SpeechLab

Heuristics
900 N. San Antonio Rd.
Los Altos, CA 94022

GETTING STARTED WITH SPEECHLAB

1. Read paragraph 2.0 on usage hints (it's short).
2. Insert Speechlab into slot 3.
3. Enter Basic (control-B, Carriage return).
4. Type LOMEM: 5500
5. Load or type in a demonstration program from Section 3.
6. Hold the microphone 1" from your mouth and speak into it, not over it.
7. Run the program and enjoy!

TABLE OF CONTENTS

1.1 INTRODUCTION

This manual contains the information required to use Heuristics Speechlab Model 20A with your Apple II computer. The manual is designed to allow the user to start using Speechlab with a minimum amount of reading or technical understanding of the product. The first section of the manual is a brief overall description of Speechlab and its capabilities. The second section of the manual contains several example basic programs and instructions, which the user can immediately load and execute. The reader should read Section 1.2, General Description, and then enter several of the programs in Section 2 and use Speechlab. These examples illustrate the various uses and capabilities of the device, and enable the user to get experience with the device before continuing to the explanations for interfacing Speechlab to the user's own programs.

Following the section on the software interface are the sections on hardware and software theory of operation, which the user may read at his leisure. These sections explain in detail how the device functions from a hardware and software point of view.

Heuristics is especially interested in receiving comments and ideas on improving the product and the documentation for the product. There is space on the warrantee card for your comments. Please note any sections of the manual that you felt were confusing or that could have been better explained, or ideas on improving the product. All comments are highly appreciated.

1.2 SPEECHLAB MODEL 20A DESCRIPTION

Speechlab Model 20A is an Apple II compatible speech recognition peripheral that allows the user to enter control information or data into a computer via spoken words. It is a combined hardware/software product in that the host computer receives raw data from the hardware board and processes the data with a program to determine the identity of the spoken word. The program to process the data is contained in a ROM located on the hardware board. When using the system, control is passed to the peripheral, which then executes its own program to determine the spoken word and returns the word to the calling program as a string. This procedure takes place in accordance with the Apple II conventions for "smart" peripherals so that data returned from Speechlab is equivalent to data entered via the keyboard or any other input device.

Technically speaking, Speechlab is a speaker-trained isolated word speech recognition system with a 32 word vocabulary. Speaker-trained means that the user must enter one example of each word in the vocabulary, and it also means that there are two modes of operation of the device. First, a training mode in which the examples of each word are provided, and then a recognition mode in which previously trained words are recognized. Isolated word means that the word must have a .1 sec. period of silence before and after the word. This is necessary to allow the computer to determine when a word has started and when it is finished. Up to 32 words may be present in the vocabulary at any given time.

Speech recognition is an exciting new technology which has just recently moved from the Artificial Intelligence Laboratory into the world of practical commercial products. These products typically cost several orders of magnitude more than Speechlab. As a result of combining judicial cost/performance analysis with an innovative research and development program, Speechlab can provide substantially identical performance at a small fraction of the cost of other presently available units.

2.0 DEMONSTRATION PROGRAMS AND APPLICATION EXAMPLES

The following section contains a number of basic programs to familiarize the user with Speechlab operation and to provide examples of the capabilities of speech input to the computer. For best results the following operational techniques should be observed.

1. Hold the microphone directly in front of the mouth as shown in Figure 1. Distance from the microphone should be one inch, and the speaker should speak into the microphone, NOT over the microphone. Figure 2 shown INCORRECT microphone position. THIS IS THE SINGLE MOST IMPORTANT KEY TO BEST RESULTS. With the microphone held in the hand as far up the microphone as possible, the user should be able to extend his thumb and touch his chin. (This feels surprisingly close, as the microphone looms rather large in the field of vision.)

2. Be consistent about the distance from the microphone to the mouth. The Speechlab program compensates to a large degree for varying amplitudes, but works best when the distance to the microphone and the volume of the speech utterance is the same for both the training sample and unknown utterance to be recognized. With a small amount of practice this becomes natural.

3. Speak naturally, and just slightly louder than usual. This works much better than trying to speak more clearly or more slowly than usual.

4. Do not move the microphone cord during or just preceding a word as this generates noise that is audible to Speechlab. Get the microphone in position before you need to speak a word.

5. Operate in a reasonably quiet environment, i.e. typical home or office environment. An optional noise-cancelling close-speaking headset microphone is available for use in extremely noisy environments.

Recognition Guidelines

Your Heuristics Speechlab is capable of recognizing up to 32 words with a high degree of accuracy. However, many words sound alike, so to get the best results you should follow a few simple rules.

First of all, the greater the acoustic difference between two words the easier it is to tell them apart, so choose words that differ by at least one full syllable. For example, "sixty" and "sixteen" don't differ in one full syllable, so they will be hard to tell apart.

Multisyllabic words are easier to recognize because they have more information than one syllable words, so use longer words when you can. In fact, you can even use phrases if they are less than 1.5 seconds long and have no pauses in them.

Some sounds are particularly hard to tell apart:

M, N

S, SH, CH, Z

L, R

so avoid words that differ only by these sounds. For instance, avoid "mine" and "nine" (they don't differ by one syllable anyway).

Low level sounds, mostly consonants, like "f" and "v" can easily be lost in background noise, so you'll get your best results by holding the microphone very close to your mouth (less than one inch away), speaking a little louder than normal, and working in a quiet room. If you must work in a noisy environment, we recommend using our optional noise cancelling microphone.

Whenever possible, you should use more than one training sample of each word to get the best performance, especially on the "hard" words. If the board confuses some words, try increasing the number of samples of those words.

These techniques require conscious consideration for the first few uses of Speechlab, after which they rapidly become second nature. By taking the time to observe the preceding points the first few times the system is used, the user guarantees the best performance from the deivce.

The following programs demonstrate the use of Speechlab in conjunction with Basic programs. If you are using Speechlab in a slot other than slot 3, the peripheral connector with the 3 directly above it, be sure to set the Basic variable SLOT equal to the slot the Speechlab is in. This statement is generally the first or second statement in each of the demonstration programs. ALSO REMEMBER TO SET LOMEM:5500 BEFORE RUNNING ANY SPEECHLAB PROGRAM. HAVE FUN!

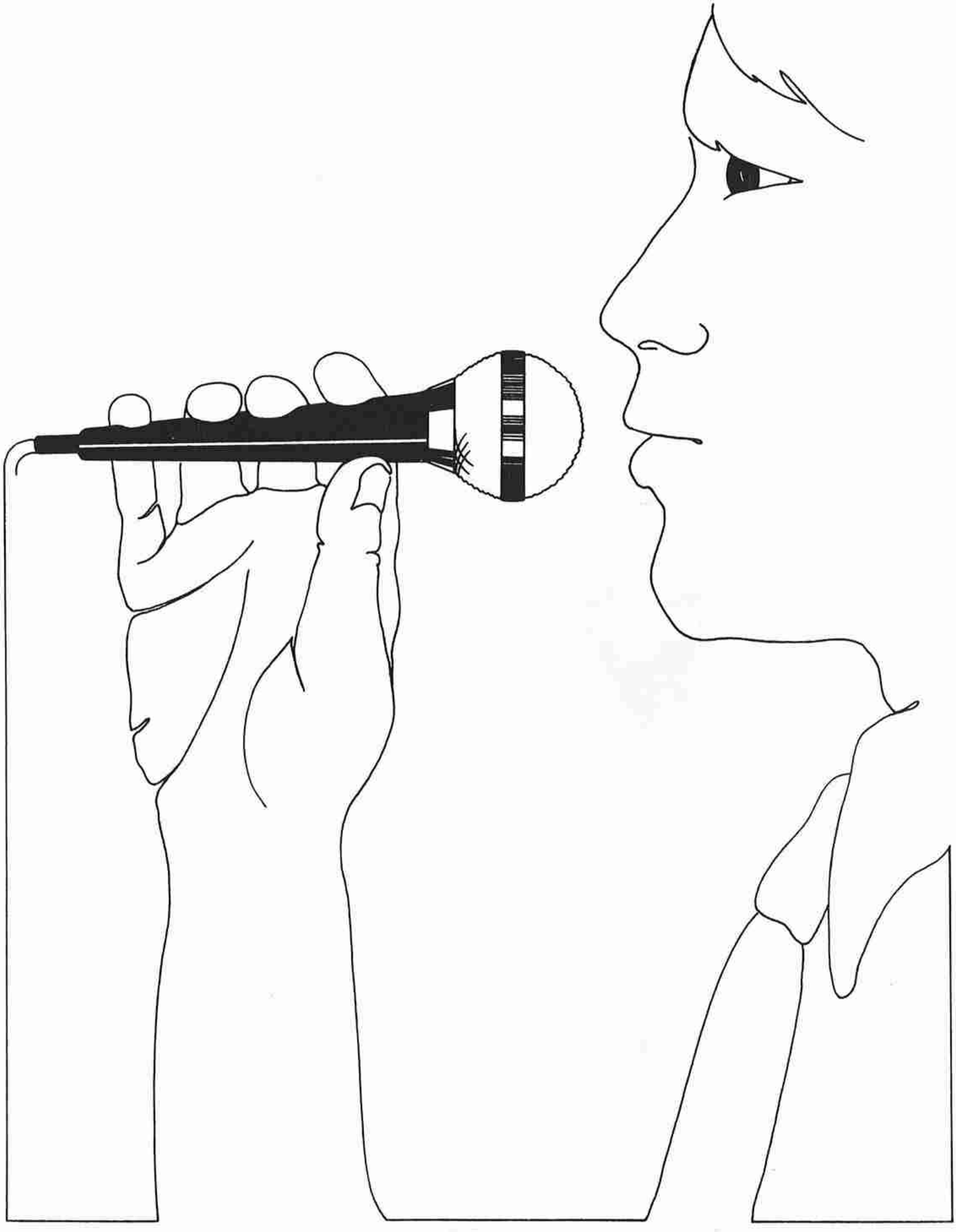


FIGURE 1
RIGHT

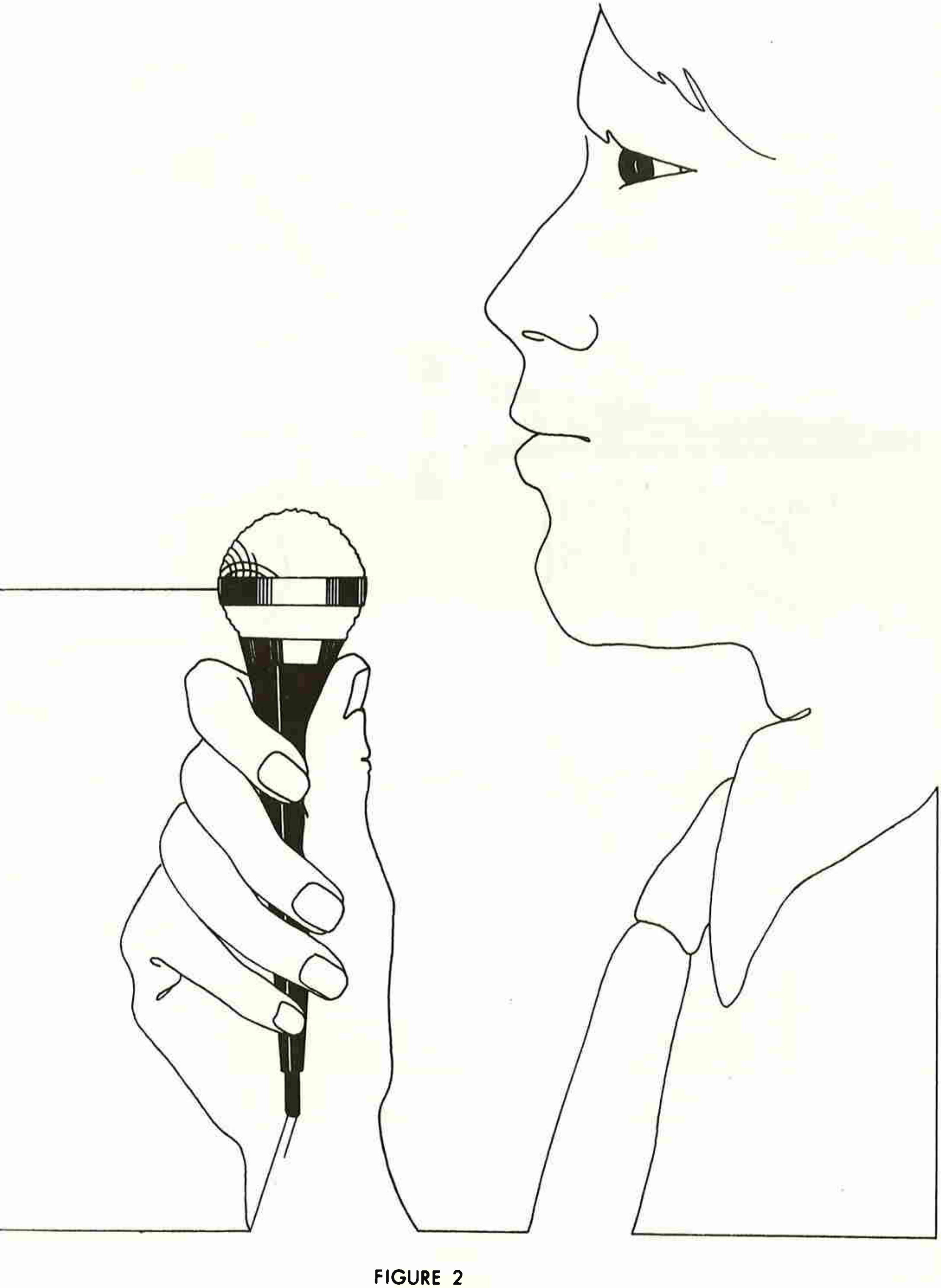


FIGURE 2
WRONG

2.1 VOICE PLOT

This program plots the voice data as seen by the Speechlab board. Every time you say something into the microphone (be sure the microphone switch is on) the program plots the energy of the utterance on the top of the screen in green, the low band frequency in the middle of the screen in orange, and the high band frequency on the bottom of the screen in blue. For example, if you speak the word "six" into the microphone, you will see a large amount of high frequency in the "SS" sound at the beginning and end of the word, and lower frequencies in the middle. This shows graphically the type of information used by Speechlab to discriminate between words.

SPEECHLAB
PLOT PROGRAM

```
 90     REM SET LOMEM:5500 BEFORE RUNNING PROGRAM
100     SLOT=3 : REM SPEECHLAB SLOT
110     DIM W$(5): BUF=2137
120     PR#SLOT: PRINT: PR#0
130     GR:PRINT "SAY SOMETHING PLEASE..."
140     PR#SLOT: PRINT "WORD":PR#0
150     L=PEEK(2133) : REM LENGTH OF UTTERANCE
160     IF L>39 THEN L=39: REM ONLY PLOT FIRST 1/2 SECOND
170     FOR I=0 TO L-1
180     COLOR=12 REM ENERGY
190     PLOT I, 8-PEEK (BUF + I*4)
200     COLOR=9 REM LOW BAND FREQUENCY
210     PLOT I, 39-PEEK (BUF+I*4+2)/4
220     COLOR=6 REM HIGH BAND FREQUENCY
230     PLOT I=39 -PEEK (BUF+I*4+3)/4
240     NEXT I
250     INPUT "ANOTHER PLOT",W$
260     IF W$(1,1)="Y" THEN 120
270     END
```

2.2 SHOOTING STARS

In this game, the player shoots at stars by calling out a position from 1 to 9, corresponding to a position on a 3x3 matrix, which contains a star. The star will disappear and new stars will appear according to the following rules:

1. If the center star is shot (5) then the stars above and below, and to the right and left of center will appear if there was no star there, or will disappear if they were there.

2. If a corner star is shot, then the stars adjacent to it including the center star will complement; that is they will appear if they were not there, or disappear if they were there.

3. If a star in the middle of a horizontal or vertical row is shot (not counting the center star) then the other stars in the horizontal or vertical row will complement.

4. The object of the game is to have all stars except the center star present, in which case you win. If all stars disappear, you lose.

SHOOTING STARS

```
1    REM SET LOMEM: 5500 BEFORE RUNNING PROGRAM
2    SLOT=3 : REM SPEECHLAB SLOT NUMBER
5    GOTO 400
10   GR
20   FOR X=6 TO 33 STEP 13:
     FOR Y=6 TO 33 STEP 13
25   T=0 : IF X=19 AND Y=19 THEN T=1
30   GOSUB 1000
40   NEXT Y,X
50   DIM S(9) : FOR I=1 TO 9:S(I)=0: NEXT I:S(5)=1
60   CALL -936
65   TAB 20:PRINT "1 2 3":TAB 20:
     PRINT "4 5 6":TAB 20:PRINT "7 8 9":
     VTAB 22: POKE 33,11: POKE 34,22: POKE 35,22
70   VTAB 22: TAB 1: PRINT "SHOOT...";:IN#SLOT
     INPUT A: IN#0: PR#0
80   IF A<1 OR A>9 THEN 70
90   IF S(A)=1 THEN 120
100  VTAB 24: POKE 33,39:
     PRINT "YOU CAN'T SHOOT A BLACK HOLE!";:
     VTAB 24: TAB 1
110  FOR I=1 TO 1500: NEXT I: VTAB 24:
     CALL -868: POKE 33,11: GOTO 60
120  X=13*((A MOD 3)-1)+6
130  Y=13*((A-1)/3)+6
150  S(A)=0
160  GOTO 200+10*A
210  S(2)=1-S(2):S(4)=1-S(4):S(5)=1-S(5):GOTO 300
220  S(1)=1-S(1):S(3)=1-S(3):GOTO 300
230  S(2)=1-S(2):S(5)=1-S(5):S(6)=1-S(6):GOTO 300
240  S(1)=1-S(1):S(7)=1-S(7):GOTO 300
250  S(2)=1-S(2):S(4)=1-S(4):S(6)=1-S(6):S(8)=1-S(8):GOTO 300
260  S(3)=1-S(3):S(9)=1-S(9):GOTO 300
270  S(4)=1-S(4):S(5)=1-S(5):S(8)=1-S(8):GOTO 300
280  S(7)=1-S(7):S(9)=1-S(9):GOTO 300
290  S(3)=1-S(8):S(5)=1-S(5):S(8)=1-S(8)
300  FOR I=1 TO 9
310  T=S(I)
320  X=13*((I-1) MOD 3)+6
330  Y=13*((I-1) /3)+6
340  GOSUB 1000
350  NEXT I
355  T=0
360  FOR I=1 TO 9:T=T+S(I):NEXT I
370  IF T THEN 380: POKE 33,39: POKE 34,20:
     POKE 35,24: CALL -936: TAB 17:
     PRINT "YOU LOSE!"
375  FOR I=1 TO 250: J=PEEK(-16336):
     NEXT I:VTAB 22:END
```

```
380    IF T#8 OR S(5)#0 THEN 70
390    POKE 33,39: POKE 34,20: POKE 35,24:
       CALL -936:TAB 17:PRINT
       "YOU WIN!":VTAB 22:END
400    TEXT : CALL -936
405    PR#SLOT:PRINT:PR#0:
       :REM INITIALIZE SPEECHLAB
410    VTAB 5:TAB 8:
       PRINT "SPEECHLAB SHOOTING STARS"
420    VTAB 10:PRINT "TURN ON MICROPHONE AND SAY...":
       VTAB 12:FOR I=1 TO 1000:NEXT I
422    FOR I=1 TO 3
425    FOR A=1 TO 9:PRINT "THE NUMBER";A;
       " ":PR#SLOT:PRINT A:PR#0:NEXT A:NEXT I
430    VTAB 23:PRINT "GOOD! LET'S START...":
       FOR I=1 TO 1000:NEXT I:GOTO 10
999    END
1000   IF T=0 THEN 1500
1005   COLOR = 15
1010   FOR DX=-2 TO 2
1020   PLOT X,Y+DX:PLOT X+DX,Y+DX:
       PLOT X+DX,Y:PLOT X-DX,Y+DX:
       NEXT DX:RETURN
1500   COLOR=0:GOSUB 1010
1510   COLOR=15:PLOT X,Y
1520   RETURN
```

2.3 MASTERMIND

The object of this game is to correctly guess the color of each of the five bars displayed on the screen in the minimum number of moves. A move consists of entering a color guess for each of the bars. After the move, the computer responds with a white square to the right of your move for each bar that is correct in both color and position, i.e., each bar that was correctly guessed. In addition, the computer responds with a grey bar if one of the colors you guessed is included in one of the five bars, but not in the position you guessed.

To enter a color guess for each bar, say the name of the color desired for that bar. The computer will change that bar to your color and automatically advance to the next bar. To change a previously set bar, use the words "advance" and "backspace" to position the dot under the desired bar, and then speak the desired color. When all bars have been set to the desired color, say the word "guess" to have the computer generate your score for that move.

SPEECHLAB

MASTERMIND

```
1    REM SET LOMEM: 5500 BEFORE RUNNING PROGRAM
2    FLASH = 1
5    SLOT=0: REM SPEECHLAB I/O SLOT
8    DIM K$(19),B(6)
10   DIM A(6),C(8),D(5),X(8),X$(8):X(1)=2:
     X(2)=12:X(3)=1:X(4)=13:X(5)=3:X(6)=9:
     X(7)=15:X(8)=5:X$="BGRYVOWX"
20   TEXT : CALL -936:PRINT "WELCOME TO THE
     GAME OF MASTERMIND! YOUR OBJECT IS TO
     GUESS 5 COLORS (WHICH"
30   PRINT "I WILL MAKE UP) IN THE MINIMUM
     NUMBER OF GUESSES. THERE ARE EIGHT
     DIFFERENT COLORS TO CHOOSE FROM."
40   PRINT "FEWER THAN 7 GUESSES--EXCELLENT":
     PRINT"   7 TO 9 GUESSES-----GOOD":
     PRINT"  10 TO 14 GUESSES----AVERAGE"
50   PRINT "MORE THAN 14 GUESSES--POOR":CALL -384:
     TAB 7:PRINT "HIT ANY KEY TO BEGIN"
52   CALL -380:IF PEEK (-16384)<132 THEN 52
60   PR#SLOT:PRINT:PR#0 :REM INIT SPEECHLAB
70   PRINT "PLEASE SAY THE FOLLOWING WORDS:"
71   FOR I=0 TO 1
72   PRINT "BLUE":PR#SLOT:PRINT"B":PR#0
74   PRINT "GREEN":PR#SLOT:PRINT"G":PR#0
76   PRINT "RED":PR#SLOT:PRINT"R":PR#0
78   PRINT "YELLOW":PR#SLOT:PRINT"Y":PR#0
80   PRINT "VIOLET":PR#SLOT:PRINT"V":PR#0
82   PRINT "ORANGE":PR#SLOT:PRINT"O":PR#0
84   PRINT "WHITE":PR#SLOT:PRINT"W":PR#0
86   PRINT "BLACK":PR#SLOT:PRINT"X":PR#0
88   PRINT "ADVANCE":PR#SLOT:PRINT">":PR#0
90   PRINT "BACKSPACE":PR#SLOT:PRINT"<":PR#0
92   PRINT "GUESS":PR#SLOT:PRINT"GUESS":PR#0
94   NEXT I
105  POKE -16368,0:GR:PRINT:FOR I=1 TO 8:
     C(I)=RND(8)+1:COLOR=X(I):HLIN I*4-2,
     I*4 AT 39:NEXT I
110  TRY=0:PRINT:PRINT "SAY COLORS FOR COLOR CHANGE"
115  PRINT "SAY ADVANCE OR BACKSPACE TO CHANGE BARS,
     GUESS TO ACCEPT GUESS"
200  Y=TRY*2 MOD 36+1:TRY=TRY+1:
     TAB 32:PRINT TRY;:COLOR=0:HLIN
     0,39 AT Y:FLASH=1:FOR N=1 TO 5:
     A(N)=8:GOSUB 1000:NEXT N:N=1
210  N=1 : GOSUB 1500
300  IN#8:INPUT K$:IF K$="" THEN 300:
     IN#0:PR#0
302  IF K$="GUESS" THEN 400
```

```
303    KEY=ASC(K$(1)):IF KEY<132 THEN 310:
       FOR I=1 TO 8: IF KEY<> ASC(X$(I)) THEN
       NEXT I
305    IF I=9 THEN 310: A(N)=I: KEY=190
310    GOSUB 1000: IF KEY=188 AND N>1 THEN N=N-1
312    IF KEY=190 AND N<5 THEN N=N+1
313    GOSUB 1500 : GOTO 300
400    COLOR=15: M=0: FOR I=1 TO 5:
       D(I)=C(I):J=I:GOSUB 2000: NEXT
       I: IF M=5 THEN 500:COLOR=5
       FOR J=1 TO 5: FOR I=1 TO 5: GOSUB 2000:
       NEXT I,J: GOTO 200
500    PRINT: PRINT "YOU GOT IT IN "; TRY;
       " TRIES(";: IF TRY <7 THEN PRINT
       "EXCELLENT";: IF TRY >6 AND TRY <10
       THEN PRINT "GOOD";
510    IF TRY >9 AND TRY <15 THEN PRINT
       "AVERAGE";: IF TRY >14 THEN PRINT
       "POOR";: PRINT ")"; CALL -384: TAB 5:
       PRINT "HIT ANY KEY TO PLAY AGAIN": GOTO 52
1000   IF N=6 THEN RETURN: COLOR=X(A(N)):
       HLIN N*4-2,N*4 AT Y: RETURN
1500   IF N=6 THEN RETURN: FOR I=1 TO 5:
       B(I)=0: NEXT I
1510   B(N)=15
1520   FOR I=1 TO 5: COLOR=B(I): HLIN I*4-1,I*4-1
       AT Y+2: NEXT I
1530   RETURN
2000   IF A(I) <> D(J) THEN RETURN:
       M=M+1: PLOT 21+M+M,Y: PRINT "";:
       A(I)=0: D(J)=9: RETURN
```

2.4 VOICE IDENTIFICATION

This demonstration takes training samples from a number of speakers. After the training mode, an unknown speaker is asked to speak three words, and Speechlab will guess the speaker.

VOICE ID DEMO

```
5     REM SET LOMEN: 5500 BEFORE RUNNING PROGRAM
10    DIM W$(25),N$(10),T1$(25),T2$(25),T3$(25)
15    DIM SP(3),ID(10)
20    N$ = "1234567890"
100   SLOT=3: REM SLOT NUMBER
105   PR#SLOT: PRINT: PR#0: CALL -936
110   PRINT "THIS PROGRAM USES SPEECH PATTERNS"
120   PRINT "TO TRY TO IDENTIFY AN UNKNOWN SPEAKER."
130   PRINT "EACH PERSON WILL BE ASKED TO SAY"
140   PRINT "A FEW WORDS TO GET TRAINING DATA."
150   PRINT "THEN THE SYSTEM WILL TRY TO DETERMINE"
160   PRINT "THE SPEAKER"
190   PRINT "HOW MANY SPEAKERS WILL THERE"
200   INPUT "BE  (MAX 10)", NS
210   IF NS>10 THEN GOTO 190
220   FOR I=1 TO NS
230   INPUT "PLEASE TYPE YOUR NAME ", W$
240   PRINT "SAY TOMATO"
250   PR#SLOT: PRINT I; W$: PR#0
260   PRINT "SAY MARIGOLD"
270   PR#SLOT: PRINT I; W$: PR#0
280   PRINT "SAY ELEPHANT"
290   PR#SLOT: PRINT I; W$: PR#0
310   NEXT I
315   CALL -936
320   PRINT "NOW WE WILL TRY TO IDENTIFY THE SPEAKER."
330   INPUT "TYPE 'GO' WHEN READY ", W$
335   CALL -936
340   PRINT "PLEASE SAY TOMATO"
350   IN#SLOT: INPUT T1$: IN#0: PR#0
365   IF T1$="" THEN 340
370   PRINT "PLEASE SAY MARIGOLD"
380   IN#SLOT: INPUT T2$: IN#0: PR#0
395   IF T2$="" THEN 370
400   PRINT "PLEASE SAY ELEPHANT"
405   IN#SLOT: INPUT T3$: IN#0: PR#0
415   IF T3$="" THEN 400
430   SP(1)=ASC(T1$(1,1))-ASC("0"): REM GET SPEAKER
440   SP(2)=ASC(T2$(1,1))-ASC("0")
445   SP(3)=ASC(T3$(1,1))-ASC("0")
450   FOR I=1 TO 3: ID(I)=0: NEXT I
460   FOR I=1 TO 3
470   ID(SP(I))=ID(SP(I))+1
480   NEXT I
490   GUESS = -1:TOT=0
500   FOR I=1 TO NS
510   IF ID(I)<TOT THEN GOTO 520
515   TOT = ID(I): GUESS=I
520   NEXT I
```

...continued...

VOICE ID DEMO (continued)

```
530     IF TOT=3 THEN PRINT "THE SPEAKER WAS...";
540     IF TOT=2 THEN PRINT "I THINK THE SPEAKER WAS...";
550     IF GUESS = ASC(T1$(1,1))-ASC("0") THEN
        PRINT T1$(2,LEN(T1$)): GOTO 330
560     IF GUESS = ASC(T2$(1,1))-ASC("0") THEN
        PRINT T2$(2,LEN(T2$)): GOTO 330
570     IF GUESS = ASC(T3$(1,1))-ASC("0") THEN
        PRINT T3$(2,LEN(T3$)):GOTO 330
580     GOTO 330
```

2.5 MOVING SPOT

This program illustrates Speechlab's ability to control objects by voice command. After training on the words up, down, left, right, the utterance of these words will move a spot around on the TV monitor by voice command. In the same manner you could control a robot through a maze.

MOVING SPOT DEMO

```
 90    REM SET LOMEM: 5500 BEFORE RUNNING PROGRAM
100    SLOT=3 : REM SLOT NUMBER
110    DIM W$(25)
120    GOSUB 9000 : REM INIT SPEECHLAB
125    CALL -936 : REM CLEAR SCREEN
130    PRINT "THIS DEMO ILLUSTRATES SPEECHLAB"
135    PRINT
140    PRINT "CONTROL OF OBJECTS.  YOU CAN MOVE AN"
145    PRINT
150    PRINT "OBJECT, A SPOT IN THIS CASE, BY VOICE"
155    PRINT
160    PRINT "COMMANDS.  FIRST WE TRAIN THE SYSTEM"
170    PRINT : PRINT
175    INPUT "TYPE 'GO' TO GET STARTED", W$
178    FOR I = 1 TO 5
180    W$ = "UP":PRINT "SAY:";W$;GOSUB 9100
190    W$ = "DOWN":PRINT "SAY: ";W$:GOSUB 9100
200    W$ = "LEFT":PRINT "SAY: ";W$:GOSUB 9100
210    W$ = "RIGHT":PRINT "SAY: ";W$:GOSUB 9100
215    NEXT I
217    CALL -936 : REM CLEAR SCREEN
220    PRINT : PRINT "THE SYSTEM IS NOW TRAINED."
225    PRINT
230    PRINT "TRY MOVING THE SPOT BY SAYING"
235    PRINT : PRINT "'UP','DOWN','LEFT',OR 'RIGHT'"
236    PRINT
237    INPUT "TYPE 'GO' TO GET STARTED", W$
240    GR
250    X=19:Y=19
255    X1=X:Y1=Y
260    GOSUB 2100:REM DISPLAY SPOT
270    X1=X:Y1=Y
280    GOSUB 9200:REM GET A WORD
290    IF W$ = "UP" THEN GOTO 340
300    IF W$ = "DOWN" THEN GOTO 350
310    IF W$ = "LEFT" THEN GOTO 360
320    IF W$ = "RIGHT" THEN GOTO 370
330    GOTO 270:REM DIDN'T RECOGNIZE WORD
340    Y=Y-3: GOTO 380
350    Y=Y+3: GOTO 380
360    X=X-3: GOTO 380
370    X=X+3: GOTO 380
380    IF Y>39 THEN Y=39
390    IF Y<0 THEN Y=0
400    IF X>39 THEN X=39
410    IF X<0 THEN X=0
420    GOSUB 2000:REM DELETE OLD SPOT
430    GOSUB 2100:REM DISPLAY NEW SPOT
440    GOTO 270
```

```
2000  COLOR=0: PLOT X1,Y1:RETURN
2100  COLOR=15: PLOT X,Y:RETURN
9000  REM INIT
9010  PR#SLOT:PRINT:PR#0
9020  RETURN
9100  REM TRAIN ROUTINE - ARG=W$
9110  PR#SLOT:PRINT W$: PR#0: RETURN
9200  REM RECOGNIZE ROUTINE
9210  IN#SLOT:INPUT W$:IF W$="" THEN 9210
9230  PR#0:IN#0:RETURN
```

2.6 BLACKJACK

This program plays blackjack. The cards are shuffled after each hand. The object of the game is to have the sum of your cards be as close to 21 as possible without going over. The game starts with your first two cards displayed and one of the computer cards. If you wish another card say "card". If you wish to keep the hand you have say "stand". After you say "stand" the computer will draw cards until it desires to stop. If you win a chip is added to your total, while if you lose a chip is subtracted from your total. Good luck!

SPEECHLAB BLACKJACK

```
5    REM SET LOMEM: 5500 BEFORE RUNNING PROGRAM
10   SLOT=3: REM SPEECHLAB SLOT
100  DIM W$(25): PLAYERCHIP=10
105  DIM DECK (52)
110  PR#SLOT: PRINT: PR#0
120  CALL -936
130  VTAB 2: TAB 6: PRINT "HEURISTICS SPEECHLAB
     BLACKJACK"
140  PRINT: PRINT "PLEASE SAY THE FOLLOWING WORDS..."
150  FOR I=1 TO 5
160  PRINT "CARD": PR#SLOT: PRINT "CARD": PR#0
170  GOSUB 7000
180  PRINT "STAND": PR#SLOT: PRINT "STAND": PR#0
190  GOSUB 7000
200  NEXT I
210  GOSUB 5000: REM INIT SCREEN
220  GOSUB 4000
230  GOSUB 3000: GOSUB 3000
240  VTAB 11: PRINT "SAY CARD OR STAND?"
250  IN#3: INPUT W$: PR#0: IN#0
260  IF W$="CARD" THEN 2000
270  IF W$="STAND" THEN 2500
280  GOTO 250
2000 GOSUB 3000: GOTO 240
2500 REM PLAYER STANDS
2510 GOSUB 4000
2520 IF DEALT>21 AND DEALACES=0 THEN 2560
2530 IF DEALT>16 AND DEALACES=0 THEN 2570
2540 IF DEALT>17 AND DEALACES>0 THEN 2570
2550 GOTO 2510
2560 PLAYERCHIP=PLAYERCHIP+1: VTAB 11:
     PRINT "I BUST, YOU WIN!!!   ": GOTO 210
2570 IF DEALT#PLAYERT THEN 2590
2580 VTAB 11: PRINT "PUSH!!!              ": GOTO 210
2590 IF DEALT>PLAYERT THEN 2610
2600 VTAB 11: PRINT "YOU WIN!!!             ":
     PLAYERCHIP=PLAYERCHIP+1: GOTO 210
2610 VTAB 11: PRINT "I WIN!!!             ":
     PLAYERCHIP=PLAYERCHIP-1
2620 IF PLAYERCHIP<1 THEN 12000
2630 GOTO 210
3000 REM DEAL PLAYER CARD
3010 CD=CD+1
3020 PLAYERPOS=PLAYERPOS+3
3030 VTAB 5: TAB PLAYERPOS: PRINT DECK (CD)
3040 IF DECK(CD)#1 THEN 3070
3050 PLAYERT=PLAYERT+10
3060 PLAYACES=PLAYACES+1
3070 PLAYERT=PLAYERT+DECK(CD)
```

```
3080    IF PLAYERT>21 AND PLAYACES>0 THEN 3150
3085    VTAB 6: TAB 8: PRINT PLAYERT
3090    IF PLAYERT>21 AND PLAYACES=0 THEN 3110
3100    RETURN
3110    VTAB 11: PRINT "YOU BUST!!!          "
3120    PLAYERCHIP=PLAYERCHIP-1
3130    IF PLAYERCHIP<1 THEN 12000
3140    POP: GOTO 210
3150    PLAYERT=PLAYERT-10: PLAYACES=PLAYACES-1:
        GOTO 3080
4000    REM DEAL COMPUTERS CARD
4010    DEALERPOS=DEALERPOS+3
4020    CD=CD+1
4030    VTAB 8: TAB DEALERPOS: PRINT DECK(CD)
4040    IF DECK(CD)#1 THEN 4065
4050    DEALT=DEALT+10
4060    DEALACES=DEALACES+1
4065    DEALT=DEALT+DECK(CD)
4070    IF DEALT>21 AND DEALACES>0 THEN 4100
4080    VTAB 9: TAB 8: PRINT DEALT
4085    GOSUB 7000
4090    RETURN
4100    DEALT=DEALT-10
4110    DEALACES=DEALACES-1
4120    GOTO 4070
5000    REM INIT SCREEN
5005    GOSUB 7000: GOSUB 7000
5010    GOSUB 6000
5020    DEALERPOS=11
5030    PLAYERPOS=9
5040    DEALACES=0: PLAYACES=0
5050    CD=1
5060    DEALT=0: PLAYERT=0
5070    CALL -936
5080    VTAB 2: TAB 6
5090    PRINT "HEURISTICS SPEECHLAB BLACKJACK"
5100    VTAB 5: PRINT "PLAYER:"
5110    VTAB 6: PRINT "TOTAL:": TAB 20: PRINT
         "CHIPS: "; PLAYERCHIP
5120    VTAB 8: PRINT "COMPUTER:"
5130    VTAB 9: PRINT "TOTAL:"
5140    RETURN
6000    REM SHUFFLE
6010    FOR I=1 TO 52
6020    C=(I MOD 13)+1
6030    IF C>10 THEN C=10
6040    DECK(I)=C
6050    NEXT I
6060    FOR I=1 TO 52
6070    DECK(I)=DECK(RND(52)+1)
6090    RETURN
7000    FOR X=0 TO 500: NEXT X: RETURN
12000   END
```

3.0 WRITING PROGRAMS FOR SPEECHLAB

The following section describes how to write Apple II basic programs which use Speechlab. The method of selecting the Speechlab board for input is discussed, and the three modes of operation (initialization, training, and recognition) are described.

3.1 SELECTING THE SPEECHLAB BOARD FOR INPUT

Following the usual conventions for Apple II peripherals Speechlab is selected for input, or recognition, and selected for output, or passing parameters to Speechlab, via the IN# and PR# Basic statements. For example, IN# 3 will direct the Apple II computer to accept all future data inputs from the peripheral in Slot 3, rather than the keyboard. If the Speechlab board has been placed in peripheral Slot 3 of the computer, then any program input will be received from the Speechlab board, rather than the keyboard.

Likewise, a PR# 3 statement will direct all program output to the peripheral in peripheral Slot 3, rather than the video display. Thus Speechlab receives its commands from the Basic program via a PR# statement in the Basic program, which "prints" command outputs to the Speechlab board. The following three modes of operation are selected by using the PR# and IN# sharp statements.

3.2 INITIALIZATION

Speechlab must be initialized before it can be used.

This initialization procedure clears out the Speechlab data tables and prepares Speechlab for subsequent use. Initialization is accomplished by selecting the board for output with a PR# statement followed by a Basic print statement with no parameters. For example, if the Speechlab board were installed in peripheral Slot 3, the following Basic statements executed in a Basic program would initialize the Speechlab board:

100 PR# 3	selects peripheral Slot 3 for output
110 PRINT	initializes the board

The statement on line 100 selects the Speechlab board for output, and the statement on line 110, a solitary print statement with nothing to print, accomplishes the initialization function.

3.3 TRAINING

To train the Speechlab board on a word or phrase, first select the board for output with a PR# statement, and then print the word to Speechlab as a string. For example, to train the system on the word "Apple", assuming Speechlab is in peripheral Slot 3, the following Basic program statements suffice:

200 PR# 3	directs output to Speechlab
210 PRINT "APPLE"	prints the string "APPLE" to Speechlab

The statement on line 200 directs program output to Speechlab rather than the television monitor, and the statement on line 210 prints the string "APPLE" to Speechlab. Speechlab will then associate the next word spoken into Speechlab with the string "APPLE".

This word must be less than 1¼ seconds in duration. In practice, the user will probably first want to print the required string to the screen to remind him which word to speak next, and then print the string to Speechlab, followed by speaking the word into the Speechlab microphone. The following program accomplishes this task:

300 PR# 0	select the TV monitor for output
310 PRINT "APPLE"	prints "APPLE" to the TV screen to prompt the user
320 PR# 3	selects the Speechlab in Slot 3 for output
330 PRINT "APPLE"	tells Speechlab to listen for "APPLE"

3.4 RECOGNITION

After the Speechlab has been trained on the desired vocabulary, it is ready to recognize spoken words. To recognize spoken words, first select the Speechlab peripheral slot for input and then perform an input command. For example, if Speechlab is in peripheral Slot 3, the following program will recognize a spoken word and return the string corresponding to the spoken word to the Basic program.

400 IN# 3	select Speechlab for input
410 INPUT A$	recognizes the next spoken word and returns the string corresponding to the word in A$.

3.5 ERRORS

Severly errors can occur while using Speechlab. As previously mentioned, the longest word or phrase that can be entered as a single vocabulary word cannot be longer than 1¼ seconds in duration. If Speechlab detects that the word has not ended 1¼ seconds after it began, it will beep the Apple II beeper, and start listening again for a word, at which time the word may be repeated. If Speechlab is used in an area with high background noise level, the Speechlab will hear continuous noise, or sounds over 1¼ seconds in duration, and will, therefore, beep every 1¼ seconds, listen some more, beep again, etc.

If Speechlab hears a word that it does not recognize, it will return an empty string, that is, a string with no characters to the Basic program.

Finally, if the user attempts to train the system on more than the maximum of 32 words, the system will return a Basic range error.

3.6 SETTING LOMEM

The speech recognition software routines need an area of RAM memory to store data about the words in its vocabulary. This RAM area is generated by moving up LOMEM, the address of the low end of Basic program memory. HENCE UPON ENTERING BASIC THE USER MUST TYPE LOMEM:5500 BEFORE RUNNING A PROGRAM WHICH USES SPEECHLAB.

3.7 GENERAL USAGE COMMENTS

In using the Speechlab board as described previously, two comments should be noted. The first is that when input is directed to come from Speechlab instead of the keyboard, all further input will come from Speechlab until the program directs otherwise. Thus, if data is to be alternately entered from Speechlab and the keyboard, the input will be commanded from Speechlab via an IN# statement, and then must be commanded from the keyboard with a subsequent IN# 0 statement, which will redirect the input to come from the keyboard.

The second comment is that the Speechlab will not listen for a word until commanded from the Basic program via an INPUT statement. Thus the input statement must first be executed, after which Speechlab will "listen" until it hears a word. Words spoken before an INPUT statement is executed will be ignored.

4.0 SPEECH RECOGNITION ALGORITHM TECHNICAL DESCRIPTION

The speech recognition software is the software contained on the on-board memory on the Speechlab hardware board. The following section describes the technical operation of this software in some detail for those interested. First is a description of the type of information made available to the software algorithm from the hardware preprocessor board. Next is a description of the method of finding the beginning and ending points of a speech utterance. This is followed by a description of the time normalization process, which compensates for different duration utterances of the same word. A description of the final parameters stored away to characterize a word and the method of matching unknown words to entries in the vocabulary table is discussed last.

4.1 HARDWARE PREPROCESSOR INFORMATION

The hardware board extracts and measures two quantities associated with an incoming speech waveform. The board splits the incoming frequency spectrum into two components, one from 100 Hz to 900 Hz and the other from 900 Hz to 5000 Hz. These are the frequency ranges of the first two resonances of the human vocal tract. Each band has a zero crossing detector which changes state every time the waveform from its filter crosses through its average level. This will occur at the rate of the dominant frequency of the filtered waveform, which in turn depends upon the resonant frequency of the vocal tract at that particular instant. Thus the rate of state changes of the zero crossing detectors is proportional to the resonant frequency of the vocal tract in the frequency range of the filter. Since there are two filters and two zero crossing detectors, the board measures the frequency of the first two resonances, or formants. These zero crossing detector outputs are made available to the computer, which counts the number of changes during a given interval of time to determine the frequency. Since these frequencies depend on the shape of the human vocal tract, which varies relatively slowly with time, these quantities are extracted every 12.5 milliseconds, or about 80 times per second.

Amplitude information is also determined by the hardware preprocessor in each of the two frequency bands. This information is only quantized to two bits of information, and is used primarily to determine beginning and ending points of speech utterances. This information is also useful for distinguishing unvoiced speech from voiced speech since unvoiced speech tends to be high frequency/low amplitude in nature, while voiced speech is lower frequency with greater amplitude. Non-voiced speech examples are the consonants "s" and "f", while voiced speech examples would be any of the vowels.

4.2 FINDING THE BEGINNING AND ENDING OF WORDS

The software uses two methods for determining the beginning points and endpoints of words. One method relies on amplitude information and the other method relies on frequency information. Words that begin with voiced sounds, such as vowels, are characterized by rapid increases in waveform energy at the beginning of the word. The software checks the amplitude of the information coming into the microphone, and if this amplitude exceeds a preset level, the program determines that the word has begun. Alternatively, words which begin with unvoiced sounds typically have low energy, but the frequency component of the waveform rises rapidly. In this case, the program checks the frequency of the information coming into the microphone, and when the frequency rises rapidly, even in the absence of any detectable amplitude, the program determines that a word has begun. By performing both of these tests simultaneously, the program can determine when an unknown utterance has started. To avoid false word starts generated by short noise pulses, either of these two conditions must be present for at least 100 milliseconds to indicate a valid word beginning. If these conditions are satisfied for less than this period of time, the program starts from the beginning and continues to look for a valid word start. After a word has begun, the program determines that the word has ended when neither of the above conditions has been met for at least 100 milliseconds. The 100 millisecond ending criterion is necessary because many words actually have periods of silence in the middle of the utterance, for example the words "six" and "eight". These periods of silence are typically 80 to 90 milliseconds long and therefore will not meet the criterion for the end of a word.

4.3 TIME NORMALIZATION

Once the beginning and endpoints of the speech utterance have been found the algorithm time normalizes the word. That is, it makes each word appear to have the same duration in time. This compensates for speakers who tend to say the same word slowly on one occasion and more rapidly on another. This is accomplished by dividing the word into 16 evenly spaced intervals and sampling the two frequency quantities and the two amplitude quantities at these sixteen points. Thus if one utterance of a particular word is twice as long as another utterance of the same word, the sampling interval for the longer word will be exactly twice as long as that for the shorter word, and exactly the same data samples will result from each word. After the normalization process is complete, each of the sixteen data points is characterized by 4 bytes of information, two bytes for the frequency in the two filter frequency ranges, and two bytes for the amplitude of each of the two frequency bands. Thus an entire word is characterized by 64 bytes of information which results from four bytes taken at each of sixteen data points. These 64 bytes are stored in a vocabulary table with an associated string to be returned to the program when that word is recognized. These 64 bytes are sometimes called a template for the word.

4.4 MATCHING UNKNOWN WORDS TO VOCABULARY WORDS

The template extraction procedures described above to generate the vocabulary reference templates during the training mode are also used to generate a template of an unknown word entered to the system during recognition mode. The problem is to determine which of the reference templates is the closest match to the unknown template, a process known as template matching. A simple but effective method of template matching is to take the difference between each byte of the unknown template and the corresponding byte of a vocabulary table template, take the absolute value of each difference, and sum the result. This sum of absolute value of the differences results in a single number expressing the "distance" between the unknown template and that vocabulary word. By repeating this differencing process between the unknown template and each of the entries in the vocabulary table a list of up to 32 numbers is obtained, and the vocabulary table entry with the smallest difference is the best candidate for being the correct word. This smallest difference is compared to a maximum allowable difference. If the closest match exceeds this maximum allowable difference, the system assumes that the word which was spoken is not in the system vocabulary table and returns the null, or empty, string. If the closest match is less than this difference, then the program returns to the Basic program with the string corresponding to that vocabulary table entry, entered into the system during training. The above algorithm is sufficiently simple to be executed in real time on a microprocessor and at the same time is effective in solving the speech recognition problem.

5.0 HARDWARE THEORY OF OPERATION

This section describes the hardware preprocessor theory of operation. Refer to the block diagram Figure 3 and the schematic diagram in Appendix C. The Speechlab Model 20 hardware consists of a microphone amplifier with frequency compensation, two analog bandpass filters, each with a signal rectification/averaging stage, and a zero crossing detector, a 1K byte ROM, and logic to run the input port and the ROM memory interface.

5.1 MICROPHONE PREAMP

The first element in the signal chain is the microphone, which produces an analog signal on the order of 10 millivolts for normal speech. The first stage of the microphone preamp "pre-emphasizes" signals up to 3.1 KHz at a rate of 6 db per octave. That is, high frequency signals are amplified more than low frequency signals, with the rate of gain increase being a factor of two for each doubling in frequency up to 3.1 KHz. From 3.1 KHz to 6.6 KHz the gain of the amplifier is constant, and the gain drops off beyond 6.6 KHz at the rate of 6 db per octave. This pre-emphasis compensates for the characteristics of the human vocal tract, which produces less energy in the higher audio frequency ranges than in the lower frequencies. The pre-emphasis tends to make all frequency components have the same order of magnitude. The output of the first stage feeds the second stage, which simply amplifies the first stage output by a factor of 60. The microphone preamp output is biased such that the output rests at about 5.5 volts with no signal into the preamp, and swings about this reference voltage by $\pm$ 5.5 volts utilizing the full dynamic range of 0 to 11 volts.

5.2 FILTERS

The output of the microphone preamp feeds two audio bandpass filters. These filters amplify and extract that portion of the speech utterance which falls within the frequency bandpass of the filter. The first filter has a frequency bandpass range of 150 Hz to 900 Hz and the second filter ranges from 900 Hz to 5 KHz. The lower frequency band corresponds to the frequency range of the lowest frequency resonance (formant) of the human vocal tract, and the higher frequency band corresponds to the frequency band of the next two vocal tract resonances. The energy and frequency in each of these frequency ranges is determined by the shape of the vocal tract which, of course, varies with time to produce time-varying outputs from these filters during a speech utterance.

The bandpass filters are each implemented as a cascade of a second degree multiple feedback lowpass filter followed by a second degree multiple feedback highpass filter to yield attenuations of 40 db/decade (12 db/octave) above and below the pass bands. The outputs of each filter are biased to the reference level of 5.5 volts. That is, in the absence of any signal, the output will rest at 5.5 volts, and in the presence of a signal will swing above and below that level. Each filter output produces sinewaves proportional in amplitude and equal in frequency to the output of the microphone signal that falls within the frequency bandpass of the filter. The output of each filter is fed to a zero crossing detector and a rectifier/averager.

5.3 ZERO CROSSING DETECTOR

The zero crossing detectors provide a means of measuring the dominant frequency component of the signal passing through each of the filters. The zero crossing detector is a comparator with one input connected to the reference voltage and the other input connected to the filter output. Since the filter output is equal to the reference voltage (5.5V) when no signal is input to the microphone, the two comparator inputs are equal when no signal is present. When a signal is present, however, the output of the filter will swing above and below the reference voltage. When the filter output is above the reference voltage, the comparator will be forced to a logic zero output. When the filter output is below the reference voltage, the comparator is forced to a logic one state. Every time the filter output crosses the reference voltage the comparator changes state. The output of each comparator is connected to the parallel output port, where they may be input to the computer. The computer reads the zero crossing detectors and determines the number of times the comparator changes state, from one to zero or vice versa, in a given period of time. This number is equal to the number of times the filter output crosses the reference voltage level. If the input to the filter were a single sinewave, the number of state changes in one second would be exactly twice the frequency, since a sinewave crosses the reference voltage twice in one period. Speech waveforms typically have several frequencies present with one frequency dominant. The dominant frequency in a frequency range is equal to the resonant frequency of the vocal tract in that frequency range. The other frequencies are attenuated by the vocal tract. The addition of small amplitude waves on the dominant wave causes sinusoidal ripples in the dominant wave. These ripples sometimes occur as the dominant wave passes through the reference voltage level at the output of the filter, producing additional zero crossings compared to the number produced by the dominant wave alone, but the difference is small and of little consequence. The output of the filter is also fed to a rectified/averager stage.

5.4 RECTIFIER/AVERAGER

The rectifier/averager circuit produces an output proprotional to the amplitude of the sinewaves coming from its associated filter. This is accomplished by rectifying the filter output relative to the reference voltage and time averaging the result. The rectification is implemented with a diode, which only passes current to the averager when the filter output is above the reference voltage. The time averager is a low pass filter ("leaky memory") with a bandwidth of 20 Hz. In the absence of new signals from the filter, the averager will decay to its no signal value, the reference voltage, with a time constant of 50 milliseconds. Thus the averager "forgets" events which occurred 50 milliseconds in the past. The output of the averager is a voltage proportional to the energy of the waveform from its associated filter. The output of the rectifier/averager is fed to the amplitude detection hardware.

5.5 AMPLITUDE DETECTION

Amplitude detection is simply accomplished by comparing the output of the rectifier/averager stage with three fixed voltages using three voltage comparators. One of the comparator reference voltages is set very close to the no-signal output of the rectifier/averager to determine the beginning and ending of words. The other two reference voltages are set at about one third and two thirds of the dynamic range of the rectifier/averager outputs. These comparator outputs, three for each of the two frequency bands, are routed to the parallel input port where they may be input to the computer.

5.6 ROM ENABLE/DISABLE

The ROM enable/disable circuitry operates in accordance with the rules for Apple II computer peripherals with on-board ROM. (See Apple II documentation.) The I/O select signal from board connector pin 1 enables the on-board ROM whenever the processor executes a memory read instruction to read from the 256 byte block of memory set aside for each peripheral card. This address block is CN00 to CNFF where N is the peripheral connector slot number, 1 to 7. Once a memory read from this address block has occurred, an RS flip-flop is set to enable mode, which allows the on-board 1K byte ROM to occupy half of the full 2K byte address space from C800H to CFFFH set aside for any peripheral. This range is detected via the I/O strobe signal which decodes this address range on the Apple II mainframe. The ROM will remain enabled until a memory access is made to the address range CF80H to CFFFH, or until a reset occurs, whichever happens first.

Heuristics
INC

900 N. SAN ANTONIO ROAD
LOS ALTOS, CA 94022
(415) 948-2542

LIMITED WARRANTY

HEURISTICS, INC., in recognition of its responsibility to provide quality components and adequate instruction for their proper assembly, warrants its products as follows:

All components sold by HEURISTICS, INC. are purchased through normal factory distribution and any part which fails because of defects in workmanship or material will be replaced at no charge for a period of one year for assembled modules, following the date of purchase. The defective part must be returned postpaid to HEURISTICS, INC. within the warranty period.

Any modules purchased from HEURISTICS, INC. as assembled units are guaranteed to meet specifications in effect at the time of manufacture for a period of at least one year following purchase. These modules are additionally guaranteed against defects in materials or workmanship for the same one year period. All warranted factory assembled units returned to HEURISTICS, INC. postpaid will be repaired and returned without charge.

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

WARRANTY REGISTRATION CARD

Name____

Street Address____

City, State, Zip____

Date Purchased____

Purchased From____

Where did you hear of Speechlab?

Magazine (which)____

Word of Mouth____

Store Display____

Note: When registration card is completed, place in envelope and mail to Heuristics Inc., 900 N. San Antonio Rd., Los Altos, CA 94022.

PARTS LIST MODEL 20A

Integrated Circuits:

IC1	LM324
IC2	LM339
IC3	8097
IC4	2708P
IC5	74LS00
IC6	LM324
IC7	LM339
IC8	8097
IC9	8093
IC10	74LS20
IC11	74LS00

Diodes:

D1	1N4148
D2	1N4148
D3	1N4148
D4	1N752

Capacitors:

1.	.01 UF
2.	100 PF
3.	.1 UF DISC
4.	.1 UF DISC
5.	240 PF
6.	.0047 UF
7.	.01 UF
8.	.022 UF
9.	.01 UF
10.	.1 UF DISC
11.	.1 UF DISC
12.	.01 UF
13.	.01 UF
14.	240 PF
15.	.0022 UF
16.	.0022 UF
17.	.001 UF
18.	.01 UF
19.	.1 UF DISC
20.	.1 UF DISC
21.	.1 UF DISC
22.	1 UF
23.	.1 UF DISC
24.	160 PF

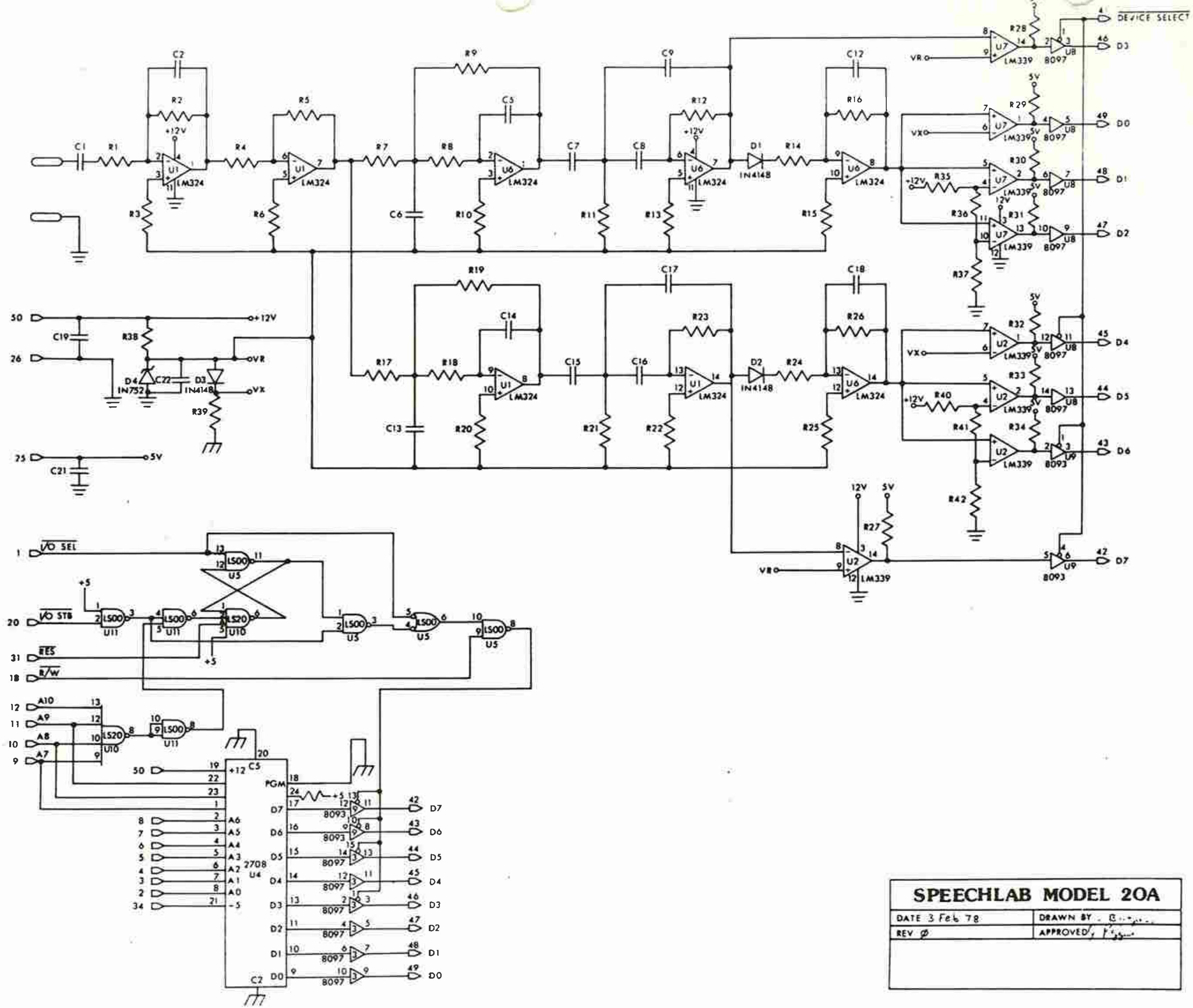
PARTS LIST MODEL 20A (continued)

Resistors:

1. 5.1K
2. 510K
3. 510K
4. 5.1K
5. 300K
6. 5.1K
7. 360K
8. 75K
9. 360K
10. 240K
11. 18K
12. 300K
13. 300K
14. 30K
15. 30K
16. 750K
17. 130K
18. 3.3K
19. 130K
20. 3.3K
21. 33K
22. 430K
23. 430K
24. 30K
25. 30K
26. 750K
27. 10K
28. 10K
29. 10K
30. 10K
31. 10K
32. 10K
33. 10K
34. 10K
35. 62K
36. 16K
37. 16K
38. 1K
39. 10K
40. 62K
41. 10K
42. 27K

Miscellaneous:

P.C. Board
Mike
Mike Connector
14 Pin Socket
24 Pin Socket
16 Pin Socket
Mod. 20A Manual



SPEECHLAB MODEL 20A	
DATE 3 Feb 78	DRAWN BY
REV Ø	APPROVED