

sp synth Pet/Cbm

Digital Speech for Pet/Cbm

Kenneth Finn Bedford, NY, magazine Compute October 1982 issue 29 page 193

With this technique, your PET/CBM (4.0 BASIC or Upgrade, 16 or 32K) can store and play digitized speech. No special hardware is required. This program lets the PET digitize, store, playback and monitor speech or other audio signals from the tape deck. It also is a beginning for the processing of the digitized audio signals and can be used for a rudimentary voice- print analysis allowing you to discriminate between different people's voices.

The machine language of the program (from \$033A to \$03BD) is called Voice-Rec. Its job is to take the information from the cassette tape and store it in memory from \$1000 to \$4000. (Note: If you change locations \$035E and \$0391 to \$30 [which is machine language BMI] the memory will be saved from \$1000 to \$8000.)

A 20k Hz Sampling Rate

This program is interesting in several ways. The first is that the sampling time for the audio signal has been reduced to about 41 microseconds. This corresponds to a sampling rate of better than 20 Khz/second. One of the ways this was accomplished was by taking the program and practically duplicating it for the high-low and low-high transitions on CA1, which is the cassette read head. The sections from \$0349-\$037B and \$037C-\$03AE are almost identical. This was done to make the sampling rate the fastest possible.

Another peculiarity of the program is that the data is packed. Each memory byte contains either a number or \$FF, which means an overflow. The number corresponds to how many 40 microsecond loops went by before the signal changed from high to low or vice versa. This packing method allowed us to store about 20 seconds of audio in the 12K of memory allocated. While this does not seem like much time, remember that about 20K samples are taken every second. Without this packing, the entire 32K PET would fill up in about one and a half seconds. This packing is made possible by the silent periods between words and the presence of other low frequency components of human voice.

A third peculiarity of this program is that the paths taken by the program for the three possible conditions - no transition, overflow, transition -have all been equalized to within four or five microseconds. This is evident by the number of NOPs or (\$EA) in the program.

The second section of the program is called Voice. It goes from \$03B0 to \$03F6. It has been previously published in the November 1981 issue of COMPUTE! but has been modified here so that it can be co-resident with Voice-Rec; the two programs go well together. Its job is to allow you to position a voice tape by monitoring or listening to what is on it. It is very useful when you are trying to get the tape set up to record a specific segment of it.

A couple of things about it are interesting. First of all, it shows you how the stop key, the CB2 line, the tape read line, and the cassette motor can all be used from machine language. Second, it has an even higher sampling rate than Voice-Rec. Both this program and Voice-Rec can be in the second cassette buffer without any trouble, or they can be separated easily. Both are also relocatable.

The third part of the program is called Voice-Play. It goes from \$033A to \$03B3, and it can play back the recorded speech from memory \$1000 to \$4000. (This program also can be modified by changing \$036C and \$0397 to \$30 or BMI, and then it will play from \$1000 to \$8000.)

It has been designed to work with Voice-Rec in a similar way. Its timing loops at 43 microseconds match closely the loops of Voice-Rec; the playback is at least uniform, if not good.

Now let's examine the process that we have been using and see what we can now (do) with our digitized voice. What we have been doing is making the PET into a one-bit analog to digital converter. Another way of describing the process is saying that we have been making a record of an infinitely clipped signal. While this method is not quite as good as using an eight-bit ADC, it at least has the benefit of allowing us to get some experience cheaply and can be improved by the use of a good amplifier with tone controls on the PET's CB2 line. Since we are not capturing the signal in a very sophisticated way, I have chosen to make the sampling rate as high as possible to make up for it. That is why the first two program sections were not merged.

Let's begin by looking at the digitized data that we made and seeing how densely it has been packed.

```
10 POKE53,13:POKE52,0:CLR
20 FORI=4096TO16384
30 S=S+PEEK(I)
40 NEXTI:PRINTS/12288
```

This little program will produce the average byte value in the program. When I ran it, I got about 32, the average number of samples packed into each byte. This is why we can compress 20 seconds of information at a 20K Hz sample rate into only 12K bytes of memory.

Voice Analysis

A second analysis of the program was to produce a histogram of the signal.

Remember that each byte represents a sort of instantaneous frequency. Thus, we want to examine what amounts of each frequency were present.

```
10 POKE53,13:POKE52,0:CLR:DIMA%(256)
20 FORI=4096TO16384
30 A%(PEEK(I))=A%(PEEK(I))+1:NEXTI
40 OPEN4,4,0
50 FORI=1TO70
60 PRINTA%(I),A%(I+70),A%(I+140),A%(I+210)
70 NEXTI:CLOSE4:END
```

This little program will produce a histogram, running down the page, on the PET printer. For the sample that I used, the majority of the important information was contained in the first 50 or so numbers running down. This is not too surprising, since the average value of the sample was 32. (Note, please, that overflow samples of 256 or \$FF were not really treated correctly in this little analysis. They should have been added to the next following byte to get the correct frequency.) This data is a kind of voice-print for a person's speech. If you have different people say the same thing into a tape recorder and then analyze each voice with our system, you will get a separate voice-print. Women's voices, since they tend to be higher, will have higher amounts of lower numbers, which correspond to the higher frequency. While this system is crude, it does provide a departure point.

A third analysis of this data is to transform the signal via differentiation. Before you wring your hands in despair, remember that we are dealing with digitized information, and all we have to do is to transform the data by taking the difference between each number in our stored data base. The ease with which we can manipulate a signal once it is in memory is why we started this project in the first place.

Another thing we can do quite easily is to filter the signal any way we like. Try adding two or three numbers to each datum, and see how each modification changes the signal. While this technique is not strictly a filter, it illustrates the idea that digital processing of speech data is useful.

Remember that once the rough parts of the work have been done in machine language, the fun parts can be done in BASIC. This makes it simple to process the data.

One final point. Up to now we have been working in the time domain. We have a representation of how the voice looks at each point in time.

There are other ways we can present this signal. While the other methods cannot mathematically tell us more about the signal, they can give us other ways to look at it.

One famous method is to transform the signal into the frequency domain by using a fourier transform. This analysis gives an altogether different type of histogram of the signal.

How To Use The PET/CBM Software Voice Synthesizer

The program is a combination BASIC loader and a runtime helper. When RUN, it loads the machine language programs from the DATA statements. Type each number carefully, and save the program before you run it, in case you've made an error (remember to change the indicated lines if you have Upgrade ROMs or 16K memory).

The program presents you with three options: Monitor Tape, Record, and Play. The Monitor program simply plays the tape. Press the RUN/STOP key to stop monitoring. You must press RUN/STOP while the tape is playing something audible, or the program won't acknowledge you. If you press it quickly, without holding down, you'll be returned to the menu of options; otherwise you'll see the message: BREAK AT LINE XXX. You can type RUN to restart the program.

When you're ready to record the tape into your computer's memory, press PLAY on the tape player first, then press R for Record. The tape will run for about 20 seconds. You can then listen to the digitized voice or sound with Play. The quality is best with an external CB2 speaker (some 4032's and (all 8032's have a built-in piezoelectric "bell" that can produce low volume, high pitched CB2 sound). You can attach an amplifier to pins M and N on the user port if you want to add CB2 sound.

Change these lines for a 16K PET/CBM

```
1090 DATA 36, 37, 112, 45, 234, 234
```

```
1160 DATA 234, 36, 37, 112, 2, 80
```

```
1290 DATA 112, 80, 160, 6, 136, 208
```

```
1370 DATA 184, 36, 37, 112, 29, 160
```

Change these lines for Upgrade ROM PET/CBM

```
1170 DATA 173, 32, 123, 252, 86, 96
```

```
1420 DATA 16, 219, 48, 153, 32, 123
```

```
100 POKE53,16:POKE52,0:CLR:GOSUB1000
105 PRINT"{CLEAR}{REV}VOICE SYNTHESIS{DOWN}":P
    OKE59468,12
110 PRINT"{REV}M{OFF}ONITOR TAPE{DOWN}":PRINT"
    {REV}R{OFF}ECORD{DOWN}":PRINT"{REV}P{
    OFF}LAY{02 DOWN}"
120 PRINT"{UP}CHOICE?":GETA$:IFA$=" "THEN120
```

```

130 ON -(A$="M")-(A$="R")*2-(A$="P")*3 GOSUB 1
    50,160,170
140 GOTO105
150 PRINT"{DOWN}PRESS {REV}RUN/STOP{OFF} TO QU
    IT":SYS958:RETURN
160 PRINT"{DOWN}NOW RECORDING...":SYS826:RETUR
    N
170 PRINT"{DOWN}PLAYING...":SYS634:RETURN
1000 FOR ADRES=634TO759:READ DATTA:POKEADRES,DA
    TTA:NEXT ADRES
1010 DATA 120, 206, 19, 232, 169, 16
1020 DATA 133, 37, 165, 0, 133, 36
1030 DATA 168, 170, 177, 36, 170, 197
1040 DATA 255, 208, 5, 32, 225, 2
1050 DATA 80, 244, 169, 204, 141, 76
1060 DATA 232, 202, 240, 7, 160, 10
1070 DATA 136, 208, 253, 240, 246, 230
1080 DATA 36, 208, 2, 230, 37, 184
1090 DATA 36, 37, 48, 45, 234, 234
1100 DATA 234, 177, 36, 170, 197, 255
1110 DATA 208, 5, 32, 225, 2, 80
1120 DATA 244, 169, 236, 141, 76, 232
1130 DATA 202, 240, 7, 160, 10, 136
1140 DATA 208, 253, 240, 246, 230, 36
1150 DATA 208, 2, 230, 37, 184, 234
1160 DATA 234, 36, 37, 48, 2, 80
1170 DATA 173, 32, 192, 252, 88, 96
1180 DATA 234, 202, 240, 7, 160, 10
1190 DATA 136, 208, 253, 240, 246, 230
1200 DATA 36, 208, 2, 230, 37, 184
1210 DATA 96, 88, 169, 52, 133, 249
1220 FOR ADRES=826TO1014:READ DATTA:POKE ADRES,
    DATTA:NEXT ADRES
1225 RETURN
1230 DATA 120, 169, 16, 133, 37, 169
1240 DATA 0, 133, 36, 170, 169, 53
1250 DATA 141, 19, 232, 173, 16, 232
1260 DATA 169, 60, 141, 17, 232, 44
1270 DATA 17, 232, 48, 21, 232, 224
1280 DATA 255, 240, 12, 184, 36, 37
1290 DATA 48, 80, 160, 6, 136, 208
1300 DATA 253, 240, 234, 234, 234, 234
1310 DATA 234, 138, 145, 36, 162, 0
1320 DATA 230, 36, 208, 2, 230, 37
1330 DATA 44, 17, 232, 16, 219, 234
1340 DATA 173, 16, 232, 169, 62, 141
1350 DATA 17, 232, 44, 17, 232, 48
1360 DATA 21, 232, 224, 255, 240, 12
1370 DATA 184, 36, 37, 48, 29, 160
1380 DATA 6, 136, 208, 253, 240, 234
1390 DATA 234, 234, 234, 234, 138, 145
1400 DATA 36, 162, 0, 230, 36, 208
1410 DATA 2, 230, 37, 44, 17, 232
1420 DATA 16, 219, 48, 153, 32, 192
1430 DATA 252, 88, 169, 52, 133, 249
1440 DATA 169, 61, 141, 19, 232, 96
1450 DATA 120, 169, 53, 141, 19, 232
1460 DATA 160, 240, 141, 16, 232, 169

```

1470 DATA 16, 45, 18, 232, 240, 224
1480 DATA 173, 16, 232, 169, 62, 141
1490 DATA 17, 232, 44, 17, 232, 16
1500 DATA 251, 169, 204, 141, 76, 232
1510 DATA 173, 16, 232, 169, 60, 141
1520 DATA 17, 232, 44, 17, 232, 16
1530 DATA 251, 169, 236, 141, 76, 232
1540 DATA 208, 206, 0, 0, 0, 0

