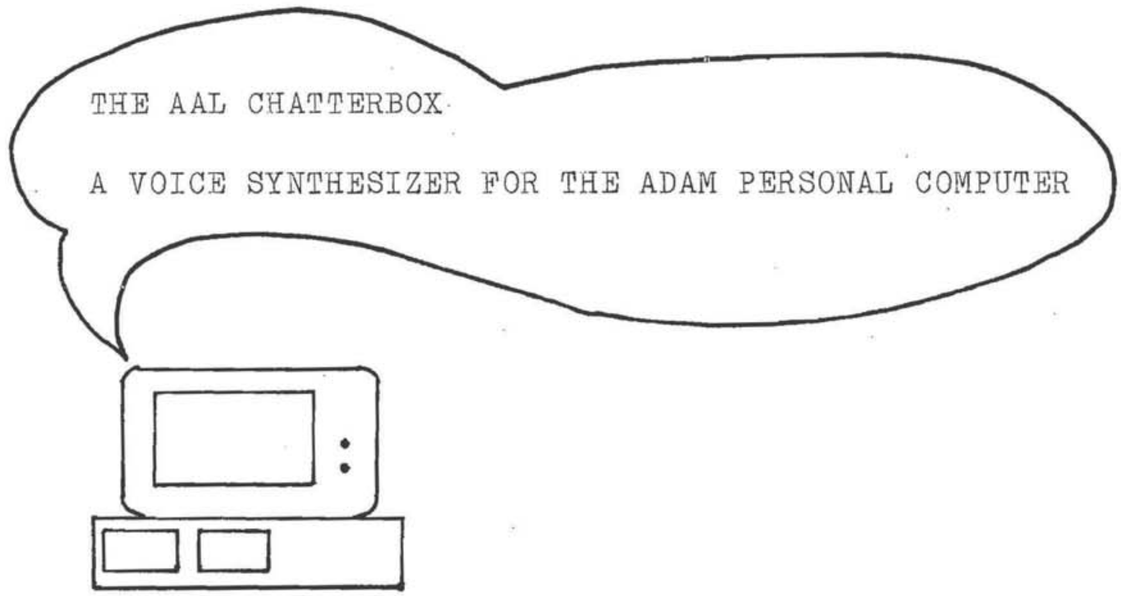

THE AAL CHATTERBOX
A VOICE SYNTHESIZER FOR THE ADAM PERSONAL COMPUTER

CONTENTS

FIGURES

INTRODUCTION

Thank you for purchasing the CHATTERBOX kit from AAL computing. We hope that your kit arrived quickly and in good condition, and that you have fun building and using this new accessory for your ADAM computer.

Please take a moment to read through this manual before starting, and check through the enclosed inventory list. Your kit was checked before dispatch but should you discover a part missing, please contact us and we will endeavour to correct the problem.

As you work through the project you should not encounter any problems as the design was built and tested by AAL. If you encounter any problems the troubleshooting guide should quickly resolve them. It's a good idea to check each stage of the project as you finish it. This way it will be easier to correct any mistakes before they become embedded in the next stage.

Good luck and thank you

UNPACKING AND INSPECTION

Please inspect the parts in your kit and check against the following list. Avoid touching the legs of the integrated circuits as static discharge could easily damage the devices.

part #1	7402	U1
part #2	7404	U2
part #3	7430	U3
part #4	7402	U4
part #5	SP0256-AL2	U5
part #6	74LS244	U6
part #7	LM380N	U7
part #8	RESISTOR 100K	R1
part #9	RESISTOR 100K	R2
part #10	CAPACITOR 0.1uF	C1
part #11	CAPACITOR 22pF	C2
part #12	CAPACITOR 22pF	C3
part #13	CAPACITOR 100uF	C4
part #14	CAPACITOR 47uF	C5
part #15	CAPACITOR 47uF	C6
part #16	DIODE 1N914	D1
part #17	CRYSTAL	XTAL
part #18	EDGE CONNECTOR	
part #19	MINI SPEAKER	
part #20	STRIPBOARD	

GETTING STARTED

Having verified that everything is present in your kit, you will want to get started on building your voice synthesizer module. We suggest you assemble the following tools which you will need to construct the unit.

1. Soldering iron- preferably with a 15W element and a fine tip for I.C. work
2. Side cutters for cutting component leads
3. snipe nosed pliers for bending component leads
4. Stripboard track removing tool
5. solder and hookup wire

The first step is to prepare the circuit board by breaking the tracks with the special tool. Refer to figure 1 for a diagram of the circuit board showing which tracks have to be removed. When cutting the track be sure to remove all of the link as it is easy to leave a short. Use a rotary action on the cutting tool to remove the metal from around the hole. Carefully check the completed board as any incorrect shorts could damage your ADAM.

When you are sure the board is correct, mount the edge connector in the appropriate position with reference to the component layout diagram in figure 2. Be sure to solder every connector pin. Although they are not all used in the circuit, they are necessary for mechanical strength when inserting or removing the board from the computer. Now check that no solder has bridged the gap between adjacent rows of tracks- it's easy to miss when doing a whole row.

The integrated circuits can now be mounted in their respective position. Ensure that the I.C.'s are correctly oriented before soldering- it's much easier than trying to desolder afterwards. Try to avoid keeping the iron on the legs of the devices for more than a few seconds when soldering as it is easy to burn out the device. After all the I.C's are in position give the board a quick check to ensure that all devices are in the correct position.

The rest of the components are now mounted on the board, again with reference to figure 2 for the location.

Now comes the part which requires the most skill and care, if your voice synthesizer is to work correctly first time : wiring in the devices to the schematic. You may have noticed that your computer is prone to causing local interference to another tv in the immediate vicinity. When a circuit board is plugged into the machine, care must be taken to ensure that no Radio Frequency Interference emissions cause a problem. The main cause of RFI is through the address and data lines of the computer being extended and acting like an antenna. For this reason we recommend a point to point wiring system on the underside of the pcb, to keep the leads as short as possible. Don't make them so short that they can't be separated, in case you need to do some changes at a later stage. Refer to figure 3 for a complete schematic diagram. It's a good idea to mark each link as you solder it in, so you do not miss any. The most common mistakes to avoid making, are joining two adjacent tracks together with a blob of solder or missing a link or putting a link to the wrong point. Also ensure that all the power and ground connections to the IC's are correct, especially the SP0256 as incorrect connection of the power will surely render the devices scrap. It's worth taking this part of the job slowly as it is difficult to trace a problem due to incorrect wiring when all the links are in.

Having completed construction of the circuit board give everything a final check to make sure that nothing is obviously wrong.....

Now you are ready to plug the unit into the computer. The next section of the manual deals with the theory of operation of the chatterbox, plus a troubleshooting guide. If you are not interested in how the unit functions, the subsequent section shows how to run test software to check the operation of the unit. WHEN PLUGGING THE UNIT INTO THE COMPUTER ALWAYS HAVE THE COMPONENT SIDE FACING UPWARDS. ALSO ENSURE THAT POWER IS OFF. Your Adam is a delicate electronic instrument : be careful.

We hope you have many happy hours of chattering !

THEORY OF OPERATION / TROUBLESHOOTING

The heart of the chatterbox is the General Instrument SP0256 voice synthesizer chip- U5. This is an N-Channel MOS Large Scale Integration circuit which uses a stored program to synthesize speech or complex sounds. There are four basic building blocks which perform the voice synthesis function

* A software programmable digital filter which models the human vocal tract.

* 16K on board ROM memory which stores the control program and the data for each sound.

* A Central Processor which controls the data flow fróm the ROM to the digital filter, the assembly of the "word strings" to link speech elements together and the amplitude and pitch data used to drive the digital filter.

* A pulse width modulator that creates a digital output, which is converted to an analog signal by an external low pass filter.

The device is controlled directly by the microprocessor in the ADAM using a set of address, control and data lines. These signals plus the power supplies are taken directly from the ADAM via the expansion module interface connector. The data lines from the processor are used to feed binary data to the address lines of the SP. This data is decoded and used to select which sound to produce from the SP. Only the six least significant bits of the ADAM data bus are needed to handle all the different combinations of sounds the SP produces to form words. Since the average sound or 'allophone' lasts only a few hundred milliseconds, and there may be ten or more used to form a word, this is why a control program must be used to correctly time the data transfer between the ADAM and the SP. When each allophone has been spoken by the SP a signal is asserted which is detected by the ADAM and used to time the next data. The way in which data is transferred by the processor is by using a special instruction of the CPU. The ADAM contains a Z80 microprocessor which has as part of its set the IN and OUT instructions. When the CPU encounters these instructions the data used by the instruction is transferred to or from an 'I/O port'. There are a possible 256 I/O ports in the ADAM. Some of these are used by the video display or the printer or the tape drive. The voice synthesizer needs 2 I/O ports, one is used to output data while the other is used to check the status of the SP. When an I/O instruction is issued, a hardware control signal IORQ is asserted which can be used to synchronize the data with another device. In this way the SP can be supplied with an enabling signal to coincide with the data coming from the ADAM.

Now that you have a basic idea of the concepts involved in using the ADAM to control a peripheral device, we will deal in detail with the schematic.

The eight least significant address lines from the Z80 appear on A0 - A7 on the edge connector. These are already buffered in the ADAM, so they can be used to drive other TTL gates directly. The circuit comprising U1 + U4 is used to provide two unique chip select pulses for I/O ports 2 and 3. An I/O address uses the same address lines as a memory address except that the memory circuits are cut off from the data when an I/O operation is in process. Instead the IORQ signal sets to logic 0 for the duration of the I/O operation. When I/O port 2 is being addressed by the Z80, address line A1 is at logic 1. A0 and A2-A7 are at logic 0. A2-A7 are all inverted by five of the gates in U2 - a 7404 Hex inverter, plus one of the gates from U1 (U1/a) wired as an inverter. This means that all inputs to U3 - an 8 input Nand gate, are at logic 1. With reference to figure 4, when all the inputs to the Nand gate are logic 1 the output is logic 0. The output of U3 is taken to the input of a Nor gate U1/b. U1 pin 6 is driven by IORQ. When this is at logic 0 and U1 pin 5 is also logic 0, the output pin U1/4 will be a logic 1. This is inverted by gate U1/c to give a logic 0 on U1/10. This point is pulsed to logic 0 when port 2 is addressed. However it is also logic 0 when port 3 is addressed, as address line A0 is not used by the circuit at this stage, and the only difference between I/O ports 2 and 3 is that port 3 causes address A0 to be set to logic 1 as well as A1. When this occurs U1/12 is at logic 0 and U1/11 is at logic 1. This causes U1/13 to be logic 0 which is inverted by U4/c to give a logic 1 at the port 2 chip select, U4 pin 10. A logic 1 is considered by the SP to be inactive, so this shows that when port 3 is addressed the SP is not enabled. However, when port 2 is addressed, the level of address line A0 is logic 0. The rest of the conditions remain the same as in the previous example. This means that U1 pins 11 and 12 are both at logic 0, causing U1/13 to be logic 1, which is again inverted by U4/c to give a logic 0 on U4/10. This enables the chip select on the SP for the duration of the I/O request. During this time, the data supplied by the program is fed to the address inputs of the SP, via the data lines D0-D5 from the ADAM. This data identifies which allophone will be spoken during this I/O operation.

During the time the allophone is being spoken, the LRQ pin of the SP (U5/9) is at logic 1. This signal is used by the SP to indicate when a new address may be loaded to start the next allophone. LRQ is connected to U6/2 - the input to a buffer. When U6/1 is at logic 1 the input to the buffer is not passed to the data bus, because the output remains in the high impedance state. However when port 3 is addressed by the ADAM, U4/4 sets to logic 0. This enables the chip select on the buffer U6/1. The LRQ pin U5/9 is connected to the ADAM data bus via the buffer in U6. This is the way the ADAM checks the status of the SP chip. Under program control the status can be used to control the flow of data to the SP. This enables a smooth utterance of the allophones used to form words.

The analog output of the SP appears on pin 24 of the device. This is in a form suitable for amplification directly by an audio amplifier U7 is a self contained amplifier which drives the speaker via C4.

TROUBLESHOOTING GUIDE

If after attempting to run the test software described in the next section, you are unable to get the unit to work, this section will help you locate the problem. You will need to examine some logic signals and check voltage readings. You will need a multimeter and a logic probe. If you do not have one or both of these they can be purchased quite cheaply from Radio Shack.

Firstly, switch off the ADAM. ALWAYS SWITCH OFF POWER BEFORE INSERTING OR REMOVING THE UNIT. Now, please recheck your wiring. It's easy to overlook a wiring error. Once you are sure that the wiring is correct, reconnect the unit to the ADAM. Switch the power on and check that the integrated circuits all have the correct voltages on their power pins. U1, U2, U3 and U4 are all 14 pin Dual In-line Packages and have pin 7 connected to 0 volts. Pin 14 is connected to +5V. U5 has 0V to pin 1 and +5V to pins 7 and 23. U6 is a 20 pin DIP, pin 10 is 0V and pin 20 is +5V. U7 is the audio amplifier and has 0V to pin 7. +12V is connected to pin 14.

If the power to the chips looks good, a step by step approach should soon locate the source of the problem. You will need to hook up your logic probe to the circuit. The only connections you need to make are to connect the +5V and 0V leads of the logic probe to a point on your circuit where you can access power. Alternatively you can take an extra set of wires from edge connector pins 1 for 0V and 58 for +5V. The probe tip is then connected to the point in the circuit you wish to measure. Your logic probe should be capable of detecting fast pulses. Most types do, and have a switch which can be used to select a positive or negative pulse. It is important to know how this works because some of the pulses you will be looking for will be very fast. We will indicate whether they are + or - going so that you can adjust your probe. Using your logic probe, check the signals on U2, pins 1,3,5,9, and 11. Also U3/1 and U1 pins 2 and 3. There should be a continuous pulsing on all these pins which should illuminate both hi and low indicators on the probe evenly. These are the address lines of the ADAM and are active whether the Chatterbox is active or not. If any of the signals is a steady hi or low level or not present, check the link to the edge connector pin for continuity. If the link is OK, the device is probably faulty, or another wire is forcing the signal to a steady state. If all the signals are present, check U2 pins 2,4,6,8 and 10, and U1 pin 1. The same signal should appear on these pins. Although it is inverted it will not look any different on the logic probe. Now follow the trail to U3, and check pins 1,2,3,4,5,6, and 11. These should all have pulses on them regardless of the state of the system. U3/12 should be a steady hi as it is strapped to +5V. The output from this eight input Nand gate is a fast negative pulse which is active whenever an address containg 2 or 3 is active. Hence there should be a continuous pulse when the system is idle, as there are plenty of background routines in operation inside the ADAM using all kinds of memory range. Check that a fast negative pulse appears on U1/5. U1/6 is the ADAM's I/O request line and this should also have a fast negative-going pulse on it all the time. Now, in order to trace the signal further it will be necessary to load the test software listed in figure 5. This is designed to access I/O port 2 and is necessary to get any pulses past U1 pin 4. Type the program as it appears in figure 5 and save it under name "voicetest". Do this before you try and run the test in case something in the hardware hangs. When you type run, you are prompted with a question mark. Type a number between 5 and 63 to attempt to send a single allophone through the circuit. When you hit return look at the signal on U1/4. At the time the allophone is spoken, address 2 will be low at the same time IORQ is low. This causes a fast positive pulse on the output of this NOR gate (see truth table- figure 4). This pulse is inverted and appears as a fast going negative-pulse on U1/10. Here it is gated with address line A0 to prevent address 3 from causing address 2 chip select to become active. Repeat the operation each time you check the next stage in the signal path. The unique I/O port 2 chip select next appears on U1/13 as a fast positive pulse. It is inverted again and finally comes out at U4/10. Here it should be a fast negative pulse. From here it is taken directly to the ALD pin on the SP. If your circuit speaks one allophone and then hangs, it could be that the status signal is not being read correctly by the ADAM. Look at U5/9. During the time the allophone is talking LRQ is at steady logic 1. When the SP is ready for a new load, U5/9 sets to logic 0. Check that the pin is at logic 0 before you send a digit in the test program. Watch it go high, then low again. If everything still seems OK, you will need to check that port 3 is working. You will need to type in the program listed in figure 6. Call it "wordtest". When you type run, it should cause the chatterbox to say the word "January". When you type run, check that U2 pin 13 has a pulse. This should appear as equal hi and lo on the logic probe. Check for a similar signal on the other side of the inverter. Now check the signal reaches U4/2. U4/3 should show a fast negative pulse when the word is being spoken. U4/1 should have a fast positive pulse when the program is running as it accesses I/O port 3, by the same principle as the port 2 example previously explained. Finally port 3 chip select is output on U4/4 and is connected to the enable pin on the buffer chip U6. When this fast negative pulse enables the buffer, the data from U5/9 is passed to the ADAM data bus where it is read by the program. In this way flow control of the allophone data is achieved.

If at any stage, you discover a signal which doesn't make it past a device, check that the wiring is OK then try another device. If everything looks good working through the logic, check the audio circuit originating from U5/24. Also check for pulses on pins 27 and 28 where the crystal feeds the SP.

Good luck with your troubleshooting.

SOFTWARE

This section deals briefly with how to incorporate the Chatterbox into your programs. Whilst there are an unlimited number of ways to program the voice synthesizer, we present one way which enables you to test the completed unit with a routine that you know at least works. If you run the following programs and the unit does not function, you know you have a hardware problem. Once you have proven that the hardware works O.K. you can experiment with the software to your heart's content. The Chatterbox is accessed by using a very simple machine language routine. This is necessary because there is no way of accessing the I/O memory from normal basic words. Figure 5 shows the listing for a program :- "Voicetest" which prompts the user for a number. This number is written to the Chatterbox to cause it to speak a single allophone. This is a useful, quick routine to test operation of the unit. The machine language instructions consist of binary data represented by decimal numbers. The numbers are stored in an area of reserved memory and accessed by the basic word "CALL". This transfers the control of the program to the machine language routine, until a "return" instruction is encountered.

Line 10 sets the top of user program memory to 39999, leaving locations 40000 onwards free to store the machine code.

Lines 20 to 50 and 100 load the machine code into memory.

Line 20 sets up a loop to use an indexed variable, x. This is used to point to the memory address- initially 40000.

Line 30 loads a variable, a, with the number representing the first machine language instruction. It gets this from the data statement in line 100.

Line 40 writes the data to the memory location pointed to by x.

Line 50 loops the routine until all 5 instructions have been transferred to memory.

Line 55 prompts the user for a number which is stored in a variable, c. This number is the one which represents which allophone will be spoken. It is written to memory location 40001 where it overwrites a dummy value of 5 in the initial set up.

Line 60 calls the program subroutine represented by the 5 instructions in memory location 40000 - 40004. The machine code is listed below in it's disassembled form

```
 LD, A, n   load accumulator with following data, n.
OUT, n, A   write the data, n, from the accumulator to port n.
RET         return to basic.
```

Line 65 returns to line 55 for another input, ad infinitum, or until you enter a "control c" character and hit return.

Line 100 contains the data representing the machine code.

The above program enables allophones to be loaded manually one at a time. Once the testing is over, the routine listed in figure 6 can be run. This one loads sequential allophones to form a word, using the status of the SP to time the data. The data in this particular example causes the chatterbox to say the word "January".

Line 10 sets the user program memory limit

Lines 100, 110, and 120 set up the same machine language routine as the voicetest program. It uses data from line 3000 which is put in memory locations 40000 to 40004. This subroutine writes the allophone data contained in memory location 40001 to the SP.

Lines 130, 140, and 150 set up another machine language routine in memory locations 40010 to 40015. The data is taken from line 3000 again. This subroutine inputs the data from the SP and stores it in memory location 40100.

Line 200 sets up a loop using a variable, d, which is set to equal the number of allophone identifiers stored in line 3010.

Line 210 loads a variable, c, with the allophone data from line 3010.

Line 220 enters a subroutine starting at line 1000 to speak the allophone, wait for status then return for more data.

Line 230 loops the routine back for the next allophone.

Line 1000 stores the allophone data loaded at line 210 into memory location 40001.

Line 1010 calls the machine code subroutine to write the allophone data to the SP.

Lines 2000 - 2030 call the subroutine which loops on the status of the SP, until it is ready to accept another allophone.

Line 3000 contains the data for both machine language subroutines.

Line 3010 contains the data for the allophones in the word.

The word can be altered by changing the count in line 200, to the new number of allophones. The new allophone data is then substituted into line 3010. Sentences can be constructed using different subroutines.

LINGUISTICS

The allophone speech synthesis technique provides the capability for an unlimited vocabulary. Any word or phrase can be formed using the appropriate combination of allophones and pauses. A few basic linguistic concepts will help you add to the library of words given in figure 7.

Firstly, there is no one-to-one correspondence between written letters and speech sounds. Secondly, speech sounds are acoustically different depending on their position within a word. Thirdly, the human ear may perceive the same acoustic signal differently in the context of different sounds. The first point compares to the problem that a child encounters when learning to read. Each sound in a language may be represented by more than one letter and, conversely each letter may represent more than one sound, as shown below :-

	Same sound represented by different letters	Different sound represented by the same letters
Vowels	mEAt	vEIn
	fEEt	forEIgn
	pEte	dEIsm
Conso-	SHip	althouGH
nants	tenSIon	GHastly
	preCIous	couGH

Because of these spelling irregularities, it is necessary to think in terms of SOUNDS, not letters when using allophones. The second and equally important point to note, is that a speech sound may differ depending upon its position within a word. For example, the initial K sound in "coop" will be acoustically different from the K's in "keep" and "speak", due to the influence of the vowels which follow them. Also the final K in speak is usually not as loud as initial K's. Finally a listener may identify the same acoustic signal differently depending on the context in which it is perceived. Don't be surprised, therefore, if an allophone word sounds slightly different when used in various phrases.

PHONEMES OF ENGLISH

The sounds of a language are called phonemes. Each language has a set which is slightly different from that of other languages. Figure 8 shows all the consonant phonemes of English, and figure 9 all the vowel phonemes. Consonants are produced by creating an occlusion or constriction in the vocal tract which produces an aperiodic sound source. If the vocal chords are vibrating at the same time, as in the case of the voiced fricatives VV, DH, ZZ and ZH (see guidelines for using allophones- figure 10) there are two sound sources: one which is aperiodic and one which is periodic. Vowels are usually produced with a relatively open vocal tract and a periodic sound source provided by the vibrating vocal chords. They are classified according to whether the front or back of the tongue is high or low (see figure 9), whether they are long or short, and whether the lips are rounded or unrounded. In English all rounded vowels are produced in or near the back of the mouth (UW, UH, OW, AO, OR, AW). Speech sounds which have features in common behave in similar ways. For example, the voiceless stop consonants PP, TT, and KK (figure 8) should be preceeded by 50-80 milliseconds of silence. The voiced stop consonants BB, DD and GG should have 10-30 mS of silence preceeding.

HOW TO USE THE ALLOPHONE SET

See figure 7 for instructions on how to create all the sample words in this section. The allophone set (see figure 11) contains 2 or 3 versions of some phonemes. It may be necessary to use one allophone of a particular phoneme for word-or-syllable-final position. A detailed set of guidelines for using the allophones is given in figure 11. Note that these are suggestions, not rules. For example, DD2 sounds good in initial positions and DD1 sounds good in final positions, as in "daughter" and "collide" respectively. One of the differences between the initial and final versions of a consonant, is that an initial version may be longer than a final version. Therefore, to create an initial SS, you can use 2 SS's instead of the usual single SS at the end of a word or syllable, as in "sister". Note that this can be done with TH, and FF, and the inherently short vowels (to be discussed below) but with no other consonants. You will want to experiment with some consonants such as STR, CL to discover which version works best in the cluster. For example, KK1 sounds good before LL as in "clown", and KK2 sounds good before WW as in "square". One allophone of a particular phoneme may sound better before or after back vowels and another before or after front vowels. KK3 sounds good before UH and KK1 sounds good before IY, as in "cookie". Some sounds (PP, BB, TT, DD, KK, GG, CH and JH) require a brief duration of silence before them. For most of these, the silence has already been added but you may decide you want to add more. Therefore there are several pauses included in the allophone set varying from 10-200 msec. To create the final sounds in the words "letter" and "little" use the allophones ER and EL.

Remember that you must always think how a word sounds, not how it is spelled. For example, the NG sound is represented by the letter N in "uncle". Also remember that some sounds may not even be represented in words by any letters, as the YY in "computer". As mentioned earlier there are some vowels which can be doubled to make longer versions for stressed syllables. These are the inherently short vowels IH, EH, AE, AX, AA, and UH. For example, in the word "extent" use one EH in the first syllable, which is unstressed and two EH's in the second syllable which is stressed. Of the inherently long vowels there is one, UW, which has a long and a short version. The short one, UW1, sounds good after YY in "computer". The long version, UW2, sounds good in monosyllabic words like "two".

Included in the vowel set is a group called R-colored vowels. These are vowel + R combinations. For example, the AR in "alarm" and the OR in "score". Of the R-colored vowels, there is one, ER which has a long and short version. The short version is good for polysyllabic words with final ER sounds like "letter", and the long version is good for monosyllabic words like "fir". One final suggestion is that you may want to add a pause of 30-50 ms between words, when creating sentences, and a pause of 100-200 msec between clauses.

Note: Every utterance must be followed by a pause in order to make the chip stop talking the last allophone.

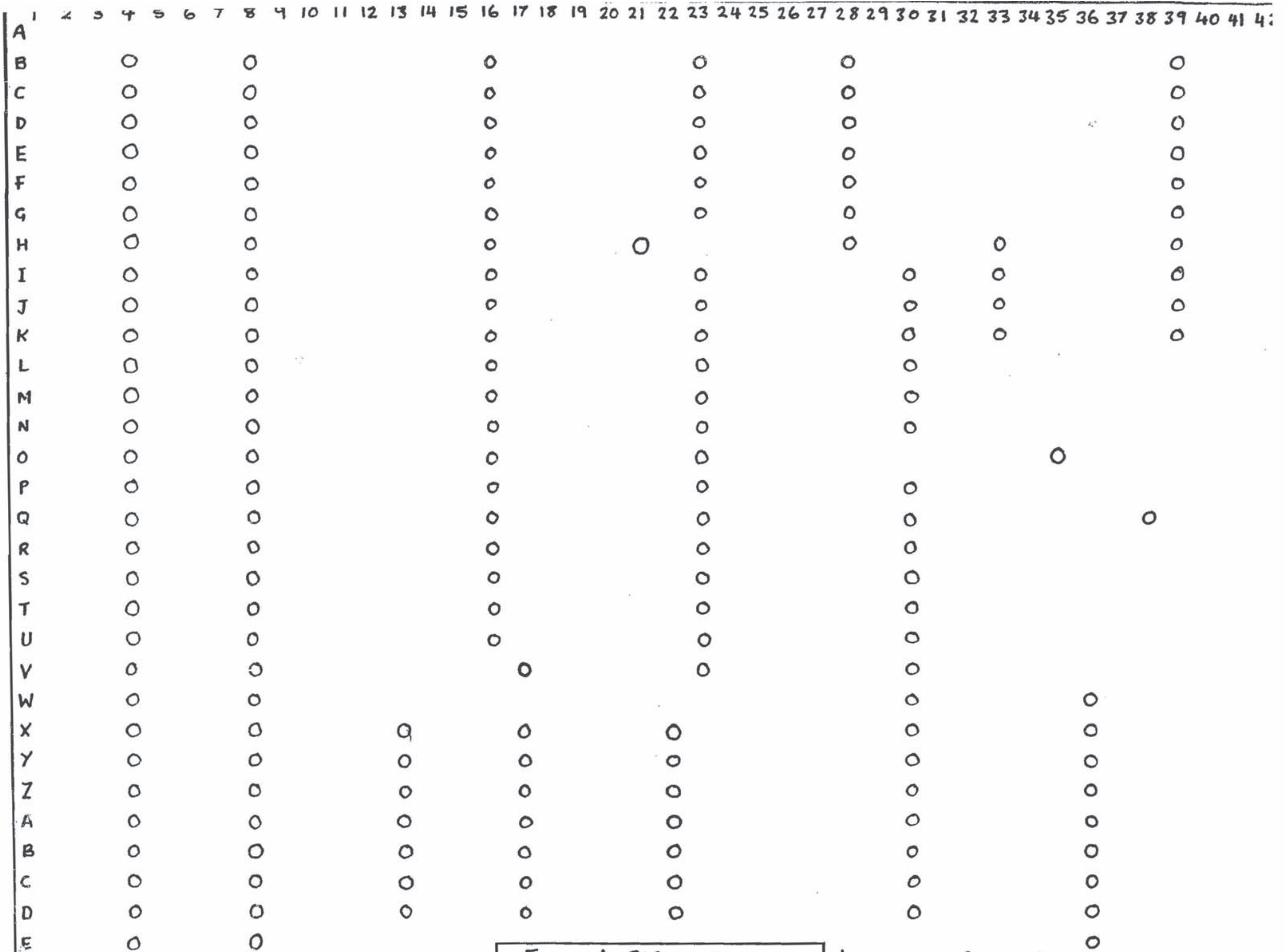


FIGURE 1 PCB TRACK CUTS.

LOOKING AT COPPER SIDE

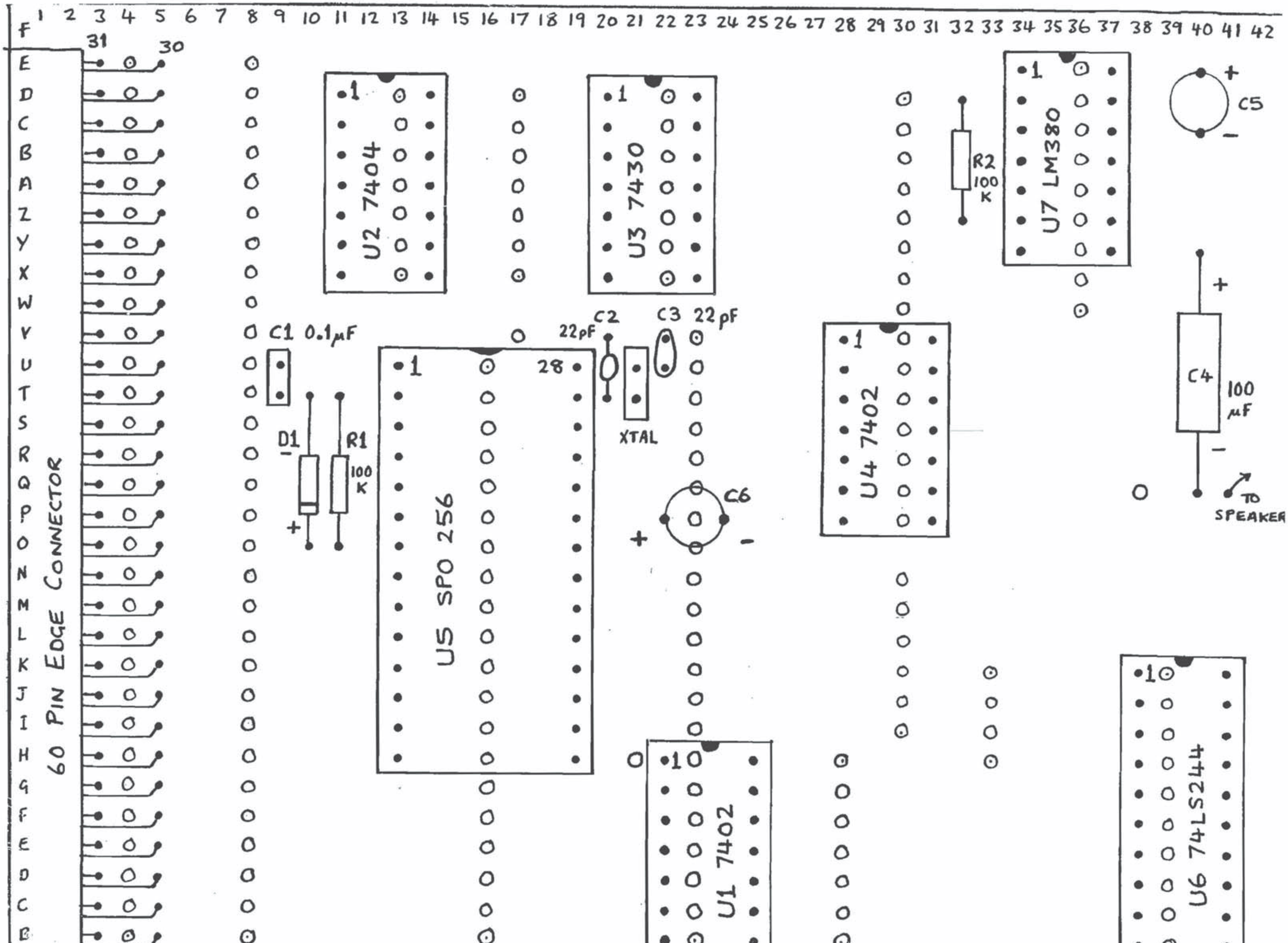
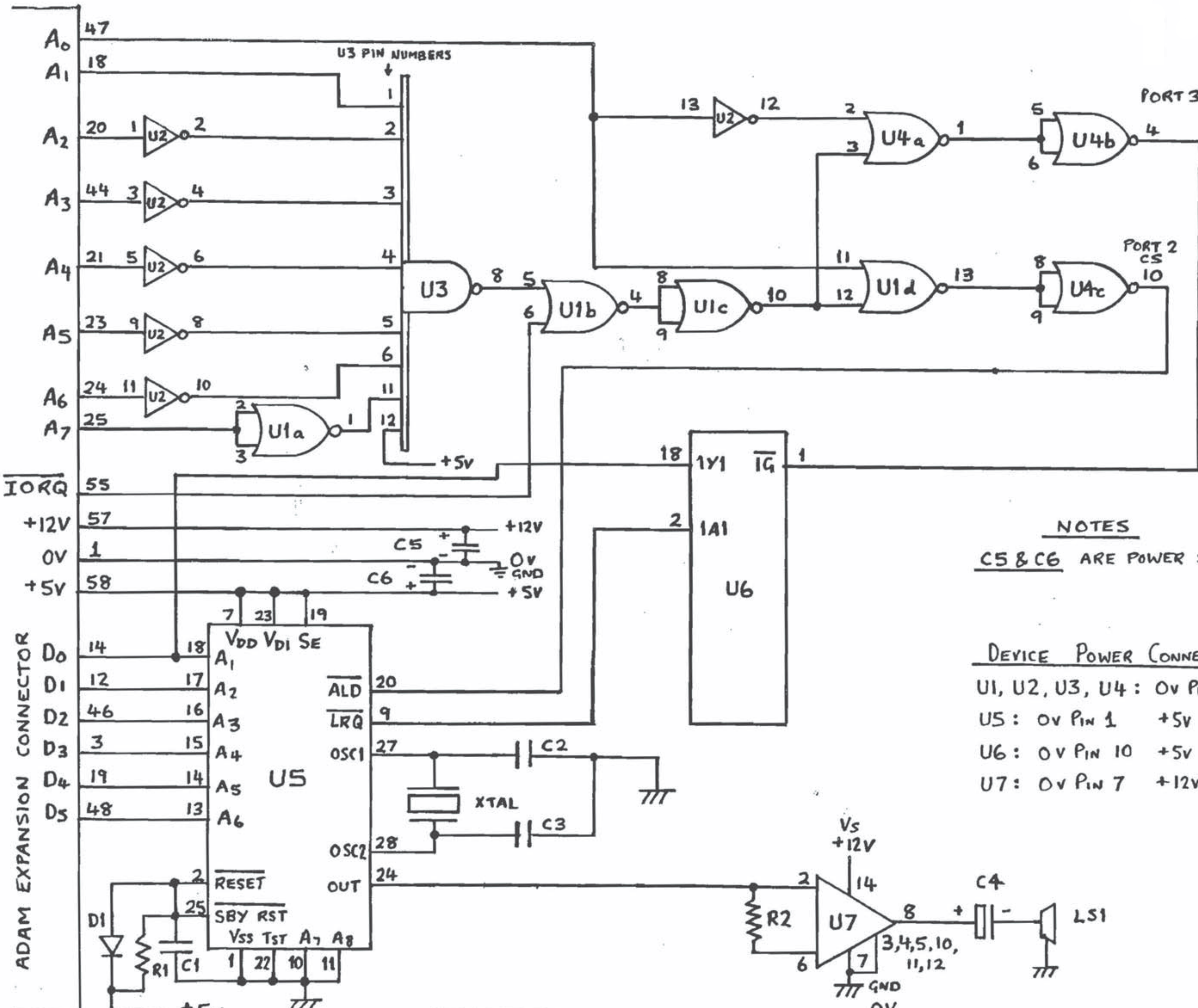


FIGURE 2 COMPONENT LAYOUT

LOOKING DOWN ON COMPONENT SIDE



NOTES

C5 & C6 ARE POWER SUPPLY DECOUPLING

DEVICE POWER CONNECTIONS

U1, U2, U3, U4:	0V PIN 7	+5V PIN 14
U5:	0V PIN 1	+5V PINS 7 & 23
U6:	0V PIN 10	+5V PIN 20
U7:	0V PIN 7	+12V PIN 14

FIGURE 3 SCHEMATIC DIAGRAM

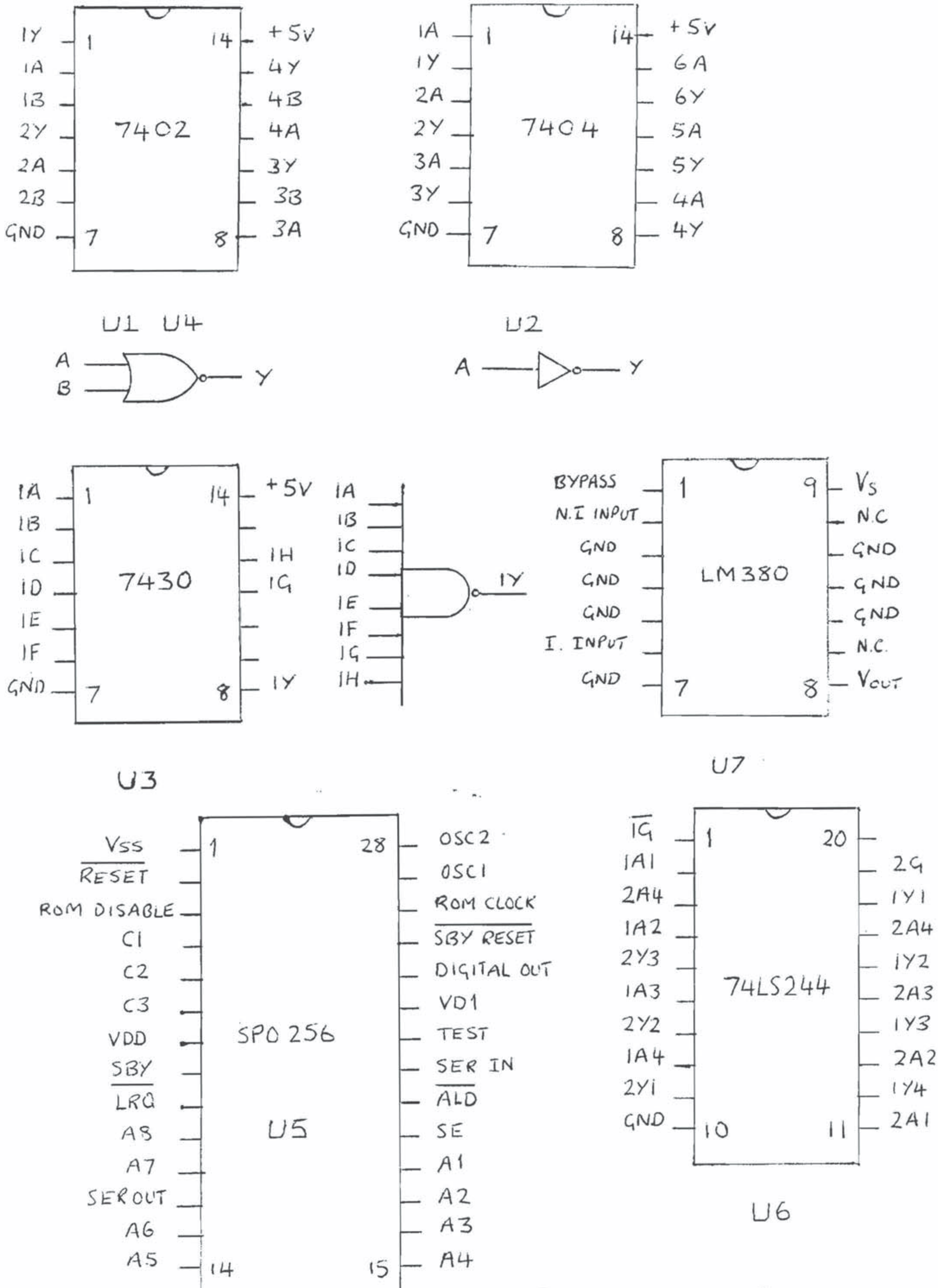


FIGURE 4 DEVICE PINOUTS

```
  10 HIMEM :39999
  20 FOR x = 40000 TO 40004
  30 READ a
  40 POKE x, a
  50 NEXT
  55 INPUT c: POKE 40001, c
  60 CALL 40000
  65 GOTO 55
 100 DATA        62,5,211,2,201
```

FIGURE 5. Voicetest Listing

```
  10 HIMEM :39999
 100 FOR x = 40000 TO 40004
 110 READ a: POKE x, a
 120 NEXT
 130 FOR x = 40010 TO 40015
 140 READ a: POKE x, a
 150 NEXT
 200 FOR d = 1 TO 8
 210 READ c
 220 GOSUB 1000
 230 NEXT
 240 END
1000 POKE 40001, c
1010 CALL 40000
2000 CALL 40010
2010 IF PEEK(40100) = 2 THEN 2030
2020 GOTO 2000
2030 RETURN
3000 DATA     62,0,211,2,201,219,3,50,164,156,201
3010 DATA     10,26,26,11,25,47,19,3
```

FIGURE 6. Wordtest Listing

FIGURE 7. DICTIONARY

NUMBERS	
zero	ZZ YR OW
one, won	WW SX AX NN1
two, to, too	TT2 UW2
three	TH RR1 IY
four, for,	FF FF OR
five	FF FF AY VV
six	SS SS IH IH PA3 KK2 SS
seven	SS SS EH EH VV IH NN1
eight, ate	EY PA3 TT2
nine	NN1 AA AY NN1
ten	TT2 EH EH NN1
eleven	IH LL EH EH VV IH NN1
twelve	TT2 WH EH EH LL VV
thirteen	TH ER1 PA2 PA3 TT2 IY NN1
fourteen	FF OR PA2 PA3 TT2 IY NN1
fifteen	FF IH FF PA2 PA3 TT2 IY NN1
sixteen	SS SS IH PA3 KK2 SS PA2 PA3 TT2 IY NN1
seventeen	SS SS EH VV TH NN1 PA2 PA3 TT2 IY NN1
eighteen	EY PA2 PA3 TT2 IY NN1
nineteen	NN1 AY NN1 PA2 PA3 TT2 IY NN1
twenty	TT2 WH EH EH NN1 PA2 PA3 TT2 IY
thirty	TH ER2 PA2 PA3 TT2 IY
forty	FF OR PA3 TT2 IY
fifty	FF FF IH FF FF PA2 PA3 TT2 IY
sixty	SS SS IH PA3 KK2 SS PA2 PA3 TT2 IY
seventy	SS SS EH VV IH NN1 PA2 PA3 TT2 IY
eighty	EY PA3 TT2 IY
ninety	NN1 AY NN1 PA3 TT2 IY
hundred	HH2 AX AX NN1 PA2 DD2 RR2 IH IH PA1 DD1
thousand	TH AA AW ZZ TH PA1 PA1 NN1 DD1

million	MM IH IH LL YY1 AX NN1
DAY OF THE WEEK	
Sunday	SS SS AX AX NN1 PA2 DD2 EY
Monday	MM AX AX NN1
Tuesday	TT2 UW2 ZZ PA2 DD2 EY
Wednesday	WW EH EH NN1 ZZ PA2 DD2 EY
	PA2 DD2 EY
Thursday	TH ER ZZ PA2 DD2 EY
Friday	FF RR2 AY PA2 DD2 EY
Saturday	SS SS AE PA3 TT2 PA2 DD2 EY
MONTHS	
January	JH AE AE NN1 YY2 XR 1Y
February	FF EH EH PA1 BR RR2 UW2 XR 1Y
March	MM AR PA3 CH
April	EY PA3 PP RR2 IH IH LL
May	MM EY
June	JH UW2 NN1
July	JH UW1 LL AY
August	AO AO PA2 GG2 AX SS PA3 TT1
September	SS SS EH PA3 PP PA3 TT2 EH EH PA1 BB2 ER1
October	AA PA2 KK2 PA3 TT2 OW PA1 BB2 ER1
November	NN2 OW VV EH EH MM PA1 BB2 ER1
December	DD2 IY SS SS EH EH MM PA1 BB2 ER1
LETTERS	
A	EY
B	BB2 IY
C	SS SS IY
D	DD2 IY
E	IY
F	EH EH FF FF
G	GH IY
H	EY PA2 PA3 CH
I	AA AY

FIGURE 7 cont'd

LETTERS

J	JH EH EY
K	KK1 EH EY
L	EH EH EL
M	EH EH MM
N	EH EH NN1
O	OW
P	PP IY
Q	KK1 YY1 UW2
R	AR
S	EH EH SS SS
T	TT2 IY
U	YY1 UW2
V	VV IY
W	DD2 AX PA2 BB2 EL YY1 UW2
X	EH EH PA3 KK2 SS SS
Y	WW AY
Z	ZZ IY

DICTIONARY

alarm	AX LL AR MM
bathe	BB2 EY DH2
bather	BB2 EY DH2 ER1
bathing	BB2 EY DH2 IH NG
beer	BB2 YR
bread	BB1 RR2 EH EH PA1 DD1
by	BB2 AA AY
calendar	KK1 AE AE LL EH NN1 PA2 DD2 ER1
clock	KK1 LL AA AA PA3 KK2
clown	KK1 LL AW NN1
check	CH EH EH PA3 KK2
checked	CH EH EH PA3 KK2 KK2 PA2 TT2
checker	CH CH EH PA3 KK1 ER1
checkers	CH EH EH PA3 KK1 ER1 ZZ
checking	CH EH EH PA3 KK1 IH NG
checks	CH EH EH PA3 KK1 SS
cognitive	KK3 AA AA GG3 NN1 IH PA3 TT2 IH VV
collide	KK3 AX LL AY DD1
computer	KK1 AX MM PP1 YY1 UW4 TT2 ER
cookie	KK3 UH KK1 IY
coop	KK3 UW2 PA3 PP
correct	KK1 ER2 EH EH PA2 KK2 PA2 TT1
corrected	KK1 ER2 EH EH PA2 KK2 PA2 TT2 IH PA2 DD1
correcting	KK1 ER2 EH EH PA2 KK2 PA2 TT2 IH NG
corrects	KK1 ER2 EH EH PA2 KK2 PA2 TT1 SS
crown	KK1 RR2 AW NN1
date	DD2 EY PA3 TT2
daughter	DD2 AO TT2 ER1
day	DD EH EY
divided	DD2 IH VV AY PA2 DD2 IH PA2 DD1
emotional	IY MM OW SH AX NN1 AX EL
engage	EH EH PA1 NN1 GG1 EY PA2 JH
engagement	EH EH PA1 NN1 GG1 EY PA2 JH MM EH EH NN1 PA2
engaging	EH EH PA1 NN1 GG1 EY PA2 JH IH NG
enrage	EH NN1 RR1 EY PA2 JH
enraging	EH NN1 RR1 EY PA2 JH IH NG
escape	EH SS SS PA3 KK1 PA2 PA3 PP
escaped	EH SS SS PA3 KK1 PA2 PA3 PP PA2 TT2
escapes	EH SS SS PA3 KK1 PA2 PA3 PP SS
escaping	EH SS SS PA3 KK1 PA2 PA3 PP IH NG
equal	IY PA2 PA3 KK3 WH AX EL
equals	IY PA2 PA3 KK3 WH AX EL ZZ
error	EH XR OR
extent	EH KK1 SS TT2 EH EH NN1 TT2
fir	FF ER2
freeze	FF FF RR1 IY ZZ
freezer	FF FF RR1 IY ZZ ER1
freezers	FF FF RR1 IY ZZ ER1 ZZ

FIGURE 7 cont'd

DICTIONARY

frozen	FF FF RR1 OW ZZ EH NN1
gauge	GG1 EY PA2 JH
gauged	GG1 EY PA2 JH PA2 DD1
gauges	GG1 EY PA2 JH IH ZZ
gauging	GG1 EY PA2 JH IH NG
hello	HH EH LL AX OW
hour	AW ER1
infinitive	IH NN1 FF FF IH IH NN1 IH PA2 PA3 TT2 IH VV
intrigue	IH NN1 PA3 TT2 RR2 IY PA1 GG3
intrigues	IH NN1 PA3 TT2 RR2 IY PA1 GG3 ZZ
intriguing	IH NN1 PA3 TT2 RR2 IY PA1 GG3 IH NG
investigate	IH IH NN1 VV EH EH SS PA2 PA3 TT2 IH PA1 GG1 EY PA2 TT2
key	KK1 IY
legislates	LL EH EH PA2 JH JH SS SS LL EY PA2 PA3 TT1 SS
legislature	LL EH EH PA2 JH JH SS SS LL EY PA2 PA3 CH ER1
letter	LL EH EH PA3 TT2 ER1
litter	LL IH IH PA3 TT2 ER1
little	LL IH IH PA3 TT2 EL
memories	MM EH EH MM ER2 IY ZZ
minute	MM IH NN1 IH PA3 TT2
month	MM AX NN1 TH
nipped	NN1 IH IH PA2 PA3 PP PA3 TT2
no	NN2 AX OW
physical	FF FF IH ZZ IH PA3 KK1 AX EL
pin	PP IH IH NN1
pinning	PP IH IH NN1 IH NG1
pledge	PP LL EH EH PA3 JH
pledged	PP LL EH EH PA3 JH PA2 DD1
plus	PP LL AX AX SS SS
rays	RR1 EH EY ZZ
ready	RR1 EH EH PA1 DD2 IY
red	RR1 EH EH PA1 DD1
robots	RR1 OW PA2 BB2 AA PA3 TT1 SS
score	SS SS PA3 KK3 OR
second	SS SS EH PA2 KK1 IH NN1 PA2 DD1
sensitivity	SS SS EH EH NN1 SS SS IH PA2 PA3 TT2 IH VV IH PA2 PA3 TT2 IY
sincere	SS SS IH IH NN1 SS SS YR
sincerity	SS SS IH IH NN1 SS SS EH EH RR1 IH PA2 PA3 TT2 IY
sister	SS SS IH IH SS PA3 TT2 ER1
speak	SS SS PA3 IY PA3 KK2
spell	SS SS PA3 PP EH EH EL
spelling	SS SS PA3 PP EH EH EL IH NG
start	SS SS PA3 TT2 AR PA3 TT2
starter	SS SS PA3 TT2 AR PA3 TT2 ER1
stop	SS SS PA3 TT1 AA AA PA3 PP
subject (noun)	SS SS AX AX PA2 BB1 PA2 JH EH PA3 KK2 PA3 TT2
subject (verb)	SS SS AX PA2 BB1 PA2 JH EH EH PA3 KK2 PA3 TT2
sweater	SS SS WW EH EH PA3 TT2 ER1
switch	SS SS WH IH IH PA3 CH
system	SS SS IH IH SS SS PA3 TT2 EH MM
talking	TT2 AO AO PA3 KK1 IH NG

FIGURE 7 cont'd

DICTIONARY

threaded	TH RR1 EH EH PA2 DD2 IH PA2 DD1	uncle	AX NG PA3 KK3 EL
		whale	WW EY EL
		whalers	WW EY LL ER1 ZZ
then	DH1 EH EH NN1	year	YY2 YR
time	TT2 AA AY MM	yes	YY2 EH EH SS SS

		LABIAL	LABIO-DENTAL	INTER-DENTAL	ALVEO-LAR	PALATAL	VELAR	GLOTTAL
Stops:	Voiceless	PP			TT		KK	
	Voiced	BB			DD		GG	
Fricatives	Voiceless	WH	FF	TH	SS	SH		HH
	Voiced		VV	DH	ZZ	ZH *		
Affricates	Voiceless					CH		
	Voiced					JH		
Nasals	Voiced	MM			NN		NG *	
Resonants	Voiced	WW			RR,LL	YY		

* These do not occur in word-initial position in English

Labial : Upper and lower lips touch or approximate
Labio-dental : Upper teeth and lower lip touch
Inter-dental : Tongue between teeth
Alveolar : Tip of tongue touches or approximates alveolar ridge - (just behind upper teeth)
Palatal : Body of tongue approximates palate (roof of mouth)
Velar : Body of tongue touches velum (posterior of roof of mouth)
Glottal : Glottis (opening between vocal chords)

FIGURE 8 Consonant phonemes of English

	FRONT	CENTRAL	BACK
High	YR IY IH *		UW # UH * #
Mid	EY EH * XR	ER AX *	OW # OY #
Low	AE *	AW # AY AR AA *	AO * # OR #

* Short Vowels
# Long Vowels

FIGURE 9 Vowel phonemes of English

SILENCE

PA1 (10mS)- before BB, DD, GG and JH
PA2 (30ms)- before BB, DD, GG and JH
PA3 (50ms)- before PP, TT, KK and CH, and between words
PA4 (100ms)- between clauses and sentences
PA5 (200ms)- between clauses and sentences

SHORT VOWELS

*/IH/ - sitting, stranded
*/EH/ - extent, gentlemen
*/AE/ - extract, acting
*/UH/ - cookie, full
*/AO/ - talking, song
*/AX/ - lapel, instruction
*/AA/ - pottery, cotton

LONG VOWELS

/IY/ - treat, people, penny
/EY/ - great, statement, tray
/AY/ - kite, sky, mighty
/OY/ - noise, toy, voice
/UW1/ - after clusters with YY: computer
/UW2/ - in monosyllabic words: two, food
/OW/ - zone, close, snow
/AW/ - sound, mouse, down
/EL/ - little, angle, gentle

R-COLORED VOWELS

/ER1/ - letter, furniture, interrupt
/ER2/ - monosyllables: bird fern, burn
/OR/ - fortune, adorn, store
/AR/ - farm, alarm, garment
/YR/ - hear, earring, irresponsible
/XR/ - hair, declare, stare

RESONANTS

/WW/ - we, warrant, linguist
/RR1/ - initial position: read, write, x-ray
/RR2/ - initial clusters: brown, crane, grease
/LL/ - like, hello, steel
/YY1/ - clusters: cute, beauty, computer
/YY2/ - initial position: yes, yarn, yo-yo

VOICED FRICATIVES

/VV/ - vest, prove, even
/DH1/ - word-initial position: this, then, they
/DH2/ - word-final and between vowels: bathe, bathing
/ZZ/ - zoo, phase
/ZH/ - beige, pleasure

VOICELESS FRICATIVES

*/FF/ -> These may be doubled for initial-position and used singly in
*/TH/ -> final position
*/SS/ ->
/SH/ - shirt, leash, nation
/HH1/ - before front vowels: YR IY, IH, EY, EH, XR, AE
/HH2/ - before back vowels: UW UH, OW, OY, AO, OR, AR
/WH/ - white, whim, twenty

VOICED STOPS

/BB1/ - final position: rib between vowels: fibber clusters: bleed, brown
/BB2/ - initial position before a vowel: beast
/DD1/ - final position: played end
/DD2/ - initial position: down clusters: drain
/GG1/ - before high front vowels YR, IY, IH, EY, EH, XR
/GG2/ - before high back vowels UW, UH, OW, OY, AX, and clusters: green, glue

FIGURE 10 Guide To Using Allophones

Figure 10 cont'd

Voiced Stops
/GG3/ - before low vowels: AE, AW, AY, AR, AA, AO, OR, ER ; and medial clusters: anger, and final position: peg

VOICELESS STOPS

/PP/	- pleasure, ample, trip
/TT1/	- final clusters before SS: tests, its
/TT2/	- all other positions test, street
/KK1/	- before front vowels: YR, IY, IH, EY, EH, XR AY, AE, ER, AX, initial clusters: cute clown scream
/KK2/	- final position: speak final clusters: task
/KK3/	- before back vowels: UW UH, OW, OY, OR, AR, AO initial clusters: crane quick, clown, scream

AFFRICATES

/CH/	- church, feature
/JH/	- judge, injure

NASAL

/MM/	- milk, alarm, ample
/NN1/	- before front and central vowels: YR, IY, IH, EY EH, XR, AE, ER, AX, AW AY, UW final clusters: earn
/NN2/	- before back vowels: UH OW, OY, OR, AR, AA
/NG/	- string, anger

* These allophones may be doubled

DECIMAL ADDRESS	ALLOPHONE	SAMPLE WORD	DURATION	DECIMAL ADDRESS	ALLOPHONE	SAMPLE WORD	DURATION
0	PA1	PAUSE	10 mS	32	/AW/	Out	370 mS
1	PA2	PAUSE	30 mS	33	/DD2/	Do	160 mS
2	PA3	PAUSE	50 mS	34	/GG3/	Wig	140 mS
3	PA4	PAUSE	100 ms	35	/VV/	Vest	190 mS
4	PA5	PAUSE	200 mS	36	/GG1/	Got	80 mS
5	/OY/	Boy	420 mS	37	/SH/	Ship	160 mS
6	/AY/	Sky	260 mS	38	/ZH/	Azure	190 mS
7	/EH/	End	70 mS	39	/RR2/	Brain	120 mS
8	/KK3/	Comb	120 mS	40	/FF/	Food	150 mS
9	/PP/	Pow	210 mS	41	/KK2/	Sky	190 mS
10	/JH/	Dodge	140 mS	42	/KK1/	Can't	160 mS
11	/NN1/	Thin	140 mS	43	/ZZ/	Zoo	210 mS
12	/IH/	Sit	70 mS	44	/NG/	Anchor	220 mS
13	/TT2/	To	140 mS	45	/LL/	Lake	110 ms
14	/RR1/	Rural	170 mS	46	/WW/	Wool	180 mS
15	/AX/	Succeed	70 mS	47	/XR/	Repair	360 mS
16	/MM/	Milk	180 mS	48	/WH/	Whig	200 mS
17	/TT1/	Part	100 mS	49	/YY1/	Yes	130 mS
18	/DH1/	They	290 mS	50	/CH/	Church	190 mS
19	/IY/	See	250 mS	51	/ER1/	Fir	160 mS
20	/EY/	Beige	280 mS	52	/ER2/	Fir	300 mS
21	/DD1/	Could	70 mS	53	/OW/	Beau	240 mS
22	/UW1/	To	100 mS	54	/DH2/	They	240 mS
23	/AO/	Aught	100 mS	55	/SS/	Vest	90 mS
24	/AA/	Hot	100 mS	56	/NN2/	No	190 mS
25	/YY2/	Yes	180 mS	57	/HH2/	Hoe	180 mS
26	/AE/	Hat	120 mS	58	/OR/	Store	330 mS
27	/HH1/	He	130 mS	59	/AR/	Alarm	290 mS
28	/BB1/	Business	80 mS	60	/YR/	Clear	350 mS
29	/TH/	Thin	180 mS	61	/GG2/	Guest	40 mS
30	/UH/	Book	100 mS	62	/EL/	Saddle	190 mS
31	/UW2/	Food	260 mS	63	/BB2/	Business	50 mS

FIGURE 11. Allophone Address Table

```
 100 FOR x = 40000 TO 40004
 110 READ a: POKE x, a
 120 NEXT
 130 FOR x = 40010 TO 40015
 140 READ a: POKE x, a
 150 NEXT
 200 FOR d = 1 TO 39
 210 READ c
 220 GOSUB 1000
 230 NEXT
 240 END
1000 POKE 40001, c
1010 CALL 40000
2000 CALL 40010
2010 IF PEEK(40100) = 2 THEN 2030
2020 GOTO 2000
2030 RETURN
3000 DATA      62,0,211,2,201,219,3,50,164,156,201
3050 DATA         27,3,6,4
3060 DATA      18,12,55,4
3070 DATA      12,43,4
3080 DATA       26,2,21,15,16,4,4,4,4,4,4
3110 DATA      27,1,26,35,3
3120 DATA      20,3
3130 DATA      56,6,2,55,55,3
3140 DATA       33,7,20,4
```

FIGURE 12. Program listing for an automatic greeting. Save it on your Smartbasic tape, under the name HELLO. Now when you load Smartbasic, Adam will greet you.

```
100 FOR x = 40000 TO 40004
110 READ a: POKE x, a
120 NEXT
130 FOR x = 40010 TO 40015
140 READ a: POKE x, a
150 NEXT
160 FOR x = 39000 TO 39049
162 READ g: POKE x, g
164 NEXT
200 INPUT "type a number "; e
210 ON e+1 GOTO 300, 310, 320, 330, 340, 350, 360, 370, 380, 390
300 FOR f = 0 TO 3
302 c = PEEK(39000+f)
304 GOSUB 1000
306 NEXT
308 GOTO 200
310 FOR f = 0 TO 4
312 c = PEEK(39004+f)
]
```

Program listing for a text-to-speech converter.

```
 314 GOSUB 1000
 316 NEXT
 318 GOTO 200
 320 FOR f = 0 TO 2
 322 c = PEEK(39009+f)
 324 GOSUB 1000
 326 NEXT
 328 GOTO 200
 330 FOR f = 0 TO 3
 332 c = PEEK(39012+f)
 334 GOSUB 1000
 336 NEXT
 338 GOTO 200
 340 FOR f = 0 TO 3
 342 c = PEEK(39016+f)
 344 GOSUB 1000
 346 NEXT
 348 GOTO 200
 350 FOR f = 0 TO 4
 352 c = PEEK(39020+f)
 354 GOSUB 1000
 356 NEXT
 358 GOTO 200
 360 FOR f = 0 TO 7
 362 c = PEEK(39025+f)
 364 GOSUB 1000
 366 NEXT
 368 GOTO 200
 370 FOR f = 0 TO 7
 372 c = PEEK(39033+f)
 374 GOSUB 1000
 376 NEXT
 378 GOTO 200
 380 FOR f = 0 TO 3
 382 c = PEEK(39041+f)
 384 GOSUB 1000
 386 NEXT
 388 GOTO 200
 390 FOR f = 0 TO 4
 392 c = PEEK(39045+f)
 394 GOSUB 1000
 396 NEXT
 398 GOTO 200
1000 POKE 40001, c
1010 CALL 40000
2000 CALL 40010
2010 IF PEEK(40100) = 2 THEN 2030
2020 GOTO 2000
2030 RETURN
3000 DATA      62,0,211,2,201,219,3,50,164,156,201
3050 DATA    43,60,53,4,46,15,15,11,4,13,31,4,29,14,19,4,40,40,58,4
3060 DATA   40,40,6,35,4,55,55,12,12,2,41,55,4,55,55,7,7,35,12,11,4,20,2,13,4,1
1,24,6,11,4

]
```

Program listing for a text-to-speech converter.