Superior Software's "Speech!"

(This page was kindly contributed by Dave Jeffrey.)

Speech! from Superior Software was such a ground-breaking piece of software, that it is difficult to overestimate the impression that it made. It was featured on television programs - for instance Micro Live presenter Fred Harris demonstrated it on the BBC 1 Saturday morning children's show "Saturday Superstore". And it also starred on records - most famously on Roger Water's (ex-Pink Floyd) solo LP "Radio Chaos". "Radio Chaos" was a concept album about someone that creates a computerised voice (a person receives a credit for operating a Master 128 on the sleeve!).

At only £9.95 on tape (or £11.95 on disk), Speech! killed Acorn's own speech synthesis hardware, which they had gone to a great deal of effort to develop. A sad side-effect of this was that because Acorn bundled their speech synthesis hardware with their ROM cartridge socket (designed to go on the left hand side of the BBC Micro's keyboard), this upgrade also effectively died too. Acorn's stock of Speech Synthesis upgrades came to a sad end - being sold for peanuts in the Micro User's Reader Offers page.

A huge amount of work and ingenuity went into writing this Speech!. Check out the following texts:

From Your Computer, January 1986, "Software Shortlist" - page 42:

Speech

>BBC * Superior * Utility * £9.95 * Simon Beesley

Speech without a speech chip. It's hard to think of a more entertaining utility than this one. You can easily insert the program's 7K of code on someone else's disc, write a short Basic loader, and ask them to boot it up. Lo! a spoken message. The surprise effect is worth the price; like seeing colour on a ZX-81 screen.

Speech's speech may not always be intelligible if the words are not printed on the screen at the same time. But it is unusually easy to use. *SAY voices any English sentence, while *PITCH sets the pitch. If you are willing to put a bit of effort into it you can improve the program's enunciation by using *SPEAK which directly accesses a set of 49 phonemes. One of the demo programs will even take a text file - from Wordwise or View, for example - and painstakingly talk it through - if you can bear to listen to it. Although speech has been generated before on the CBM-64 and Spectrum, through the sound chip alone, this is the first time on the BBC. Elsewhere in this issue Speech's author, David Hoskins, explains how.

From BBC English, Your Computer, January 1986 - pages 84-86:

BBC English

Speech without a speech chip. David Hoskins explains how to do it.

The BBC has very easy to use and versatile sound facilities. The envelope command enables you to change the pitch and volume of any sound played. But there is one draw-back, you cannot change the actual waveform of the sound. So you always get that spikey, tinny kind of sound that's always recognisable as a BBC.

Digitise speech

About six months ago I was very impressed by some speech on a Commodore 64 game, and wondered if it was possible to do it on the BBC. The first problem - and quite a big one - was to find a way of encoding a waveform so I can literally play it from the BBC speaker! I will briefly explain how this can be achieved.

First, set the sound chip to it's highest possible frequency, which is inaudible to the human ear. The more channels used the louder the waveform will be. The waveform which varies in amplitude from 0 to 15 is then sent nibble by nibble to the volume control of each channel. The fastest speed the BBC can do this is about 10000 times a second, the faster you go the clearer the sound will become. What was needed now was the waveform data to feed the program. To do this I used the Interbeeb module from DCP Microdevelopments which sampled - recorded - data at a rate of 10000 times pre second, this was perfect for the job. After many weeks of frustration and fiddling I eventually had a result...

"It worked!" I exclaimed and ran downstairs to tell everyone. The scope for experiment was awesome. I spent a long time recording and playing sounds, at different speeds, backwards, with an echo, while mixing with other sounds, making drum beats, and even playing tunes with samples like some £280000+ synthesisers !

I had digitised about five seconds of music at a high sample rate, and it was now playing through the BBC's speaker under complete software control! To digitise speech I use a sampling rate of about 5.5khz and a filter to cut out any unwanted sounds above that frequency. I speak the words into a microphone which are converted from analogue form to digital by the Interbeeb, and sent to the BBC which in turn stacks the data in memory. The data can then be saved and played back at any time without using any hardware.

A phoneme solution

Since then I have been developing a program called Speech! which is being marketed by Superior Software. This system allows the user to type in English or phoneme phrases (explained later), which are then spoken by the program, e.g.*SAY Hello I am a speech synthesiser. The above command can be typed directly to the computer or put into a program. Speech! will translate this command and speak it through the BBC's speaker. This program, like a few hardware systems on the market, has an unlimited vocabulary. So how did I manage to digitise every word inthe English language in 32K?

The answer is in phonemes: the sounds that we string together in order to make speech. Take the words "mat" and "cat" for example, they both use exactly the same "a" sound. So in the sentence "The cat sat on the mat" all the "a" sounds can be just one block of speech data. The same thing applies to the "t"'s, "th"'s, "e"'s, and so on until you have covered as many phonemes needed for understandable speech. In Speech! I have used 49 different phonemes, which are merged to form whatever word or sentence you like.

Unfortunately to convert English phrases into phoneme data is a little more complicated. The trouble with the English language is that it is occasionally very illogical. For instance, the word "laughter" almost completely changes pronunciation when an "s" is prefixed to make "slaughter". Why is the word "rough" different from "bough" and "tower" different from "mower" merely because of the prefixing consonant? The reason why we know the differences and exceptions in the English language is because we have had years of training. The computer hasn't, and all it had to rely on is logic. Basically there can be at least 400 rules for text/speech conversion with a few thousand exceptions! Obviously to fit Speech! into 7.5K I have cut down on the exceptions, but any mispronounced words can be easily altered with a little thought.

Glottal pulse

To digitise each phoneme sound that uses the vocal chords for speech, i.e. "OO" not "S" I prolonged the sound (like when the doctor says "Say aaahh") into the microphone and sampled a small section of it. On analysis it can be seen that there is a close repetition in the waveform, every time the vocal chords produce what is known as a glottal pulse. By changing the pitch of my voice with the highest sample rate my program could handle I found that I could get the waveform to repeat every 128 samples. Unfortunately it is very difficult to remember and keep a steady pitch of voice throughout the recordings and it took many hours to complete the voiced - glottal pulse phonemes - sounds.

Voiced fricatives

Explosive sounds like "P" or "D" as in "Dear" never repeat so the 128 sample waveform is only played once. But this is not all. Before we make the explosive sound air pressure is built up behind the lips, making a

slight pause before pronunciation. This pause was reconstructed just before playback of the phoneme. Similarly, because a relatively small sample of the sound was taken a small pause was also needed after the phoneme as well.

The biggest problem with sampling and repeating waveforms is that it does not work with fricatives. These are phonemes which use "white noise" or resonated hiss e.g. "S" as in "Salt". Voiced fricatives use the resonant hiss but are also overlayed by a glottal pulse waveform. You cannot use the BBC's own white noise generator because it has only one set sound and doesn't resemble any of the phonemes.

To get around this problem I digitised 128 samples of spoken fricative, which fitted nicely into the rest of the data and could be easily tabled. To play the fricative Speech! picks a random number between 0 and 127 along the data of the sound and then plays a waveform block of 16 samples from that point. The process is then repeated by choosing the random number again. This is continued until the desired time of the phoneme is over. The result is quite and an accurate reproduction of the original fricative. The only problem left (soundwise) was the voiced fricatives.

No white noise

I found that it was possible to say these phonemes and not actually sound the white noise into the microphone. After digitising and tabling this sound I worked out that the fricative sound could be added to the waveform after it has played one section of about 100 samples. For example, in the phoneme "Z" as in "ZEBRA" SPEECH! plays about 100 samples of the wave (without the hiss) and then it plays 28 samples of the fricative "S". The reason why "S" was used in the phoneme "Z" was because it's sharp and loud, whereas "V" would need a mellow and quiet sound like "H" - as in "House".

Vary the pitch

The use of Speech! can also control the overall pitch of the voice and individual phonemes. Inserting full-stops and question marks vary the pitch of the voice towards the end of the sentence. To change the pitch Speech! simply plays back the phoneme at different speeds. But . . . the problem was that when you change the pitch of the voice in the phrase it suddenly changes where it should vary slowly. Also, after a voiced phoneme was played it suddenly jumped into the next one making it very hard to understand at the best of times! What was need was some kind of merging technique which will slowly change one waveform into another while they are playing.

Fairly accurate

With the phonemes "AA" in "Yard" and "EE" as in "Feet" the merging was achieved as follows. When the "AA" phoneme starts, 128 samples are played and then 0 samples of "EE". Then 127 samples of "AA" are played and 1 of "EE". This continues until the end of the "AA" phoneme when 0 samples are played of "AA" and 127 of "EE", you see! This merge is timed to finish exactly when the first phoneme has finished its cycle, no matter how long it is played.

The result is a gradually varied and fairly accurate reproduction of speech. An example of Speech! can be heard on two games from Superior Software: Citadel and Repton 2. Speech! and six programs and utilities can be bought at £9.95 on tape and £11.95 on disk.

```
10 REM     Waveform Encoded strange sounds program
20 REM             (C) 1985 DAVID J HOSKINS
30 REM                 for YOUR COMPUTER
40 REM   line 110 inserts a sinewave into waveform table
50 REM   this can be changed to create any sound
60 REM   once data is in memory line may be skipped
70
80 MODE4
90 PROCinit
100 MOVE0,448
110 FORA%=0TO255:Y%=7.9+8*SINRAD(360/256*A%)
120 DRAWA%*4,Y%*64:A%?table=Y%:NEXT
```

```
130 CALLstart
140 END
150
160 DEFPROCinit        :REM assemble code
170 temp=&71:table=&4000:sped=&70
180 tim=&72:coun=&74
190 FOR pass=0 TO 2 STEP2:P%=&4100
200
210 [OPT pass
220
230 .start
240
250 SEI
260 LDA #193:JSR in:LDA #0:JSR in     \ set up sound channels
270 LDA #129:JSR in:LDA #0:JSR in     \  "   "   "       "
280 LDA #161:JSR in:LDA #0:JSR in     \  "   "   "       "
290 LDA #0
300 STA tim
310 STA tim+1
320 STA sped
330
340 .loop
350 LDY#0
360 STY coun
370
380 .loop2
390 LDA table,Y
400 AND #15
410 JSR send
420 LDA tim
430 AND #127
440 CLC
450 ADC#1
460 TAX
470 .wait
480 DEX
490 BNE wait
500 TYA
510 CLC
520 ADC sped
530 TAY
540 CLC
550 ADC #3
560 STA tim
570 DEC coun
580 BNE loop2
590 LDA sped
600 AND #7
610 ADC tim+1
620 ADC tim
630 STA sped
640 INC tim+1
650 BNE loop
660 CLI
670 RTS
680
690 .send \send wave form to sound chip
700
710 EOR #15
720 STA temp
730 ORA #208
740 JSR in
750 LDA temp
760 ORA #144
770 JSR in
780 LDA temp
790 ORA #176
800 JSR in
810 RTS
```

```
820
830 .in
840
850 LDX #255
860 STX &FE43
870 STA &FE41
880 INX
890 STX &FE40
900 LDA &FE40
910 ORA #8
920 STA &FE40
930 RTS
940
950 ]
960 NEXT
970 ENDPROC
```