976).

The first electrical synthesizer was demonstrated at the 1939 World's Fair. Built by Bell Telephone Laboratories, the VODER (Voice Operated Demonstrator) consisted of a periodic buzz oscillator to simulate the vocal chords, and a wide-band noise source to simulate constricted air flow for fricative production. These sounds were modified by passing them through a resonance control box containing 10 continuous bandpass filters that spanned the frequency range of normal speech. Ten finger keys activated gain control potentiometers which modulated the outputs of the filters. Three additional keys provided a transient excitation of selected filters to simulate three types of plosive sound: t-d, p-b, k-g. A wrist bar selected noise or buzz source and a foot pedal controlled the pitch of the buzz oscillator. Figure 2 shows a schematic of the VODER.

The VODER was limited only to demonstrational use due to its complex controls and bulky size. The development of the digital computer provided a boost to the production of practical speech synthesizers. The computer took over the control functions as well as provided greater freedom in modeling the vocal system because of its high speed computational abilities.

Technical advances in integrated circuit design also contributed to the feasibility of practical speech synthesis. Higher density integrated circuits have enabled designers to include more functions in a smaller amount of space.
Fig. 2. Block Diagram of the VODER (Flanagan 1972a)
A speech synthesizer on a single LSI chip is presently a reality. Texas Instruments has produced a P-channel MOS chip, called the TMCO280, which synthesizes speech through a technique called linear predictive coding (LPC). This is a method of analyzing and storing human speech by extracting information from an input speech waveform and storing it in memory. Speech reproduction is accomplished by reconstructing speech from the information stored. This system uses a TMS1000 microprocessor along with two 16 K-byte ROMs, each with the ability to store over 100 seconds of speech. The fact that this circuit has been fabricated on a single chip indicates that very small and inexpensive speech synthesizers will shortly become a reality (Wiggins and Brantingham 1978).
CHAPTER 3

PHONETICS

In order to design or program an electronic speech synthesizer, it is essential to have a knowledge of phonetics.

Phonetics is the scientific study of speech sounds from the standpoint of their production, reception, and symbolization. A word is phonetically described by breaking it into distinctive sound units called phonemes. There are 38 phonemes in the English language (Atmar 1976). However, subtle differences in a phoneme call for another classification, the allophone. These are the slightly varying sounds of a single phoneme usually determined by the position in a word and phonemes preceding and following it. Table 1 gives a list of International Phonetic Alphabet (IPA) symbols and key words demonstrating their use.

In a sense, phonetics is a language in itself. For example, the following are some common American English words with their phonetic transcriptions:

* the əi shall ʃeɪl
  and ænd should ʃʊd
  but ʌt was ʊəz
  to tu can ɡæn
  by ɹaI as æz

Thus, "ʃi waz ʌt houm" is a phonetic transcription of "she was at home."
### TABLE 1
IPA PHONETIC SYMBOLS (Leutenegger 1963)

<table>
<thead>
<tr>
<th>Phonetic Symbol</th>
<th>Key Words</th>
<th>Additional Spellings of Sound</th>
</tr>
</thead>
<tbody>
<tr>
<td>[p]</td>
<td>part</td>
<td>rope, stopped, hiccough</td>
</tr>
<tr>
<td>[b]</td>
<td>be</td>
<td>rubbed, cupboard</td>
</tr>
<tr>
<td>[t]</td>
<td>to</td>
<td>butt, debt, raked, indict, yacht, Thomas, receipt, might</td>
</tr>
<tr>
<td>[d]</td>
<td>do</td>
<td>add, rubbed, could</td>
</tr>
<tr>
<td>[k]</td>
<td>keep</td>
<td>cue, sick, account, lake, ache, walk, khaki</td>
</tr>
<tr>
<td>[g]</td>
<td>give</td>
<td>egg, ghost, guest</td>
</tr>
<tr>
<td>[f]</td>
<td>fame</td>
<td>off, phrase, laugh</td>
</tr>
<tr>
<td>[v]</td>
<td>vest</td>
<td>of, have, Stephen</td>
</tr>
<tr>
<td>[θ]</td>
<td>thin</td>
<td>miss, scent, schism, cinder, psalm, sword, waltz</td>
</tr>
<tr>
<td>[s]</td>
<td>see</td>
<td>zoo, fuzz, raise, scissors, xylophone</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ship, issue, sugar, pension, gracious, ration, champagne, anxious, schottische, conscious</td>
</tr>
<tr>
<td>[rz]</td>
<td>lesion</td>
<td>leisure, azure, negligee</td>
</tr>
<tr>
<td>[ŋ]</td>
<td>he</td>
<td>whole</td>
</tr>
<tr>
<td>[m]</td>
<td>milk</td>
<td>summer, comb</td>
</tr>
<tr>
<td>[n]</td>
<td>no</td>
<td>inn, pneumonia, Wednesday, mnemonic, knife, gnash</td>
</tr>
<tr>
<td>[l]</td>
<td>lake</td>
<td>tell</td>
</tr>
<tr>
<td>[w]</td>
<td>wig</td>
<td>language</td>
</tr>
<tr>
<td>[r]</td>
<td>red</td>
<td>merry, rhetoric, wrist</td>
</tr>
<tr>
<td>[j]</td>
<td>yes</td>
<td>onion</td>
</tr>
<tr>
<td>[tʃ]</td>
<td>chew</td>
<td>cello, witch, feature</td>
</tr>
<tr>
<td>[dʒ]</td>
<td>just</td>
<td>rage, gem, dodge, soldier</td>
</tr>
<tr>
<td>[ŋ]</td>
<td>sing</td>
<td>anchor, handkerchief, tongue</td>
</tr>
<tr>
<td>[i]</td>
<td>see</td>
<td>eat, people, chief, perceive, be, key, phoenix, ravine, Caesar</td>
</tr>
<tr>
<td>[I]</td>
<td>sit</td>
<td>here, hear, sieve, hymn, business, women, guild</td>
</tr>
<tr>
<td>[e]</td>
<td>ache</td>
<td>ain, beige, great, play, grey, gauge</td>
</tr>
<tr>
<td>[ɛ]</td>
<td>end</td>
<td>said, pear, says, heir, leopard, friend, any</td>
</tr>
<tr>
<td>[æ]</td>
<td>can't</td>
<td>laugh, half, as</td>
</tr>
<tr>
<td>[ɜ]</td>
<td>earn</td>
<td>worst, fir, fur, purr, germ, myrtle, journey, colonel</td>
</tr>
<tr>
<td>[ə]</td>
<td>ladder</td>
<td>surprise, sailor, liar</td>
</tr>
<tr>
<td>[ʌ]</td>
<td>up</td>
<td>son, tough, gull, does, blood</td>
</tr>
<tr>
<td>[ɔ]</td>
<td>sofa</td>
<td>succeed, famous, bargain, specimen</td>
</tr>
<tr>
<td>Phonetic Symbol</td>
<td>Key Words</td>
<td>Additional Spellings of Sound</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>[u]</td>
<td>food</td>
<td>rude, whose, through, threw, shoes, group, blue</td>
</tr>
<tr>
<td>[U]</td>
<td>book</td>
<td>could, fully, wolf</td>
</tr>
<tr>
<td>[O]</td>
<td>rope</td>
<td>oak, though, sown, sew, goes, yeomen, shoulder, beau</td>
</tr>
<tr>
<td>[ɔ]</td>
<td>aught</td>
<td>raw, cough, abroad, gone, thought, all</td>
</tr>
<tr>
<td>[ɑ]</td>
<td>farm</td>
<td>hot, honest</td>
</tr>
<tr>
<td>[aɪ]</td>
<td>sky</td>
<td>write, height, aisle, buy, lye, eye, aye, pie, sigh</td>
</tr>
<tr>
<td>[aʊ]</td>
<td>out</td>
<td>bough, crowd, hour, sauerkraut</td>
</tr>
<tr>
<td>[ɔɪ]</td>
<td>boy</td>
<td>broil</td>
</tr>
</tbody>
</table>
Phonemes can be divided into five classes as shown in Table 2. Some definitions are required for a better understanding of the table. A voiced sound was defined earlier as a sound in which air from the lungs causes the vocal chords to vibrate. The pulsating air stream is then resonated in the throat, mouth, and/or nasal cavity.

A diphthong is a voiced speech sound which changes smoothly from one vowel to another in the same syllable. This smooth transition is also called gliding. An example of a diphthong is the "oy" in boy.

A stop is caused by a complete momentary closure of the vocal tract, and then an explosive release of the built up pressure. Thus a voiced stop is a sound in which the released airstream of a stop has a vibratory element.

Nasal sounds result from resonance in the nasal cavity due to a complete closure of the mouth.

Fricatives result from a constriction of the air passage of the vocal tract, resulting in turbulence in the air stream. Fricatives can be voiced or unvoiced, and can include stops.

The h sound is an aspirant, also called a glottal fricative. It is produced by the airstream in the glottis, the opening between the vocal folds.

Affricates are consonant sounds produced by beginning with the vocal organs in the position of a stop. The corresponding pressure build up is released through a fricative constriction. Affricates can be either voiced(dz) or unvoiced(tʃ)(Wise 1958).

A speech synthesizer must be able to produce these sounds and
connect them in such a way that the transition from one phoneme to another is accomplished as smoothly and naturally as possible.

**TABLE 2**

**PHONEME CLASSES**

1. **Voiced**
   a. vowels and diphthongs of vowels
      e, i, o, u, ə, ɛ, æ, ɔ, a, ʊ, ai, au, ɔi, i, ə
   b. liquid consonants
      r, l, w, j

2. **Stops (Plosives)**
   a. voiced
      b, d, g
   b. unvoiced
      t, p, k

3. **Nasal closures**
   m, n, ŋ

4. **Fricatives**
   a. voiced
      z, ʒ, v, ɹ
   b. unvoiced
      s, ʃ, f, θ
   c. glottal (aspirant)
      h

5. **Affricates**
   a. voiced
      ðt
   b. unvoiced
      tʃ
CHAPTER 4

ELECTRONIC SPEECH SYNTHESIS

The main components of a speech synthesis system are shown in Figure 3. The machine is required to speak a message, typically an English text. A synthesis program may access from a vocabulary store, or a set of vocabulary rules, to obtain a description of the required sequence of words. This description is then transferred to a synthesis device which transforms the information into speech.

Many different approaches to speech synthesis exist; however, the techniques are distinguished mainly by the memory storage requirements for the vocabulary and by the complexity of the control rules for generating the speech. Four different techniques illustrate the range of complexity. They are: adaptive differential pulse-code modulation (ADPCM), linear predictive coding (LPC), formant synthesis, and text synthesis. Typical bit-rate comparisons are shown in Table 3.

Adaptive Differential Pulse Code Modulation

The first and simplest technique utilizes a vocabulary composed of human-spoken words whose waveforms are digitally coded. The speech signal is band-limited to $f_N$ Hz (the lowest frequency from which intelligible speech can be reconstructed). The signal is sampled at a rate of at least $2f_N$ samples per second (the Nyquist rate). In communication systems the most familiar example of this
Fig. 3. Block Diagram of a Computer Voice Response System
TABLE 3

COMPARISON OF DATA RATES FOR STORAGE OF SPEECH VOCABULARIES (Flannagan 1976)

<table>
<thead>
<tr>
<th>CODING</th>
<th>BIT RATE</th>
<th>DURATION OF SPEECH IN $10^6$ BITS OF STORAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADPCM</td>
<td>20 k-bits/sec</td>
<td>1 min</td>
</tr>
<tr>
<td>LPC</td>
<td>2.4 k-bits/sec</td>
<td>7 min</td>
</tr>
<tr>
<td>FORMANT</td>
<td>500 bits/sec</td>
<td>30 min</td>
</tr>
<tr>
<td>TEXT</td>
<td>75 bits/sec</td>
<td>240 min</td>
</tr>
</tbody>
</table>

The process is called pulse-code modulation (PCM). In most PCM representations each sample can take on only $2^B$ possible values, where $B$ is the length of the binary code words that represent a quantized sample. The bit rate or information rate equals $2f_NB$, since $B$ bits are required for each sample. The quantization step size, $\Delta$, is normally set so that $\Delta \cdot 2^B$ spans the expected peak-to-peak amplitude range of the input (Rabiner and Schafer 1976).

A better representation of the signal can be obtained using differential rather than uniform quantization. The simplest case is when the quantizer has only two levels. Since a single-bit word can be used to represent these levels, the bit rate equals the sample rate. Such systems are called delta modulators (DM). If the step size of the one-bit quantizer varies to match the amplitude of the difference signal, the system is called an adaptive DM (ADM).
For a sampling rate at just the Nyquist rate, a multibit quantizer must be used. This is differential pulse-code modulation (DPCM). If the quantizer step size varies with speech signal level, we have an adaptive DPCM system (ADPCM) (Rabiner and Schafer 1976). Figure 4 shows a block diagram of an ADPCM system. Logic "L" observes the bit stream produced by the coder and adjusts the quantizer step size to minimize slope overload and distortion. By comparison of Figures 5(a) and 5(b) it is seen that ADPCM provides a better estimate of the signal than does DPCM (Flanagan 1976).

For message assembly, the synthesis program merely pulls the digitally coded words from storage, and supplies them to an ADPCM decoder to produce an analog output. The disadvantage of this system is the large amount of data storage required, due to the high bit rate, as shown previously in Table 3.

**Linear Predictive Coding**

Linear Predictive Coding (LPC) represents a substantial storage saving over ADPCM. In LPC speech synthesis, the bit rate is reduced by a factor of 8 or more.

This is done by using coefficients, calculated from an input voice waveform, to control a time-varying digital filter. These coefficients, called reflection coefficients, are found from a set of predictor coefficients which minimize error. This minimization of error is simply a method of obtaining a parametric representation of the speech signal. The reflection coefficients, along with pitch
Fig. 4. Block Diagram of Adaptive Differential Coding of Speech (Flanagan 1976)
Fig. 5. Waveforms Coded by 3-Bit
(a) DPCM and (b) ADPCM (Flanagan 1976)
period and amplitude information, are transmitted to the synthesizer. Because of the physical constraints of the speech production process, these parameters need to be updated only every 10 to 20 milliseconds, thus resulting in a much lower bit-rate than necessary for ADPCM synthesis.

In LPC synthesis, the digital filter requires two sound sources. A white noise generator is used for unvoiced sound production, and a periodic source is used for voiced sounds. The output of the digital filter is converted from a digital signal to speech by a digital-to-analog converter. Figure 6 shows a diagram of how LPC is used in the TMC0280 synthesizer chip. Markel and Gray(1976), and Morgan and Craig(1976) present more detailed discussions of LPC and include computer programs for its use.

**Formant Synthesis**

The formant synthesizer, unlike the ADPCM and LPC types, does not require a voice input to be digitized and stored. Instead, a circuit is designed to reproduce sounds (usually phonemes and allophones) by manipulating filters and applying rules to produce the proper sound when used in the context of a word or sentence. Figure 7 shows block diagrams of serial and parallel formant synthesizers. Serial and parallel synthesizers both have their relative advantages and disadvantages; thus many different designs have been produced. See Atmar(1976), Holmes(1972), Klatt(1972), and Rice(1976).
Fig. 6. Linear Predictive Coding Used in the TMC0280 Single Chip Synthesizer (Wiggins and Brantingham 1978)
Fig. 7. Formant Synthesizer Block
Diagrams (a) Series (b) Parallel (Atmar 1976)
To model the vocal tract, a formant synthesizer typically has two excitation sources: an impulse generator to simulate a voicing source, and a white noise generator as a fricative source.

For vowel production the output of the impulse generator is fed into the formant filters. For minimal quality speech synthesis, the first three formants, F1, F2, and F3 are required. These are produced by variable digitally controlled resonator circuits. Higher formants are usually produced using fixed resonant circuits, since there is little change in the fourth and higher formants.

Most of the vowels of American English can be produced by the steady state formant frequencies shown in Table 4. These frequencies are approximate, and actual formant resonances vary from individual to individual.

TABLE 4

STEADY STATE ENGLISH VOWEL FORMANTS
(FREQUENCIES IN Hz)

<table>
<thead>
<tr>
<th></th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
</tr>
</thead>
<tbody>
<tr>
<td>heed</td>
<td>250</td>
<td>2300</td>
<td>3000</td>
</tr>
<tr>
<td>hid</td>
<td>375</td>
<td>2150</td>
<td>2800</td>
</tr>
<tr>
<td>head</td>
<td>550</td>
<td>1950</td>
<td>2600</td>
</tr>
<tr>
<td>had</td>
<td>700</td>
<td>1800</td>
<td>2550</td>
</tr>
<tr>
<td>hod</td>
<td>775</td>
<td>1100</td>
<td>2500</td>
</tr>
<tr>
<td>paw</td>
<td>575</td>
<td>900</td>
<td>2450</td>
</tr>
<tr>
<td>hood</td>
<td>425</td>
<td>1000</td>
<td>2400</td>
</tr>
<tr>
<td>who</td>
<td>275</td>
<td>850</td>
<td>2400</td>
</tr>
</tbody>
</table>
The distinctions between various vowel sounds can be seen more clearly by plotting them on a two dimensional graph as in Figure 8. F3 is not shown here because it varies only slightly for all vowels (except those with very high F2, where it is somewhat higher).

Some English vowels, the diphthongs, are characterized by rapid sweeps across the formant frequency space rather than relatively stable positions. Approximate traces of formants F1 and F2 are shown in Figure 9 for the vowels in bay, boy, buy, hoe, and how. These sweeps occur in 150 to 250 ms, depending on the speaking rate. The controlling computer must change the codes to the formant filters in such a way as to provide the desired transitions.

Stop consonant sounds "p," "t," "k," "b," "d," and "g" are formed by the white noise source. The amplified breakdown noise of a zener diode can be used as a simple fricative source. Figure 10 gives the formant frequency patterns necessary to produce the stop consonants when followed by the vowel "ah." The release of the stop closure (start of the noise pulse) is marked by "REL," and the beginning of the voicing sounds is marked by "VO."

A second group of consonant sounds consists of the liquids "w," "j," "r," and "l." These sounds are actually more like vowels. Thus, "w" and "j" can be associated with vowels "u" and "i" respectively. Timing makes the difference; for example, if the vowel "u" is immediately followed by vowel "a" and the rate of F1 and F2 transitions is increased, the result will sound like "wa." For the other liquids, "l" is marked by a brief increase of F3, while "r" is indicated by a
Fig. 8. Relationships Between the First Two Formants of Steady State English Vowels (Rice 1976)
Fig. 9. Formant Sweeps for the English Diphthongs (Rice 1976)
Fig. 10. Stop Consonant Patterns (Rice 1976)
sharp drop in F3 almost to the level of F2.

The nasal consonants "m," "n," and "ŋ" are very similar to the voiced stops "b," "d," and "g" respectively, except for the addition of a fixed nasal formant. This extra formant is generated by an additional resonator tuned to approximately 1400 Hz and having a fairly wide bandwidth. It is only necessary to control the amplitude of this extra resonator during the "closure" period to achieve the nasal quality in the synthesizer output.

Affricates "tʃ" and "dʒ" consist of the patterns for "t" and "d" followed immediately by the fricative "ʃ" or "ʒ" respectively, that is, "tʃ" = "t + ʃ" and "dʒ" = "d + ʒ." The fricatives "s," "ʃ," "z," "ʒ," "f," "v," "s," and "θ" are characterized by a pulse of high frequency noise lasting from 50 to 150 milliseconds. Fricatives and affricates are classed as voiced or non-voiced. The first classification is according to voicing amplitude during the noise pulse, as seen in Figure 10 for voiced stops. Thus "s," "ʃ," "f," "tʃ," and the "θ" in "thin" have no voicing during the noise pulse, while "z," "ʒ," "v," "dʒ," and the "s" in "then" have high voice amplitude. Thus different fricatives and affricates within a group are distinguished by the spectral characteristics of the noise pulse. Table 5 gives the fricative resonator settings needed to produce the various fricative and affricate consonants. Fricative noise amplitude settings are shown on a scale of 0 to 1 (Rice 1976).

Figure 11 shows a partial circuit for a parallel formant synthesizer. This circuit is used in the Ai Cybernetic Systems Model
Fig. 11. Parallel Formant Circuit Used in the Ai Cybernetics Systems Model 1000 Speech Synthesizer (Atmar 1976)
TABLE 5
FRICATIVE SPECTRA

<table>
<thead>
<tr>
<th>RESONATOR FREQUENCY</th>
<th>FRICATIVE AMPLITUDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>( i, \xi )</td>
<td>2500 Hz</td>
</tr>
<tr>
<td>( s, z )</td>
<td>5000 Hz</td>
</tr>
<tr>
<td>( f, v )</td>
<td>6500 Hz</td>
</tr>
<tr>
<td>( \theta \phi )</td>
<td>8000 Hz</td>
</tr>
</tbody>
</table>

1000 Speech Synthesizer (Atmar 1976).

In order to build a quality formant synthesizer, the designer must consider much more than has been presented in the above discussion. The computer interface as well as the controlling software requirements must also be considered. Finally, detailed rules for phoneme and allophone production, pitch, duration, etc. must be programmed into the computer. A discussion of these rules is beyond the scope of this report; however, the interested reader can find this information given by the following authors: Klatt (1976), House (1961), Friedman (1975), Rao and Thosar (1974), and O'Shaugnessy (1974).

Text Synthesis

The final class of synthesizer to be considered here is a text synthesizer, shown in block diagram form in Figure 12. In its simplest form this type of synthesizer consists of a vocabulary stored
Fig. 12. Block Diagram of a Text Synthesizer
in a pronouncing dictionary, each entry of which has the English word and a phonetic transcription of the word.

When a word is to be output or "spoken," the controlling program outputs the sequence of codes to a synthesizer (such as a formant synthesizer) to be converted to an analog signal (Allen 1976). This type of text synthesizer has a limited vocabulary; however, it has the desired feature of converting English text directly into synthesized speech. Although the vocabulary is limited, over 700 words can be stored in 6.6 kilobytes of memory. This is the type of synthesizer under study in the next chapter.

In a more sophisticated form, a text synthesizer can have an unlimited vocabulary. It is able to take any English text, and through an appropriate set of rules, convert the text to speech. Such a synthesizer must therefore use similar rules that a human subconsciously applies when reading. An unrestricted text synthesizer has been built which operates with an accuracy of 90% (Elowitz, et al. 1976).
CHAPTER 5
MICROCOMPUTER CONTROLLED TEXT SYNTHESIZER

This chapter describes a microcomputer controlled, limited vocabulary text synthesizer. A Southwest Technical Products Corporation (SWTPC) 6800 microcomputer is used to control a Votrax (Federal Screw Works) speech synthesizer. The system has the capability of synthesizing speech using one of two methods: (1) via text supplied through a keyboard, or (2) by a table of codes stored in memory.

The keyboard method involves typing in a sentence or series of sentences. The computer looks up each word in a vocabulary dictionary and sends a string of codes to the Votrax unit which outputs the synthesized speech.

In the other method a table of codes must be stored in memory. To produce speech, the computer simply sends these codes to the synthesizer as before. This method enables the programmer to store messages which can be used in any program for prompting or responses by the computer.

Hardware

The heart of the system is the Votrax synthesizer module. This module is a sealed unit which contains circuitry to convert a 6-bit code into one of 64 phonemes or allophones. Thus by proper coding it has the capability of speaking English, Spanish, French, and many
other languages, although the module has been optimized for English speech (The Digital Group 1978). Table 6 lists the phonemes in IPA and Votrax notation and gives the hexadecimal code for each. Note that in the Votrax notation some phonemes end in a number, for example A1. Phonemes with increasing number values decrease in duration. Thus A2 is of shorter duration than A1.

The Votrax module is mounted on a circuit board, the VOX-1, built by The Digital Group. This board was specifically designed to plug directly into a Z80 microcomputer I/O slot, however it can be used with any computer system which has an 8-bit parallel output port and an 8-bit parallel input port. The SWTPC 6800 has parallel I/O ports via a Motorola MC6820 peripheral interface adaptor (PIA); thus the board required no modifications.

In addition to the Votrax module, the board has two FIFO (first in-first out) memory buffer IC's (3351-2) which reduce processor overhead, a single chip audio amplifier (LM384) to drive a speaker and a dual retrigerable one shot (74L123) to indicate the buffer status to the microprocessor. The board also includes a limited speech recognition circuit. Figure 13 shows a schematic of the speech synthesis portion of the VOX-1 board.

It works as follows: hex phoneme codes (V-codes) are sent to the memory buffer IC's via a parallel 8-bit output port. Since the hex codes are 6 bits, the two extra bits are used for buffer control. Bit 7 is used to strobe codes into the buffer, and bit 6 is a master reset. The buffers hold up to 80 codes. These codes propagate
<table>
<thead>
<tr>
<th>VOTRAX</th>
<th>IPA</th>
<th>HEX CODE</th>
<th>VOTRAX</th>
<th>IPA</th>
<th>HEX CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA0</td>
<td>Pause</td>
<td>83</td>
<td>IU</td>
<td></td>
<td>B6</td>
</tr>
<tr>
<td>PA1</td>
<td>e</td>
<td>BE</td>
<td>J</td>
<td>d7</td>
<td>9A</td>
</tr>
<tr>
<td>A</td>
<td>A0</td>
<td>K</td>
<td>k</td>
<td>99</td>
<td>99</td>
</tr>
<tr>
<td>A1</td>
<td>86</td>
<td>L</td>
<td>l</td>
<td>9B</td>
<td>98</td>
</tr>
<tr>
<td>A2</td>
<td>85</td>
<td>M</td>
<td>m</td>
<td>8C</td>
<td>8D</td>
</tr>
<tr>
<td>AE</td>
<td>AE</td>
<td>N</td>
<td>n</td>
<td>8D</td>
<td>94</td>
</tr>
<tr>
<td>AE1</td>
<td>AF</td>
<td>NG</td>
<td>n</td>
<td>8D</td>
<td>94</td>
</tr>
<tr>
<td>AH</td>
<td>a</td>
<td>A4</td>
<td>O</td>
<td>o</td>
<td>A6</td>
</tr>
<tr>
<td>AH1</td>
<td>95</td>
<td>01</td>
<td>B5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AH2</td>
<td>88</td>
<td>02</td>
<td>B4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AW</td>
<td>ò</td>
<td>BD</td>
<td>00</td>
<td>U</td>
<td>97</td>
</tr>
<tr>
<td>AW1</td>
<td>93</td>
<td>001</td>
<td>96</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AW2</td>
<td>B0</td>
<td>P</td>
<td>p</td>
<td>A5</td>
<td></td>
</tr>
<tr>
<td>AY</td>
<td>A1</td>
<td>R</td>
<td>r</td>
<td>AB</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>b</td>
<td>8E</td>
<td>S</td>
<td>s</td>
<td>9F</td>
</tr>
<tr>
<td>CH</td>
<td>tʃ</td>
<td>90</td>
<td>SH</td>
<td>s</td>
<td>91</td>
</tr>
<tr>
<td>D</td>
<td>d</td>
<td>9E</td>
<td>T</td>
<td>t</td>
<td>AA</td>
</tr>
<tr>
<td>DT</td>
<td>84</td>
<td>TH</td>
<td>θ</td>
<td>B9</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>i</td>
<td>AC</td>
<td>THV</td>
<td>θ</td>
<td>B8</td>
</tr>
<tr>
<td>E1</td>
<td>BC</td>
<td>U</td>
<td>u</td>
<td>A8</td>
<td></td>
</tr>
<tr>
<td>EH</td>
<td>e</td>
<td>BB</td>
<td>U1</td>
<td>B7</td>
<td></td>
</tr>
<tr>
<td>EH1</td>
<td>82</td>
<td>UH</td>
<td>A</td>
<td>or</td>
<td>a</td>
</tr>
<tr>
<td>EH2</td>
<td>81</td>
<td>UH1</td>
<td>B2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EH3</td>
<td>80</td>
<td>UH2</td>
<td>B1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ER</td>
<td>ɹ or ɹ</td>
<td>BA</td>
<td>UH3</td>
<td>A3</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>f</td>
<td>9D</td>
<td>V</td>
<td>v</td>
<td>8F</td>
</tr>
<tr>
<td>G</td>
<td>g</td>
<td>9C</td>
<td>W</td>
<td>w</td>
<td>AD</td>
</tr>
<tr>
<td>H</td>
<td>h</td>
<td>9B</td>
<td>Y</td>
<td>j</td>
<td>A9</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>AY</td>
<td>Y1</td>
<td>A2</td>
<td></td>
</tr>
<tr>
<td>I1</td>
<td>88</td>
<td>Z</td>
<td>z</td>
<td>92</td>
<td></td>
</tr>
<tr>
<td>I2</td>
<td>8A</td>
<td>ZH</td>
<td>ζ</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>I3</td>
<td>89</td>
<td>SILENCE/EOM*</td>
<td>BF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Hex BF is used in software to denote an end of spoken message.
Fig. 13. Synthesizer Circuit of the VOX-1 Board
(Property of The Digital Group 1978)
through the buffers and are input to the Votrax module. The module then produces an audio output tone corresponding to the phoneme code and signals the buffers when it is ready for another phoneme.

The audio output of the Votrax module can either be used as an auxiliary input to an external audio amplifier or to drive the on-board audio amplifier. A 5 k ohm potentiometer is used to control the volume.

The dual retriggerable one-shot is used to indicate to the microprocessor, through the input port, whether or not the memory buffers are empty or half full. In addition, a line from FIFO 1 indicates whether or not the buffers are completely full. The microprocessor must monitor the input port and feed phoneme codes to the buffers whenever necessary (The Digital Group 1978).

The SWTPC 6800 computer system has a microcomputer with 32 k-bytes of memory and a dual mini floppy disk system. For input and output a Beehive Medical Electronics Video terminal and a Digital Equipment Corporation Decwriter II were used. The Decwriter enabled hard copies of assembly listings and program output to be obtained at a reasonable rate of 300 baud.

Software

The synthesizer software was designed by the author with two objectives in mind. The first was to find a method of checking out the VOX-1 board and Votrax speech synthesizer. The second objective was to produce a program which could be incorporated in the SWTPC 6800 system as a prompting and verbal message package.
The software can be divided into several parts. The main program is a monitor program which enables the user to choose between several options. Typing a "T" causes a previously stored text (in V-codes) to be spoken. If "L" is typed a list of words in the vocabulary, their memory addresses, and their V-codes are output. An "E" enables the user to input, via the keyboard, sentences or groups of sentences composed of words in the vocabulary list and immediately have them output as synthesized speech. Typing a "D" causes control to be returned to the disk operating system.

The voice message subroutine, VMSG, is responsible for the actual control of the Votrax synthesizer. This subroutine uses an address supplied via the index register, as the starting address of a table of hex V-codes. It sends these codes to the VOX-1 buffers and polls the buffer control lines in order to stop transmission should the buffers become full. VMSG can be used to output synthesized speech of a message stored in hex V-codes in a table anywhere in memory. This is done by simply loading the index register with the address of the first entry and then jumping to the subroutine. The table of V-codes must end with a $BF (a $ is used to denote a hex number) to signify the end of message.

A dictionary of words is located in memory starting at $100. Appendix A gives a 6800 assembly listing of the dictionary. Note that each letter of the alphabet has a 256 byte block of memory allocated to it. This can be called a letter block. The dictionary can be expanded easily by inserting words and codes until a letter
block is filled. The starting address of each letter block increases by $100 which allows quick processor access. For example in the keyboard subroutine when a word is typed in, the first letter (ASCII) of the word is taken and the two most significant bits are masked out. Then two zeros are catenated to the end to form the letter block starting address. Thus if the word "at" is entered, the first 2 bits of the ASCII code for "a", $41 (binary 0100 0001), are masked off leaving $01. When the two zeros are catenated an address of $0100 is given as the start of the "a" block. In this way, each letter block is accessed without going through preceding letter blocks. Each letter block ends with a $BF to signify its end. This method of access provides efficient use of processor time, especially if the dictionary is full.

Table 7 presents an ASCII code conversion chart for ASCII codes with the parity bit (bit 7) equal to zero. By referring back to Table 6 it is seen that all V-codes are greater than $7F. ASCII codes are less than or equal to $7F, thus the processor can distinguish between the ASCII codes and V-codes for words in the dictionary.

The "L" monitor command causes the address, words, and V-codes of all the dictionary words to be printed. A portion of such a listing is shown in Figure 14.

The most complicated subroutine is subroutine ENGLSH which accepts text input from a keyboard and converts it to speech. This subroutine stores the input sentence in a table in ASCII form. When
L
0100 A 86 85 A1 A9
0105 ABLE 85 85 A9 8E 98
010E ABOUT B1 8E 95 96 AA
0118 ADDRESS AE 80 9E AB 82 9F
0125 ADVANCE B1 9E 8F AF 81 8D 9F
0133 AFTER AE 9D AA BA
013C AGAIN B2 9C BB 8D
0145 ALL BD A3 98
014B ALPHABET AF B1 98 9D B2 8E BB AA
015B AM 81 82 80 8C 8C
0162 AN AF 80 89 8D
0165 AND AF 80 89 8D 9E
0170 ANY 82 82 8D 8D BC BC
0179 ARE 95 A3 BA
017F AS AE AF 92
0184 AT AF 80 AA
0189 AVAILABLE B1 8F 86 A1 98 A3 SE A3 98
0200 B 8E BC AC
0204 BAD 8E AE 9E
020A BE 8E AC AC
  
1727 WHEN AD 82 8D
172E WHERE AD 81 85 AB
1737 WHICH AD 8B AA 90
1740 WILL AD 8A A3 98
1748 WILLING AD 8A A3 98 8B 94
1755 WITH AD 8A B9
175C WORKING AD BA AB 99 AC 94
1769 WOULD AD B7 B6 B6 9E
1800 X 81 82 89 99 83 9F
1807 XRAY 80 99 9F AB A1 A1
1900 Y AD 95 80 89 A9
1906 YALL A9 BD A3 98
190E YES A9 82 80 9F
1915 YOU A9 B6 AB
1A00 Z 92 BC AC
1A04 ZERO 92 A1 89 AB A3 B4 B7

NUMBERS 0 THROUGH 9 ARE ALSO AVAILABLE

Fig. 14. Partial Results of the "L" Command
### TABLE 7

**ASCII CODE CONVERSION CHART (PARITY BIT ZERO)**

<table>
<thead>
<tr>
<th>BITS 4 thru 6</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>NUL</td>
<td>DLE</td>
<td>SP</td>
<td>0</td>
<td>@</td>
<td>P</td>
<td>p</td>
<td></td>
</tr>
<tr>
<td>1 1</td>
<td>SOH</td>
<td>DC1</td>
<td>!</td>
<td>1</td>
<td>A</td>
<td>Q</td>
<td>a</td>
<td>q</td>
</tr>
<tr>
<td>2 2</td>
<td>STX</td>
<td>DC2</td>
<td>&quot;</td>
<td>2</td>
<td>B</td>
<td>R</td>
<td>b</td>
<td>r</td>
</tr>
<tr>
<td>3 3</td>
<td>ETX</td>
<td>DC3</td>
<td>#</td>
<td>3</td>
<td>C</td>
<td>S</td>
<td>c</td>
<td>s</td>
</tr>
<tr>
<td>4 4</td>
<td>EOT</td>
<td>DC4</td>
<td>$</td>
<td>4</td>
<td>D</td>
<td>T</td>
<td>d</td>
<td>t</td>
</tr>
<tr>
<td>5 5</td>
<td>ENQ</td>
<td>NAK</td>
<td>%</td>
<td>5</td>
<td>E</td>
<td>U</td>
<td>e</td>
<td>u</td>
</tr>
<tr>
<td>6 6</td>
<td>ACK</td>
<td>SYN</td>
<td>&amp;</td>
<td>6</td>
<td>F</td>
<td>V</td>
<td>f</td>
<td>v</td>
</tr>
<tr>
<td>7 7</td>
<td>BEL</td>
<td>ETB</td>
<td>'</td>
<td>7</td>
<td>G</td>
<td>W</td>
<td>g</td>
<td>w</td>
</tr>
<tr>
<td>8 8</td>
<td>BS</td>
<td>CAN</td>
<td>(</td>
<td>8</td>
<td>H</td>
<td>X</td>
<td>h</td>
<td>x</td>
</tr>
<tr>
<td>9 9</td>
<td>HT</td>
<td>EM</td>
<td>)</td>
<td>9</td>
<td>I</td>
<td>Y</td>
<td>i</td>
<td>y</td>
</tr>
<tr>
<td>A LF</td>
<td>SUB</td>
<td>*</td>
<td>:</td>
<td>J</td>
<td>Z</td>
<td>j</td>
<td>z</td>
<td></td>
</tr>
<tr>
<td>B VT</td>
<td>ESC</td>
<td>+</td>
<td>;</td>
<td>K</td>
<td>[</td>
<td>k</td>
<td>{</td>
<td></td>
</tr>
<tr>
<td>C FF</td>
<td>FS</td>
<td>,</td>
<td>&lt;</td>
<td>L</td>
<td>/</td>
<td>1</td>
<td>/</td>
<td></td>
</tr>
<tr>
<td>D CR</td>
<td>GS</td>
<td>=</td>
<td>M</td>
<td>]</td>
<td>m</td>
<td>}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E SO</td>
<td>RS</td>
<td>&gt;</td>
<td>N</td>
<td>~</td>
<td>n</td>
<td>≈</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F SI</td>
<td>US</td>
<td>/</td>
<td>?</td>
<td>O</td>
<td>—</td>
<td>o</td>
<td>DEL</td>
<td></td>
</tr>
</tbody>
</table>

input, each word must be separated by a space or punctuation mark (, . ?). These denote word boundaries. Each word is looked at by the processor in the sequence entered. As shown above, the first letter determines the letter block. The processor compares the
words within this block with the entered word. If the input word matches a word in the dictionary, the V-codes are stored in a table. If the input word is not in the vocabulary an error message is output on the monitor.

The user denotes the end of an entry by hitting the ESC key. If all words entered are in the dictionary the speech is synthesized by outputting the table of V-codes via subroutine VMSG.

Some of the features of subroutine ENGLISH are:

1. More than one line can be entered as long as each line ends with a space or punctuation, and a carriage return(CR) and line feed(LF)

2. Mistakes in text entry can be deleted via the "RUBOUT" key. The computer will respond with a "#" for each rubout, and the ASCII characters will not be stored in memory

3. A punctuation mark must precede ESC

4. Numbers are treated as words, thus there must be a space or punctuation mark between each single number

5. More than one space or punctuation mark, or adjacent space and punctuation mark between words are not allowed

Appendix B gives a complete 6800 assembly listing for the synthesizer software. Note that the word dictionary and operating program are separated by a large block of memory. This is because the disk operating system software resides in this space. Also observe that there are three tables; ATABLE, VTABLE, and TTABLE, each consisting of a block of 500 bytes. This is to facilitate large text entries. ASCII keyboard entries are stored in ATABLE, V-codes are stored in VTABLE prior to output to the Votrax module, and text is
stored in TTABLE. These tables could be reduced considerably if large text entries are not anticipated.

**Results**

The speech synthesizer system tested works very well and gives good quality speech output. Figure 15 shows a sequence which demonstrates the features of the system software. In (a) the disk operating system is initiated when in the SWTPC monitor mode (denoted by $) by typing a "D." When the disk operating system (FDOS) is ready, the word dictionary, WORDS, is loaded and the synthesis program (VOTRAX) is run. The computer speaks a prompting message, "Enter T, L, D, or E."

When an "E" is entered, the computer responds with, "ENTER SENTENCE. HIT ESCAPE." on the monitor. In (d) a sentence is entered. Note that words as well as numbers are separated by a space or a punctuation mark which denotes a word boundary. When the "ESC" key is pressed the sentence is synthesized by the computer.

Entry (e) shows how a rubout is used to correct a typing error in the word "the." When a word is not in the dictionary of words the computer will respond with an error message as in (f) where the word address is spelled incorrectly. This is also demonstrated in (g).

The software automatically places a pause between words, however extra pauses may be desired. These are denoted as P1 and P2, and their use is shown in (h) and (i). P1 is equivalent to a PA1
a) $D\text{ SWTPC V1.0 (C) 1977}$

FDOS READY

b) LOAD WORDS

FDOS READY

c) RUN VOTRAX

E

ENTER SENTENCE, HIT ESCAPE.
THE ADDRESS IS 1 2 A F.

E

d) ENTER SENTENCE, HIT ESCAPE.
THE ADDRESS IS 1 2 A F.

E

e) ENTER SENTENCE, HIT ESCAPE.
THE ADDRESS IS 1 2 A F.

E

f) ENTER SENTENCE, HIT ESCAPE.
THE ADDRESS IS 1 2 A F.
ADDRESS IS NOT IN VOCABULARY.

E

g) ENTER SENTENCE, HIT ESCAPE.
THE ADDRESS IS 1 2 A F.
ADDRESS IS NOT IN VOCABULARY.

E

h) ENTER SENTENCE, HIT ESCAPE.
THE ADDRESS IS 1 F1 2 P1 A F1 F.

E

i) ENTER SENTENCE, HIT ESCAPE.
THE ADDRESS IS 1 F2 2 P2 A F2 F.

E

j) ENTER SENTENCE, HIT ESCAPE.
I AM A SIXTY 8 HUNDRED MICRO COMPUTER.

E

k) ENTER SENTENCE, HIT ESCAPE.
I AM A VOTRAX VOICE SYNTHESIZER.
I AM TALKING TO YOU NOW.
DO YOU UNDERSTAND?

E

l) ENTER SENTENCE, HIT ESCAPE.
HA HA THAT IS A GOOD ONE.

E

m) ENTER SENTENCE, HIT ESCAPE.
HA HA THAT IS A GOOD ONE.

E

n) ENTER SENTENCE, HIT ESCAPE.
HA HA THAT IS A GOOD ONE.

E

o) FDOS READY

Fig. 15. Illustration of Program Features
shown in TABLE 6 and a P2 is twice the duration of P1.

If a letter other than T, L, D, or E is entered the computer responds with a verbal message, "Error! You must enter T, L, D, or E." This occurs in (j) where "I" is entered.

More than one line can be entered by using a carriage return (CR) and line feed (LF) at the end of a line. The CR and LF are ignored, and to the computer the entry looks like one continuous line. The multiple line entry feature is shown in (l).

Entries (m) and (n) demonstrate that a word can be given different symbols, but will still sound the same since the V-codes are the same.

In (o), the "D" key has been pressed. This returns control to the disk operating system. The listing command, "L," was shown previously in Figure 14. The text "T" command cannot be shown since all output is verbal.

The synthesized speech can be understood very well. It is basically monotonic since the Votrax module used did not have pitch or intensity controls.

There were some problems with the "p" and "q" phoneme sounds, which tended to degrade the speech quality. The "p" sound is synthesized as a "t." Since "p" and "b" are in the same class and are very similar except for duration, it was felt that substitution of the "b" phoneme might be an improvement over the "t" sound. This, however, was not the case.
The "q" sound was also of poor quality. A "q" is made up of "k + w." When used in a word the synthesized sound is that of a "tw." Thus the word "quite" sounds like "twite." This is apparent in every case where a "q" sound is desired. A "g" sound was substituted for the "k," but did not improve the "q" sound.

Since the "p" and "q" sounds are characteristic of the Votrax module, no immediate solution to the problem was found. The Votrax circuit itself would have to be modified, and since the circuit is in a sealed unit, this is not a feasible solution.
CONCLUSIONS

Many different kinds of speech synthesizers are available today. This report has described several different types, and a hardware and software realization of a limited text synthesizer has been presented. This system has shown that very good quality speech synthesis can be achieved on a small scale computer. The number of words in a vocabulary is constrained only by memory size. The system described has the capability of over 700 words in 6.6 k-bytes of memory. In many applications only certain words or phrases are required, and very little memory is utilized, since these can be stored in V-code form in memory tables. The routine to output a message is very short (62 bytes) and can easily be stored in a ROM to be incorporated as a permanent feature of a small scale computer system.

The application of speech synthesis in many different areas is immediately apparent. The speed of the computer in accessing data coupled with a voice output makes telephone information systems a possibility. A Touchtone keyboard can be used as an input device, and the computer can provide requested data by speaking over the phone. There are many applications in the aviation industry, including air traffic control and on-board aircraft warning devices. Likewise, such verbal warnings might be used in automobiles to signal faults in vital systems such as brakes. Another speech synthesizer use includes training and learning devices which
incorporate a processor to evaluate performance and give verbal prompting, encouragement, or correction to the trainee.

Essentially, any area which calls for verbal information output coupled with the processing capabilities of a computer, is a prime candidate for speech synthesis. The reduction in the physical size and increased performance of computers as well as synthesizers points to an exciting era of speaking machines in the near future.
APPENDIX A

TEXT SYNTHESIZER VOCABULARY
ASSEMBLY LISTING
APPENDIX B

TEXT SYNTHESIZER PROGRAM
ASSEMBLY LISTING
TABLE FCB

`MONITOR PROGRAM`

`OPTIONS:

T O U T P U T S  T E X T  L O C A T E D  I N  T T A B L E
L P O S T S  L I S T  O F  W O R D S  A N D  C O D E S
E  E N G L I S H  S E N T E N C E  I N P U T  I S  S P O K E N.
`D  R E T U R N S  C O N T R O L  T O  D O S.

`3400  ORG $3400`
`3400  CE FF04 BEGIN LDX $FF04 000000230`
`3403  FF 8014 STX $8014 INITIALIZE OUT PORT`
`3406  CE 0004 LDX $0004 000000242`
`3409  FF 8016 STX $8016 INITIALIZE IN PORT`
`3413  B6 00 LDA A $0 00000250`
`3411  B7 8014 STA A $8014 00000254`
`3411  DE 3600 LDG $3600 INITIALIZE STACK`
`3414  B6 0D LDA A $1D 00000262`
`3416  BD E1D1 JSR $E1D1 OUTPUT CR`
`3419  B6 0A LDA A $9A 00000270`
`341B  BD E1D1 JSR $E1D1 OUTPUT LF`
`341E  CE 4100 LDX $MNHG 00000278`
`3421  BD 435E JSR $VMSG SPEAK MSG`
`3424  BD E1AC JSR $E1AC GET CHAR`
`3427  BD 44 TST CMP A $44 #?
`3429  B6 03 BNE TTST 00000294`
`342B  BD 2400 JMP $2400 0000029B`
`342E  BD 5A CMP A $5A 000002A4`
`3430  B6 0C BNE TTST 00000306`
`3432  CE 3F00 LDX #TABLE INDEX GETS TAB ADDR.
`3435  BD 435E JSR $VMSG SPEAK MSG`
`343B  BD 435E JSR $ENDCHK 00000316`
`343B  BD 4300 JMP $4300 00000318`
`343E  BD 41C TST CMP A $4C INPUT=LT
`3440  B6 06 BNE ETST 00000326`
`3442  BD 454B JSR $WLST 00000330`
`3445  BD 4300 JMP $4300 00000334`
`3448  BD 45 TEST CMP A $45 INPUT=GT
`344A  B6 06 BNE EMS 00000342`
`344C  BD 4300 JSR $ENDCHK 00000346`
`344F  BD 4300 JMP $4300 00000350`
`3452  CE 411B EMSG LD $MNHG 00000354`
`3455  BD 435E JSR $VMSG SPEAK MSG
`345B  BD 4316 JSR $ENDCHK 00000362`
`345B  BD 4300 JMP $4300 00000366`

`#ROUTINE TO OUTPUT SPOKEN MSG OR TEXT`
`435E  B6 00 VMSG LDA A $0 00000378`
`4360  B7 8014 STA A $8014 RESET FIFOs`
`4363  B6 A0 LDA A $A0 00000386`
`4365  B7 8014 STA A $8014 PREPARE FOR DATA ENTRY`
`436B  B6 8016 PHOEN LDA A $8016 00000394`
`436B  BD 00 AND A $BD MASK FOR BIT 7`
`436D  BD 00 CMA A $0 CHECK FOR FIFO FULL"
436F 27 F7 BEQ PHON  LOOP IF FULL
4371 B6 8016 LDA A $8016  00000400
4374 B4 50 AND A $50  MASK FOR BIT 5
4376 B1 00 CMP A $00  CHECK FOR FIFO EMPTY
4379 B0 2A BNE PHON2  00000422
437A B6 C3 LDA A $8C3  DUMMY PAUSE
437D B7 8014 STA A $8014  00000430
437F B6 43 LDA A $43  RESET BIT 7
4381 B7 8014 STA A $8014  00000438
4384 A6 00 PHON2 LDA A $00  GET CHAR
4386 0A 60 ORA A $60  MASK OFF RESET AND STROBE
4388 B7 8014 STA A $8014  SENG CODE
438B B4 7F AND A $7F  TUNK OFF BIT 7 STROBE
4390 B7 8014 STA A $8014  00000458
4393 B4 3F AND A $3F  MASK BITS 6-7
4395 B2 3F CMP A $3F  END OF TABLE?
4398 B6 01 BNE CONTIN  00000470
439B 39 RTS  00000474
439D 70 CONTIN INX  INCR TABLE ADDR
439F 7E 436B JMP PHON  CONTINUE
43C5 39 RTS  00000486

*ROUTINE TO ENCODE & STORE V CODES IN TABLE
439C A6 00 ENCODE LDA A $00  A GETS VCODE
439E 0B INX  00000502
439F B1 DF CMP A $DF  END OF CODES?
43A1 27 12 BEQ CODEND  00000510
43A4 FF 41AF STX VX  00000514
43A6 FE 41AF LXD VX  00000518
43A9 E7 00 STA A $00  STORE IN V TABLE
43AB 00 INX  00000526
43AC FF 41AF STX VX  00000530
43AF FE 41AF LXD VX  00000534
43B2 7E 439C JMP ENCODE  00000538
43B5 39 CODEND RTS  00000542

*SUBROUTINE ENDCBK
*CHECKS FOR END OF SPOKEN MSG
43B6 B6 8016 ENDCBK LDA A $8016  00000542
43B9 B4 20 AND A $20  MASK FOR BIT 5
43BB B1 00 CMP A $00  CHECK FOR FIFO EMPTY
43BD 26 F7 BNE ENDCBK  00000574
43F1 3F RTS  00000578

*SUBROUTINE ENCGN
*CONVERTS ENGLISH INPUT VIA KEYBOARD
*TO SPEECH
*USE CR AND LF FOR A NEW LINE
43C0 CE 41A0 ENGLNS LXD  $MSG2  00000646
43C3 BD E07E JSR $E07E PROMPTING HSG
43C6 CE 3B00 LXD $TABLE  00000654
43C9 BD E1AC GETIN JSR $E1AC KEYBOARD INPUT
43CC B1 0D CMP A $0D  CR?
43CE 27 F9 DEQ GETIN  YES
43D0 B1 0A CMP A $0A  LF?
43D2 B7 F5 DEQ GETIN  YES
43D4 01 7F CMP A $7F  RUBOUT?
43D6 26 13 BNE SKIP2  00000692
43DB 8C 3B00 CPX $TABLE FIRST ENTRY?
43DD 27 EC DEQ GETIN  00000696
43DF 09 DEX  INCR X TO WRITE OVER ERROR
43E0 B6 0B LDA A $0B  0000059B
43E3 BD E1D1 JSR $E1D1  00000702
43E5 BD E1D1 LDA A $123  00000706
43E8 43C9 JMP GETIN  00000710
43EC 01 1B BNE INEND2  00000714
43ED 27 06 STA A $00  STORE ASCII CHAR IN TABLE
43F0 08 INX  INCR. ADDR
43F2 7E 43C9 JMP GEIN  00000722
43F5 B6 04 INEND2 LDA A $14  END OF TABLE MARK
43F7 A7 11 STA A $0X  00000742
43F9 CE 3000 LXD $TABLE X GETS BEGIN OF VOTRAX TABLE
43FC FF 414E STX VX  00000750
43FF CE 4128 LXD $TEMP X GETS TEMPORARY STORE ADDR
4402 FF 4148 STX TX  00000758
4405 CE 3B00 LXD $TABLE X GETS ASCII TABLE ADDR
4408 A6 00 EOUT LDA A $0X  00000766
440A 08 INX  00000770
440B B1 04 CMP A $04  EMT
440D 26 11 BNE SPCHK2  00000778
440F FE 414E LXD VX  00000782
4412 B6 BF LDA A $BF  END OF V TABLE MARK
4414 A7 00 STA A $0X  00000790
4417 CE 3300 LXD $TABLE START OF TEXT
4419 BD 435E JSR $VMSG OUTPUT SPOKEN TEXT
441C BD 436B JSR ENDCBK  00000802
441F 39 RTS  00000806
**T W L I S T:**

**S U R R O U N T I N E**

**T O O U T P U T A**

**L I S T**

**O F W O R D S**

*IN VOCABULARY.*

**T H E W O R D A D D R, W O R D T H E N D**

**A N D H E X C O D E S**

**A R E P R I N T E D.**

---

<table>
<thead>
<tr>
<th>Address</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4507 E7</td>
<td>00</td>
<td>STA B 0X STORE PAUSE IN V TABLE</td>
</tr>
<tr>
<td>4509 0B</td>
<td>00</td>
<td>INX</td>
</tr>
<tr>
<td>450A FF</td>
<td>414E</td>
<td>STX VX</td>
</tr>
<tr>
<td>450F CE</td>
<td>412B</td>
<td>LDX #TEMP</td>
</tr>
<tr>
<td>4510 FF</td>
<td>4149</td>
<td>STX TX</td>
</tr>
<tr>
<td>4511 FF</td>
<td>4146</td>
<td>LDX AX</td>
</tr>
<tr>
<td>4516 7E</td>
<td>4400</td>
<td>JMP EOUT</td>
</tr>
<tr>
<td>4519 7E</td>
<td>440B</td>
<td>FASJMP JMP NOTW D</td>
</tr>
<tr>
<td>451C CE</td>
<td>412B</td>
<td>RSTR LDX #TEMP</td>
</tr>
<tr>
<td>451F FF</td>
<td>414A</td>
<td>STX TX</td>
</tr>
<tr>
<td>4522 FE</td>
<td>414C</td>
<td>LDX RX</td>
</tr>
<tr>
<td>4525 E6</td>
<td>00</td>
<td>GETB1 LDA B 0X</td>
</tr>
<tr>
<td>4527 0B</td>
<td>00</td>
<td>INX</td>
</tr>
<tr>
<td>4529 C1</td>
<td>7F</td>
<td>CMP B #$7F VCODE?</td>
</tr>
<tr>
<td>452A 23</td>
<td>E9</td>
<td>BLS GETB1 BRANCH IF ASCII</td>
</tr>
<tr>
<td>452C E6</td>
<td>00</td>
<td>GETB2 LDA B 0X</td>
</tr>
<tr>
<td>452E 0B</td>
<td>00</td>
<td>INX</td>
</tr>
<tr>
<td>452F C1</td>
<td>7F</td>
<td>CMP B #$7F VCODE?</td>
</tr>
<tr>
<td>4531 27</td>
<td>E6</td>
<td>RED FASJMP</td>
</tr>
<tr>
<td>4533 C1</td>
<td>7F</td>
<td>CMP B #$7F VCODE?</td>
</tr>
<tr>
<td>4535 22</td>
<td>F5</td>
<td>BHI GETB2 BRANCH IF VCODE</td>
</tr>
<tr>
<td>4537 09</td>
<td></td>
<td>BEX BEX INDEX</td>
</tr>
<tr>
<td>453A FF</td>
<td>414C</td>
<td>STX RX</td>
</tr>
<tr>
<td>453B FE</td>
<td>4140</td>
<td>LDX TX</td>
</tr>
<tr>
<td>453E A6</td>
<td>00</td>
<td>LDA A 0X</td>
</tr>
<tr>
<td>4540 AB</td>
<td>0B</td>
<td>INX</td>
</tr>
<tr>
<td>4541 FF</td>
<td>4149</td>
<td>STX TX</td>
</tr>
<tr>
<td>4544 7E</td>
<td>440B</td>
<td>JMP CPNULL</td>
</tr>
<tr>
<td>4547 39</td>
<td></td>
<td>RIS RETURN</td>
</tr>
</tbody>
</table>

---

**#W L I S T: SUBROUTINE TO OUTPUT A LIST OF WORDS**

**#I N V O C A B U L A R Y.**


<table>
<thead>
<tr>
<th>Address</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>454B B6</td>
<td>00</td>
<td>W L I S T LDA A #$10 CR CODE</td>
</tr>
<tr>
<td>454A BD</td>
<td>01D1</td>
<td>J S R #E1D1 OUTPUT CR</td>
</tr>
<tr>
<td>454B B6</td>
<td>0A</td>
<td>LDA A #$4A LF CODE</td>
</tr>
<tr>
<td>454F BD</td>
<td>01D1</td>
<td>J S R #E1D1 OUTPUT LF</td>
</tr>
<tr>
<td>4547 4F</td>
<td></td>
<td>C L R A A = 0</td>
</tr>
<tr>
<td>4551 4C</td>
<td>ADD1 INC</td>
<td>A = A+1</td>
</tr>
<tr>
<td>4554 B7</td>
<td>4141</td>
<td>STA A SAVA 00001386</td>
</tr>
<tr>
<td>4557 FE</td>
<td>4141</td>
<td>L D X SAVA 00001390</td>
</tr>
<tr>
<td>455A FF</td>
<td>4143</td>
<td>S T X SAVA 00001394</td>
</tr>
<tr>
<td>455D CE</td>
<td>4143 Logic LDX #$1A OUTPUT ADDR</td>
<td></td>
</tr>
<tr>
<td>4566 BD</td>
<td>00</td>
<td>E O O C H JSR #E0CC OUTPUT SPACE</td>
</tr>
<tr>
<td>4567 FE</td>
<td>4143</td>
<td>L D X SAVA 00001410</td>
</tr>
<tr>
<td>4569 A6</td>
<td>00</td>
<td>N C H A R LDA A 0X SAVA 00001414</td>
</tr>
<tr>
<td>456B BD</td>
<td>07</td>
<td>C M P A #$7F ASCII</td>
</tr>
<tr>
<td>456D BD</td>
<td>00</td>
<td>E D J I D L JSR #E1D1 OUTPUT CHAR</td>
</tr>
<tr>
<td>4572 0B</td>
<td></td>
<td>INX INCR INDEX</td>
</tr>
<tr>
<td>4573 7E</td>
<td>4569</td>
<td>J M P N C H A R 00001434</td>
</tr>
<tr>
<td>4576 BD</td>
<td>E415</td>
<td>CODES STA A #T C O D E 00001454</td>
</tr>
<tr>
<td>4579 0B</td>
<td></td>
<td>INX</td>
</tr>
<tr>
<td>457A FF</td>
<td>4143</td>
<td>S T X SAVA 00001466</td>
</tr>
<tr>
<td>457B BD</td>
<td>00</td>
<td>E O O C H JSR #E0CC OUTPUT SPACE</td>
</tr>
<tr>
<td>4580 CE</td>
<td>4145</td>
<td>L D X #$T C O D E 00001454</td>
</tr>
<tr>
<td>4583 BD</td>
<td>00</td>
<td>E O O C H JSR #E0CA OUT2HS (OUT HEX VCODE)</td>
</tr>
<tr>
<td>4586 FE</td>
<td>4143</td>
<td>L D X SAVA 00001462</td>
</tr>
<tr>
<td>458B A6</td>
<td>00</td>
<td>L D A A 0X GET CHAR</td>
</tr>
<tr>
<td>45B0 01</td>
<td>BF</td>
<td>C M P A #$BF END OF TABLE?</td>
</tr>
<tr>
<td>45B0 27</td>
<td>11</td>
<td>B E R CKSAVA 00001474</td>
</tr>
<tr>
<td>45B8 BD</td>
<td>00</td>
<td>L D X RX</td>
</tr>
<tr>
<td>45B9 BD</td>
<td>00</td>
<td>L D X RX</td>
</tr>
<tr>
<td>45B9 BD</td>
<td>00</td>
<td>L D X RX</td>
</tr>
<tr>
<td>45BA BD</td>
<td>01D1</td>
<td>J S R #E1D1 OUTPUT CR</td>
</tr>
<tr>
<td>45BC BD</td>
<td>00</td>
<td>L D X RX</td>
</tr>
<tr>
<td>45BD 7E</td>
<td>455D</td>
<td>L D X SPF 00001502</td>
</tr>
<tr>
<td>45B5 BD</td>
<td>00</td>
<td>L D X RX</td>
</tr>
<tr>
<td>45D2 BD</td>
<td>01D1</td>
<td>J S R #E1D1 CR</td>
</tr>
<tr>
<td>45D5 BD</td>
<td>0A</td>
<td>L D A A #$A 00001514</td>
</tr>
<tr>
<td>45D7 BD</td>
<td>01D1</td>
<td>J S R #E1D1 LF</td>
</tr>
<tr>
<td>45DA BD</td>
<td>01D1</td>
<td>J S R #E1D1 OUTPUT LF</td>
</tr>
<tr>
<td>45DF BD</td>
<td>00</td>
<td>L D X RX</td>
</tr>
<tr>
<td>45D8 BD</td>
<td>00</td>
<td>L D X RX</td>
</tr>
<tr>
<td>45D9 BD</td>
<td>00</td>
<td>L D X RX</td>
</tr>
<tr>
<td>45DA BD</td>
<td>00</td>
<td>L D X RX</td>
</tr>
<tr>
<td>45DB BD</td>
<td>00</td>
<td>L D X RX</td>
</tr>
<tr>
<td>45F6 BD</td>
<td>00</td>
<td>L D X RX</td>
</tr>
<tr>
<td>45F1 CE</td>
<td>41DA</td>
<td>L D X #$MS4 00001534</td>
</tr>
<tr>
<td>45F4 BD</td>
<td>00</td>
<td>L D X RX</td>
</tr>
<tr>
<td>45FB 39</td>
<td></td>
<td>END 00001542</td>
</tr>
</tbody>
</table>
REFERENCES CITED


