Published in Electronics and Computing Monthly october 1983
Mike Furminger of Nen College

SPEECH RECOGNITION SYSTEMS

Mike Furminger, of Nene College, Northampton, outlines the basic principles of a micro based speech recognition unit.

- The idea of talking to a computer then getting an answer in plain English has long been a popular one in science fiction. The reality of talking to a computer though is not quite as easy as having a conversation with another person. Speech Recognition is being used in modern helicopters and commercial planes, however, and enables pilots to call up desired displays on a VDU rather than looking for an instrument amongst many required by the aircraft.

- Alternative Approaches

- Computers can understand certain words when an operator has 'taught' the computer the word before or, can analyse each syllable of the sound and try to create a reasonable match to the that word against a stored library of sound. It is possible to buy speech recognition hoards which will recognise up to 100 words with 99% reliability, however these are expensive and unsuitable for home use.
- To make a small home computer recognise words is a little tricky, but reasonable results can be obtained if the computer is not expected to discriminate between similar words spoken by the same person. In order to appreciate how machines can understand the spoken word it's necessary to examine the basics of human speech.
- In principle spoken words are made up of a series of basic elements. There are 'Voiced' and 'Unvoiced' sounds, ie. sound which uses the mouth to form the noise and sound which is just blown from the lungs. An example of unvoiced sound is the word 'Oh'. The voiced sounds are very complex but simply can be identified as one of three major groups ie. fricative, sibilant and plosive sound. Fricative sound uses the tongue to make the sound as in 'fur', sibilant sound is the 's' sound and plosive sound is a sudden release of air by the mouth as in the word 'pop'. Although the above is not a complete analysis of spoken sound it is adequate for an introduction.

- Analysis Techniques

- To analyse these different sounds so that a computer can recognise different words it's necessary to use some technique which breaks down the complex wave form into a manageable spectrum. If we had a large mainframe computer to hand then we could try, for example the linear predictive coding (LPC) technique used by the Texas company's speech chips. In principle LPC creates a series of speech frames (eg 40 per second) in which the frequency, amplitude, duration and sound type is converted to a binary code. This binary code is then saved in a relatively small memory. Recognition would occur if an incoming speech pattern matched a known one.
- The techniques of LPC are beyond the scope of a small microcomputer but simpler approaches to the problem can be used A spectrum analyser, such as used to control disco lights, would be a suitable method The spectrum analyser breaks down a complex spoken word or musical note into its frequency elements. This is similar to the way in which a prism breaks white light into different colours but instead a sound is broken down to elements like notes on a piano keyboard.

- Spectrum Analysers

- Many designs for spectrum analysers have bee&publishedby various magarines These are either based on a series op-amp filters or use a switched capacitor filter such as the MF10. Either approach will work well enough, so it a spectrum analyser is already to hand then this is another use for it.

Figure 1: Block diagram of a Spectrum Analyser. After amplification, the speech signal is divided into four frequency bands

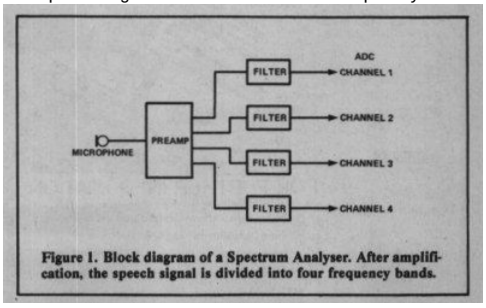


Figure 1. Block diagram of a Spectrum Analyser. After amplification, the speech signal is divided into four frequency bands.

- If you have not got a spectrum analyser then building a basic model is not too difficult With a BBC model B computer a very simple device could be built using filters into each of the 4 ADC channels. Suitable filter frequencies are mentioned later. All other computers will require a multiplexed ADC to accept the signal data.

Figure 2: Each of the block diagram's filter blocks could take the form of an active filter based around an op-amp

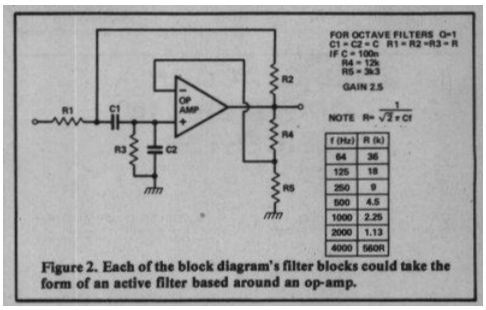


Figure 2. Each of the block diagram's filter blocks could take the form of an active filter based around an op-amp.

- Another approach is to use an MF10 controlled with various clock rates to reproduce many filters in one as shown in Fig. 3.

Figure 3: An alternative to an op-amp filter
this filter is built around a Switched Capacitor block

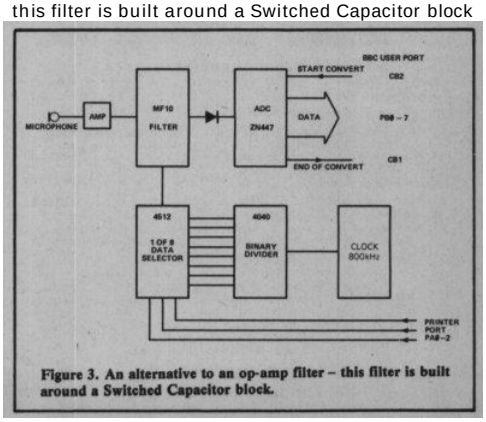


Figure 3. An alternative to an op-amp filter – this filter is built around a Switched Capacitor block.

Figure 4: Circuit details of the filter of the block diagram of Fig. 3

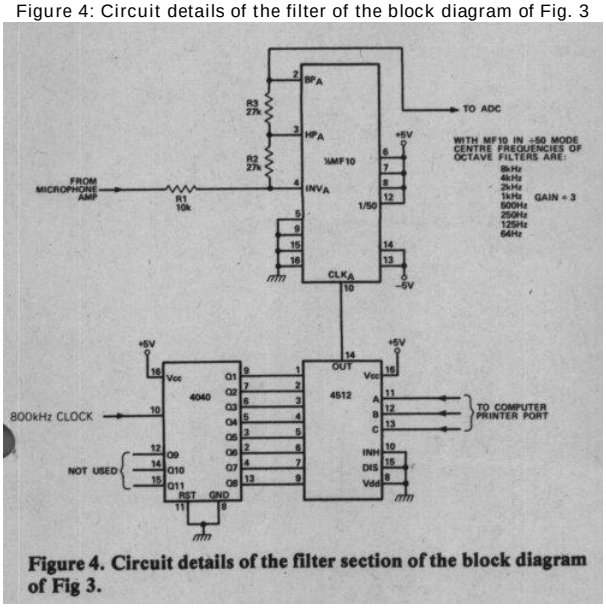


Figure 4. Circuit details of the filter section of the block diagram of Fig 3.

- Sound Experiments

- The author used a PET computer fitted with an Eventide 30 1/3 octave filter spectrum analyser and consequently all examples are quoted for a PET computer, the method described though can be transferred to any other machine. The properties of speed will dictate which filters are significant
- Unvoiced sound tends to have a broad spectrum with a maximum around 500 Hz, the exact frequency depending on the physical size of the speaker. Voiced sounds can be identified by their properties, since sibilant sound is a high frequency hiss around8 KHz, plosives are low frequency sounds around 150 Hz and fricative sound tends to be more difficult to identify but mostly its components are around 1 KHz. There is no useful information to be found below 50 KHz or above 10 KHz. If only a few channels are used then it is a good idea to keep the above points in mind.
- A suitable 8 channel octave set of filters would have centre frequencies at 62, 125,250,500, 1000,2000,4000 and 8000 Hz.

- Software Outlines

- The software for speech recognition on a microcomputer would first store a copy of a word, then compare any incoming word with the stored copy. The more comparisons that are required the slower will be the response. The program elements for saving a copy of a word are shown in fig. 5a An example of these procedures on the PET computer are given in program 1, lines 330 to 570. The program fills an array with the stored data from the spectrum analyser, sorts for peaks and then sorts again into channel order with a bubble sort.

Figure 5a (above) shows the program elements for saving a copy of a word.
Fig 5b (right) shows the flow diagram of a Spectrum Analyser based on an MF10 Filter

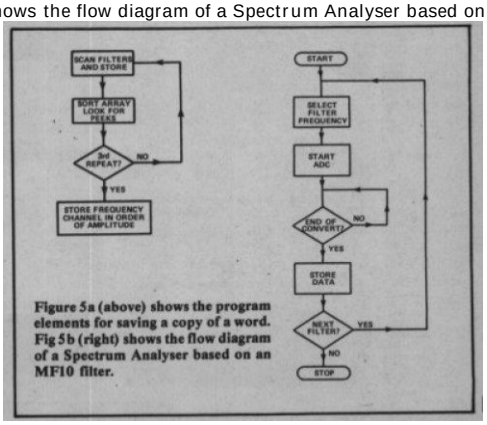


Figure 5a (above) shows the program elements for saving a copy of a word. Fig 5b (right) shows the flow diagram of a Spectrum Analyser based on an MF10 filter.

Program 1: This illustrates the way in which words can be saved as in the outline of Fig 5a.

Program 1. This illustrates the way in which words can be saved as in the outline of Fig 5a.

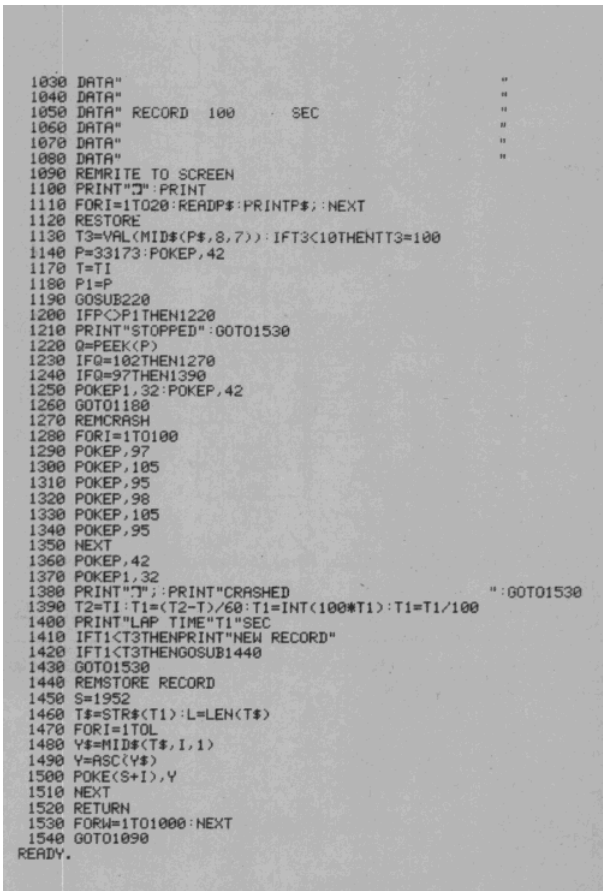
```
100 PRINT"□■■■■■■■HAVE YOU TURNED ON THE"
110 PRINT"■■REAL TIME SPECTRUM ANALYZER■?"
120 GETA$:IFA$<>"Y"THEN120
130 DIMA(30),B(30),C(30),D(30),E(30),F(10)
140 SYS4096*11
150 PRINT"□■■THIS PROGRAM TAKES THREE COPIES OF"
160 PRINT"ONE WORD AND SAVES IT ON DISC"
170 INPUT"■■■■■ENTER WORD AND PRESS ■RETURN■";W$
180 FORM=1TO3
190 POKE46080,1:REM GAIN MAX
200 A=USR(5):POKE931,255:A=USR(3):REM CLEAR
210 A=USR(7):A=USR(8)
220 A=USR(4)
230 POKE931,0:REM FREEZE
240 PRINT"□■■■■■■HEN YOU ARE READY FOR TEST ";M;" PRESS ■SPACE■ AND SAY ■";W$
250 GETA$:IFA$<>CHR$(32)THEN250
260 FORI=0TO300:NEXT
270 A=USR(1)
280 FORX=1TO50
290 A=USR(3):REM SCAN FILTERS
300 A=USR(2)
310 NEXT
320 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■SORTING DATA"

READY.
READY.

330 FORF=0TO29:REM FILL ARRAY
340 A(F)=PEEK(F+27+F)
350 NEXT
360 KK=3:REM FIND 4 PEAKS FROM 50HZ
370 FORII=0TO12
380 KK=KK+1
390 IFKK>28THEN480
400 IFA(KK)>A(KK+1)THEN420
410 IFA(KK)<=A(KK+1)THEN380
420 B(II)=KK+1:C(II)=A(KK)
430 KK=KK+1
440 IFKK>28THEN480
450 IFA(KK)<=A(KK+1)THEN470
460 GOTO430
470 NEXTII
480 FORI=0TO12:SW=0:REM BUBBLE SORT
490 FORJ=0TO12
500 IFC(J)>=C(J+1)THEN550
510 T=C(J):C(J)=C(J+1):C(J+1)=T:SW=1
520 T=B(J):B(J)=B(J+1):B(J+1)=T
550 NEXTJ
560 IFSW=0THENI=12
570 NEXTI
575 IFM=1THENFORI=0TO12:D(I)=B(I):NEXT
576 IFM=2THENFORI=0TO12:E(I)=B(I):NEXT
580 NEXTM
590 FORS=0TO9:REM COMPARE
600 F(S)=B(S)
610 IFD(S)=E(S)THENF(S)=D(S)
620 IFF(S)=0THENF(S)=B(S)
630 NEXTS
640 FL$="@0:"+W$+",SEQ,WRITE"
650 OPEN1,8,15:OPEN2,8,14:CLOSE2
660 OPEN2,8,14,FL$
670 FORW=0TO9:PRINT#2,F(W);CHR$(13);:NEXT
680 CLOSE2:CLOSE1
690 PRINT"□■■ANOTHER WORD?"
700 GETA$:IFA$="Y"THEN170
710 IFA$="N"THENEND
720 GOTO700
READY.
```

- It is a good idea to loop round the recording program 3 times and take the hest match. This ensures any odd noises are removed, as lines 575-630.

Program 2: This illustrates the way in which a computer
could perform a simple 'sex test' based on the
fact that males have peak vocal frequencies
of less than 500 Hz while females have higher dominant frequencies

Program 2. This illustrates the way in which a computer could perform a simple 'sex test' based on the fact that males have peak vocal frequencies of less than 500 Hz while females have higher dominant frequencies.

```
100 PRINT"[illegible]HAVE YOU TURNED ON THE"
110 PRINT"[illegible]REAL TIME SPECTRUM ANALYZER[illegible]?"
120 GETA$:IFA$<>"Y"THEN120
125 PRINT"[illegible]COLLECTING DATA"
130 DIMA(30),B(30),C(30)
140 SYS4096*11
150 K=1:W$="UP":GOSUB700
160 K=2:W$="DOWN":GOSUB700
170 K=3:W$="LEFT":GOSUB700
180 K=4:W$="RIGHT":GOSUB700
190 PRINT"[illegible]THE COMPUTER IS NOW READY TO PLAY"
210 FORI=1TO2000:NEXT:GOTO1100
220 POKE46080,1:REM GAIN MAX
230 A=USR(5):POKE931,255:A=USR(3):REM CLEAR
240 A=USR(7):A=USR(8)
250 A=USR(4)
260 POKE931,0:REM FREEZE
280 PRINT"[illegible]YES?"
285 A=USR(3):IF PEEK(826)<50THEN285
290 FORK=1TO100
300 A=USR(3)
310 NEXT
320 PRINT"[illegible]     "
330 FORF=0TO29STEP2
340 A(F)=PEEK(827+F)+PEEK(828+F)+PEEK(829+F)
350 NEXT
360 KK=0:REM FIND PEEKS FROM 50HZ
370 FORII=0TO12
380 KK=KK+1
390 IFKK>28THEN480
400 IFA(KK)>A(KK+1)THEN420
410 IFA(KK)<=A(KK+1)THEN380
420 B(II)=KK+1:C(II)=A(KK)
430 KK=KK+1
440 IFKK>28THEN480
450 IFA(KK)<=A(KK+1)THEN470
460 GOTO430
470 NEXTII
480 FORI=0TO12:SW=0:REM BUBBLE SORT
490 FORJ=0TO12
500 IFC(J)>=C(J+1)THEN530
510 T=C(J):C(J)=C(J+1):C(J+1)=T:SW=1
520 T=B(J):B(J)=B(J+1):B(J+1)=T
530 NEXTJ
540 IFSW=0THENI=12
550 NEXTI
570 R0=0:R1=0:R2=0:R3=0
580 FORI=0TO9
590 IFH0(I)=B(I)THENR0=R0+1
600 IFH1(I)=B(I)THENR1=R1+1
610 IFH2(I)=B(I)THENR2=R2+1
620 IFH3(I)=B(I)THENR3=R3+1
630 NEXTI
640 IFR0>=R1ANDR0>=R2ANDR0>=R3THENP=P+40:RETURN
650 IFR1>=R0ANDR1>=R2ANDR1>=R3THENP=P-40:RETURN
660 IFR2>=R0ANDR2>=R1ANDR2>=R3THENP=P-1:RETURN
670 IFR3>=R0ANDR3>=R1ANDR3>=R2THENP=P+1:RETURN
680 GOTO220
690 END
700 REM COLLECT FILES
710 OPEN1,8,15
720 FL$="0:"+W$+",SEQ,READ"
730 OPEN3,8,13,FL$
740 FORW=0TO9
750 INPUT#3,H(W)
760 ONKGOSUB810,820,830,840
770 NEXTW
780 CLOSE3
790 CLOSE1
800 RETURN
810 H0(W)=H(W):RETURN
820 H1(W)=H(W):RETURN
830 H2(W)=H(W):RETURN
840 H3(W)=H(W):RETURN
850 REM GAME
860 DATA"[illegible]"
870 DATA"[illegible]"
880 DATA"[illegible]"
890 DATA"[illegible]"
900 DATA"[illegible]"
910 DATA"[illegible]"
920 DATA"[illegible]"
930 DATA"[illegible]"
940 DATA"[illegible]"
950 DATA"[illegible]"
960 DATA"[illegible]"
970 DATA"[illegible]"
980 DATA"[illegible]"
990 DATA"[illegible]"
1000 DATA"[illegible]"
1010 DATA"[illegible]"
1020 DATA"[illegible]"
```



```
1030 DATA"                                          "
1040 DATA"                                          "
1050 DATA" RECORD  100      SEC                     "
1060 DATA"                                          "
1070 DATA"                                          "
1080 DATA"                                          "
1090 REMRITE TO SCREEN
1100 PRINT"□":PRINT
1110 FORI=1TO20:READP$:PRINTP$;:NEXT
1120 RESTORE
1130 T3=VAL(MID$(P$,8,7)):IFT3<10THENTT3=100
1140 P=33173:POKEP,42
1170 T=TI
1180 P1=P
1190 GOSUB220
1200 IFP<>P1THEN1220
1210 PRINT"STOPPED":GOTO1530
1220 Q=PEEK(P)
1230 IFQ=102THEN1270
1240 IFQ=97THEN1390
1250 POKEP1,32:POKEP,42
1260 GOTO1180
1270 REMCRASH
1280 FORI=1TO100
1290 POKEP,97
1300 POKEP,105
1310 POKEP,95
1320 POKEP,98
1330 POKEP,105
1340 POKEP,95
1350 NEXT
1360 POKEP,42
1370 POKEP1,32
1380 PRINT"□";:PRINT"CRASHED                       ":GOTO1530
1390 T2=TI:T1=(T2-T)/60:T1=INT(100*T1):T1=T1/100
1400 PRINT"LAP TIME"T1"SEC
1410 IFT1<T3THENPRINT"NEW RECORD"
1420 IFT1<T3THENGOSUB1440
1430 GOTO1530
1440 REMSTORE RECORD
1450 S=1952
1460 T$=STR$(T1):L=LEN(T$)
1470 FORI=1TOL
1480 Y$=MID$(T$,I,1)
1490 Y=ASC(Y$)
1500 POKE(S+I),Y
1510 NEXT
1520 RETURN
1530 FORW=1TO1000:NEXT
1540 GOTO1090
READY.
```

- A simple 'sex test' could now be performed since the maximum energy channel would indicate the frequency which is significant to the speaker. Males have peak frequencies of less than 500 Hz, females and children have a higher dominant frequency. This is not foolproof but good fun. Speech recognition with this system is possible if the order of channels is saved in a file. Then when an incoming word is required to be recognised the stored files are compared with the channel order of the incoming word. This can be a 50% match being OK or finding the best match. Neither system is particularly good in the simple form. The problem is that BASIC is slow and if several matches are made for one word(about 16 per second would be a good idea) then the sort and compare becomes too slow. A job for machine code! If only simple words or noises are compared then a single sample will work well. To demonstrate this try out a musical instrument and see if different notes are recognised. A game can be written where words~or music drives a car round a track, or the words are used to control a simple robot. This system only allows prerecorded words to he compared whereas some professional systems attempt to recognise words as they are spoken. The principles of this system are simple. The challenge is for you to improve the system.E"CM
- RDjan2001

Back *Back to* Richard Davies NLnet Home Page