DAVID A. WARD

Computerized voice synthesizers are turning up everywhere. Perhaps you've heard one at the grocery store check-out stand, in an automobile, or from an educational toy. Other uses include text-to-speech converters for the visually impaired, talking clocks, calculators, radar detectors, chess and other games, blood-sugar and pressure-monitoring devices, and automotive test equipment.

It's a lot of fun experimenting with voice synthesizers; in fact, the author has built and experimented with four different voice synthesizer IC's, and has listened to at least ten different synthesizers in all.

So that you can share in the fun too, we'll present theory and construction details of a stored-word speech system that you can connect to any personal computer having a parallel printer port. A simple BASIC program then uses LPRINT statements to create speech output. A number of terms relevant to electronics are included: ampere, kilo, milli, volt, circuit, connect, farad, hertz, meg, mega, micro, nano, ohms, pico, as well as letters of the alphabet, numbers, and numerous others. The project can be built for about $75.

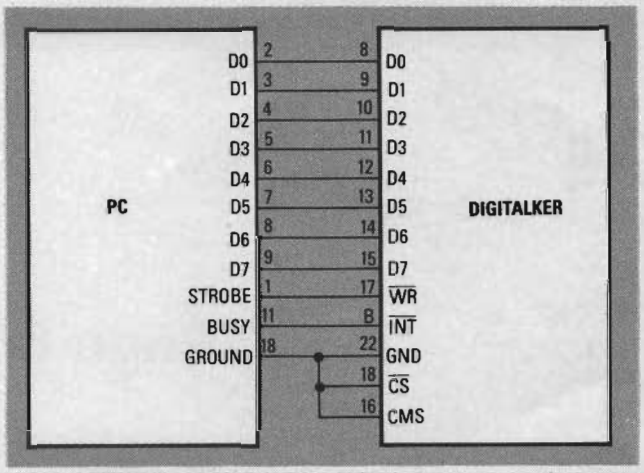


FIG. 1—THE PARALLEL PRINTER PORT of any personal computer can drive the Digitalker.

Speech systems

Most speech synthesis systems operate in one of two ways: the stored-speech method or the allophone method. The allophone method uses *allophones,* little chunks of sound that can be combined to form words. The stored-word system stores entire words and phrases.

Each system has advantages and disadvantages. Allophone synthesis can offer an unlimited vocabulary and yet require very little memory. However, allophone speech synthesis is usually artificial sounding, monotone, and difficult for the untrained ear to understand. Probably the best application for allophone synthesis is in converting text to speech. Text-to-speech conversion can be a great aid for the visually impaired, allowing them to operate word processors and other computer programs.

By contrast, a stored-word synthesizer can offer excellent speech quality with intonation or feeling. However, a stored-word system requires tremendous amounts of memory for just a few minutes of speech. Typically, that limits a stored-word system to a vocabulary of several dozen words. The best application for a stored-word synthesizer is one that requires the clearest possible speech and a limited vocabulary, such as in an automobile, or a supermarket check-out stand. A stored-word

TABLE 1—WORD LIST (SSR1 AND SSR2)

THIS IS
DIGITALKER 0
ONE 1
TWO 2
THREE 3
FOUR 4
FIVE 5
SIX 6
SEVEN 7
EIGHT 8
NINE 9
TEN 10
ELEVEN 11
TWELVE 12
THIRTEEN 13
FOURTEEN 14
FIFTEEN 15
SIXTEEN 16
SEVENTEEN 17
EIGHTEEN 18
NINETEEN 19
TWENTY 20
THIRTY 21
FORTY 22
FIFTY 23
SIXTY 24
SEVENTY 25
EIGHTY 26
NINETY 27
HUNDRED 28
THOUSAND 29
MILLION 30
ZERO 31
A 32
B 33
C 34
D 35
E 36
F 37
G 38
H 39
I 40
J 41
K 42
L 43
M 44
N 45
O 46
P 47
Q 48
R 49
S 50
T 51
U 52
V 53
W 54
X 55
Y 56
Z 57
AGAIN 58
AMPERE 59
AND 60
AT 61
CANCEL 62
CASE 63
CENT 64
400 HZ TONE 65
80 HZ TONE 66
20 mS SILENCE 67
40 mS SILENCE 68
80 mS SILENCE 69
160 mS SILENCE 70
320 mS SILENCE 71
CENTI 72
CHECK 73
COMMA 74
CONTROL 75
DANGER 76
DEGREE 77
DOLLAR 78
DOWN 79
EQUAL 80
ERROR 81
FEET 82
FLOW 83
FUEL 84
GALLON 85
GO 86
GRAM 87
GREAT 88
GREATER 89
HAVE 90
HIGH 91
HIGHER 92
HOUR 93
IN 94
INCHES 95
IS 96
IT 97
KILO 98
LEFT 99
LESS 100
LESSER 101
LIMIT 102
LOW 103
LOWER 104
MARK 105
METER 106
MILE 107
MILLI 108
MINUS 109
MINUTE 110
NEAR 111
NUMBER 112
OF 113
OFF 114
ON 115
OUT 116
OVER 117
PARENTHESIS 118
PERCENT 119
PLEASE 120
PLUS 121
POINT 122
POUND 123
PULSES 124
RATE 125
RE 126
READY 127
RIGHT 128
SS (NOTE 1) 129
SECOND 130
SET 131
SPACE 132
SPEED 133
STAR 134
START 135
STOP 136
THAN 137
THE 138
TIME 139
TRY 140
UP 141
VOLT 142
WEIGHT 143

NOTE 1: "SS" (#129) can be used to make singular words plural

TABLE 2—WORD LIST SSR5 AND SSR6

ABORT 0
ADD 1
ADJUST 2
ALARM 3
ALERT 4
ALL 5
ASK 6
ASSISTANCE 7
ATTENTION 8
BRAKE 9
BUTTON 10
BUY 11
CALL 12
CAUTION 13
CHANGE 14
CIRCUIT 15
CLEAR 16
CLOSE 17
COMPLETE 18
CONNECT 19
CONTINUE 20
COPY 21
CORRECT 22
DATE 23
DAY 24
DECREASE 25
DEPOSIT 26
DIAL 27
DIVIDE 28
DOOR 29
EAST 30
ED (NOTE 1) 31
ED (NOTE 1) 32
ED (NOTE 1) 33
ED (NOTE 1) 34
EMERGENCY 35
END 36
ENTER 37
ENTRY 38
ER 39
EVACUATE 40
EXIT 41
FAIL 42
FAILURE 43
FARAD 44
FAST 45
FASTER 46
FIFTH 47
FIRE 48
FIRST 49
FLOOR 50
FORWARD 51
FROM 52
GAS 53
GET 54
GOING 55
HALF 56
HELLO 57
HELP 58
HERTZ 59
HOLD 60
INCORRECT 61
INCREASE 62
INTRUDER 63
JUST 64
KEY 65
LEVEL 66
LOAD 67
LOCK 68
MEG 69
MEGA 70
MICRO 71
MORE 72
MOVE 73
NANO 74
NEED 75
NEXT 76
NO 77
NORMAL 78
NORTH 79
NOT 80
NOTICE 81
OHMS 82
ONWARD 83
OPEN 84
OPERATOR 85
OR 86
PASS 87
PER 88
PICO 89
PLACE 90
PRESS 91
PRESSURE 92
QUARTER 93
RANGE 94
REACH 95
RECEIVE 96
RECORD 97
REPLACE 98
REVERSE 99
ROOM 100
SAFE 101
SECURE 102
SELECT 103
SEND 104
SERVICE 105
SIDE 106
SLOW 107
SLOWER 108
SMOKE 109
SOUTH 110
STATION 111
SWITCH 112
SYSTEM 113
TEST 114
TH (NOTE 2) 115
THANK 116
THIRD 117
THIS 118
TOTAL 119
TURN 120
USE 121
UTH (NOTE 3) 122
WAITING 123
WARNING 124
WATER 125
WEST 126
SWITCH 127
WINDOW 128
YES 129
ZONE 130

NOTE 1: "ED's" 31 and 32 work best with words that end with "T" or "D". "ED" 34 works best with words that end with soft sounds.

NOTE 2: "TH" (#115) can be added to words like; six, seven, and eight to make sixth, seventh, and eighth etc.

NOTE 3: "UTH" (#122) can be added to twenty, thirty, and forty to make twentieth, thirtieth, and fortieth, etc.

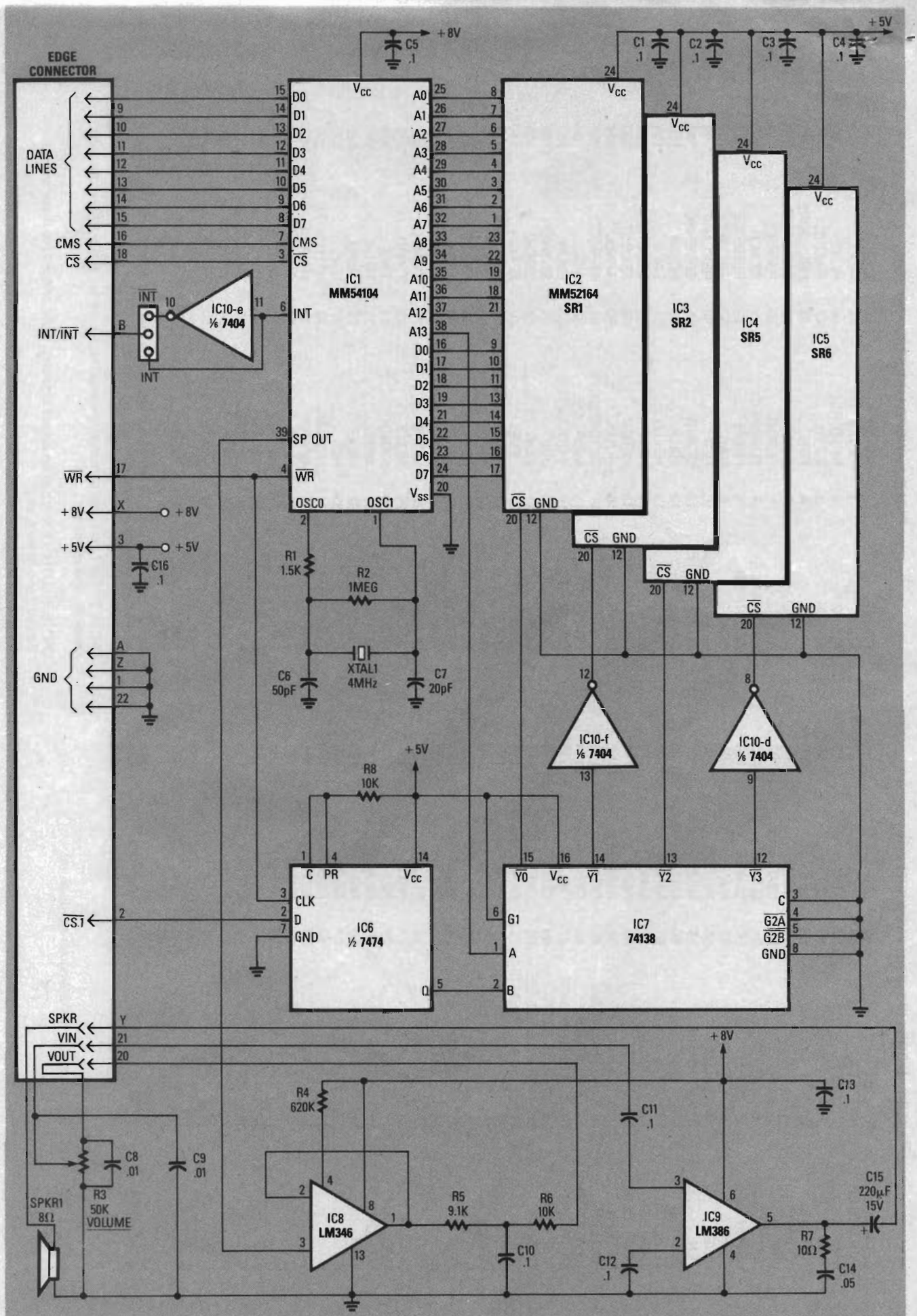


FIG. 2—SCHEMATIC DIAGRAM OF THE DIGITALKER. The speech processor (IC1) reads data from the ROM's (IC2–IC5) and delivers speech output via pin 39.

synthesizer is useless for text-to-speech conversion because of the large amount of memory that would be required.

The Digitalker

National Semiconductor's Digitalker is a stored-word speech synthesis system that produces an exceptionally clear "voice." In fact, the Digitalker's quality exceeds Texas Instrument's *Speak & Spell* speech synthesizer. The Digitalker's voice has intonation or feeling, is not monotone, and even uses a female voice for the phrase "This is Digitalker."

The MM54104 SPC (*S*peech *P*rocessor *C*hip) is the heart of the Digitalker system. It's a 40-pin IC having 8 data lines (pins 8–15) that can be programmed manually with switches, or by connecting the device to a computer. For best results, a computer should be used to control the SPC so that sentences can be formed by stringing words together rapidly.

The SPC also has 14 ROM address lines (A0–A13, pins 25–38) that are to address ROM's containing speech data. Through those 14 address lines, the SPC can directly access 128K bits of speech data, which is good for about one minute of continuous speech. The SPC receives its data from the ROM's through eight data lines (pins 16–19 and pins 21–24). A number of other lines (pins 3, 4, and 7) are used for handshaking with a host computer, for connecting an external crystal oscillator (pins 1 and 2), and for speech output (pin 39)—which is connected to a filter and an audio amplifier. For more information on the SPC, see National's 1982 Linear Databook.

The right words

One key to a good stored-word speech-synthesis system is to choose the right words to store, convert them from an analog source, and then compress them into digital data suitable for the SPC.

National Semiconductor will convert analog tapes into custom digital data for customers, but that's an expensive proposition for hobbyists. However, the company has developed four general-purpose 64K-bit ROM's that contain data for 273 words, phrases, tones, and pauses. National's Linear databooks list several different ROM sets, but the SSR1, SSR2, SSR5, and SSR6 provide the best selection of words and are easy to obtain. The four ROM's together contain nearly two minutes of continuous speech; the words contained in each ROM set are shown in Tables 1 and 2.

Hooking it up

As shown in Fig. 1, the simplest way to use the Digitalker is to connect it to your computer's printer port. There are several advantages to doing so. First, handshaking between the computer and the Digitalker is automatic, so it isn't necessary to place timing loops in the software.

Second, most printer ports have a $\overline{\text{STROBE}}$ line that goes low when data at the port is valid. The strobe line can be connected to the SPC's $\overline{\text{WR}}$ line. When it is asserted, the SPC reads the ROM data for the selected word over its eight data lines (D0–D8), and then delivers the word to the audio output (pin 39).

The SPC's INTR line (pin 6) goes high after the entire word has been pronounced. By connecting the INTR line (or, if necessary, the inverted $\overline{\text{INTR}}$) to the printer port's BUSY input,

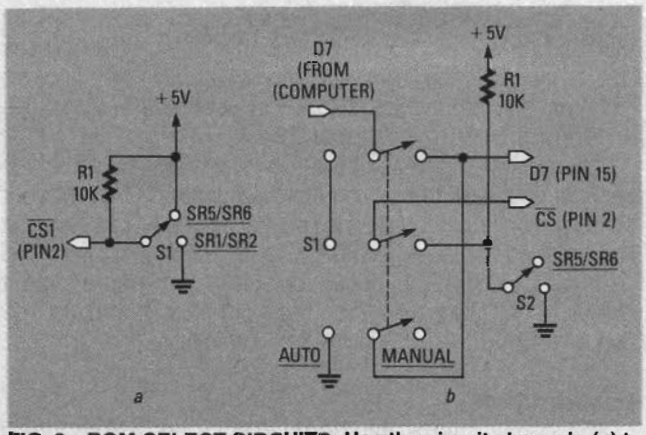


FIG. 3—ROM-SELECT CIRCUITS: Use the circuit shown in (*a*) to select between ROM sets manually. The circuit shown in (*b*) allows manual or automatic computer control, but only the first 128 words and phrases are accessible in the auto mode.

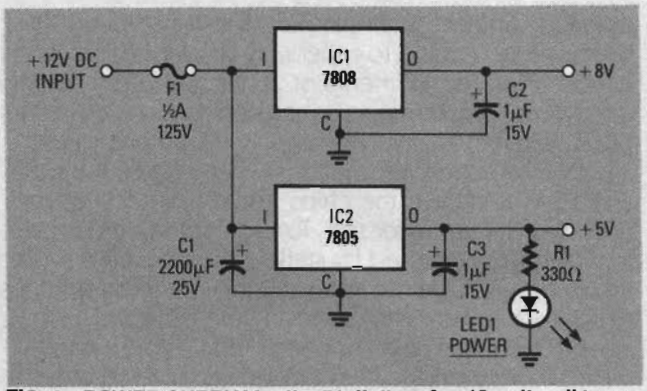


FIG. 4—POWER SUPPLY for the Digitalker. A +12-volt wall transformer provides the raw DC input.

the host computer will wait until each word has been spoken before sending more data to to the SPC.

Two SPC pins provide options. First, $\overline{\text{CS}}$ is the chip-select line;it must be grounded momentarily when the computer addresses the SPC. $\overline{\text{CS}}$ is provided to allow the SPC to share the data bus with other devices.

Second, CMS (command select) resets the interrupt and starts a speech sequence when it is low, and only resets the interrupt when it is high.

The PC board layout brings both $\overline{\text{CS}}$ and CMS out to the edge connector. For normal operation from a parallel-printer port, it's most convenient to ground both pins at the edge-card connector.

Now let's look at the circuit, shown in Fig. 2. The SPC's speech output drives IC8, which buffers the audio signal and drives a volume control. Final audio output is provided by IC9.

Flip-flop IC6 and 3-to-8 line decoder IC7 select the speech ROM's, depending on whether SPC address line AD13 is high or low, and on the state of the $\overline{\text{CS1}}$ signal (edge connector pin 2). AD13 picks the high or low ROM of a pair, and $\overline{\text{CS1}}$ picks one pair or the other.

There are several ways to select which ROM pair you want to use. If you have an extra output bit available on your PC (perhaps a bit from a second parallel port), you can program $\overline{\text{CS1}}$ directly. Otherwise, you can use a manual switch, as shown in Fig. 3-*a*.

A combination approach is shown in Fig. 3-*b*. With switch S1 in the Manual position, you can use S2 to switch

between ROM's. But with S1 in the Auto position, you can switch between ROM's using a single eight-bit port. The upper data bit (D7) provides the switching function, so only the first 128 words (0–127) in each ROM set will be accessible using that approach.

The power-supply schematic is shown in Fig. 4. An inexpensive wall transformer provides the raw DC power. Voltage regulators inside the project's cabinet provide the required voltages: +5-volts DC for the digital circuits, and +8-volts DC for the audio circuitry. The entire circuit draws about 300 mA when the volume is turned up, so use a +12-volt DC, 500-mA power supply.

Construction

PC board patterns are shown in PC Service. An etched and drilled PC board is also available from the source given in the Parts List. Figure 5 shows how the parts are mounted on the board. **Note:** six jumper wires must be soldered to the circuit board before the IC sockets are installed. An additional jumper must be soldered from the center INT terminal to either INT or $\overline{\text{INT}}$, depending on the handshaking requirements of your computer's parallel port. Most computers use an active-high BUSY signal, so try the INT setting first if you're not sure which one to use.

Observe normal precautions when handling the SPC and ROM IC's.Leave the chips in their protective "rugs" until they are ready for use. To protect the components against damage caused by static electricity, make sure to ground yourself before removing the IC's from from their rugs, or when handling or moving the PC board.

After mounting all components, check your work carefully for solder bridges and cold joints. Fix any problems before applying power to the board.

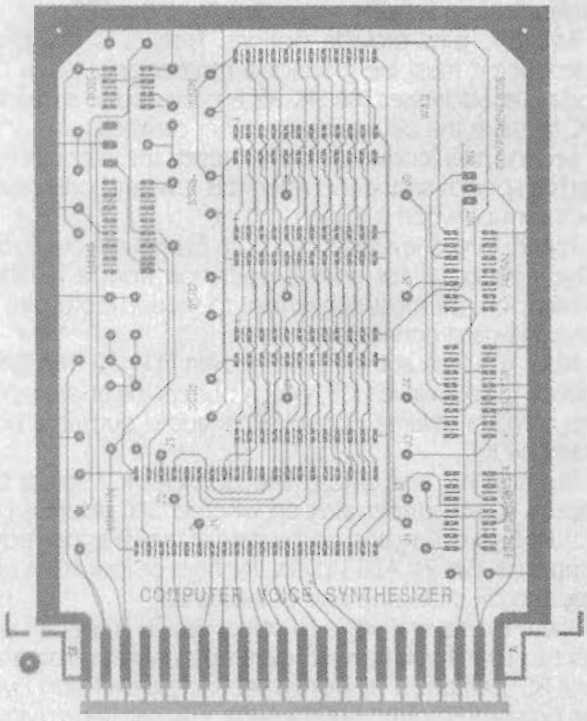


FIG. 5—PARTS LAYOUT: Note that six jumpers must be installed on the component side of the board before installing the IC sockets. (Sockets mount over five of the six jumpers.)

DIGITALKER PARTS LIST

All resistors are ¼-watt, 5% unless otherwise noted.
R1—1500 ohms
R2—1 megohm
R3—50,000 ohms, potentiometer
R4—620,000 ohms
R5—9100 ohms
R6, R8—10,000 ohms
R7—10 ohms
All capacitors are rated 15 volts or higher
C1–C5, C10–C13, C16—0.1 μF, ceramic disc
C6—50 pF, ceramic disc
C7—20 pF, ceramic disc
C8, C9—0.01 μF, ceramic disc
C14—0.05 μF, ceramic disc
C15—220 μF, 15 volts, electrolytic
Semiconductors
IC1—MM54104, speech processor
IC2—MM52164-SSR1, speech ROM
IC3—MM52164-SSR2, speech ROM
IC4—MM52164-SSR5, speech ROM
IC5—MM52164-SSR6, speech ROM
IC6—7474, dual D flip-flop
IC7—74138, 3-to-8 line decoder
IC8—LM346, programmable op-amp
IC9—LM386, audio power amplifiere
IC10—7404, hex inverter
Other components
XTAL1—4.00 MHz crystal

POWER SUPPLY PARTS LIST

R1—330 ohms
IC1—7808 8-volt regulator
IC2—7805 5-volt regulator
C1—2200 μF, 25 volts, electrolytic
C2, C3—1 μF, 15 volts, tantalum
F1—fuse, 0.5 amp, 125 volts
LED1—light-emitting diode
Note: An etched and drilled PC board is available for $15.95 from David A. Ward, 2261 W. Skyview, Cedar City, UT 84720-2233. All orders add $2.00 shipping and handling; Utah residents add 6% sales tax.

LISTING 1

```
10 REM This program will make the
20 REM Digitalker pronounce all words
30 REM in SSR1 and SSR2 (CS1 is low)
40 FOR X = 0 to 143
50 LPRINT CHR$(X);
60 NEXT X
70 END
```

LISTING 2

```
10 REM This program will make the
20 REM Digitalker pronounce all words
30 REM in SSR5 and SSR6 (CS1 is high)
40 FOR X=0 to 130
50 LPRINT CHR$(X);
60 NEXT X
70 END
```

LISTING 3

```
10 REM REAL TIME CLOCK PROGRAM
20 CLS
30 PRINT"HOW OFTEN DO YOU WANT THE TIME ANNOUNCED?"
40 PRINT:PRINT
50 PRINT"ENTER 1 FOR 1 MINUTE INTERVALS..."
60 PRINT"ENTER 5 FOR 5 MINUTE INTERVALS..."
70 PRINT"ENTER 30 FOR 30 MINUTE INTERVALS..."
80 INPUT"",I
90 TIME$=TIME$
100 T$=LEFT$(TIME$,2)
110 T1$=MID$(TIME$,4,2)
120 H$=LEFT$(T$,1)
130 H1$=RIGHT$(T$,1)
140 H=ASC(H$)
150 H1=ASC(H1$)
160 H=H-48
170 H1=H1-48
180 H=H*10
190 HT=H+H1
200 IF HT>12 THEN HT=HT-12:P=47:GOTO 220
210 P=32
220 IF HT=12 THEN P=47
230 IF HT=0 THEN HT=12:P=32
240 M$=LEFT$(T1$,1)
250 M1$=RIGHT$(T1$,1)
260 M=ASC(M$)
270 M1=ASC(M1$)
280 M=M-48
290 M1=M1-48
300 IF M=0 AND M1=0 THEN M=68:M1=68
310 IF M=0 AND M1>0 THEN M=46
320 IF M=1 AND M1=0 THEN M=10:M1=68
330 IF M=1 AND M1=1 THEN M=11:M1=68
340 IF M=1 AND M1=2 THEN M=12:M1=68
350 IF M=1 AND M1=3 THEN M=13:M1=68
360 IF M=1 AND M1=4 THEN M=14:M1=68
370 IF M=1 AND M1=5 THEN M=15:M1=68
380 IF M=1 AND M1=6 THEN M=16:M1=68
390 IF M=1 AND M1=7 THEN M=17:M1=68
400 IF M=1 AND M1=8 THEN M=18:M1=68
410 IF M=1 AND M1=9 THEN M=19:M1=68
420 IF M=2 THEN M=20
430 IF M=3 THEN M=21
440 IF M=4 THEN M=22
450 IF M=5 THEN M=23
460 IF M1=0 THEN M1=68
470LPRINT CHR$(0);CHR$(138);CHR$(67);CHR$(139);CHR$(67);
CHR$(96);CHR$(71);CHR$(HT);CHR$(69);CHR$(M);CHR$(M1);
CHR$(71);CHR$(P);CHR$(44);CHR$(71);CHR$(71);
480 PRINT TIME$
490 GOSUB 510
500 GOTO 90
510 IF I=1 THEN I=60
520 IF I=5 THEN I=300
530 IF I=10 THEN I=600
540 IF I=30 THEN I=1800
550 Z=TIMER
560 Y=TIMER
570 IF Y-Z<I THEN 560
580 RETURN
```

Making the connection

Connecting the Digitalker to your computer is as simple as plugging it into your computer's parallel printer port. For testing purposes, wire a ROM-select switch as shown in Fig. 3-*a*.

It's easy to program the Digitalker. For example, simply by typing

```
LPRINT CHR$(0);
```

the Digitalker will say the phrase "This is Digitalker" if $\overline{CS1}$ is low, or "abort" if $\overline{CS1}$ is high.

Listing 1 and Listing 2 are test programs that sequentially pronounces all words contained in the selected ROM set. Both programs were written in GW-BASIC; they were tested on a Kaypro PC.

More sophisticated applications are not difficult. For example, the author has written BASIC programs that do the following; announce the time from the computer's real-time clock, pronounce the corresponding letter of the alphabet as a key is typed (great for a small child learning his ABC's), pronounce phone numbers as names are typed in, and prompt the user for input in various programs. The talking clock program is shown in Listing 3.

There are a couple of things to be aware of when programming the Digitalker. First, addressing a word with a number higher than that listed in the word lists will produce unintelligible speech, but will not damage the SPC or ROM chips. Second, the semicolons following the LPRINT statements are essential. If they are not present the Digitalker will pronounce *thirteen* and then *ten* after each word is spoken. That occurs because an ASCII 13 is a carriage return, and an ASCII 10 is a linefeed. The semicolon (;) eliminates the carriage return and linefeed.

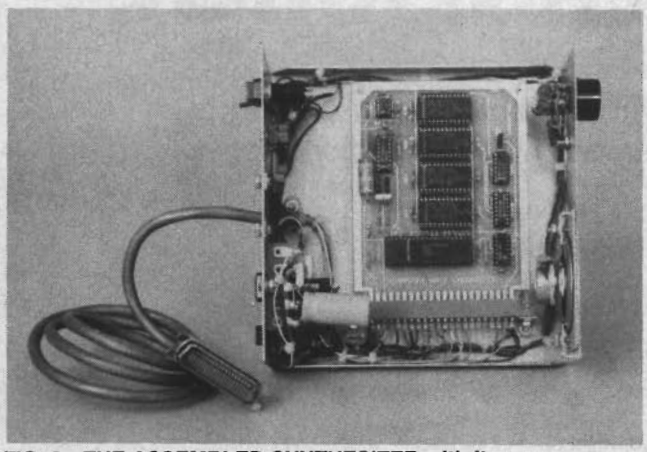
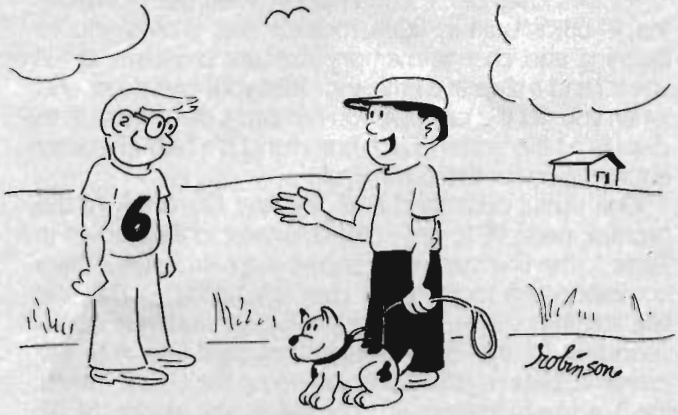


FIG. 6—THE ASSEMBLED SYNTHESIZER with its cover removed.

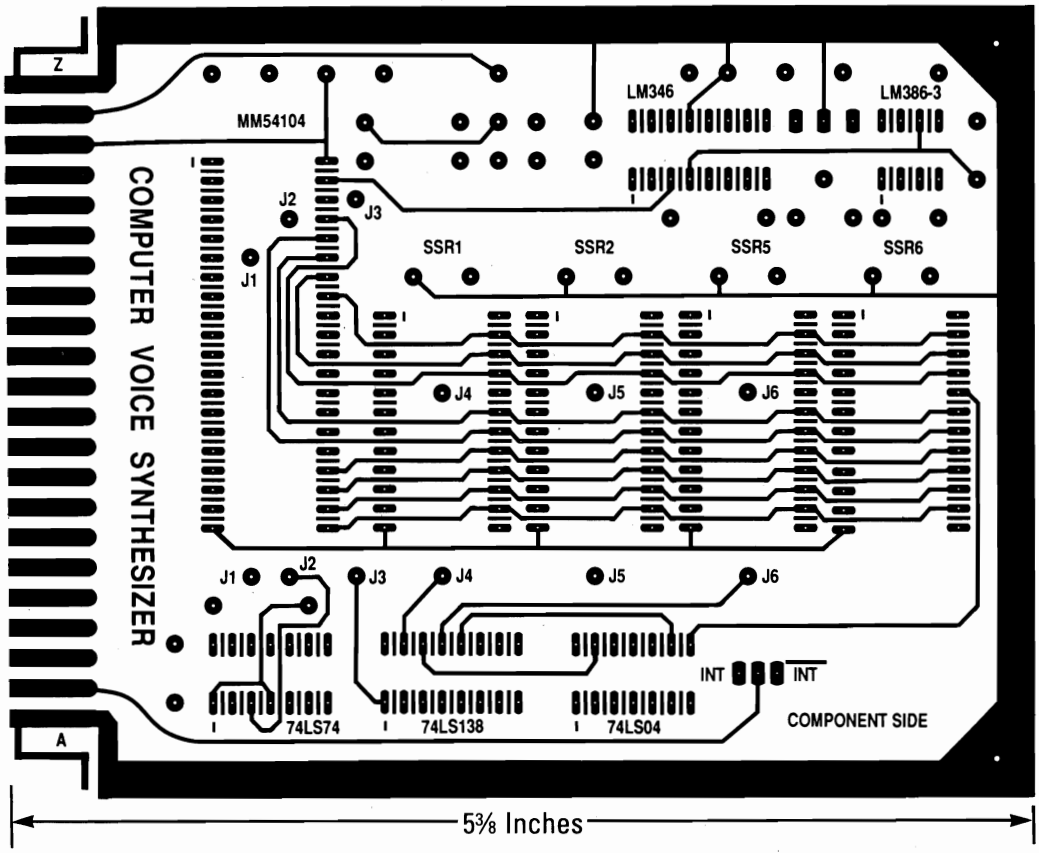
Applications ideas

Computer voice synthesis can be a very natural way for computers to communicate with people. For example, a synthesizer could be used to warn a pilot that the plane's altitude is critically low, or that the fuel level is low. A visually impaired person could compose documents with a word processor, or compute math problems with a calculator.

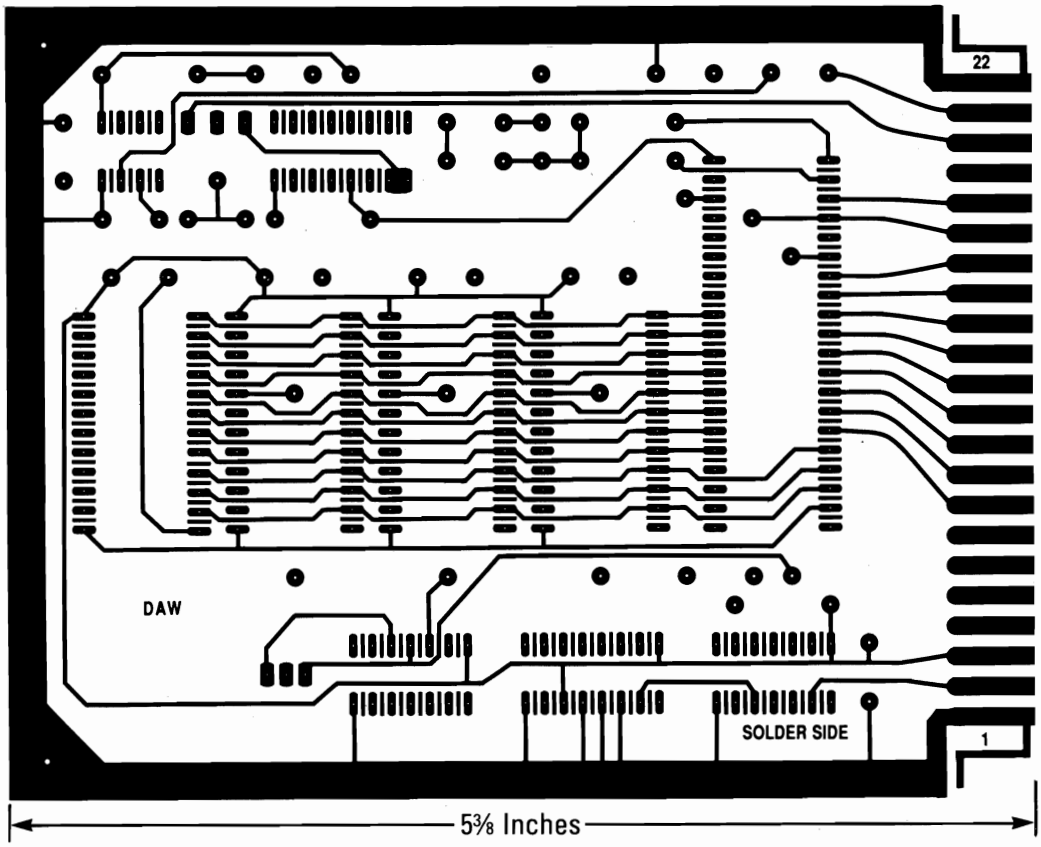
Undoubtedly, there are many other uses for computerized voice synthesis in cash registers, automatic teller machines, emergency warning systems, automobiles, telephone systems, etc. Have fun finding them!



"The man at the pet shop said he can store up to 20 separate commands."



THE COMPONENT SIDE of the voice-synthesizer board.



THE SOLDER SIDE of the voice-synthesizer board.