CT-1
Hardware User's Manual

Computalker Consultants
P.O. Box 1951
Santa Monica, California 90406

SYNTHESIZING SPEECH BY RULE WITH THE COMPUTALKER MODEL CT-1

Synthesis-by-Rule is a method of producing synthetic speech which is considerably easier than computer/hand analysis of recorded human speech. The word or phrase to be synthesized is entered in the form of a phonetic code to a software system which generates the control parameters for the CT-1 Synthesizer board. The result is speech which is understandable to most people in all but the most difficult perceptual situations with high noise levels or speech material having completely unexpected content.

The demonstration cassette contains a portion of the Gettysburg Address synthesized using a system of software rules. Such a set of software acoustic-phonetic rules is available from Computalker Consultants coded for the 8080 CPU. This software system accepts a string of ASCII coded phonetic symbols with stresses marked, and produces a set of control parameters for the Model CT-1 Synthesizer. The example on the cassette was generated using a previous version of this software system, coded in FORTRAN, and running on a DEC PDP-12. As the parameter data was generated, it was punched on paper tape in the data format as described in the CT-1 Hardware User's Manual, and then read into the IMSAI 8080 for playback. That program, as run on the larger machine, was originally written for a different speech synthesizer, and some parameters required special treatment for conversion to the CT-1 parameter format. In some cases, this conversion was not accurately fine-tuned for the CT-1, and the direct output of the 8080 version of the program is somewhat clearer in some of the fine details.

The CSR1 Synthesis-by-Rule software system is organized around the philosophy of attempting to produce natural sounding, human quality speech, rather than trying to produce a stereotypical robot-like sound. Because the true structure of real human speech is not yet correctly represented in the software rules, the resulting speech sometimes has an eerie quality that makes the listener try to assign human-like traits and qualities to the "speaker" behind the voice. This psychological reaction to the voice does not occur when it is synthesized in a "robot" stereotype having little or no pitch variation and abrupt, blocky formant frequency transitions. The pitch control parameter (FØ) can easily be held to a constant value if the speech output sounds better to you that way. The CSR1 software system is structured around phonological, phonetic and acoustic principles in such a way that it can be modified to keep pace with the state of the art of synthesis of natural speech. The Model CT-1 has been designed as a general acoustic synthesizer so that the hardware will not pose limitations to further improvements in the obtainable speech output quality.

The CSR1 software system is set up as a general callable subroutine which accepts a string argument containing the phonetic text, and on completion, plays the speech data in the buffer directly to the CT-1. With this structure, CSR1 may be called either from a keyboard input loop (supplied with the code) giving an on-line phonetic synthesizer, or from another system such as BASIC or an operating system, which passes a stored or computed string argument containing the material to be synthesized. On return, the buffer contains the actual CT-1 data as synthesized, which may be written out to cassette or paper tape for editing with the CTMON Monitor/Editor program. The 8080 assembly code version of CSR1 fits in less than 6K bytes of memory, including all phoneme feature and target tables. This code may be located in ROM or RAM. Additional RAM will be required for parameter data storage during the actual synthesis. The buffer space required is 900 bytes per second of speech. By comparison, the introductory phrase, "Hello, I'm Computalker, A speech synthesizer designed to plug into the standard bus on your 8080 microcomputer" is less than 7K bytes long. CSR1 version 1.0 completes the computation of parameter data before beginning playback. An interrupt driven version is currently under development, which will begin playback as soon as sufficient data has been computed and stored in the buffer.

HOW TO GET NATURAL SOUNDING SPEECH OUTPUT FROM THE COMPUTALKER MODEL CT-1

The demonstration cassette, "Sounds of Computalker", illustrates several methods of obtaining the control parameters to operate the Computalker Model CT-1 Speech Synthesizer. High quality speech output, as exemplified by the introductory phrases, "Hello, I'm Computalker. A speech synthesizer ... ", involves computer processing of recorded human speech followed by a fair amount of hand work. The recordings were initially digitized at 10K samples/second and then analysed using a linear prediction algorithm to extract the formant frequencies, and a cepstrum algorithm to measure the fundamental frequency. These techniques are described in several texts on speech analysis (Flanagan,J.L., Speech Analysis, Synthesis, and Perception, 2nd Ed., Springer Verlag 1972; Markel,J.D. and Gray,A.H.,Jr., Linear Prediction of Speech, Springer Verlag 1976). In addition to these analyses, the amplitude was measured by RMS averaging a smooth window each 10 msec. to obtain the AV parameter. Some editing of the formant frequency data was done by hand to eliminate falsely detected peaks and fill in occasional gaps in the true formant data before converting the frequency data to the Computalker parameters F1, F2, and F3. Since the CT-1 control parameters consist of numerical values within the range of 0—255, all frequency and amplitude data is converted so that it stays within this range. All the above steps required approximately 6 hours of time on a DEC PDP-12 set up for speech analysis processing to produce the original data for the introductory phrases on the cassette. At this stage, this data was punched on paper tape and then read into the CT-1 Control Monitor program running on my IMSAI 8080. From that point, I spent several more evenings entering the data for parameters AH, AF, FF, and AN, and a bit more touching up of the other parameters.

Given the frequency vs. time information obtained from the initial computer analysis, the remaining aspiration and frication data can be inserted by fairly straight-forward procedures. These procedures will be described in the completed CT-1 Hardware User's Manual. The Manual will also discuss the approximate formant frequency patterns needed to construct the sounds of the various phonemes of English. It would be feasible (although tedious work) to construct intelligible sounds by hand editing based on this data. However, it is still quite difficult to form these patterns to make natural sounding speech without access to a spectrum analysis process of some kind. Such an analysis gives you the frequency structure as a function of time, i.e. retaining the natural timing structure.

It is my plan to publish more extensive descriptions of the above mentioned speech analysis techniques, to make them accessible to a wider audience than they now have. The recent developments in floating point hardware with multiplication in the 50-100 microsec. range make it reasonable to do this sort of analysis on a microcomputer. The setup would require a filter and A/D converter capable of sampling the speech at at least 10K samples/sec. The low-pass speech filter ahead of the A/D converter should be reasonably flat to at least ⅓ of the sampling rate, and then down by at least 30-40 db at ½ the sampling rate. 32K of RAM memory would allow sampling up to 3 seconds continuously, which is a workable sized chunk. Without floating point hardware, the analysis would proceed quite slowly, but in many cases, that is not a drawback on a micro system.

Alternatively, for a modest consulting fee, Computalker Consultants could supply the basic, rough formant frequency, FØ and AV data from your tape recording, leaving out the aspiration, frication and nasal values, which must be added by hand. As a preliminary estimate, I believe this work could be done for approx. $25 per second of speech material to be analyzed. Working from this basic data, the desired speech could be produced following the tables and information given in the CT-1 Hardware User's Manual, using the CTMON Monitor/Editor to synthesize speech from the data as the work progresses.

Friends, Humans, and Countryrobots: Lend me your Ears

D Lloyd Rice
Computalker Consultants
821 Pacific St #4
Santa Monica CA 90405

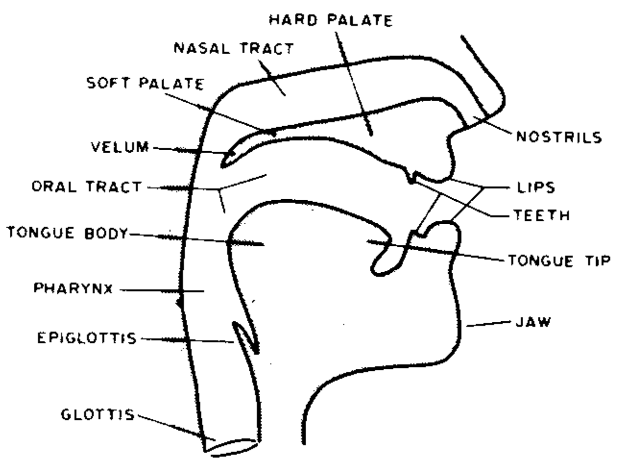


Figure 1: The Human Vocal Tract. The human vocal tract is roughly described as a tube approximately 17.4 cm long with varying resonance characteristics as muscles control the shape. The tract splits into two parts, nasal and oral, at the top, with a valve called the velum providing flexible control of the nasal resonances in given utterance. An electronic model of this natural organ roughly parallels the function of the tract.

Reprinted with permission from
BYTE, Number 12, August 1976

You've got your microcomputer running and you invite your friends in to show off the new toy. You ask Charlie to sit down and type in his name. When he does, a loudspeaker on the shelf booms out a hearty "Hello, Charlie!" Charlie then starts a game of Star Trek and as he warps around thru the galaxy searching for invaders, each alarming new development is announced by the ship's computer in a warning voice, "Shield power low!", "Torpedo damage on lower decks!"

The device that makes this possible is a peripheral with truly unlimited applications, the speech synthesizer. This article describes what a speech synthesizer is like, how it works and a general outline of how to control it with a microcomputer. We will look at the structure of human speech and see how that structure can be generated by a computer controlled device.

How can you generate speech sounds artificially, under computer control? Let's look at some of the alternatives. Simplest of all, with a fast enough digital to analog converter (DAC) you can generate any sound you like. A 7 or 8 bit DAC can produce good quality sound, while somewhere around 4 or 5 bits the quantization noise starts to be bothersome. This noise is produced because with a 5 bit data value it is possible to represent only 32 discrete steps or voltage levels at the converted analog output. Instead of a smoothly rising voltage slope, you would get a series of steps as in figure 2. As for the speed of the DAC, a conversion rate of 8,000 to 10,000 conversions per second [The sample rate in conversions per second or samples per second is often quoted in units of Hertz. We will use that terminology here, although conversions per second is a generalization of the concept of cycles per second] is sufficient for fairly good quality speech. With sample rates below about 6 kHz the speech quality begins to deteriorate badly because of inadequate frequency response.

Almost any microprocessor can easily handle the data rates described above to keep the DAC going. The next question is, where do the samples come from? One way to get them would be by sampling a real speech signal with a matching analog to digital converter (ADC) running at the same sample rate. You then have a complicated and expensive, but very flexible, recording system. Each second of speech requires 8 K to 10 K bytes of storage. If you want only a few words or short phrases, you could store the samples on a ROM or two and dump then sequentially to the DAC. Such a system appears in figure 3.

If you want more than a second or two of speech output, however, the amount of ROM storage required quickly becomes impractical. What can be done to minimize storage? Many words appear to have parts that could be recombined in different ways to make other words. Could a lot of memory be saved this way? A given vowel sound normally consists of several repetitions of nearly identical waveform segments with the period of repetition corresponding to the speech fundamental frequency or pitch. Figure 4 shows such a waveform. Within limits, an acceptable sound is produced if we store only one such cycle and construct the vowel sound by repeating this waveform cycle for the duration of the desired vowel. Of course, the pitch will be precisely constant over that entire interval. This will sound rather unnatural, especially for longer vowel durations, because the period of repetition in a naturally spoken vowel is never precisely constant, but fluctuates slightly. In natural speech the pitch is nearly always changing, whether drifting slowly or

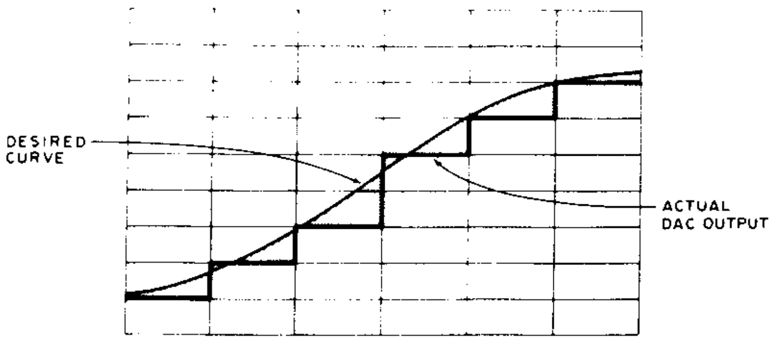


Figure 2: DAC Quantization Errors. The actual output of a computer to the analog world is a step function (in the absence of any filtering). This leads to the problem of quantization errors, depicted conceptually here by the shaded areas in between the smooth analog function and its closest step function approximation. Low precision digital to analog conversions accentuate this problem.

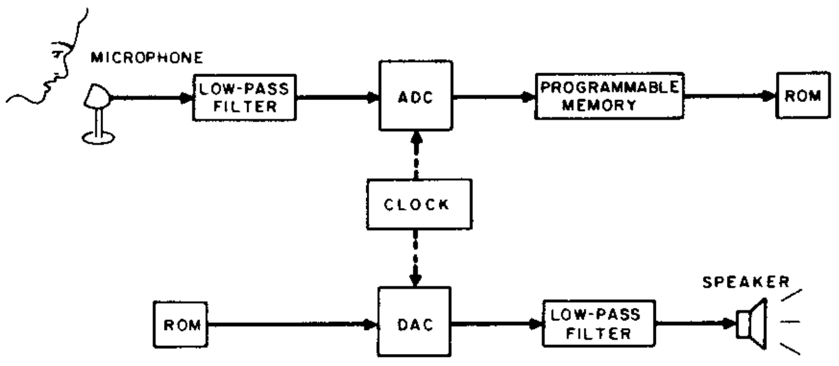


Figure 3: Waveform Playback from ROM Storage. One way to achieve a digitally controlled vocal output is to first digitize a passage of human speech, then store the digital pattern in memory. For a commercial product, such as a talking calculator, the limited vocabulary required makes this a feasible avenue of design, especially when a single mass produced ROM can be used in the final product. In an experimenter's system, the ROM is not needed, and programmable memory can be substituted during experiments. This is probably the least expensive way to augment an existing computer's capability with vocal output, but the memory requirements limit its use to small vocabularies. The quality of the result varies with the ADC (and DAC) sampling rate and precision.

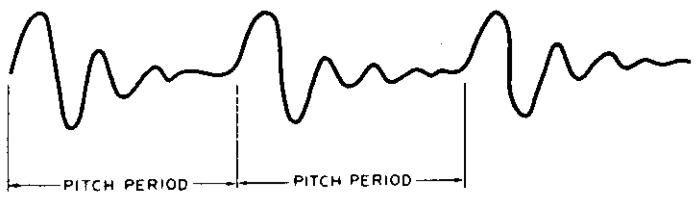


Figure 4: Typical Vowel Waveform. In principle, a vowel is a fairly long sustained passage of sound with repetitive characteristics. The vowel sounds are produced physiologically by the resonances of the vocal tract, and are controlled electronically by the formant filters which produce the equivalent of vocal tract resonances.

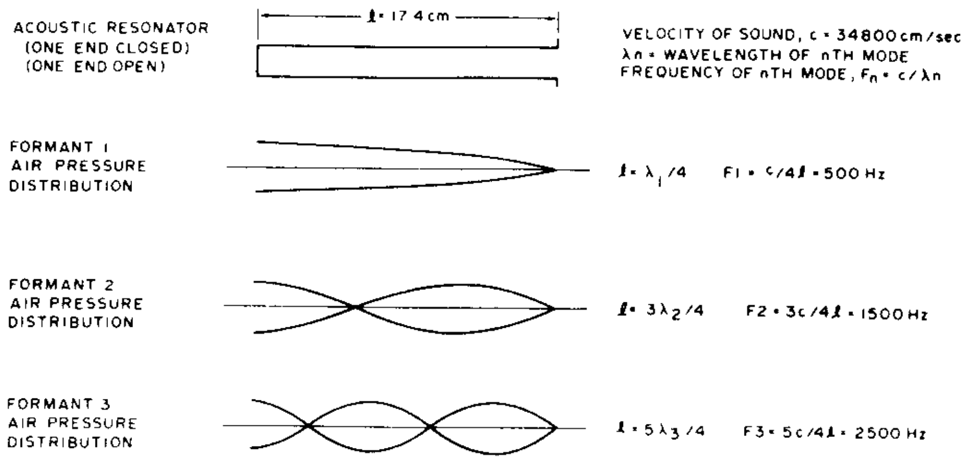


Figure 5: Tube Resonances. Temporarily ignore the complicated shape of the vocal tract and simplify it to a tube 17.4 cm long. Applying the equations of physics to acoustic waves in air gives resonances at several modes or natural frequencies. The standing waves along the tube at each frequency are shown, and identified as formant 1, formant 2 and formant 3. In the actual vocal tract, a more complicated and time varying geometry changes the resonances as a sound is created.

sweeping rapidly to a new level. It is of interest that this jitter and movement of the pitch rate has a direct effect on the perception of speech because of the harmonic structure of the speech signal. In fact, accurate and realistic modelling of the natural pitch structure is probably the one most important ingredient of good quality synthetic speech. In order to have smooth pitch changes across whole sentences, the number of separate stored waveform cycles still gets unreasonable very quickly. From these observations of the cyclic nature of vowels, let us move in for a closer look at the structure of the speech signal and explore more sophisticated possibilities for generating synthetic speech.

How Do We Talk?

The human vocal tract consists of an air filled tube about 16 to 18 cm long, together with several connected structures which make the air in the tube respond in different ways (see figure 1). The tube begins at the vocal cords, or glottis, where the flow of air up from the lungs is broken up into a series of sharp pulses of air by the vibration of the vocal cords. Each time the glottis snaps shut, ending the driving pulse with a rapidly falling edge, the air in the tube above vibrates or rings for a few thousandths of a second. The glottis then opens and the airflow starts again, setting up conditions for the next cycle.

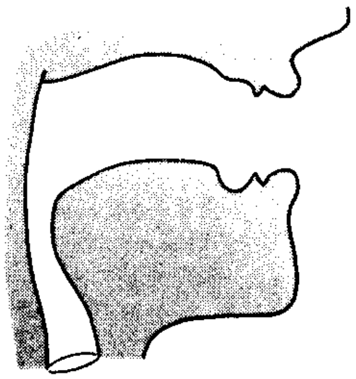


Figure 6: "ah" as in "father." In figure 1, the vocal tract was shown in schematic form. Here is a similar figure showing how the tract has been modified to produce the vowel sound "ah." The human typically closes off the nasal cavity and widens out the oral cavity by opening the mouth during this sound.

The length of this vibrating air column is the distance from the closed glottis up along the length of the tongue and ending at the lips, where the air vibrations are coupled to the surrounding air. If we now consider the frequency response of such a column of air, we see that it vibrates in several modes or resonant frequencies corresponding to different multiples of the acoustic quarter wavelength. There is a strong resonance or energy peak at a frequency such that the length of the tube is one quarter wavelength, another energy peak where the tube is three quarter wavelengths, and so on at every odd multiple of the quarter wavelength. If a tube 17.4 cm long had a constant diameter from bottom to top, these resonant energy peaks would have frequencies of 500 Hz, 1500 Hz, 2500 Hz and so on. These resonant energy peaks are known as the formant frequencies. Figure 5 illustrates the simple acoustic resonator and related physical equations.

The vocal tract tube, however, does not have a constant diameter from one end to the other. Since the tube does not have constant shape, the resonances are not fixed at 1000 Hz intervals as described above, but can be swept higher or lower according to the shape. When you move your tongue down to say "ah," as in figure 6, the back part is pushed back toward the walls of the throat and in the front part of the mouth the size of the opening is increased. The effect of changing the shape of the tube in this way is to raise the frequency of the first resonance or formant 1 (F1) by several hundred Hz, while the frequency of formant 2 (F2) is lowered slightly. On the other hand, if you move your tongue forward and upward to say "ee," as in figure 7, the size of the tube at the front, just behind the teeth, is much smaller, while at the back the tongue has been pulled away from the walls of the throat, leaving a large resonant cavity in that region. This results in a sharp drop in F1 down to as low as 200 or 250 Hz, with F2 being increased to as much as 2200 or 2300 Hz.

We now have enough information to put together the circuit for the oral tract branch of a basic formant frequency synthesizer. After discussing that circuit, we will continue on in this way, describing additional properties of the speech mechanism and building up the remaining branches of the synthesizer circuit.

A Speech Synthesizer Circuit

To start with, we must have a train of driving pulses, known as the voicing source, which represents the pulses of air flowing up thru the vibrating glottis. This could be simply a rectified sine wave as in figure 8. To get different voice qualities, the circuit may be modified to generate different waveform shapes.

This glottal pulse is then fed to a sequence of resonators which represent the formant frequency resonances of the vocal tract. These could be simple operational amplifier bandpass filters which are tunable over the range of each respective formant. Figure 9 shows the concept of a typical resonator circuit which meets our requirements. IC1, IC2 and IC4 form the actual bandpass filter, while IC3 acts as a digitally controlled resistance element serving to vary the resonant frequency of the filter. Several such resonator circuits are then combined as in figure 10 to form the vocal tract simulator. The voicing amplitude control, AV, is another digitally controlled resistance similar to IC3 of figure 9.

This gain controlled amplifier configuration is the means by which the digital computer achieves its control of speech signal elements. The data of one byte drives the switches to set the gain level of the amplifier in question. In figures 10, 13 and 15 of this article, this same variable resistance under digital control is shown symbolically as a resistor with a parameter name,

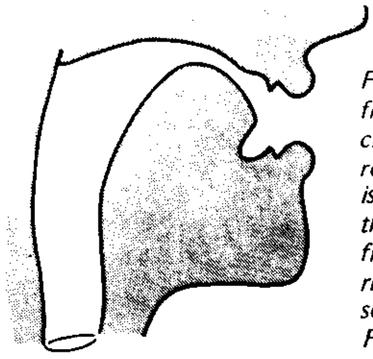


Figure 7: "ee" as in "heed." In contrast to figure 6, when the "ee" vowel sound is created, the mouth opening tends to be narrowed; and the upper end of the vocal tract is restricted. This lowers the frequency of the first resonant mode and raises the frequencies of the second and third. Referring to table 1, the "ee" vowel sound has some of the highest resonances for formants F2 and F3 and the lowest for F1.

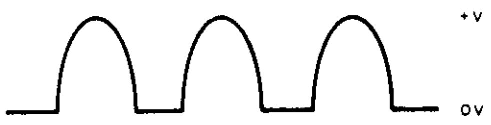


Figure 8: Voiced Sounds from the Glottis. Sounds which have definite pitch are called voiced sounds. In the natural larynx, these sounds are generated by the vocal chords and drive the vocal tract at the glottis. In an electronic analog, the voiced sounds can be generated by a programmable counter (to set the frequency) which in turn creates a sine wave of the same frequency. A rectified sine wave is a good source for the glottal pulses used in the electronic model of a larynx used in the author's approach to speech generation.

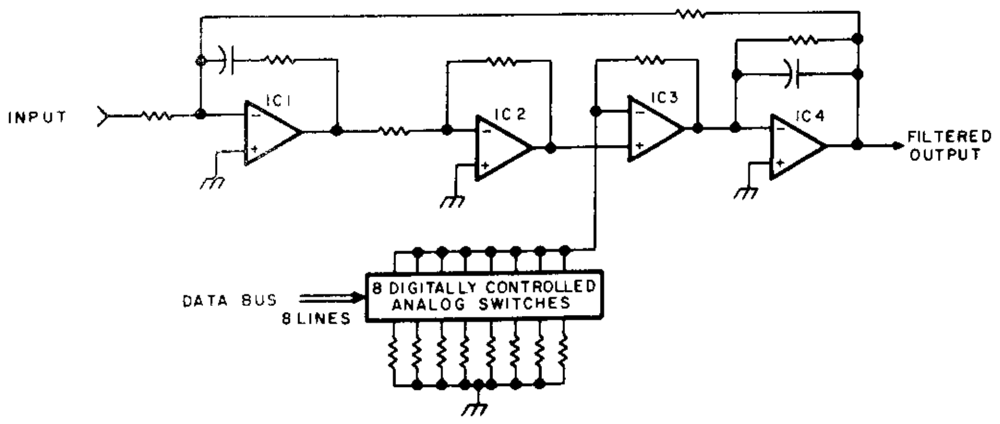


Figure 9: Typical Formant Resonator Circuit. A digitally controlled band pass filter can be built from four operational amplifiers and 8 digitally controlled analog switches. The filter characteristics are set by the choice of the resistance and capacitance elements as well as the digital control word. The operational amplifier IC3 serves as a gain controlled amplifier in the feedback loop, which alters the filter resonance.

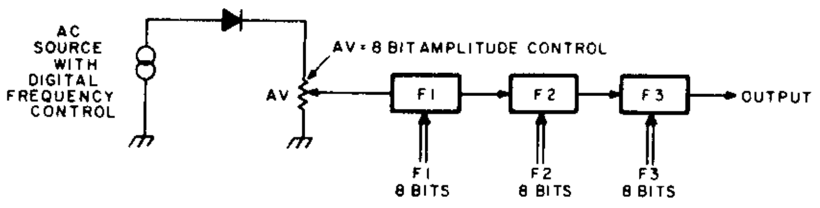

AC SOURCE WITH DIGITAL FREQUENCY CONTROL
AV = 8 BIT AMPLITUDE CONTROL
AV
F1
F2
F3
OUTPUT
F1 8 BITS
F2 8 BITS
F3 8 BITS

Figure 10: A first approximation of the voice synthesizer can be constructed by using three formant filters in series with differing resonance settings all controlled by 8 bit digital words. The resistance indicated as AV is an operational amplifier circuit (see IC3 of figure 9) with a digital gain control input. It is thus a programmable element of gain less than unity, in other words the electronically controlled equivalent of a variable resistance. This notation of a controlled resistance is used in figures 13 and 15 as well.

	F1	F2	F3
heed	250	2300	3000
hid	375	2150	2800
head	550	1950	2600
had	700	1800	2550
hod	775	1100	2500
paw	575	900	2450
hood	425	1000	2400
who	275	850	2400

Table 1: Steady State English Vowels. The vowel sounds are made by adjusting the formant resonances of the human vocal tract to the frequencies listed in this table. These figures are approximate, and actual formant resonances vary from individual to individual. In a speech synthesizer based upon an electronic model of the vocal tract, the formant frequencies are set digitally using operational amplifier filters with adjustable resonant peaks.

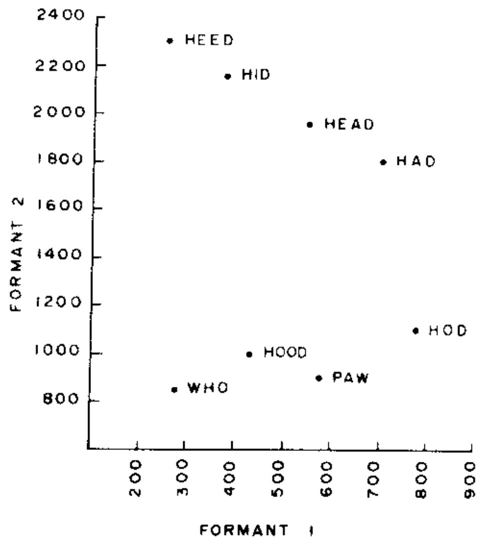


Figure 11: The Steady State English Vowels. The distinctions between various vowel sounds can be illustrated by plotting them on a two dimensional graph. The horizontal axis is the formant 1 frequency, the vertical axis is the formant 2 frequency. A location for each vowel utterance can be determined experimentally by locating the resonance peaks with an audio spectrum analyzer.

rather than as an operational amplifier with analog switches.

Generating Vowel Sounds

The vocal tract circuit as shown thus far is sufficient to generate any vowel sound in any human language (no porpoise talk, yet). Most of the vowels of American English can be produced by fixed, steady state formant frequencies as given in table 1. A common word is given to clearly identify each vowel. The formant frequency values shown here may occasionally be modified by adjacent consonants.

An alternative way to describe the formant relationships among the vowels is by plotting formant frequencies F1 vs F2 as in figure 11. F3 is not shown here because it varies only slightly for all vowels (except those with very high F2, where it is somewhat higher).

The F1-F2 plot provides a convenient space in which to study the effects of different dialects and different languages. For example, in some sections of the United States, the vowels in "hod" and "paw" are pronounced the same, just above and to the right of "paw" on the graph. Also, many people from the western states pronounce the sounds in "head" and "hid" alike, about halfway between the two points plotted for these vowels on the graph.

A few English vowels are characterized by rapid sweeps across the formant frequency

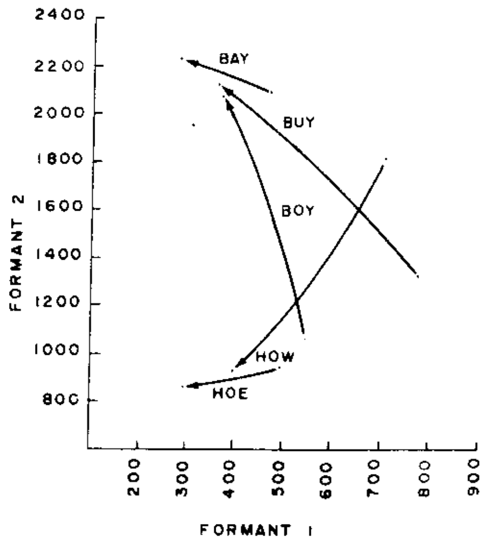


Figure 12: English Diphthongs. A diphthong is a sound which represents a smooth transition from one vowel sound to another during an utterance. The time duration of the swap from one point to another in formant space is typically 150 to 250 ms. This graph shows typical starting and ending points for several common diphthong sounds.

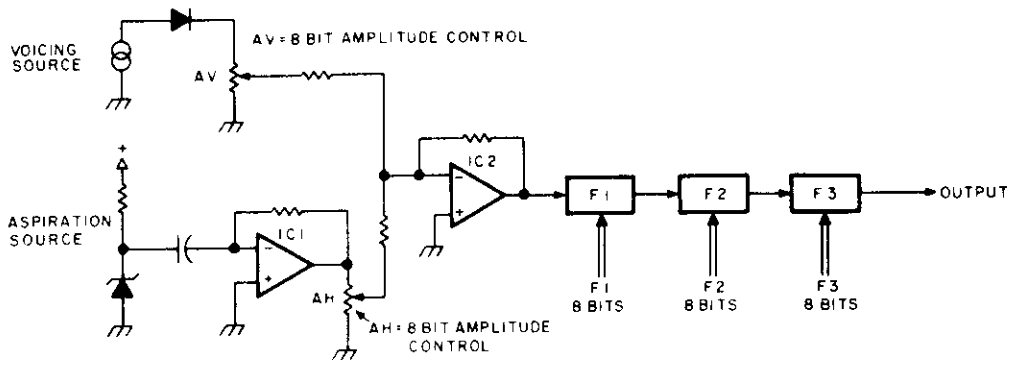


Figure 13: Synthesizer with Aspiration Noise Generator. Not all utterances are vowels. By adding a digitally controlled noise generator to the circuit of figure 10, it is possible to synthesize the consonant sounds known as "stops." In this circuit, the amplitude versus time characteristics of the noise pulse are determined by an 8 bit programmable gain control AH (shown symbolically as a resistor). The output of the noise source is mixed with the voicing source with the analog sum being routed to the formant filters. The noise generator is a zener diode.

space rather than the relatively stable positions of those given in table 1. These sweeps are produced by moving the tongue rapidly from one position to another during the production of that vowel sound. Approximate traces of the frequency sweeps of formants F1 and F2 are shown in figure 12 for the vowels in "bay," "boy," "buy," "hoe" and "how." These sweeps occur in 150 to 250 ms roughly depending on the speaking rate.

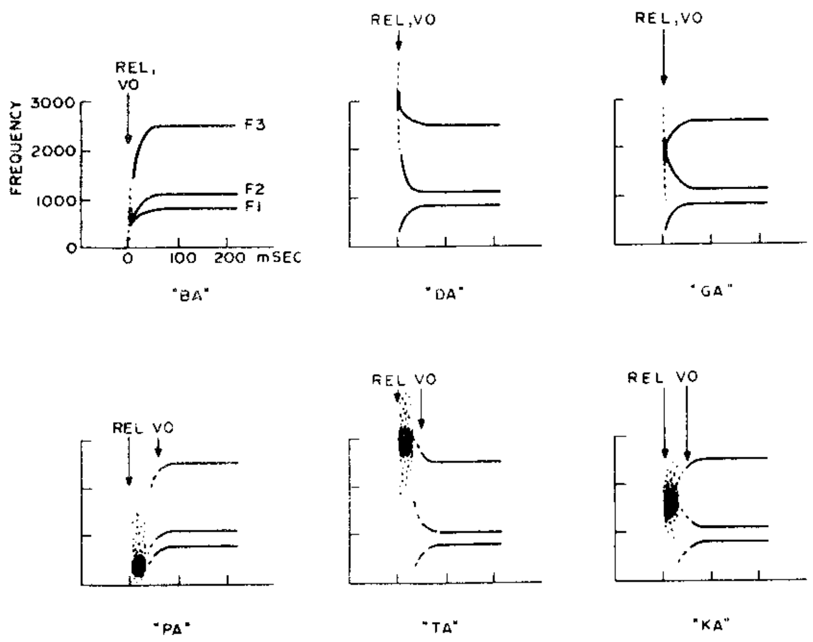


Figure 14: Stop Consonant Patterns. This figure illustrates 6 different stop consonant patterns. The release of the stop closure (start of noise pulse) is at the point marked by "Rel" and the beginning of the voicing sounds is marked by "VO". Note the typical transition of the vowel formants as the steady state is reached.

Consonant Sounds

Consonant sounds consist mostly of various pops, hisses and interruptions imposed on the vibrating column of air by the actions of several components of the vocal tract shown in figure 1. We will divide them into four classes: 1) stops, 2) liquids, 3) nasals, and 4) fricatives and affricates. Considering first the basic 'stop consonants,' "p," "t," "k," "b," "d" and "g," the air stream is closed off, or stopped, momentarily at some point along its length, either at the lips, by the tongue tip just behind the teeth or by the tongue body touching the soft palate near the velum. Stopping the air flow briefly has the effect of producing a short period of silence or near silence, followed by a pulse of noise as the burst of air rushes out of the narrow opening.

The shape of the vocal tract with the narrow opening at different points determines the spectral shape of the noise pulse as well as the formant locations when voicing is started. Both the noise burst spectrum and the rapid sweeps of formant frequency as the F1-F2 point moves into position for the following vowel are perceived as characteristic cues to the location of the tongue as the stop closure is released. We need only add a digitally controlled noise generator to the vocal tract circuit of figure 10 to simulate the noise of the burst of air at the closure release and we can then generate all the stop consonants as well as the vowels. Figure 13 shows the speech synthesizer with such a noise generator added. The breakdown noise of a zener diode is amplified by IC1 and amplitude is set by the digitally controlled resistor AH. IC2 is a mixer amplifier which combines the glottal source and aspiration

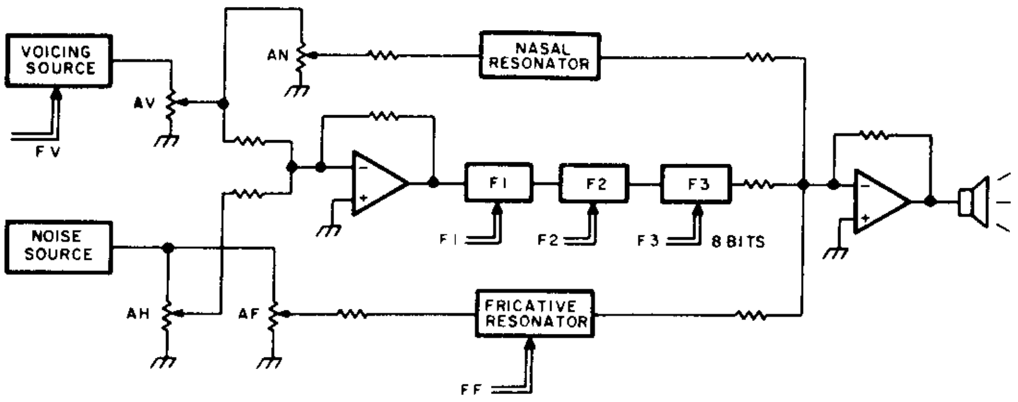


Figure 15: The Complete Synthesizer. This diagram shows the organization of a complete synthesizer which includes a wide variety of parameters. The voicing frequency and amplitude are set by parameters FV and AV. The noise pulses of stop consonants are generated with the programmable gain element AH. The fricative resonator with amplitude AF and frequency resonance FF are used to generate fricatives like "s" and "sh." The normal vowel sounds are generated by control of the formant frequencies F1, F2 and F3, and a nasal resonator with amplitude AN and fixed frequency characteristics is used to add varying amounts of nasal sounds. The result of signals processed through the nasal, formant and fricative paths is summed by a final operational amplifier and used to drive the output speaker.

noise at the input to the formant resonators.

It is important to notice at this point the range of different sounds that can be generated by small changes in the relative timing of the control parameters. The most useful of these timing details is the relationship between the pulse of aspiration noise and a sharp increase in the amplitude of voicing (see figure 14). For example, if we set the noise generator to come on for a noise pulse about 40 ms long and immediately after this pulse, F1 sweeps rapidly from 300 up to 775 Hz and F2 moves from 2000 down to 1100 Hz, the sound generated will correspond to moving the tip of the tongue down rapidly from the roof of the mouth. Observe, however, that the formant output is silent after the noise pulse until the voicing amplitude is turned up. If voicing is turned on before or during a short noise burst, the circuit generates the sound "da," whereas if the voicing comes on later, after a longer burst and during the formant frequency sweeps, the output sounds like "ta." This same timing distinction characterizes the sounds "ba" vs "pa" and "ga" vs "ka," as well as several other pairs which we will explore later. Figure 14 gives the formant frequency patterns needed to produce all the stop consonants when followed by the vowel "ah." When the consonant is followed by a different vowel, the formants must move to different positions corresponding to that vowel.

The important thing to note about a stop transition is that the starting points of the frequency sweeps correspond to the point of closure in the vocal tract, even though these sweeps may be partially silent for the unvoiced stops "p," "t" and "k," where the voicing amplitude comes on after the sweep has begun.

The second consonant group comprises the liquids, "w," "y," "r" and "l." These sounds are actually more like vowels than any of the other consonants except that the timing of formant movements is crucial to the liquid quality. "W" and "y" can be associated with the vowels "oo" and "ee," respectively. The difference is one of timing. If the vowel "oo" is immediately followed by the vowel "ah," and then the rate of F1 and F2 transitions is increased, the result will sound like "wa." A comparison of the resulting traces of F1 and F2 vs time in "wa" with the transition pattern for "ba" in figure 14 points out a further similarity. The

	Resonator Frequency (FF)	Fricative Amplitude (AF)
sh, zh	2500	.9
s, z	5000	.7
f, v	6500	.4
th	8000	.2

Table 2: Fricative Spectra. A fricative sound typically consists of a pulse of high frequency noise. The various types of fricatives are classified according to the spectral profile of the pulse. For the electronic model described here, the fricative amplitude and resonator frequency for several sounds are listed in this table.

direction of movement is basically the same, only the rate of transition of "ba" is still faster than for "wa." Thus we see the parallelism in the acoustic signal due to the common factor of lip closeness in the three sounds "ua," "wa" and "ba." "Y" can be compared with the vowel "ee" in the same way, so the difference between "ia" and "ya" is only a matter of transition rates. Generally, "l" is marked by a brief increase of F3, while "r" is indicated by a sharp drop in F3, in many cases, almost to the level of F2.

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

The fourth series of consonants to be described are the fricatives, "s," "sh," "z," "zh," "f," "v" and "th" and the related affricates "ch" and "j." The affricates "ch" and "j" consist of the patterns for "t" and "d" followed immediately by the fricative "sh" or "zh," respectively, that is, "ch" = "t+sh" and "j" = "d+zh." The sound "zh" is otherwise rare in English. An example occurs in the word "azure." With the letters "th," two different sounds are represented, as contained in the words "then" and "thin." All the fricatives are characterized by a pulse of high frequency noise lasting from 50 to 150 msec. The first subclassification of fricatives is according to voicing amplitude during the noise pulse, just as previously described for the stop consonants. Thus, "s," "sh," "f," "ch" and "th" as in "thin" have no voicing during the noise pulse, while "z," "zh," "v," "j" and "th" as in "then" have high voice amplitude. When a voiceless fricative is followed by a vowel, the voicing comes on during the formant sweeps to the vowel position, just as in the case of the voiceless stops. The different fricatives within each voice group are distinguished by the spectral characteristics of the fricative noise pulse. This noise signal differs from that previously described for the stop bursts in that it does not go thru the formant resonators, but is mixed directly into the output after spectral shaping by a single-pole filter. Table 2 gives the fricative resonator settings needed to produce the various fricative and affricate consonants. Fricative noise amplitude settings are shown on a scale of 0 to 1.

Product Information

At the time this article goes to press, a synthesizer module incorporating several detail refinements and improvements over the circuits of this article is being developed by the author and associates. A detailed user's guide will be supplied with the Computalker module which illustrates the timing relationships needed to produce all the consonant-vowel and vowel-consonant combinations which occur in natural speech. This can serve as a reference guide for creating your speech output software which generates the proper control patterns from text inputs. Write to Computalker, 821 Pacific St No. 4, Santa Monica CA 90405 for the latest information on this module.

The Complete Synthesizer

The system level diagram of a complete synthesizer for voice outputs is summarized in figure 15. The information contained in this article should be sufficiently complete for individual readers to begin experimenting with the circuitry needed to produce speech outputs. In constructing a synthesizer on this model, the result will be a device which is controlled in real time by the following parameters:

AV =	amplitude of the voicing source, 8 bits
FV =	frequency of the voicing source, 8 bits
AH =	amplitude of the aspiration noise component, 8 bits
AN =	amplitude of the nasal resonator component, 8 bits
AF =	amplitude of the fricative noise component, 8 bits
F1 =	frequency of the formant 1 filter, 8 bit setting.
F2 =	frequency of the formant 2 filter, 8 bit setting.
F3 =	frequency of the formant 3 filter, 8 bit setting.
FF =	frequency of fricative resonator filter, 8 bit setting.

This is the basic hardware of a system to synthesize sound; in order to complete the system, a set of detailed time series for settings for these parameters must be determined (by a combination of the theory in this article and references, plus experiment with the hardware). Then, software must be written for your own computer to present the right time series of settings for each sound you want to produce. Commercial synthesizers often come with a predefined set of "phonemes" which are accessed by an appropriate binary code. The problem of creating and documenting such a set of phonemes is beyond the scope of this introductory article, but is well within the dollar and time budgets of an experimenter.■

BIBLIOGRAPHY

1. Erman, Lee, ed, IEEE Symposium on Speech Recognition, April, 1974, Contributed Papers, IEEE Catalog No. 74CH0878-9 AE.

2. Flanagan, J L, and Rabiner, L R, eds, *Speech Synthesis*, Benchmark Papers in Acoustics, Dowden, Hutchinson & Ross, Inc, 1973.

3. Lehiste, Ilse, ed, *Readings in Acoustic Phonetics*, MIT Press, 1967.

4. Moschytz, George S, *Linear Integrated Networks Design*, Van Nostrand, New York, 1975.

Hardware & software for speech synthesis

by Lloyd Rice   Computalker Consultants
P. O. Box 1951   Santa Monica, CA 90406

The process of generating voice output with a computer can be broken down into several steps. We will examine the operations at each step to determine the flow of information into and out of the step. This examination will give us the background needed to decide which parts of the overall process should be wired into a hardware device, and which parts should be kept as software to retain flexibility and control over the process. Perhaps the easiest way to carry out such an examination is by following an example phrase thru the system as it is transformed into a speech signal and sent to the loudspeaker.

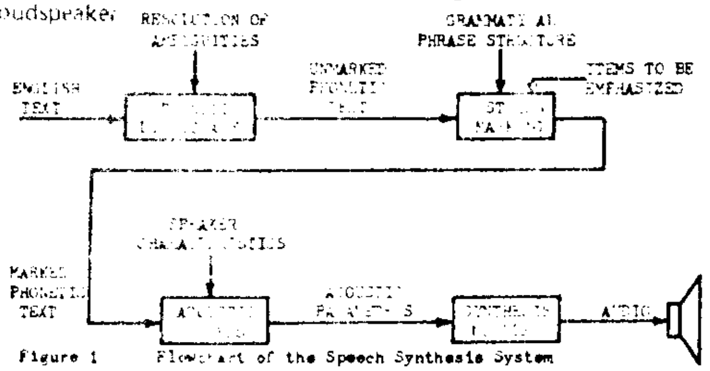


Figure 1   Flowchart of the Speech Synthesis System

Figure 1 shows a flowchart of the speech output system to be described. We will see that the kinds of external information needed for the first 2 steps is quite difficult to obtain and can require large amounts of processing, whereas the information needed in the third step is easily determined, and in most applications can be set as constants in the system. Finally, the acoustic parameters contain all the information necessary to control the last step, the actual synthesizer, to produce audio output. As a result of these observations, we will see that in most cases, one should specify the material to be synthesized in the form of marked phonetic text rather than raw English text. In order to present a more complete description, however, we will begin with the first step shown in Figure 1, input of English text.

Beginning with the sample text, "This is computer speech.", we first consult a phonetic dictionary, which performs a direct translation to phonetic text. A phonetic coding scheme suitable for this purpose which is compatible with the ASCII character set and Teletype output was developed by the Advanced Research Projects Agency (ARPA) as a part of a recent speech recognition study. That phonetic code, known as ARPABET, is listed in Table 1. The output of the phonetic dictionary in our example would be, "DHIHS/IHZ/KAHM-PYUWTER/SPIYCH #".

The main problem which arises at this stage is due to homographs, words that are spelled the same but pronounced differently. Two different types of homographs, however, present quite different problems. The first type consists mainly of short words such as "bow," pronounced either as in "tied a bow" and "bow and arrow," or as in "off the starboard bow." In these cases, the pronounciation can usually be resolved by examining the surrounding context. The other type of ambiguity is a lesser noticed but very widespread phenomenon in English: the situation where a word has a different stress pattern depending on whether it is used as a noun or a verb. As an example of this, notice the difference in "It was a dull subject," and "They were going to subject him to cruel punishment." It is not evident from the spelling which usage is intended, and requires that a fairly complete grammatical analysis be carried out to make that decision. One advantage at this point is that good use can be made of the recovered grammatical structure in the next step, where a more elaborate assignment of stress is performed.

The second step in the synthesis process deals with the assignment of sentence stress levels to the phonetic text string. To clarify that operation we will first have a closer look at the nature of the linguistic feature known as stress. The stress will be coded as a numerical value attached to a vowel in the phonetic string. That value will be realized later by the synthesizer in three different ways: as an increase in the pitch frequency, as a lengthening of the vowel duration, and to some extent as an increase in amplitude. The primary or highest level of stress is marked as a "1" following the vowel symbol. Secondary stress, marked with a "2," has less extreme acoustic effects than primary stress. As many as 3 or 4 distinct levels of stress may be marked in a sentence.

With regard to its communicative value, stress serves two quite distinct functions. The sentence stress pattern, together with timing and intonation, serves to communicate the grammatical structure to the listener. One can think of the grammatical structure as being transformed into a stress and intonation pattern by the speaker which is then decoded back into the phrase structure by the listener. Using the term "grammar," I am here including several kinds of information about words, such as the noun-verb distinction discussed above as well as syntactic information about the phrase and clause structure. The second function of the stress pattern is to indicate which item or items in a sentence are to be given special emphasis. The meaning of a sentence can be shifted around by emphasizing different items. The sources of information needed for marking these two components of the stress pattern are quite different and must be considered separately. Our example, with the stress pattern marked, would be something like, "DHIH3S/IHZ/KAHMPYUW1TER/SPIY2CH #". Notice also that the word and utterance boundary markers have been kept explicitly in the text string.

The purpose of the portion of the system described thus far is simply to generate strings of phonetic text with marked stress patterns which are to be synthesized by the 2 steps in the bottom row of Figure 1. Marked phonetic text strings can be obtained in other ways, of course. In the case of predetermined phrases, marked phonetic strings can be stored instead of raw English text, making the synthesis task much simpler. On the other hand, consider synthesis of speech from an information network of some kind. The grammatical information could come from a phrase structure grammar which is being driven by relationships in the network. Items in the network would be coded as phonetic strings, or in essence, references to the phonetic dictionary described above. There are many significant problems remaining with this approach, but it is perhaps one of the more exciting applications of synthetic speech. The third box in the flowchart in Figure 1 is the acoustic rules section. In order to describe what the acoustic rules are and what they do, we must first look at the acoustic structure of speech. The speech code must be broken down into components so it can be synthesized by controlling, in real time, a limited number of parameter values. To a good approximation, speech can be represented by the model shown in Figure 2.

This model requires 9 parameter control values consisting of 5 frequency controls and 4 amplitude controls. The box labeled "pulse source" is a controllable frequency oscillator which is adjusted dynamically to determine the voice pitch. The boxes labeled "resonator" are tunable, single-pole, bandpass resonators which determine the frequency or spectral shape of the speech signal in different ways. The data bus symbol used to represent the control inputs indicates that each parameter can be controlled by at most 8 bits from the computer's output bus. The data rates needed to control the

Table 1

COMPUTER

PHONETIC

REPRESENTATIONS

NOTE: Spaces are ignored except within escapes.

Phoneme	Computer Representation		Example	Phoneme	Computer Representation		Example
	1-Character	2-Characters			1-Character	2-Characters	
i	i	IY	beat	p	p	P	pet
ɪ	I	IH	bit	t	t	T	ten
e	e	EY	bait	k	k	K	kit
ɛ	E	EH	bet	b	b	B	bet
æ	@	AE	bat	d	d	D	debt
ɑ	a	AA	Bob	g	g	G	get
ʌ	A	AH	but	h	h	HH	hat
ɔ	o	AO	bought	f	f	F	fat
o	o	OW	boat	θ	T	TH	thing
ʊ	U	UH	book	s	s	S	sat
u	u	UW	boot	š or ʃ	S	SH	shut
ə	x	AX	about	v	v	V	vat
ɨ	X	IX	roses	ð	D	DH	that
ɝ	R	ER	bird	z	z	Z	zoo
aʊ or aw	W	AW	down	ž or ʒ	Z	ZH	azure
aɪ or ay	Y	AY	buy	č	C	CH	church
ɔɪ or ɔy	O	OY	boy	ǰ	j	JH	judge
y	y	Y	you	ʍ	H	WH	which
w	w	W	wit	syl l,l̩	L	EL	battle
r	r	R	rent	syl m,m̩	M	EM	bottom
l	l	L	let	syl n,n̩	N	EN	button
m	m	M	met	flapped t,ɾ	F	DX	batter
n	n	N	net	glottal stop,ʔ	Q	Q	
ŋ	G	NX	sing	silence	-	-	
				non-speech segment	!	!	laugh, etc.

AUXILIARY SYMBOLS (1- AND 2-CHARACTER CODES ARE IDENTICAL)

Symbol	Meaning	Symbol	Meaning
+	Morpheme boundary	:3 or ,	Fall-rise or non-term juncture
/	Word boundary	* **	Comment (anything except * or **)
#	Utterance boundary	' '	Apos-surround special symbol in comment
:	Tone group boundary		
:1 or .	Falling or decl. juncture	( )	Phoneme class information
:2 or ?	Rising or inter. juncture	< >	Phonetic or allophonic escape

STRESS REPRESENTATIONS (IF PRESENT, MUST IMMEDIATELY FOLLOW THE VOWEL)

Value	Stress Assignment	Value	Stress Assignment
0	No stress	3	Tertiary stress
1	Primary stress	.	(Etc.)
2	Secondary stress	.	

parameters are quite low, the highest rate needed for any parameter being less than 100 new settings per second.

I will not go into detail here describing the actual parameter values needed to represent particular speech sounds. An article to appear in the August, 1976 issue of *Byte Magazine* goes into some detail on the nature of the different kinds of speech sounds and how they can be generated by controlling the parameter values in such a model. Such information would, of course, be necessary to write a software implementation of the acoustic rules. For our present purposes, we consider the 9 control values as outlined above to represent an acoustic parameter model of speech. We can now turn to a discussion of the acoustic rules and the tasks they must perform to generate controls for this model.

Each phoneme, as encoded in the phonetic text string, is a symbol representing one or more acoustic speech segments, each such segment being produced by a particular pattern of values on the control parameters. Each pattern, or configuration of control values, must be held for a specific length of time before changing to the next pattern. As a first approximation then, the rules would consist of a series of table look-ups to convert each phoneme into a sequence of parameter patterns, along with the duration each pattern is to be held.

Now comes the catch! This first approximation makes rather poor speech. The problem is that the transitions between parameter values are often more important than the actual values at any given time. The flow of parameter values must be more carefully orchestrated. Actually, the only tough problem here is that correct transitions *between* phonemes are just as important as having the correct temporal structure *within* a phoneme. This means that phonemes cannot be coded as independent sets of parameter time functions which be merely joined together sequentially, but that some interaction must take place between phoneme patterns before they are sent out to the synthesizer module. Briefly, the different phonemes of a language can be classified according to the effects of boundary interactions. The transition of each parameter value across a given phoneme boundary can then be determined from the boundary characteristics of each of the neighboring phonemes. Such boundary behavior information can be stored in phoneme look-up tables.

In addition to assigning initial parameter values and mapping the transitions across boundaries, the acoustic rules must also assign and modify durations. For example, a stressed vowel is given a longer duration than the same vowel in an unstressed position.

A third function the acoustic rules must perform--probably the most important for natural sounding speech--is to assign the time pattern of values to the pitch frequency parameter. First, an archetypal intonation pattern is chosen on the basis of punctuation (retained in the phonetic text just for this purpose). A period selects a falling pitch, a comma signals a level pause, and a question mark indicates a rising pattern. Other diacritical marks could be defined in the phonetic string to generate more complex pitch patterns such as singing. The selected archetypal pitch pattern is then modified locally by specific phonemes. Such local modification of the pitch pattern is one of the effects of a stress level marked on a vowel. Also, some consonants affect the pitch value slightly.

To complete the synthesis process, the acoustic parameters generated by the acoustic rules are output, in real time, to a synthesizer module such as sketched in Figure 2 and

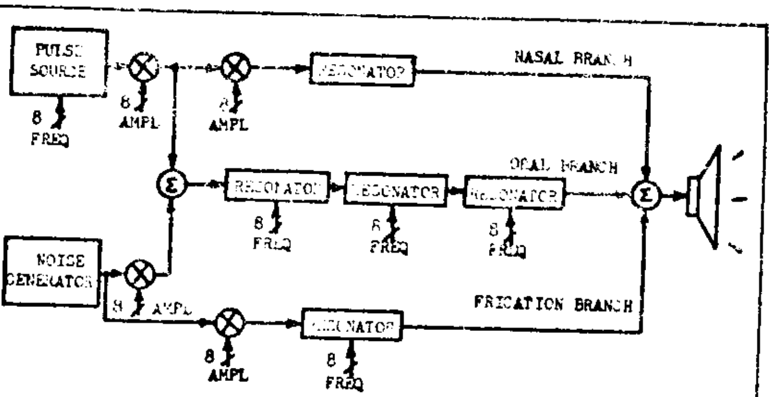


Figure 2 A Model of the Structure of Speech

described in the forthcoming *Byte* article. The synthesizer constructs an audio frequency signal as specified by the control parameters. The audio signal is then sent to a loudspeaker as the speech output.

It would be impractical to consider simulating the synthesizer module in software because of the speed needed to generate speech in real time. That task is much more appropriately handled by analog hardware. Such a hardware synthesizer module is currently being developed by Computalker, P. O. Box 1951 , Santa Monica CA 90406. The Computalker synthesizer module would be driven by the microcomputer output data bus as described above. The software interface consists either of a direct, manually-controlled parameter pattern generator or an implementation of the acoustic rules. Software for the acoustic rules will also be developed by Computalker as the hardware becomes available.

I believe it is important to consider at this point some of the trade-offs involved in implementing the acoustic rules in software rather than hardware. A synthesizer system such as the Votrax VS-6 contains a hardware implementation of the basic acoustic rules. As a result, the language available for coding the phonetic text is fixed and cannot be extended. In addition, the phoneme table values are fixed so that each phoneme has a set phonetic quality. By implementing these rules in software you could retain the flexibility over pitch patterns and speech rate and also have control over the phonetic qualities which determine the language and dialect. The acoustic rules determine a number of qualities in the resulting speech which are characteristic of a particular speaker, such as the sex and age, and other qualities which vary from occasion to occasion, such as voice quality, speaking rate, distinctness of articulation, etc. Because of time constraints, a software version of the acoustic rules may not have time to handle all these possibilities as on-line variables. Of course, it is cheaper to produce a synthesizer module if a hardware acoustic rules system is not included.

How could speech output from a microcomputer be used? Several applications come to mind for the hobbyist environment, such as responses in games, voice readout of measurement data, system status warnings, etc., etc. Other applications might include telephone answering and intrusion warnings. What about generating audio tape labels automatically? Each of these applications makes its own demands for quality, naturalness and range of vocabulary needed. I would very much like to hear of your interest in computer speech output. What applications do you have in mind? What problems do you foresee? A note to the above address will assure that you receive further information as it becomes available.

CTMON, The Computalker Control Monitor

CTMON is a combined speech data editor and CT-1 playback monitor. The CT-1 control parameters are displayed as columns of decimal values on the video monitor. The speech parameter data can be read from paper tape or audio cassette, edited as desired, and written back out onto audio cassette. The speech data in the buffer may be played out to the CT-1 Synthesizer beginning at any desired data frame and played for any desired number of frames. The rate of playback may be varied, resulting in fast speech at up to 10 times the normal speaking rate, or slow speech, drawing the words out to more than 5 times the normal length of time.

VIDEO DISPLAY

CTMON is currently assembled to use a Polymorphic Systems VIDEO display board and can easily be patched to use a Processor Technology VDM-1 display system. It would not be easy to modify the program as it now stands to be able to use a TVT or Teletype-replacement type of video display. The reason for this is that I wanted to be able to scroll up or down thru the frames of data quickly to be able to move around in the data as fast as you could hit the "N" or "B" keys (for Next or Back). I did not want to take the time to rewrite the entire screen for each scroll. A scrolling feature such as the Processor Technology VDM-1 would not be of much help either, because I wanted to keep several lines fixed at the top and bottom of the screen. I usually run my video board (Polymorphic) with the width control set back so there are about 40 chars across the screen. As a result, I wrote the frame display code so it would work with less than 64 characters width (being too short sighted, tho, I did not quite get it in 32). This detail could be changed by modifying the subtraction and sign test code just before the label CT2A at location Ø186H and similarly in the subroutines DISFRM at Ø57BH and ARROW at Ø5B8H. Also the column headings should then be changed. The text lines FRMTX, FORTX, and AMPTX should then be combined into one line at FRMTX. This would affect the character count at Ø13BH.

The program was assembled using Polymorphic's suggested video buffer address of 88ØØH. To change this to the address of your video display requires changing three locations as follows. If the symbol VIDBUF is the page number of the first video location, then the full 16 bit address of that location will be VIDBUF*256. With a 1K byte video buffer, VIDBUF+4 will be the next page beyond the end of video memory and VIDBUF/4 is just the bit pattern of VIDBUF shifted right 2 bits. The examples shown below would address the screen properly for VIDBUF EQU ØEØH.

location	now contains	change to	symbol
Ø694H	88H	ØEØH	VIDBUF
Ø69AH	8CH	ØE4H	VIDBUF+4
Ø6EDH	22H	38H	VIDBUF/4

The Polymorphic VIDEO board requires that the most significant bit be set high to get text mode instead of graphics. With a VDM-1, this will give either positive or negative characters depending on the setting of the on-board switch. If you must have the most significant bit low for your board, this may be done by changing the following locations as shown.

location	now contains	change to
Ø1BAH	9BH	ØDH
Ø1D2H	ØAØH	2ØH
Ø59DH	8ØH	ØØ
Ø5CDH	8ØH	ØØ
Ø681H	8ØH	ØØ
Ø696H	ØAØH	2ØH
Ø6ACH	ØAØH	2ØH
Ø6FCH	8ØH	ØØ
Ø737H	8ØH	ØØ
Ø73FH	8ØH	ØØ

Of these, location Ø1BAH contains the symbol to be displayed as a left-pointer just to the right of the selected parameter data value. The Polymorphic board with a 6571 character generator ROM has a back-arrow at code 9BH. The VDM-1 which I tried this system on had a suitable character at code ØDH. You may have to put a different code here depending on your character generator ROM.

OTHER I/O

Reading and writing of speech parameter data is done under the control of separate monitor commands for each device.

command	action
TR	Read data from paper tape & replace current buffer contents
TA	Read data from paper tape & append it to data currently in buffer
TW	Punch buffer contents on paper tape (not yet implemented)
TY	List buffer contents on TTY or printer
CR	Read data from cassette & replace current buffer contents
CA	Read data from cassette & append it to data currently in buffer
CW	Write buffer contents onto cassette

Code is currently written for Teletype input (using MITS definitions) and input and output using Tarbell cassette definitions. All instructions specific to an I/O device are located near the end of the listing at locations Ø74FH thru Ø7B4H. These routines are all reasonably well commented as far as which way flags are set and what's being returned, etc. In general, it will only be necessary to change the OUT and IN addresses to those needed for your device assignments. In most of these routines, space has been left for a NOP or CMA before the bit-masking, depending on whether your device has active high or active low status.

GETC	Waits for strobe from console keyboard, then gets 1 ASCII char and returns with ms bit low in A register.
KEYTST	This only checks whether a key has been hit, it does not get the character. This leaves the strobe still active for GETC. If a key was hit, KEYTST returns with a non-zero condition.
TTYOUT	This has not yet been coded. It is called from location Ø35CH on the command "TY", which would produce a formatted listing as opposed to a parameter data format dump to paper tape.

Two Teletype/paper tape output command processors have not been completed. TTYLIS would construct a formatted listing as mentioned above, and TPDATA would punch the buffer contents on tape in parameter data format. For Teletype operation, these would both call TTYOUT, and would call separate subroutines if a separate paper tape punch were available.

PTRCLR This initializes the paper tape reader at the beginning of each input transmission. Both this and PTRIN below are currently set up for a MITS 2SIO board.

PTRIN This subroutine waits for the paper tape reader ready flag to go on, or a time limit to expire, and if ready, gets 1 data byte from the reader and returns it in A. Note that this is 8 bit data, not ASCII, so the ms bit should not be masked off. A timer is included so you don't hang forever on an unready ready flag. If the flag does not go on within approx. ½ second, the routine returns zero condition. If the data byte was read normally, it returns non-zero condition with the byte in A. Registers BC are used, but restored before return.

CASSETTE I/O

The cassette routines are assembled using Don Tarbell's standard definitions of 6EH (status) and 6FH (data).

CASCLR This initializes the cassette reader board and clears the B register for use as a checksum accumulator. It is called once at the beginning of each input transmission, on either command "CR" or "CA". Note that the cassette should be moving forward reading leader at the time either of these commands is given.

CASIN This subroutine reads 1 byte from cassette. It is called with the read address in the HL registers and puts the byte just read directly into memory at (HL). The address in HL is then incremented. In addition, the value of the byte read is added to the contents of the B register, which is accumulating the checksum for the record. All registers and flags except the B register are restored before return.

CASLDR This sets up everything for a cassette write transfer including leader and all sync bytes. My Tarbell board is wired with a relay which closes by setting status bit Ø to a 1. This connects the audio to the cassette input. If your system is set up so that starts the cassette motor, too, you may want to add more than the 2 second delay I used. I should have cleared B here also, but that's done back in CWDATA where this is called.

CASOUT This writes 1 byte from A onto the cassette. Note that it does not write directly from memory as with CASIN. CASOUT also accumulates a checksum in the B register.

BEFORE USING THE TARBELL I/O ROUTINES, CHANGE THE FOLLOWING LOCATION:

LOC	NOW CONTAINS	CHANGE TO
Ø3C7	81 H	96 H

CASSETTE I/O Modifications for MITS ACR

The following modifications should be patched into CTMON in order to use MITS ACR cassette, assuming MITS standard port assignments, flags=Ø6, data=Ø7. When using this cassette I/O to write a record, the drive should be connected in RECORD mode for several seconds before typing the "CW" command. When reading a cassette data record, enter the command "CR" or "CA" while the head is reading this leader.

location	now contains	change to		symbolic code
Ø3A8	CD 81 Ø7	CD 96 Ø7		CALL CASCLR2
Ø3C6	CD 81 Ø7	CD A5 Ø7		CALL CASLDR
Ø3EØ	AF	C9		RET
Ø787	DB 6E	DB Ø6		IN CASF
Ø789	E6 1Ø	E6 Ø1		ANI Ø1
Ø78E	DB 6F	DB Ø7		IN CASD
Ø796	3E Ø1	D3 Ø6	CASCLR2	OUT CASF
Ø798	D3 6E	DB Ø7		IN CASD * reset UART
Ø79A	CD 53 Ø6	21 81 Ø7		LXI H,CASCLR *unused loc
Ø79D	3E 3C CD	CD 86 Ø7		CALL CASIN *read & ignore
Ø7AØ	A7 Ø7	Ø6 ØØ		MVI B,Ø start byte
Ø7A2	3E	C9		RET
Ø7A5	A7 Ø7	3E FF		MVI A,ØFFH * write start byte
Ø7A8	DB 6E	DB Ø6		IN CASF
Ø7AA	E6 2Ø	E6 8Ø		ANI 8ØH
Ø7BØ	D3 6F	D3 Ø7		OUT CASD

PLAYING and EDITING SPEECH DATA with CTMON

When you first bring up the Control Monitor system, you see frame 1 displayed across the screen with Ø's shown in each data field. Put the demonstration data tape "HELLO" in the paper tape reader and then type the command "TR". Now start the tape reader. At the end of the tape, the command TR should disappear from the screen and several columns of data should be in view, beginning with frame 1. The total length of data in the buffer is shown to be 746 frames.

TYPE "P" TO HEAR IT

With the CT-1 Synthesizer board in place and connected to an audio amplifier, you should now be able to type the command "P" and hear the demonstration phrase. The playback sequence can be interrupted any time in the middle of the phrase by typing either Rubout or Escape. These keys serve as the general abort function for any playback operation or for falsely entered commands. Up to this point, the commands described had no arguments and were executed immediately upon typing the command letter(s). We now consider the second type of command, having a numerical argument.

One of the two kinds of timing delays controlled by CTMON is the time between frame updates to the synthesizer. This time is normally set at 10 milliseconds and can be changed using the command "H". Type "H", followed by a 5 (for 5 msec), and then a carriage return. Now type "P" again. This time the playback is at twice the normal speaking rate, with data frames being updated to the synthesizer every 5 msec. Also try H20 and again P, playing back the speech at half speed. This is not like changing the speed of a tape recording because the frequencies of the voice do not change here, only the rate of speaking changes. Remember that any command with an argument must be ended with a return.

The second kind of delay is the length of time between repeats. Type the command "R". You will hear the entire phrase and then after about 1 second pause, the phrase will repeat. This will continue until you hit any other key (like rubout or escape). By typing a "W" followed by the desired wait time in msec, and return, you can change the length of time between repeats up to 65 seconds.

Using the "S" (Start at frame) and "L" (Length to play) commands, you can control the section of data to be played over the synthesizer. Enter the commands S600 return, and L146 return. Now when you type P you hear only the phrase "8080 microcomputer". To find where a certain sound is in the data, enter a length of approximately 50 frames and then change the starting frame by 100's until you hear the piece you want. Sometimes you will hear a thump if the playback starts or ends in the middle of a vowel. That is because of the unusually rapid onset of amplitude to the synthesizer.

MAKE IT SOUND DIFFERENT

Watch the screen as you type the command "AH". You will see the "current parameter" pointer move over to column AH, the amplitude of aspiration parameter. The display should still be positioned at frame 1. If it is not, move to frame 1 by typing the move command M1 return. Observe the column of values beginning at frame 1; 10, 20, 30, 40, ..., and locate this column of data in the listing of the demonstration phrase "HELLO". The values greater than Ø in the AH column represent the amplitude of the hissing sound produced during the h sound, frames 1 thru 8. Set the playback pointers to include just the word "hello", that is, S1 return, L50 return. Now listen carefully to the word "hello" and observe the hissing sound of the h.

You can change the value of the current parameter at the current frame by typing the desired value and a return. Note that typing only a return is the same as Ø return. This is for convenience since Ø is by far the most frequent value you will want to enter, but it is also a bit risky because stray returns seem to pop up every now and then. This would zero a location in your data. Entering a new data value into the buffer has one other effect: it moves you forward to the next frame. Now type a series of returns, at least 8, until you see all Ø's in the AH column. Again play the word "'ello" and notice the Cockney accent with the missing "h". Move back to frame 1, either by typing M1 return or a series of B's. Then reenter a new set of amplitude values, say in the range 80 to 100, to make it much louder. Notice that there is a fairly critical threshhold of amplitude where it becomes too loud and no longer sounds like an h in the word, but just sounds like a hissing noise separated from the sound of the word itself. A smooth decay over 3 or 4 frames instead of a "square corner" also has quite a lot to do with this disassociation effect.

Now experiment with the pitch control, FØ. Type "FØ" to select that parameter and then go down the column from frame 1 setting each frame's FØ value to a constant, say 60. An easy way to do this is by typing a 60 in the usual way for the first frame and then typing the ditto command (") until you have set the desired number of frames. Continue this thru at least frame 44. Now listen to "hello" and hear that it has lost all intonation and is pronounced in a monotone. Experiment with different pitch values and try constucting a new contour for the FØ parameter. The original FØ control for "hello" has a peak around frame 12. Try moving the major peak down to about frame 35. You may find it helpful to sketch the curve you want on graph paper and read off the new values to be entered.

Look around thru the data and try changing other parts to see what effect it has. For example, thru most of the entire demonstration phrase, flattening out the F3 control doesn't make very much difference as long as it's set roughly to a mid-range value. However, if you reduce the value of F3 at frames 583-593, the "r" sound in "your" will disappear. Notice the peaks of F3 at frame 508 in "standard" and at frame 689 in "micro". This is typical of the "r" sound.

The CTMON Control Monitor software system is supplied with the CT-1 Synthesizer board both as ASCII source and assembled and punched in Intel Hex object format. The code was assembled using a modified version* of the popular Microtec/Processor Tech/IMSAI assembler. One characteristic of that system that I am not particularly happy with is that every line must begin with its line number. That takes a lot of paper tape. Also, the lines are formatted nicely with a lot of spaces. Even with the comments in, the actual source code would be about half that much paper if fields were marked with just a single space or tab character. For the most part the syntax and symbols used are those in common use elsewhere. Some things which may be undefined in other assemblers are DT for define text, definition of the full symbols PSW and SP, and the use of multiplication, division and modulus operators in operand expressions.

At the time this software was written, I did not have the facilities to assemble it on my own machine, and had the job done at several different places. That is my excuse for the missing parts and its generally incomplete state of development. In one sense it's a makeshift system anyway, because of the lack of a cheap high-resolution graphics capability, which would allow the editing to be done graphically instead of numerically. Such an editor would be much easier to work with. In any case, I will fully support this software system, which means that I will be sending out update packages to all purchasers. That will include the Insert and Delete command processors in the near future, and of course, patches for any bugs discovered as more people begin using the system.

*PDS-1, modified by Steve Zook, at the Computer Store, Santa Monica, CA

COMPUTALKER CONTROL MONITOR -- COMMAND SUMMARY
FOR VERSION 1.10   DEC 13, 76

INDICATES A DECIMAL NUMBER 0 THRU 65535
<CR> INDICATES A CARRIAGE RETURN

#<CR>	CHANGE VALUE OF CURRENT PARAMETER TO # AND STEP FORWARD ONE FRAME
"	REPEAT LAST #<CR> COMMAND WITH SAME #
AF	OPEN AF FOR EDITING
AH	OPEN AH FOR EDITING
AN	OPEN AN FOR EDITING
AV	OPEN AV FOR EDITING
B	STEP BACK ONE FRAME
CA	APPEND DATA TO BUFFER FROM CASSETTE
CR	READ FROM CASSETTE & REPLACE BUFFER
CW	WRITE BUFFER CONTENTS ONTO CASSETTE
D#<CR>	DELETE # FRAMES, CURRENT FRAMF THRU CURRENT+#-1 *
FØ	OPEN F0 FOR EDITING
F1	OPEN F1 FOR EDITING
F2	OPEN F2 FOR EDITING
F3	OPEN F3 FOR EDITING
FF	OPEN FF FOR EDITING
H#<CR>	HANG # MSEC BETWEEN FRAMES DURING PLAY
I#<CR>	INSERT # FRAMES BEFORE CURRENT FRAME * (CURRENT FRAME IS DUPLICATED # TIMES)
KILL	KILL THE ENTIRE BUFFER CONTENTS
L#<CR>	SET TO PLAY # FRAMES ONLY
M#<CR>	MOVE TO FRAME #
N	STEP FORWARD ONE FRAME
P	PLAY
R	REPEAT PLAY
S#<CR>	SETUP TO START PLAY AT FRAME #
TA	APPEND TO BUFFER FROM PAPER TAPE
TR	READ PAPER TAPE & REPLACE BUFFER CONTENTS
TW	WRITE BUFFER CONTENTS TO PAPER TAPE (PUNCH) * (WILL BE TP IN NEXT VERSION)
TY	LIST BUFFER CONTENTS ON TTY * (WILL BE TL IN NEXT VERSION)
W#<CR>	WAIT # MSEC BETWEEN REPEAT PLAYS
X	EXIT TO USER DEFINED LOCATION *

IN THE NEXT VERSION, THE SYNTAX OF # COMMANDS WILL BE CHANGED TO #M INSTEAD OF THE PRESENT M#<CR>. THE COMMAND WILL BE EXECUTED IMMEDIATELY ON TYPING THE COMMAND LETTER AND WILL NOT WAIT FOR THE <CR>.

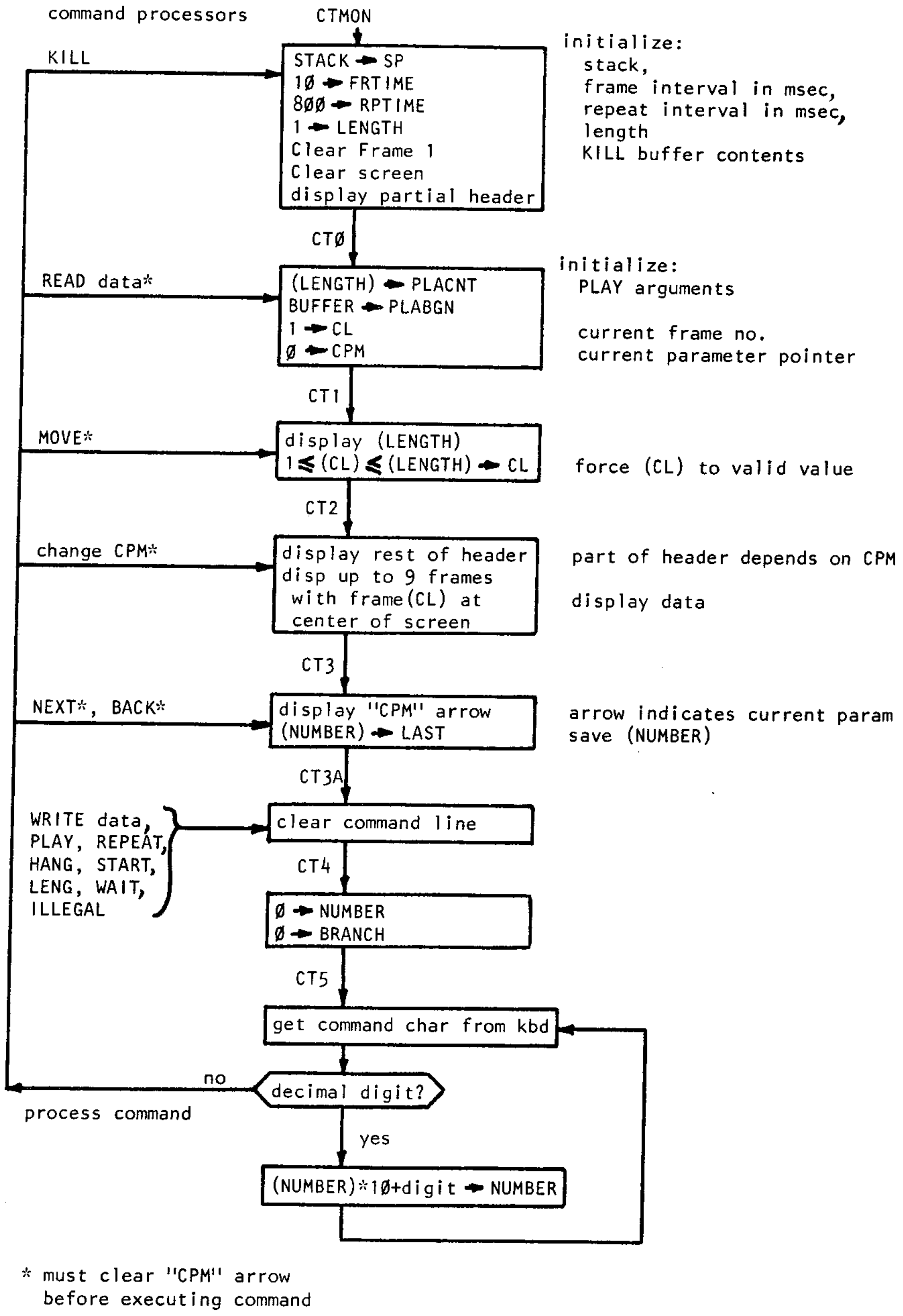
* not implemented
in this version

The Vector Graphics FLASHWRITER is another memory-mapped video system that CTMON will work with. The board uses memory locations D000H thru D3FFH. To patch the standard version of CTMON assembeled at address 0100H, make the following changes:

location	now contains	change to
01BAH	9BH	0DH
01D2H	A0H	20H
059DH	80H	00H
05CDH	80H	00H
0681H	80H	00H
0694H	88H	D0H
0696H	A0H	20H
069AH	8CH	D4H
06ACH	A0H	20H
06EDH	22H	34H
06FCH	80H	00H
0737H	80H	00H
073FH	80H	00H

To patch the special version of CTMON assembeled for Micropolis systems at address 2A00H, make the following changes:

location	now contains	change to
2BBAH	9BH	0DH
2BD2H	A0H	20H
2F9DH	80H	00H
2FCDH	80H	00H
3081H	80H	00H
3094H	88H	D0H
3096H	A0H	20H
309AH	8CH	D4H
30ACH	A0H	20H
30EDH	22H	34H
30FCH	80H	00H
3137H	80H	00H
313FH	80H	00H



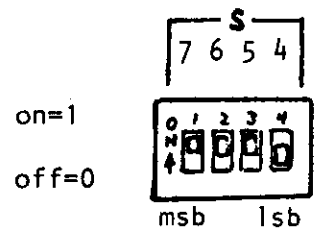
* must clear "CPM" arrow
before executing command

"HOW THE CT-1 INTERFACE WORKS"

During the period of time that Addresses A4-A7 compare with the setting of the DIP-Switch, and both $\overline{\text{PWR}}$ and SOUT are active, the data D0-D7 is strobed into an 8 bit register; also the Address A0-A3 is then strobed into a 4 bit latch. As the data bits are stored, they are also passed to the 8 bit Digital-to-Analog Converter (DAC). When $\overline{\text{PWR}}$ or SOUT go inactive, the voltage from the DAC is strobed into one of 14 sample-and-holds, the destination to be determined by the contents of the 4 bit address latch. Nine of these sample-and-holds are used to control the CT-1 parameters listed below, one is used to turn the CT-1 audio output (which appears on a RCA connector at the top edge of the board) on and off, and 4 are reserved for future additions.

Address A3---AØ	Channel (hex)	Mnemonic	Parameter
Ø Ø Ø Ø	Ø	AV	Voicing Amplitude
Ø Ø Ø 1	1	FØ	Voicing Frequency
Ø Ø 1 Ø	2	F1	Formant 1 Frequency
Ø Ø 1 1	3	F2	Formant 2 Frequency
Ø 1 Ø Ø	4	F3	Formant 3 Frequency
Ø 1 Ø 1	5	AH	Aspiration Amplitude
Ø 1 1 Ø	6	AF	Frication Amplitude
Ø 1 1 1	7	FF	Frication Frequency
1 Ø Ø Ø	8	AN	Nasal Amplitude
1 Ø Ø 1	9	}	Reserved for Future Expansion
1 Ø 1 Ø	A		
1 Ø 1 1	B		
1 1 Ø Ø	C		
1 1 Ø 1	D		not used
1 1 1 Ø	E		not used
1 1 1 1	F	SW	Audio On/Off Switch

Orientation of the output port selector switch
(for serial numbers 031 and up)



Output port numbers are the above selected hex digit followed by the channel number. Hex channel numbers are as follows:

Channel	Mnemonic	Description
Ø	AV	Amplitude of Voicing
1	FØ	Frequency of Voicing (fundamental)
2	F1	Formant 1 Frequency
3	F2	Formant 2 Frequency
4	F3	Formant 3 Frequency
5	AH	Amplitude of Aspiration (hiss)
6	AF	Amplitude of Frication
7	FF	Frequency of Frication
8	AN	Amplitude of Nasal
9		reserved for future expansion
A		
B		
C		
D		
E		
F	SW	Audio on-off Switch

Your Model CT-1 was shipped with the selector switch set to accept data on ports EØ - EF, as shown above. The Control Monitor software system has been assembled to output data on the same ports. To modify the code to output data on another set of ports, redefine the variable CTBASE to the desired address of the parameter AV.

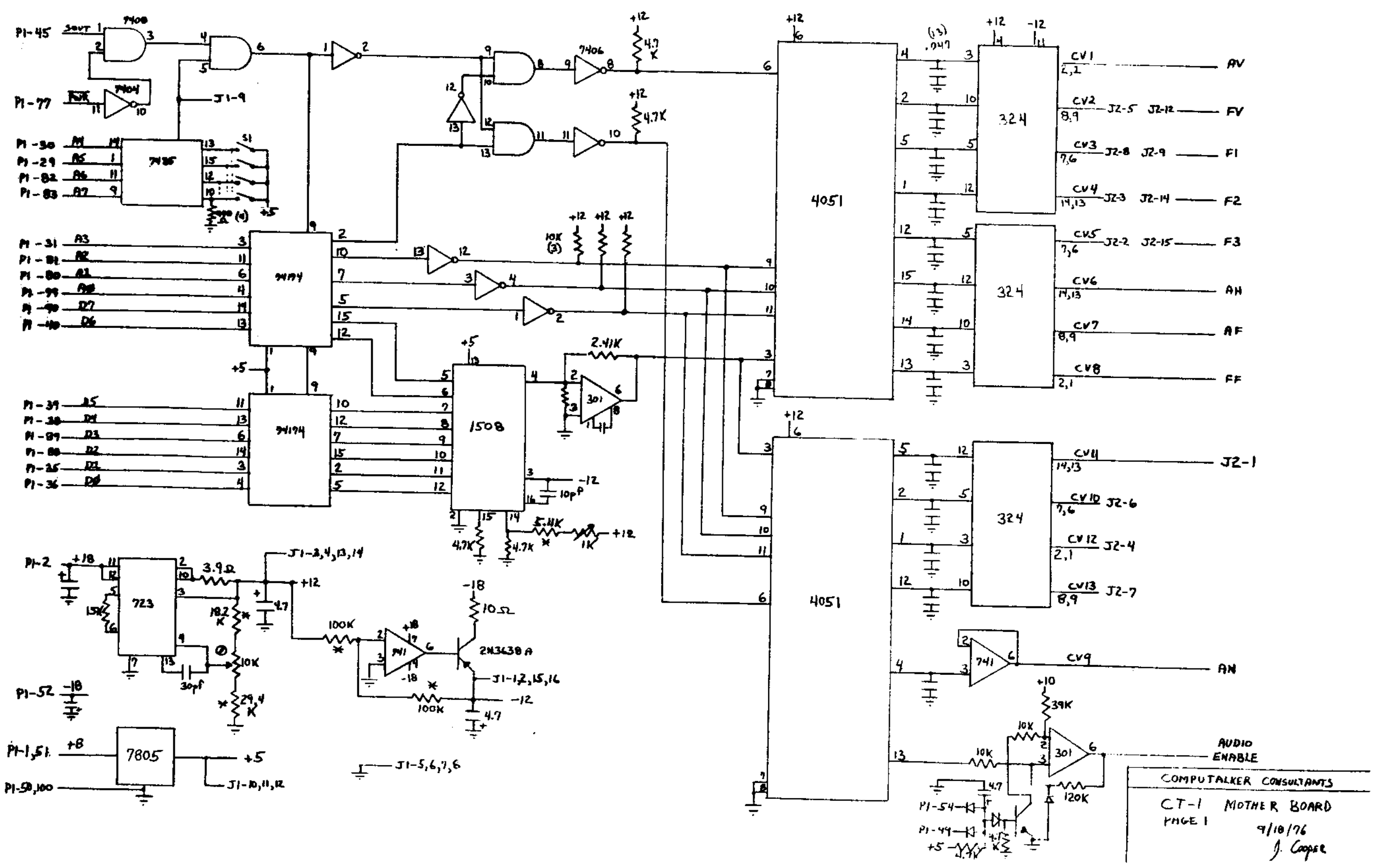
COMPUTALKER CT-1 BUS REQUIREMENTS

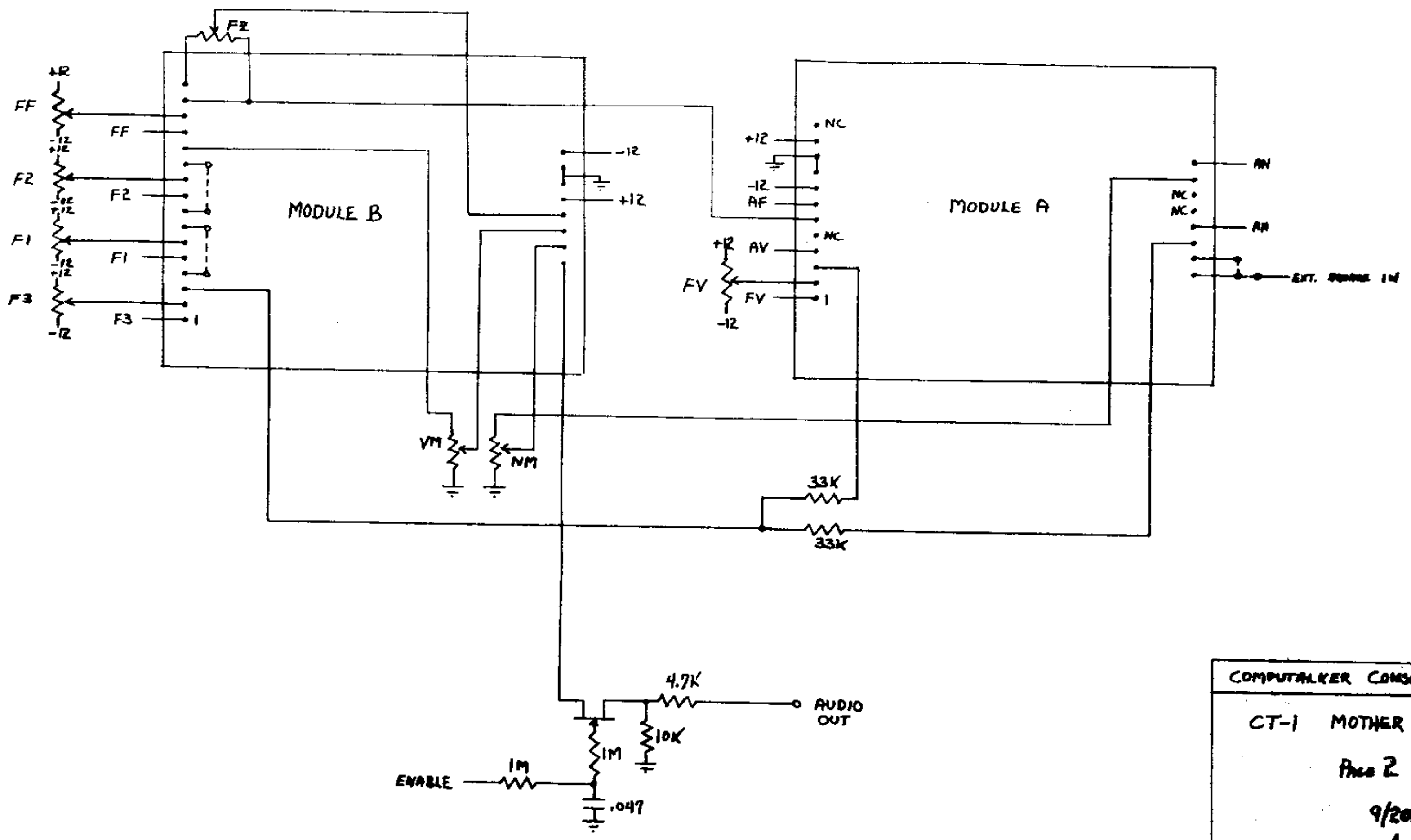
Following are the 8800-type (S-100) Bus signals used by the CT-1 Speech Synthesizer:

PIN	NAME	ACTIVE	DESCRIPTION	
36	D0	HIGH	LSB	8 bit parameter data from computer to CT-1.
35	D1	"		
88	D2	"		
89	D3	"		
38	D4	"		
39	D5	"		
40	D6	"		
90	D7	"	MSB	
79	A0	HIGH	LSB	4 bit address from computer selects 1 of 16 possible parameter outputs to be updated.
80	A1	"		
81	A2	"		
31	A3	"	MSB	
30	A4	HIGH	LSB	4 bit address from computer selects which block of 16 output ports is used. This address is compared against the on-board DIP switch to enable transfer of data.
29	A5	"		
82	A6	"		
83	A7	"	MSB	
77	$\overline{\text{PWR}}$	LOW		These two signals in active state allow transfer of data from computer to CT-1.
45	SOUT	HIGH		

COMPUTALKER CT-1 BUS REQUIREMENTS (cont.)

PIN	NAME	ACTIVE	DESCRIPTION
54	$\overline{\text{EXT CLR}}$	LOW	When either of these signals is brought to its active level, the on-board audio switch is turned off.
99	$\overline{\text{POC}}$	LOW	
1,51	+8V		These three un-regulated voltages are supplied by the computer (or external supply) to the CT-1. If desired, +5 Volts may be used instead of +8V by a simple change on the board.
50,100	Ground		
2	+16V		
52	-16V		


COMPUTALKER CONSULTANTS
CT-1 MOTHER BOARD
PAGE 1
9/18/76
J. Cooper
AUDIO ENABLE
4051
324
74174
1508
723
7805
741
301
2N3638A
AV
FV
F1
F2
F3
AH
AF
FF
J2-1
AN
P1-45
P1-77
J1-9
S1



COMPUTALKER CONSULTANTS

CT-1 MOTHER BOARD

Page 2

9/20/76

J. Cooper

Five of the nine synthesizer control parameters have a frequency-related control function. These are Frequency of Voicing (FØ), Frequency of Formant 1 (F1), Frequency of Formant 2 (F2), Frequency of Formant 3 (F3), and Frequency of Frication (FF). Of these, Frequency of Voicing (FØ) moves as a direct exponential function of the control value, while the other four, F1, F2, F3, and FF, move as an inverse exponential function of their control values.

If C represents the control value as sent on the computer bus for one of these frequency control parameters, and F represents the resulting frequency in hz, the following relations hold between C and F. To compute the frequency as a function of the control value, use the equation

$$F = A * \mathrm{EXP}(B * C)$$

and to determine the control value needed to produce a desired frequency, use the equation

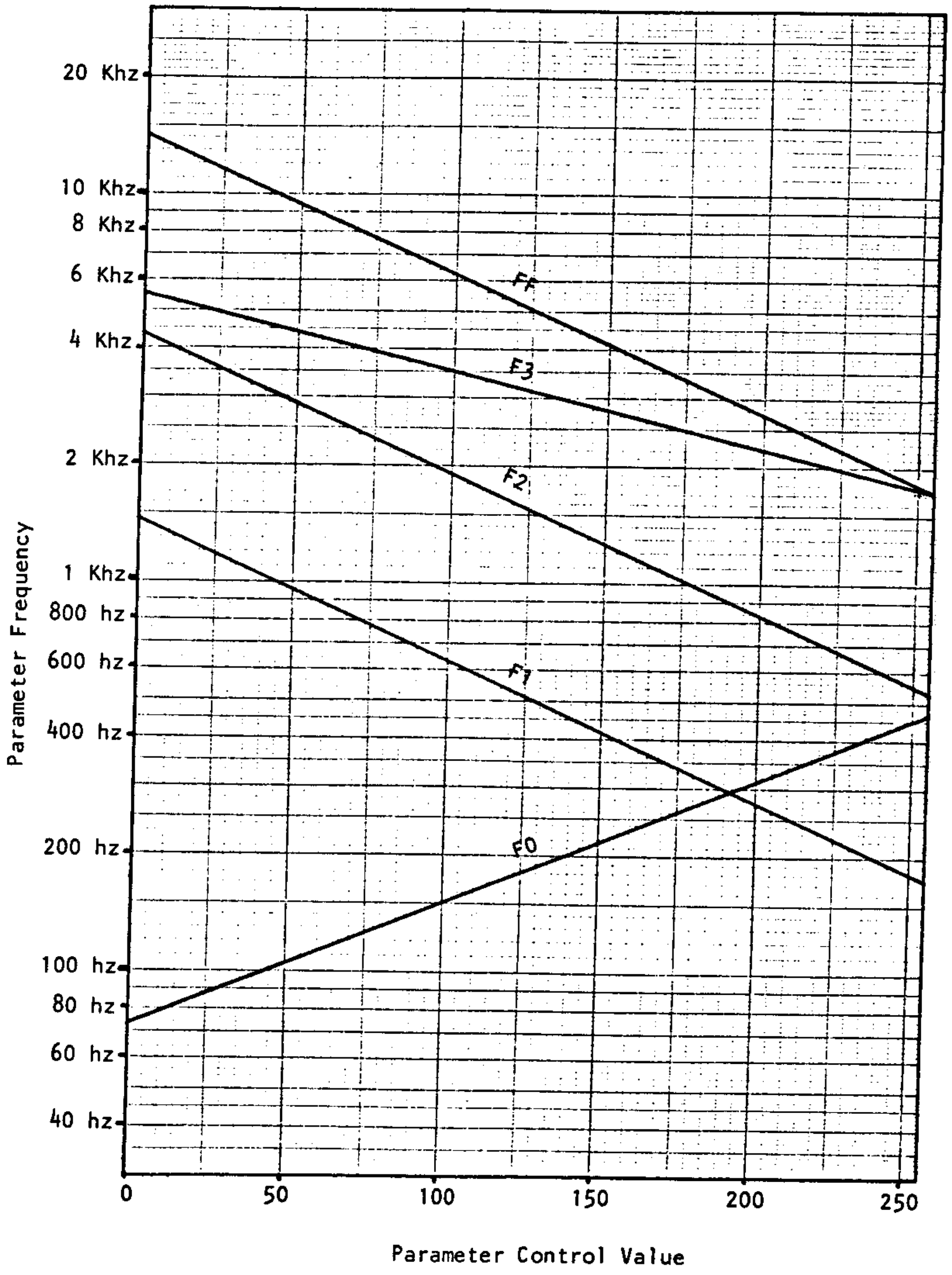
$$C = \frac{\mathrm{LOG}_e(F/A)}{B}.$$

Values for the constants A and B for the five parameters are:

	A	B
FØ	73.4	.00722
F1	1452	-.0083
F2	4356	-.0083
F3	5508	-.0046
FF	14160	-.0083

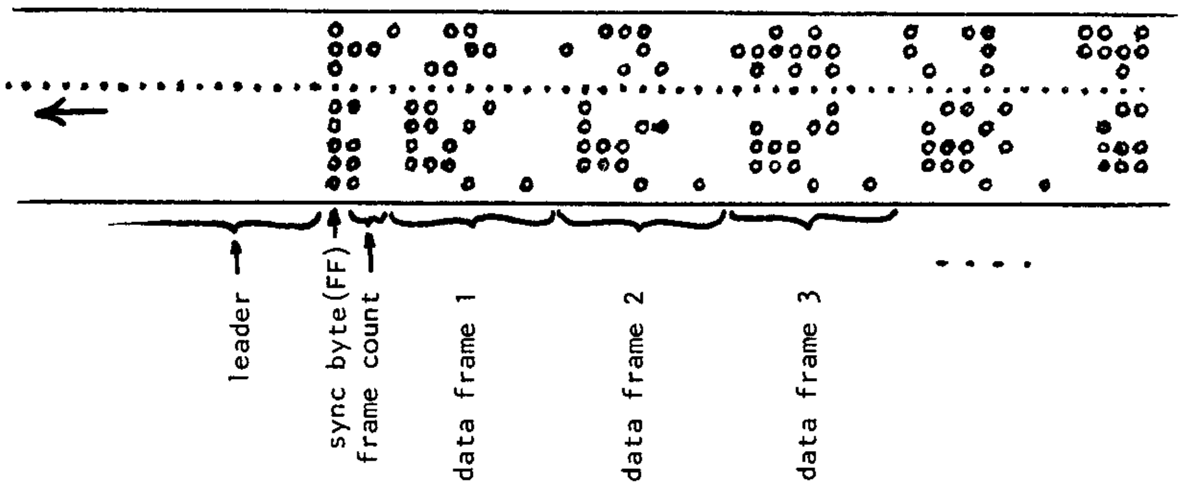
These conversion functions are also given in graphical form on the next page.

Frequency Conversion for Parameters F0, F1, F2, F3, and FF



The CT-1 speech parameter data format

Paper tape parameter data punched for or by the Control Monitor, CTMON, has the following format:



The frame count is punched low order byte first. In this example, the frame count is Ø2EA hex, or 746 decimal frames.

Each data frame consists of 9 bytes with the parameter values in the order AV, FØ, F1, F2, F3, AH, AF, FF, and AN. There is no restriction on the value of any parameter within the range Ø-255.

DATA RATE:

Within each frame, the 9 individual parameters should be sent out to the CT-1 using the appropriate OUT address. At least 20 microseconds should be allowed between individual parameter updates. This time is needed for the sample-and-hold capacitor to be fully charged to the new value in the worst case of a sweep to the opposite extreme value.

The data is coded so that the time between frame refreshes for a normal speech rate is 10 milliseconds. The CT-1 board may be updated with new frame data at any desired rate, of course, to vary the rate of speech. At an update rate faster than approx. 2 msec per frame, the speech looses intelligibility. The maximum length of time between frame updates is approx. 50 to 100 milliseconds, at which time the sample-and-hold capacitors cannot hold the value steady.

"HELLO"

FRAME	phonetic symbol	AV	F0	F1	F2	F3	AH	AF	FF	AN	
1		1	120	92	101	147	10	0	128	0	
2		2	120	97	101	147	20	0	128	0	
3		2	118	99	102	147	30	0	128	0	
4	h	3	116	104	105	146	40	0	128	0	H
5		3	115	110	107	151	40	0	128	0	
6		11	115	115	110	155	30	0	128	0	
7		76	116	107	115	156	20	0	128	0	
8		117	116	101	120	158	10	0	128	0	
9		149	119	102	127	160	0	0	128	0	
10		159	123	104	137	159	0	0	128	0	
11		155	124	109	149	156	0	0	128	0	
12		146	124	119	159	155	0	0	128	0	
13		138	123	132	170	159	0	0	128	0	
14		127	113	142	176	162	0	0	128	0	
15	ɛ	101	101	147	182	164	0	0	128	0	e
16		122	85	141	179	162	0	0	128	0	
17		149	76	119	174	160	0	0	128	0	
18		167	64	110	167	157	0	0	128	0	
19		175	59	104	165	158	0	0	128	0	
20		178	56	101	160	159	0	0	128	0	
21		179	53	99	158	160	0	0	128	0	
22		179	51	98	158	164	0	0	128	0	
23		178	48	97	156	167	0	0	128	0	
24		177	48	98	157	165	0	0	128	0	
25		176	48	101	160	167	0	0	128	0	l
26	l	174	48	103	163	166	0	0	128	0	
27		172	48	105	165	166	0	0	128	0	l
28		169	48	106	166	164	0	0	128	0	
29		167	49	108	171	165	0	0	128	0	
30		164	50	110	175	166	0	0	128	0	
31		163	52	112	180	164	0	0	128	0	
32		162	54	114	182	164	0	0	128	0	
33		162	58	117	183	162	0	0	128	0	
34		162	63	119	184	163	0	0	128	0	
35		162	70	121	185	161	0	0	128	0	
36		161	75	123	185	159	0	0	128	0	
37	o	158	80	124	184	158	0	0	128	0	
38		152	87	126	184	158	0	0	128	0	
39		141	93	132	183	160	0	0	128	0	o
40		106	96	139	183	163	0	0	128	0	
41	w	76	98	151	183	168	0	0	128	0	
42		57	100	164	182	173	0	0	128	0	
43		29	100	174	180	177	0	0	128	0	
44		12	101	180	178	178	0	0	128	0	
45		0	102	183	176	178	0	0	128	0	,
46		0	102	187	175	178	0	0	128	0	
47		0	102	188	173	178	0	0	128	0	
48		0	101	189	171	178	0	0	128	0	
49		0	100	186	170	178	0	0	128	0	
50		0	99	185	167	178	0	0	128	0	
51		0	98	182	165	178	0	0	128	0	
52		0	96	180	163	178	0	0	128	0	
53		0	93	177	162	178	0	0	128	0	
54		0	91	172	161	178	0	0	128	0	
55		0	89	169	160	178	0	0	128	0	

```
 56            0      86     165     159     178       0       0     128       0
 57            0      84     161     158     178       0       0     128       0
 58            0      80     155     157     178       0       0     128       0
 59            0      77     150     156     178       0       0     128       0
 60            0      73     145     156     178       0       0     128       0
 61            0      69     138     156     177       0       0     128       0
 62            0      66     132     155     176       0       0     128       0
 63            0      63     126     155     174       0       0     128       0
 64            0      61     119     155     172       0       0     128       0
 65            0      58     113     155     170       0       0     128       0
 66            0      55     106     155     167       0       0     128       0
 67            0      52      99     155     163       0       0     128       0
 68            0      49      91     155     159       0       0     128       0
 69            0      46      85     155     154       0       0     128       0
 70            0      44      81     155     148       0       0     128       0
 71            0      40      74     155     144       0       0     128       0
 72            0      39      68     155     140       0       0     128       0
 73            0      37      65     155     137       0       0     128       0
 74            6      37      62     157     135       0       0     128       0
 75           14      42      62     157     135       0       0     128       0
 76           30      50      61     156     135       0       0     128       0
 77           63      54      64     156     136       0       0     128       0
 78          118      62      62     154     141       0       0     128       0
 79          156      71      61     151     143       0       0     128       0
 80          183      80      62     147     146       0       0     128       0
 81          200      87      60     141     147       0       0     128       0
 82    a     208      90      60     137     149       0       0     128       0
 83          212      95      62     132     151       0       0     128       0
 84          214     100      63     125     159       0       0     128       0  I'
 85          214     104      69     117     160       0       0     128       0
 86          213     107      87     111     158       0       0     128       0
 87          211     111     101     107     154       0       0     128       0
 88          205     114     125     104     152       0       0     128       0
 89          194     116     142     102     154       0       0     128       0
 90          176     118     160     101     160       0       0     128       0
 91    l     146     119     177     100     166       0       0     128       0
 92          137     118     176      98     166       0       0     128      26
 93          133     115     177      96     168       0       0     128      61
 94          127     114     177      94     170       0       0     128      91
 95          125     114     178      91     172       0       0     128     106
 96          122     114     178      89     173       0       0     128     111
 97          120     114     179      87     174       0       0     128     114
 98          116     113     178      85     175       0       0     128     114
 99    m     110     112     178      83     178       0       0     128     115
100          101     109     181      81     182       0       0     128     112  m
101           76      96     184      80     187       0       0     128     103
102           53      84     196      80     193       0       0     128      89
103           21      95     224      85     198       0       0     150      57
104            0     101     214      93     200       0       0     200      24
105            0     104     193     103     199       0       0     250       0
106    k       0     108     170     116     196       0       0     250       0
107  RBL →    34     110     142     122     191      35      40     250       0
108            0     112     123     127     186      30      30     250       0  C
109            0     113     111     131     180      30      20     250       0
110            0     113      97     135     175      30      10     250       0
111            0     113      86     139     172      30       0     250       0
112           13     111      78     140     167      20       0     250       0
113           68     111      73     142     163       0       0     250       0
```

114		190	114	70	145	159	0	0	200	0	
115		195	117	66	147	155	0	0	200	0	
116		198	125	62	149	152	0	0	200	0	
117	a	199	132	66	151	148	0	0	200	0	
118		196	134	76	152	146	0	0	128	0	o
119		190	137	87	146	149	0	0	128	0	
120		164	139	109	140	148	0	0	128	0	
121		129	139	137	136	149	0	0	128	0	
122		70	142	163	134	160	0	0	128	26	
123		90	144	167	133	172	0	0	128	53	
124	m	94	144	162	132	185	0	0	128	75	m
125		90	141	173	137	192	0	0	128	80	
126		71	128	198	139	189	0	0	128	82	
127		26	117	220	140	181	0	0	128	74	
128		4	116	215	135	160	0	0	128	54	
129		17	112	167	128	153	0	0	128	22	
130	p REL →	31	109	99	121	149	30	30	128	0	
131		29	104	88	112	147	25	0	128	0	
132		18	101	86	106	148	20	0	128	0	p
133		5	97	163	100	151	15	0	128	0	
134	y	92	91	160	98	159	10	0	128	0	
135		116	81	153	100	161	0	0	128	0	
136		155	77	154	105	166	0	0	128	0	
137		141	76	156	109	170	0	0	128	0	
138	@	113	73	157	112	173	0	0	128	0	u
139		92	70	165	115	168	0	0	128	0	
140		15	71	214	114	163	0	0	128	0	
141		7	70	217	113	160	0	0	128	0	
142		2	69	176	110	158	0	0	128	0	
143		1	68	79	109	156	0	0	128	0	
144	REL →	30	67	163	110	160	30	30	128	0	
145	t	22	67	118	111	166	25	0	128	0	t
146		12	67	109	116	172	20	0	128	0	
147		17	68	55	121	178	15	0	128	0	
148		15	67	72	131	182	10	0	128	0	
149		53	65	131	141	182	0	0	128	0	
150		76	65	107	152	179	0	0	128	0	
151		130	63	97	160	177	0	0	128	0	
152		226	60	93	166	175	0	0	128	0	
153		227	58	94	170	172	0	0	128	0	
154		213	57	90	174	169	0	0	128	0	a
155	ɔ	213	54	88	175	173	0	0	128	0	
156		208	51	90	174	175	0	0	128	0	
157		164	48	94	173	177	0	0	128	0	
158		149	43	100	169	180	0	0	128	0	l
159		99	40	105	169	184	0	0	128	0	
160		88	38	111	165	185	0	0	128	0	
161		47	37	123	163	187	0	0	128	0	
162		17	35	145	162	189	0	0	128	0	
163		9	34	150	159	185	0	0	128	0	
164		2	33	157	157	179	0	0	128	0	
165		1	31	147	151	179	0	0	128	0	
166		1	29	130	144	180	0	0	128	0	
167		1	26	109	136	183	0	0	128	0	
168		1	23	88	114	189	0	0	128	0	k
169	K REL →	44	22	80	114	192	30	30	128	0	
170		11	19	141	113	215	25	0	128	0	
171		35	19	142	115	222	20	0	128	0	

```
172          43      18      130      120      232      15       0      128       0
173          44      16      124      124      237      10       0      128       0
174          44      14      123      124      239       0       0      128       0
175          43      11      123      124      242       0       0      128       0
176          40      11      131      125      246       0       0      128       0
177          37       7      135      125      245       0       0      128       0
178          32       5      136      124      247       0       0      128       0
179          29       5      134      122      246       0       0      128       0
180          22       5      136      123      246       0       0      128       0
181          17       5      136      125      247       0       0      128       0
182           7       6      135      126      255       0       0      128       0
183           4       7      136      125      249       0       0      128       0
184           0       8      135      126      250       0       0      128       0
185           0       9      130      127      239       0       0      128       0
186           0      10      120      128      240       0       0      128       0
187           0      12      120      128      240       0       0      128       0
188           0      12      120      128      240       0       0      128       0
189           0      13      120      128      240       0       0      128       0
190           0      14      120      128      240       0       0      128       0
191           0      16      120      128      240       0       0      128       0
192           0      17      120      128      240       0       0      128       0
193           0      18      120      128      240       0       0      128       0
194           0      19      120      128      240       0       0      128       0
195           0      20      120      128      240       0       0      128       0
196           0      21      120      128      240       0       0      128       0
197           0      22      120      128      240       0       0      128       0
198           0      22      120      128      240       0       0      128       0
199           0      25      120      128      240       0       0      128       0
200           0      25      120      128      240       0       0      128       0
201           0      26      120      128      240       0       0      128       0
202           0      26      120      128      240       0       0      128       0
203           0      27      120      128      240       0       0      128       0
204           0      28      120      128      240       0       0      128       0
205           0      28      120      128      240       0       0      128       0
206           0      29      120      128      238       0       0      128       0
207           0      30      120      128      236       0       0      128       0
208           0      30      120      128      234       0       0      128       0
209           0      30      120      128      231       0       0      128       0
210           0      30      120      128      229       0       0      128       0
211           0      30      120      128      226       0       0      128       0
212           0      30      120      129      224       0       0      128       0
213           0      30      120      131      220       0       0      128       0
214           0      30      120      132      218       0       0      128       0
215           0      30      120      134      216       0       0      128       0
216           0      30      120      136      212       0       0      128       0
217           0      30      120      138      208       0       0      128       0
218           0      30      120      140      205       0       0      128       0
219           0      30      120      141      201       0       0      128       0
220           0      30      120      144      198       0       0      128       0
221           0      30      120      146      195       0       0      128       0
222           0      30      120      149      191       0       0      128       0
223           0      30      120      151      187       0       0      128       0
224           0      31      120      153      184       0       0      128       0
225           0      32      120      153      180       0       0      128       0
226           0      33      120      154      176       0       0      128       0
227           0      35      110      155      173       0       0      128       0
228           0      40      100      152      170       0       0      128       0
229           8      45       90      150      167       0       0      128       0
```

230		73	50	85	147	164	0	0	128	0	
231		88	52	88	143	163	0	0	128	0	
232		113	54	83	140	161	0	0	128	0	
233	ə	113	54	93	137	160	0	0	128	0	a
234		100	56	112	134	159	0	0	100	0	
235		80	61	124	128	158	0	0	80	0	
236		39	63	142	121	156	0	20	80	0	
237		17	69	84	114	156	0	30	80	0	
238		0	73	86	113	156	0	50	80	0	
239		0	78	84	109	155	0	50	80	0	
240		0	83	90	105	154	0	50	80	0	
241	s	0	87	77	103	153	0	50	80	0	s
242		0	91	67	102	154	0	50	80	0	
243		0	95	68	100	154	0	40	80	0	
244		0	98	82	99	153	0	40	80	0	
245		0	104	74	99	154	0	40	80	0	
246		0	108	75	100	153	0	40	80	0	
247		0	113	83	105	153	0	30	80	0	
248		0	117	81	109	153	0	20	80	0	
249		0	121	82	119	153	0	0	80	0	
250		0	125	84	139	154	0	0	80	0	
251		0	130	79	142	155	0	0	128	0	
252		0	134	80	142	158	0	0	128	0	
253		0	136	85	142	159	0	0	128	0	
254		0	138	98	139	161	0	0	128	0	
255	p	0	139	121	134	164	0	0	128	0	
256	$\xrightarrow[\text{REL}]{}$	38	139	186	127	163	0	50	128	0	p
257		176	139	172	79	146	0	0	128	0	
258		186	142	169	71	137	0	0	128	0	
259		190	144	180	68	136	0	0	128	0	
260		192	146	194	65	136	0	0	128	0	
261	i	195	150	204	63	134	0	0	128	0	
262		195	151	212	64	143	0	0	128	0	e
263		189	151	214	68	139	0	0	128	0	
264		177	151	218	70	139	0	0	128	0	e
265		158	150	215	72	138	0	0	128	0	
266		81	151	225	79	136	0	0	150	0	
267		47	150	246	81	133	0	0	150	0	
268		29	149	247	83	136	0	0	150	0	
269	tʃ	11	147	152	84	136	40	30	150	0	c
270		3	145	125	86	140	40	30	150	0	
271		0	142	112	88	139	40	30	150	0	h
272		0	139	106	91	144	40	30	150	0	
273		0	137	104	94	147	40	30	150	0	
274		0	134	102	97	147	40	30	150	0	
275		0	133	102	98	147	30	30	125	0	
276		0	130	102	100	144	30	40	100	0	
277		0	128	102	107	144	20	40	100	0	
278	s	0	127	108	105	143	20	40	100	0	
279		0	125	113	105	138	20	40	100	0	s
280		0	124	122	110	147	20	40	100	0	
281		3	124	127	108	145	20	40	100	0	
282		16	125	178	106	146	20	30	100	0	
283		180	125	167	109	153	0	0	100	0	
284		225	125	163	106	153	0	0	100	0	
285	ɪ	219	125	158	106	151	0	0	120	0	y
286		208	124	156	107	145	0	0	128	30	
287		194	123	179	110	143	0	0	128	40	

288		170	122	167	113	148	0	0	128	50	
289	n	151	118	189	113	152	0	0	128	100	
290		132	114	186	113	152	0	0	128	100	n
291		110	101	194	111	153	0	0	128	100	
292		85	102	200	109	155	0	0	128	50	
293		54	104	220	108	158	0	0	200	0	
294		25	105	219	107	155	0	0	200	0	
295		5	106	162	109	150	0	0	200	0	
296		0	105	86	112	144	0	50	200	0	t
297	θ	0	104	116	117	140	0	50	200	0	
298		0	102	97	114	142	0	0	200	0	h
299		10	96	151	115	145	0	0	200	0	
300		90	91	149	122	159	0	0	200	0	
301		153	81	147	123	159	0	0	100	0	
302	ə	152	73	152	125	156	0	0	100	0	e
303		136	67	154	127	158	0	0	100	0	
304		91	57	153	125	158	0	20	100	0	
305		15	57	158	116	153	20	40	100	0	
306		2	58	153	115	141	30	50	100	0	
307		0	60	107	113	137	30	50	100	0	
308		0	62	84	114	143	30	50	100	0	
309	s	0	63	60	112	144	30	50	100	0	
310		0	67	90	112	149	20	50	100	0	s
311		0	67	86	113	151	10	40	100	0	
312		0	75	68	113	154	0	30	100	0	
313		0	83	81	115	155	0	20	100	0	
314		10	85	149	139	161	0	0	100	0	
315		100	80	134	140	163	0	0	100	0	
316		170	74	118	141	165	0	0	100	0	
317	a	178	67	103	143	167	0	0	100	0	
318		220	60	86	148	171	0	0	128	0	
319		243	58	74	150	180	0	0	128	0	
320		253	57	75	146	166	0	0	128	0	
321		255	55	74	147	162	0	0	128	0	
322		227	55	73	145	163	0	0	128	0	i
323		212	54	73	141	163	0	0	128	0	
324		187	51	73	135	165	0	0	128	0	
325	ɩ	165	49	75	129	169	0	0	128	0	
326		141	46	79	126	174	0	0	128	0	
327		115	43	86	117	175	0	0	128	0	
328		102	39	96	118	171	0	0	128	0	
329		97	38	117	115	166	0	0	128	0	
330		91	39	120	113	162	0	0	128	0	
331		83	35	132	113	165	0	0	128	0	
332		78	33	146	115	169	0	0	128	0	
333		72	25	154	114	172	0	0	100	0	
334		53	17	164	118	175	0	0	100	0	
335		29	11	180	121	178	0	0	100	0	
336		20	6	188	120	179	20	20	100	0	
337		20	13	190	121	176	30	40	100	0	
338		20	22	185	120	174	30	40	100	0	
339	z	20	29	177	119	170	30	40	100	0	
340		20	35	177	119	167	30	40	100	0	z
341		20	40	170	120	166	30	40	100	0	
342		28	40	165	120	169	20	30	100	0	
343		36	39	165	121	174	0	20	100	0	
344		44	35	157	123	180	0	0	100	0	
345		52	32	155	124	196	0	0	100	0	

```
346          50    27   150   127   240    0    0   100     0
347          50    27   152   128   252    0    0   128     0
348          54    27   151   130   255    0    0   128     0
349   ɚ      69    27   157   130   255    0    0   128     0  e
350          67    23   162   131   255    0    0   128     0
351          56    22   167   132   255    0    0   128     0  r
352          47    21   181   128   255    0    0   128     0
353          45    20   186   125   255    0    0   128     0
354          28    20   182   119   247    0    0   128     0
355          30    17   204   114   231    0    0   128     0
356          22    17   216   106   174    0    0   128     0
357          20    17   214   107   166    0    0   128     0
358          16    22   211   110   164    0    0   128     0
359          15    26   207   109   158    0    0   128     0
360          12    32   208   113   159    0    0   128     0
361          14    41   212   116   159    0    0   128     0
362          12    49   196   117   160    0    0   128     0
363   d      22    52   181   112   160    0    0   128     0  d
364  REL →   29    54   172   109   157   50   70   128     0
365          57    55   173   113   157    0    0   128     0
366          82    57   169   119   159    0    0   128     0
367   l      93    56   168   121   162    0    0   128     0
368          79    54   169   123   165    0    0   128     0  e
369          55    49   173   124   169    0   20   128     0
370          27    49   188   124   169    0   50   128     0
371          13    53   218   124   170    0   60   128     0
372   z      10    60   240   123   169    0   80   128     0  s
373          10    69   219   120   166    0   80   128     0
374          10    76   179   117   168    0   80   128     0
375          22    81   150   119   164    0   50   128     0
376          19    84   135   128   157    0   20   128     0
377         131    85   125   135   159    0    0   128     0
378         168    84   111   138   161    0    0   128     0
379         200    84    94   142   161    0    0   128     0
380         210    85    84   143   158    0    0   128     0  i
381   a     210    88    80   145   161    0    0   128     0
382         208    92    76   146   162    0    0   128     0
383         207    93    72   145   158    0    0   128     0
384         204    97    67   144   155    0    0   128     0
385         200   100    76   144   159    0    0   128     0
386         194   101    79   143   160    0    0   128     0
387         187   105    79   139   157    0    0   128     0
388         179   105    81   135   157    0    0   128     0
389         173   106    89   133   157    0    0   128     0
390   l     166   105   100   121   156    0    0   128     0
391         158   105   112   112   152    0    0   128     0
392         150   104   124   110   153    0    0   128     0  g
393         142   101   137   102   153    0    0   128     0
394         135   101   155    99   157    0    0   128     0
395         124   100   175    97   166    0    0   128    50
396         112    96   192    97   187    0    0   128   100  n
397          95    88   195    98   191    0    0   128   100
398          76    83   199    98   189    0    0   128   100
399          51    79   201   100   179    0    0   128   100
400  n(d)    29    78   205   101   172    0    0   128    50
401          16    77   213   104   158    0    0   128     0  e
402           6    78   200   107   144    0    0   128     0
403  REL →    1    76   181   108   142   30   40   128     0  d
```

404	t	16	76	164	110	148	25	0	128	0	
405		78	75	153	115	158	20	0	128	0	
406		146	73	146	121	159	15	0	128	0	t
407	ə	149	69	142	136	169	10	0	128	0	
408		185	67	149	141	176	0	0	128	0	o
409		29	68	163	143	176	0	0	128	0	
410		10	69	189	144	173	0	0	128	0	
411		3	71	157	144	170	0	0	128	0	
412		0	73	65	144	171	0	0	128	0	
413	p	0	74	56	142	173	0	0	128	0	
414		0	75	64	142	178	0	0	200	0	
415	→	0	77	37	154	182	30	40	200	0	p
416	REL	0	79	107	150	186	25	0	200	0	
417		0	82	110	136	189	20	0	200	0	
418		0	83	123	128	189	15	0	200	0	
419		0	85	101	128	182	10	0	200	0	
420		0	88	101	144	172	0	0	200	0	
421		0	90	146	155	158	0	0	200	0	
422		62	92	159	169	155	0	0	128	0	
423	l	161	92	123	162	153	0	0	128	0	
424		191	90	109	163	155	0	0	128	0	
425		209	92	105	150	152	0	0	128	0	l
426		210	92	105	145	158	0	0	128	0	
427		211	93	105	142	161	0	0	128	0	
428		211	94	110	135	160	0	0	128	0	
429		209	91	114	130	160	0	0	128	0	
430		207	90	118	126	162	0	0	128	0	
431	ʌ	202	92	140	114	163	0	0	128	0	
432		193	92	163	103	170	0	0	128	0	u
433		72	92	164	94	176	0	0	128	0	
434		38	93	205	88	175	0	0	128	0	
435		34	93	203	82	175	0	0	128	0	
436	g	28	93	211	79	173	0	0	128	0	
437		18	95	192	77	173	0	0	128	0	g
438	→	55	94	177	78	172	0	50	128	0	
439	REL	134	92	170	80	165	0	0	128	0	
440		174	91	157	85	163	0	0	128	0	
441	ı	193	89	147	88	160	0	0	128	0	
442		199	88	147	90	159	0	0	128	0	
443		200	88	152	93	158	0	0	128	0	i
444		195	88	167	100	158	0	0	128	0	
445		187	85	196	103	157	0	0	128	0	
446		168	83	198	105	157	0	0	128	0	
447	n	143	80	198	107	161	0	0	128	0	
448		112	78	195	108	162	0	0	128	0	n
449		69	76	190	108	163	0	0	128	0	
450		36	75	213	108	162	0	0	128	0	
451		11	76	185	107	157	0	0	128	0	
452	t	0	76	152	107	151	0	0	128	0	
453	→	0	75	139	106	146	30	40	128	0	t
454	REL	16	73	175	107	153	25	0	128	0	
455		71	69	156	113	158	20	0	128	0	
456		120	64	149	115	160	15	0	128	0	
457	ə	165	58	147	117	165	10	0	128	0	
458		159	54	151	121	170	0	0	128	0	o
459		133	49	154	127	168	0	0	128	0	
460		63	43	182	130	172	0	0	128	0	
461		32	35	221	132	180	0	0	170	0	

```
462        28    32   217   125   169     0     0   170     0
463        23    43   223   117   168     0     0   170     0
464   ð    21    48   255   116   172     0    10   170     0  t
465        22    52   170   117   170     0    20   170     0
466        41    53   164   117   172     0    30   160     0  h
467        71    56   154   122   170     0    30   140     0
468        73    58   154   121   168    20    30   130     0
469   ə    70    61   151   117   168    30    40   120     0  e
470        13    63    75   113   167    30    40   120     0
471        10    63    43   113   169    20    40   120     0
472        13    63    31   105   164    10    40   120     0
473   s    13    64    30   100   167     0    40   120     0
474         8    66    37   100   156     0    30   120     0  s
475         8    66    41   100   147     0    20   120     0
476         6    67    44   101   155     0     0   120     0
477         4    70    33   106   165     0     0   120     0
478         3    72    39   105   157     0     0   128     0
479         3    73    47   107   153     0     0   128     0
480         3    75    55   106   151     0     0   128     0
481         3    77    67   109   144     0     0   128     0
482   t     2    77    78   107   140     0     0   128     0  t
483   →    13    78   109   104   143    30    40   128     0
484        34    79   141   106   158    20     0   128     0
485        64    77   135   112   166     0     0   128     0
486       152    75   116   112   169     0     0   128     0
487       171    74   101   111   169     0     0   128     0
488   æ   180    73    90   110   168     0     0   128     0  a
489       183    74    84   110   165     0     0   128     0
490       183    74    86   111   162     0     0   128     0
491       182    77    90   110   160     0     0   128     0
492       179    80    89   107   162     0     0   128     0
493       176    81   107   105   160     0     0   128     0
494       170    84   121   104   157     0     0   128     0
495       163    84   135   107   156     0     0   128     0
496       154    83   151   109   157     0     0   128     0
497       143    81   200   109   158     0     0   128     0  n
498   n̂d  135    81   201   109   160     0     0   128     0
499       109    80   204   109   165     0     0   128     0
500        78    77   207   109   171     0     0   128     0
501        50    74   208   109   179     0     0   128     0  d
502        38    72   195   109   180     0     0   128     0
503        50    71   162   110   185     0     0   128     0
504        88    72   157   112   194     0     0   128     0
505       123    71   151   113   206     0     0   128     0
506       171    69   148   114   221     0     0   128     0
507   ə   166    67   147   115   234     0     0   128     0  a
508       154    66   151   117   242     0     0   128     0
509       120    61   160   118   242     0     0   128     0  r
510        91    58   184   111   235     0     0   128     0
511        45    56   211   112   224     0     0   128     0
512        38    55   201   113   213     0     0   128     0
513        37    54   204   111   208     0     0   128     0
514        34    50   202   112   201     0     0   128     0
515        30    47   215   116   194     0     0   128     0
516        28    44   207   119   194     0     0   128     0
517        24    42   214   126   200     0     0   128     0  d
518   d̂b   25    39   214   129   202     0     0   128     0
519        23    35   218   131   207     0     0   128     0
```

```
520        20    32   227   141   208     0     0   128     0
521        20    30   224   147   204     0     0   128     0
522        16    36   225   152   202     0     0   128     0
523        15    47   156   151   186     0     0   128     0  b
524       110    54   117   150   174     0     0   128     0
525       166    57   102   153   170     0     0   128     0
526       195    57    96   154   167     0     0   128     0
527       216    57    89   153   164     0     0   128     0
528       223    57    84   153   161     0     0   128     0
529       228    57    81   153   161     0     0   128     0
530   ʌ   230    56    79   153   160     0     0   128     0
531       230    59    81   150   162     0     0   128     0  u
532       215    62    85   147   163     0     0   128     0
533       200    64    92   146   167     0     0   128     0
534       180    64    93   143   173     0     0   128     0
535       167    66    99   138   167     0     0   128     0
536       140    67   109   135   168     0     0   100     0
537       115    67   117   133   166     0     0   100     0
538        79    65   130   130   167     0     0   100     0
539        35    61   137   125   164    20    30   100     0
540        12    56   140   120   167    30    50   100     0
541         2    56   142   113   167    30    50   100     0
542         0    60   147   107   169    30    50   100     0
543   s     0    60   147   102   169    30    50   100     0
544         0    61   150   101   175    30    50   100     0
545         0    62   151   102   167    30    50   100     0  s
546         0    62   151   107   158    30    50   100     0
547         0    63   146   116   151    30    50   100     0
548         0    63   137   136   162    20    30   100     0
549        12    64   131   143   167     0     0   100     0
550        48    64   111   143   171     0     0   100     0
551        73    55    99   143   173     0     0   100     0
552        90    48    86   145   176     0     0   128     0
553   ɔ   105    51    79   149   176     0     0   128     0
554       117    54    72   152   173     0     0   128     0  o
555       131    54    68   153   165     0     0   128     0
556       138    57    68   150   163     0     0   128     0
557       140    57    72   146   166     0     0   128     0
558       138    58    76   143   165     0     0   128     0
559       135    57    79   139   168     0     0   128     0
560       132    57    91   133   165     0     0   128     0
561       131    57   103   123   164     0     0   128     0
562       128    57   122   114   164     0     0   128     0
563       126    57   148   107   164     0     0   128     0
564       123    57   172   103   166     0     0   128     0
565       121    56   201   100   167     0     0   128     0
566   n   119    55   211    98   166     0     0   128     0  n
567       118    54   213    96   164     0     0   128     0
568       119    55   217    94   162     0     0   128     0
569       120    55   217    91   161     0     0   128     0
570       123    55   219    90   161     0     0   128     0
571       125    54   214    89   164     0     0   128     0
572       125    56   209    89   171     0     0   128     0
573   y   126    54   208    91   181     0     0   128     0
574       129    54   183    95   209     0     0   128     0  y
575       131    52   160   106   218     0     0   128     0
576       134    54   146   131   211     0     0   128     0
577       139    54   140   150   211     0     0   128     0
```

578		143	54	133	166	214	0	0	128	0	
579		149	52	124	173	214	0	0	128	0	
580		157	52	123	182	214	0	0	128	0	
581	o	161	52	123	188	211	0	0	128	0	
582		167	51	121	186	214	0	0	128	0	o
583		171	51	121	185	222	0	0	128	0	
584		172	51	118	185	232	0	0	128	0	
585		174	52	112	175	234	0	0	128	0	
586		173	51	110	169	255	0	0	128	0	
587	ə	172	54	110	167	255	0	0	128	0	
588		166	57	109	158	255	0	0	128	0	u
589		157	57	112	151	255	0	0	128	0	
590		131	58	113	140	254	0	0	128	0	r
591		71	60	112	128	245	0	0	128	0	
592		19	62	113	123	238	0	0	128	0	
593		11	64	114	113	224	0	0	128	0	
594		4	66	118	98	213	0	0	128	0	
595		1	67	147	89	204	0	0	128	0	
596		0	69	149	86	203	0	0	128	0	
597		0	72	148	84	196	0	0	128	0	
598		0	74	147	83	193	0	0	128	0	
599		0	76	147	81	190	0	0	128	0	
600		0	79	142	80	189	0	0	128	0	
601		0	81	136	80	186	0	0	128	0	
602		2	82	137	80	181	0	0	128	0	
603		6	84	141	79	173	0	0	128	0	
604		16	86	142	78	168	0	0	128	0	
605		48	88	142	78	169	0	0	128	0	
606	eⁱ	82	89	147	79	172	0	0	128	0	
607		118	92	152	79	172	0	0	128	0	
608		154	93	151	76	163	0	0	128	0	e
609		172	100	150	74	158	0	0	128	0	
610		184	101	149	70	152	0	0	128	0	
611		193	103	155	68	146	0	0	128	0	
612		199	105	166	67	141	0	0	128	0	i
613		205	108	171	64	141	0	0	128	0	
614		206	108	175	66	141	0	0	128	0	
615		203	109	178	67	140	0	0	128	0	g
616		194	108	180	69	141	0	0	128	0	h
617		120	108	186	73	142	0	0	128	0	
618		52	108	186	76	140	0	0	128	0	
619	ʌ	50	105	186	77	139	0	0	128	0	
620		96	110	183	76	136	0	0	128	0	t
621		123	114	181	69	135	0	0	128	0	
622		134	113	180	65	133	0	0	128	0	
623		137	109	182	60	129	0	0	128	0	
624	i	139	109	183	59	121	0	0	128	0	
625		141	109	185	59	120	0	0	128	0	
626		139	97	185	62	118	0	0	128	0	y
627		127	85	186	62	120	0	0	128	0	
628		105	68	183	65	123	0	0	128	0	
629		62	55	164	70	128	0	0	128	0	
630		38	46	157	71	133	0	0	128	0	
631		25	46	147	74	136	0	0	128	0	
632		30	74	133	78	141	0	0	128	0	
633		36	80	126	80	146	0	0	128	0	
634		51	80	128	79	152	0	0	128	0	
635		62	81	131	80	156	0	0	128	0	

636		113	81	131	81	159	0	0	128	0
637		141	81	140	80	164	0	0	128	0
638		172	84	155	80	163	0	0	128	0
639		199	85	159	79	164	0	0	128	0 e
640	e^i	222	88	161	76	159	0	0	128	0
641		234	86	163	75	154	0	0	128	0
642		239	85	165	74	157	0	0	128	0
643		242	84	168	72	156	0	0	128	0 l
644		237	84	170	74	156	0	0	128	0
645		216	81	172	80	155	0	0	128	0 g
646		186	81	175	88	159	0	0	128	0
647		53	82	185	99	162	0	0	128	0 h
648	d	38	84	183	100	158	0	0	128	0
649		97	84	179	81	151	0	0	128	0 t
650		161	85	186	75	145	0	0	128	0
651		179	85	191	69	140	0	0	128	0
652		186	86	194	68	139	0	0	128	0
653	i	163	88	196	65	140	0	0	128	0 y
654		139	88	197	64	142	0	0	128	0
655		123	88	196	66	144	0	0	128	0
656		93	88	194	66	146	0	0	128	0
657		66	86	186	69	151	0	0	128	0
658		30	85	184	75	160	0	0	128	0
659	m	30	77	201	88	167	0	0	128	50 m
660		68	68	209	116	174	0	0	128	70
661		79	66	210	123	176	0	0	128	80
662		85	63	212	120	176	0	0	128	80
663		90	61	214	116	179	0	0	128	80
664		95	60	218	112	179	0	0	128	50
665		110	60	219	113	177	0	0	128	0
666	a	120	60	219	112	175	0	0	128	0
667		130	61	217	113	173	0	0	128	0
668		144	61	116	143	171	0	0	128	0
669		153	62	79	155	172	0	0	128	0
670		166	63	75	142	170	0	0	128	0
671		182	64	73	136	172	0	0	128	0 i
672		195	64	72	133	173	0	0	128	0
673		210	67	75	124	176	0	0	128	0
674	z	212	74	91	118	182	0	0	128	0
675		212	76	105	113	188	0	0	128	0
676		208	77	133	106	194	0	0	128	0
677		198	76	148	102	203	0	0	128	0
678		167	76	170	98	205	0	0	128	0
679		47	79	172	100	206	0	0	100	0
680		9	78	167	96	200	0	0	100	0
681		0	78	171	102	188	0	0	100	0
682	K	0	79	151	118	193	0	0	100	0 c
683	$\longrightarrow$	0	77	125	118	198	30	40	100	0
684	REL	0	77	101	122	203	25	0	100	0
685		0	77	102	135	210	20	0	100	0
686		15	76	120	157	216	15	0	100	0
687		38	76	160	160	235	10	0	100	0 r
688	r	99	77	150	163	245	0	0	100	0
689		169	74	138	163	247	0	0	100	0
690		195	72	134	164	237	0	0	100	0
691		206	71	132	166	230	0	0	100	0
692		212	68	133	167	213	0	0	100	0 o
693	O	211	67	134	168	198	0	0	100	0

694		203	62	137	171	188	0	0	100	0
695		161	60	143	178	181	0	0	100	0
696		144	60	152	178	174	0	0	100	0
697		61	56	169	179	175	0	0	100	0
698		19	51	163	182	178	0	0	100	0
699		6	51	149	176	178	0	0	100	0
700		0	51	125	154	171	0	0	100	0
701	k	0	52	91	118	157	0	0	100	0 c
702		0	55	69	108	148	0	0	100	0
703	$\xrightarrow[\text{REL}]{}$	0	56	110	111	141	30	40	100	0
704		0	56	146	115	146	25	0	100	0
705		0	56	145	116	176	20	0	100	0
706	a	23	57	162	130	193	15	0	100	0
707		45	57	173	136	197	10	0	100	0 o
708		64	57	202	167	194	0	0	100	0
709		78	57	207	161	190	0	0	100	30
710		90	55	206	161	189	0	0	100	40
711	m	73	54	203	158	192	0	0	100	70 m
712		60	53	190	149	192	0	0	128	100
713		34	53	188	126	192	0	0	128	100
714		13	55	221	120	191	0	0	128	50
715		0	56	214	114	192	0	0	128	0
716	p	0	58	126	107	190	0	0	128	0 p
717	$\xrightarrow[\text{REL}]{}$	0	61	124	101	181	25	40	128	0
718		0	63	119	99	174	18	0	128	0
719		0	64	150	93	174	10	0	128	0
720	y	65	64	174	88	174	5	0	128	0
721		118	64	179	86	174	0	0	128	0
722		171	64	177	88	175	0	0	128	0 u
723		195	67	175	95	180	0	0	128	0
724	u	207	67	171	105	183	0	0	128	0
725		211	67	170	112	187	0	0	128	0
726		211	67	166	118	188	0	0	128	0
727		210	64	159	123	190	0	0	128	0
728		208	63	155	127	193	0	0	128	0
729		205	58	147	128	195	0	0	128	0
730		198	54	146	128	198	0	0	128	0
731		198	46	148	129	204	0	0	128	0
732		146	41	157	128	212	0	0	128	0
733		120	35	160	127	219	0	0	128	0
734	l	103	32	164	126	231	0	0	128	0 t
735		92	29	150	126	240	0	0	128	0
736		84	24	142	126	247	0	0	128	0
737		73	17	147	126	248	0	0	128	0
738		64	13	139	128	250	0	0	128	0
739		58	8	141	128	253	0	0	128	0
740		49	5	142	129	255	0	0	128	0 e
741	r	41	0	149	129	255	0	0	128	0
742		32	0	148	131	255	0	0	128	0 r
743		25	0	147	132	255	0	0	128	0
744		18	0	152	134	255	0	0	128	0
745		11	0	153	139	255	0	0	128	0
746		7	0	154	142	255	0	0	128	0

Descriptions of COMPUTALKER Speech Parameters
Data Tapes DIGITS and LETTERS

DIGITS data dated July 30, 1977:

Length 400 frames (occupies 3600 bytes).

Contains parameter data for the 10 digits Ø,1,2,3,4,5,6,7,8, and 9 in CT-1 Speech Parameter Data format.

digit	begins at frame
Ø	1
1	41
2	81
3	121
4	161
5	201
6	241
7	281
8	321
9	361

All digits are 40 frames long

LETTERS data dated Mar 28, 1977:

Length 1040 frames (occupies 9360 bytes).

Contains parameter data for the 26 letters A thru Z in CT-1 Speech Parameter Data format.

letter	begins at frame	letter	begins at frame
A	1	N	521
B	41	O	561
C	81	P	601
D	121	Q	641
E	161	R	681
F	201	S	721
G	241	T	761
H	281	U	801
I	321	V	841
J	361	W	881
K	401	X	921
L	441	Y	961
M	481	Z	1001

All letters are 40 frames long

INTEL HEX FORMAT

The assembled Hex Object paper tape of the CSR1 system is punched in INTEL HEX format. The following is a definition of that format.

Frame 0	Record Mark. Signals the start of a record. The ASCII character colon (":" HEX 3A) is used as the record mark.
Frames 1,2 (0-9,A-F)	Record Length. Two ASCII characters representing a hexadecimal number in the range 0 to 'FF' (0 to 255). This is the count of actual data bytes in the record type or checksum. A record length of 0 indicates end of file.
Frames 3 to 6	Load Address. Four ASCII characters that represent the initial memory location where the data following will be loaded. The first data byte is stored in the location pointed to by the load address; succeeding data bytes are loaded into ascending addresses.
Frames 7, 8	Record Type. Two ASCII characters. Currently all records are type 0. This field is reserved for future expansion.
Frames 9 to 9+2* (Record Length) -1	Data. Each 8 bit memory word is represented by two frames containing the ASCII characters (0 to 9, A to F) to represent a hexadecimal value 0 to 'FF'H (0 to 255).
Frames 9+2* (Record Length) to 9+2* (Record Length) +1	Checksum. The checksum is the negative of the sum of all 8 bit bytes in the record since the record mark (":") evaluated modulus 256. That is, if you add together all the 8 bit bytes, ignoring all carries out of an 8-bit sum, then add the checksum, the result is zero.

Example: If memory locations 1 through 3 contain 53F8EC, the format of the hex file produced when these locations are punched is:

```
:0300010053F8ECC5
```

```
ØØ8Ø             *  INTEL HEX FORMAT LOADER
ØØ8Ø             *
ØØ8Ø             *  COPIED FROM IMSAI SCS REV 1
ØØ8Ø             *  TO USE THIS LOADER, FIRST KEY IT IN, STARTING
ØØ8Ø             *  AT LOCATION ØØ8ØH.  THEN MOUNT THE TAPE IN THE
ØØ8Ø             *  TTY READER, SET THE ADDRESS SWITCHES TO ØØ8ØH,
ØØ8Ø             *  THEN PRESS 'STOP','RESET','EXAMINE', AND 'RUN'.
ØØ8Ø             *  THE TTY READER SHOULD START AUTOMATICALLY.
ØØ8Ø             *  FOR A DIFFERENT READER INPUT ADDRESS, MODIFY
ØØ8Ø             *  ROUTINE 'CHRIN' AT ØØEØH.
ØØ8Ø             *
ØØ8Ø             *
ØØ8Ø             TTYS    EQU   Ø
ØØ8Ø             TTYD    EQU   1
ØØ8Ø             *
ØØ8Ø 31 8Ø ØØ    HXLOAD  LXI   SP,$     * SET UP THE STACK
ØØ83 3E CE               MVI   A,ØCEH
ØØ85 D3 ØØ               OUT   TTYS     * SET TTY MODE
ØØ87 3E 17               MVI   A,17H
ØØ89 D3 ØØ               OUT   TTYS
ØØ8B 3E 11               MVI   A,11H
ØØ8D D3 Ø1               OUT   TTYD     * X-ON TO READER
ØØ8F CD 95 ØØ            CALL  HEXIN    * READ THE TAPE
ØØ92 C3 92 ØØ            JMP   $        * THEN HANG
ØØ95             *
ØØ95             *  HEX LOAD SUBROUTINE BEGINS HERE
ØØ95 CD EØ ØØ    HEXIN   CALL  CHRIN    * GET A CHAR
ØØ98 FE 3A               CPI   ':'
ØØ9A C2 95 ØØ            JNZ   HEXIN    * WAIT FOR COLON
ØØ9D ØE ØØ               MVI   C,Ø      * CLEAR CHECKSUM
ØØ9F CD C2 ØØ            CALL  GEBYT    * GET THE COUNT
ØØA2 B7                  ORA   A        * SET THE FLAGS
ØØA3 C8                  RZ    *        * RET ZERO IF EOF RECORD
ØØA4 47                  MOV   B,A      * ELSE PUT COUNT IN B
ØØA5 CD C2 ØØ            CALL  GEBYT    * GET HI BYTE OF ADDRESS
ØØA8 67                  MOV   H,A
ØØA9 CD C2 ØØ            CALL  GEBYT    * GET LO BYTE OF ADDRESS
ØØAC 6F                  MOV   L,A      * ADDRESS IS IN HL
ØØAD CD C2 ØØ            CALL  GEBYT    * GET & IGNORE RECORD TYPE
ØØBØ             * GO THRU THIS LOOP ONCE FOR EACH DATA BYTE IN A RECORD
ØØBØ CD C2 ØØ    LOOP    CALL  GEBYT    * GET A DATA BYTE
ØØB3 77                  MOV   M,A      * STORE IT
ØØB4 23                  INX   H        * BUMP ADDR
ØØB5 Ø5                  DCR   B        * DECR COUNT
ØØB6 C2 BØ ØØ            JNZ   LOOP     * DO IT AGAIN
ØØB9 CD C2 ØØ            CALL  GEBYT    * GET THE CHECKSUM
ØØBC 79                  MOV   A,C
ØØBD B7                  ORA   A        * SET FLAGS, RET IF NON-ZERO
ØØBE CØ                  RNZ
ØØBF C3 95 ØØ            JMP   HEXIN    * ELSE GO GET NEXT RECORD
```

```
ØØC2                *
ØØC2                *  THIS ROUTINE READS TWO CHARS FROM THE TAPE
ØØC2                *  AND ASSEMBLES THEM INTO A BYTE, WHICH IS
ØØC2                *  RETURNED IN THE A REGISTER.
ØØC2 CD D3 ØØ       GEBYT   CALL  GEDIG    * GET A HEX DIGIT
ØØC5 87                     ADD   A        * SHIFT LEFT 4 BITS
ØØC6 87                     ADD   A
ØØC7 87                     ADD   A
ØØC8 87                     ADD   A
ØØC9 57                     MOV   D,A      * SAVE IT IN D
ØØCA CD D3 ØØ               CALL  GEDIG    * GET ANOTHER DIGIT
ØØCD B2                     ORA   D
ØØCE 57                     MOV   D,A      * SAVE BYTE IN D
ØØCF 81                     ADD   C
ØØDØ 4F                     MOV   C,A      * ACCUMULATE CHECKSUM
ØØD1 7A                     MOV   A,D      * GET DATA BYTE BACK
ØØD2 C9                     RET
ØØD3                *
ØØD3                *  THIS ROUTINE READS A HEX DIGIT FROM THE TAPE.
ØØD3                *  NOTE THAT IT DOES NO VALIDITY CHECKING.
ØØD3 CD EØ ØØ       GEDIG   CALL  CHRIN    * GET A CHAR FROM THE TAPE
ØØD6 FE 3A                  CPI   '9'+1
ØØD8 DA DD ØØ               JC    GENUM    * SKIP ADI IF NUMERIC
ØØDB C6 Ø9                  ADI   9
ØØDD E6 ØF          GENUM   ANI   ØFH      * MASK OUT UPPER 4 BITS
ØØDF C9                     RET
ØØEØ                *
ØØEØ                *  THIS ROUTINE READS A CHAR FROM THE TAPE READER
ØØEØ DB ØØ          CHRIN   IN    TTYS     * GET TTY STATUS
ØØE2 E6 4Ø                  ANI   4ØH
ØØE4 CA EØ ØØ               JZ    CHRIN    * WAIT TILL READY
ØØE7 DB Ø1                  IN    TTYD     * GET THE CHAR
ØØE9 E6 7F                  ANI   7FH      * KILL THE PARITY BIT
ØØEB C9                     RET
ØØEC                *
ØØEC                *  END OF HEX LOADER
```

```
Ø1ØØ     *  CTPLAY
Ø1ØØ     *
Ø1ØØ     *    CT-1 DEMONSTRATION PLAYBACK PROGRAM
Ø1ØØ     *
Ø1ØØ     *  READ SPEECH PARAMETER DATA TAPE AND PLAY TO
Ø1ØØ     *  CT-1 SYNTHESIZER UNDER CONSOLE SWITCH CONTROL
Ø1ØØ     *
Ø1ØØ     *
Ø1ØØ     *  CTPLAY IS A MINIMAL CT-1 CONTROL ROUTINE.  IT CONTAINS
Ø1ØØ     *  THE BASIC CT-1 SETUP AND PLAYBACK SUBROUTINES WHICH
Ø1ØØ     *  MAY BE INCORPORATED INTO OTHER PROGRAMS.  THE ROUTINE
Ø1ØØ     *  'CTOUT', AS INCLUDED HERE IS SET UP TO RUN IN RAM ONLY.
Ø1ØØ     *  LOCATION CTO+1 IS MODIFIED DURING CT-1 PLAYBACK.
Ø1ØØ     *  THE PROGRAMS CTMON AND CTEDIT CONTAIN EXAMPLES WHERE
Ø1ØØ     *  THIS OUTPUT INSTR. IS CONSTRUCTED IN RAM SO THE CODE
Ø1ØØ     *  ITSELF CAN BE RUN IN ROM.
Ø1ØØ     *
Ø1ØØ     *
Ø1ØØ     *   USING CTPLAY:
Ø1ØØ     *      PUT ALL SWITCHES DOWN
Ø1ØØ     *      RAISE SW Ø TO READ A PAPER TAPE INTO THE BUFFER
Ø1ØØ     *      RAISE SW 6 TO REPEAT CT-1 PLAYBACK
Ø1ØØ     *      RAISE SW 7 TO PLAY DATA TO CT-1
Ø1ØØ     *
Ø1ØØ     *
Ø1ØØ     *
Ø1ØØ     *    WRITTEN   DEC 2Ø, 1976
Ø1ØØ     *    VERSION 1.1     REVISED SEP 22, 77
Ø1ØØ     *    BY LLOYD RICE
Ø1ØØ     *       COMPUTALKER CONSULTANTS
Ø1ØØ     *       P.O. BOX 1951
Ø1ØØ     *       SANTA MONICA, CA  9Ø4Ø6
Ø1ØØ     *
Ø1ØØ     *
Ø1ØØ     *
Ø1ØØ     *  THIS CONTROL MONITOR SOFTWARE IS FURNISHED TO PURCHASERS
Ø1ØØ     *  OF THE COMPUTALKER MODEL CT-1 SPEECH SYNTHESIZER.
Ø1ØØ     *  IT MAY BE COPIED OR MODIFIED AS DESIRED WITHOUT
Ø1ØØ     *  SPECIFIC PERMISSION.  COMPUTALKER'S RESPONSIBILITY FOR
Ø1ØØ     *  MAINTENANCE APPLIES ONLY TO THE ORIGINAL VERSION OF
Ø1ØØ     *  THE CODE (AS LISTED HERE) AND ONLY WHEN IN USE BY THE
Ø1ØØ     *  ORIGINAL PURCHASER.
Ø1ØØ     *
Ø1ØØ     *  COMPUTALKER OFFERS TO ASSIST WITH OTHER VERSIONS
Ø1ØØ     *  WHEREVER REASONABLE AND POSSIBLE.
Ø1ØØ     *
```

```
Ø1ØØ                *
Ø1ØØ                CTBASE  EQU   ØEØH
Ø1ØØ                *
Ø1ØØ                *
Ø1ØØ                STACK   EQU   $
Ø1ØØ                *
Ø1ØØ 31 ØØ Ø1       CTPLAY  LXI   SP,STACK
Ø1Ø3 3E 55          START   MVI   A,55H      * SET CONSOLE LIGHTS TO 1Ø1Ø1Ø1Ø
Ø1Ø5 D3 FF                  OUT   ØFFH
Ø1Ø7 Ø1 ØØ 1Ø       SDTIME  LXI   B,1ØØØH    * ALL SWITCHES MUST BE DOWN
Ø1ØA DB FF          SWDN    IN    ØFFH       *   FOR 1ØØ MSEC CONTINUOUSLY
Ø1ØC B7                     ORA   A
Ø1ØD C2 Ø7 Ø1               JNZ   SDTIME
Ø11Ø ØB                     DCX   B
Ø111 78                     MOV   A,B
Ø112 B1                     ORA   C
Ø113 C2 ØA Ø1               JNZ   SWDN
Ø116 DB FF          SWUP    IN    ØFFH       * LOOP HERE UNTIL A SW IS UP
Ø118 B7                     ORA   A
Ø119 CA 16 Ø1               JZ    SWUP
Ø11C 17                     RAL
Ø11D DA 3F Ø1               JC    PLAY       * SW 7 UP, PLAYBACK
Ø12Ø 17                     RAL
Ø121 DA 47 Ø1               JC    REPEAT     * SW 6 UP, REPEAT
Ø124 17                     RAL
Ø125 DA Ø3 Ø1               JC    START      * SW 5 UP,
Ø128 17                     RAL
Ø129 DA Ø3 Ø1               JC    START      * SW 4 UP,
Ø12C 17                     RAL
Ø12D DA Ø3 Ø1               JC    START      * SW 3 UP,
Ø13Ø 17                     RAL
Ø131 DA Ø3 Ø1               JC    START      * SW 2 UP,
Ø134 17                     RAL
Ø135 DA Ø3 Ø1               JC    START      * SW 1 UP,
Ø138 17                     RAL
Ø139 DA 95 Ø1               JC    READ       * SW Ø UP, READ PAPER TAPE
Ø13C C3 Ø7 Ø1               JMP   SDTIME     * HOW DID WE GET HERE??
Ø13F                *
Ø13F                *
Ø13F                *  CT-1 PLAYBACK ROUTINES
Ø13F                *
Ø13F                *  THE CT-1 PLAYBACK CODE ASSUMES A FRAMECOUNT IS STORED
Ø13F                *  IN THE 2 BYTES JUST PRECEEDING THE DATA BUFFER.
Ø13F                *
Ø13F                *
Ø13F                *  PLAYBACK
Ø13F                *
Ø13F 3E 7F          PLAY    MVI   A,7FH      * PUT 8Ø IN CONSOLE LEDS
Ø141 D3 FF                  OUT   ØFFH
```

```
Ø143 AF                    XRA    A
Ø144 C3 4D Ø1              JMP    PLABK       * CLEAR REPEAT SW
Ø147               *
Ø147               *  REPEAT
Ø147               *
Ø147 3E BF         REPEAT  MVI    A,ØBFH      * PUT 4Ø IN CONSOLE LEDS
Ø149 D3 FF                 OUT    ØFFH
Ø14B 3E 4Ø                 MVI    A,4ØH
Ø14D               *
Ø14D               *  COMMON PLAYBACK LOOP
Ø14D               *
Ø14D 32 6E Ø1      PLABK   STA    RPTSW+1     * REPEAT SW = 4ØH TO REPEAT, ELSE Ø
Ø15Ø 21 E9 Ø1              LXI    H,BUFFER    * GET PLAYBACK ARGUMENTS
Ø153 5E                    MOV    E,M
Ø154 23                    INX    H
Ø155 56                    MOV    D,M         * FRAME COUNT TO DE
Ø156 23                    INX    H           * DATA ADDRESS TO HL
Ø157 CD 77 Ø1              CALL   CTOUT       * SET CT-1 PARAMS FROM FRAME 1
Ø15A 1B                    DCX    D           * COUNT THAT FRAME
Ø15B 3E FF                 MVI    A,255       * TURN ON THE CT-1
Ø15D D3 EF                 OUT    CTBASE+15
Ø15F CD 77 Ø1      PLOOP   CALL   CTOUT       * PLAY A FRAME
Ø162 CD 89 Ø1              CALL   DLY1Ø       * DELAY 1Ø MSEC
Ø165 1B                    DCX    D
Ø166 7A                    MOV    A,D
Ø167 B3                    ORA    E
Ø168 C2 5F Ø1              JNZ    PLOOP       * DONE?
Ø16B DB FF                 IN     ØFFH        * TEST REPEAT SW
Ø16D E6 ØØ         RPTSW   ANI    Ø           * SEE IF SW 6 IS STILL UP
Ø16F C2 4D Ø1              JNZ    PLABK       * YES, PLAY AGAIN
Ø172 D3 EF                 OUT    CTBASE+15   * NO, TURN OFF CT-1
Ø174 C3 Ø3 Ø1              JMP    START       * AND GO BACK TO CMMD LOOP
Ø177               *
Ø177               *  CTOUT PLAYS 1 DATA FRAME FROM THE BUFFER.
Ø177               *  ON ENTRY: HL POINTS TO AV OF THE FRAME TO PLAY
Ø177               *  CONTENTS OF DE NOT CHANGED BY THIS SUBR
Ø177               *
Ø177 Ø6 EØ         CTOUT   MVI    B,CTBASE    * RE-INITIALIZE OUTPUT ADDR
Ø179 ØE Ø9                 MVI    C,9
Ø17B 78            CTLP    MOV    A,B
Ø17C 32 81 Ø1              STA    CTO+1
Ø17F 7E                    MOV    A,M
Ø18Ø D3 EØ         CTO     OUT    CTBASE      * OUTPUT THE PARAMETER
Ø182 23                    INX    H
Ø183 Ø4                    INR    B
Ø184 ØD                    DCR    C
Ø185 C2 7B Ø1              JNZ    CTLP        * GO AROUND 9 TIMES
Ø188 C9                    RET
Ø189               *
```

```
Ø189              *  DELAY 1Ø MILLISECONDS (ASSUMES 2 MHZ CLOCK)
Ø189              *  (A) CHANGED, ALL ELSE RESTORED
Ø189              *
Ø189 E5           DLY1Ø    PUSH  H
Ø18A 21 2Ø Ø3              LXI   H,8ØØ
Ø18D 2B                    DCX   H
Ø18E 7C                    MOV   A,H
Ø18F B5                    ORA   L
Ø19Ø C2 8D Ø1              JNZ   $-3
Ø193 E1                    POP   H
Ø194 C9                    RET
Ø195              *
Ø195              *
Ø195              *  PAPER TAPE READER LOOP
Ø195              *
Ø195 3E FE        READ     MVI   A,ØFEH     * PUT Ø1 IN CONSOLE PORT
Ø197 D3 FF                 OUT   ØFFH
Ø199 CD CA Ø1              CALL  PTRCLR     * CLEAR READER FLAG
Ø19C CD D3 Ø1     PT1      CALL  PTRIN      * IGNORE THE 1ST BYTE
Ø19F CA 9C Ø1              JZ    PT1        * IF IT TAKES FOREVER
Ø1A2 CD D3 Ø1     PT2      CALL  PTRIN      * IGNORE NULLS (LEADER)
Ø1A5 CA Ø3 Ø1              JZ    START      * ZERO IF TIMER RAN OUT
Ø1A8 B7                    ORA   A
Ø1A9 CA A2 Ø1              JZ    PT2
Ø1AC 21 E9 Ø1              LXI   H,BUFFER   * ALSO IGNORE 1ST NON-ZERO BYTE
Ø1AF CD D3 Ø1              CALL  PTRIN      * READ 2 FRAME COUNT BYTES
Ø1B2 CA Ø3 Ø1              JZ    START
Ø1B5 77                    MOV   M,A
Ø1B6 23                    INX   H
Ø1B7 CD D3 Ø1              CALL  PTRIN
Ø1BA CA Ø3 Ø1              JZ    START
Ø1BD 77                    MOV   M,A
Ø1BE 23                    INX   H
Ø1BF CD D3 Ø1     PT3      CALL  PTRIN      * THEN READ IN THE DATA
Ø1C2 CA Ø3 Ø1              JZ    START
Ø1C5 77                    MOV   M,A
Ø1C6 23                    INX   H
Ø1C7 C3 BF Ø1              JMP   PT3        * READ UNTIL TIME-OUT
Ø1CA              *
Ø1CA              *
Ø1CA              *  PAPER TAPE I/O
Ø1CA              *
Ø1CA              PTRF     EQU   16H        * PTR FLAGS PORT
Ø1CA              PTRD     EQU   17H        * PTR DATA PORT
Ø1CA              *
```

```
Ø1CA 26 Ø3        PTRCLR  MVI    H,3
Ø1CC D3 16                OUT    PTRF
Ø1CE 26 11                MVI    H,11H
Ø1DØ D3 16                OUT    PTRF
Ø1D2 C9                   RET
Ø1D3              *
Ø1D3              *  WAIT FOR PTR FLAG & READ 1 BYTE.  IF FLAG IS NOT
Ø1D3              *  READY WITHIN 4ØØ MSEC, RETURN WITH Z FLAG SET.
Ø1D3              *  IF DATA READ NORMALLY, RETURN WITH Z=Ø, &
Ø1D3              *  THE DATA BYTE IS IN A.
Ø1D3              *
Ø1D3 C5           PTRIN   PUSH   B          * SAVE B
Ø1D4 Ø1 ØØ 4Ø             LXI    B,4ØØØH    * SET TIMER
Ø1D7 DB 16        P1      IN     PTRF       * WAIT FOR FLAG
Ø1D9 ØØ                   NOP    * CMA FOR ACTIVE LOW STATUS
Ø1DA E6 Ø1                ANI    1
Ø1DC C2 E5 Ø1             JNZ    P2
Ø1DF ØB                   DCX    B          * NOT READY, CHECK THE TIME
Ø1EØ 78                   MOV    A,B
Ø1E1 B1                   ORA    C
Ø1E2 C2 D7 Ø1             JNZ    P1
Ø1E5 DB 17        P2      IN     PTRD       * CONDITION ZERO IF TIME OUT
Ø1E7 C1                   POP    B          * RESTORE B
Ø1E8 C9                   RET
Ø1E9              *
Ø1E9              *
Ø1E9              *  CT-1 PARAMETER BUFFER
Ø1E9              *
Ø1E9 ØØ           BUFFER  DB     Ø          * CT-1 DATA BUFFER BEGINS HERE
Ø1EA              *                         * 1ST 2 BYTES OF BUFFER ARE FRAME COUNT
Ø1EA              *                         * THEN FOLLOWS THE PARAMETER DATA
Ø1EA              *
```

```
SYMB


CTBASE     ØØEØ : STACK     Ø1ØØ : CTPLAY     Ø1ØØ
START      Ø1Ø3 : SDTIME    Ø1Ø7 : SWDN       Ø1ØA
SWUP       Ø116 : PLAY      Ø13F : REPEAT     Ø147
PLABK      Ø14D : PLOOP     Ø15F : RPTSW      Ø16D
CTOUT      Ø177 : CTLP      Ø17B : CTO        Ø18Ø
DLY1Ø      Ø189 : READ      Ø195 : PT1        Ø19C
PT2        Ø1A2 : PT3       Ø1BF : PTRF       ØØ16
PTRD       ØØ17 : PTRCLR    Ø1CA : PTRIN      Ø1D3
P1         Ø1D7 : P2        Ø1E5 : BUFFER     Ø1E9
```

PAGE 01

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

0000            0000 *  CT-1 CONTROL MONITOR
0000            0005 *  CTMON  VERSION 1.10
0000            0010 *
0000            0015 *  ALLOWS READING, EDITING, AND STORING CT-1
0000            0020 *  SPEECH PARAMETER DATA FILES IN DECIMAL USING
0000            0025 *  A POLYMORPHIC OR PROCESSOR TECH VIDEO DISPLAY.
0000            0030 *
0000            0035 *  SUPPORTS STORAGE ON PAPER TAPE OR AUDIO
0000            0040 *  CASSETTE IN THE TARBELL FORMAT.
0000            0045 *
0000            0050 *
0000            0055 *
0000            0060 *    WRITTEN   DEC 18, 1976
0000            0065 *    BY LLOYD RICE
0000            0070 *       COMPUTALKER CONSULTANTS
0000            0075 *       P.O. BOX 1951
0000            0080 *       SANTA MONICA, CA  90406
0000            0085 *
0000            0090 *
0000            0095 *
0000            0100 *  THIS CONTROL MONITOR SOFTWARE IS FURNISHED TO
0000            0105 *  PURCHASERS OF THE COMPUTALKER MODEL CT-1
0000            0110 *  SPEECH SYNTHESIZER.  IT MAY BE COPIED OR
0000            0115 *  MODIFIED AS DESIRED WITHOUT SPECIFIC PERMISSION.
0000            0120 *  COMPUTALKER'S RESPONSIBILITY FOR MAINTENANCE
0000            0125 *  APPLIES ONLY TO THE ORIGINAL VERSION OF THE CODE
0000            0130 *  AND ONLY WHEN IN USE BY THE ORIGINAL PURCHASER.
0000            0135 *
0000            0140 *  COMPUTALKER OFFERS TO ASSIST WITH OTHER VERSIONS
0000            0145 *  WHEREVER REASONABLE AND POSSIBLE.
0000            0150 *
0000            0155        ORG    100H
0100            0160 *
0100            0165 VIDBUF EQU    88H        * HIGH ORDER VIDEO ADDR
0100            0170 CTBASE EQU    0E0H       * CT-1 CHAN AV ADDRESS
0100            0175 *
0100 21 01 00   0180 CTMON  LXI    H,1
0103 22 4C 06   0185        SHLD   LENGTH     * INIT LENGTH TO 1 FRAME
0106 06 09      0190        MVI    B,9
0108 21 4E 08   0195        LXI    H,BUFFER   * ZERO THE 1ST FRAME
010B 36 00      0200 CTMA   MVI    M,0
010D 23         0205        INX    H
010E 05         0210        DCR    B
010F C2 0B 01   0215        JNZ    CTMA
0112 21 40 00   0220        LXI    H,PLADAT
0115 36 D3      0225        MVI    M,0D3H     * SET UP PLADAT SUBROUTINE
0117 23         0230        INX    H
0118 23         0235        INX    H
0119 36 C9      0240        MVI    M,0C9H     * SET UP RETURN
011B 31 FF 00   0245 CTRST  LXI    SP,STACK   * INITIALIZE THE STACK
011E 3E 0A      0250        MVI    A,0AH
0120 32 49 00   0255        STA    FRTIME     * INIT FRAME TIME TO 10 MSEC
0123 21 20 03   0260        LXI    H,320H
0126 22 4B 00   0265        SHLD   RPTIME     * INIT REPEAT DELAY TO 800 MSEC
0129 CD 90 06   0270        CALL   CLRVID     * CLEAR THE SCREEN
012C 21 DE 07   0275        LXI    H,HDRTX    * DISPLAY LINE 0 HEADER
```

PAGE 02

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

012F 01 18 00   0280        LXI    B,18H
0132 11 00 06   0285        LXI    D,600H     * LINE # IN E, CHAR POS IN D
0135 CD F7 06   0290        CALL   DSTX
0138 21 F9 07   0295        LXI    H,FRNTX    * DISPLAY PARTIAL LINE 1 HEADER
013B 01 10 00   0300        LXI    B,10H
013E 11 01 01   0305        LXI    D,101H
0141 CD F7 06   0310        CALL   DSTX
0144 21 09 08   0315        LXI    H,CMDTX    * DISPLAY COMMAND PROMPT
0147 01 18 00   0320        LXI    B,18H
014A 11 0F 05   0325        LXI    D,50FH
014D CD F7 06   0330        CALL   DSTX
0150 2A 4C 06   0335 CT0    LHLD   LENGTH     * INIT PLAY PARAMETERS
0153 22 56 00   0340        SHLD   PLACNT     * FRAME COUNT <= LENGTH
0156 21 01 00   0345        LXI    H,1
0159 22 54 00   0350        SHLD   PLABGN     * BEGIN AT FRAME 1
015C 22 4F 00   0355        SHLD   CL         * INIT CURRENT LINE TO LINE 1
015F AF         0360        XRA    A
0160 32 4D 00   0365        STA    CPM        * INIT CURRENT PARAM TO AV
0163 11 0F 0F   0370 CT1    LXI    D,0F0FH
0166 CD E1 06   0375        CALL   VBUFAD
0169 06 20      0380        MVI    B,' '
016B 2A 4C 06   0385        LHLD   LENGTH
016E CD 07 07   0390        CALL   DISPI4     * DISPLAY BUFFFER LENGTH
0171 2A 4F 00   0395        LHLD   CL
0174 CD D1 05   0400        CALL   LIMRNG     * FORCE CL TO WITHIN LEGAL LIMITS
0177 22 4F 00   0405        SHLD   CL
017A 11 01 14   0410 CT2    LXI    D,1401H    * LINE #1, CHAR POS 20
017D 21 21 08   0415        LXI    H,FORTX    * DISPLAY REST OF HEADER
0180 01 11 00   0420        LXI    B,11H
0183 3A 4D 00   0425        LDA    CPM        * DEPENDING ON CURRENT PARAMETER
0186 FE 05      0430        CPI    5
0188 FA 8C 01   0435        JM     CT2A       * CP<5, DISPLAY FORMANTS
018B 09         0440        DAD    B          * CT>4, DISPLAY AMPLITUDES
018C CD F7 06   0445 CT2A   CALL   DSTX
018F 11 FC FF   0450        LXI    D,-4       * TOP DISPLAY LINE IS FRAME CL-4
0192 2A 4F 00   0455        LHLD   CL
0195 19         0460        DAD    D
0196 44         0465        MOV    B,H        * PUT THAT FRAME NO. IN BC
0197 4D         0470        MOV    C,L
0198 1E 03      0475        MVI    E,3        * BEGIN DISPLAY AT LINE 3
019A 3E 09      0480        MVI    A,9        * DISPLAY LINE COUNTER
019C 32 FF 00   0485        STA    STACK      * SAVE IN A TEMP
019F 60         0490 CT2C   MOV    H,B
01A0 69         0495        MOV    L,C
01A1 CD D1 05   0500        CALL   LIMRNG
01A4 FA AD 01   0505        JM     CT2D       * FRAME OUT OF RANGE, CLEAR LINE
01A7 CD 53 05   0510        CALL   DISFRM     * DISPLAY 1 FRAME OF DATA
01AA C3 B0 01   0515        JMP    CT2E
01AD CD A1 06   0520 CT2D   CALL   CLRLIN     * DISPLAY A BLANK LINE
01B0 03         0525 CT2E   INX    B          * INCREMENT FRAME NO.
01B1 1C         0530        INR    E          * INCR. DISPLAY LINE NO.
01B2 21 FF 00   0535        LXI    H,STACK    * DECREMENT LINE COUNT
01B5 35         0540        DCR    M
01B6 C2 9F 01   0545        JNZ    CT2C       * LOOP TO DISPLAY ALL LINES
01B9 3E 5B      0550 CT3    MVI    A,5BH      * PLACE ARROW BY CURRENT PARAM.
01BB CD B3 06   0555        CALL   ARROW      * HORIZONTAL POSITION DEPENDS
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

01BE 2A 45 00   0560           LHLD  NUMBER      *  ON PARAMETERS DISPLAYED.
01C1 22 43 00   0565           SHLD  LAST        * SET LAST=NUMBER
01C4 1E 0F      0570 CT3A      MVI   E,0FH
01C6 16 1D      0575           MVI   D,1DH
01C8 CD E1 06   0580           CALL  VBUFAD      * GET CMMD LINE VIDEO BUFF ADDR
01CB EB         0585           XCHG
01CC 22 51 00   0590           SHLD  CMDADR      * SAVE IT
01CF 06 0A      0595           MVI   B,0AH
01D1 36 A0      0600 CT3B      MVI   M,' '+128   * AND CLEAR THE COMMAND LINE
01D3 23         0605           INX   H
01D4 05         0610           DCR   B
01D5 C2 D1 01   0615           JNZ   CT3B
01D8 21 00 00   0620 CT4       LXI   H,0
01DB 22 45 00   0625           SHLD  NUMBER      * SET NUMBER=0
01DE AF         0630           XRA   A
01DF 32 47 00   0635 CT4A      STA   BRANCH      * SET BRANCH=0
01E2 CD 6E 06   0640 CT5       CALL  GETCMC      * GET A CHAR FROM KEYBOARD
01E5 FE 30      0645           CPI   '0'
01E7 DA 0F 02   0650           JC    CT5A        * IT'S < ASCII '0'
01EA FE 3A      0655           CPI   '9'+1
01EC D2 0F 02   0660           JNC   CT5A        * IT'S > ASCII '9'
01EF E6 0F      0665           ANI   0FH         * KEEP THE DECIMAL DIGIT
01F1 4F         0670           MOV   C,A
01F2 06 00      0675           MVI   B,0
01F4 2A 45 00   0680           LHLD  NUMBER
01F7 29         0685           DAD   H
01F8 54         0690           MOV   D,H
01F9 5D         0695           MOV   E,L
01FA 29         0700           DAD   H
01FB 29         0705           DAD   H
01FC 19         0710           DAD   D           * MULTIPLY NUMBER BY 10
01FD 09         0715           DAD   B           * AND ADD DIGIT
01FE 22 45 00   0720           SHLD  NUMBER
0201 3A 47 00   0725           LDA   BRANCH
0204 B7         0730           ORA   A
0205 C2 E2 01   0735           JNZ   CT5
0208 3C         0740           INR   A
0209 32 47 00   0745           STA   BRANCH
020C C3 E2 01   0750           JMP   CT5
020F FE 0D      0755 CT5A      CPI   0DH         * IS CHAR A CR ?
0211 C2 2B 02   0760           JNZ   CT5C        * NO
0214 3A 47 00   0765           LDA   BRANCH      * YES, DO BRANCH OPERATION
0217 B7         0770           ORA   A
0218 C2 1C 02   0775           JNZ   CT5B
021B 3C         0780           INR   A           * IF NO BRANCH SET, MAKE IT A 1
021C 87         0785 CT5B      ADD   A
021D 4F         0790           MOV   C,A
021E 06 00      0795           MVI   B,0
0220 21 BA 07   0800           LXI   H,BRNCTB-2
0223 09         0805           DAD   B           * BUILD POINTER TO BRANCH TABLE
0224 EB         0810           XCHG
0225 1A         0815           LDAX  D
0226 6F         0820           MOV   L,A
0227 13         0825           INX   D
0228 1A         0830           LDAX  D
0229 67         0835           MOV   H,A
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

022A E9         0840           PCHL  *           * DO BRANCH
022B 47         0845 CT5C      MOV   B,A         *SAVE CHAR IN B
022C 3A 47 00   0850           LDA   BRANCH
022F B7         0855           ORA   A
0230 C2 AA 02   0860           JNZ   BADCMD      * BRANCH NOT 0, BAD
0233 78         0865           MOV   A,B         * GET THE CHAR BACK
0234 06 02      0870           MVI   B,2
0236 0E 07      0875           MVI   C,7         * BRANCH LIST COUNTER
0238 21 B5 07   0880           LXI   H,BRNCHR
023B BE         0885 CT5E      CMP   M           * SEARCH BRANCH CHAR TABLE
023C CA 48 02   0890           JZ    CT5F
023F 04         0895           INR   B
0240 23         0900           INX   H
0241 0D         0905           DCR   C
0242 C2 3B 02   0910           JNZ   CT5E
0245 C3 4C 02   0915           JMP   CT5G        * WAS NOT A BRANCH COMMAND
0248 78         0920 CT5F      MOV   A,B
0249 C3 DF 01   0925           JMP   CT4A        * SET BRANCH TO CORRESP. VALUE
024C FE 41      0930 CT5G      CPI   'A'
024E C2 59 02   0935           JNZ   CT5H
0251 21 CC 07   0940           LXI   H,APTAB     * SET POINTERS FOR "A-" PARAMETER
0254 06 04      0945           MVI   B,4         * APTAB COUNTER
0256 C3 63 02   0950           JMP   CT5I
0259 FE 46      0955 CT5H      CPI   'F'
025B C2 82 02   0960           JNZ   CT5L
025E 21 D4 07   0965           LXI   H,FPTAB     * SET POINTERS FOR "F-" PARAMETER
0261 06 05      0970           MVI   B,5         * FPTAB COUNTER
0263 CD 6E 06   0975 CT5I      CALL  GETCMC
0266 BE         0980 CT5J      CMP   M
0267 23         0985           INX   H
0268 CA 7A 02   0990           JZ    KAR2
026B 23         0995           INX   H
026C 05         1000           DCR   B
026D C2 66 02   1005           JNZ   CT5J
0270 C3 AA 02   1010           JMP   BADCMD      * WAS NOT A GOOD PARAMETER NAME
0273 7E         1015 CT5K      MOV   A,M
0274 32 4D 00   1020           STA   CPM         * SET CPM TO NEW PARAMETER
0277 C3 7A 01   1025           JMP   CT2
027A 3E 20      1030 KAR2      MVI   A,' '       * WHEN CHANGING PARAMETERS,
027C CD B3 03   1035           CALL  ARROW       * KILL THE ARROW
027F C3 73 02   1040           JMP   CT5K
0282 FE 43      1045 CT5L      CPI   'C'
0284 CA DD 02   1050           JZ    CASIO       * CASSETTE
0287 FE 54      1055           CPI   'T'
0289 CA 25 03   1060           JZ    TAPIO       * TAPE OR TTY
028C FE 50      1065           CPI   'P'
028E CA EA 03   1070           JZ    PLAY
0291 FE 52      1075           CPI   'R'
0293 CA E5 03   1080           JZ    REPEAT
0296 FE 42      1085           CPI   'B'
0298 CA 9D 04   1090           JZ    BACK
029B FE 4E      1095           CPI   'N'
029D CA 69 04   1100           JZ    NEXT
02A0 FE 22      1105           CPI   '"'
02A2 CA 48 04   1110           JZ    DITTO
02A5 FE 4B      1115           CPI   'K'
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

02A7 CA C2 02    1120         JZ    KILCHK
02AA 31 FF 00    1125 BADCMD  LXI   SP,STACK    * RESET STACK
02AD 21 43 08    1130         LXI   H,ILLEG
02B0 01 09 00    1135         LXI   B,9
02B3 11 0F 1D    1140         LXI   D,1D0FH
02B6 CD F7 06    1145         CALL  DSTX        * DISPLAY "ILLEGAL !"
02B9 CD 5B 07    1150 CT5M    CALL  KEYTST      * UNTIL A KEY IS HIT
02BC CA B9 02    1155         JZ    CT5M
02BF C3 C4 01    1160         JMP   CT3A
02C2 CD 6E 06    1165 KILCHK  CALL  GETCMC      * TO KILL REQUIRES NAME "KILL"
02C5 FE 49       1170         CPI   'I'
02C7 C2 AA 02    1175         JNZ   BADCMD
02CA CD 6E 06    1180         CALL  GETCMC
02CD FE 4C       1185         CPI   'L'
02CF C2 AA 02    1190         JNZ   BADCMD
02D2 CD 6E 06    1195         CALL  GETCMC
02D5 FE 4C       1200         CPI   'L'
02D7 C2 AA 02    1205         JNZ   BADCMD
02DA C3 00 01    1210         JMP   CTMON       * START FROM SCRATCH
02DD             1215 *
02DD             1220 * COMMAND PROCESSORS:     CASSETTE I/O HANDLERS
02DD             1225 *
02DD CD 6E 06    1230 CASIO   CALL  GETCMC
02E0 FE 41       1235         CPI   'A'
02E2 CA F2 02    1240         JZ    CASAP
02E5 FE 52       1245         CPI   'R'
02E7 CA 0B 03    1250         JZ    CASRD
02EA FE 57       1255         CPI   'W'
02EC CA 1A 03    1260         JZ    CASWR
02EF C3 AA 02    1265         JMP   BADCMD
02F2 CD 62 03    1270 CASAP   CALL  SETAP       * SETUP TO APPEND DATA
02F5 CD A5 03    1275         CALL  CRDATA      * READ IT
02F8 2A 56 00    1280 CAS1    LHLD  PLACNT
02FB EB          1285         XCHG
02FC 2A 4C 08    1290         LHLD  LENGTH      * LENGTH=LENGTH+
02FF 19          1295         DAD   D           * NO. OF FRAMES READ
0300 22 4C 08    1300         SHLD  LENGTH
0303 3E 20       1305 KARROW  MVI   A,' '       * AFTER READING IN NEW DATA,
0305 CD B3 05    1310         CALL  ARROW       * KILL THE ARROW
0308 C3 50 01    1315         JMP   CT0
030B 21 4E 08    1320 CASRD   LXI   H,BUFFER
030E CD A5 03    1325         CALL  CRDATA      * READ DATA
0311 2A 56 00    1330 CAS2    LHLD  PLACNT
0314 22 4C 08    1335         SHLD  LENGTH      * LENGTH=NO. OF FRAMES READ
0317 C3 03 03    1340         JMP   KARROW
031A CD 6A 03    1345 CASWR   CALL  SETWR       * SETUP BYTE COUNT IN DE
031D 13          1350         INX   D           * INCLUDE FRAME COUNT BYTES
031E 13          1355         INX   D
031F CD C6 03    1360         CALL  CWDATA      * WRITE IT
0322 C3 C4 01    1365         JMP   CT3A        * DATA IN BUFFER IS STILL OK
0325             1370 *
0325             1375 * PAPER TAPE AND TTY HANDLERS
0325             1380 *
0325 CD 6E 06    1385 TAPIO   CALL  GETCMC
0328 FE 41       1390         CPI   'A'
032A CA 3F 03    1395         JZ    TAPAP
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

032D FE 52       1400         CPI   'R'
032F CA 48 03    1405         JZ    TAPRD
0332 FE 57       1410         CPI   'W'
0334 CA 51 03    1415         JZ    TAPWR
0337 FE 59       1420         CPI   'Y'
0339 CA 5C 03    1425         JZ    TTYLIS
033C C3 AA 02    1430         JMP   BADCMD
033F CD 62 03    1435 TAPAP   CALL  SETAP       * SETUP TO APPEND DATA
0342 CD 75 03    1440         CALL  TRDATA      * READ IT
0345 C3 F8 02    1445         JMP   CAS1        * SET LENGTH & GO BACK
0348 21 4E 08    1450 TAPRD   LXI   H,BUFFER
034B CD 75 03    1455         CALL  TRDATA      * READ DATA
034E C3 11 03    1460         JMP   CAS2        * SET LENGTH & GO BACK
0351 CD 6A 03    1465 TAPWR   CALL  SETWR       * SETUP BYTE COUNT IN DE
0354 13          1470         INX   D           * INCLUDE FRAME COUNT BYTES
0355 13          1475         INX   D
0356 CD A4 03    1480         CALL  TPDATA      * PUNCH IT
0359 C3 C4 01    1485         JMP   CT3A        * & GO BACK TO MONITOR
035C CD 61 07    1490 TTYLIS  CALL  TTYOUT      * LIST DATA ON TTY
035F C3 C4 01    1495         JMP   CT3A
0362 CD 6A 03    1500 SETAP   CALL  SETWR       * BYTE COUNT ALREADY IN BUFFER
0365 21 4E 08    1505         LXI   H,BUFFER    * + BUFFER ADDR
0368 19          1510         DAD   D
0369 C9          1515         RET
036A 2A 4C 08    1520 SETWR   LHLD  LENGTH      * (LENGTH)*9 INTO DE
036D 54          1525 SW2     MOV   D,H
036E 5D          1530         MOV   E,L
036F 29          1535         DAD   H
0370 29          1540         DAD   H
0371 29          1545         DAD   H
0372 19          1550         DAD   D
0373 EB          1555         XCHG
0374 C9          1560         RET
0375             1565 *
0375             1570 * TTY AND PAPER TAPE I/O ROUTINES
0375             1575 *
0375 E5          1580 TRDATA  PUSH  H
0376 AF          1585         XRA   A
0377 CD 62 07    1590         CALL  PTRCLR      * CLEAR READER STATUS BITS
037A CD 5B 07    1595 TR1     CALL  KEYTST
037D C0          1600         RNZ   *           * KEYBOARD INTERRUPTION
037E CD 6B 07    1605         CALL  PTRIN       * IGNORE THE 1ST CHAR
0381 CA 7A 03    1610         JZ    TR1         * IF IT TAKES FOREVER
0384 CD 6B 07    1615 TR2     CALL  PTRIN       * IGNORE NULLS (LEADER)
0387 C8          1620         RZ    *           * ZERO IF TIMER RAN OUT
0388 B7          1625         ORA   A
0389 CA 84 03    1630         JZ    TR2
038C 21 56 00    1635         LXI   H,PLACNT    * ALSO IGNORE 1ST NON-0 BYTE
038F CD 6B 07    1640         CALL  PTRIN       * NOW READ 2 BYTES
0392 C8          1645         RZ
0393 77          1650         MOV   M,A         * INTO PLACNT
0394 23          1655         INX   H
0395 CD 6B 07    1660         CALL  PTRIN
0398 C8          1665         RZ
0399 77          1670         MOV   M,A         * AND PLACNT+1
039A E1          1675         POP   H
```

```
ADDR B1 B2 B3 E LINE  LABEL    OPCD  OPERAND

039B CD 6B 07   1680 TR3      CALL  PTRIN      * THEN READ THE REST
039E C8         1685          RZ
039F 77         1690          MOV   M,A        * INTO THE BUFFER
03A0 23         1695          INX   H
03A1 C3 9B 03   1700          JMP   TR3
03A4            1701 *
03A4            1702 * PAPER TAPE PUNCH ROUTINE (PRESENTLY UNIMPLEMENTED)
03A4            1703 *
03A4 C9         1705 TPDATA RET
03A5            1710 *
03A5            1715 * CASSETTE I/O ROUTINES
03A5            1720 *
03A5 E5         1725 CRDATA PUSH  H          * SAVE DATA START ADDR
03A6 3E 10      1730          MVI   A,10H
03A8 CD 81 07   1735          CALL  CASCLR     * INITIALIZE CASSETTE READER
03AB 21 56 00   1740          LXI   H,PLACNT
03AE CD 86 07   1745          CALL  CASIN      * READ FRAME COUNT INTO PLACNT
03B1 CD 86 07   1750          CALL  CASIN
03B4 2A 56 00   1755          LHLD  PLACNT
03B7 CD 6D 03   1760          CALL  SW2        * MULTIPLY BY 9 & INTO DE
03BA E1         1765          POP   H
03BB CD 86 07   1770 CR1      CALL  CASIN      * READ THE DATA
03BE 1B         1775          DCX   D
03BF AF         1780          XRA   A
03C0 B2         1785          ORA   D
03C1 B3         1790          ORA   E
03C2 C2 BB 03   1795          JNZ   CR1
03C5 C9         1800          RET
03C6 CD 81 07   1805 CWDATA CALL  CASCLR     * WRITE SOME LEADER
03C9 21 4C 00   1810          LXI   H,BUFFER-2
03CC 06 00      1815          MVI   B,0
03CE 7E         1820 CW1      MOV   A,M
03CF CD A7 07   1825          CALL  CASOUT     * WRITE THE DATA
03D2 23         1830          INX   H
03D3 1B         1835          DCX   D
03D4 7A         1845          MOV   A,D
03D5 B3         1850          ORA   E
03D6 C2 CE 03   1855          JNZ   CW1
03D9 78         1860          MOV   A,B
03DA CD A7 07   1865          CALL  CASOUT
03DD CD 53 06   1870          CALL  DLY25      * WRITE SOME TRAILER
03E0 AF         1875          XRA   A
03E1 CD 81 07   1880          CALL  CASCLR
03E4 C9         1885          RET
03E5            1890 *
03E5            1895 * PLAY & REPEAT PROCESSORS
03E5            1900 *
03E5 3E 01      1905 REPEAT MVI   A,1
03E7 C3 EB 03   1910          JMP   PLAY+1     * SET REPEAT FLAG TO 1
03EA AF         1915 PLAY     XRA   A          * CLEAR REPEAT FLAG
03EB 32 53 00   1920          STA   RPTFLG
03EE CD 18 05   1925 PL1      CALL  CTINIT     * TURN ON CT-1
03F1 2A 54 00   1930          LHLD  PLABGN
03F4 CD F2 05   1935          CALL  FRTOAD     * CONVERT BEGIN FRAME
03F7 22 58 00   1940          SHLD  FRMADR     * TO BUFFER ADDRESS
03FA 2A 56 00   1945          LHLD  PLACNT
```

```
ADDR B1 B2 B3 E LINE  LABEL    OPCD  OPERAND

03FD 22 5A 00   1950          SHLD  FRMCTR     * SET FRAME COUNTER
0400 CD 5B 07   1955 PL2      CALL  KEYTST     * ANY KEYS BEEN HIT?
0403 CA 0D 04   1960          JZ    PL3        * NO
0406 AF         1965 PL2A     XRA   A
0407 32 53 00   1970          STA   RPTFLG     * YES, KILL ANY REPEATS
040A C3 25 04   1975          JMP   PL4
040D 3A 49 00   1980 PL3      LDA   FRTIME
0410 6F         1985          MOV   L,A
0411 26 00      1990          MVI   H,0
0413 CD 56 06   1995          CALL  DLYHL      * DELAY (FRTIME) MSEC
0416 CD 29 05   2000          CALL  CTOUT      * PLAY 1 DATA FRAME
0419 C2 00 04   2005          JNZ   PL2
041C 3A 49 00   2010          LDA   FRTIME
041F 6F         2015          MOV   L,A
0420 26 00      2020          MVI   H,0
0422 CD 56 06   2025          CALL  DLYHL      * WAIT FOR LAST FRAME
0425 CD 4F 05   2030 PL4      CALL  CTOFF      * TURN OFF CT-1
0428 3A 53 00   2035          LDA   RPTFLG
042B B7         2040          ORA   A
042C CA 04 03   2045          JZ    CTSA       * NO REPEAT, GO BACK
042F 2A 46 00   2050          LHLD  RPTIME
0432 EB         2055          XCHG
0433 21 01 00   2060 PL5      LXI   H,1
0436 7A         2070          MOV   A,D
0437 B3         2075          ORA   E
0438 CA EE 03   2080          JZ    PL1        * END OF WAIT, PLAY AGAIN
043B CD 56 06   2085          CALL  DLYHL
043E CD 5B 07   2090          CALL  KEYTST
0441 C2 06 04   2095          JNZ   PL2A       * KEY "INTERRUPT", GO BACK
0444 1B         2100          DCX   D
0445 C3 33 04   2105          JMP   PL5
0448            2110 *
0448            2115 * PARAMETER MODIFY, DITTO & NEXT LINE PROCESSORS
0448            2120 *
0448 2A 43 00   2125 DITTO    LHLD  LAST       * SET NUM. BACK TO PREV VALUE
044B 22 45 00   2130          SHLD  NUMBER
044E 2A 4F 00   2135 PRMMOD LHLD  CL
0451 CD F2 05   2140          CALL  FRTOAD     * COMPUTE ADDR OF CURRENT FRAM
0454 3A 4D 00   2145          LDA   CPN
0457 5F         2150          MOV   E,A
0458 16 00      2155          MVI   D,0
045A 19         2160          DAD   D          * ADD CURRENT PARAM
045B 3A 45 00   2165          LDA   NUMBER
045E 77         2170          MOV   M,A        * AND SET DATA = NUMBER
045F 1E 07      2175          MVI   E,7
0461 2A 4F 00   2180          LHLD  CL
0464 44         2185          MOV   B,H
0465 4D         2190          MOV   C,L
0466 CD 53 05   2195          CALL  DISFRM     * REDISPLAY CURRENT FRAME
0469 3E 20      2200 NEXT     MVI   A,
046B CD 63 05   2205          CALL  ARROW
046E 2A 4F 00   2210          LHLD  CL         * SEE IF WE CAN BUMP CL
0471 23         2215          INX   H
0472 CD [illegible] 2220      CALL  LINKNG
0475 DA 82 04   2225          JC    CTS        * TOO BIG, WE CAN'T GO FORWARD
0478 22 4F 00   2230          SHLD  CL         * IT'S OK, SET CL=CL+1
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

047B 01 04 00    2235           LXI   B,4
047E 09          2240           DAD   B
047F 44          2245           MOV   B,H        * SAVE NUMBER OF NEW
0480 4D          2250           MOV   C,L        * BOTTOM LINE IN BC
0481 1E 04       2255           MVI   E,4
0483 16 08       2260           MVI   D,8
0485 CD B9 06    2265 NX1       CALL  LINUP      * MOVE ALL BUT TOP LINE UP
0488 1C          2270           INR   E
0489 15          2275           DCR   D
048A C2 85 04    2280           JNZ   NX1
048D 2A 4C 00    2285           LHLD  LENGTH     * COMPL. LENGTH = -(LENGTH)-1 /
0490 7C          2290           MOV   A,H
0491 2F          2295           CMA
0492 67          2300           MOV   H,A
0493 7D          2305           MOV   A,L
0494 2F          2310           CMA
0495 6F          2315           MOV   L,A
0496 09          2320           DAD   B          * COMPUTE BOTTOM # -(LENGTH)-1
0497 1E 0B       2325           MVI   E,0BH      * COORDINATE FOR BOTTOM LINE
0499 7C          2330           MOV   A,H
049A C3 C6 04    2335           JMP   BK2
049D             2340 *
049D             2345 * MOVE BACK 1 FRAME
049D             2350 *
049D 3E 20       2355 BACK      MVI   A,' '
049F CD B3 05    2360           CALL  ARROW
04A2 2A 4F 00    2365           LHLD  CL         * TRY DECREMENTING CL
04A5 2B          2370           DCX   H
04A6 CD D1 05    2375           CALL  LIMRNG
04A9 FA B9 01    2380           JM    CT3        * =0, WE CAN'T MOVE BACK
04AC 22 4F 00    2385           SHLD  CL         * OK, SET CL=CL-1
04AF 1E 0A       2390           MVI   E,0AH
04B1 06 08       2395           MVI   B,8
04B3 CD BF 06    2400 BK1       CALL  LINDN      * MOVE LINES 3-10 DOWN A NOTCH
04B6 1D          2405           DCR   E
04B7 05          2410           DCR   B
04B8 C2 B3 04    2415           JNZ   BK1
04BB 11 FB FF    2420           LXI   D,-5
04BE 19          2425           DAD   D          * COMPUTE FRAME NO. OF TOP LINE
04BF 7C          2430           MOV   A,H
04C0 2F          2435           CMA
04C1 23          2440           INX   H
04C2 44          2445           MOV   B,H
04C3 4D          2450           MOV   C,L
04C4 1E 03       2455           MVI   E,3        * SET DISFRM LINE NO.
04C6 B7          2460 BK2       ORA   A          * IF OFF END-OF-BUFFER, S=0
04C7 F4 A1 06    2465           CP    CLRLIN     * CLRLIN ALWAYS RETURNS WITH S=0
04CA FC 53 05    2470           CM    DISFRM     * IF S=1, DISPLAY THE FRAME
04CD C3 B9 01    2475           JMP   CT3
04D0             2480 *
04D0             2485 * INSERT AND DELETE WILL BE IMPLEMENTED
04D0             2490 * IN A LATER VERSION
04D0             2495 *
04D0 C3 50 01    2500 INSERT JMP   CT0
04D3 C3 50 01    2505 DELETE JMP   CT0
04D6             2510 *
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

04D6             2515 * MISC COMMAND PROCESSORS
04D6             2520 *
04D6 3E 20       2525 MOVE      MVI   A,' '
04D8 CD B3 05    2530           CALL  ARROW
04DB 2A 45 00    2535           LHLD  NUMBER     * SET CURR. LINE TO NUMERIC INPUT
04DE 22 4F 00    2540           SHLD  CL
04E1 C3 63 01    2545           JMP   CT1
04E4 2A 45 00    2550 AAAG      LHLD  NUMBER     * SET FRAME DELAY INTERVAL
04E7 7C          2555           MOV   A,H
04E8 B7          2560           ORA   A
04E9 CA EE 04    2565           JZ    AA2
04EC 2E FF       2570           MVI   L,0FFH
04EE 7D          2575 AA2       MOV   A,L        * MIN (NUMBER,255)
04EF B7          2580           ORA   A
04F0 C2 F4 04    2585           JNZ   AA3
04F3 3C          2590           INR   A          * AND IT MUST BE >0
04F4 32 49 00    2595 AA3       STA   FRTIME
04F7 C3 C4 01    2600           JMP   CT3A
04FA 2A 45 00    2605 START     LHLD  NUMBER     * FIX START FRAME TO LEGAL RANGE
04FD CD D1 05    2610           CALL  LIMRNG
0500 22 54 00    2615           SHLD  PLABGN
0503 C3 C4 01    2620           JMP   CT3A
0506 2A 45 00    2625 LENG      LHLD  NUMBER     * SET LENGTH
0509 22 56 00    2630           SHLD  PLACNT     * (NO RANGE CHECK)
050C C3 C4 01    2635           JMP   CT3A
050F 2A 45 00    2640 WAIT      LHLD  NUMBER     * SET INTER-REPEAT DELAY
0512 22 48 00    2645           SHLD  RPTIME     * (NO RANGE CHECK)
0515 C3 C4 01    2650           JMP   CT3A
0518             2655 * ===============================================*
0518             2660 *  SUBROUTINES                                   *
0518             2665 *                                                *
0518             2670 *  COMPUTALKER HANDLER ROUTINES                  *
0518             2675 *  INITIALIZATION:  SETUP CT-1 FROM 1ST DATA FRAME *
0518             2680 *  AND TURN ON THE AUDIO SWITCH                  *
0518             2685 * ===============================================*
0518             2686 *
0518 2A 54 00    2690 CTINIT LHLD  PLABGN     * SET ARGS TO PLAY 1ST FRAME
051B CD F2 05    2695           CALL  FRTOAD
051E 22 58 00    2700           SHLD  FRMADR
0521 CD 29 05    2705           CALL  CTOUT      * PLAY 1ST FRAME
0524 3E FF       2710           MVI   A,0FFH
0526 D3 EF       2715           OUT   CTBASE+0FH * THEN TURN ON THE AUDIO
0528 C9          2720           RET
0529             2725 *
0529             2730 *  COMPUTALKER HANDLER ROUTINES
0529             2735 *  PLAY 1 DATA FRAME
0529             2740 *    ADDRESS OF FRAME TO PLAY IS IN FRMADR
0529             2745 *    (FRNCTR) IS CHECKED TO SEE IF THIS FRAME
0529             2750 *    WAS THE LAST TO BE PLAYED
0529             2755 *    BOTH (FRMADR) AND (FRNCTR) ARE UPDATED
0529             2790 *
0529 2A 5A 00    2795 CTOUT     LHLD  FRNCTR     * GET FRAME COUNT IN DE
052C EB          2800           XCHG
052D 2A 58 00    2805           LHLD  FRMADR     * GET DATA ADDR IN HL
0530 06 09       2810           MVI   B,9
0532 3E E0       2815           MVI   A,CTBASE
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

0534 32 41 00   2820 CT01   STA   PLADAT+1   * SET CT-1 OUTPUT PORT
0537 7E         2825        MOV   A,M
0538 CD 40 00   2830        CALL  PLADAT
053B 23         2835        INX   H
053C 3A 41 00   2840        LDA   PLADAT+1
053F 3C         2845        INR   A
0540 05         2850        DCR   B
0541 C2 34 05   2855        JNZ   CT01       * LOOP FOR 9 PARAMS
0544 22 58 00   2860        SHLD  FRMADR     * UPDATE ADDRESS
0547 1B         2865        DCX   D
0548 EB         2870        XCHG
0549 22 5A 00   2875        SHLD  FRMCTR     * UPDATE COUNT
054C 7C         2885        MOV   A,H
054D B5         2890        ORA   L          * Z FLAG SET IF DONE
054E C9         2895        RET
054F            2900 *
054F            2905 *  COMPUTALKER HANDLER ROUTINES
054F            2910 *  TURN OFF CT-1
054F            2915 *
054F AF         2920 CTOFF  XRA   A
0550 D3 EF      2925        OUT   CTBASE+0FH
0552 C9         2930        RET
0553            2935 *
0553            2940 *  DISPLAY 1 FRAME OF DATA
0553            2945 *
0553 D5         2950 DISFRM PUSH  D          * PUSH SCREEN LINE NO (IN E)
0554 C5         2955        PUSH  B          * PUSH FRAME NO.
0555 C5         2960        PUSH  B          * MAKE 2 COPIES, WE'LL USE 1 HERE
0556 16 01      2965        MVI   D,1        * HORIZ CHAR POS. =1
0558 CD E1 06   2970        CALL  VBUFAD     * CONVERT TO VIDEO BUF ADDR
055B 60         2975        MOV   H,B        * GET FRAME NO. IN HL
055C 69         2980        MOV   L,C
055D 06 20      2985        MVI   B,' '
055F CD 07 07   2990        CALL  DISPI4     * DISPLAY (H) IN LEFTMOST COLUMN
0562 13         2995        INX   D
0563 13         3000        INX   D
0564 13         3005        INX   D          * BUMP DE TO 1ST DATA COL.
0565 C1         3010        POP   B
0566 60         3015        MOV   H,B
0567 69         3020        MOV   L,C        * THE ADDR OF THE DATA IS
0568 29         3025        DAD   H          *    (FRAME NO.-1)*9+DATA ADDRESS
0569 29         3030        DAD   H          *    WHICH EQUALS
056A 29         3035        DAD   H
056B 09         3040        DAD   B          *    FRAME NO. *9
056C 01 45 00   3045        LXI   B,BUFFER-9 *    + BUFFER ADDR-9
056F 09         3050        DAD   B
0570 06 20      3055        MVI   B,' '
0572 CD A8 05   3060        CALL  DISDAT     * DISPLAY AV
0575 CD A8 05   3065        CALL  DISDAT     * DISPLAY F0
0578 3A 4D 00   3070        LDA   CPM
057B FE 05      3075        CPI   5
057D FA 92 05   3080        JM    DFORM      * CPM<5, DISPLAY FORMANTS
0580 23         3085        INX   H
0581 23         3090        INX   H
0582 23         3095        INX   H          * BUMP HL PAST FORMANTS
0583 CD A8 05   3100        CALL  DISDAT     * DISPLAY AH
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

0586 CD A8 05   3105        CALL  DISDAT     * DISPLAY AF
0589 CD A8 05   3110        CALL  DISDAT     * DISPLAY FF
058C CD A8 05   3115        CALL  DISDAT     * DISPLAY AN
058F C1         3120        POP   B
0590 D1         3125        POP   D
0591 C9         3130        RET
0592 CD A8 05   3135 DFORM  CALL  DISDAT     * DISPLAY F1
0595 CD A8 05   3140        CALL  DISDAT     * DISPLAY F2
0598 CD A8 05   3145        CALL  DISDAT     * DISPLAY F3
059B 78         3150        MOV   A,B        * GET BLANK IN A
059C F6 80      3155        ORI   128        * SET MSB HIGH
059E 12         3160        STAX  D          * & PUT IN NEXT 4 POSITIONS
059F 13         3165        INX   D
05A0 12         3170        STAX  D          * IT'S NOT WORTH A LOOP
05A1 13         3175        INX   D
05A2 12         3180        STAX  D
05A3 13         3185        INX   D
05A4 12         3190        STAX  D
05A5 C1         3195        POP   B
05A6 D1         3200        POP   D
05A7 C9         3205        RET
05A8            3210 *
05A8            3215 *  DISPL. 1 PARAMETER DATA ITEM
05A8            3220 *
05A8 E5         3225 DISDAT PUSH  H          * SAVE DATA ADDR
05A9 6E         3230        MOV   L,M        * GET DATA VALUE IN HL
05AA 26 00      3235        MVI   H,0
05AC CD 07 07   3240        CALL  DISPI4     * DISPLAY DECIMAL
05AF 13         3245        INX   D
05B0 E1         3250        POP   H
05B1 23         3255        INX   H
05B2 C9         3260        RET
05B3            3265 *
05B3            3270 *  CHAR IN A IS DISPLAYED RIGHT OF CURRENT PARAM
05B3            3275 *     ALL REGISTERS SAVED
05B3            3280 *
05B3 D5         3285 ARROW  PUSH  D
05B4 F5         3290        PUSH  PSW
05B5 3A 4D 00   3295        LDA   CPM
05B8 FE 05      3300        CPI   5
05BA FA BF 05   3305        JM    AR2        * ADJUST HOR. CHAR POS. FOR
05BD D6 03      3310        SUI   3          * EITHER FORMANTS OR AMPL
05BF 57         3315 AR2    MOV   D,A
05C0 87         3320        ADD   A
05C1 87         3325        ADD   A
05C2 82         3330        ADD   D
05C3 C6 0C      3335        ADI   0CH        * ADJUSTED CPM *5+12
05C5 57         3340        MOV   D,A
05C6 1E 07      3345        MVI   E,7        * ALWAYS ON LINE 7
05C8 CD E1 06   3350        CALL  VBUFAD     * CONVERT TO VIDEO ADDR
05CB F1         3355        POP   PSW
05CC F6 80      3360        ORI   128        * SET MSB HIGH
05CE 12         3365        STAX  D          * PUT CHAR ON SCREEN
05CF D1         3370        POP   D
05D0 C9         3375        RET
05D1            3380 *
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

05D1            3385 * LIMRNG:  FORCE (HL) TO WITHIN THE LIMITS 1<=X<=(LENGTH)
05D1            3390 * IF (HL) WAS CHANGED, A IS NEG ON RETURN & FLAG S IS SET
05D1            3395 *  ELSE A IS POS. & S=0
05D1            3400 *  REG B,C,D & E RESTORED
05D1            3405 *
05D1 2B         3410 LIMRNG DCX    H
05D2 7C         3415        MOV    A,H
05D3 B7         3420        ORA    A             * SEE IF (HL)<1
05D4 F2 DB 05   3425        JP     LM1           * NO, IT'S > 0
05D7 21 01 00   3430        LXI    H,1
05DA C9         3435        RET    *             * YES, SET IT TO 1 & RETURN
05DB D5         3440 LM1    PUSH   D
05DC E5         3445        PUSH   H
05DD 7C         3450        MOV    A,H
05DE 2F         3455        CMA
05DF 57         3460        MOV    D,A
05E0 7D         3465        MOV    A,L
05E1 2F         3470        CMA
05E2 5F         3475        MOV    E,A           * PUT -(HL) IN DE
05E3 2A 4C 08   3480        LHLD   LENGTH
05E6 EB         3485        XCHG   *             * KEEP (LENGTH) IN DE
05E7 19         3490        DAD    D             * (LENGTH)-(HL) SHOULD BE +
05E8 7C         3495        MOV    A,H
05E9 B7         3500        ORA    A
05EA E1         3505        POP    H             * GET ORIGINAL (HL) BACK
05EB 23         3510        INX    H
05EC F2 F0 05   3515        JP     LM2
05EF EB         3520        XCHG   *             * IT WAS TOO BIG, USE (LENGTH)
05F0 D1         3525 LM2    POP    D
05F1 C9         3530        RET
05F2            3535 *
05F2            3540 *  CONVERT FRAME NO. TO BUFFER ADDRESS
05F2            3545 *
05F2 CD 6D 03   3550 FRTOAD CALL   SW2
05F5 21 45 08   3555        LXI    H,BUFFER-9
05F8 19         3560        DAD    D
05F9 C9         3565        RET
05FA            3570 *
05FA            3575 *  CONVERT BINARY (HL) TO BCD IN HL
05FA            3580 *
05FA C5         3585 BINBCD PUSH   B
05FB D5         3590        PUSH   D
05FC 11 00 00   3595        LXI    D,0           * CLEAR DIGIT REGISTERS DE
05FF 01 18 FC   3600        LXI    B,-1000
0602 AF         3605        XRA    A
0603 09         3610 D1     DAD    B             * SUBTRACT 1000 FROM NUMBER
0604 B4         3615        ORA    H             * SET A7 IF H NEG
0605 FA 17 06   3620        JM     D2
0608 13         3625        INX    D             * STILL +, INCR DIGIT
0609 7B         3630        MOV    A,E
060A FE 0A      3635        CPI    0AH
060C C2 03 06   3640        JNZ    D1
060F 21 00 00   3645        LXI    H,0           * ERROR RETURN
0612 F6 FF      3650        ORI    0FFH
0614 C3 50 06   3655        JMP    POPS
0617 7A         3660 D2     MOV    A,E           * MULTIPLY COUNT BY 16
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

0618 87         3665        ADD    H             * (IE. SHIFT IT LEFT 1 DIGIT)
0619 87         3670        ADD    H
061A 87         3675        ADD    H
061B 87         3680        ADD    H
061C 5F         3685        MOV    E,A
061D 01 E8 03   3690        LXI    B,1000
0620 09         3695        DAD    B
0621 01 9C FF   3700        LXI    B,-100
0624 AF         3705        XRA    A
0625 09         3710 D2A    DAD    B             * SUBTRACT 100 FROM NUMBER
0626 B4         3715        ORA    H
0627 FA 2E 06   3720        JM     D3
062A 13         3725        INX    D             * TILL +, INCR 2ND DIGIT
062B C3 25 06   3730        JMP    D2A
062E EB         3735 D3     XCHG   *             * WE'LL USE HL TO MULTIPLY NOW
062F 29         3740        DAD    H             * (HL)=2*PREVIOUS 2 DIGITS
0630 29         3745        DAD    H             *   4*
0631 29         3750        DAD    H             *   8*
0632 29         3755        DAD    H             *  16*
0633 EB         3760        XCHG   *             * RESULT BACK TO DE
0634 01 64 00   3765        LXI    B,100
0637 09         3770        DAD    B
0638 01 F6 FF   3775        LXI    B,-10
063B AF         3780        XRA    A
063C 09         3785 D3A    DAD    B             * SUBTRACT 10 FROM THE NUMBER
063D B4         3790        ORA    H
063E FA 45 06   3795        JM     D4
0641 13         3800        INX    D             * STILL +, INCR 3RD DIGIT
0642 C3 3C 06   3805        JMP    D3A
0645 01 0A 00   3810 D4     LXI    B,10          * ADD 10 BACK TO NUMBER
0648 09         3815        DAD    B
0649 EB         3820        XCHG
064A 29         3825        DAD    H             * MULTIPLY DIGITS BY 16 AGAIN
064B 29         3830        DAD    H
064C 29         3835        DAD    H
064D 29         3840        DAD    H
064E 19         3845        DAD    D             * ADD ON LAST DIGIT (THE NUMBER)
064F AF         3850        XRA    A             * CLEAR ACC (MEANS ALL'S WELL)
0650 D1         3855 POPS   POP    D
0651 C1         3860        POP    B
0652 C9         3865        RET
0653            3870 *
0653            3875 *  PROGRAM LOOP DELAY SUBROUTINES
0653            3880 *  ASSUME BASIC 8080 CLOCK RATE OF 2 MHZ
0653            3885 *  DELAY 2 SECONDS
0653            3890 *
0653 21 D0 07   3895 DLY2S  LXI    H,2000
0656            3900 *
0656            3905 *  DELAY (HL) MSEC
0656            3910 *
0656 F5         3915 DLYHL  PUSH   PSW
0657 C3 60 06   3920        JMP    DLY2
065A 3E 06      3925 DLY1   MVI    A,6
065C 3D         3930        DCR    A
065D C2 5C 06   3935        JNZ    $-1
0660 3E 7C      3940 DLY2   MVI    A,124
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

0662 3D          3945           DCR   A
0663 C2 62 06    3950           JNZ   $-1
0666 2B          3955           DCX   H
0667 B4          3960           ORA   H
0668 B5          3965           ORA   L
0669 C2 5A 06    3970           JNZ   DLY1
066C F1          3975           POP   PSW
066D C9          3980           RET
066E             3985 *
066E             3990 * GET A COMMAND CHAR FROM KBD INTO A
066E             3995 * AND DISPLAY IT IN THE COMMAND LINE
066E             4000 *
066E E5          4005 GETCMC PUSH H
066F CD 4F 07    4010           CALL  GETC
0672 FE 1B       4015           CPI   1BH        * CHECK FOR ESC
0674 CA 8A 06    4020           JZ    GC1
0677 FE 7F       4025           CPI   7FH        * CHECK FOR RUBOUT
0679 CA 8A 06    4030           JZ    GC1
067C 2A 51 00    4035           LHLD  CMDADR
067F F5          4040           PUSH  PSW
0680 F6 80       4045           ORI   128        * SET MSB HIGH
0682 77          4050           MOV   M,A
0683 F1          4055           POP   PSW
0684 23          4060           INX   H
0685 22 51 00    4065           SHLD  CMDADR
0688 E1          4070           POP   H
0689 C9          4075           RET
068A 31 FF 00    4080 GC1       LXI   SP,STACK   * RESET STACK & BACK TO LEVEL 0
068D C3 C4 01    4085           JMP   CT3A
0690             4086 *
0690             4090 *  VIDEO DISPLAY SUBROUTINES:
0690             4091 *
0690             4095 *  CLEAR THE SCREEN
0690             4096 *
0690 F5          4100 CLRVID PUSH PSW
0691 E5          4105           PUSH  H
0692 21 00 88    4110           LXI   H,256*VIDBUF
0695 36 A0       4115 CLR       MVI   M,' '+128  * SET MSB HIGH
0697 23          4120           INX   H
0698 7C          4125           MOV   A,H
0699 FE 8C       4130           CPI   VIDBUF+4   * COMPARE FOR END OF PAGE
069B C2 95 06    4135           JNZ   CLR
069E E1          4140           POP   H
069F F1          4145           POP   PSW
06A0 C9          4150           RET
06A1             4151 *
06A1             4155 *  CLEAR ONE LINE ACROSS THE SCREEN
06A1             4156 *
06A1 E5          4160 CLRLIN PUSH H
06A2 D5          4165           PUSH  D
06A3 16 00       4170           MVI   D,0
06A5 CD E1 06    4175           CALL  VBUFAD
06A8 EB          4180           XCHG
06A9 16 40       4185           MVI   D,40H
06AB 36 A0       4190 CLN       MVI   M,' '+128  * SET MSB HIGH
06AD 23          4195           INX   H
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

06AE 15          4200           DCR   D
06AF C2 AB 06    4205           JNZ   CLN
06B2 D1          4210           POP   D
06B3 E1          4215           POP   H
06B4 C9          4220           RET
06B5             4221 *
06B5             4225 *  MOVE 1 LINE UP OR DOWN (E) SCREEN LINES (+UP -DOWN)
06B5             4226 *
06B5 F5          4230 MOVLIN PUSH PSW
06B6 C3 C2 06    4235           JMP   LIN1
06B9             4236 *
06B9             4240 *  MOVE 1 ASCII LINE UP OR DOWN 1 LINE ON SCREEN
06B9             4241 *
06B9 F5          4245 LINUP     PUSH  PSW        * MOVE A LINE UP
06BA 3E FF       4250           MVI   A,-1
06BC C3 C2 06    4255           JMP   LIN1
06BF F5          4260 LINDN     PUSH  PSW        * MOVE A LINE DOWN
06C0 3E 01       4265           MVI   A,1
06C2 E5          4270 LIN1      PUSH  H
06C3 D5          4275           PUSH  D
06C4 C5          4280           PUSH  B
06C5 83          4285           ADD   E
06C6 5F          4290           MOV   E,A        * DESTINATION LINE # IN E
06C7 16 00       4295           MVI   D,0        * SOURCE COORDINATES IN DE
06C9 CD E1 06    4300           CALL  VBUFAD     * CONVERT TO VIDEO BUF. ADDR
06CC EB          4305           XCHG  *          * & PUT IN HL
06CD 16 00       4310           MVI   D,0
06CF CD E1 06    4315           CALL  VBUFAD     * DESTINATION COORDINATES IN DE
06D2 06 40       4320           MVI   B,40H
06D4 7E          4325 LIN2      MOV   A,M
06D5 12          4330           STAX  D
06D6 23          4335           INX   H
06D7 13          4340           INX   D
06D8 05          4345           DCR   B
06D9 C2 D4 06    4350           JNZ   LIN2
06DC C1          4355           POP   B
06DD D1          4360           POP   D
06DE E1          4365           POP   H
06DF F1          4370           POP   PSW
06E0 C9          4375           RET
06E1             4376 *
06E1             4380 *  CONVERT VIDEO COORDS (LINE NO. & CHAR POS.)
06E1             4385 *      TO VIDEO BUFFER ADDRESS
06E1             4386 *
06E1 F5          4395 VBUFAD PUSH PSW
06E2 E5          4400           PUSH  H
06E3 7A          4405           MOV   A,D        * GET CHAR POS
06E4 E6 3F       4410           ANI   3FH
06E6 57          4435           MOV   D,A        * SAVE CHAR POS IN D
06E7 6B          4440           MOV   L,E        * GET LINE NO.
06E8 29          4445           DAD   H
06E9 29          4450           DAD   H
06EA 29          4455           DAD   H
06EB 29          4460           DAD   H          * SHIFT 4 LEFT
06EC 26 22       4465           MVI   H,VIDBUF/4 * PUT VIDEO ADDR IN HIGH BITS
06EE 29          4470           DAD   H
```

```
ADDR B1 B2 B3 E LINE  LABEL    OPCD  OPERAND

06EF 29          4475          DAD   H          * SHIFT 2 MORE
06F0 7D          4480          MOV   A,L
06F1 82          4485          ADD   D          * ADD CHAR POS
06F2 6F          4490          MOV   L,A
06F3 EB          4495          XCHG  *          * LEAVE RESULT IN DE
06F4 E1          4500          POP   H
06F5 F1          4505          POP   PSW
06F6 C9          4510          RET
06F7             4511 *
06F7             4515 * DISPLAY A LINE OF TEXT BEGINNING AT A GIVEN SCREEN COORD
06F7             4516 *
06F7 CD E1 06    4520 DSTX     CALL  VBUFAD     * CONVERT COORD TO VIDEO ADDRESS
06FA 7E          4525          MOV   A,M
06FB F6 80       4530          ORI   128        * SET MSB HIGH
06FD 12          4535          STAX  D          * PUT A CHAR IN SCREEN BUFFER
06FE 23          4540          INX   H
06FF 13          4545          INX   D
0700 0B          4550          DCX   B          * DECREMENT BYTE COUNT
0701 78          4560          MOV   A,B
0702 B1          4565          ORA   C
0703 C2 FA 06    4570          JNZ   DSTX+3     * NOT DONE YET
0706 C9          4575          RET
0707             4576 *
0707             4580 *  CONVERT (HL) TO DECIMAL & DISPLAY AT VIDEO BUFF ADDR
0707             4585 *  IN DE.  DISPLAY FORMAT(I4) WITH LEADING CHAR =(B)
0707             4586 *
0707 F5          4590 DISPI4 PUSH  PSW
0708 C5          4595          PUSH  B
0709 CD FA 05    4600          CALL  BINBCD
070C EB          4605          XCHG  *          * BCD DAT TO DE, VIDEO ADDR TO HL
070D 0E 30       4610          MVI   C,'0'
070F 7A          4615          MOV   A,D
0710 0F          4620          RRC
0711 0F          4625          RRC
0712 0F          4630          RRC
0713 0F          4635          RRC
0714 CD 2D 07    4640          CALL  BCDASC     * CONVERT BCD TO ASCII & DISPLAY
0717 7A          4645          MOV   A,D
0718 CD 2D 07    4650          CALL  BCDASC
071B 7B          4655          MOV   A,E
071C 0F          4660          RRC
071D 0F          4665          RRC
071E 0F          4670          RRC
071F 0F          4675          RRC
0720 CD 2D 07    4680          CALL  BCDASC
0723 7B          4685          MOV   A,E
0724 0E 00       4690          MVI   C,0        * KILL LEAD CHAR FOR LAST DIGIT
0726 CD 2D 07    4695          CALL  BCDASC
0729 EB          4700          XCHG
072A C1          4705          POP   B
072B F1          4710          POP   PSW
072C C9          4715          RET
072D CD 43 07    4720 BCDASC   CALL  KAASC
0730 B9          4725          CMP   C
0731 C2 3C 07    4730          JNZ   DDIG
0734 F5          4735          PUSH  PSW
```

```
ADDR B1 B2 B3 E LINE  LABEL    OPCD  OPERAND

0735 79          4740          MOV   A,C
0736 F6 80       4745          ORI   128        * SET MSB HIGH
0738 77          4750          MOV   M,A
0739 F1          4755          POP   PSW
073A 23          4760          INX   H
073B C9          4765          RET
073C 0E 00       4770 DDIG     MVI   C,0
073E F6 80       4775          ORI   128        * SET MSB HIGH
0740 77          4780          MOV   M,A
0741 23          4785          INX   H
0742 C9          4790          RET
0743             4791 *
0743             4795 *  CONVERT KA(A) TO ASCII W/BIT 7 SET
0743             4796 *
0743 E6 0F       4800 KAASC    ANI   0FH
0745 D6 0A       4805          SUI   10
0747 DA 4C 07    4810          JC    DECIM
074A C6 07       4815          ADI   7
074C C6 3A       4820 DECIM    ADI   '9'+1
074E C9          4825          RET
074F             4826 *
074F             4830 *  KEYBOARD INPUT SUBROUTINES
074F             4831 *
074F             4835 KBF      EQU   0D8H
074F             4840 KBD      EQU   0D9H
074F             4841 *
074F             4845 *  WAIT FOR KBD STROBE & RETURN WITH ASCII CHAR IN A
074F             4846 *
074F DB D8       4850 GETC     IN    KBF
0751 E6 01       4855          ANI   1
0753 C2 4F 07    4860          JNZ   GETC
0756 DB D9       4865          IN    KBD
0758 E6 7F       4870          ANI   127
075A C9          4875          RET
075B             4876 *
075B             4880 *  TEST KBD
075B             4885 *  IF NO KEY PRESSED : Z=1
075B             4890 *  IF    KEY PRESSED : Z=0
075B             4891 *
075B DB D8       4895 KEYTST   IN    KBF
075D 2F          4900          CMA   *          * NOP FOR ACTIVE HIGH STATUS
075E E6 01       4905          ANI   1
0760 C9          4910          RET
0761             4911 *
0761             4915 *  TTY OUTPUT SUBROUTINES
0761             4916 *
0761             4920 TTYF     EQU   16
0761             4925 TTYD     EQU   17
0761 C9          4930 TTYOUT   RET
0762             4931 *
0762             4935 *  PAPER TAPE READER SUBROUTINES
0762             4936 *
0762             4940 PTRF     EQU   TTYF
0762             4945 PTRD     EQU   TTYD
0762             4947 *
0762             4950 *  INITIALIZE PAPER TAPE READER
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

0762              4951 *
0762 3E 03        4955 PTRCLR MVI    A,3
0764 D3 10        4960        OUT    TTYF
0766 3E 11        4965        MVI    A,11H
0768 D3 10        4970        OUT    TTYF
076A C9           4975        RET
076B              4976 *
076B              4980 *  WAIT FOR READY FLAG & READ 1 BYTE
076B              4981 *
076B C5           4985 PTRIN  PUSH   B
076C 01 00 40     4990        LXI    B,4000H     * SET TIMER
076F DB 10        4995 PTR1   IN     PTRF        * WAIT FOR FLAG
0771 00           5000        NOP    *           * CMA FOR ACTIVE LOW STATUS
0772 E6 01        5005        ANI    1
0774 C2 7D 07     5010        JNZ    PTR2
0777 0B           5015        DCX    B           * NOT READY, CHECK THE TIME OUT
0778 78           5020        MOV    A,B
0779 B1           5025        ORA    C
077A C2 6F 07     5030        JNZ    PTR1
077D DB 11        5035 PTR2   IN     PTRD        * CONDITION Z=1 IF TIME-OUT
077F C1           5040        POP    B
0780 C9           5045        RET
0781              5047 *
0781              5050 *  CASSETTE I/O SUBROUTINES FOR TARBELL CASSETTE
0781              5051 *
0781              5055 CASF   EQU    6EH
0781              5060 CASD   EQU    6FH
0781              5061 *
0781              5065 * PUT (A) INTO CASSETTE STATUS BITS & CLEAR B (FOR CHKSUM)
0781              5066 *
0781 D3 6E        5070 CASCLR OUT    CASF
0783 06 00        5075        MVI    B,0
0785 C9           5080        RET
0786              5081 *
0786              5085 *  READ 1 BYTE FROM CASSETTE
0786              5090 *  CALL WITH READ ADDRESS IN HL
0786              5095 *      1. READS ONE BYTE INTO (HL)
0786              5100 *      2. INCREMENTS HL
0786              5105 *      3. ACCUM CHECKSUM IN B, IE. SET B=B+BYTE READ IN
0786              5110 *      4. RESTORES A AND FLAGS
0786              5111 *
0786 F5           5115 CASIN  PUSH   PSW
0787 DB 6E        5120        IN     CASF
0789 E6 10        5125        ANI    10H
078B C2 87 07     5130        JNZ    CASIN+1
078E DB 6F        5135        IN     CASD
0790 77           5140        MOV    M,A
0791 80           5145        ADD    B
0792 47           5150        MOV    B,A
0793 23           5155        INX    H
0794 F1           5160        POP    PSW
0795 C9           5165        RET
0796              5166 *
0796              5170 *  WRITE CASSETTE LEADER, SYNC AND START BYTES
0796              5171 *
0796 3E 01        5175 CASLDR MVI    A,1
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

0798 D3 6E        5180        OUT    CASF        * TURN ON WRITE RELAY
079A CD 53 06     5185        CALL   DLY2S       * WRITE LEADER FOR 2 SEC
079D 3E 3C        5190        MVI    A,3CH
079F CD A7 07     5195        CALL   CASOUT      * WRITE START BYTE
07A2 3E E6        5200        MVI    A,0E6H
07A4 C3 A7 07     5205        JMP    CASOUT      * WRITE SYNC BYTE
07A7              5211 *
07A7              5215 *  WRITE ONE DATA BYTE ONTO CASSETTE
07A7              5220 *  CALL WITH DATA TO BE WRITTEN IN A
07A7              5225 *      1. WRITES ONE BYTE FROM A
07A7              5230 *      2. ACCUM CKSUM IN B, IE. SETS B=B+BYTE WRITTEN
07A7              5231 *
07A7 F5           5235 CASOUT PUSH   PSW
07A8 DB 6E        5240        IN     CASF
07AA E6 20        5245        ANI    20H
07AC C2 A8 07     5250        JNZ    CASOUT+1
07AF F1           5255        POP    PSW
07B0 D3 6F        5260        OUT    CASD
07B2 80           5265        ADD    B
07B3 47           5270        MOV    B,A
07B4 C9           5275        RET
07B5              5276 *
07B5              5280 *  CHAR TABLES, BRANCH TABLES, ETC TABLES, ETC.
07B5              5281 *
07B5 4D 48 57     5285 BRNCHR DT     'MHWSLID'
07B8 53 4C 49
07BB 44
07BC 4E 04        5290 BRNCTB DW     PRMMOD      * BRANCH=1
07BE D6 04        5295        DW     MOVE        *       =2
07C0 E4 04        5300        DW     HANG        *       =3
07C2 0F 05        5305        DW     WAIT        *       =4
07C4 FA 04        5310        DW     START       *       =5
07C6 06 05        5315        DW     LENG        *       =6
07C8 D0 04        5320        DW     INSERT      *       =7
07CA D3 04        5325        DW     DELETE      *       =8
07CC 56           5330 HPTAB  DB     'V'
07CD 00           5335        DB     0
07CE 48           5340        DB     'H'
07CF 05           5345        DB     5
07D0 46           5350        DB     'F'
07D1 06           5355        DB     6
07D2 4E           5360        DB     'N'
07D3 08           5365        DB     8
07D4 30           5370 FPTAB  DB     '0'
07D5 01           5375        DB     1
07D6 31           5380        DB     '1'
07D7 02           5385        DB     2
07D8 32           5390        DB     '2'
07D9 03           5395        DB     3
07DA 33           5400        DB     '3'
07DB 04           5405        DB     4
07DC 46           5410        DB     'F'
07DD 07           5415        DB     7
07DE              5416 *
07DE 43 4F 4D     5420 HDRTX  DT     'COMPUTALKER CONTROL MONITOR'
07E1 50 55 54
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

07E4 41 4C 4B
07E7 45 52 20
07EA 43 4F 4E
07ED 54 52 4F
07F0 4C 20 4D
07F3 4F 4E 49
07F6 54 4F 52
07F9 46 52 41    5425 FRMTX  DT     'FRAME    AV   F0'
07FC 4D 45 20
07FF 20 20 20
0802 41 56 20
0805 20 20 46
0808 30
0809 42 55 46    5430 CMDTX  DT     'BUFF LNG =       COMMAND:
080C 46 20 4C
080F 4E 47 20
0812 3D 20 20
0815 20 20 2C
0818 20 43 4F
081B 4D 4D 41
081E 4E 44 3A
0821 46 31 20    5435 FORTX  DT     'F1   F2   F3
0824 20 20 46
0827 32 20 20
082A 20 46 33
082D 20 20 20
0830 20 20
0832 41 48 20    5440 AMPTX  DT     'AH   AF   FF   AN'
0835 20 20 41
0838 46 20 20
083B 20 46 46
083E 20 20 20
0841 41 4E
0843 49 4C 4C    5445 ILLEG  DT      ILLEGAL
0846 45 47 41
0849 4C 20 21
084C             5446 *
084C             5450 LENGTH DS     2          * 16 BIT FRAME COUNT PUT HERE
084E             5455 BUFFER EQU    $          * BUFFER EXTENDS THRU REST OF RAM
084E             5456 *
084E             5460 * COMMON AREA:  STACK, VARIABLES, CONSTANTS, ETC.
084E             5461 *
084E             5465        ORG    40H     * RESERVE SPACE FOR INTERRUPT VECTORS
0840             5470 PLADAT DS     3       * CT OUTPUT SUBROUTINE
0843             5475 LAST   DS     2       * PREVIOUS 16 BIT COMMAND ARGUMENT
0845             5480 NUMBER DS     2       * 16 BIT COMMAND ARGUMENT ACCUMULATOR
0847             5485 BRANCH DS     2       * CMD STATE FLAG, USED AS TEMP IN CT2
0849             5490 FRTIME DS     2       * INTERFACE INTERVAL IN MSEC
084B             5495 RPTIME DS     2       * 16 BIT INTERSET TIME IN MSEC
084D             5500 CPM    DS     2       * CURRENT PARAMETER
084F             5505 CL     DS     2       * 16 BIT CURRENT FRAME NUMBER
0851             5510 CMDADR DS     2       * VIDEO ADDR OF CURRENT COMMAND CHAR
0853             5515 RPTFLG DS     1       * SET=1 TO REPEAT
0854             5520 PLABGN DS     2       * STARTING FRAME USED BY PLAY COMMAND
0856             5525 PLACNT DS     2       * FRAME COUNT USED BY PLAY COMMAND
0858             5530 FRMADR DS     2
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

085A             5535 FRMCTR DS     2
085C             5540 TOS    EQU    $           * TOP OF STACK
085C             5541 *
085C             5542 * STACK ALSO USED AS A TEMPORARY STORAGE LOCATION
085C             5543 * IN CT2 ROUTINES
085C             5544 *
085C             5545 STACK  EQU    CTMON-1     * STACK
```

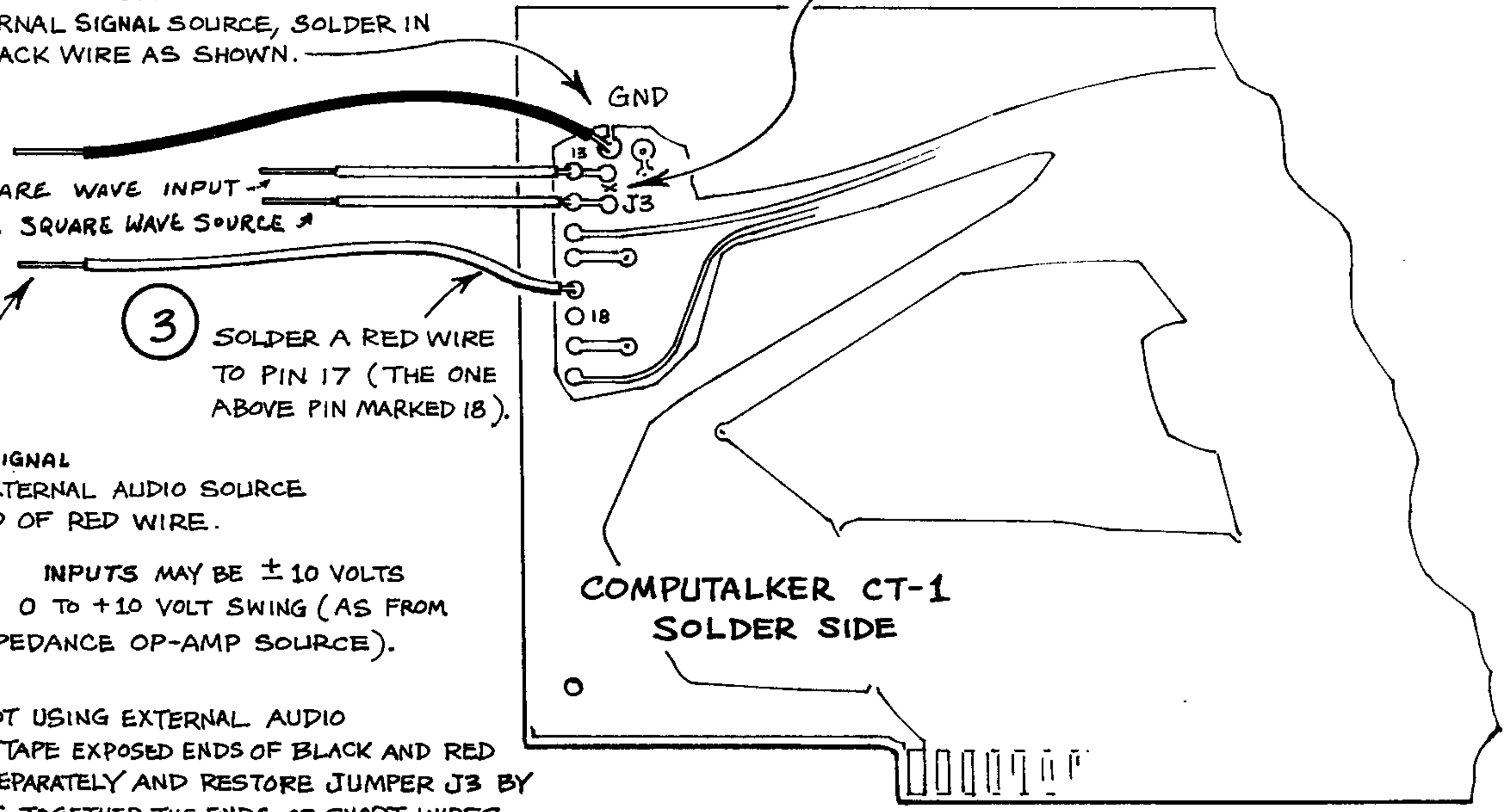
TO ENTER AN AUDIO SQUARE WAVE FROM EXTERNAL SOURCE, REPLACING AV AND FØ PARAMETERS, FOLLOW PROCEDURE BELOW

1. REMOVE JUMPER J3; THEN SOLDER TWO SHORT WIRES AT SAME PINS (FOR RESTORING JUMPER CONNECTION LATER).

2. FOR COMMON GROUND CONNECTION TO EXTERNAL SIGNAL SOURCE, SOLDER IN A BLACK WIRE AS SHOWN.

3. SOLDER A RED WIRE TO PIN 17 (THE ONE ABOVE PIN MARKED 18).

4. APPLY SIGNAL FROM EXTERNAL AUDIO SOURCE AT LEAD OF RED WIRE.



NOTE: INPUTS MAY BE $\pm$10 VOLTS OR HAVE 0 TO +10 VOLT SWING (AS FROM LOW IMPEDANCE OP-AMP SOURCE).

WHEN NOT USING EXTERNAL AUDIO SOURCE, TAPE EXPOSED ENDS OF BLACK AND RED WIRES SEPARATELY AND RESTORE JUMPER J3 BY CLIPPING TOGETHER THE ENDS OF SHORT WIRES.

Model CT-1 Warranty

The COMPUTALKER Model CT-1 Speech Synthesizer is warranted by Computalker Consultants against defects in workmanship and materials for a period of six (6) months from the date of delivery.

During the warranty period, Computalker Consultants will repair, or at its option, replace at no charge components that prove to be defective provided that the board is returned, shipping prepaid, to:

(if by U.S.Postal Serv.)	(if by private delivery serv.)
COMPUTALKER CONSULTANTS P.O. Box 1951 Santa Monica, CA 90406	COMPUTALKER CONSULTANTS 1730 A 21st Street Santa Monica, CA 90404

This warranty does not apply if the board has been damaged by accident or misuse, or as a result of repairs or modifications made by other than authorized personnel at the above captioned service facility.

No other warranty is expressed or implied. Computalker Consultants is not liable for consequential damages.

Because of the critical tuning required for a number of the analog circuits used in the Model CT-1, Computalker Consultants strongly recommends that the board be returned to us for any repairs needed. Beyond the period of the warranty, such repairs will be made with a charge for parts and labor. Computalker Consultants will cover the shipping costs to return the board.