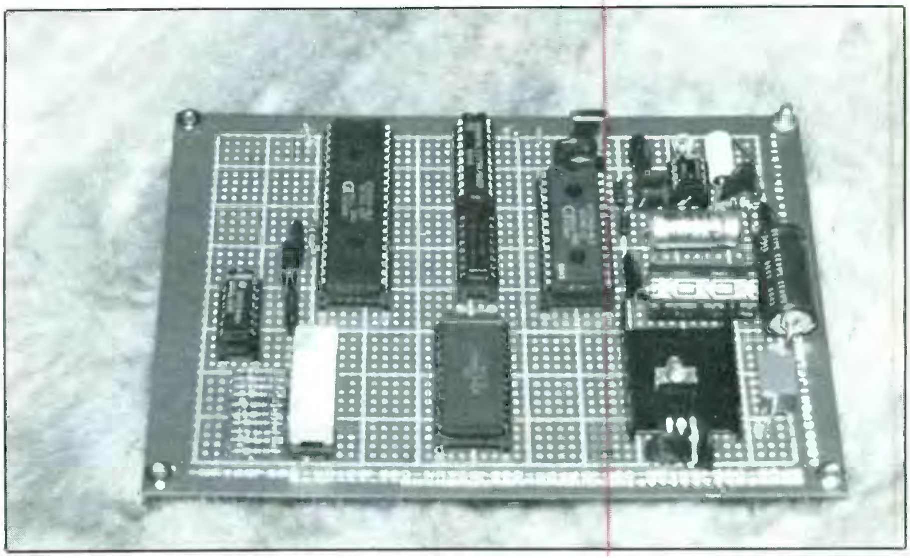
A General-Purpose Speech Synthesizer

This versatile ASCII-to-speech processor can be used with a variety of devices to generate human-sounding speech

By Barry L. Ives

Artifically generated—or so-called "synthesized"—speech has found applications in many areas as a useful way of providing machine-to-human feedback. One obvious application of a speech synthesizer is as a device to assist sightless and/or mute people to communicate with others. In other applications, the speech synthesizer can be used to vocalize instructions to a computer operator in situations where it is not convenient to display characters on a video screen or where the operator is unable to keep a continuous eye on the screen. It can also be used as a teaching/learning tool for children and other individuals to learn spelling, mathematics and even how to read. One popular use is in providing aural feedback in interactive video games. In fact, the uses to which a speech synthesizer like the one to be described are limited only by the imagination of the user.



Our speech synthesizer project accepts ASCII test information from any computer and converts it into vocalized speech. Because it translates the text words fed to it into sounds phonetically with an accuracy of 85 percent or more, the synthesizer's vocabulary is virtually unlimited. For maximum versatility, the project can be configured to accept serial or parallel ASCII data fed to it at any rate between 50 and 9,600 baud. It accommodates both TTL and RS-232C signal levels and any combination of bits and parity and can be set up to handle the user's choice of hardware or software handshaking, with a three-line or X-line serial interface. If you wish, you can even use an ASCII-encoded keyboard to feed data to the synthesizer in place of a computer.

About the Circuit

The speech synthesizer is built around the General Instruments CTS256A-AL2 code-to-speech converter processor, which is basically a dedicated microprocessor, and

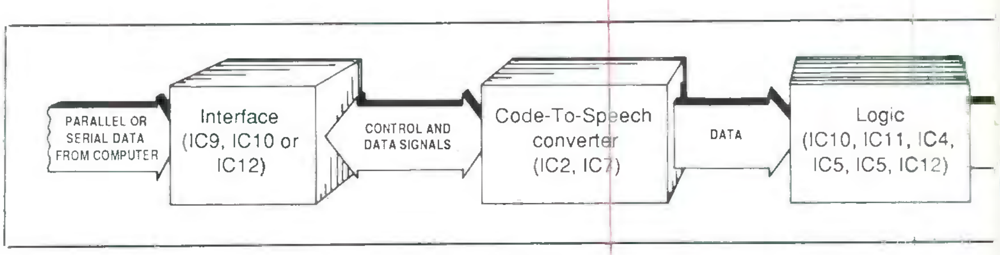


Fig. 1. Block diagram of overall speech synthesizer,

Table 1. Standard RS-232C Signals (partial list)

DB-25 Pin No.	Abbreviation	Name	Direction
1	P. GND	Chassis ground	Both
2	TX	Transmitted data	To modem
3	RX	Received data	From modem
4	RTS	Request to send	To modem
5	CTS	Clear to send	From modem
6	DSR	Data set ready	From modem
7	S. GND	Signal ground	Both
8	DCD	Data carrier detect	From modem
20	DTR	Data terminal ready	To modem

SPO256A-AL2 speech-processor integrated circuits. Both of these chips are readily available from Radio Shack stores and other sources. The block diagram of the synthesizer, minus its ac-operated power supply, is shown in Fig. 1.

Character data from a computer or other ASCII-encoded source is fed into dedicated 8-bit microprocessor *IC2* in Fig. 2. Upon receipt of the ASCII data, *IC2*'s internal program searches through a series of letter-to-sound rules to locate the proper allophone addresses to feed to speech-processor chip *IC3*. The latter chip uses the addresses fed to it to form the required phonetic sounds by means of an internal programmable digital filter and pulse-width modulator configured to imitate the sounds of a human voice.

After passing through an external two-pole filter, the sound signal from *IC2* is amplified by audio amplifier chip *IC8*, which then drives the speaker to produce the vocalized sounds of the typed-in data. (For more information on allophone speech synthesis, see "This is Your Computer Speaking," *Modern Electronics*, June 1986.)

Figure 3 is the schematic diagram of the circuit required to interface the synthesizer to a computer through a *true* RS-232C serial port. As you can see from this interface circuit, you need quad receiver *IC9* and quad transmitter *IC10*. Standard RS-232C uses the signals detailed in Table 1. Some of these signals are for handshaking control to and from the computer and "modem." However, the RX and TX lines carry the actual data.

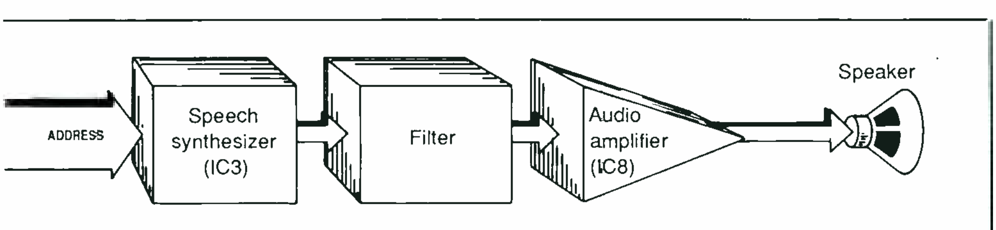
Pin numbers for interface connector *J1* into which the computer/synthesizer link is made are for the standard DB-25 connector commonly used in computer systems. If your computer uses a different connector, refer to the computer's manual to obtain the information needed and adjust accordingly when preparing the linking cable.

Code-to-speech converter *IC2* in Fig. 2 has an input port for ASCII data as well as an output port for software handshaking and a BUSY output line for hardware handshaking. The input port at pin 16 of *IC2* should be connected to the computer's TX data output. The output port at pin 37 of *IC2*, if it is to be used for software XON/XOFF handshaking, should connect to the computer's CTS input. These are the primary signals to and from the speech synthesizer. In the Fig. 3 RS-232C interface adapter circuit, the DCD and DSR lines are pulled high to signal the computer at all times.

If TTL-level signals are used, the connections should be made directly to DB-25 connector *J1* instead of going through the Fig. 3 circuit. The connections shown permit a straight-through interconnecting cable to be used, assuming the computer is configured as a data terminal. (If you plan to use the project in this manner, check the computer's programming manual to be certain that the pin numbers correspond and adjust as necessary. Information on programming the RS-232C port in the computer can also be found in the manual.)

As provided, the CTS256A is programmed to read 7-bit ASCII data with two stop bits and no parity. Baud rate, however, is selectable according to settings of the sections of DIP switch *S3*, as shown in Table 2. If your computer cannot be set up to accommodate these parameters, the optional latch circuit consisting of octal latch *IC11* in Fig. 4 can be added to the basic project to provide additional choices of bits and parity per Table 2. Inclusion of the Fig. 4 programmable parameters circuit requires installation of the jumper wire that connects pin 9 of *IC2* to +5 volts (see Fig. 2).

The parallel interface port circuit shown in Fig. 5 can be used to interface the project with computers that have a parallel but no serial port or simply if you wish to use parallel



including all options but not power supply.

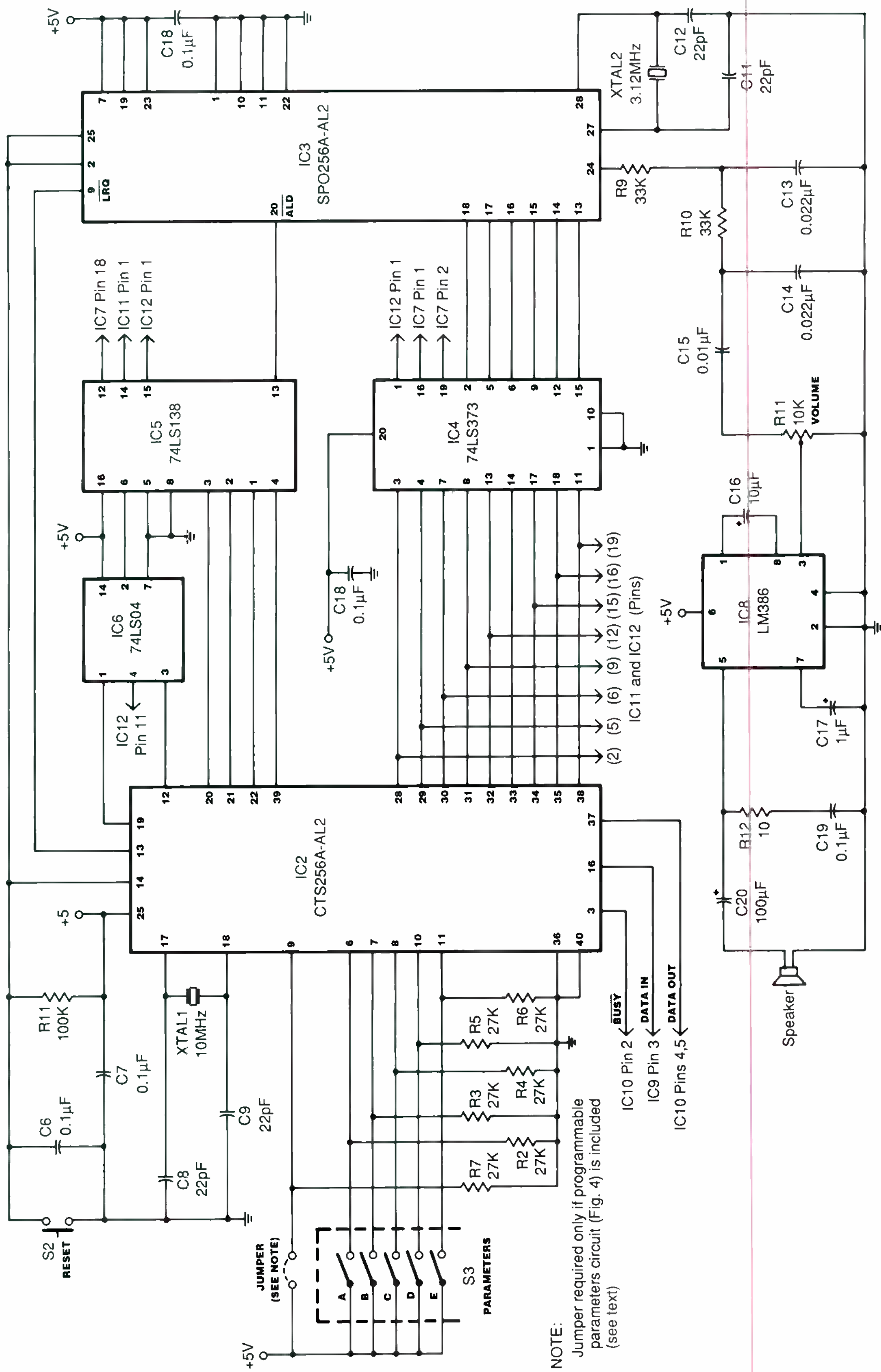


Fig. 2. Basic speech-processing circuitry.

Table 2. Parameters Option Wiring and Setting

S3 Section	To	Settings	
A	IC2, pin6	000 = Parallel	
B	IC2, pin 7	001 = Serial (50 baud)	
C	IC2, pin 8	010 = Serial (110 baud)	
		011 = Serial (300 baud)	
		100 = Serial (1,200 baud)	
		101 = Serial (2,400 baud)	
		110 = Serial (4,800 baud)	
		111 = Serial (9,600 baud)	
	Under "Settings," read A, B, C left to right.		
D	IC2, pin 10	1 = External Ram; 0 = internal RAM	
E	IC2, pin 11	1 = Any delimiter; 0 = Carriage return only	
F	IC11, pin 3	1 = 2 stop bits; 0 = 1 stop bit	
G	IC11, pin 7	1 = Even parity; 0 = odd party	
H	IC11, pin 8	1 = Parity; 0 = No parity	
J	IC11, pin 13	00 = Not used	10 = 7 bits
K	IC11, pin 14	01 = 6 bits	11 = 8 bits
	Under Settings," read J, K left to right.		

feed. This circuit latches eight bits corresponding to an ASCII character into *IC12* when the S̄ (STROBE) line is brought low. The character is then fed into *IC2* (Fig. 2) and RAM memory. Connections can be made through pins 10 through 18 and pin 7 ground of *J1*.

Though *IC2* comes with a 20-byte input buffer and 26-byte output buffer, this is not sufficient storage space to avoid crashing the system as a result of output buffer overflow. Therefore, the external RAM circuit shown in Fig. 6 is included in the basic design of the speech synthesizer to permit a half-screen full of data to be accepted and spoken from beginning to end.

The final circuit that makes up the speech synthesizer is the ac-line-operated power supply shown schematically in Fig. 7. This is a straightforward regulated +5-volt dc power supply. Transformer *T1* steps down the incoming 117 volts ac to 12.6 volts ac, which is then rectified to pulsating dc by filter capacitor *C1*. After this, the dc is regulated to +5 volts by *IC1* and is further filtered to pure dc by *C3*. Light-emitting diode *LED1* serves as a power ON indicator and is an option you can leave out of the circuit, along with *R1*, if you wish to economize.

Construction

Because of the complexity of the wiring required to assemble the speech

PARTS LIST

Semiconductors

LED1—Red panel-mount light-emitting diode
IC1—7805T +5-volt regulator
IC2—CTS256A-AL2 code-to-speech converter
IC3—SPO256A-AL2 speech processor
IC4,IC11—74LS373 octal latch
IC5—74LS138 1-of-8 decoder
IC6—74LS04 hex inverter
IC7—TMS4016-25 or 6116LP-3 static RAM
IC8—LM386 audio amplifier
IC9—MC1489 quad receiver
IC10—MC1488 quad transmitter
IC12—74LS374 octal D flip-flop
RECT1—50-volt, 1-ampere bridge rectifier

Capacitors

C1,C3—1,000-μF, 25-volt electrolytic
C2,C20—100-μF, 15-volt electrolytic
C4 thru C7,C10,C15,C18,C19—0.0-μF ceramic disc
C8,C9,C11,C12,C13,C14—0.022-μF Mylar
C16—10-μF, 15-volt electrolytic
C17—1-μF, 15-volt electrolytic
C21 thru C25—100-pF disc

Resistors (¼-watt, 5% tolerance)

R1—400 ohms
R2 thru R7, R14 thru R18—2,700 ohms
R8—100,000 ohms
R9,R10—33,000 ohms
R12—10 ohms
R13—1,000 ohms
R11—10,000-ohm audio-taper potentiometer

Miscellaneous

F1—1-ampere fast-blow fuse
J1—Female panel-mount DB-25S connector
S1—Spst toggle switch
S2—Spst normally-open pushbutton switch
S3—10-section DIP switch
SPKR—2" replacement speaker
T1—12.6-volt, 1-ampere power transformer
XTAL1—10.00-MHz crystal
XTAL2—3.12-MHz crystal
6.5" × 4.5" prototyping board (see text); Wire Wrap sockets for IC2 thru IC12 and S3 (use low-profile socket for IC8); suitable enclosure (8.5" × 5" × 2.5" or larger); TO-220 heat sink for IC1; block-type fuse holder for F1; push-in solder posts; ac line cord with plug; rubber grommet; ½" spacers for circuit board (4); lettering kit; 4-40 machine hardware; assorted wire and cable; solder; etc.

Note: All components are available from local Radio Shack stores and/or Jameco Electronics, 1355 Shoreway Rd., Belmont, CA 94002.

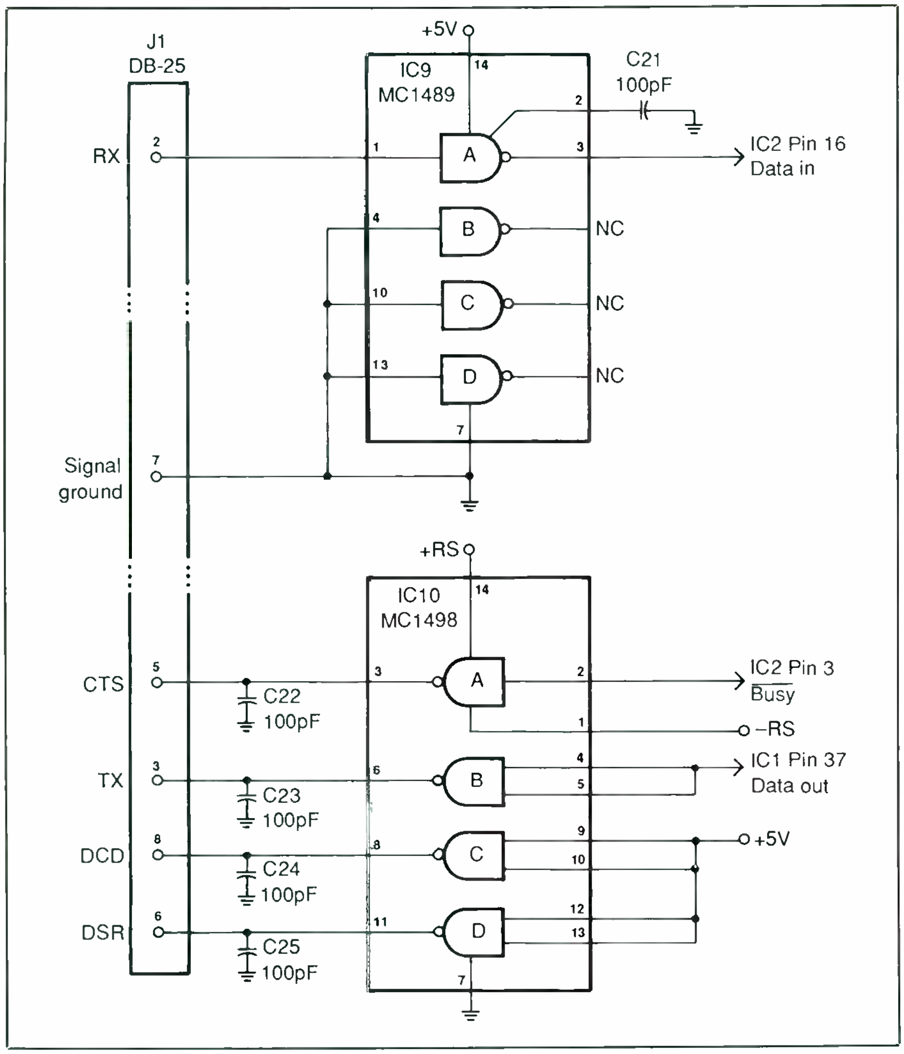


Fig. 3. RS-232C interface circuitry. Include this circuit for true serial feed from computer.

Before beginning to wire the project, determine which portions of the circuit you will need for use with your computer or other input device. At the very least, you need the basic circuit shown in Fig. 2 and the external RAM circuit shown in Fig. 6. synthesizer, I used a combination of Wire Wrap and traditional point-to-point wiring techniques. Almost all components (exceptions are the speaker, *F1* and its holder, *T1, J1, S1, S2, R11* and *LED1*) mount neatly on a 6.5″ × 4.5″ prototyping board with plated-through holes on 0.1″ centers but otherwise not clad with copper.

Whatever remaining circuitry you need will depend on the I/O capabilities of the input device that will be used to feed ASCII data to the speech synthesizer. For example, if you need only a parallel interface, all you need add to the basic circuit is the Fig. 5 circuit. In this case, you do not need the Figs. 3 and 4 circuits. On the other hand, if you need a serial interface, you need the Fig. 3 circuit (and the programmable parameters Fig. 4 circuit if your computer has no means for setting baud rate, parity and other parameters) but can omit the Fig. 5 parallel circuit.

The simplest approach, if your computer can support it, is a direct feed without an interface built into the speech synthesizer. To be able to do this, however, your computer must be able to provide 5-volt dc TTL signal levels through a serial or pseudo-RS-232C port or through an asynchronous receiver/transmitter I/O port. Only if your computer has a true RS-232C port and you want to feed ASCII data to the synthesizer via a serial line is the Fig. 3 circuit needed.

From the forgoing, it should be obvious that whatever circuits are not needed can be omitted from the synthesizer project if you wish to economize. However, the cost of the additional components to build in all the options described to meet any interfacing situation you might encounter is reasonably low enough that you might want to consider a full-blown version.

A suggested component layout on the circuit board is shown in Fig. 8. Plan on using Wire Wrap sockets for all ICs except *IC1* and DIP switch *S3*. Install only the sockets—*not* the ICs themselves—during the wiring phase.

Start stuffing the circuit board by plugging in the IC sockets and sparingly soldering one or two pins on both sides of each to physically anchor them into place as per Fig. 8. Make sure to minimize the length of pin that gets coated with solder. Also, use a small sharp knife to scrape away all traces of flux after soldering; otherwise, when you wrap wires to the pins, the flux is likely to insulate the pins from the wires.

Use 18- to 22-gauge solid wire for the runs from the +5-volt and ground power-supply input pins to the IC sockets and wherever else on the board the power buses are to go.

Once the IC sockets are in place and the wires for dc power distribution have been connected, install all 0.1-microfarad bypass capacitors as close as possible to the V_{CC} (+5-volt) pins of the sockets.

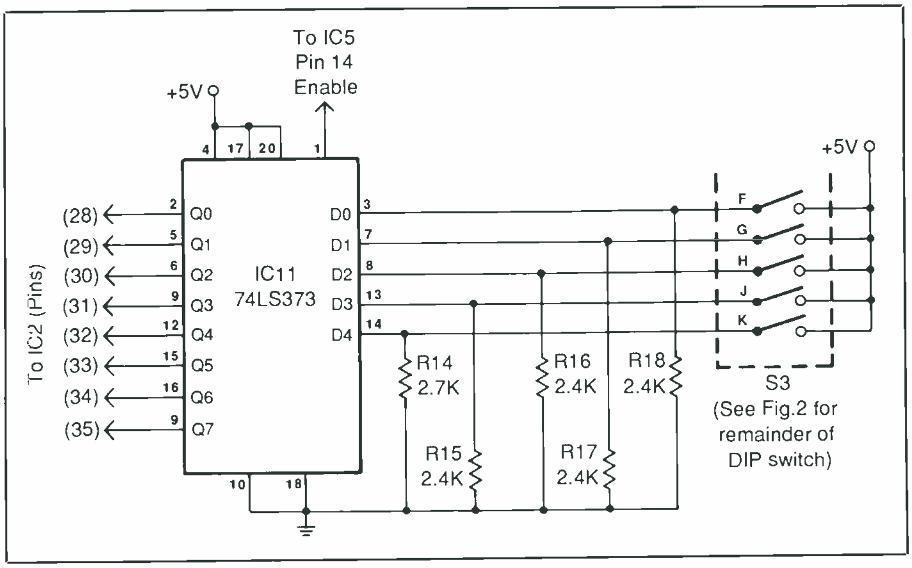


Fig. 4. Optional programmable parameters circuit provides means for setting baud rate, parity, etc. if these facilities are not built into computer with which project is to be used.

Mount voltage regulator *IC1* on a TO-220 heat sink and secure both to the board using a 4-40 × ½″ machine screw, lockwasher and nut. Connect *C4* and *C5* right at the pins of *IC1*, using as little lead length as possible.

Wire all signal lines with 30-gauge Kynar wire with the aid of a Wire Wrap tool. A good way to keep track of all connections and wire runs is to go over photocopies of the schematic diagrams with a red pen or pencil as each is made to avoid confusion and guard against missed wires.

Use push-in pins and solder to make all component lead connections for the *IC8* audio amplifier section. Ditto for the tap-off points for off-the-board components and connections.

Once the circuit is fully wired and all wiring has been double checked (do *not* install the ICs in their sockets until after you have made voltage checks and are sure that your wiring is correct), mount it inside a suitably sized enclosure. The selected enclosure should be large enough to accommodate all components, including the speaker, without crowding.

Figure 9 illustrates a suggested layout. It shows the circuit-board assembly, power transformer and fuse holder mounted to the floor and the speaker mounted to the lid of the enclosure. Use ½″ spacers and 4-40 × ¾″ machine screws, lockwashers and nuts to mount the circuit-board assembly and be sure to drill a number of holes in the enclosure's lid to allow the sound from the speaker to project outward. Infrequently used RESET switch *S2* and DB-25 connector *J1* are best mounted on the rear panel of the enclosure, which should also have a rubber-grommet-lined hole for entry of the ac line cord. Tie a knot in the line cord inside the enclosure to serve as a strain relief. On the front panel of

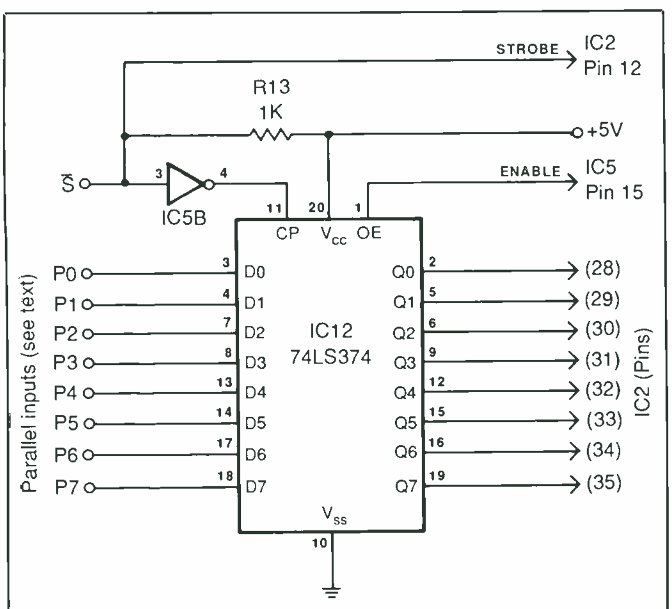


Fig. 5. Circuitry needed for parallel feed from computer.

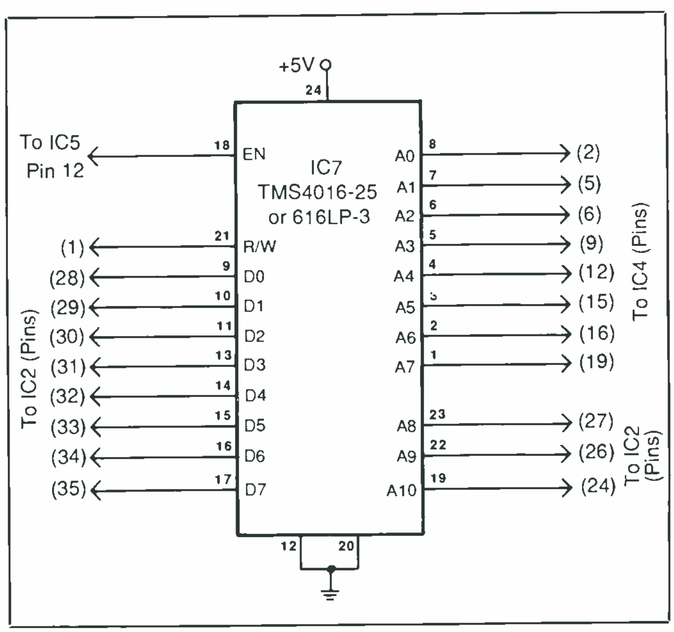


Fig. 6. External RAM memory circuit.

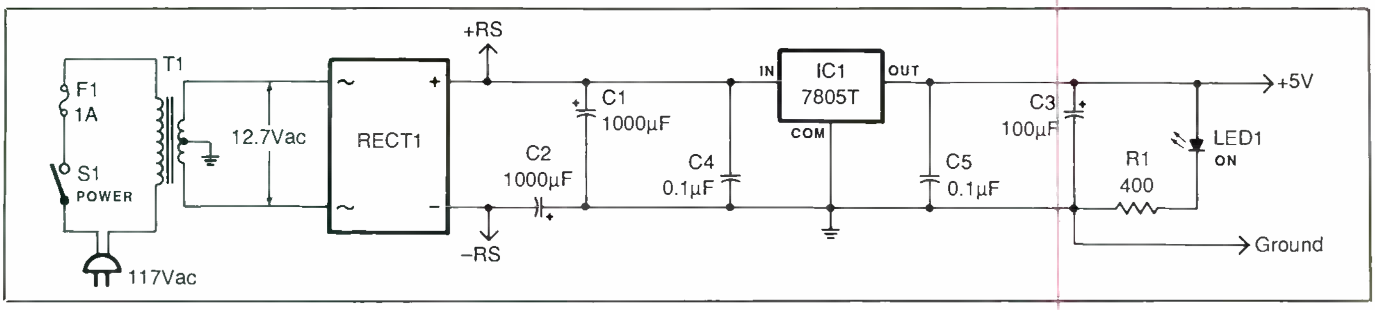


Fig. 7. Schematic diagram of ac-line-operated power supply for speech synthesizer.

the enclosure should be mounted VOLUME control *R11*, POWER switch *S1* and power ON indicator *LED1*.

After the project has been fully wired and assembled (still without the ICs in their sockets), label the POWER and RESET switches, VOLUME control; *J1* input and ON LED.

If you do not have a suitable cable to use between your computer and the speech synthesizer, you will have to make one yourself or have one made by a service that does this sort of thing. To make the cable, you need a male DB-25P connector and hood on one end and the proper connector for your computer on the other end. Any stranded wire or ribbon cable between the two connectors will work properly. However, for very long lines (in excess of, say, 12 feet), you should use shielded cable for the data lines.

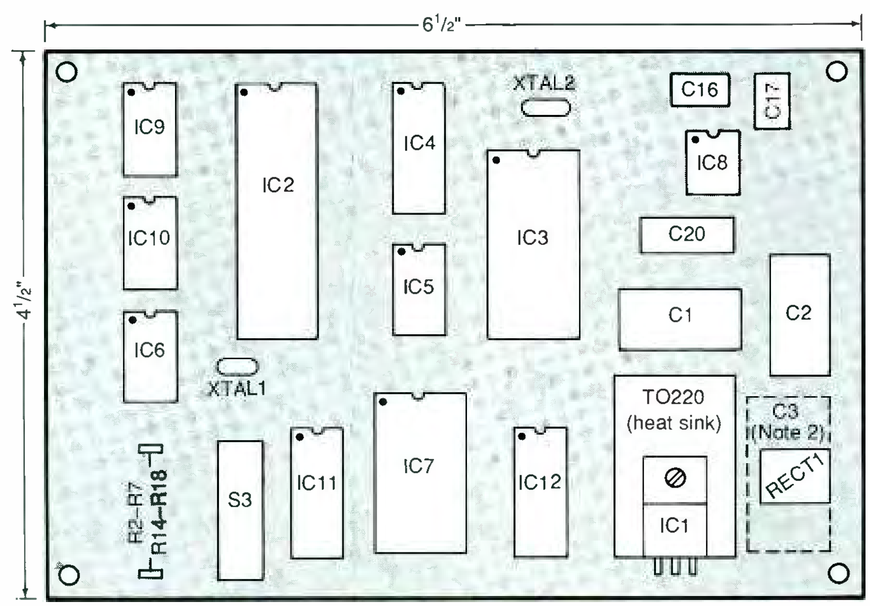
Checkout and Use

With the ICs still not installed in their sockets, plug the speech synthesizer's line cord into an ac outlet and turn on the power. The ON LED should light. If it does not, power down the project and rectify the problem. Then repower the synthesizer and use a multimeter set to read at least 5 volts dc to check for the proper voltages at points in the circuit as follows. First connect the meter's common probe to circuit ground at the negative side of *C3*. Then touch the meter's positive probe (observe the meter indication) to:

- ☐ positive (+) side of C3
- ☐ pin 25 of IC2
- ☐ pin 20 of IC3
- ☐ pin 20 of IC4
- ☐ pin 16 of IC5
- ☐ pin 14 of IC6
- ☐ pin 20 of IC7
- ☐ pin 6 of IC8
- ☐ pin 14 of IC9
- ☐ pin 14 of IC10
- ☐ pins 4,17,20 of IC11
- ☐ pin 20 of IC12

You should obtain a reading of +5 volts in all cases. If you do not obtain this reading, power down the circuit, pull the line cord from the ac outlet and recheck all wiring to correct the problem before proceeding further.

Once you do obtain the proper readings in all cases, turn off the power and wait a minute or so to allow the charges to bleed off the electrolytic capacitors in the power supply. Then install the ICs in their respective sockets. Handle these ICs with the same care as you would any other MOS-type device to avoid damaging them with static electricity. Make sure the proper ICs go into the sockets, that they are properly oriented (see Fig. 9) and that no pins



NOTES:
1. Use sockets for IC2 through IC12, S3.
2. Mount C3 on bottom of board.

Fig. 8. Most components that make up project mount on relatively small circuit board. In this layout on prototyping board, all options are shown installed.

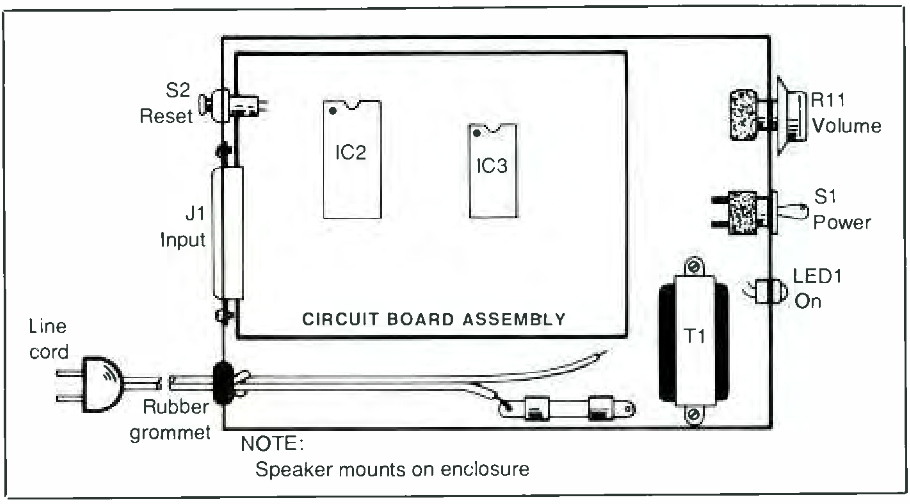


Fig. 9. Suggested component layout inside enclosure.

overhang the sockets or fold under between IC and socket as each is pushed solidly into place.

The speech synthesizer is very easy to use. Two operating modes are available, selectable by toggling section E of DIP switch *S3*. In the "any delimiter" mode (switch closed), any ASCII character that is not a letter or number, including the space, will cause the project to immediately vocalize the characters stored in the output buffer. For example, if you were to type "Modern Electronics." (without the quotes, of course), the synthesizer will begin "speaking" as soon as the space is reached and once again when the period is reached. This mode requires continuous typing unless the strings are stored in the computer's memory. The result will be a halting speech pattern, rather than a smooth flow of words and phrases.

Switching to "carriage return only" mode (section E of *S3* open) makes for more convenient operation. In this mode, long strings of characters (up to 600 characters at 1,200 baud) with all punctuation are sent to the synthesizer and automatically stored in the external RAM memory. Only upon receiving a carriage return (hex 0D) will the synthesizer vocalize what has been typed.

Sending an "escape" (ESC, hex 1B) at any time from the computer will clear the synthesizer's RAM and terminate all speech. If the buffer becomes overloaded, all you need do is press and release the RESET button on the rear of the synthesizer to get back into action. Doing this clears the synthesizer and returns an "OKAY" message on the computer's display screen. More information on the speech processor and code-to-speech converter is given in the data sheets Radio Shack provides with these chips.

A final note: Bear in mind that the speech synthesizer is a literal *phonetic system*. It will properly vocalize most text character strings sent to it. However, it cannot second guess non-phonetically spelled words in which characters are silent ("phlegmatic," for example) or are not pronounced as spelled ("Sioux," for example). Therefore, you will have to experiment on the proper misspellings to make such words come out sounding the way they should, such as "flematic" and "soo" or "sue." As you experiment with various words, you will discover that quite a number, even some that appear to sound exactly as they are spelled, will have to be judiciously misspelled to make them sound as they should. **ME**