J.C.B. (MICROSYSTEMS)
29 Southbourne Road, Bournemouth, Dorset, BH6 5AE
Tel: (0202) 423973

J.C.B. (MICROSYSTEMS)

SPEECH SYNTHESIS MODULE

for the

DRAGON 32 COMPUTER

USER MANUAL

CONTENTS

1) INTRODUCTION

Thank you for purchasing our Speech Synthesis Module for the Dragon 32. We hope you enjoy the extra dimension it gives to your computing, as the possibilities for its use are many - in games, in educational programs, in introducing pre-school children to computing etc.

The J.C.B.(MICROSYSTEMS) Speech Module is based on the General Instrument SP0256 Speech Processor, which simulates speech electronically from a set of 64 allophones or individual speech sounds. It is this extra hardware that makes the Module much more intelligible than software based voice synthesisers. To successfully use a set of allophone sounds to synthesise words, you must always think about how a word sounds, rather than how it is spelled. We shall be producing, on cassette, a program which relates text to pronunciation, for those who would like this facility. Meanwhile, if you require any further information, please don't hesitate to get in touch with us.

2) GENERAL INFORMATION

The J.C.B. Speech Module is very easy to use, as it is operated by a set of BASIC commands which are introduced when the module is plugged in. REMEMBER to switch OFF the computer before inserting the module into, or removing it from, the cartridge port.

The commands available allow you to enter words in two formats:-

a) You may enter words in their normal written format, as long as the word is amongst those listed in the table of pre-defined words in section 9. You may also instruct the computer to say any number by entering it in numeric form. (see Section 3)

b) You may enter words as a sequence of allophones, by typing the allophones themselves rather than numbers, as this makes for ease of editing whilst the best combination of allophones is being selected. It also saves having to look up two sets of data to formulate the word. (section 4)

It is possible to use either format or to mix them when putting words together to form sentences - even paragraphs - which will then be spoken on the use of a speech command. (see Section 5)

The Module also incorporates a facility whereby it can speak irrespective of whether the computer may be doing other things at the same time. (see also Sections 6 and 8) However, the Module will wait for any existing speech to finish before executing a further SAY or SPEAK command.

3) USE OF 'SAY' COMMAND

3.1

The command 'SAY' is used to make the Module speak one of the pre-defined words it contains, or to say any number that the computer understands. A complete list of pre-defined words is given in alphabetical order in Section 9(a).

3.2 Immediate Mode

The syntax requirements of the SAY command are very simple. All words and pauses <u>must</u> be separated by commas. If a comma is omitted, the next word will be ignored.
Try this:-

```
SAY "HELLO,4,WHAT,4,IS,4,YOUR,4,NAME,4"
```

The number 4 represents a pause, and can be any number within the range 1 to 5 (section 9(b)).
Note the pause at the end of the example. It is necessary to end your word, or series of words, with a pause, otherwise the Speech Module will continue to speak the last allophone used. However, you may leave out the pause inbetween two words and thus add them together.
e.g. SAY "GOOD,MORNING,4" will say 'Good Morning'

There are several suffixes included amongst the pre-defined words, such as ---ing, ---s, ---est. These can be added onto words by omitting the pause.
e.g. SAY "LONG,EST,4" will say 'longest'

Spaces are always ignored, so the computer will understand both of the following:-
SAY "HE LL O ,4" and SAY "HELLO , 4"

The Module only uses the first four letters of a word to recognise it. Therefore, it may not matter if a word is not spelled entirely accurately, as long as the first four letters are correct. Thus:-

SAY "NUMB,4" will say 'Number'

3.3 Using Strings

The SAY command is compatible with all legal string handling commands on the Dragon. Try this:-

```
10  B$="GO,4"
20  A$(1)="NORTH,4"
30  A$(2)="SOUTH,4"
40  A$(3)="EAST,4"
50  A$(4)="WEST,4"
60  SAY B$,A$(RND(4))
70  GOTO 60
```

Complete phrases may be stored within strings, which may be sliced or concatenated, and then added together, by a command such as SAY A$,B$,MID$(C$,4,4),D$

3.3 contd

Words and phrases may be strung together like this until the speech buffer is full. The buffer will hold up to about 40 or 50 words, depending on their length. If the buffer should overflow, the program will stop with an error. (see Section 7)

You should be careful when slicing or adding strings, to ensure that the correct commas and pauses are included, and that you do not cut words in half.

Here is a short program for you to enter:-

```
10  REM***SAY THE ALPHABET***
20  FOR N=65 TO 90
30  A$=CHR$(N)
40  A$=A$+",4"
50  SAY A$
60  NEXT N
```

3.4 Input of strings

If you wish to input to a program either pre-defined words or allophones, then use the command LINE INPUT rather than just INPUT, as INPUT would ignore anything after the first comma. Try this:-

```
10  LINE INPUT A$
20  A$=A$+",4"
30  SAY A$
40  GOTO 10
```

When this program is RUN, you could use it to hear the pre-defined words. e.g. when prompted by the cursor, enter WHICH or NUMBER or WHICH,4,NUMBER

3.5 Numbers

For speaking numbers, use the SAY command followed by# e.g. SAY #12 will say 'twelve'

Note that it is <u>not</u> necessary here to put a pause at the end, as this is done automatically.

The SAY# command will say any number within the computer's range. It will also evaluate an expression such as 3*3 to say 'nine'. SAY # 55↑10 will say the result in exponential form.

3.5 contd

SAY# may be used with numeric variables and is compatible with all the normal Dragon numeric functions. Try this:-

```
10  FOR N=&H8000 TO &H8010
20  SAY# PEEK(N)
30  NEXT N
```

The commands SAY and SAY# may be intermixed on one line e.g. SAY"HELLO,3,I,3,AM,3,A,3,DRAGON,3",# 32,"COMPUTER,4"

<u>or</u> SAY A$,B$,"HELLO,4",C$,# N,D$,# 100 etc.

The following program will say the multiplication tables up to 12.

```
10  FOR N=1 TO 12
20  FOR M=1 TO 12
30  SAY# N
40  SAY "TIME,SZ,4"
50  SAY# M
60  SAY "EQUAL,SZ,4"
70  SAY# N*M
80  FOR X=1 TO 200: NEXT X
90  NEXT M,N
```

Another example program:-

```
    5 REM*SUBROUTINE TO SPEAK A NUMBER IN HEXADECIMAL*
   10 INPUT A
   20 GOSUB 10000
   30 GOTO 10
10000 REM*A=THE NUMBER TO BE SPOKEN IN HEX (A IS A DECIMAL NO)
10010 AA$=HEX$(A)
10020 HH$="HH1,EH,2,KK2,SS,4"
10030 LL=LEN(AA$)
10040 FOR L=1 TO LL
10050 NN$=MID$(AA$,L,1)
10060 NN=ASC(NN$)
10070 IF NN>47 AND NN<58 THEN GOTO 10140
10080 NN$=NN$+",4"
10090 SAY NN$
10100 NEXT L
10110 SPEAK HH$
10120 RETURN
10130 REM*THIS PART CONVERTS THE ASCII CODE TO A VALID NO.*
10140 NN=NN-48
10150 SAY# NN
10160 GOTO 10100
```

4) USE OF 'SPEAK' COMMAND

4.1

The SPEAK command is used for saying words formulated using the allophones. A list of allophones, together with some guidelines for their use, is included in Section 9(b). Examples are given there to illustrate the sound produced, but only experience will ultimately enable you to formulate the best sounding combination of allophones. We have added some hints, based on our own experience.

4.2 Immediate mode

The syntax required is the same as for the SAY command, in that allophones and pauses must be separated by commas, but it is essential for the allophone mnemonic to be spelled correctly, otherwise an error message will appear. Try this:-

```
SPEAK "AY,4,AE,1,MM,4,DH1,AX,4,SS,SS,AX,NN1,SH,AY,NN1,4
      AX,VV,4,YY|,OR,4,LL,AY,FF,FF,4"
```

4.3 Using strings

The SPEAK command also is compatible with all legal string handling commands on the Dragon. The sentence in 4.2 could be split up and put into strings as follows:-

```
10  A$="AY,4,AE,1,MM,4"
20  B$="DH1,AX,4,SS,SS,AX,NN1,SH,AY,NN1,4"
30  C$="AX,VV,4,YY|,OR,4,LL,AY,FF,FF,4"
40  SPEAK A$,B$,C$
```

N.B. Remember to use LINE INPUT if you wish to input allophones to a program (see 3.4).

5) USE OF 'ADD' COMMAND

When constructing phrases or sentences, it is likely that you may want to intermix the ease of using pre-defined words with the flexibility of formulating words from allophones. The ADD command has been included for this purpose.

The method of using the ADD command enables you to bring together any number of SAY and SPEAK commands, which will then all be spoken consecutively, as if only one command had been used.

5) contd

e.g. ADD;SAY"HELLO,4";SPEAK A$;SAY"HOW,4,ARE,4,YOU,4";SPEAK A$

Assuming that A$ has already been defined as 'John' for example, this command will say 'Hello John How are you John'

Note the syntax of the ADD command. Each statement is separated by a semi-colon.
Only the SAY and SPEAK commands may be mixed in this way.
The ADD command does not operate on any other BASIC commands.

6) USE OF 'WAIT' COMMANDS

6.1

The SAY and SPEAK commands do not halt the program whilst the speech is taking place, which enables you to mix speech and graphics very effectively. However, all speech commands will ensure that any existing speech is finished before starting a new speech command. This will not hold up normal BASIC unless a second speech command is encountered before the unit finishes speaking the first command. If this happens then a Wait state is entered until the current speech ends. You may, however, want the speech to finish before the computer continues with the next command, and in this case, you should use the WAIT ON command.

WAIT ON forces the processor to stop whilst the speech unit is talking, and the computer will stay in this mode until the WAIT OFF command is used.

The following program ensures that the number cannot be input until the speech module has finished saying it.

```
10  WAIT ON
20  CLS
30  PRINT"ENTER THE NUMBER WHICH IS SPOKEN"
40  A=RND(1000)
50  SAY#A
60  INPUT N
70  IF N=A THEN SAY"O,3,K,4":FOR D=1 TO 100:
    NEXT D: GOTO 20
80  SAY "WRONG,4": GOTO 50
```

6.2

If you wish to include a line in your program to check whether the unit is still talking, try:-

```
100  IF(PEEK(&HFF43) AND 1)=1 THEN GOTO xxx
                                where xxx is a line before 100
```

This will check the interrupt status of the Module's PIA and the expression will equal 1 whilst the module is active. This can be used to repeat a graphic sequence, for example, until the speech has finished.

7) ERRORS

7.1

As an aid to programming, we have incorporated into the Dragon's normal error routines, various messages relating to the Speech Module. These are followed by the report, ?FC ERROR, and a line number where relevant.
The error messages are as follows:-

a) ILLEGAL WORD - occurs when a word is either mis-spelled or is not defined in the table in section 9. This will only occur with the SAY command.

b) ILLEGAL ALLOPHONE - occurs when an allophone mnemonic is spelled incorrectly.

c) ILLEGAL PAUSE - occurs when a number greater than 5 is used for a pause.

d) UNDEFINED VARIABLE - occurs when you try to use SAY or SPEAK with a string variable which is empty.

e) PHRASE TOO LONG - occurs when the phrase or sentence being spoken is too long to fit into the speech buffer. The remedy is to split it into separate phrases.

7.2

The omission of commas or semi-colons will lead to ?SN ERROR reports.

7.3

This short program could be of use to include in programs where there are so many words or allophones involved, that trying to find your mistake could give you double vision!

```
19995  END
19996  REM*THIS ROUTINE WILL PRINT AN ILLEGAL WORD OR
ALLOPHONE TO THE SCREEN*
19997  REM*ENTER THE FOLLOWING PROGRAM LINES AT THE END OF
YOUR PROGRAM WHILST THE PROGRAM IS BEING WRITTEN*
19998  REM*WHEN THE PROGRAM IS COMPLETE, DELETE LINES 19995
ONWARDS*
19999  REM*USE BY TYPING 'RUN 20000' WHEN YOU GET AN ERROR
MESSAGE*

20000 CLS:PRINT"ILLEGAL WORD/ALLOPHONE IS-------------'";
20010 TT=&H1DØ
20020 CC=PEEK(TT)
20030 IF CC=>128 THEN 20060
20040 PRINT CHR$(CC);
20050 TT=TT+1:GOTO 20020
20060 PRINT"'"
20070 STOP
```

8) TECHNICAL INFORMATION

8.1

The speech module uses the cassette buffer to store the speech data and also uses the non-maskable interrupt (NMI) to continue speech whilst BASIC is running as normal (unless WAIT ON is used). Therefore, do not load from, or save to, the cassette whilst speech is in progress.
If you should wish to speak commands in order to instruct the user to prepare the cassette for data input or ouput, then ensure that WAIT ON is used to prevent the user from starting the cassette prematurely.

8.2

All audio/analogue signals are routed through one selector chip in the Dragon - therefore, use of the joysticks or SOUND/PLAY commands, whilst speech is in progress, is to be avoided, as it will switch off the speech. Joysticks may of course be used in a program in which speech occurs, as long as they are not used whilst the module is actually producing speech.

8.3

Addresses &H144 - &H147 are used to store 4 flags. These addresses are normally used to store the USR call addresses for USR 8 & 9, but as these calls do not work (due to a bug in the Dragon's ROM) you should have no reason to use them from BASIC. However, the command DEF USR 8=&HFFFF will in fact store the address FFFF at location &H144/5 and will corrupt the flags which are stored there; similarly so for DEF USR 9.

8.4 For Machine Code Programmers

8.4.1

The table of allophones in Section 9(b) has a number in decimal/hex beside it. This is the number which, when written to the speech synthesiser IC will generate the allophone to which it refers.
The synthesiser is addressed through a port located within the input/output area of memory space (i.e. from $FF00 to $FFFF) and thus you actually address the port, rather than the synthesiser. The port used is a 6821 PIA, which is the same as that used inside the Dragon for addressing the keyboard and generating sound (amongst other things!). The 6821 has built-in facilities for providing the various signal inputs and outputs to the synthesiser apart from the normal data bus. This means that when a byte of data is written to the synthesiser via the PIA, the PIA will provide all the necessary strobe signals etc. thus simplifying the process for the programmer.
The interrupt output from the PIA is configured to provide an interrupt whilst the synthesiser is not busy. This means that the interrupt will be generated immediately the synthesiser ceases speaking.
The interrupt driven speech generation routines may be used from machine code, and the details which follow should be of use to experienced programmers. Further details on the operation of the 6821 PIA may be obtained from books such as Leventhal's "6809 Assembly Language Programming."

8.4.2 Interrupt driven speech generation

The interrupt may be enabled by setting bit Ø of the PIACR at $FF43. It is disabled by resetting bit Ø. This is done very simply using the commands INC PIACR to enable
and DEC PIACR to disable

8.4.2. contd

The interrupt routine, when enabled, will look at a storage buffer which is pointed to by the contents of $2AC. It will load the data at this address (one byte) into the synthesiser, increment the pointer at $2AC, decrement a counter at $144, and if the counter is not zero it will return, else it will disable the interrupt and then return. The method of use is therefore quite simple:-

a) Set aside a section of RAM of maximum length 255 bytes, where the allophone data will be held.

b) When the data is loaded into this buffer area, store the start address at $2AC.

c) Store the number of allophones (max 255) at $144.

d) Use the command INC PIACR when you wish speech to commence.

Although the interrupt routine will disable itself when it has finished speaking the allophones from the buffer, you may wish to detect whether or not it is still talking. This is very easily done by testing the counter at $144 and checking for zero, or by reading the PIACR and testing bit Ø.

8.4.3 Non-interrupt controlled speech

If you should wish to enter allophone data to the synthesiser without using the interrupt routine, then you need some means of checking to see if the current allophone is still being spoken, as machine code runs many times faster than the speech produced! This can be accomplished by reading the PIACR. Bit 7 will be set when the synthesiser finishes speaking the current allophone. This Bit is used as a means for the PIA to remember that it has received an interrupt input and it can only be cleared by reading the PIADR which in this case is $FF42.

The format of a routine to write data to the synthesiser should be based on the following outlines:-

```
START:-    WRITE 1 Byte of data to $FF42
           READ $FF42    (to clear Bit 7)
LOOP       READ $FF43
           TEST BIT 7
           IF BIT 7=Ø THEN LOOP
           BRANCH TO START
```

The following short Assembler program will speak the allophones available (except pauses).

```
7000          100          ORG      $7000
7000          110  **THIS SWITCHES ON DRAGON SOUND CHANNEL**
7000  B6 FF01 120          LDA      $FF01
7003  84 F7   130          ANDA     #$F7
7005  B7 FF01 140          STA      $FF01
7008  B6 FF03 150          LDA      $FF03
700B  8A 38   160          ORA      #$38
700D  B7 FF03 170          STA      $FF03
7010  B6 FF23 180          LDA      $FF23
7013  8A 38   190          ORA      #$38
7015  B7 FF23 200          STA      $FF23
7018          210  **MAIN ROUTINE TO SPEAK EACH ALLOPHONE**
7018  86 05   220          LDA      #5
701A  B7 FF42 230  LOOP    STA      $FF42
701D  7D FF42 240          TST      $FF42
7020  7D FF43 250  WAIT    TST      $FF43
7023  2A FB   260          BPL      WAIT
7025  4C      270          INCA
7026  81 40   280          CMPA     #64
7028  26 F0   290          BNE      LOOP
702A          300  **ENTER A PAUSE TO STOP ANY SPEECH**
702A  7F FF42 310          CLR      $FF42
702D  39      320          RTS
```

9) TABLES

a) Pre-defined words

A
ADD
AFTER
AGAIN
AGE
AIM
ALL
ALLOW
ALPHA
AM
AND
ANOTHER
ANSWER
ARROW
AS
AT

B
BACK
BET
BLACK
BLUE
BOTTOM
BY

C
CAN
CASSETTE
CHECK
CIRCLE
CLOCK
COLOUR
COMMAND
COMPUTER
CONTINUE
COPY
CORRECT
CURSOR

D
DATA
DATE
DAY
DD
DECIDE
DIFFICULT
DIVIDE
DO
DOWN
DRAGON
DRAW

Pre-defined words contd

E
EACH
EAST
EASY
ED
ELSE
END
ENTER
EQUAL
ERROR
EST
EXAMPLE
EXPLAIN

F
FAIL
FALSE
FAST
FINAL
FINISH
FIRST
FROM

G
GAME
GET
GO
GOOD
GREEN

H
HALF
HARD
HAVE
HE
HELLO
HELP
HERE
HIGH
HOUR
HOW

I
IF
IN
ING
INSTRUCTION
IS
IT

J

K
KEY

L
LARGE
LAST
LEARN
LEFT
LETTER
LEVEL
LINE
LIST
LITTLE
LONG
LOSE
LOW

M
MAKE
MANY
MAY
MINUS
MORE
MORNING
MOST
MOVE
MULTIPLY
MUST
MY

N
NAME
NEVER
NEXT
NO
NORTH
NOT
NOW
NUMBER

O
ODD
OF
OFF
ON
ONLY
OPTION
OR
ORANGE
OVER

P
PASS
PENCE
PLEASE
POINT
POUND
PRESS
PRINT
PROGRAM
PUT

Q
QUARTER
QUESTION

R
READ
RED
RETURN
RIGHT
RUN

S
SAME
SAVE
SAY
SCORE
SELECT
SENTENCE
SHORT
SLOW
SOUTH
SPACE
SPELL
SS
START
STOP
SUBTRACT
SUM
SZ

T
TALL
THANK
THAT
THE
THEN
THINK
THIS
TIME
TOTAL
TRUE
TRY
TYPE

U
UNDER
UP
USE

V
VALUE
VERY
VOICE

W
WANT
WAS
WERE
WEST
WHAT
WHICH
WHITE
WILL
WIN
WORD
WRITE
WRONG

X

Y
YELLOW
YES
YOUR

Z

b) Allophones

The sounds of a language are called phonemes, but there are acoustic differences depending on the position of the phoneme in the word. This is why there are two or three versions of some phonemes included in the allophone set.

Guidelines for using the allophones:-

Silence

PA1 (10ms)	1	before BB, DD, GG and JH
PA2 (30ms)	2	before BB, DD, GG and JH
PA3 (50ms)	3	before PP, TT, KK and CH and between words
PA4 (100ms)	4	between clauses and sentences
PA5 (200ms)	5	between clauses and sentences

Short Vowels

*/IH/	sitting, stranded
*/EH/	extent, gentlemen
*/AE/	extract, acting
*/UH/	cookie, full
*/AO/	talking, song
*/AX/	lapel, instruct
*/AA/	pottery, cotton

Long Vowels

/IY/	treat, people, penny
/EY/	great, statement, tray
/AY/	kite, sky, mighty
/OY/	noise, toy, voice
/UW1/	computer
/UW2/	two, food
/OW/	zone, close, snow
/AW/	sound, mouse, down

/ER1/	letter, furniture, interrupt
/ER2/	bird, fern, burn
/OR/	fortune, adorn, store
/AR/	farm, alarm, garment
/YR/	hear, earring, irresponsible
/XR/	hair, declare, stare

* These allophones can be doubled

9(b) Allophones contd

/WW/	we, warrant, linguist
/RR1/	read, write, x-ray
/RR2/	brown, crane, grease
/LL/	like, hello, steel
/EL/	little, angle, gentlemen
/YY1/	cute, beauty, computer
/YY2/	yes, yarn, yo-yo

/VV/	vest, prove, even
/DH1/	this, then, they
/DH2/	bathe, bathing
/ZZ/	zoo, phase
/ZH/	beige, pleasure

*/FF/	These may be doubled for initial
*/TH/	position and used singly in
*/SS/	final position.
/SH/	shirt, leash, nation
/HH1/	before front vowels YR,IY,IH,EY,EH,XR,AE
/HH2/	before back vowels UW,UH,OW,OY,AO,OR,AR
/WH/	white, whim, twenty

/BB1/	rib, fibber, bleed, brown
/BB2/	initial position before vowel, beast
/DD1/	final position, played, end
/DD2/	initial position, down, drain
/GG1/	before YR,IY,IH,EY,EH,XR
/GG2/	before UW,UH,OW,OY,AX, green, glue
/GG3/	before AE,AW,AY,AR,AA,AO,OR,ER, anger, peg

/PP/	pleasure, ample, trip
/TT1/	before SS, tests, its
/TT2/	test, street
/KK1/	before YR,IY,IH,EY,EH,XR,AY,AE,ER,AX, cute,
/KK2/	speak, task
/KK3/	before UW,UH,OW,OY,OR,AR,AO, quick, clown

* These allophones can be doubled

continued - - - - -

9(b) Allophones contd

/CH/	church, feature
/JH/	judge, injure
/MM/	milk, alarm, ample
/NN1/	before YR,IY,IH,EY,EH,XR,AE,ER,AX,AW,AY,UW earn
/NN2/	before UH,OW,OY,OR,AR,AA
/NG/	string, anger

ALLOPHONE ADDRESS TABLE

Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex	
0	00	Pause 1	16	10	MM	32	20	AW	48	30	WH
1	01	Pause 2	17	11	TT1	33	21	DD2	49	31	YY1
2	02	Pause 3	18	12	DH1	34	22	GG3	50	32	CH
3	03	Pause 4	19	13	IY	35	23	VV	51	33	ER1
4	04	Pause 5	20	14	EY	36	24	GG1	52	34	ER2
5	05	OY	21	15	DD1	37	25	SH	53	35	OW
6	06	AY	22	16	UW1	38	26	ZH	54	36	DH2
7	07	EH	23	17	AO	39	27	RR2	55	37	SS
8	08	KK3	24	18	AA	40	28	FF	56	38	NN2
9	09	PP	25	19	YY2	41	29	KK2	57	39	HH2
10	0A	JH	26	1A	AE	42	2A	KK1	58	3A	OR
11	0B	NN1	27	1B	HH1	43	2B	ZZ	59	3B	AR
12	0C	IH	28	1C	BB1	44	2C	NG	60	3C	YR
13	0D	TT2	29	1D	TH	45	2D	LL	61	3D	GG2
14	0E	RR1	30	1E	UH	46	2E	WW	62	3E	EL
15	0F	AX	31	1F	UW2	47	2F	XR	63	3F	BB2

Although examples have been given to illustrate the type of sound produced by each allophone, these are only suggestions, and your own experience will enable you to effectively combine the allophones.

The following notes are based on our own experience:-

1) BB KK GG TT DD PP CH JH - The sound of a word containing any of these allophones will nearly always be improved if a pause is inserted immediately prior to the allophone, usually a pause from 1 to 3. e.g.SPEAK "KK1,AR,3,DD1,4" sounds (to us!) much more intelligible than SPEAK "KK1,AR,DD1,4"

9(b) contd

2) For words beginning with DE- such as Delimit, Deduce, Decimal, the 'e' often sounds more like 'i'. The following prefixes may be useful:-

DD1,IY ---- De-limit
DD1,IH ---- De-duce
DD1,EH ---- De-cimal

3) Although FF,TH and SS are designed to be doubled up in particular cases, some of the vowels also may be extended in this way. Try experimenting!

4) Pauses between words and sentences should not be the same, and, in fact, some words require a shorter pause between them than others. For instance, the phrase 'How are you' seems better with a shorter pause between 'How' and 'are' than between 'are' and 'you', Pauses can of course be combined to give the required length.

5) Many words ending in LL will sound better if the allophone EL is used instead of LL. Try forming the word 'bill' to see what happens.

6) The allophones BB,RR and GG are perhaps the most difficult to use, especially when followed by RR, and a little experimentation can provide unexpected improvements in the speech quality.

7) The 'a' in 'ball' is particularly difficult to pronounce, as the speech chip uses American phonetics. Sometimes the allophone OR may be used, although the 'r' is usually too pronounced, and AO may have to be used in the end, despite the American 'twang.'

8) When adding --s to a word, you will sometimes find that ZZ should be used instead of SS, depending on how you would normally say the word. e.g. 'Drives' will use ZZ and 'Biscuits' will use SS

REMEMBER - always think of the way you would say a word, not how it is written, when assembling allophones to form words.

NOTES