1 Introduction

"Chatterbox" is a self-contained Speech Synthesiser, including amplifier speaker and bidirectional computer interface. "Chatterbox" plugs directly into Sinclair ZX81 and Spectrum computers using the standard bus connector.

"Chatterbox" does not contain a fixed vocabulary, but uses phonemes, or individual speech sounds, to build up complete words under programme control. The quality of phoneme-synthesised speech is not usually as good as pre-programmed speech, but the flexibility is infinite. With care your "Chatterbox" can be made to produce words, phrases or sentences for any application.

2 Use of Phonemes

A complete list of phonemes, together with examples of their use and their code numbers used to command "Chatterbox", is given in Appendix B. Note that phonemes have been given mnemonics (normally two or three capital letters) in order to assist identification.

It is worth noting the following guidelines:

. The letters which form an English word do not necessarily identify the sounds required from "Chatterbox". Example: "enough" would be spoken using phonemes EH-NN2-UX-FF.

. Some letters require slightly different sounds, depending on whether they start or terminate a word. Example: d in "do" uses phoneme DD2, d in "could" uses phoneme DD1, which is less pronounced.

. Most of the vowel sounds can be stressed, or lengthened, by repeating the phoneme. Example: "extent" using EH-KK1-SS-TT2-EH-EH-NN1-TT2.

. Some phonemes exist already in long and short versions: Example: "letter" uses short phoneme ER, "fir" uses long phoneme ERR.

. Pauses (silence) may be used between words, and must be used to terminate speech, otherwise "Chatterbox" is liable to carry on repeating the last phoneme.

. Appendix A gives examples of words and their phonemes.

3 Programming for "Chatterbox"

"Chatterbox" is controlled as an output device, address 127 on Sinclair computers. Data in the form of phoneme codes is transferred under programme control, and "Chatterbox" can be interrogated to check that it is ready for the next command. Since output devices are difficult to handle in Sinclair Basic, we have provided a short machine-code routine.
The total number and length of the strings to be stored can be changed by altering lines 8 and 9, but the previously stored strings must then be ERASEd.

Remember: for best results some experimentation will be needed, to arrive at the best phoneme sequence for new words. Start by entering and listening to some of the examples in Appendix A.

4 Extending your CHATTERBOX

You have purchased more than just a Speech Synthesiser.

"Chatterbox" contains a complete interface for the well-known "Big Ears" Speech Recognition System, an audio input for amplifying sound from the Music Synthesiser, and finally an audio output for an external speaker, amplifier or tape recorder.

We hope that "Chatterbox" will give you many hours of enjoyment.

Interfacing "Chatterbox" to other computers

Hardware Connection (to data bus or PIA)

cable colour          signal          RIBBON CODE
mauve                DT-YELLOW        1
blue                 D6-BLACK         A
green                D5-WHITE         2
yellow               D4-MAUVE         3
orange               D3-GRAY          4
red                  D2-BROWN         B
brown                D1-RED           C
black                D0-ORANGE        D
white                CS1-YELLOW        0
grey                 CS2-CRANGE        1
pink                 RD-RED           2
pale blue            WR-BROWN         3
red/yellow           +5v-BLUE         4
screen               0v-GREEN         5

Programme Model
Write 0 0 x x x x x x x

Wait for Load Request
=0 before writing
next phoneme.

Note: CS1 and CS2 are both low for unit select.
On Sinclair computers
CS1= A7
CS2=TORQ

=1 out

1 = out

Wait for Load Request
Ears
Read  x x 1 1 1 1 x x
Load Request
Idle
To enter this routine, type in the following:

```
ZX81
1 REM 123456789012345
2 LET MEM=16514
3 LET TALK=MEM+1
   POKE 16515,219
   POKE 16516,127
   POKE 16517,230
   POKE 16518,1
   POKE 16519,32
   POKE 16520,250
   POKE 16521,58
   POKE 16522,130
   POKE 16523,64
   POKE 16524,211
   POKE 16525,127
   POKE 16526,201
```

Now SAVE on tape.

You can now test your "Chatterbox" by adding

```
10 POKE MEM,60
11 LET X=USR TALK
20 POKE MEM,0
21 LET X=USR TALK
RUN
```

"Chatterbox" should say "ear" (phoneme no.60). Note that lines 20 and 21 command silence, needed to prevent "Chatterbox" from repeating the sound indefinitely.

Now add

```
8 POKE MEM,63
9 LET X=USR TALK
RUN
```

"Chatterbox" should say "beer" (phonemes 63 and 60)

Clearly the above method of programming will be somewhat tedious, and a Speech Edit Programme is given in Appendix C. This allows you to compose, play and save 50 words or phrases of up to 20 phonemes each, including the silences. Each set of phonemes is stored in a "string", and is composed by entering the phoneme codes in sequence, using code 64 to terminate. A string may be examined by entering NEWLINEs instead of phoneme codes, and the compose mode is terminated by typing an X.

**NB** The first time you run Speech Editor you must answer "y" to the question "ERASE DATA?"

Once the words or phrases have been set up they may be tested by selecting Play.

To cause your own programmes (which you can add to the Speech Editor) to speak, use the following instructions:

```
.. LET WN=string number
.. GOSUB 300
```
Appendix A - Example Words

DD2-AO-TT2-ER (33-23-13-51)  "daughter"
KK3-UX-LL-AY-DD1       "collide"
SS-SS-IH-SS-TT2-ER      "sister"
KK1-LL-OW-NN1           "clown"
SS-KK3-WW-AIR           "square"
UX-NG-KK3-EL            "uncle"
KK1-UX-MM-PP-YY1-UW1-TT2-ER "computer"
TT2-UW2                 "two"
UX-LL-AR-MM             "alarm"
SS-KK3-OR               "score"

LDA A $FF  > PIAB  A9 FF
STA A $9.122  > OUT  8D 22.91
LDA $10  > READ  A9 10
STA A $9120  > PIA A 8D 20.91
LDA $00  > INPUTS  A9 00
STA A $9123  > 8D 23.91
CHECK LDA $01  STATUS BIT A9 01
CHECK AND A9 04
BEQ CHECK
LDA $04  > WRITE  2D 21.91
STA A $9120  > PIA A 8D F9
LDA $FF  > PIA A 8D 23.91
LDA 9123  > OUTPUTS  A9 FF
STA 9123  > 8D 23.91
LDA 033C  > MEM  AD 38 03
STA 9121  > 8D 21.91
RETI 60

169, 255, 141, 34, 145, 169, 16, 141
32, 145, 169, 0, 141, 35, 145, 169, 1
45, 33, 145, 240, 249, 169, 14
141, 32, 145, 169, 255, 141, 35
145, 173, 60, 3, 141, 33, 14
<table>
<thead>
<tr>
<th>letter</th>
<th>mnemonic</th>
<th>example</th>
<th>code</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>AA</td>
<td>ma</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>AE</td>
<td>hat</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>AO</td>
<td>aught</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>EY</td>
<td>page</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>AR</td>
<td>arm</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td>UX</td>
<td>above</td>
<td>15</td>
</tr>
<tr>
<td>b</td>
<td>BB2</td>
<td>rib</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>BB</td>
<td>builder</td>
<td>28</td>
</tr>
<tr>
<td>c</td>
<td>KK1</td>
<td>can</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>KK2</td>
<td>sky</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>KK3</td>
<td>comb</td>
<td>8</td>
</tr>
<tr>
<td>ch</td>
<td>CH</td>
<td>chuch</td>
<td>50</td>
</tr>
<tr>
<td>d</td>
<td>DD2</td>
<td>do</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>DD1</td>
<td>could</td>
<td>21</td>
</tr>
<tr>
<td>e</td>
<td>EH</td>
<td>end</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>EE</td>
<td>see</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>EAR</td>
<td>ear</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>ER</td>
<td>letter</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>ERR</td>
<td>err</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>AY</td>
<td>eye</td>
<td>6</td>
</tr>
<tr>
<td>f</td>
<td>FF</td>
<td>food</td>
<td>40</td>
</tr>
<tr>
<td>g</td>
<td>GG2</td>
<td>got</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>GG1</td>
<td>guest</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>GG3</td>
<td>wig</td>
<td>34</td>
</tr>
<tr>
<td>h</td>
<td>HH2</td>
<td>hoe</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>HH1</td>
<td>he</td>
<td>27</td>
</tr>
<tr>
<td>i</td>
<td>IH</td>
<td>sit</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>AY</td>
<td>size</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>ERR</td>
<td>fir</td>
<td>52</td>
</tr>
<tr>
<td>j</td>
<td>JH</td>
<td>dodge</td>
<td>10</td>
</tr>
<tr>
<td>k</td>
<td>KK3</td>
<td>comb</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>KK2</td>
<td>sky</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>KK1</td>
<td>can</td>
<td>42</td>
</tr>
<tr>
<td>l</td>
<td>LL</td>
<td>lake</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>EL</td>
<td>paddle</td>
<td>62</td>
</tr>
<tr>
<td>m</td>
<td>MM</td>
<td>milk</td>
<td>16</td>
</tr>
<tr>
<td>n</td>
<td>NN1</td>
<td>thin</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>NN2</td>
<td>no</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>NG</td>
<td>anchor</td>
<td>44</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>letter</th>
<th>mnemonic</th>
<th>example</th>
<th>code</th>
</tr>
</thead>
<tbody>
<tr>
<td>o</td>
<td>AA</td>
<td>hot</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>OH</td>
<td>low</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>AO</td>
<td>aught</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>UX</td>
<td>come</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>OR</td>
<td>store</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>UH</td>
<td>book</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>UW2</td>
<td>food</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>OW</td>
<td>out</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>OY</td>
<td>boy</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>UW1</td>
<td>to</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>PP</td>
<td>pig</td>
<td>9</td>
</tr>
<tr>
<td>qu</td>
<td>KK3-WW</td>
<td>square</td>
<td>8,46</td>
</tr>
<tr>
<td>r</td>
<td>RR1</td>
<td>real</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>RR2</td>
<td>brain</td>
<td>39</td>
</tr>
<tr>
<td>s</td>
<td>SS</td>
<td>test</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>ZH</td>
<td>pause</td>
<td>38</td>
</tr>
<tr>
<td>sh</td>
<td>SH</td>
<td>ship</td>
<td>37</td>
</tr>
<tr>
<td>t</td>
<td>TT1</td>
<td>part</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>TT2</td>
<td>two</td>
<td>13</td>
</tr>
<tr>
<td>th</td>
<td>TH</td>
<td>thin</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>DH1</td>
<td>this</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>DH2</td>
<td>this</td>
<td>54</td>
</tr>
<tr>
<td>u</td>
<td>UW1</td>
<td>put</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>UX</td>
<td>up</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>UH</td>
<td>full</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>UW2</td>
<td>rude</td>
<td>31</td>
</tr>
<tr>
<td>v</td>
<td>VV</td>
<td>vest</td>
<td>35</td>
</tr>
<tr>
<td>w</td>
<td>WW</td>
<td>wool</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>WH</td>
<td>which</td>
<td>48</td>
</tr>
<tr>
<td>x</td>
<td>KK1-SS</td>
<td>expert</td>
<td>42,55</td>
</tr>
<tr>
<td>y</td>
<td>YY1</td>
<td>yes</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>YY2</td>
<td>yes</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>AY</td>
<td>sky</td>
<td>6</td>
</tr>
<tr>
<td>z</td>
<td>ZZ</td>
<td>zoo</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>ZH</td>
<td>azure</td>
<td>38</td>
</tr>
</tbody>
</table>

Note: This Table is Copyright. It may not be reproduced in any way without written permission from Audio Vision Systems Limited.
Appendix C - SPEECH EDIT

NB Lines 1, 2 & 3 must be detailed in Chapter 3.

ZX81

1 REM RND:=A NEW "4 IF U/RNDP
2 LET STAN=45
3 LET TALK=16515
4 PRINT "ERASE DATA? (Y/N)"
5 INPUT A$
6 IF A$="Y" THEN GOTO 20
7 LET LU=20
8 DIM U$(50,LU)
9 PRINT "ALL GONE"
10 GOTO 21
11 PRINT "COMPOSE SAVE OR PLAY ?(C/S/P)"
12 INPUT A$
13 IF A$="C" THEN GOTO 50
14 IF A$="S" THEN GOTO 6000
15 GOTO 200
16 REM
17 REM
18 REM "COMPOSE/EXAMINE"
19 PRINT "STRING NO:");
20 INPUT UN
21 IF UN=0 THEN GOTO 20
22 PRINT UN
23 LET CN=0
24 PRINT
25 LET CN=CN+1
26 PRINT CODE U$(UN,CN);" >>"
27 INPUT C$
28 IF C$="X" THEN GOTO 20
29 IF C$="X" THEN GOTO 20
30 LET CC=VAL C$
31 PRINT CC
32 LET U$(UN,CN)=$CHR CC
33 IF CN=LU THEN GOTO 20
34 GOTO 62
35 REM
36 REM "SPEAK STRING NO:");
37 PRINT UN
38 IF UN=0 THEN GOTO 20
39 PRINT UN
40 GOSUB 300
41 GOTO 20
42 REM "WORD OUTPUT"
43 LET CN=1
44 POKE MEM, CODE U$(UN,CN)
45 IF PEEK MEM>63 THEN RETURN
46 LET X=USR TALK
47 LET CN=CN+1
48 IF CN=LU THEN RETURN
49 GOTO 310
50 REM
51 REM "TAPE ON, THEN RETURN"
52 PRINT "Chatterbox demonstration"
53 PRINT "(C) 1982 AudioVision Systems"
54 PRINT GOTO 1