Cheep Talk

Build your own speech synthesizer

16K Cassette or Disk

by Lee Brilliant, M.D.

What do *2001*, *Star Trek* and *Star Wars* all have in common? Among other things, they all have that wonderful invention of the future, the talking computer! *2001* had HAL, the monotone malevolent; *Star Trek*'s counterpart was decidedly female; and from *Star Wars*, we have the proper-English-speaking C3-PO and his "binary"-speaking (don't *you* speak binary?) pal, R2-D2.

Now, your very own Atari home computer can be in the same class as these. All you need is a few hours and a few dollars to enter the world of **Cheep Talk**. Why a do-it-yourself speech synthesizer when there are several excellent speech products on the market already? Well, aside from the personal satisfaction of building it yourself, add-on devices are expensive in money, and software speakers are expensive in RAM. For under forty dollars, you can build your own **Cheep Talk**—it's easy to use and occupies little memory.

First, a little theory.

The subject of speech synthesis is very complex, and, though I make no pretense of this being a complete discourse on the subject, you need to know some language theory.

The English alphabet has twenty-six letters in it and about forty separate sounds called *phonemes*. The reason that there are more sounds than letters is that many letters have multiple sounds, or that multiple letters make single sounds. For example, the letter *g* can be hard, as in *go*, or soft, as in *gem*.

Actually, there are hundreds of variations of these phonemes, depending on how finely you can divide sounds. A sound such as a hard *g* can have slight differences in intonation and duration, depending on its location within a word and which vowels it associates with. These variations of phonemes are called *allophones*. **Cheep Talk** uses fifty-nine allophones.

How sounds are formed.

If you type in *SOUND 0,121,10,10*, you'll get a C note that sounds like a toy organ. The sound thus produced is a single sine wave, and only the pitch and volume can be changed.

Music made by the computer does not sound like a violin or a piano. Why not? The formation of music is more complicated, so that instruments differ from each other in sound quality in three ways: (1) the waveform, (2) harmonic content, and (3) the envelope shape.

Waveform is the primary sound determinant. A sawtooth waveform sounds harsh and sirenlike in comparison to a sine wave which sounds purer. Sharp, angular waves tend to have more harmonics, which are the second determinant in sound. A harmonic is a frequency which is a multiple (or fraction) of a primary frequency. That is, 200 Hz (cycles per second) is the first harmonic of 100 Hz (see Figure 1). Some instruments, like the organ, have few harmonics, while others—stringed instruments, for example—tend to have more.

Finally, the envelope is the overall sound/volume shape. A piano has a sharp rise in volume with a slow decay time, while a violin has a slower rise in volume and no decay, as long as the bow is moving across the string.

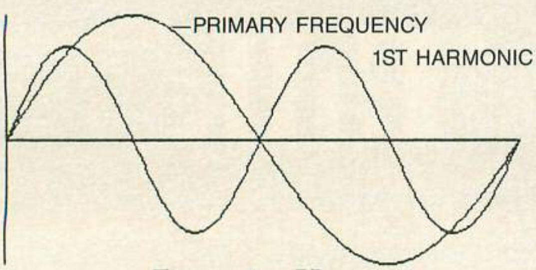


Figure 1. Harmonics.

Human speech is created in much the same manner as music. The vocal cords provide a primary wave shape called the *glottal pulse* (see Figure 2).

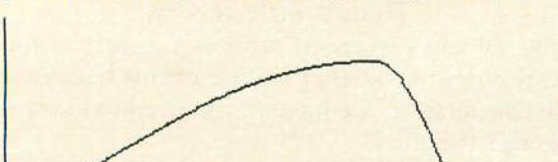


Figure 2. Glottal pulse.

The male pitch is around 141 Hz, while the female pitch is around 233 Hz. As with musical instruments, the tonal quality of a voice depends on the harmonics, which are created by the cavities of the vocal tract, including the sinuses, mouth, tongue, throat, etc. You can alter the sounds of speech by altering the size of these cavity resonators.

For example, say "aaah" while opening and pursing your lips. Doing this changes the "aaah" to "oh." Saying "aaah" and moving your tongue up and down changes the "aaah" to "eeee." Some sounds, such as "b," use the vocal cords, but some, like "s," do not. The ones that do are called *voiced sounds*, and the ones that don't are called *voiceless*.

If you take a look at the entire harmonics of human speech, you find that there are two major harmonic frequencies, plus the primary frequency, ranging from 100 to 2000 Hz. The relationship of these in volume and pitch give speech its characteristic sounds. These relationships, in turn, are controlled by all the parts of the vocal tract. Human speech is a complicated collection of variables, including wave shape, envelope shape and harmonics.

How do computers talk?

There are three major ways the computer forms speech: digital encoding, analog and linear predictive coding (LPC).

Digital encoding is best compared to today's digital audio. A voice signal is chopped up into slices, and the volume of that segment is converted from an analog or voltage to a binary number (see Figure 3). Each byte is stored in memory. When these bytes are read back in the same sequence and converted back to a voltage, it is integrated into a copy of the original signal.

To be of good quality, the signal must be sampled at a rate double the highest frequency to be used. If we cut off speech at 5000 Hz, then the sample rate should be at least 10000 bytes per second! At that rate, the word *hello* could use 5 to 10K of memory!

An example of this type of speech reproduction (not true synthesis) is the phone company's automated operators. While the reproduction quality is excellent, the vocabulary is limited. **SAM** (Software Automated Mouth) is a software example of this technique. It drives the sound channels directly, in machine language (see *De Re Atari*, section 7-21). Instead of a vocabulary of fixed words, **SAM**'s is made up of phonemes which can be hooked together to give a large vocabulary. Nevertheless, it's still memory hungry.

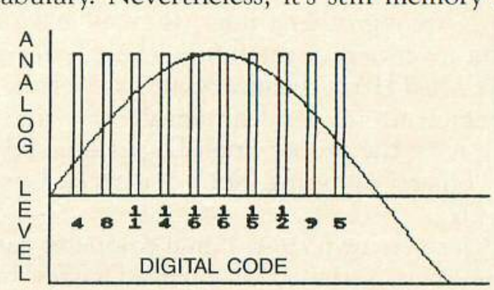


Figure 3.

Analog speech is based on a low frequency pulse generator for voiced sounds and a white noise generator for voiceless sounds. These are passed through tunable filters to select which harmonics will go on to the amplifier, much like today's multichannel audio equalizers.

The pitch and volume of the generators can also be varied, giving about nine separate adjustments which need to be continuously varied to create reproductions of voice. It would be difficult, indeed, for a human to twiddle all those knobs fast enough to make intelligible speech.

Instead, the computer can feed the processor appropriate parameters. Of course, the more frequently you feed the parameters, the more accurately you can define the sounds, and the more intelligible the speech. Also. . .the more memory is used. Thus, when a series of computer-defined parameters is passed to the processor, it sounds like speech.

LPC is somewhat "between" the two previous techniques. In structure, it is similar to analog synthesis using parameters to control 12-stage filters, volume controls and oscillator frequencies. Speech produced is better than analog, because the sample rate is higher. But, despite the high sample rate, computer memory requirements are very low.

This feat is accomplished by an onboard microprocessor and 8 to 16K of built-in ROM. The processor calculates most of the control settings based on complicated formulae in its ROM. So, while the control parameters may be updated from ROM every twenty or so milliseconds, the onboard microprocessor is calculating hundreds of intermediate values.

Your Atari doing the same thing in BASIC would take five to ten minutes to calculate what the speech processor does every twenty milliseconds. In this manner, the LPC system greatly improves the sample rate and the quality of speech, without the memory overhead of the digital system.

The SPO256-AL2 speech processor used in **Cheep Talk** is of the LPC type, as is the Texas Instruments **Speak 'n Spell**. LPC speech is not truly synthetic, because the control settings are obtained from digitally encoded speech patterns, and so are modeled after an original source.

The SPO256-AL2 does not have a fixed vocabulary; rather, it has a set of allophones programmed into its ROM which can be strung together to make words. Vocabulary is almost unlimited.

Building Cheep Talk.

Construction is fairly straightforward. Except for the 22pf capacitors, all parts are available at Radio Shack. The manufacturers of the SPO256-AL2 call for a 3.12 MHz crystal which can be specially ordered, but I used a 3.579 MHz color TV crystal without any difficulty. Be sure you get the right speech chip; Radio Shack carries two versions of the SPO256 speech processor.

The 28-pin socket and all components should be soldered in place *before* the integrated circuit is installed. Wiring can be done on perf board with point-to-point wiring or on a printed circuit. If you're into making your own board, then use the pattern shown.

After all parts except the chip are installed, wire in the joystick plugs. If you can obtain a couple of cords from old joysticks, and if they have all six wires called for, then use them. Otherwise, use 9-pin plugs and 6-conductor ribbon cable made by splitting the 25-conductor cable (see parts list).

Pinouts.

Connections for the joystick plugs are simple. If you are using the printed circuit, then the connections are numbered 1 to 4, plus 7 and 8. These correspond to the pin numbers on the plugs.

Figures 4, 5 and 6 show the actual pin connections. The views of the plugs are looking at the end which interfaces with the computer, so that the cable goes away from you. The term *port A* corresponds to joystick plug 1; *B* is plug 2.

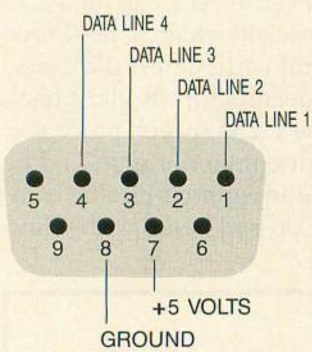


Figure 4.
Joystick plug 1.

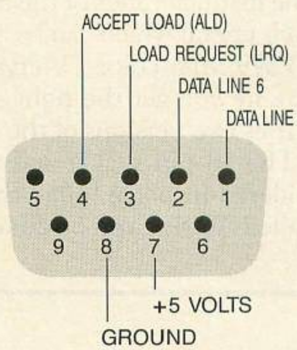
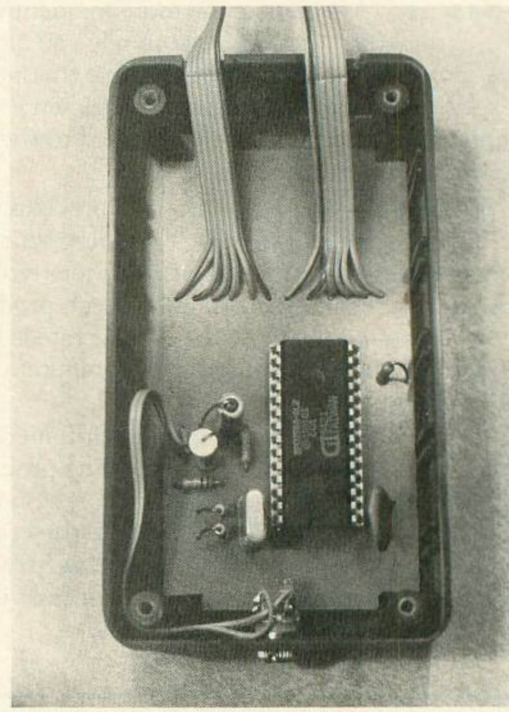


Figure 5.
Joystick plug 2.

The hoods for these plugs will not fit the computer without your removing the tabs at the ends, which hold the plug. Therefore, you need to use small flat-head machine screws to secure the plugs to the hoods.

Double check the order of your pin connections and verify that no solder has bridged any connections on the circuit board. Mount the circuit board and earphone jack in the box and attach the audio output leads to the jack. Make an opening for the joystick cables and label the plugs, so you put them in the right sockets.



Internal arrangement of the Cheep Talk voice synthesizer.

Triple check all wiring, then install the chip. *Note* that there is a small dot on the top of the chip over pin 1. Locate the chip properly and install in the socket by pushing gently and evenly, being sure not to bend any pins. While digital chips are very forgiving of wrong wiring, they give up when faced with static electricity. So ground yourself before handling the chip and leave it in its black conductive foam until the last minute. Close up the case.

PLUG #2				PLUG #1		
JOYSTICK PIN	FUNCTION	CHIP PIN	SPO-256	CHIP PIN	FUNCTION PIN	JOYSTICK
PIN 4	ACCEPT LOAD	9		18	DATA 1	PIN 1
PIN 3	LOAD REQUEST	20		17	DATA 2	PIN 2
PIN 2	DATA 6	13		16	DATA 3	PIN 3
PIN 1	DATA 5	14		15	DATA 4	PIN 4
PIN 7	+5 VOLTS				+5 VOLTS	PIN 7
PIN 8	GROUND				GROUND	PIN 8

Figure 6. Pinout schematic.

Once assembled, plug in the joystick cables and connect up to an amplifier. I use a small unit sold by Radio Shack, which has its own amplifier and speaker in a compact case. You can also connect **Cheep Talk** to your Atari's cassette audio input line, so it will play through the TV.

To do this, either put a clip lead on pin 11 of the serial jack, or open the plug on your serial cable and solder a wire to the number 11 connector, bringing it out the back of the plug. Attach it to the center lead of a miniature phone plug and connect to **Cheep Talk**. No ground wire is needed.

If the volume isn't enough, jump C3 to pin 24 of the IC and eliminate C2 and R2. Stay away from stereo equipment or plug-in amplifiers, unless they're properly grounded or isolated. Failure to heed this warning may fry your computer! Power for **Cheep Talk** comes from your Atari through the joystick ports.

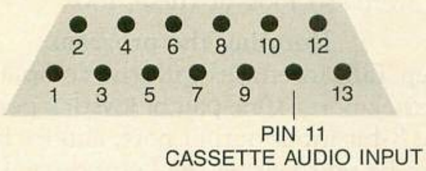


Figure 7. Serial plug.

Using Cheep Talk.

To use your new toy, just type in the short program and run it with **Cheep Talk** installed. If everything is right, the computer should talk to you and say, "Hello. This is **ANALOG Computing** magazine **Cheep Talk**." If it doesn't, recheck all wiring, especially the order of the wires in the joystick plugs and, finally, your program typing. If it works, onward!

To make **Cheep Talk** say what you want, we must return to language theory briefly. The SPO256-A2 has fifty-nine allophones (or speech sounds), and five pauses numbered 0 to 63. The trick is to know which ones to use.

Table 1 contains a listing of the allophones by phonetic groupings. Don't worry, it's not that hard. To code a word into data statements, you first need to break the word down to its sounds. Spelling has *no bearing* on sounds! For example, you will find no letter *c*. Instead, you use *s* if it is a soft sound, and *k* if it is hard.

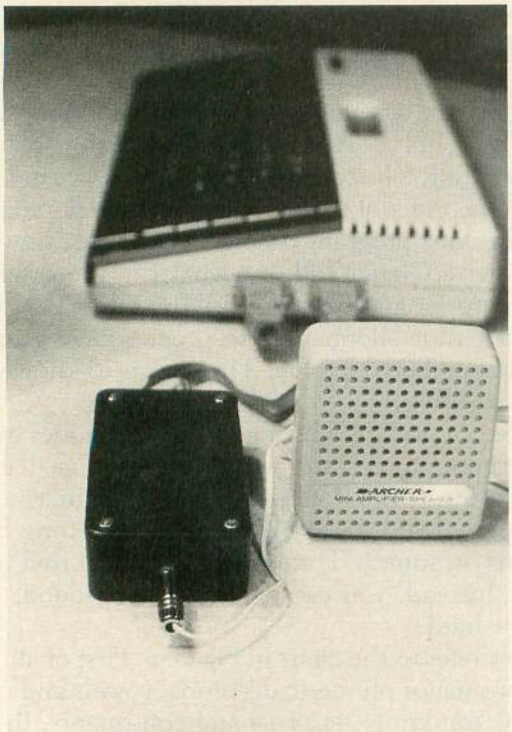
Now refer to the chart in Figure 8. First of all, there are two major phonetic divisions: vowels and consonants. You know—*a,e,i,o,u* and consonants. In turn, there are three types of vowels: long (like in *bE*), short (like in *bEd*) and diphthongs or blends (like in *bEAr*).

(continued on next page)

Table 1. Allophone listing.

PAUSES			
0	10 ms.	(PA1)	Before *p, t, k, ch* and between words
1	30 ms.	(PA2)	Same
2	50 ms.	(PA3)	Same
3	100 ms.	(PA4)	Between clauses and sentences
4	200 ms.	(PA5)	Same
VOWELS			
SHORT:			
* a	26	(AE)	bAt, mAp
* e	7	(EH)	sEt, tEnt
* i	12	(IH)	sIt, kItten
* o	24	(AA)	pOt, mOp
* u	15	(AX)	Up, lApel, trUck
LONG:			
a	20	(EY)	cAke, grEAt, grAte
e	19	(IY)	spEAk, pEEk, pEOple, pennY
i	6	(AY)	kIte, skY, mIght
o	53	(OW)	gO, snOW, clOse, zOne
u	22	(UW1)	After *y* sounds: yOUth
	31	(UW2)	In monosyllable words: twO, tOO, tO, shOE, fOOd
DIPHTHONGS (BLENDS)			
R COLORED:			
ar	59	(AR)	fARm, gARment
air	47	(XR)	hAIR, stARE
er	51	(ER1)	stIRring, fURniture, lettER
	52	(ER2)	In monosyllable words: bIRd, fERn, bURn
ear	60	(YR)	hEAR, pEER, IRresistible
or	58	(OR)	fORturn, stORe
OTHERS:			
* aw	23	(AO)	AWful, sOng, tALk
oi	5	(OY)	vOIce, tOY
ow	32	(AW)	sOUnd, dOWn
* uh	39	(UH)	cOOkie, fUll
ul	62	(EL)	littLE, angLE, gentLE
CONSONANTS			
STOPS — VOICED:			
b	28	(BB1)	Final position, between vowels and in blends with other consonants: riB, fiBer, BLend
	63	(BB2)	Initial position before a vowel: Bat
d	21	(DD1)	Final position: saiD, enD
	33	(DD2)	Initial position and in blends with other consonants: Down, DRain
g	36	(GG1)	Before the vowels ā, ē, ĕ, ĭ, *ear*, and *air*: Gear, Gift, Gate
	61	(GG2)	Before the vowels ō, *oi*, ū, ŭ, *uh*, and in blends with other consonants: Gun, GReen
	34	(GG3)	Before the vowels ă, ī, ŏ, *ar, er, or, aw, ow*, blends in the middle of words, and final position: Gap, aGRee, peG
CONSONANTS continued			
STOPS — UNVOICED:			
p	9	(PP)	Please, amPle, triP
t	17	(TT1)	Before *s* and in final position blends with other consonants: gaTeS, STop
	13	(TT2)	ALl other positions
k	42	(KK1)	Before vowels ā, ă, ē, ĕ, ī, ĭ, ŭ, *er, air, ear*, and initial blends with other consonant: Cute, Kit, Came, sCream, Clown
	41	(KK2)	Final position and final blends with other consonants: speaK, cliCK, taSK
	8	(KK3)	Before vowels *ar, aw*, ō, *oi, or*, ū, *uh*, and in initial blends with consonants: Coin, Collide, sCream, Clown
FRICATIVES — VOICED:			
v	35	(VV)	Vest, proVe, eVen
dh	18	(DH1)	Initial *th* sound: THis, THey, THen
	54	(DH2)	Final position and between vowels: baTHing
z	43	(ZZ)	Zoo, phaSE
zh	38	(ZH)	beiGE, pleaSure
FRICATIVES — VOICELESS:			
* f	40	(FF)	Find, Finger
* th	29	(TH)	THin, wiTH
* s	55	(SS)	Sit, Single
sh	37	(SH)	SHirt, wiSH
h	27	(HH1)	Before vowels ā, ă, ē, ĕ, ĭ, *ear, air*: Hat, Hair
	57	(HH2)	Before vowels ō, *oi*, ū, *uh, aw, or, ar*: Harm, Hoist, Home
wh	48	(WH)	WHite, WHim, tWenty
RESONANTS			
w	46	(WW)	We, War, langUage
r	14	(RR1)	Initial position: Red, Robot, WRite
	39	(RR2)	In initial blends with consonants: BRown, GRease. In the middle, use R-colored vowels.
l	45	(LL)	Like, heLLo, steeL
y	49	(YY1)	In blends: bEAuty, comp(Y)uter, c(Y)ute
	25	(YY2)	Initial position: Yes, Yarn
NASALS			
m	16	(MM)	Milk, alarM, aMple
n	11	(NN1)	Before vowels ā, ă, ē, ĕ, ī, ĭ, ū, ŭ, *er, ear, ow, air*, and final consonant blends: Name, Now, Nervous, earN, turN
	56	(NN2)	Before vowels ō,ŏ,*oi, or, ar, uh*: Note, North, Noise
bg	44	(NG)	striNG, aNGer
JH-CH			
jh	10	(JH)	fuDGe, inJure
ch	50	(CH)	CHurch, CHeer, feaTure, matCH

* *These sounds can be doubled for long sounds.*



The finished Cheep Talk synthesizer.

Consonants are either stops, which are short, explosive sounds like *buh* or *guh*, fricatives like *v*, resonants like *r*, or nasals like *m*, which you can't say with your nose pinched. Stops and fricatives can be voiced or voiceless, depending on whether the vocal cords are used—as in *b* or whispered like *p*. There are two consonant blends which do not fit: *ch* and *j*, because they are both fricatives and voiced stops.

To use Table 1, you must break your words down into sounds. Decide if each is a vowel or consonant, and which type, then look it up in the table. Some sounds have several versions, such as g, so read the short rules next to the sound. Once you have the right sound, find its number and place it in order in the data statement. Also included are the phonetic code and some example words.

Let's try "hello." The sounds are *h, e, l, o*. *H* is a consonant, a fricative type, which is voiceless. Its number is 27. There are two *h* sounds. Number 27 goes with the short *e* (code EH), while 57 goes with other vowels. The *e* is a short vowel whose number is 7; *l* is a resonant consonant numbered 45; and, finally *o* is a long vowel numbered 53. So "hello" is really 27,7,45,53. The dictionary that comes with the chip adds an extra vowel before the *o*. Some experimentation may be needed. Pauses are sometimes used within words so look at these, too.

Running the program.

Cheep Talk interfaces with the computer through the joystick ports. One pair of joystick ports make up a single 8-bit input/output port, and each bit of this port can be set to either send or receive. Line 10 sets this up for joysticks 1 and 2 (you can use locations 54017 and 54019 for joysticks 3 and 4 on the Atari 400s or 800s).

The first 7 bits of the port are set to transmit, and the eighth bit to receive. The sixty-four allophones and pauses use the first 6 bits to code them in binary. By dropping from +5 volts to 0, bit 7 tells the processor that an allophone number is on the first six lines. The processor accepts the load and begins

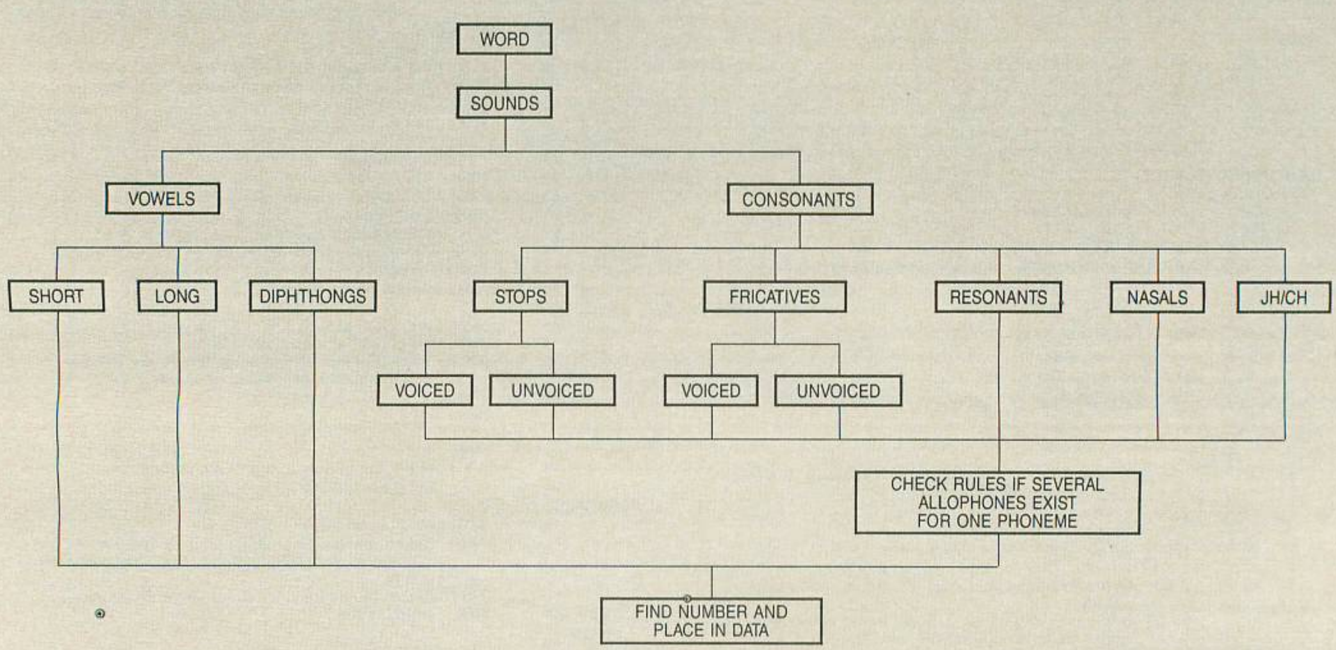


Figure 8. Logic chart.

PARTS LIST

#	Value	Radio Shack #
C1	.1 μf 50-volt capacitor	272-1069
C2	.02 μf 50-volt capacitor	272-1066
C3	1 μf 16-volt capacitor	272-1434
C4,C5	22 pf capacitors	
IC1	SP0256-AL2 speech chip	276-1784
J1	miniature phone jack	274-251
R1	100K ¼-watt resistor	271-1347
R2	33K ¼-watt resistor	271-1341
R3	10K ¼-watt resistor	271-1335
XTAL1	3.579 MHz TV crystal	272-1310
MISCELLANEOUS		
2	9-pin female D plugs	276-1538
2	hoods for plugs	276-1539
1	ribbon cable	278-772
1	plastic case	270-222
1	circuit board	276-162
1	28-pin DIP socket	276-1997
OPTIONAL		
Amplifier		277-1008

talking. At the same time, it sets its load request (LRQ) line to high, until it's done making its sound.

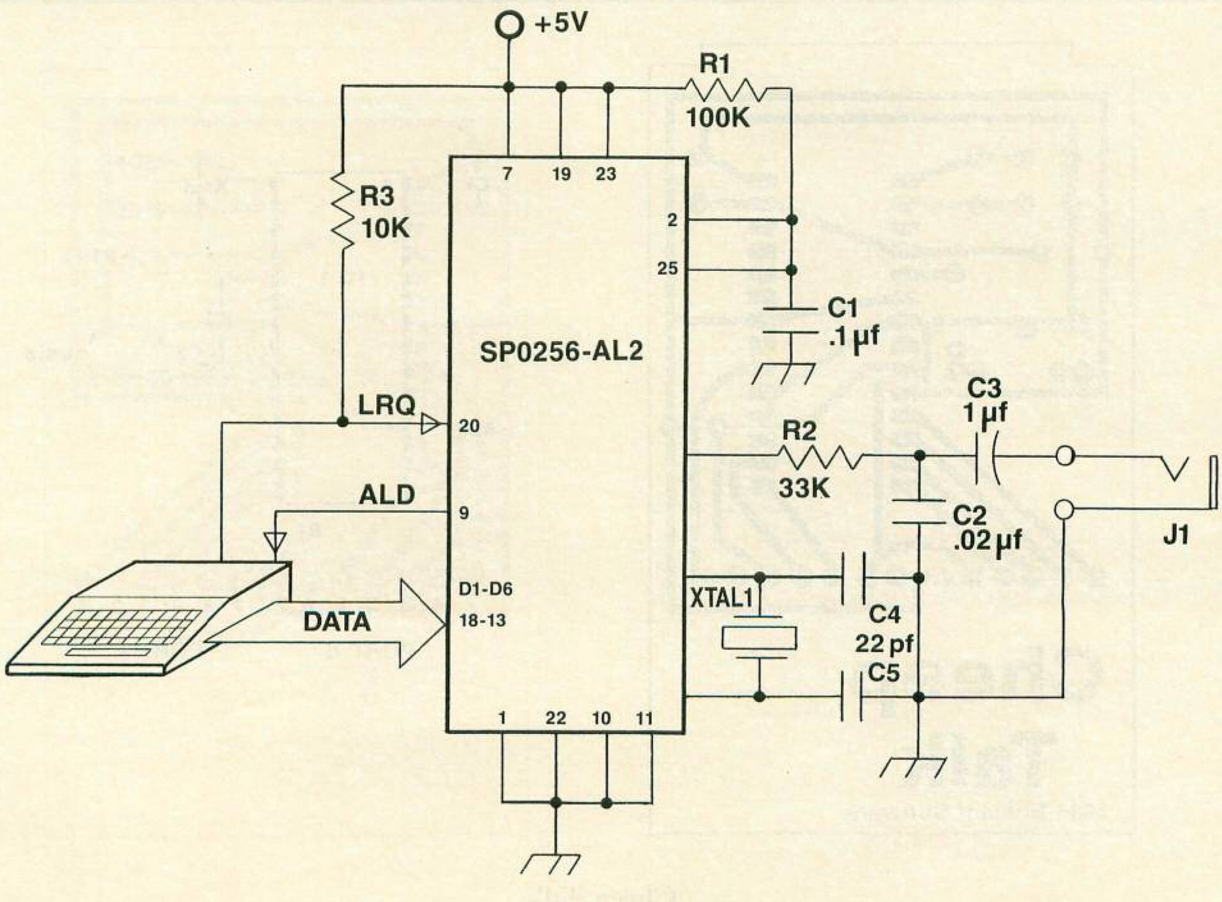
Program Line 130 reads LRQ on bit number 8 and keeps the Atari from forcing a new load until the processor is done. When finished, LRQ drops low, and your computer responds by loading the next allophone address and strobing bit 7. This "handshaking" keeps things in order.

One final word—as with most other new areas of exploration, it takes a lot of practice to be good. Don't get discouraged if, at first, **Cheep Talk** is hard to grasp; just keep at it. If nothing else, you might understand why English is one of the most difficult languages to master. . .especially to those who speak German or Spanish, where there are twenty-six letters and only twenty-six phonemes!

The uses for **Cheep Talk** are many. Consider the possibilities of programs for the unsighted, instructional tutorials, verbal instructions for your programs, or interactive games. Beam me up, Scotty! □

(Program listings and circuit board design start on page 66)

Reference: *Electronically Speaking: Computer Speech Generation*. John P. Cater, publ. Howard W. Sams & Co. 1983.



Cheep Talk Schematic.

Listing 1.

```
100 P=PEEK(54018):POKE 54018,P-4:POKE
54016,127:POKE 54018,P
110 TRAP 500
120 FOR S=1 TO 99:READ D
130 IF PEEK(54016)>128 THEN 130
140 POKE 54016,D+64:POKE 54016,D:NEXT
S
290 DATA 27,7,45,15,53,4,4,18,12,55,55
,55,3,12,43,3,26,11,15,15,45,24,24,1,3
4,3,8,24,16,0,9,22,13,12,44,55,3
300 DATA 50,19,1,9,3,13,23,23,1,41,0
500 END
```

•

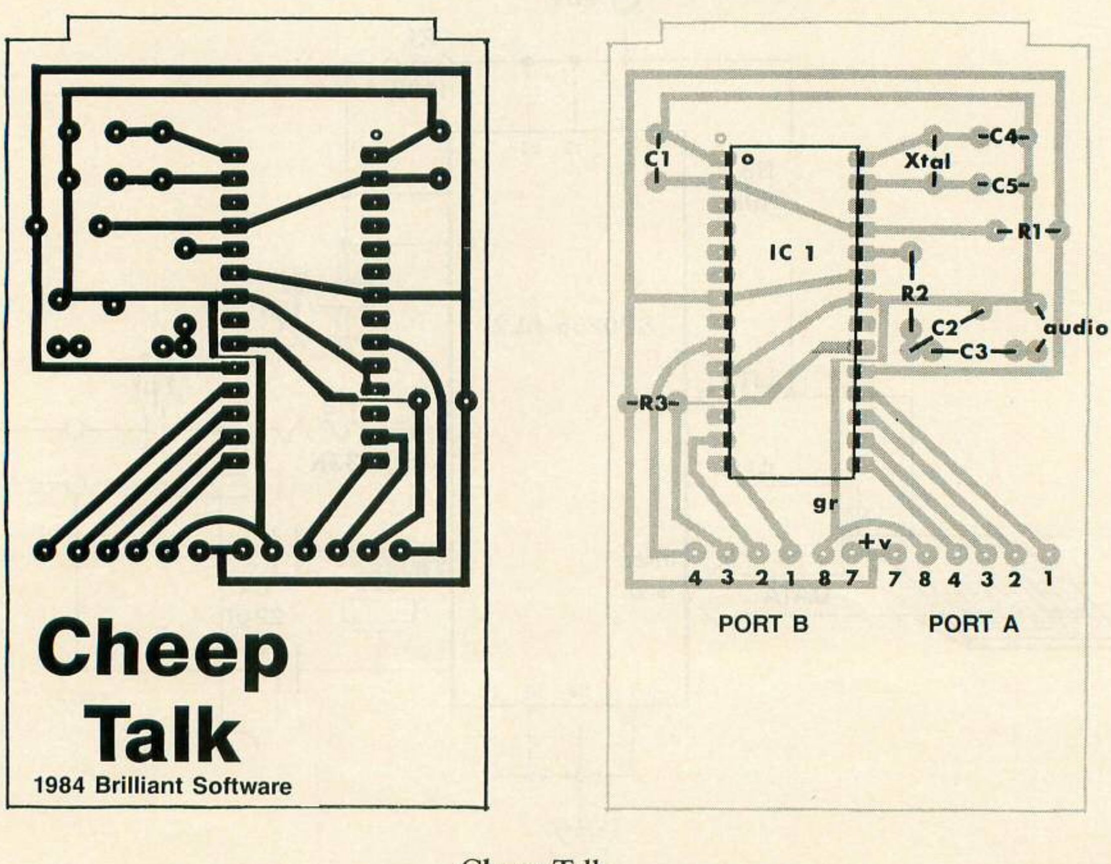
CHECKSUM DATA.
(see page 32)

```
100 DATA 46,701,119,904,847,970,960,34
,4581
```

•

Listing 2.
First Words.

```
0 REM ************************
1 REM * FIRST WORDS          *
2 REM * by Lee Brilliant MD. *
3 REM * for CHEEP TALK       *
4 REM * 1984                 *
5 REM ************************
10 GOTO 100
20 READ N
25 FOR S=1 TO N:READ D
30 IF PEEK(54016)>127 THEN 30
35 POKE 54016,D+64:POKE 54016,D:NEXT S
:A=1^1:POKE 54016,64:POKE 54016,0:RETU
RN
48 DATA 4,43,60,53
49 DATA 3,46,15,11
50 DATA 2,13,31
51 DATA 4,29,14,19
52 DATA 3,40,40,58
53 DATA 5,40,40,6,35,2
54 DATA 7,55,55,12,12,2,41,55
55 DATA 7,55,55,7,7,35,12,11
56 DATA 3,20,2,13
57 DATA 4,11,24,6,11
65 DATA 1,20
66 DATA 2,63,19
67 DATA 3,55,55,19
68 DATA 2,33,19
69 DATA 1,19
70 DATA 4,7,7,40,40
71 DATA 2,10,19
72 DATA 4,20,1,2,50
73 DATA 2,24,6
74 DATA 3,10,20,20
75 DATA 3,42,7,20
76 DATA 3,7,7,62
77 DATA 3,7,7,16
78 DATA 3,7,7,11
79 DATA 1,53
80 DATA 2,9,19
81 DATA 3,42,49,31
82 DATA 1,59
83 DATA 4,7,7,55,55
84 DATA 2,13,19
85 DATA 2,49,31
```



Cheep Talk
Printed Circuit Board Layout.

```
86 DATA 2,35,19
87 DATA 7,33,15,1,63,62,49,31
88 DATA 6,7,7,2,41,55,55
89 DATA 2,46,6
90 DATA 2,43,19
95 DATA 18,14,23,44,3,42,19,4,4,17,39,
6,4,15,2,36,7,7,11
100 P=PEEK(54018):POKE 54018,P-4:POKE
54016,127:POKE 54018,P
110 X=7:Y=3
120 GRAPHICS 18:POKE 712,44:GOSUB 500
125 POKE 16,64:POKE 53774,64
150 OPEN #1,4,0,"K:"
160 GET #1,K
170 IF PEEK(694)=128 THEN POKE 694,0:K
=K-128
180 IF PEEK(702)<>64 THEN POKE 702,64:
K=K-32
200 IF K>47 AND K<59 THEN 230
210 IF K>64 AND K<91 THEN 230
220 SOUND 1,100,10,10:A=1^1:SOUND 1,20
0,10,10:A=1^1:SOUND 1,0,0,0:RESTORE 95
:GOSUB 20:POKE 764,255:GOTO 160
230 POSITION X+2,Y+2:? #6;CHR$(K)
240 RESTORE K:GOSUB 20
250 X=INT(RND(0)*16):Y=INT(RND(0)*8):?
 #6;"K":GOSUB 500:POKE 764,255:GOTO 16
0
500 POKE 712,4+16*INT(RND(0)*16):POKE
708,16*INT(RND(0)*16)+10
510 COLOR 42:PLOT X,Y:DRAWTO X+4,Y:DRA
WTO X+4,Y+4:DRAWTO X,Y+4:DRAWTO X,Y:RE
TURN
```

•

CHECKSUM DATA.

(see page 32)

```
0 DATA 1,415,675,809,718,11,435,332,13
2,511,895,198,179,859,162,6332
52 DATA 177,733,157,314,966,264,585,89
6,199,893,603,411,865,396,681,8140
74 DATA 155,983,154,160,153,611,706,17
8,598,444,879,895,893,194,799,7802
89 DATA 709,878,763,46,485,948,639,273
,568,348,319,543,525,861,291,8196
240 DATA 309,458,155,119,1041
```

•