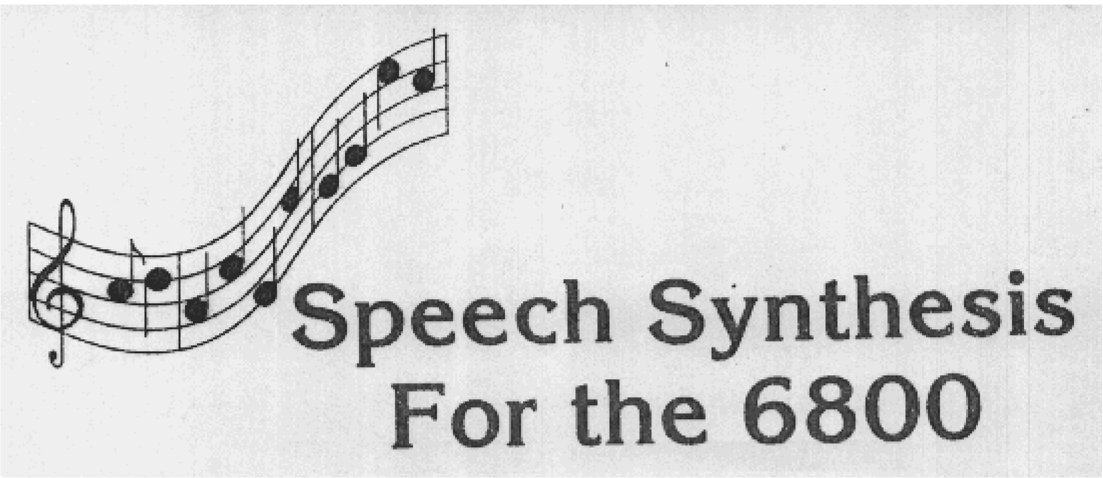
sp synth 6800

Speech Synthesis For the 6800

Give your SWTP the gift of gab for uses ranging from computer-aided instruction to audible I/O far the handicapped.

Dropbox



by Terry L. Mayhugh, 11632 Midhurst Drive, Concord, TN 37922
published in Microcomputing December 1980

A speech synthesizer Is one of the most exciting peripherals that a hobbyist can add to a home computer system. Computerized speech synthesis opens the door to an almost unlimited number of new uses for a micro, ranging from preschool computer-aided instruction to audible I/O for blind users.

Until recently, genuine speech synthesis equipment was not only difficult to find, but far beyond the budget of the average experimenter. However, researchers from the Votrax Division of the Federal Screw Works have produced a scaledown version of one of their more expensive units. Designed specifically to interface to the TRS-80 through its expansion connector, the unit is marketed by Radio Shack and distributed through their many dealers.

Fortunately, due to its simple parallel Interface requirements, it can be easily interfaced to practically any computer. Hardware enthusiasts can connect it to any eight-bit parallel port. This article includes details for an interface to an MPLA card in an SWTP 6800 computer.

Fig. 1. Votrax ribbon cable connector.

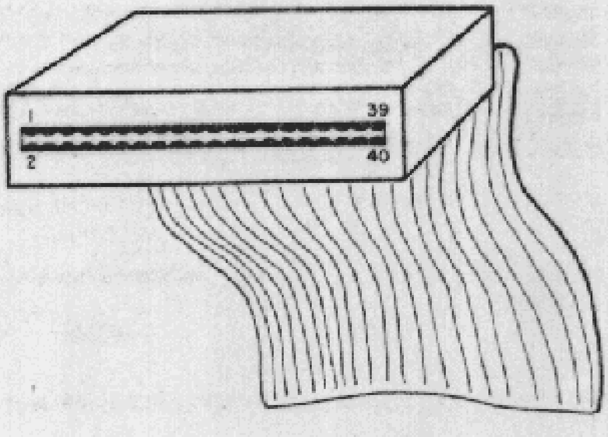


Fig. 1. Votrax ribbon cable connector.

120 Microcomputing, December 1980

Synthesis by Rule

The Votrax device uses a "synthesis by rule" technique of voicing words by synthesizing individual units of speech called phonemes. With suitable software, the phonemes are sequentially output to the synthesizer to produce intelligible speech.

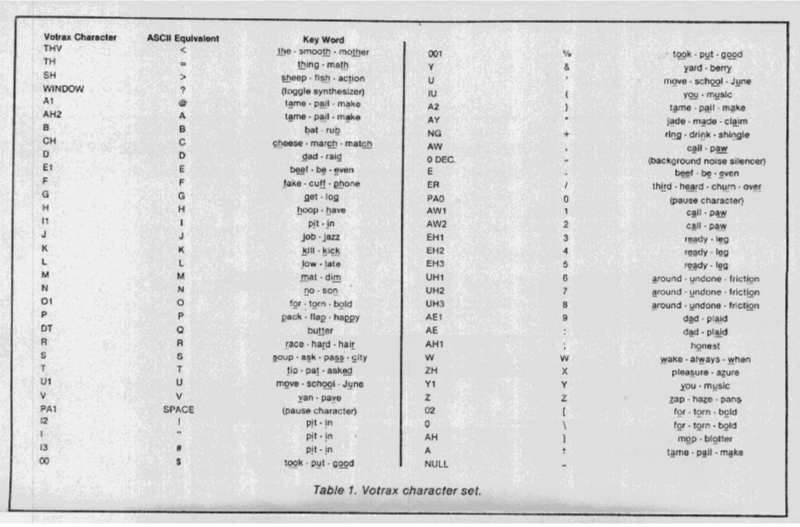
Such an approach, in contrast to the ROMed vocabulary techniques of other units, such as Ti's Speak and Spell module, allows unlimited vocabulary production in any language. This particular model is capable of producing 62 different phonemes, which, if properly sequenced, can produce nearly any word In the English language with reasonable clarity. Synthesis of many foreign words and phrases is also possible.

The Votrax hardware does most of the work, and so you need little overhead in the way of computer memory or software. In fact, voicing words through BASIC can be as simple as outputting their phonetic spellings using simple PRINT statements.

The input of the synthesizing unit contains a character FIFO buffer, which has been memory-mapped into the last 32 locations of the TRS-80 video display refresh memory. Phoneme characters shifted Into this buffer from the computer determine the specific sounds that the unit voices. The synthesizer takes care of processing the characters In the buffer at the proper rate until it is empty.

Table 1 lists the complete Votrax phoneme character set, along with the corresponding ASCII equivalent character set.

You will notice that some of the vowel phonemes have several listings such as EH1, EH2, EH3. This is to allow for various durations (40 to 160 ms), in this case, of the "ea" sound. The longer duration vowel (EH1) is normally used in monosyllabic words or as the stressed syllable of longer words. Phonetic groups, such as stop plosives, fricatives, nasals and semivowels, and their uses are discussed in a manual that comes wIth the synthesizer. Also included are several examples of phonetic programming in BASIC.



Votrax Character	ASCII Equivalent	Key Word			
THV	<	the - smooth - mother	001	%	took - put - good
TH	=	thing - math	Y	&	yard - berry
SH	>	sheep - fish - action	U	'	move - school - June
WINDOW	?	(toggle synthesizer)	IU	(	you - music
A1	@	tame - pail - make	A2	)	tame - pail - make
AH2	A	tame - pail - make	AY	*	jade - made - claim
B	B	bat - rub	NG	+	ring - drink - shingle
CH	C	cheese - march - match	AW	,	call - paw
D	D	dad - raid	0 DEC.	-	(background noise silencer)
E1	E	beef - be - even	E	.	beef - be - even
F	F	fake - cuff - phone	ER	/	third - heard - churn - over
G	G	get - log	PA0	0	(pause character)
H	H	hoop - have	AW1	1	call - paw
I1	I	pit - in	AW2	2	call - paw
J	J	job - jazz	EH1	3	ready - leg
K	K	kill - kick	EH2	4	ready - leg
L	L	low - late	EH3	5	ready - leg
M	M	mat - dim	UH1	6	around - undone - friction
N	N	no - son	UH2	7	around - undone - friction
O1	O	for - torn - bold	UH3	8	around - undone - friction
P	P	pack - flap - happy	AE1	9	dad - plaid
DT	Q	butter	AE	:	dad - plaid
R	R	race - hard - hair	AH1	;	honest
S	S	soup - ask - pass - city	W	W	wake - always - when
T	T	tip - pat - asked	ZH	X	pleasure - azure
U1	U	move - school - June	Y1	Y	you - music
V	V	van - pave	Z	Z	zap - haze - pans
PA1	SPACE	(pause character)	02	[	for - torn - bold
I2	!	pit - in	0	\	for - torn - bold
I	"	pit - in	AH	]	mop - blotter
I3	#	pit - in	A	↑	tame - pail - make
00	$	took - put - good	NULL	_	

Table 1. Votrax character set.

Interfaclng the Synthesizer

The Votrax synthesizer operates from 110 V ac and comes assembled in its own cabinet. Computer interfacing to the unit Is accomplished through a 40-pin edge card connector on the end of a two-foot ribbon cable. Connection to the synthesizer is most easily done by soldering the computer connections to a piece of suitably cut PC edge card stock from the junk box. The PC board is then slid into the connector as it was in-tended to be used.

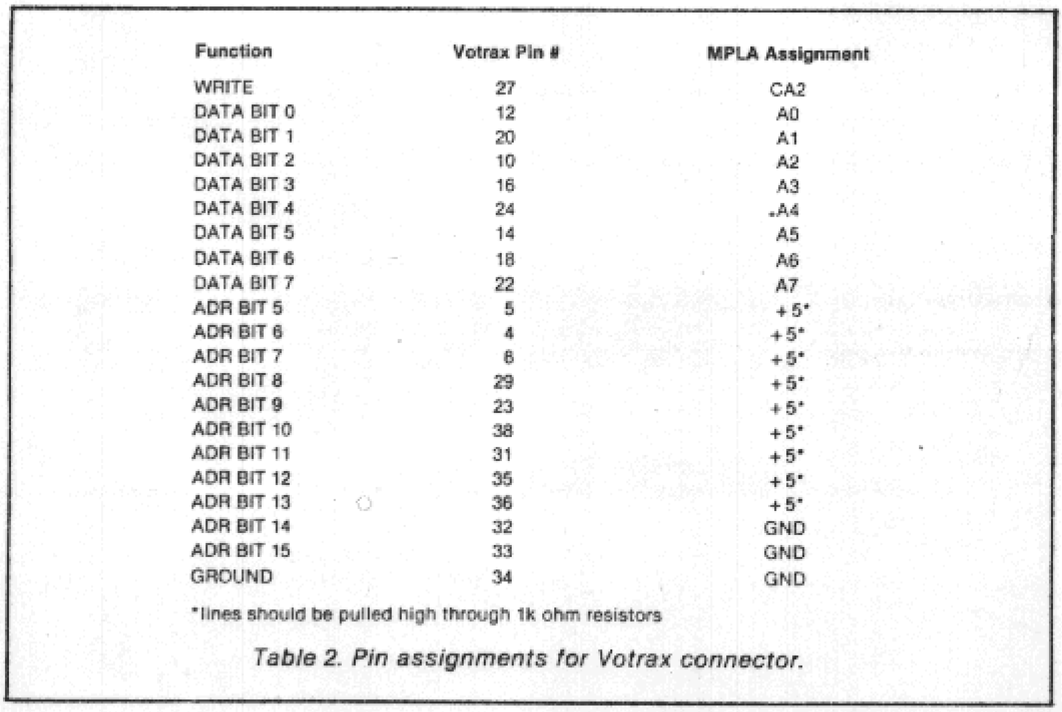
As shown in Fig. 1, the Votrax connector pins are numbered from left to right, with the odd-numbered pins on the top and the even-numbered pins on the bottom. These numbers do not correspond to those of the TRS-80 expansion connector edge card. The numbers referred to in this article are those of the Votrax connector.

Table 2 shows the pin assignments grouped according to function, along with a parallel port assignment for an MPLA card in an SWTP 6800 computer. In this assignment the synthesizer is interfaced to the A side (configured for computer output) of the 6820/21 PIA. Tying the Votrax address bits 5 through 13 high and bits 14 and 15 low correctly addresses the synthesizer so that it can accept data bytes from the PIA. Data is strobed into the FIFO of the synthesizer on the positive edge of a WRITE pulse (500 ns minimum), which must occur after each byte is loaded into the PIA data register. The CA2 line is used for this purpose.

The assembled listing in listing 1 is a suitable driver routine that properly configures an MPLA card in slot 3. Phonemes may be sent to the synthesizer by loading the data register of the PIA with the appropriate ASCII equivalents, using simple LDA and STA instructions. This driver may be used alone or called within a BASIC interpreter, using the CALL or USER commands.

If you are using Computerware's random disk BASIC (BBAS version 9D), you may output phonemes directly with PRINT statements (i.e., PRINT #3 if using slot 3) by poking locations $0218 and $*0219 within the interpreter with NOP's at the beginning of your program. This bypasses the normal handshaking and prevents the computer from qetting into an endless loop while waiting for an acknowledge signal from the synthesizer. (POKE locations $0227 and $0228 with NOP's in Smoke Signal's BASIC version 5.9.)

Because Radio Shack chose not to include any handshaking with the synthesizer, you must not output too many characters in succession or the 32 character buffer might overflow and cause data loss. Since the phonemes last from 40 to 160 ms, a full buffer may take from one to five seconds to empty. This is best handled experimentally by inserting software delays between words or phrases as required to prevent overspill. Simple FOR-NEXT delay loops may be used in BASIC.



Function	Votrax Pin #	MPLA Assignment
WRITE	27	CA2
DATA BIT 0	12	A0
DATA BIT 1	20	A1
DATA BIT 2	10	A2
DATA BIT 3	16	A3
DATA BIT 4	24	.A4
DATA BIT 5	14	A5
DATA BIT 6	18	A6
DATA BIT 7	22	A7
ADR BIT 5	5	+5*
ADR BIT 6	4	+5*
ADR BIT 7	6	+5*
ADR BIT 8	29	+5*
ADR BIT 9	23	+5*
ADR BIT 10	38	+5*
ADR BIT 11	31	+5*
ADR BIT 12	35	+5*
ADR BIT 13	36	+5*
ADR BIT 14	32	GND
ADR BIT 15	33	GND
GROUND	34	GND

*lines should be pulled high through 1k ohm resistors

Table 2. Pin assignments for Votrax connector.

Microcomputing, December 1980 121

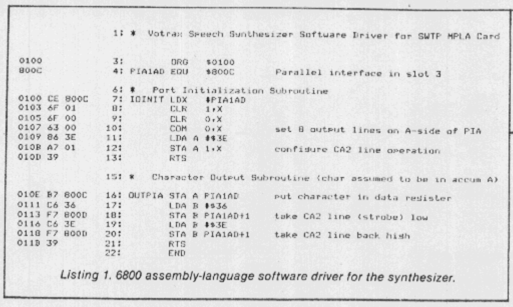
Using the Synthesiser

The easiest way to use the synthesizer Is to install a suitable driver routine, such as the one in Listing 1, and then communicate with the unit In BASIC. The synthesizer is toggled on and off by outputting a "?". An LED on the front of the unit indicates when it has been toggled on. After toggling the unit on, it Is Important to first send a pause phoneme (space), which synchronizes the synthesizer after it has been idle. Since the unit does not have a suitable power-on reset, this may also have to be done to keep it from jabbering when power is first applied.

Because the synthesiser continually voices the last phoneme residing in its buffer, you should also transmit a final pause at the end of each data block to silence it.

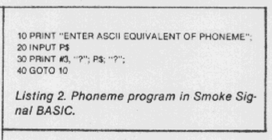
A simple program recommended by Radio Shack (and converted here to Smoke Signal BASIC) to familiarize the user with some of the phonemes the unit is capable of producing is shown in Listing 2. The subroutine in Listing 3 may be used to output the phonetic string K$.

This program pulls phonemes one at a time from the phonetic string K$ and sends them to the synthesizer. For example, to output the word "ready," K$ is set equal to 'R35D8' by a previous statement.



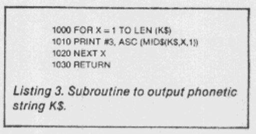
```
                  *  Votrax Speech Synthesizer Software Driver for SWTP MPLA Card

0100                       ORG    $0100
800C              PIA1AD EQU    $800C       Parallel interface in slot 3

                  *   Port Initialization Subroutine
0100 CE 800C      IOINIT LDX    #PIA1AD
0103 6F 01               CLR    1,X
0105 6F 00               CLR    0,X
0107 63 00               COM    0,X         set 8 output lines on A-side of PIA
0109 86 3E               LDA A #$3E
010B A7 01               STA A 1,X          configure CA2 line operation
010D 39                  RTS

                  *   Character Output Subroutine (char assumed to be in accum A)

010E B7 800C      OUTPIA STA A PIA1AD       put character in data register
0111 C6 36               LDA B #$36
0113 F7 800D             STA B PIA1AD+1     take CA2 line (strobe) low
0116 C6 3E               LDA B #$3E
0118 F7 800D             STA B PIA1AD+1     take CA2 line back high
011B 39                  RTS
                         END
```

Listing 1. 6800 assembly-language software driver for the synthesizer.



```
10 PRINT "ENTER ASCII EQUIVALENT OF PHONEME";
20 INPUT P$
30 PRINT #3, "?"; P$; "?";
40 GOTO 10
```

Listing 2. Phoneme program in Smoke Signal BASIC.



```
1000 FOR X = 1 TO LEN (K$)
1010 PRINT #3, ASC (MID$(K$,X,1))
1020 NEXT X
1030 RETURN
```

Listing 3. Subroutine to output phonetic string K$.