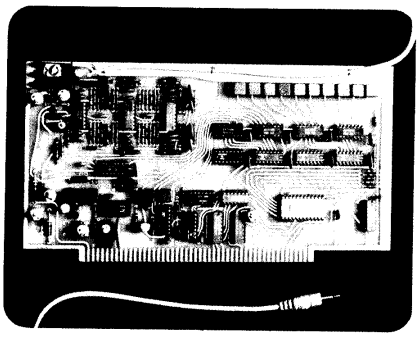
Ai Cybernetic Systems
Model 1000
Speech Synthesizer

September 1976

A Real Conversation Piece . . . .

Ai Cybernetic Systems Model 1000 Speech Synthesizer



The Ai Cybernetic Systems Model 1000 Speech Synthesizer is a revolutionary form of peripheral for the small computer. Interactive computing will never again be the same once your computer has actually spoken to you.

FEATURES

- Very easy to program. The keyboard has been phoneticized in a most intuitive manner.
- Very slow information transfer rate required, typically 25 words/sec.
- Only minimal software support necessary. Less than 50 bytes of assembly or 5 lines of BASIC.
- It can say anything.
- Requires only one Altair ® 8800 bus slot.
- Directly mechanically and electrically compatible with any computer using the Altair bus structure.
- Extremely low price.
- Words and sentences are formed by supplying strings of ASCII characters as would be done in any printing peripheral.
- The phonemes of standard American English are programmed into onboard ROM's. Any one phoneme is requested by supplying its ASCII equivalent character to the synthesizer.
- Speech rate and vocal pitch are easily adjusted to personal preferences.
- Output is a high-level 0.6 V P-P signal capable of driving any standard audio amplifier.
- 12-foot output cable supplied terminating in a standard minature audio plug.

SYSTEM DESCRIPTION

The Model 1000 Speech Synthesizer is a hardwired analog of the human vocal tract. Various portions of the circuit simulate the vocal cords, the lungs, and the variable-frequency resonant acoustic cavity of the mouth, tongue, lips, and teeth.

All of the information necessary to produce the speech sounds of American English has been programmed into ROM's which reside on the synthesizer board. The unit accepts a string of ASCII characters (each character representing a particular speech sound or phoneme) in exactly the same fashion as if it were a printing device.

The small size of the Ai Cybernetic Systems Speech Synthesizer was achieved through the intensive use of modern high density, multi-circuit analog and digital integrated circuits. Because the synthesizer is primarily an analog circuit which is commanded digitally, new programming information is required only at the end of each completed phoneme. The maximum information transfer rate is about 50 bytes/sec (25 bytes/sec typical), figures well within the capacity of any small computer.

Ai cybernetic systems

SYSTEM SPECIFICATIONS

PHYSICAL CHARACTERISTICS

Completely Mechanically and Electrically Compatible with the MITS Altair 8800, IMSAI 8080, and Polymorphic Systems Poly-88 bus structure.

Width:	10.0" (25, 4 cm)
Height:	5.5" (14, 0 cm)
Depth:	0.60" (1, 5 cm) max.

Double-row 50 pin connector required, 0.124" (0, 3175 cm) spacing (not supplied).

POWER REQUIREMENTS

Input Voltages: 7-8 VDC unregulated
+16-19 VDC unregulated
-16-19 VDC unregulated

Power: 2.3 W maximum

INPUTS

8 bit parallel address line
(prewired to address 254_{10})

7 bit parallel data line for transfer of ASCII characters and status information

Status Input and Status Output Lines are used to signal software driver routine.

OUTPUT

Voltage: 0.6 V P-P signal

Impedance: Approx. 1K

Frequency span: 150 Hz minimum
4500 Hz maximum

SYSTEM SUPPORT

Manuals Furnished

Programming Manual, including a Glossary of Commonly Used Words,
Schematics and Theory of Operation

ORDERING INFORMATION

Orders may be placed directly with Ai Cybernetic Systems or with dealers nationwide. Ai Cybernetic Systems will accept Bank Americard or Master Charge. Certified checks or Money Orders are preferred due to increased speed in processing. Delivery 45 days after receipt of payment.

Ai
cybernetic systems
P.O. Box 4691
University Park, New Mexico 88003

Printed by RB Printing Co., Las Cruces, N. M., U.S.A., 5/76.

LIST OF PHONEMES

Phoneme	ASCII Symbol	Usage
Vowels:		
a	A	*Stay, ate, pace*
ae	&	*and, Altair*
ah	(	*father, all*
aw	)	*bought, not, robot*
e	É	*zebra, tea*
eh	'	*enter, Mexico*
er	No.	*number, bird*
oo	U	*moon, too*
o	O	*go, flow*
i	I	*hit, six*
uh	!	*the, computer, shut*
Plosives:		
p	P	*puff, deep*
k	K	*come, phonetic*
t	T	*rip, not, favored*
b	B	*bill, bought*
d	D	*dog, draw*
g	G	*go, gone*
Fricatives:		
f	F	*fancy, puff*
s	S	*saw, less, cycle*
h	H	*hat, hose*
sh	/	*shut, rich (RI.T/)*
th	+	*thaw, earth*
z	Z	*zebra, is (IZ)*
v	V	*vow, David*
Semi-Vowels:		
w	W	*want, one (W!N)*
y	Y	*yacht, yaw*
Liquids:		
l	L	*law, all*
r	R	*raw, robot*
Nasals:		
m	M	*moon, am*
n	N	*not, an*
Others:		
Glottal stop	.	Aspiration before hard sounds (e.g., RO.B . . T)
Pause	space	Normal word spacing
Draw	-	Extension of voiced (vowel, semi-vowel) sound with decay.

PROGRAMMING EXAMPLE

Normal ASCII string to be printed:
W$="I AM A TALKING ROBOT."

Phonetic ASCII string to be spoken:
P$="&&IE AM AE T) . . KEN- RO.B) . . T"

Notice that two *long* vowels, I and A, occur in the sentence. These long vowels are actually combinations (diphthongs) of the pure vowels listed above. The long vowels are programmed as:

AE- EE- &IE OU- EU-

The numbers from one to ten are programmed as:

W!N TOU- T+#E- FO#- F&IE.V
SI . . KZ S'-VIN AE . . T N&IEN T ' ' N

Because there are generally far fewer ways of saying a particular sound than spelling it, the proper combination of phonemes are quickly learned by the programmer, allowing the machine to say anything. Examples are the words "won" and "one", "two", "too", and "to" and "pair" and "pare". Only one phonetic spelling exists for each of these three word homonyms. Programming proficiency comes with the recognition of these phonetic subwords in ordinary speech.

INTRODUCTION

The Model 1000 speech synthesizer is a physiological model of the human vocal tract. It is a true speech synthesizer. And, at the time of this writing, the Model 1000 is by far and away the lowest priced speech synthesizer available.

Because the Model 1000 is a true speech synthesizer, it can say almost anything. Its speech is generated by a string of ASCII characters, each of which has been given a phonetic representation. The produced speech is not digitized human voice. The spoken words produced by the Model 1000 were never spoken by a person. The words are completely synthesized from the basic phonetic units of speech associated with American English.

The information and memory requirements of the controlling computer are lessened substantially from that which would be required to reproduce prerecorded digital speech. Moreover, the vocabulary of the device is essentially limitless rather than being fixed to that which was prerecorded. And the vocabulary can be changed in a very few minutes through reprogramming.

No major software packages are required to operate the Model 1000. The rules of speech synthesis are contained in the hard-wired construction of the device. The goal throughout the design of the Model 1000 was to produce a unit which is accessed and commanded exactly as if it were a printing peripheral rather than a speaking one.

A STATEMENT CONCERNING INTELLIGIBILITY

Intelligibility is both a cultural and contextual process. We tend to recognize speech only when the speaker possesses the same general dialect as we do. It is also a matter of context. Words spoken completely out of context are generally not understood at all. In intelligibility tests we have conducted, we have programmed the computer with 200 different words and let the machine speak them at random. Experienced listeners never mistook a word, but they knew the machine's vocabulary beforehand. Naive listeners who had not heard the machine before, nor knew its vocabulary, correctly understood only a small fraction of the spoken words at first hearing.

Yet, once a listener does grow accustomed to the machine's speech, it becomes surprisingly clear and human-like. And, at that point, the machine is communicating with people in their language, not the machine's. We have found that when the machine speaks, you pay immediate attention to what it has to say far better and more reliably than any system of buzzers and bells. Rather than being part of the general background noise, people recognize the voice instantly and can recall what was said even when concentrating on other matters.

When operating the Model 1000, listen to it at a moderate volume and, ifpossible, perpendicular to the direction of speaker radiation. Listening directly into the speaker will tend to promote concentration on the process of speech synthesis rather than on what is being said. Allow a slight emphasis to the treble and deemphasize the bass. The goal is not high-fidelity reproduction. Rather, it is to create an acoustical radiation pattern similar to that of a human speaker.

WARRANTY

Ai Cybernetic Systems, in recognition of its responsibilites to provide quality products, warrants its products as follows:

All units are guaranteed to meet specifications in effect at the time of manufacture for a period of at least 6 months following purchase. These units are additionally guaranteed to be free of defects in materials or workmanship for the same 6 month period. All warranted units returned to Ai Cybernetic Systems postpaid will be repaired and returned without charge.

This warranty is made in lieu of all other warranties, expressed or implied, and is limited in any case to the repair or replacement of the unit involved.

RECEIVING INSPECTION

Upon arrival, examine the shipping container for signs of possible damage to the contents during shipping. Then inspect the contents for damage. We suggest that you save the shipping container for possible return use. Should the Model 1000 be damaged in transit, please write us at once describing the condition of the unit so that we may take appropriate action.

SPECIAL CARE

The Model 1000 contains MOS integrated circuits (IC8,13,14,15 and 16). The board, and especially these ICs, must be handled with with the appropriate MOS anti-static handling procedures.

SPEECH AND ITS REPRESENTATION

We often tend to think of speech as a uniquely human quality. It is not, of course. Many animals quite clearly communicate their attitudes and desires by means of voiced sounds.

Undeniably, communication by oral speech reaches its pinnacle in humanity. Most significantly, we are the only animal to abstract and symbolize its speech by means of written signs. But that is a most recent development in our history. Moreover, the complex vocabularies of Indo-European languages are even more recent developments. Any language certainly must derive its basic roots from a "look-see" type of speech. Many of these forms of languages still exist today. These forms are characteristic of the tribal languages of Africa, Australia, and the Americas. In these languages, short monosyllabic words are often combined into longer words which take on a completely new meaning of their own. Inflection and context are also varied to provide even further meanings to the same words or word groupings.

Quite often, the translation of each of the basic words of a grouped-word word into English does not convey the proper meaning or provides a somewhat "silly" translation (university in Chinese is simply big-school in its most literal translation). This basic difference in construction of the languages leads to the pidgin-English translations which lend a falsely illiterate or child-like sound to the translated language.

Indo-European languages have taken a different course in their evolution. Generally, a completely new word is either invented or assimilated from an alternate language to clearly demark the word and its meaning from all other like words. The words kindergarten, school, seminary, institute, academy, lyceum, conservatory, uni-

versity and college all precisely define a particular kind of schooling. The words are, for the most part, not related to each other in origin nor are they basic words which readily form alternative, larger words with drastically altered meanings.

English is an especially hybridized language -- a hodge-podge of assimilated languages. We tend to have a specific word for every event or phenomenon. In fact, we often have many. The word cougar appears to have been derived from the Brazilian Tupi Indian word "sussuarana" while puma is from the Quechua language of Peru. North American colonists named the cougar: painter (derived from a mispronunciation of the Greek word "panther"), catamount (derived from the expression "cat-a-mountain"), lion (in confusion with the African lion), Mexican lion, silver lion, mountain lion, mountain devil, mountain demon, mountain screamer, brown tiger, red tiger, deer killer, Indian devil, purple feather, king cat and sneak cat.

But because English is so especially a hybridized language, the basic symbolisms which have evolved to represent a particular sound within a particular language no longer necessarily carry their original pronunciations. In English, the F of OFF is not the v-sound of OF nor is the S of HISS that of the z-sound of HIS. The sound to be associated with written letters simply must be learned almost on a word by word basis in English.

Many phonetic languages do currently exist though. Japanese Katakana script is a purely phonetic representation of a syllabic language. Polynesian languages never developed a written form on their own but rather inherited a phonetic representation from their contacts with European missionaries. In both of these languages, written text can be converted into spoken sounds almost flawlessly once the phonetic representations are learned (a process to which computer-generated speech directly from text would be most amenable).

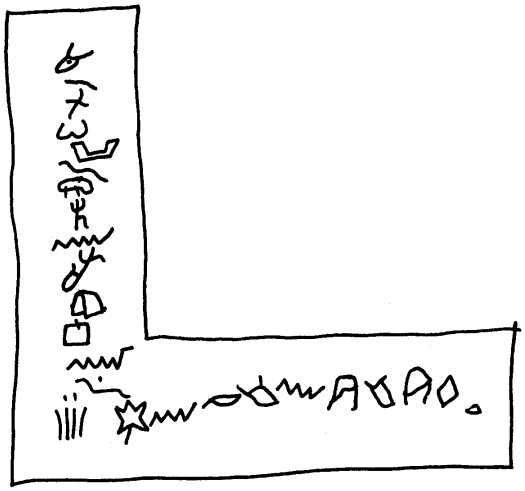
	M	N	S	Z	P	B	T	D	K	G	Y	R	H	W
A	ア	マ	ナ	サ	ザ	パ	バ	ダ	カ	ガ	ヤ	ラ	ハ	ワ
I	イ	ミ	ニ	シ	ジ	ピ	ビ	チ	ギ	キ	イ	リ	ヒ	ヰ
U	ウ	ム	ヌ	ス	ズ	プ	ブ	ツ	ヅ	ク	グ	ユ	ル	フ	ウ
E	エ	メ	ネ	セ	ゼ	ペ	ベ	テ	デ	ケ	ゲ	エ	レ	ヘ	エ
O	オ	モ	ノ	ソ	ゾ	ポ	ボ	ト	ド	コ	ゴ	ヨ	ロ	ホ	ヲ
UN	ン														

JAPANESE KATAKANA SYLLABARY

Three distinct forms of written symbolisms have evolved to represent speech. Undoubtedly, all started as simply aids to memory in keeping track of the seasons, numbers, and the like. More often than not, numbers were represented as multiple strokes while common objects were denoted pictorially (as evidenced in cave paintings, the petroglyphs of North America and Australia, and in the hieroglyphs of Egypt). This sign-writing persists today in chemical formulations, electronics, mathematics, and ordinary English. For example, the and sign (&) and the plus sign (+) have no relationship to their spoken sounds. In this form of language representation every word must have a different symbol and this fact alone carries a tremendous social price. Only a few in any such culture can ever become truly literate in their native language. Often too, a particular sign will retain its meaning while it gathers new spoken sounds as it disperses. A sign spoken one way in the north of China cannot be generally understood in the south even though the sign continues to mean exactly the same thing. But this is true of European languages as well. The +, &, and 1,2,3 ... symbols mean the same thing throughout Europe but are of course pronounced quite differently.

The transition from sign-writing to phonetic (syllabic) writing has occurred apparently independently many places -- in Asia minor, in Senegal, in Cyprus, in Crete, and in Japan. The process of syllabic writing collapses the total number of necessary symbols to only the number of syllables used, a thousand-fold lessening of the symbol set that must be learned.

But even this is unwieldly for a language such as those of the Indo-European group which contain better than a thousand different syllables. Alphabetic writing, necessary for a complex language, has evolved only once in the history of the world and in its earliest form was solely a consonantal script. The earliest writings appear in the Egyptian mines by Hebrew slaves 1500 B.C. Vowels were assumed as we often do now in the commonly occurring pronounciations of "TWX" (twix) and "NOP" (no-op). The greek-speaking world, which had limped



A WORKMAN'S INSCRIPTION ON A MINE SHAFT, SINAI PENINSULA. (ca. 1500 B.C.)

along with the syllabaries of Crete and Cyprus, readily assimilated the plethora of consonantal signs from the Semitic Phoenicians. The earliest Greek writings are the parent of Etruscan, Latin, and other Italic scripts (and hence English).

The script we now use on a day to day basis is the Roman alphabet augmented with the Greek signs K,Y and Z. Yet, even this alphabet does not provide for all of the vagaries of Anglo-American speech -- principally because of the heritage of several Teutonic consonants. To represent these additional consonants, two-letter consonants are used: th (as in thin), dh (as in that), sh (as in shin), zh (like the si in collusion), nh (like the ng in singer, not the ng in finger), with two being compound sounds: tsh (as in catch) and dzh (the j in jam). These representations are the representations which most closely recreate these consonants utilizing the alphabet we have. (But, they are not necessarily the representations that must be used with the Model 1000 to recreate the same or similar sounds.)

All human speech is similar in that the range of possible utterances is determined by the physical and physiological constraints associated with the human vocal tract. But, all human languages are obviously not alike. Only a fraction of all possible sounds are incorporated into any one language. Many African languages contain clicks and whistles which make them almost impossible for an American to reproduce. And of course English contains many distinct phonetic sounds (called phonemes) which are not contained in other languages which provide unending difficulty to a newly-arrived foreigner. To clearly and sharply reproduce any one phonetic sound generally requires a great deal of practice, often years. The differences in phonetic pronounciation between people are learned responses. The differences in produced speech by a speech synthesizer (when accurately programmed) occur because the machine is not totally recreating the physiological conditions of the human vocal tract.

The consonantal phonemes we associate with human speech (see Table II.1 for General American English) are said to be either VOICED or VOICELESS, depending on the action of the larynx. If the vocal cords, which are an integral part of the larynx, vibrate rapidly to produce a buzzing pulse train, the speech is termed voiced. If not, the sound is said to be voiceless and is characterized by the white noise of rushing, turbulent, aspirated air. It is possible to arrange many of the English consonants into a paired series, voiced and voiceless, thusly:

voiced:	b	d	v	g	z	dh	zh
voiceless:	p	t	f	k	s	th	sh

Interestingly, a direct feature of this separation is that it points out an important feature of human speech. Although we write and say "caps", we write "cabs" but we say "cabz". Likewise we write

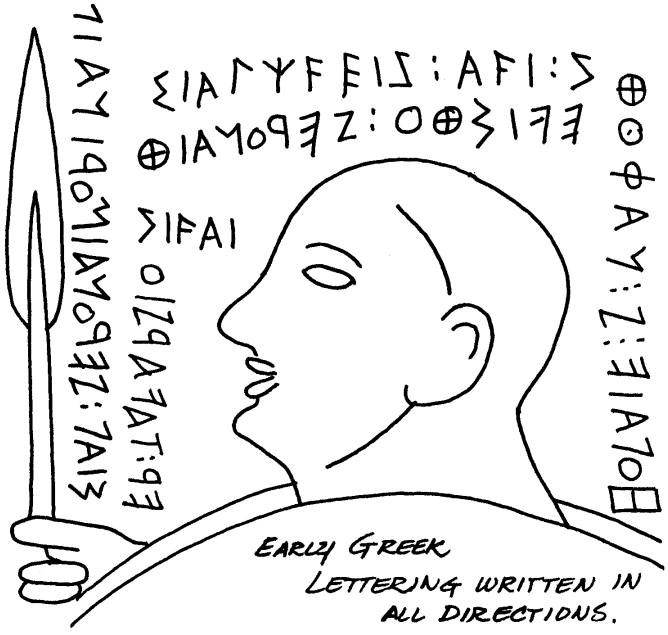


TABLE II.1

THE PHONEMES OF GENERAL AMERICAN ENGLISH

Vowels	Consonants	
ee as in heat	t as in tee	s as in see
i as in hit	p as in pee	sh as in shell
e as in head	k as in key	h as in he
ae as in had	b as in bee	v as in view
ah as in father	d as in dawn	th as in then
aw as in call	g as in go	z as in zoo
u as in put	m as in me	zh as in measure
oo as in cool	n as in no	l as in law
uh as in the	ng as in sing	r as in red
er as in bird	+ as in thin	y as in you
oi as in toil	f as in fee	w as in we
au as in shout		
ei as in take		
ou as in load		
ai as in might		

and say "cats" but we write "cads" and say "cadz". We cannot rapidly transist in our speech from one line to the other. This phenomenon tightly constrains the range of possible speech patterns we can produce.

Vowels are the phonetic sounds produced by the buzzing of the vocal cords in the absence of aspirated air. When the mouth, tongue, teeth, and nasal opening are held in a fixed position, the produced resonant acoustical cavity passes a set of three, four, five, or six preferred frequencies (called formant frequencies). These fixed-position vowels are called "pure vowels" or "steady-state vowels". Vowels which are produced as a direct result of the rapid transition between steady-state vowels are termed "diphthongs". The pronounciation of a certain set of vowels is a learned response. Our inability to adequately reproduce the vowels of another language stems directly from our lack of practice in shaping our mouth, tongue, and lips in the proper fashion at the necessary rate.

It is important to note that the vowel representations that we have in English are not sufficient. There are many more spoken vowels than the symbols A, E, I, O, and U. As can be seen in Table II.1, any one of these symbols can generally represent either a steady-state vowel (as the i in hit) or a diphthong (the i in ice). The proper subsitution of the proper sound into a spoken English word is simply learned.

In many languages, the vowels have only one sound for each written representation. That is, unfortunately, not true of English because of its extraordinarily diverse heritage. Nevertheless, the written, alphabetized word is a direct phonetic representation (with ambiguities, of course) of the spoken word. For the purposes of computer-generated speech, the ambiguities must be removed. Every word must have its own unique representation. But to do so is surprisingly easy -- a new alphabet must be created -- and that is the subject of what follows.

PHONETIC PROGRAMMING

As discussed in Section II, English is represented by an alphabetic script where each symbol more or less is associated with a particular English phoneme. Through time, the symbols have come to be less and less accurate representations of their spoken sounds. Moreover, dialectical changes in the language have provided any one symbol with several alternate sounds (for example, the vowel "er" as it is spoken in the northeastern, southeastern, and western regions of the United States in words such as park, car, and cotter).

For the purposes of speech synthesis, these ambiguities must be removed. The Model 1000 has been programmed with phonemes which are generally reminiscent of General American English, the dialect of English spoken in the midwestern and southwestern regions of the U.S. Each phoneme must be, by necessity, provided with a unique symbolic representation. These representations form the new alphabet (Table III.1) by which speech will be represented. With this alphabet (or something very much like it), any English word can be written down exactly as it is spoken. Adaptation to this form of written speech is particularly easy. And, if it were done universally, as has been often proposed, English would be a much simpler language to learn and spell correctly.

The consonants are generally the commonly occurring consonants as there is only one sound associated with each. There are exceptions, of course. No C, Q, X, or J appear in the list. The C is either spoken as an S or as a K. Thus, the C is a redundant symbol in modern English. The Q is comprised of a transition group of phonemes and is itself pronounced as "KKEUUU".

TABLE III.1

PHONEMES OF THE MODEL 1000

		VOICED	UNVOICED	DURATION
VOWELS				
A	pace	+	-	VL
E	key	+	-	VL
I	hit	+	-	L
O	oh	+	-	VL
U	boo	+	-	VL
&	and	+	-	L
'	eh	+	-	L
#	word, bird	+	-	VL
(	father	+	-	VL
)	caw	+	-	VL
!	the	+	-	L
CONSONANTS				
B	bat	+	+	VS
C	(shortened /)	-	+	VS
D	date	+	+	VS
F	fat	-	+	VL
G	go	+	+	VS
H	hat	-	+	VL
K	key	-	+	VS
L	lot	+	+	L
M	mat	+	+	L
N	not	+	+	M
P	pot	-	+	L
R	robot	+	+	L
S	saw	-	+	VL
T	tee	-	+	VS
V	vee	-	+	VL
W	we	+	+	M
Y	yaw	+	+	M
Z	zero	-	+	VL
+	thin	-	+	VL
/	shoot	-	+	VL
PAUSES				
.	glottal stop	-	-	M
blank	normal word spacing	-	-	VL

VS=very short S=short M=medium L=long VL=very long

Likewise, the X is pronounced as "!'...KZZ", where the symbol "!" represents the vowel "eh". The J sound of JAM is also the "soft" G sound of GENERAL and is simulated in the Model 1000 as a rapid transition between the unvoiced phonemes T, C, and the following vowel. And finally, although a Y is listed in the set of phonemes, the Y takes on several different sounds even when it is not a vowel. The pre-programmed phoneme Y works best with the Y sound of YAW or YACHT. To pronounce these words, the Y is distinctly aspirated. But, in the word YOU, the Y is not aspirated and is best simulated by the rapid transition of the purely voiced sounds represented by the "test" phonemes of 9 and 0 as in 90UUU. It is slightly less well simulated by EUUU.

There are many more vowels than the A, E, I, O, amd U of English. And each must have its own unique representation. Table III.1 lists the vowels of the Model 1000. If the object of these symbolizations were the complete and accurate representation of all of the vowels of English, this list would not be sufficient -- but it is close, and it does represent the principal vowels. The remaining vowel sounds are slight variations of these and are the source of the commonly-occurring dialects.

Every effort was made to choose ASCII symbols which carry a great amount of intuitive weight in the representation of a particular vowel. The number symbol, "#", was chosen to represent the vowel "er" as in NUMBER; the and sign, "&", was chosen to represent the vowel "ae" as in AND. The symbols "(" and ")" represent the vowels "ah" and "aw" in the manner that the tongue is placed in their being spoken (to the back and front of the mouth, respectively). The exclamation mark, "!", is used to indicate the sharp sound of the vowel "uh" while the apostrophe, "'", is used to represent the vowel "eh".

The "long" vowels are, for the most part, combinations of these basic phonemic vowels (termed diphthongs). The long I is programmed as &&IEE; the long A is AE; the long O is OUU; and, the long U is 9OUUU. Only the long E is a basic (or pure) vowel. Pronounce each of these vowels using the phonetic representations of Table III.1. Begin by slowly pronouncing each phoneme carefully and then begin to increase your rate of speech. The phonemes will blend into the sounds we associate with each of the long vowels.

Timing Considerations

The proper selection of timing in the produced speech is only secondary in importance to the proper selection of phonemes. The phonemes are pre-programmed to be of variable lengths (Table III.1). In each instance, the phoneme is programmed for the SHORTEST duration that it is called upon in normal speech. To extend the duration of any single phoneme, all that must be done is simply repeat the symbol as many times as necessary. EEE is 3 times as long as E.

The vowels, as can be seen in Table III.1, are generally longer than the consonants as programmed. Even so, they often must be repeated several times to more accurately simulate their normal pronunciations. The word ENTER is typed as '"N..T###. In this word, the consonant T is a very short burst of high-frequency noise. But, it is highly distinct and is quite noticeable in its absence.

The number of times a phoneme must be repeated is, of course, a function of the speech rate. The SPEECH RATE control (see Section IV) controls the frequency of an on-board phoneme clock. In general, more accurate speech simulations can

be produced by a faster clock and repeated symbols. This is particularly true of words such as ARE when it is encoded as (('###. With a much slower clocking rate, ARE would simply be (## -- without the trill the "eh" sound produces in the rolled R of ARE. A faster clock does, though, require more symbols and hence more string space in the computer (perhaps twice as much). If memory is not an extremely critical problem, we recommend that the faster clocking rate be used. The phonetic representations in the following Phonetic Glossary have been written for a fast clock setting.

Glottal Stops

Glottal stops are periods of speech where all sound is silenced for a brief period. They are particularly characteristic of the period right before a hard consonant. Glottal stops are symbolized in the Model 1000 by the period, ".". As a rule, a consonant becomes "harder" as the duration of a glottal stop is increased between the consonant and its preceeding or following vowels. As an example, have the machine say T))T, T)).T, T))..T, T))...T. In the first word, the following T will not be heard. In the second, it will be heard as a trailing D. In the third and fourth words, it will sound more like a T. This soundless period can be distinctly sensed when you pronounce the words TOD and TOT. In the same manner, a glottal stop between a preceeding consonant and the following vowel(s) will make the consonant appear more distinct and more heavily aspirated. The K in K.AEE will be more apparent than in KAEE.

Plosives

Plosives are one category of consonants. They are so named because they are produced with an "explosion" of air from between

the lips. The unvoiced plosives are P, K, and T; the voiced plosives are B, G, and D. As mentioned in Section II, the plosives form associative pairs on the basis of their spectral qualities (B/P, G/K, and D/T).

In all phonetic programming, it will be important to recognize whether a consonant is voiced or unvoiced. To do so, Table III.1 must be essentially memorized. This is important because of the following general rule:

> PROGRAMMING RULE NO. 1: GLOTTAL STOPS CANNOT FALL BETWEEN TWO VOICED PHONEMES (VOWEL OR CONSONANT) OF THE SAME SYLLABLE.

As examples of this, POT is spelled as P))..T (or P.))...T or P))...T). but POD cannot be spelled as P))..D. Rather, a trailing D is actually pronounced as a soft T. Thus, POD is spelled as P)).T. (In demonstration of this, compare the words GUEST and GUESSED).

The same holds true for words like BIG (BII.K), BOB (B)).P), and BAD (B&&'.T). In each, the trailing voiced plosive must be simulated with its unvoiced counterpart. Again, if more glottal stops are inserted, the trailing plosive becomes harder and begins to sound more like the letter it normally represents.

Frequently, a leading B, G, or D will not appear to be pronounced sufficiently strongly. Often, depending on the following vowel, the addition of one "eh" phoneme directly following the voiced plosive will help (B'RAEE...K for BREAK instead of BRAEE...K). But, bear in mind that this is only an option, not a hard and fast rule.

Fricatives

The fricatives are very much like the plosives and obey similar rules. They too can be grouped into voiced and unvoiced pairs in natural speech. But, they are not totally simulated in that fashion in the Model 1000. The unvoiced fricatives are S, F, / (as in SLASH), + (as in THIN), and H. The voiced fricatives are Z, V, ZH (not present in the Model 1000) and TH (simulated as T.& for words such as THAT (T.&&'..T) and THE (T.&!!)). The voiced fricatives are not as strongly voiced as are the voiced plosives. For that reason, a more accurate simulation is obtained by having two of them, the Z and the V, remain unvoiced.

PROGRAMMING RULE NO. 2: FRICATIVES PLACED BETWEEN TWO VOWELS DO NOT GENERALLY REQUIRE GLOTTAL STOPS TO DIVIDE A WORD INTO ITS PROPER SYLLABLES UNLESS A PLOSIVE FOLLOWS.

Examples of Rule No. 2 are words such as SPACECRAFT (S.PAESS...-K.R&&FF...T) and OCEAN (OU//!!N).

There are two phonemes in English which are not singular sounds but are rather a combination of two distinct and separate phonemes. The J has been discussed earlier. The other is the CH, which is actually a TSH sound. This latter phoneme is simulated well in the Model 1000 using the programmed phoneme sequence TC. The CH phoneme may occur at the beginning or end of a word. CHINA is programmed as TC.&&IEN!! and PATCH is P&&..TC. In each of these words, the CH is a "hard" sound. When it is made softer, it begins to sound as if it were the soft G of JET (TC'''..T).

Nasals

Three nasal consonants exist in English, the M, N, and NG.

The first two are used as they would be expected to be. Often, they will have to be repeated in order to produce words of the proper duration (as AAMMM for the word AM). The third nasal, NG, does not exist in the Model 1000. Rather, it has to be simulated by the EEN sequence (as in N&IEN..TEEEN for NINETEEN).

Laterals

The two English laterals are R and L. It is important to note:

> PROGRAMMING RULE NO. 3: THE LATERAL CONSONANTS L AND R ARE PRECEEDING CONSONANTS ONLY. THE TRAILING R OF NUMBER IS ACTUALLY THE VOWEL "ER" AND THE TRAILING L OF BELL IS NOT THE L OF LOT.

To simulate the trailing L, some modification of the sequence '''IILL (ELL) must be used (as in H'''LL.L(OU for HELLO). As was true of the voiced plosives, the R may not always be found to be sufficiently distinct when pronounced. For some words, the foremost R must be "rolled". To do this, one of these two sequences often works: 'R as in B'RAEE...K or the "test" phonemes 46 as in 46OU..B))...T for ROBOT. The leading R is not pronounced the same in all words. The R phoneme works quite well for READY (RR'''.TEE). Compare the pronounciation to yourself of these three words -- READY, ROBOT, BRAKE. In most American dialects, there is a difference in the trill associated with these Rs.

Semi-vowels

The semi-vowels are the final category of consonants. There are two of them, the W and the Y. As with the laterals, the following

is true:

> PROGRAMMING RULE NO. 4: THE SEMI-VOWELS W AND Y ARE PRECEEDING CONSONANTS ONLY. WHEN THEY TRAIL, THEY ARE INCORPORATED INTO THE PRECEEDING VOWEL (AS IN CAW, WHY, OR BATTY).

The W works almost always as it would be expected to. The Y does not because there exist two Ys in English, an aspirated Y (YACHT) and a non-aspirated Y (YOU). For the second, the sequence 90 works well (as in 90UUU for YOU).

The "Test" Phonemes

Ten test phonemes have been programmed into the read-only memory to allow the testing of the ten active filters of the Model 1000. Each "test" phoneme is programmed to be a short voice-only burst through the selected filter network. The filters are numbered from 1 to 0, 1 being the lowest frequency and 0 being the highest. The numbers (in ASCII) are the command codes for these phonemes. Although they were originally programmed only for test purposes, they have been found to work quite well in the simulation of some English phonemes.

Pitch

The pitch bit (bit No. 6, decimal 64) is to be added to each ASCII character to determine the rise and fall of pitch as a word is pronounced. Generally, this is done by specifying a second string of characters and writing a routine to set the pitch bit on each phonetic symbol by melding the information contained in the pitch and phoneme strings.

Only voiced phonemes need have the pitch specified. On voiceless phonemes, the information is not used.

> PROGRAMMING RULE NO. 5: USE THE PITCH BIT SPARINGLY, GENERALLY AT THE BEGINNING OF IMPORTANT WORDS OR SYLLABLES IN A SENTENCE. PITCH VARIATIONS HELP MAKE THE BEGINNING AND END OF WORDS MORE DISTINCT. GENERALLY, THE LAST WORD OF A SENTENCE IS ALWAYS SET AT LOW PITCH, EXCEPT IN THE CASE OF A QUESTION WHERE IN THE LAST HALF OF THE LAST WORD THE PITCH IS SET TO RISE ABRUPTLY.

General Advice

Become familiar with the pronounciation symbols of a dictionary. The phonetic spellings presented there offer excellent clues as to the specific phonetic spellings to be used with the Model 1000. No timing information is given, of course. That, along with the location and duration of the glottal stops must be worked out for each word. But, using the rules presented here and the examples which follow in this Section, the procedure becomes quite easy with practice.

At first, do not spend long periods programming the machine. You may find it somewhat difficult in the beginning and after an intensive period of listening, every word will come to sound very good or very bad. Breaks help. They offer a chance to forget previous predujices and allow you to listen to the programmed sentences more objectively.

Pay special attention to both word and sentence timing. It is something we do in our own speech without thinking. But it is extremely important in producing phrases of maximum intelligibility.

Proper pitch control also augments intelligibility greatly.

Play the device at a low to moderate volume, about that of natural speech. As the volume is reduced, the noise level should be increased relative to the voice source (see Adjustments, Section IV). Whispering is simply the passage of only noise in the absence of voiced excitation. The design of the Model 1000 is such that it will always seem to be slightly more intelligible at low to moderate volumes relative to the general background. Overly loud volumes allow you to too heavily concentrate on the process of speech synthesis rather than on what is being said.

Use short phrases and sentences. If possible, use words which tend to contain unvoiced consonants. These are the words which contain hard sounds. They are easy to say and understand. Brand names are often chosen for these properties (TEKTRONIX is T'''...K.TR(NII...KZZ and AMPEX is &&MM..P'''...KZZ).

And finally, listen to whay you actually say rather than to how a word is spelled. COMMAND is said as if its first letters were K."uh" rather than CO. Programming will become particularly easy once you begin to identify the proper phonemic sounds.

INTEGRATION INTO HIGH-LEVEL LANGUAGES

The Model 1000 has been designed to be as easy as possible to integrate into existing high-level language programs. But because several options are available, this section sould be read carefully.

The address of the Model 1000 has been hardwired to be decimal 254, the lower half of the sense switch address in the Altair 8800 (TM) and the IMSAI 8080 (TM) computers. To command the Model 1000 to produce any one phoneme (say perhaps an E, ASCII decimal 69), all that need be done is output the symbol to address 254 decimal.

In 8080 machine code, this would be:

```
076  MVI,A   LOAD A
105  069     WITH CHARACTER
323  OUT     OUTPUT TO
376  254     MODEL 1000
```

In MITS BASIC, it is simply:

```
OUT 254,69
```

Under these conditions, the E phoneme would be produced indefinitely. A phoneme duration clock is contained on the board as an integral part of the Model 1000. Its output is a single bit. A "one" indicates that the device is busy, a "zero" that the machine is ready for new data.

To produce properly timed speech, new phoneme symbols must

be outputted to the Model 1000 as quickly as possible once the device signals that it has completed the previous phoneme. The busy flag is outputted to the edge connector on the DIO bus line and (optionally) on the XRDY line. The use of the XRDY line allows a smaller resident program to supply phonetic characters on demand.

Rather than sense the busy flag with software, when the XRDY line is pulled, the computer is expected to come to a NOT READY state and stop execution for the duration of the phoneme string. This strategy works particularly well with BASIC. Consider a string of phonetic characters, V$. By parsing each character out of the string one at a time, a tight loop can be written thusly:

```
FOR II=1 TO LEN(V$)
C$=MID$(V$,II,1)      PARSE ONE CHARACTER
WD=ASC(C$) AND 63     MASK OFF UPPER BITS
OUT 254,WD            OUTPUT TO DEVICE
NEXT II
```

For the duration of the phoneme, the computer comes to a halt on the OUT 254,WD statement. No other operation can be performed, of course, during this time, but this is the very simplest method of integrating the device into BASIC. It DOES NOT work with dynamic memories (see Section IV). To establish this mode of interfacing, a jumper must be wired on the board (again, see Section IV).

It is possible to determine the status of the Model 1000 in BASIC (with XRDY not connected) in this fashion:

```
100 FOR II=1 TO LEN(V$)
101 WD=ASC(MID$(V$,II,1)) AND 63
102 OUT 254,WD
```

```
103 CK=INP(254) AND 1
104 IF CK=1 GOTO 103
105 NEXT II
```

But BASIC is slow. It does not work nearly as well as a machine code subroutine.

To establish a machine code subroutine in BASIC, a link is built through the USR command. In MITS BASIC, locations 73 and 74 must be set to contain the starting address of the machine language routine. For the programming examples of Appendix I, the upper limit of BASIC is set at 12200. The subroutine is loaded at 12201 (47,169 in split-address decimal where 47*256+196=12201). The machine-language routine presented on the opposite page accepts an ASCII character which has been stored in the E register by the USR routine, checks device status, outputs the character, and then returns to BASIC control. The routine also waits through 20 dummy loop cycles before outputting the character to the device. This is to allow any residual energy stored in the filters from the last vocal pulse to decay before the filters are switched. This feature is somewhat optional, but it is recommended. Clicks in the speech may sometimes occur otherwise.

Other than for the first few op code commands, this routine can be easily integrated into an assembly language program. All that need be done in this latter instance is establish a routine to provide the phonetic characters one at a time to this subroutine.

While the routines discussed here are specific to 8080 machine code and MITS BASIC, similar routines can be easily written in almost any programming language.

MACHINE CODE SUBROUTINE OF APPENDIX I PROGRAMS

DECIMAL LOCATION	OCTAL LOCATION	OCTAL CODE	DECIMAL CODE	OP CODE	COMMENTS
12201	57,251	041	033	LXI, H	----------
12202	57,252	261	177	026	INITIATION
12203	57,253	057	047	057	RECOMMENDED
12204	57,254	345	229	PUSH H	BY
12205	57,255	052	042	LHLD	
12206	57,256	004	004	004	MITS FOR
12207	57,257	000	000	000	BASIC
12208	57,260	351	233	PCHL	----------
12209	57,261	333	219	IN	INPUT
12210	57,262	376	254	376	DEVICE STATUS
12211	57,263	346	230	ANI	
12212	57,264	001	001	001	MASK &
12213	57,265	312	202	JZ	TRY AGAIN
12214	57,266	261	177	261	IF BUSY
12215	57,267	057	047	057	
12216	57,270	346	230	ANI	CLEAR A
12217	57,271	000	000	000	
12218	57,272	306	198	ADI	SET UP
12219	57,273	024	020	024	DELAY &
12220	57,274	326	214	SUI	LOOP FOR
12221	57,275	001	001	001	20
12222	57,276	302	194	JNZ	TIMES
12223	57,277	274	188	274	
12224	57,300	057	047	057	
12225	57,301	173	123	MOV A,E	FETCH PHONEME
12226	57,302	323	211	OUT	& OUTPUT TO
12227	57,303	376	254	376	DEVICE
12228	57,304	311	201	RET	

A PHONETIC GLOSSARY OF COMMONLY USED WORDS

A glossary of slightly over 100 commonly used words is presented here. The words are spelled phonetically for a fast clock setting and were written to stand alone. When placed in the midst of a sentence, some of the words may need to be shortened to better simulate the way they are spoken in that context. These words were chosen because they are words we have used a great deal and because they contain a wide variety of sounds. By comparing a desired word to a word in the list which has a similar pronunciation, the synthesis of the unknown word should be made easy.

GOOD	G'!!.T
BAD	B'&&'.T
OK	OU...K.AEE
NEXT	NN''...KZ.T
COMMAND	K.!!.M&&IN.T
TALK	T)...K
TALKING	T)...KEEN
SPEAK	S.PEE...K
COMPUTER	K.!!M..PEU..T###
ROBOT	46OU..B)...T
Ai	AEE &IEE
CYBERNETIC	SS&IE.B##..N''..TII...K
SYSTEMS	SSIISS..T.''MMZZ
ALTAIR	&&L..T''###
IMSAI	IIMMSS&&IEE
BASIC	BAE.SII...K
FORTRAN	F.O##..T.R&&NN
ASSEMBLY	((.SS''MM..BLEE
MACHINE	M!!.//IENN

PROGRAM	P.ROU..GRR&&MM
PHONETIC	F.OO.N''..TII...K
ASCII	&&ZZ...KEE
STRING	ST.RAENN
NUMBERS	N!!M..B##ZZ
ELECTRONIC	EE..L''...K.TR(NII...K
ELECTRIC	EE..L''...K.TRII...K
LOGIC	L(..TCII...K
CRASH	K.R&'/C
LANDING	L&&NN.TENN
ENTERPRISE	''N..T###..P.R&&IEZZ
KLINGON	K.LENN..G'((NN
SENSOR	SS'''NNSS###
SCAN	S.K&&INN
SECTOR	SS'''...K.T##
STARBASE	S.T(('##..BAESS
PLANET	P.L&&..N''..T
ATLANTIC	&&..T.L&NN..TII...K
PACIFIC	P.!!SSIFII...K
NORTH	NOO##.++
SOUTH	SS(U.++
POSITIVE	P.((ZZII..TII.V
NEGATIVE	N'''..G''..TII.V
HELLO	H'''LL.L(OO
HOW	H.(UU
YOU	90UUU
ENTER	'''N..T###
TAKE	T.AE...K
THE	T.&!!
THAT	T.&&'..T
AN	ANN
IT	II..T
IS	IIZZ
AM	AAMMM

ARE	(('###
WAS	W!!ZZ
WERE	W!!###
BE	B'EEE
BEEN	B'''INN
HAS	H&&ZZ
HAVE	H&&'V
HAD	H&&'.T
DO	DEUU
DOES	D'&!!ZZ
DID	T8)II.T
SHALL	/&&''LL
WILL	W'''ILL
SHOULD	/(!!.T
WOULD	W(!!.T
MAY	MAEE
MIGHT	M&&IE..T
MUST	M!!SS..T
CAN	K.''IINN
COULD	K.(!!.T

A	AAEE
B	B'EEE
C	SSEEE
D	DEEE
E	EEEE
F	'''.FF
G	TC'EEE
H	AE..T/
I	&&&IEE
J	TCAAEE
K	KAAEE
L	''IILLL
M	'''MMMM

N	''IIN
O	OOUUU
P	P.EEEE
Q	KKEUUU
R	((''###
S	'''SSS
T	T.'EEE
U	9OUUU
V	V'EEEE
W	D!!..B!!LL..9OUUU
X	''...KZZ
Y	WW&IEE
Z	Z'EEEE

ZERO	ZZ##.ROU
ONE	W!!!N
TWO	T.OUU
THREE	T.+#EE
FOUR	F.O##
FIVE	F.&IE.V
SIX	SSI...KZZ
SEVEN	SS'''.VIIN
EIGHT	AE..T
NINE	N&IEN
TEN	T''IN
ELEVEN	ELL'''VIIN
TWELVE	TW'''LL.V
THIRTEEN	++##..TIENN
FOURTEEN	F.O##..TIENN
FIFTEEN	F.IIFF..TIENN
SIXTEEN	SSII...KZZ.TIENN
SEVENTEEN	S'''.VIN..TIENN
EIGHTEEN	AE..TIENN
NINETEEN	N&IEN..TIENN

TWENTY	T.W'EN..TEE
THIRTY	++##..TEE
FORTY	F.O##..TEE
FIFTY	F.IIFF..TEE
SIXTY	SSI...KZZ..TEE
SEVENTY	S'''.VIN..TEE
EIGHTY	AE..TEE
NINETY	N&IEN..TEE
HUNDRED	H!!NN..DR!!.T
THOUSAND	+.(OUSS''N.T
MILLION	M'ILL90''NN

THEORY OF OPERATION

The Model 1000 has been designed to be perfectly compatible with the Altair (TM) bus structure. Moreover, it has been prewired to operate at address 254, the lower (control) half of the sense switch address. Data is transferred to the device through the use of SOUT. Likewise, data is outputted onto the bus with the use of SINP. No separation exists between control functions and data; received data activates the proper control signals.

IC4 continuously monitors the address bus. IC4 comes true (goes low) when address 254 is sensed. If SOUT is detected to be simultaneously true (by IC5, part A), a one-shot multivibrator constructed from IC5 (parts B and C) is triggered. The phoneme duration clock (IC6) is initiated with the triggering of the one-shot. Switches 13/14, 11/12, and 9/10 of IC16 set in the duration of the clock from information contained in the read-only memories (ROM's). The status of the clock (busy (0) or not busy (1)) is placed on the DIO bus line when requested by a SINP command at address 254. At the user's option, XRDY can also be pulled low by wiring jumper J1 closed. This modification lessens the software demands necessary to monitor the status of the Model 1000 (see Section III), but it appears to work well only when the computer uses only static memory. Because most dynamic memories also use the XRDY line, memory refresh cycles appear to become disturbed when some manufacturer's dynamic memories are in place.

A second jumper, J2, is also available. This jumper superimposes the vocal source pulses from the voice generator, IC21, on the device status lines, DIO or XRDY. This jumper is prewired to be in place by the factory. It's being there allows the

resident software to sense that a vocal pulse is in progress and thus delay a change in phoneme until the present one is completed. Not doing so generates "clicks" in the outputted speech because of a rapid change in the stored energy in the following formant filters (see Section III).

Any desired phoneme is transferred to the device as an ASCII character on the Data Out bus. The received data is latched on the board by IC8. The ASCII character is used as the address to the ROM's (IC9, IC10, IC11, and IC12). The ROM's are each a 32 X 8 matrix of encoded information necessary to adjust the resonant filters, vocal sources, and duration timing associated with a particular phoneme. The ROM's interface their encoded information to the analog circuit through an array of 16 analog switches, IC13, IC14, IC15, and IC16 (shown digitally as a solid-line block and as analog components as a dotted-line block.)

The remainder of the circuit is an analog model of the human vocal tract. Two vocal sources are used to represent (1) the voiced sounds produced by the larynx (IC21 and IC20, part B) and (2) the non-voiced sounds of rushing (aspirated) air (CR4, Q2, and IC20, part A). The selection of either or both of these sources is determined by the programmed information contained in the ROM's. These vocal excitation sources are combined in an adder circuit (IC19, part D) and are then fed to an array of ten active filters (IC17, IC18, and IC19) which simulate the formant frequencies associated with the preferred energy passage of the resonant cavities of the mouth, nose, tonque and teeth. The resonant frequencies of these filters have been chosen so as to generate each of the phonemes of American English when commanded to do so by the ROM's. The "Q" (quality factor) of each filter network has been adjusted to allow for the proper decay rate during a phoneme change and thus produce a smooth transition between phonemes.

The outputs of the active filters are summed by IC19 (part C) where the produced speech is spectrally conpensated to more closely match the acoustical properties of a human voice.

The pitch of the voice source is modulated by two separate inputs. The first is by information contained in the ROM's activating the analog switch IC16, pins 16/15. This pitch variation is automatic and simulates the rise and decay in pitch with each voiced syllable. The second source of additional variation is induced externally with Data Bit No. 6. An FET, Q1, is modulated in conductivity by the presence or absence of bit 6 in order to change the pitch frequency as desired by the programmer.

Four potentiometers located toward the bottom left of the board (component side) control (from left to right): noise level, pitch frequency, voice level, and speech rate (Fig. 4.1). These have been adjusted by the factory to optimal levels, although changes in pitch frequency and speech rate do not significantly effect the quality of speech and can be adjusted to personal preferences.

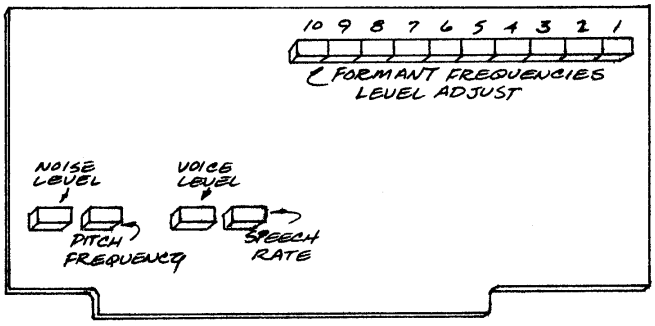


FIG. 4.1. Potentiometer Locations.

ADJUSTING THE MODEL 1000

We strongly recommend that you do not perform any adjustments to your Model 1000 unitl you have thoroughly read this manual and have had sufficient time to acclimate to its particular mode of speech.

Of all of the potentiometers, only the bottom left four should ever be adjusted. The Formant Frequency Level Adjust potentiometers have been set by the factory for optimum operation.

Two of the potentiometers, Speech Rate and Pitch Frequency, are moderately insensitive to adjustments in that speech quality is not greatly degraded. These, too, have been adjusted to levels at which it is felt that maximally intelligible speech is produced. After a period of time, some experimentation with these adjustments may produce a speech which you may find to be more personally pleasing.

The remaining two potentiometers, Noise Level and Voice Level, are very critical adjustments in determining the quality of produced speech. We have found that when the synthesized speech is outputted through a small speaker, in general, the noise level has to be increased to produce maximally intelligible speech. This control is particularly sensitive. Small excursions will make a great deal of difference. The control should be turned CW with the machine saying something like "I HAVE NINE PROGRAMS THAT TALK" (&&IEE H&&.V N&IEN P.ROU..GR&&MMZZ T.&&'..T T)...K) (See Programming Example No. 3, Appendix I) until the N's, M's, and R's begin to take on a distinctly rough sound and the V of HAVE and

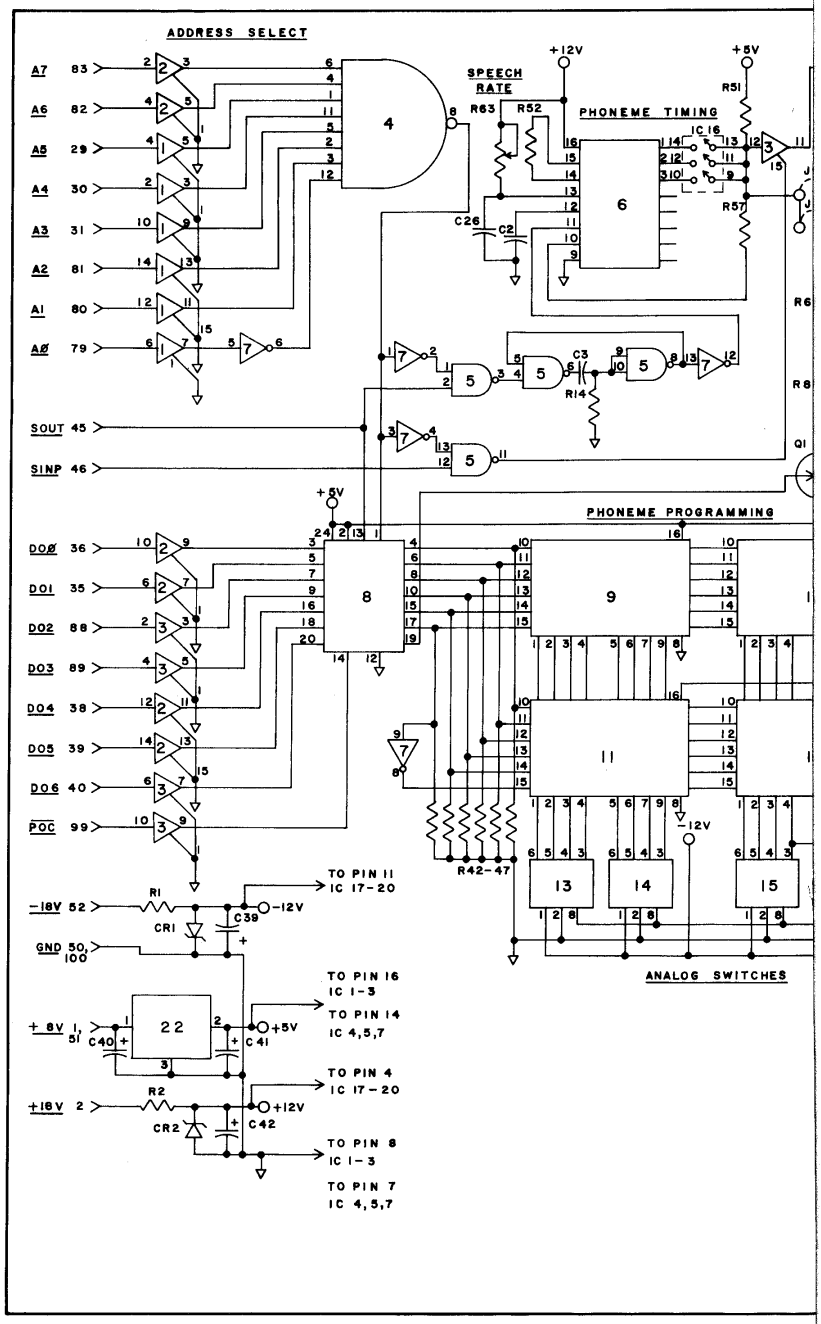
the K of TALK become overly apparent. Reduce (CCW) the noise level until the roughness disappears in the N's and the K and V become more natural.

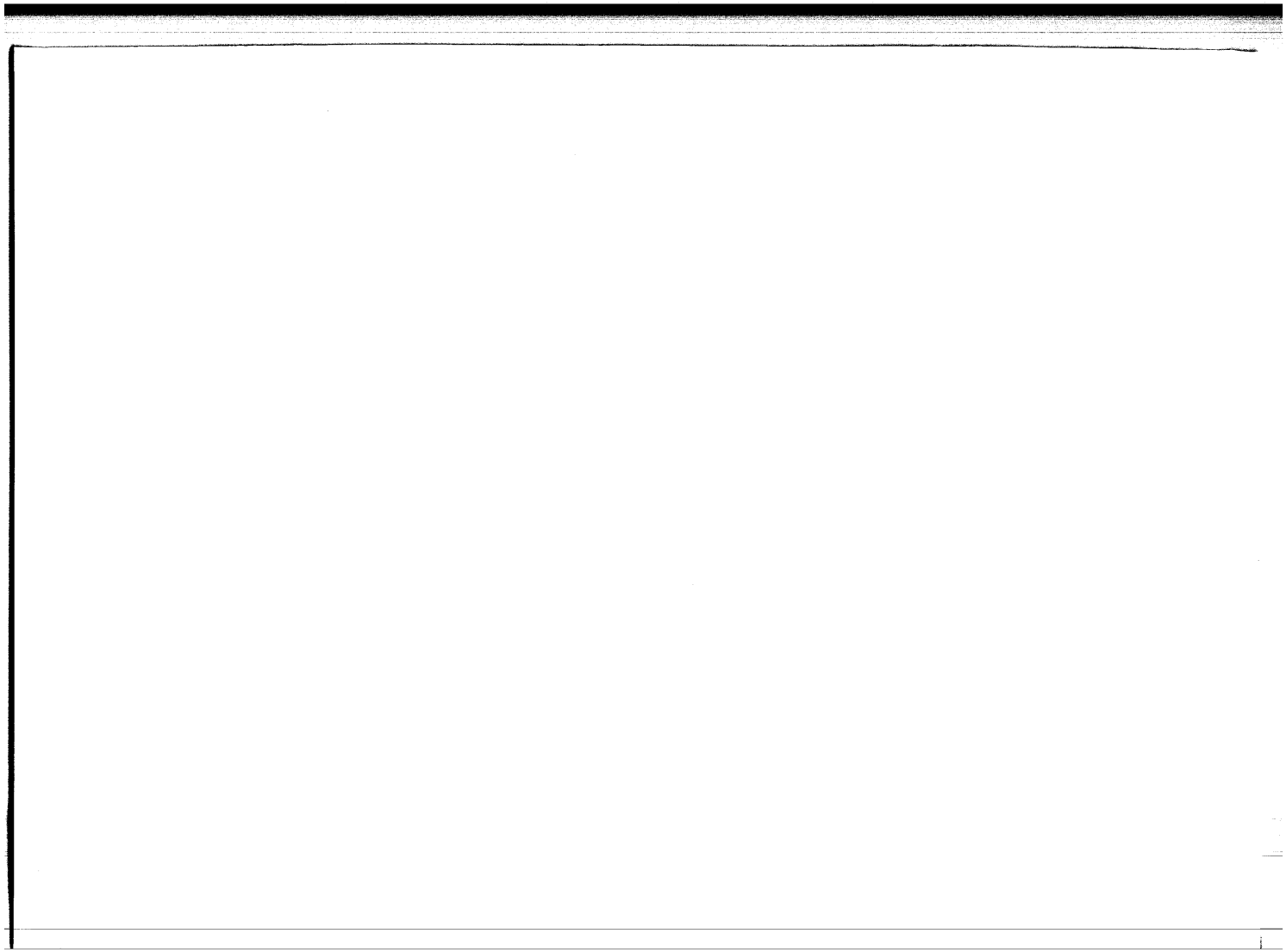
This adjustment is very heavily moderated by the kind of speaker system through which the device is played. Remember, the goal is NOT high-fidelity reproduction. Rather, it is to provide an acoustical radiation pattern similar to that of a human speaker. For this purpose, a small speaker and amplifier may work quite well (such as that associated with a television set), depending of course on the quality of the unit.

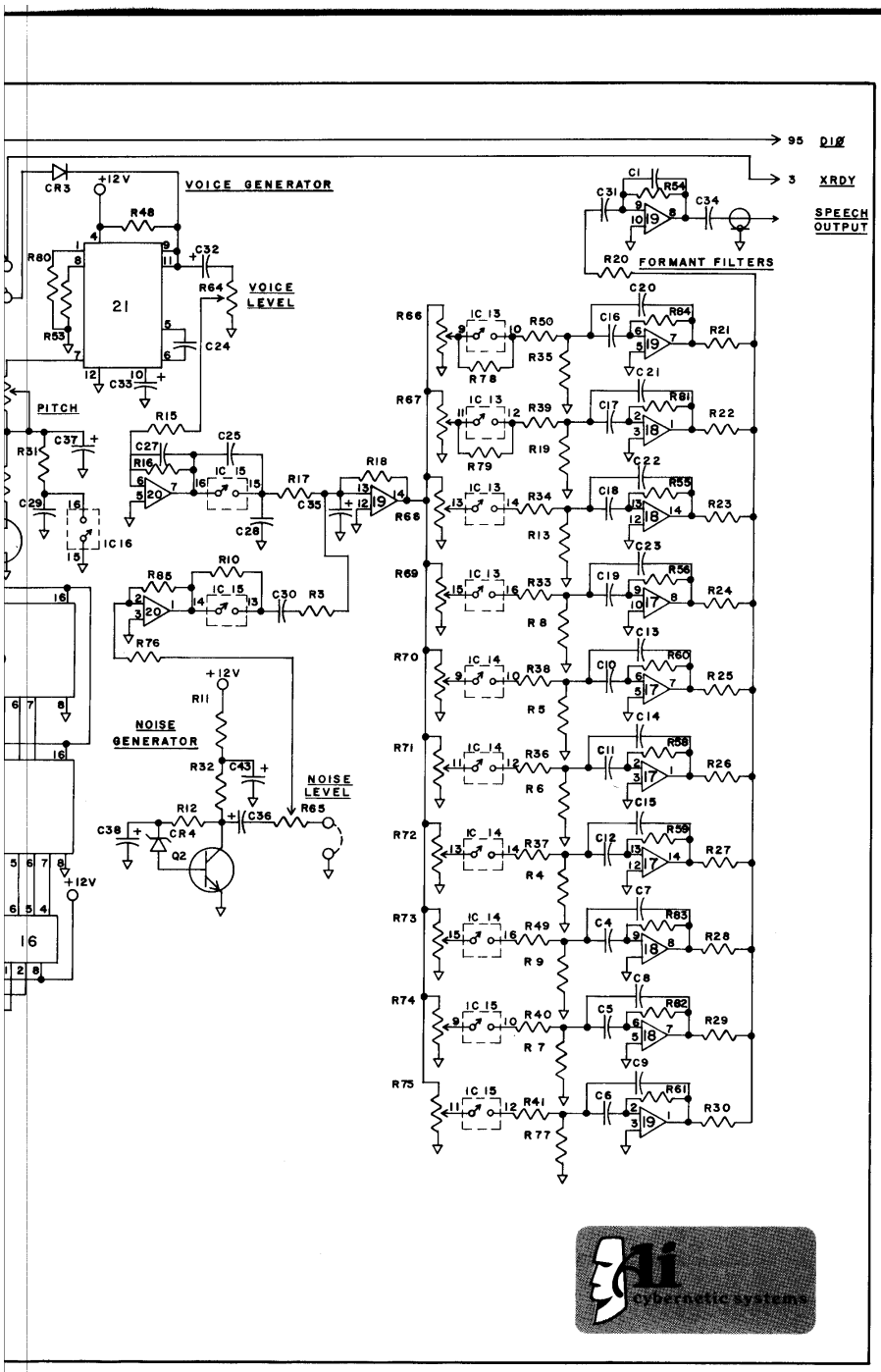
The output of the Model 1000 has been set to about 0.6-0.8V p-p (95% of which is voiced sound, 5% unvoiced (noise)). Should this level overdrive the following amplifier, it can be reduced through first adjusting the voice level control and then the noise level as described on the previous page.

Please feel free to call Ai Cybernetic Systems should you have particular questions or problems concerning the adjustment of your unit. A complete readjustment service is provided by the factory for a nominal charge.

SCHEMATIC OF THE MODEL 1000


ADDRESS SELECT
A7 83
A6 82
A5 29
A4 30
A3 31
A2 81
A1 80
AØ 79
SOUT 45
SINP 46
DOØ 36
DO1 35
DO2 88
DO3 89
DO4 38
DO5 39
DO6 40
POC 99
-18V 52
GND 50, 100
+8V 1, 51
+18V 2
SPEECH RATE
PHONEME TIMING
PHONEME PROGRAMMING
ANALOG SWITCHES
+12V
+5V
-12V
R63
R52
R51
R57
R14
R42-47
C26
C2
C3
C39
C40
C41
C42
R1
R2
CR1
CR2
TO PIN 11 IC 17-20
TO PIN 16 IC 1-3
TO PIN 14 IC 4,5,7
TO PIN 4 IC 17-20
TO PIN 8 IC 1-3
TO PIN 7 IC 4,5,7




95 DIØ
3 XRDY
SPEECH OUTPUT
VOICE GENERATOR
+12V
CR3
R48
R80
R53
21
C32
R64
VOICE LEVEL
C24
C33
PITCH
R51
C37
C29
IC16
R15
C27
R16
C25
IC 15
R17
C35
C28
R18
R66
IC 13
R10
R85
C30
R3
R76
NOISE GENERATOR
R11
R32
C43
NOISE LEVEL
R12
C38
CR4
Q2
C36
R65
+12V
16
C1
C31
R54
C34
R20
FORMANT FILTERS
C20
R84
C16
R21
C21
R81
C17
R22
C22
R55
C18
R23
C23
R56
C19
R24
C13
R60
C10
R25
C14
R58
C11
R26
C15
R59
C12
R27
C7
R83
C4
R28
C8
R82
C5
R29
C9
R61
C6
R30
R50
R35
R78
R39
R19
R79
R34
R13
R33
R8
R38
R5
R36
R6
R37
R4
R49
R9
R40
R7
R41
R77
R67
R68
R69
R70
R71
R72
R73
R74
R75
IC 14
cybernetic systems

PARTIAL COMPONENT LIST *

Integrated Circuits

IC1,2,3	8T97 INTERFACE
IC4	SN7430 8-INPUT NAND
IC5	SN7400 QUAD 2-INPUT NAND
IC6	XR 2240 TIMER
IC7	SN7404 HEX INVERTER
IC8	8212 BI-DIRECTIONAL PORT
IC9,10,11,12	74S288 PROM
IC13,14,15,16	AD7510K ANALOG SWITCHES
IC17,18,19,20	836 OPERATIONAL AMPLIFIERS
IC21	XR 2206 FUNCTION GENERATOR
IC22	7805 +5V REGULATOR

Transistors

Q1	2N4124 NPN (HIGH BETA)
Q2	HEP1036 P-CHANNEL FET

Diodes

CR1,2	12V, 1W ZENER
CR3	1N4154 SIGNAL DIODE
CR4	ZO211 5.1V, 0.5W ZENER

* Resistor and capacitor values are not listed because a great many of the values are determined during factory tuning. Because the values are critical, we have found that variations between manufacturers demand factory-selected values. All components are clearly marked on each board as to value.

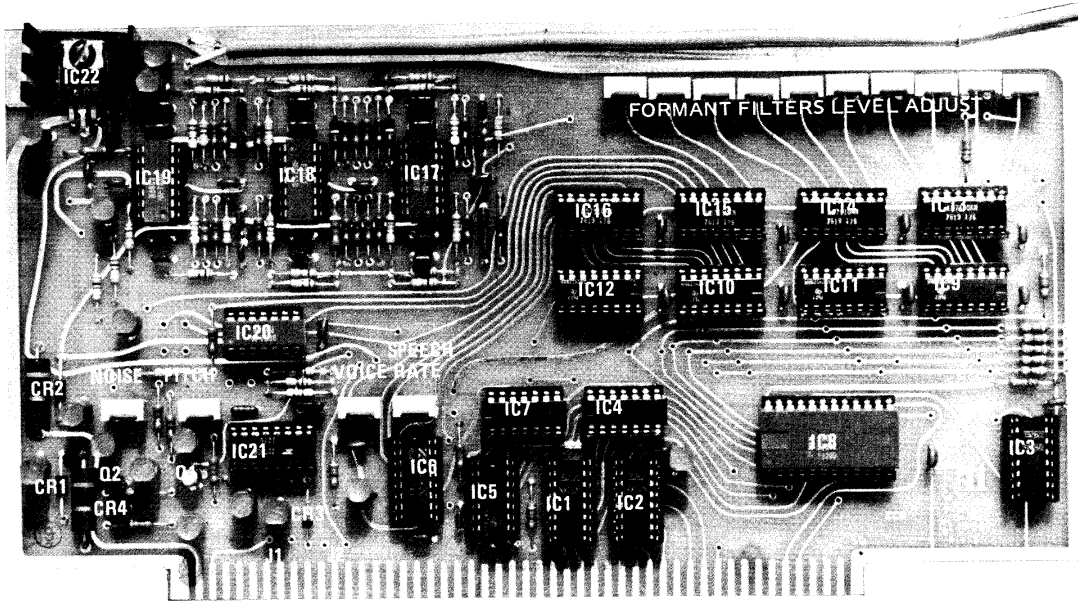


FIG. 4-2, COMPONENT LOCATIONS

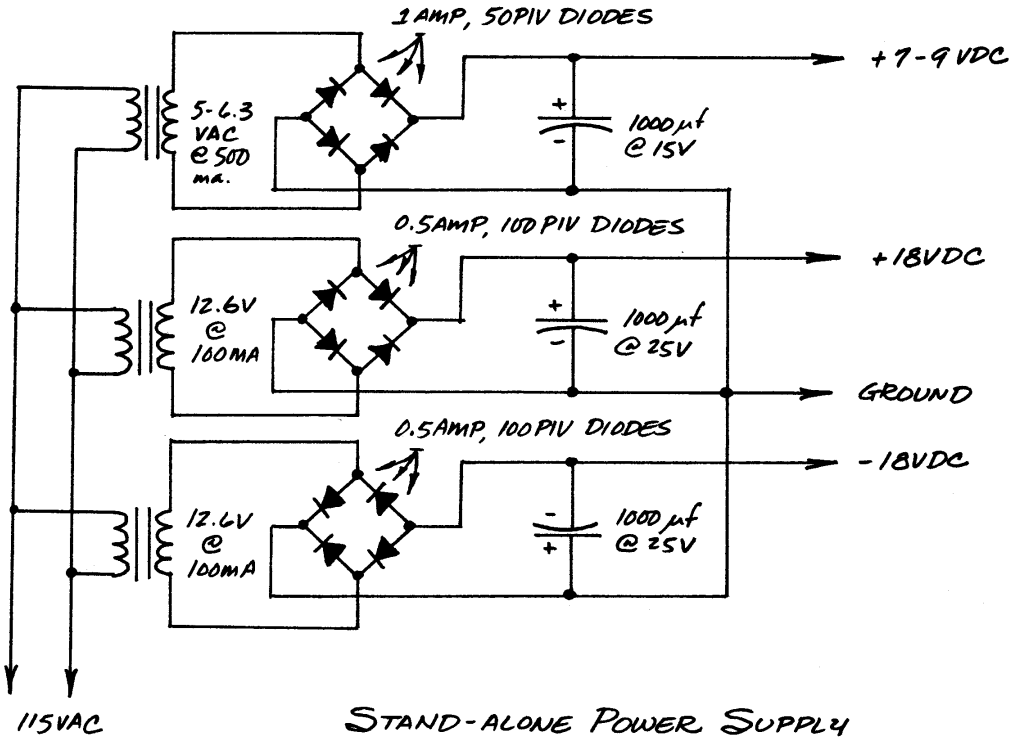
INTERFACING

The Model 1000 was designed to be electrically and mechanically compatible with the Altair (TM) bus structure. But there is little difficulty in wiring it to work with any parallel output device.

Power to the board need not be well-regulated, although ripple should not exceed 10% of the desired voltage. The power supply shown on the next page, or something very much like it, will work quite well in providing power. No component types are given because of the plethora of available equivalent components.

To transfer an ASCII character to the Model 1000, the character must be first set up on the D00-D06 bus lines. The lines D00-D05 carry the ASCII character with its two upper bits masked off. D06 contains pitch information. A "one" (+5V) lowers the pitch. D07 is not used. (Edge connector numbers for these lines are shown on the schematic, Section IV). Once the ASCII character is in place and settled -- about 1 uS, a duration dependent on the addressing interface -- the ASCII character can be transferred to the Model 1000 by setting the address to 254 (decimal) and by bringing SOUT high. If the application is considerably removed from a bus structure configuration, perhaps the most efficient use of the lines might come from tying all but one of the A1-A7 lines high and grounding the A0 line. Then, if the remaining floating address line is connected to SOUT, data will be transferred from the bus each time SOUT/AX goes high. Only one transfer takes place per upward transition as the address detecting circuitry is followed by a one-shot.

The phoneme clock signals that the phoneme has been completed



STAND-ALONE POWER SUPPLY

via the DIO bus line (or optionally, XRDY). To interrogate the DIO bus line, the SINP line is made to go high for the duration of the interrogation. The XRDY information appears without interrogation if jumper J1 is wired in place. This jumper is not normally installed at the factory.

To silence the unit, transferring an all zero field to the device as a normal character will suffice. It is also possible to clear the 8212 bi-directional data latch (IC8) by pulling $\overline{\text{POC}}$ ($\overline{\text{Power-On Clear}}$) low. If the $\overline{\text{POC}}$ line is not used, then it must be wired high to allow data transfer into the 8212.

The output signal is a 0.6V p-p medium impedance (about 1K) signal. As in any normal audio signal, clipping and distortion due to overdriving a following amplifier must be avoided (see Section IV, Adjustments).

PROGRAMMING EXAMPLE NO. 1

LUNAR LANDER

As discussed in Section III, several techniques exist for interfacing the Model 1000 to existing software. These following programs utilize the most elaborate method, which of course, provides the most satisfactory results. Two subroutines are used to sense the device-busy flag and output the phonetic characters to the device one at a time. A machine language program (lines 100-190 in Lunar Lander) is loaded by the BASIC program above the top of allocated memory (set at 12200) to check device status. Machine language is used preferentially because of its speed. A BASIC subroutine (lines 2000-3050) is used to separate the phonetic characters from the outputted word string and set the pitch bit (bit no. 6, decimal 64). The BASIC subroutine links to the machine language subroutine through the USR command. To have the computer speak, two strings must be set up (for example, lines 1545 and 1546). One string will contain phonetic characters; the other, pitch information.

This program plays the Lunar Lander game with the computer telling you your present hieight from the surface of the Moon. The game has been programmed on an Altair 8800 (T.M.) computer equipped with 12K of memory.

```
0100 POKE 73,169
0110 POKE 74,47
0120 ST=12201
0130 LN=28
0140 DATA 33,177,47,229,42,4,0,233,219,254
0150 DATA 230,1,202,177,47,230,0,198
0151 DATA 20,214
0152 DATA 1,194,188,47,123,211,254,201
0160 FOR II=ST TO ST+LN-1
0170 READ WD
0180 POKE II,WD
0190 NEXT II
1098 DIM O$(29)
1099 DIM N$(29)
1100 N$(0)=".."
1110 N$(1)="W!!N "
1111 O$(1)="1111"
1120 N$(2)="TOUU "
1121 O$(2)="2211"
1130 N$(3)="T+#EE "
1131 O$(3)="22211"
1140 N$(4)="FOU## "
1141 O$(4)="11122"
1150 N$(5)="F&&IEV "
1151 O$(5)="222111"
1160 N$(6)="SSII..KZZ "
1170 N$(7)="S'''.VIN "
1171 O$(7)="22211111"
1180 N$(8)="AE..T "
1181 O$(8)="21111"
1190 N$(9)="N&IEN "
1191 O$(9)="22111"
1200 N$(10)="TI''N "
1201 O$(10)="222111"
1210 N$(11)="ELL'''VIIN "
1211 O$(11)="2222111111"
1220 N$(12)="TW''LL.V "
1221 O$(12)="222111111"
1230 N$(13)="++##..TIENN "
1231 O$(13)="22211111222"
1240 N$(14)="FO##..TIENN "
1241 O$(14)="22111111222"
1250 N$(15)="FIIFF..TIENN "
1251 O$(15) "222221111222"
1260 N$(16) "SSII..KZZ.TIENN "
```

```
1261 O$(16)="222221111111222"
1270 N$(17)="S'''.V'N..TIENN "
1271 O$(17)="2221111111122222"
1280 N$(18)="AE..TIENN "
1281 O$(18)="221111222"
1290 N$(19)="N&IEN..TIENN "
1291 O$(19)="222111111122"
1310 N$(21)=N$(10)
1320 N$(22)="T.W'EN..TEE "
1321 O$(22)="2221111222"
1330 N$(23)="++##..TEE "
1331 O$(23)="221111122"
1340 N$(24)="FFO##.TEE "
1341 O$(24)="22111222"
1350 N$(25)="FFIIFF..TEE "
1351 O$(25)="2211111122"
1360 N$(26)="SSII..KZ.TEE "
1361 O$(26)="222111111122 "
1370 N$(27)="S'''.V'N..TEE "
1371 O$(27)="2221111111222"
1380 N$(28)="AE..TEE "
1381 O$(28)="2111122"
1390 N$(29)="N&IEN..TEE "
1391 O$(29)="2221111122"
1400 ST=63
1410 SF=64
1420 ZO=0
1430 WN=1
1500 D$="                "
1520 B$=" --------------------------------"
1530 PRINT CHR$(12):PRINT:PRINT:PRINT D$+B$:PRINT:PRINT
1540 PRINT D$+"THIS IS THE SPACECRAFT COMPUTER"
1545 V$="T.'IISS  ''ZZ  T.&&!  S.PAEZ..KR&FF..T  K!M..PEU..T###  "
1546 P$="1111111222222222222111111222222222222222111111111111111"
1550 GOSUB2000:PRINT:PRINT:PRINT D$+B$
1560 PRINT:PRINT "READY";
1561 V$="..RR'''.TE "
1562 P$="1111111222"
1563 GOSUB 2000
1570 INPUT A$:Y$="YES":IF A$.LT..GT.Y$ THEN 1500
1999 GOTO 4100
2000 FOR II=1 TO LEN(V$)
2010 C$=MID$(V$,II,1)
2011 U$=MID$(P$,II,1)
2030 WD=ASC(C$) AND ST
```

```
2035 IF U$="1" THEN WD=WD+SF
2040 X=USR(WD)
2070 NEXT II
2080 OUT 254,0
3050 RETURN
4100 PRINT CHR$(12)
4130 L=0:V=1:A=120:M=32500:N=16500
4170 G=.001:Z=1.8
4210 GOSUB 8000:GOSUB 6000
4220 PRINT:PRINT:PRINT "DESIRED FUEL RATE (LBS/SEC)";:INPUT K
4225 T=10
4230 IF K.LT.0 THEN 4590
4235 IF K=0 THEN 4310
4240 IF K.LT.8 THEN 4260
4250 IF K.LT.=200 THEN 4310
4260 PRINT "NOT POSSIBLE, RE-ENTER RATE";
4270 INPUT K: GOTO 4230
4310 IF M-N-.001.LT.0 THEN 4420
4320 IF T.LT..001 THEN 4210
4330 S=T:IF N+S*K.LT.M THEN 4350
4340 S=(M-N)/K
4350 I0=1:GOTO 4900
4360 IF I.LT.=0 GOTO 4710
4370 IF V.LT.=0 THEN 4380
4375 IF J.LT.0 THEN 4810
4380 LET I0=1: GOTO 4600
4420 S=(-V+SQR(V*V+2*A*G))/G
4430 V=V+G*S
4440 L L+S
4510 PRINT CHR$(12):PRINT "ON THE MOON AT";L;"SECS"
4511 W=3600*V
4512 PRINT "IMPACT VELOCITY OF";W;"MPH"
4520 PRINT "FUEL LEFT";M-N;"LBS"
4530 IF W.GT.=1 THEN 4550
4535 V$="P##..F''..K  L&&NN.TENN          "
4536 P$="22221111111122221111222"
4540 PRINT "PERFECT LANDING. CONGRATULATIONS.":GOTO 4590
4550 IF W=.GT.10 THEN 4560
4551 V$="NN)..T  B&&'.T      "
4552 P$="221111122211111"
4555 PRINT "NOT BAD.  BUT NOT PERFECT YET.":GOTO 4590
4560 IF W.GT.=25 THEN4570
4561 V$="OU..KAE         ":P$="2111222"
4562 PRINT " A FAIR LANDING. NO CRAFT DAMAGE".: GOTO 4SQR90
4570 IF W.GT.=60 THEN 4580
```

```
4572 PRINT "CRAFT DAMAGE."
4573 V$="OU..KAE          "
4574 P$="2111222"
4575 GOTO 4590
4580 PRINT "SORRY. BUT THERE WERE NO SURVIVORS."
4585 V$="K.R&'/  L&&NN.TENN          "
4586 P$="22221111222211112222"
4590 GOSUB 2000:GOSUB 2000:GOSUB 2000:PRINT
4591 PRINT "TRY AGAIN";
4593 INPUT A$:Y$="YES":IF Y$=A$ THEN PRINT CHR$(12):GOTO4100
4594 PRINT:PRINT "COMPUTER OUT":GOTO 5800
4600 L=L+S
4610 T=T-S
4620 M=M-S*K
4630 A=I
4640 V=J
4650 IF I0=1 THEN 4310
4660 IF I0=3 THEN 4850
4710 IF S.LT..005 THEN 4510
4720 S=2*A/(V+SQR(V*V+2*A*(G-Z*K/M)))
4730 I0=2:GOTO 4900
4810 W=(1-M*G/(Z*K))/2
4820 S=M*V/(Z*K*(W+SQR(W*W+V/Z)))+0.05
4825 I0=3:GOTO 4900
4830 IF I.LT.=0 THEN 4710
4840 GOTO 4600
4850 IF J.GT.=0 THEN 4310
4860 IF V.LT.=0 THEN 4310
4870 GOTO 4810
4900 Q=S*K/M
4905 IF Q.LT.=0 THEN 5000
4910 J V+G*S+Z*(-Q*(1+Q*(1/2+Q*(1/3+Q*(1/4+Q*(1/5))))))
4920 I=Z*S*(Q*(1/2+Q*(1/6+Q*(1/12+Q*(1/20+Q*(1/30))))))
4925 I=I+A-G*S*S/2-V*S
4930 IF I0=1 THEN 4360
4940 IF I0=2 THEN 4600
4950 IF I0=3 THEN 4830
5000 J=V+G*S
5010 I=A-G*S*S/2-V*S
5020 GOTO 4930
5800 V$="K↓M..PEU..T###   (U...T "
5801 P$="22221111112222221111"
5802 GOSUB 2000:END
6000 AL=INT(A+0.1)
6005 PRINT:PRINT "          ";
```

```
6010 IF AL=0 THEN 6130
6015 PRINT AL;
6020 IF AL.LT.100 THEN 6030
6021 V$="W!!N  HH!N.TR''.T  ":P$="1111222221111111"
6022 GOSUB 2000:AL AL-100
6030 IF AL.LT.20 AND AL.GT.9 THEN V$=N$(AL):P$=O$(AL):GOSUB2000:GOTO6110
6035 IF AL.LT.10 THEN 6080
6040 FOR MM=3 TO 10
6050 IF AL.LT.MM*10 THEN V$=N$(MM+19):P$=O$(MM+19):GOSUB2000:GOTO6070
6060 NEXT MM
6070 AL=AL-(MM-1)*10
6080 FOR MM=0 TO 9
6090 IF AL=MM THEN V$=N$(MM):P$=O$(MM):GOSUB2000:GOTO 6110
6100 NEXT MM
6110 PRINT "MILES FROM SURFACE--"
6111 V$="M&I'LLLZ  FFRR!MM  SS##.FF!'ZS "
6112 P$="22211111111111111111222211111"
6115 GOSUB 2000
6120 GOTO 6200
6130 PRINT "LESS THAN ONE MILE FROM SURFACE"
6135 V$="L''SS T+&&NN  W!!N  M&I''LL FFR!M  SS##..F!!SS "
6136 P$="2211112222111222212222221112222212222222222111"
6140 GOSUB 2000
6200 RETURN
8000 PRINT CHR$(12):PRINT CHR$(12):PRINT
8001 PRINT D$+"   -LEM CONTROL BOARD-"
8005 PRINT
8010 PRINT "CLOCK TIME","ALTIMETER","VELOCITY","  FUEL"
8020 PRINT "(SECONDS)","(FEET)","(MPH)","(LBS. LEFT)"
8030 PRINT:PRINT INT(L+.5),A*5280,V*3600,M-N
8040 PRINT:PRINT "        ---------------------------------"
8500 RETURN
```

PROGRAMMING EXAMPLE NO. 2

A DEMONSTRATION ROUTINE

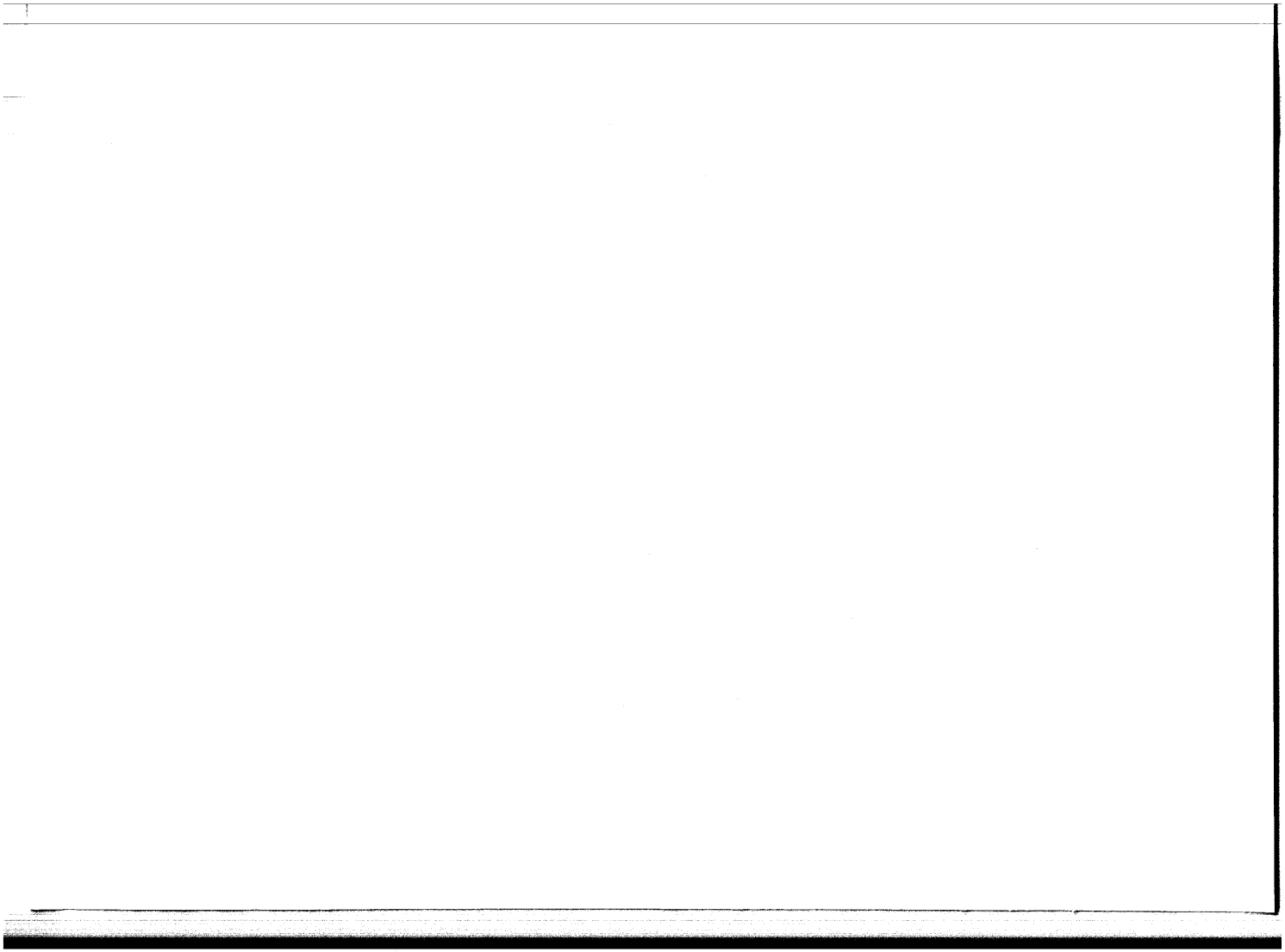
This simple program illustrates a useful basic technique in storing phonetic information, particularly where a large vocabulary might be desired to be rearranged to match the situation of the moment. As in Program No. 1, a 12K machine was used. Approximately 45 seconds of speech is produced by this demonstration program which simultaneously writes on a CRT and speaks the same words. The words are arranged in a word string array (W$) and the corresponding phonemes are in a phonetic string array (P$). No pitch information is provided for each word. Rather, an auto-pitch program (line 6082) is used to start each word at a high pitch and let it drift lower. The technique obviously does not require much programming room but it works surprisingly well.

```
0040 DIM W$(65)
0050 DIM P$(65)
0060 DIM A$(65)
0100 POKE 73,169
0110 POKE 74,47
0120 ST=12201
0130 LN=28
0140 DATA 33,177,47,229,42,4,0,233,219,254
0150 DATA 230,1,202,177,47,230,0,198
0151 DATA 20,214
0152 DATA 1,194,188,47,123,211,254,201
0160 FOR II ST TO ST+LN-1
0170 READ WD
0180 POKE II,WD
0190 NEXT II
2000 W$(1)="-------------------------------- GOOD"
2010 W$(2)="DAY"
2020 W$(3)="PEOPLE."
2030 W$(4)="I"
2040 W$(5)="AM"
2050 W$(6)="A"
2060 W$(7)="TALKING"
2070 W$(8)="ROBOT"
2080 W$(9)="MADE"
2090 W$(10)="FROM"
2100 W$(11)="AN"
2110 W$(12)="ALTAIR"
2120 W$(13)="-8800-"
2130 W$(14)="COMPUTER."
2140 W$(15)="I"
2150 W$(16)="WAS"
2160 W$(17)="BORN"
2170 W$(18)="IN"
2180 W$(19)="ALBUQUERQUE"
2190 W$(20)="AT"
2200 W$(21)="THE"
2210 W$(22)="-M."
2220 W$(23)="I."
2230 W$(24)="T."
2240 W$(25)="S.-"
2250 W$(26)="FACTORY"
2260 W$(27)="BUT"
2270 W$(28)="I"
2280 W$(29)="LEARNED"
2290 W$(30)="TO"
2300 W$(31)="TALK"
```

```
2310 W$(32)="IN"
2320 W$(33)="LAS"
2330 W$(34)="CRUCES,"
2340 W$(35)="NEW"
2350 W$(36)="MEXICO."
2360 W$(37)=" AS"
2370 W$(38)="YOU"
2380 W$(39)="HAVE"
2390 W$(40)="PROBABLY"
2400 W$(41)="NOTICED,"
2410 W$(42)="I"
2420 W$(43)="HAVE"
2430 W$(44)="A"
2440 W$(45)="STRONG"
2450 W$(46)="ACCENT."
2460 W$(47)=" BUT,"
2470 W$(48)="I"
2480 W$(49)="CAN"
2490 W$(50)="SAY"
2500 W$(51)="ANYTHING."
2510 W$(52)=" MY"
2520 W$(53)="SPEECH"
2530 W$(54)="IS"
2540 W$(55)="PROGRAMMED"
2550 W$(56)="FROM"
2560 W$(57)="A"
2570 W$(58)="STRING"
2580 W$(59)="OF"
2590 W$(60)="PHONETIC"
2600 W$(61)="CHARACTERS."
2610 W$(62)=""
2620 W$(63)="          "
2630 W$(64)="------------------------------"
4000 P$(1)="...G!U..T"
4010 P$(2)="DAE"
4020 P$(3)="PEE..P''LL"
4030 P$(4)="&&IE"
4040 P$(5)="AM"
4050 P$(6)="AE"
4060 P$(7)="T)..KENN"
4070 P$(8)="ROU..B)..T"
4080 P$(9)="MAE..T"
4090 P$(10)="FR!MM"
4100 P$(11)="AANN"
4110 P$(12)="&L.T''##"
4120 P$(13)="AE.TEE AE..T HH!N.TR''.T"
```

```
4120 P$(13)="AE.TEE AE..T HH!N.TR''.T"
4130 P$(14)="KK!M..PEU..T##"
4140 P$(15)="   &&IE"
4150 P$(16)="W!ZZ"
4160 P$(17)="BO#NN"
4170 P$(18)="IIN"
4180 P$(19)="&L.BU..K##..KEE"
4190 P$(20)="&&..T"
4200 P$(21)="T.&&!"
4210 P$(22)="''MM"
4220 P$(23)="&&IE"
4230 P$(24)="TEE"
4240 P$(25) "''SS"
4250 P$(26)="F&&.K.T##E "
4260 P$(27)=" B!..T"
4270 P$(28)="&&IE"
4280 P$(29)=".L''#N.T"
4290 P$(30)="TOU"
4300 P$(31)="T)..K"
4310 P$(32)="IIN"
4320 P$(33)="L(ZZZ"
4330 P$(34)="KRUSS'SS"
4340 P$(35)="..NEUU"
4350 P$(36)="M''.KZE..KOO"
4360 P$(37)="   &&ZZZ"
4370 P$(38)=".EUU"
4380 P$(39)="H&&VV"
4390 P$(40)="PR).B!.BLEE"
4400 P$(41)="NO.TIS..T"
4410 P$(42)="&&IE"
4420 P$(43)="H&&.VV"
4430 P$(44)="AE"
4440 P$(45)="ST.R)!NN"
4450 P$(46)="&..KS'N..T"
4460 P$(47)="  B!..T"
4470 P$(48)="..&&IE"
4480 P$(49)="KAAN"
4490 P$(50)="SAEE"
4500 P$(51)="ANEE+TIENN"
4510 P$(52)="   M&IE"
4520 P$(53)="SP.EE..T//"
4530 P$(54)="IIZZZ"
4540 P$(55)=".PRO.GR&MM.T"
4560 P$(56)=".FR!MM"
4570 P$(57)=".AE"
```

```
4580 P$(58)=".ST.RAEN-"
4590 P$(59)=".VV"
4600 P$(60) ".FO.N''..TI..K"
4610 P$(61)=".K&#&..K.T#ZZ"
4620 P$(62)="."
4630 P$(63) "."
4640 P$(64)="          "
5998 PRINT CHR$(12)
5999 SL=64
6000 FOR II=1 TO SL
6010 FOR JJ=1 TO LEN(W$(II))
6020 C$ MID$(W$(II),JJ,1):IF C$="" THEN PRINT:GOTO 6040
6030 PRINT C$;
6040 NEXT JJ
6050 PRINT " ";
6060 FOR JJ=1 TO LEN(P$(II))
6062 C$=MID$(P$(II),JJ,1)
6064 FOR KK=1 TO 5:DD=1:NEXT KK
6070 WD=ASC(C$)
6080 WD=63 AND WD
6082 IF JJ.GT.3 THEN WD=WD+64
6084 X=USR(WD)
6140 NEXT JJ
6150 OUT 254,64
6160 FOR KK=1 TO 15
6170 CT=12
6180 NEXT KK
7000 NEXT II
8000 GOTO 5998
9999 END
```



PROGRAMMING EXAMPLE NO. 3

SPEECH TRIAL PROGRAM

This program has been written to evaluate phonetic string representations of words and phrases. Two strings are to be inputted; A$, which contains the phonetic characters, and P$, which contains the pitch information.

Once the information has been inputted, the program cycles endlessly from line 20 to line 100 (line 95 is a time delay routine). Because this routine uses the same machine language subroutine (lines 110-200) as the preceeding two programs, phrases developed here can be immediately transferred to the other programs without modification.

SPEECH TRIAL PROGRAM

```
0001 PRINT CHR$(12)
0005 CLEAR 500
0006 WT=128
0007 SF=64
0008 ST=63
0009 GOTO 110
0010 INPUT A$
0012 PRINT "  ";:PRINT A$
0014 INPUT P$
0020 FOR II=1 TO LEN(A$)
0030 C$=MID$(A$,II,1)
0035 U$=MID$(P$,II,1)
0050 WD=ASC(C$) AND ST
0055 IF U$="1" THEN WD=WD+SF
0060 X=USR(WD)
0090 NEXT II
0092 OUT 254,0
0095 FOR II=1 TO 20:TD=1313131313131313:NEXT II
0100 GOTO 20
0110 POKE 73,169
0111 POKE 74,47
0120 SA=12201
0130 LN=28
0140 DATA 33,177,47,229,42,4,0,233,219,254
0150 DATA 230,1,202,177,47,230,0,198
0151 DATA 20,214
0152 DATA 1,194,188,47,123,211,254,201
0160 FOR II=SA TO SA+LN-1
0170 READ WD
0180 POKE II,WD
0190 NEXT II
0200 GOTO 10
```

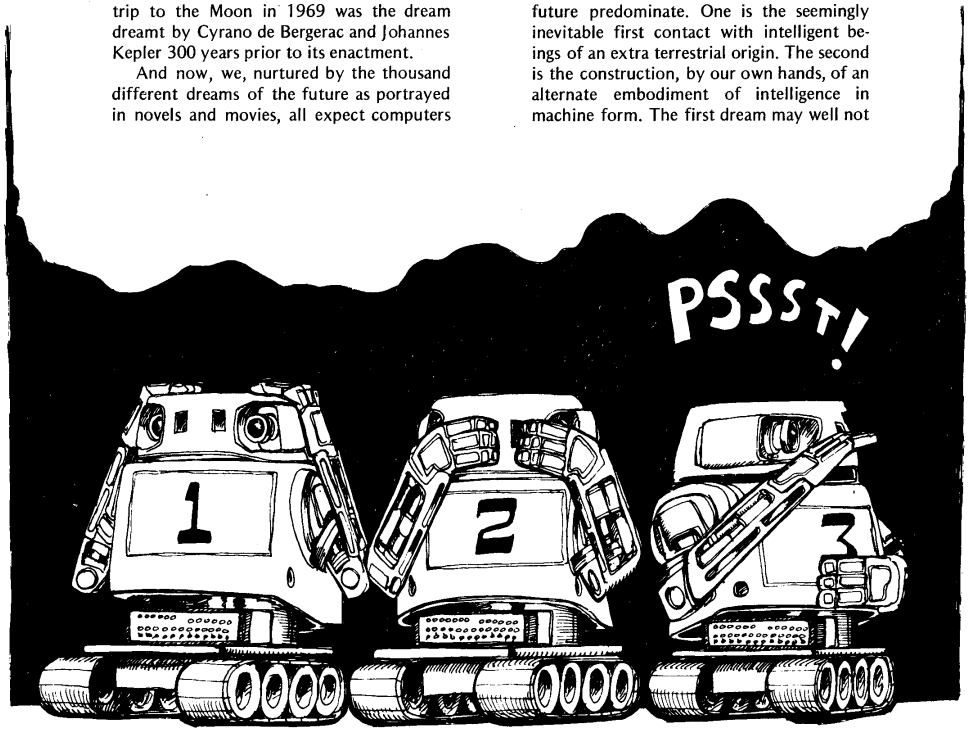
The Time Has Come to Talk

Wirt Atmar
Ai Cybernetic Systems
PO Box 4691
University Park NM 88003

The extent to which art and literature, particularly science fiction, affect the future course of civilization remains a persistent and perplexing question. Must a dream, by necessity, occur decades before its realization? Or does the presence of the dream itself generate its own reality? Mankind's trip to the Moon in 1969 was the dream dreamt by Cyrano de Bergerac and Johannes Kepler 300 years prior to its enactment.

And now, we, nurtured by the thousand different dreams of the future as portrayed in novels and movies, all expect computers to be able to talk in the near future. Whether we see the computer becoming the benign and obedient servant of man or wildly out of control, we all tend to see the computer becoming more anthropomorphic, more humanlike in behavior and form.

In science fiction two great dreams of the future predominate. One is the seemingly inevitable first contact with intelligent beings of an extra terrestrial origin. The second is the construction, by our own hands, of an alternate embodiment of intelligence in machine form. The first dream may well not



"The time has come," the Walrus said,
"To talk of many things:
Of shoes — and ships — and sealing wax —
Of cabbages — and kings —
And why the sea is boiling hot —
And whether pigs have wings."

— Lewis Carroll, 1871, in **Through the Looking-Glass.**

occur within the lifetime of our civilization; the second would seem to be almost guaranteed within the next 100 years.

The addition of speech to the computer's behavioral repertoire makes the computer no more intelligent nor aware than it was before. It remains a simple machine. But it undeniably takes on a human characteristic that it never possessed before. An observer finds it impossible not to personify the machine with an identity and a distinct personality. While the addition of speech is only a minor step toward achievement of a truly self-organizing, artifically intelligent machine, it is a psychologically important one. The computer, once it speaks, *seems* to be intelligent. But again, the dream of machine produced speech is much older than its reality. The ancient Greco-Roman civilization was fascinated with the idea of *deus ex machina.* Stone gods were often hollowed to allow a priest to speak from within, a practice that persisted well into the Christian era.

The first known practical realization of machine generated speech was accomplished in 1791 by a most ingenious engineer, Wolfgang von Kempelen, of the Hungarian government. Von Kempelen's machine was based on a surprisingly detailed understanding of the mechanisms of human speech production, but he was not taken seriously by his peers due to a previous well publicized deception in which he built a nearly unbeatable chess playing automaton. The "automaton" was unfortunately later discovered to actually conceal a legless Polish army ex-commander who was a master chess player.

By 1820, a machine was constructed which could carry on a normal conversation when operated by an exceptionally skilled person. Built by Joseph Faber, a Viennese professor, the machine was demonstrated in London where it sang "God Save the Queen." Both the Von Kempelen and Faber machines were mechanical analogs of the human vocal tract. A bellows was provided to simulate the action of lungs; reeds were used to simulate the vocal cords, and variable resonant cavities served to simulate the mouth and nasal passages.

The basic method, modelling the human vocal tract, remains to this time the only practical method of actually synthesizing speech. In the 20th century, such modelling is done electronically. The approach was first put in electrical analog form by Bell Laboratories in the late 1930s. The Bell Telephone VODER (Voice Operation DEmonstratoR) was initially shown at the 1939 New York's World Fair where it drew large crowds and considerable attention. The VODER consisted of a buzz source (similar to human vocal cords or mechanical synthesizers), a hiss source to simulate the rush of aspirated air, and a series of frequency filters to imitate the three, four, five or six preferred frequencies (called formant frequencies) passed by the resonant cavities formed by the mouth, tongue and nose.

The original VODER was played by highly trained operators using a keyboard, wrist switches, and pedals at an organ-like console. Twenty four telephone operators were trained six hours a day over a 12 month period for the 1939 World's Fair. The VODER itself was a full rack in height.

With the advent of digital computers, however, the synthesis of speech has been made much easier. All the information necessary to repeatedly and reliably generate any one speech sound (a "phoneme") can now be programmed into the machine. Through the proper connection of phonemes, a digital computer could be made to say words and sentences.

General American English, the dialect spoken in the midwest and southwestern parts of the United States, contains 38 distinct phonemes. These speech sounds can be divided into the following classes:

Pure vowels: produced by a constant excitation of the larynx and the mouth held in a steady position; eg: "ē".

Diphthongs: a transition from one

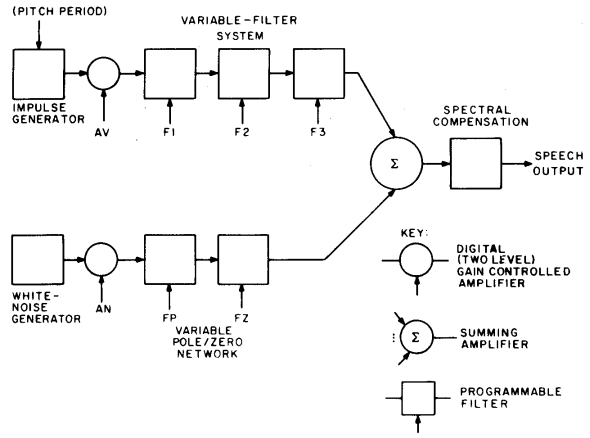


Figure 1: The serial analog speech synthesizer in block diagram form.

pure vowel to another, thus are not always considered as separate phonemes; "ī", "ū".

Fricatives: consonants produced by a rush of aspirated air through the vocal passages: "f", "s".

Plosives: explosive bursts of air: "p", "k", "t".

Semi-vowels: "w", "y".

Laterals: "l", "r".

Nasals: "n", "m".

To produce speech, a separate circuit, or combination of circuits, must be provided to generate each of the above classes of phonemes.

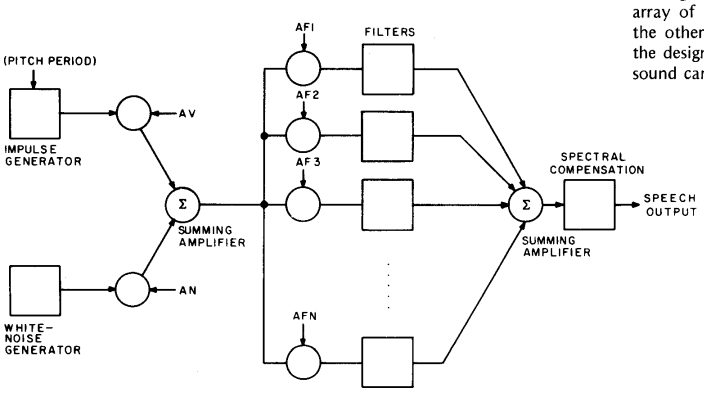


Figure 2: The parallel analog speech synthesizer in block diagram form.

Among possible realizations of such a synthesizer, there are the serial analog and parallel analog forms. Figure 1 illustrates a block diagram of a serial analog design, and figure 2 shows the general organization of a parallel analog synthesizer.

The parallel analog method was the realization chosen by Ai Cybernetic Systems for its synthesizer module. The parallel realization was chosen because of the low digital information transfer rate and the smaller number of bits required to control the filters which simulate the resonant cavity of the vocal tract.

In the Ai Cybernetic Systems design, the rush of aspirated air is generated by the noise of a zener diode operated at its knee, amplified many times, as shown in figure 3. The action of the larynx is simulated by an integrated circuit function generator. One or both of these circuits is selected to produce the excitation necessary to generate any one class of phonemes. The actual phoneme perceived is determined by the duration of the excitation and the selected formant filters. Figure 4 shows the typical formant filter circuits which are digitally activated by analog switches.

The control of the several analog switches is provided by a read only memory which is addressed by the ASCII bit patterns identified in table 1.

No hard and fast rules exist in the design of the circuitry to generate a phoneme. In fact, small changes in component values can often make large differences in the phoneme which is actually heard. Because no set rules exist, a steady stream of listeners must parade before the machine while it is being designed in order to determine which phoneme the synthesizer is really saying. The phenomenon of "tired ears" rapidly sets in; and a person will begin, after a bit, hearing any one speech sound as a whole array of possible phonemes. Suggestion, on the other hand, is an ever obtuse enemy to the designer. Surprisingly, almost any speech sound can be suggested to sound like a great

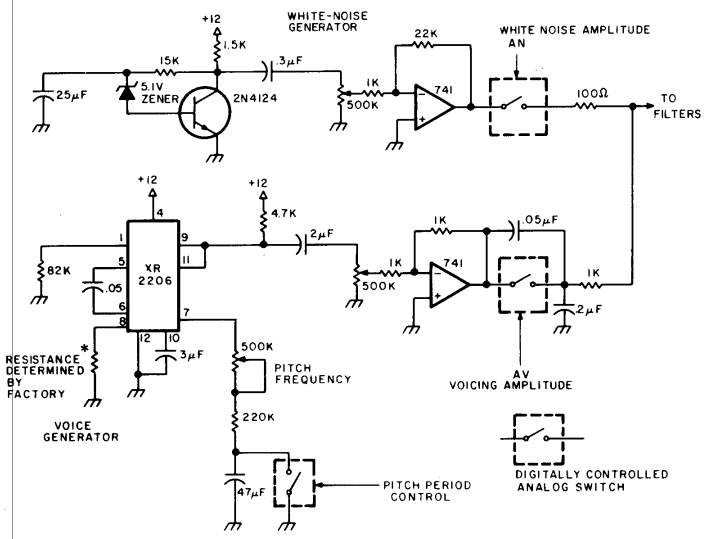


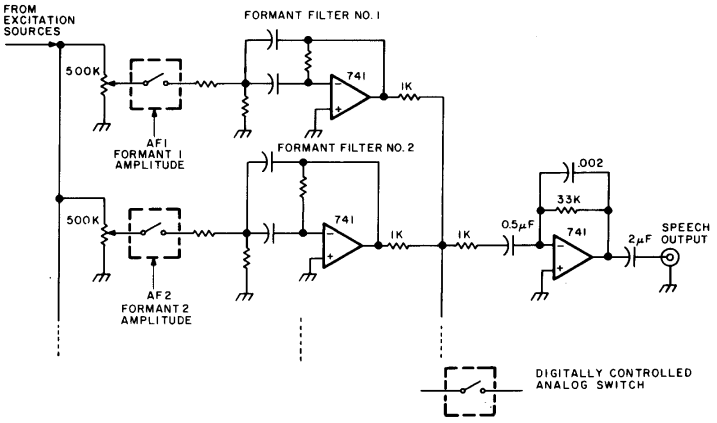
Figure 3: The excitation sources of the Ai Cybernetic Systems Model 1000 Speech Synthesizer. The rush of air through the vocal passages is simulated in the upper branch while the action of the larynx is simulated in the lower branch.

number of alternate phonemes, especially after 20 to 30 minutes of intense listening.

Once the design is experimentally determined, careful procedures must be followed to insure that when the circuit is duplicated, it produces each phoneme properly. This means precision components must be used, as small changes in values can make the difference between moderately distinct speech and a fairly mushy speech.

Analog simulation of the vocal tract is the only method of true speech synthesis known. A popular alternate method of speech production (actually, reproduction) is the storage of digitized speech in a ROM. When the stored information is clocked out of the ROM at the proper rate and smoothed by a low pass filter, the generated speech can be quite clear and distinct. But it is important to note that this is not synthesized speech. In effect, this method is no different than any other method of recording speech. Yet, the method does have the advantage of producing readily understood words by a

Figure 4: The parallel filter network of the Model 1000. The filter frequencies and quality factors chosen depend on the number of filters used to divide the voice frequency spectrum. Ideally, the center frequencies of the filters should lie somewhere near the commonly occurring formant frequencies.



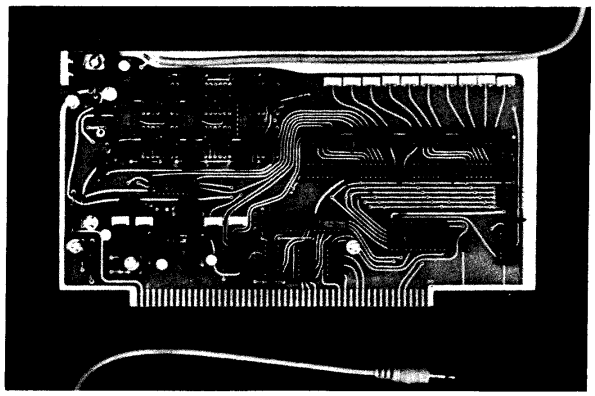


Photo 1: The Ai Cybernetic Systems Model 1000 Speech Synthesizer. The synthesizer is primarily an analog circuit controlled digitally. Ten active filters composed of 15 operational amplifiers are mounted in the upper left corner of the board. Directly beneath these resonant-cavity simulating filters are the vocal excitation circuits. The right half of the board is composed of the ASCII character decoding circuits and phoneme memories. Four 32 x 8 ROMs control the 16 analog switches to select the proper combination of circuits to generate any one phoneme. A device-busy flag is returned for the duration of the phoneme.

computer or calculator. However, the vocabulary is totally predefined and must remain small due to the high cost of storing this kind of generated speech. Moreover, the repertoire of this kind of speech is limited to the person who initially spoke the recorded words.

Synthetic speech, on the other hand, is generally not as clear and distinct. The proper transitions from phoneme to phoneme, the automatic emphasis given to leading or terminating consonants, and the intonation of a rhythm in speech which is associated with a word's importance or placement, are all facets of human speech which are difficult to properly recreate in machine produced speech. The determination of accurate rules to account for these factors has been the subject of active and intense research at centers here, and in Europe and Japan, including Bell Telephone Laboratories, the Haskins Laboratories of New York, the Royal Institute of Technology in Sweden, and the Musashino Electrical Communication Laboratory in Tokyo. On the whole, totally satisfactory rules have not yet been worked out although a great deal of progress has been made in the last 20 years. Machines which do incorporate the known rules quickly become elaborate and expensive (in the tens of thousands of dollars).

Simplified speech rules can be incorporated in a much smaller machine, but the burden of intelligibility now falls upon the listener. The produced speech is not natural speech. It sounds for all the world like the speech produced by the robots of 1950s grade B science fiction movies. But it is intelligible and it is quickly learned. Because the machine pronounces every phoneme in the same fashion each time it occurs, a listener quickly gains a feeling for the speech. The process is not unlike learning to listen to a newly-arrived foreigner who possesses a strong accent. The fashion by which he mispronounces the English phonemes is quickly learned and intelligibility increases rapidly. The difference with synthetic speech is that the speech is truly an alien form of speech, not often heard before by many of us.

As to the naturalness of synthetic speech, M D McIlroy of Bell Telephone Labs wrote this in 1974 *[in "Synthetic English Speech by Rule," Computer Science Technical Report No. 14, Bell Telephone Laboratories]:*

> The Computer Science Center at this laboratory has experimented with an inexpensive speech synthesizer [presumed to be the Votrax] as a regular output device in a general purpose computing system. Our intention was not to do speech research or to create artificial speech as an end in itself. In the present state of the art, those goals require much more elaborate facilities than we have at our disposal.
>
> We wished to see what uses might evolve when speech became available more or less on a par with printed output. For this goal, "naturalness" was not a prerequisite, any more than it is for printed output. Most computers still print mainly in upper case, are incapable of printing mathematical notation, and normally produce cryptic codes or tabular stuff that require considerable indulgence to be understood. Since printed gobbledygook is so widely accepted from computers – and fed into them, witness any manufacturer's operating system manual – we suspected that spoken gobbledygook might be quite passable, too, except for one severe difficulty: Being ephemeral, sounds must be understood at first hearing. As it turns out, long speeches *are* hard to understand, as are extremely short utterances of very simple words out of context. But given a little familiarity

with the machine's "accent", one finds short sentences to be quite intelligible.

The phonemes generated by the Model 1000 synthesizer appear in table 1. Each phoneme has been assigned an ASCII character to represent its particular sound. The assignment was done in the most intuitive manner possible; the consonants are generally the consonants as they appear on the keyboard, but there are many more vowels than a, e, i, o and u. Non-alphanumeric characters were chosen to represent the remaining vowels and consonants in such a manner that they could be easily associated with their sound. As examples of this, the number symbol, "#" is used to signify the vowel *er* as in num*ber*, "&" for the vowel *ae* as in *a*nd "(" for *ah* and ")" for *aw*

Table 1: List of Phonemes.

	Phoneme	ASCII Symbol	Usage
Vowels:			
	a	A	pace, bay
	ae	&	and, Altair
	ah	(	father, all
	aw	)	bought, robot
	e	E	see, harmony
	eh	'	excessive, ten
	er	#	number, bird
	i	I	hit, six
	o	O	Mexico, over
	oo	U	too, sue
	uh	!	the, computer
	^	↑	putt, up
Semi-Vowels:			
	w	W	water, wind
	y	Y	yaw, yacht
Plosives:			
	p	P	pop, deep
	k	K	computer, Atlantic
	t	T	top, pot
	b	B	boy, bird
	d	D	dog, died
	g	G	go, great
Fricatives:			
	f	F	puff, food
	h	H	how, had
	s	S	saw, miss
	v	V	David, vow
	sh	/	slash, shoot
	th	+	thaw, Earth
	z	Z	zero, is
Liquids:			
	1	L	low, all
	r	R	row, round
Nasals:			
	m	M	miss, am
	n	N	now, nine
Others:			
	Glottal Stop	.	The pause associated with aspiration
	Draw Bar	-	An extended vowel with decay
	Pause	(space)	Normal word spacing

representing the position of the tongue when these vowels are spoken, "!" for the sharp sound of *uh*, "+" for the fricative consonant *th* as in *th*aw, and "/" for the *sh* in sla*sh*.

The Model 1000 accepts a string of ASCII characters as if it were a normal printing device. Read only memories on the board convert the incoming ASCII symbol into specific control information which in turn determines the vocal source, duration and frequency content of the spoken phoneme. Less than 50 bytes of machine code or 8 lines of the typical BASIC are all that is required to generate a subroutine to accept a string of characters and output it character-by-character to the synthesizer.

For example, to write the phrase "I am a talking robot" on a printer or display peripheral, an ASCII character string is set up and sent to the output device. In BASIC, if C$ is the argument of the output subroutine, the setup would be:

C$ = "I AM A TALKING ROBOT."

To have the synthesizer say the same phrase, the setup for the phonetic output routine with argument P$ might be:

P$ = "&IE AM AE T). .KEN- RO.B). .T"

(The ASCII symbols are taken from table 1.) The long vowels I and A occur in this passage. As a rule, most of the long vowels are not really vowels at all but rather diphthongs composed of a sequence of pure vowels. Pronounce out loud each of the phonemes in the phrase above, referring to table 1 as necessary. Remember that each phoneme has only one specific sound. Playing the part of a synthesizer yourself, you will find that you can say any English word with the phonemes of table 1.

Programming the Model 1000 synthesizer is easy once you actually begin to listen to what you say and learn to rely less on how a word is written. English is a hodge podge of languages and carries with it all the alternate symbolisms of the pronunciations of its root languages. Purely phonetic languages such as the Polynesian languages of Samoa or Tonga could be made to be spoken almost as they are written. This is unfortunately not true of English; homonyms such as "won" and "one" and "two", "too" and "to" abound.

Generally, only one phonetic spelling exists for any one word regardless of the number of alternate written spellings. It becomes important to identify the sounds that you actually are saying when a word is pronounced. The word "one" is phoneticized using the phonemes of table 1 as W!N in similarity to the word "won"; "two" is programmed as TOU- more as if it were the

written word "too". For most Americans, there is no difference in the way these words are pronounced.

Proceeding in the same fashion, the remaining numbers up to ten are typed in as:

T+#E- FO#- F&IE..V SI..KZ
S'-VIN AE..T N&IEN T'N

Again, pronounce these phonetic spellings to yourself. As you will discover, phonetic spellings are quickly deduced and learned.

In a very short period of time, it becomes possible to make the machine say anything. At that point, conversational computing takes on a whole new meaning. Interactive computing will never again be the same once your computer has actually spoken to you.■

BIBLIOGRAPHY

1. *Speech Synthesis, Benchmark Papers In Acoustics,* 1973. J L Flanagan and L R Rabiner, eds. Dowden, Hutchinson and Ross, Stroudsburg PA. A collection of the best papers on speech synthesis over the past 35 years.
2. "Synthetic Voices for Computers," 1970. J L Flanagan, C H Coker, L R Rabiner, R W Schater, N Umeda in IEEE *Spectrum* 7:22-45. An authoritative overview of the speech synthesis procedure.
3. "The Synthesis of Speech," 1972. J L Flanagan, *Scientific American* 226:48-58. A simplified rework of the IEEE *Spectrum* article above.
4. *IEEE 1974 Speech Recognition,* Proceedings, 1974. L Erman, ed. IEEE, NY. A bit too technical for a first introduction but a good measure of where things are going.

COMMERCIAL PRODUCTS

At the present time, two speech synthesizers are both commercially available and affordable by the hobbyist. One is the Votrax produced by:

Vocal Interface Division
Federal Screw Works
500 Stephenson Dr
Troy MI 48084
Price, approximately $2,000
Interfacing: Parallel or Serial (RS-232)

The second is the Model 1000 manufactured by:

Ai Cybernetic Systems
PO Box 4691
University Park NM 88003
Price, $425
Interfacing: Electrically and mechanically compatible with Altair/IMSAI/ Poly-88 bus structure.

Either company will be pleased to provide literature free of charge. A demonstration tape is available from Ai Cybernetic Systems for $5 and a complete programming guide, theory of operation manual and phonetic glossary is available for $2.50.