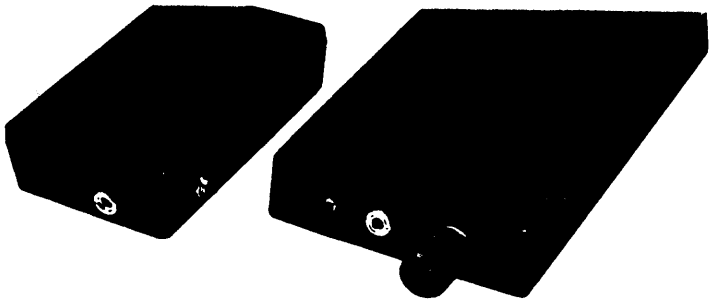
A Pocket-Size, *Talking* Morse Code Practice Computer



Here's a go-anywhere Morse-code companion to teach and entertain you.

By Ken Staton, KD6LHB
PO Box 2601
El Granada, CA 94018
Photos by Kirk Kleinschmidt, NTØZ

Have you ever wished you could practice copying Morse code during that boring commute to and from work, during a lunch break, or even while exercising? With the Morse Code Practice Computer (CPC), you can fulfill that wish. This pocket-sized Morse code generator can go almost anywhere you do. The CPC not only generates random Morse characters or groups, it *speaks* the code sent to provide feedback! The CPC costs about $50 to build—about the same price as a portable cassette player and set of Morse code tapes.[1] The CPC is self-contained, there are no tapes to lose or break and—because it generates random code—you'll never become familiar with the character sequence. Rather than the raucous tones produced by some code-practice devices, the CPC generates pleasant *sinusoidal* tones. You can set the CPC's code rate to Farnsworth speeds of 5 wpm (character rate of 16 wpm), 13 wpm (18 wpm character rate), or 20 wpm (23 wpm character rate). CPC software is available (see Note 1) and is easily modified.

Circuit Description

The CPC circuit (Figure 1) is a simple computer with an analog output section. U1 is a relatively inexpensive 8031 microcontroller. The 8031 addresses a full 64 kB of external program memory in a replaceable EPROM (U2), which makes software changes easy. U3 latches the least-significant 8 bits of the address bus from I/O port 0, which multiplexes the address and data.

To minimize the parts count and cost, I use digitized speech rather than a speech synthesizer IC. U4, the 8-bit digital-to-analog converter (DAC), is a single-supply-voltage IC that can be configured for voltage output. To operate U4 in the voltage-output mode, the R/2R ladder is driven backward by U6, a 2.5-V reference connected to U4's current-mode output pin. The DAC voltage output is obtained from the reference pin. U5A buffers the DAC output and drives a Sallen-Key low-pass active filter built around U5B. R4 through R6 set the output level to 0.5 V P-P across an 8 to 16-Ω load. Regulator U7 allows operation from a vehicle battery and 12 and 9-V sources.

A brief description of the software, and the process used to digitize speech, are presented in the sidebar.

Construction

The CPC is easy to build using any construction technique. Aim to keep the analog output circuitry away from the digital circuitry. Using the point-to-point wiring technique, you'll achieve the smallest possible assembly. It's possible to fit the entire circuit in a 0.995×2.25×3.6-inch box (see Figure 2), including a socket for the EPROM. (Use a socket for the EPROM so you can easily make software changes.) A slightly larger version of the CPC (Figure 3) uses a PC board, in a Serpac 051-I box. The PC board is single-sided to encourage individual fabrication.[2]

If you mount the CPC in an enclosure, you'll need to change the **VOLUME** control (R5) to a panel-mount potentiometer, or set the PC-board-mounted R5 for maximum output and use an in-line stereo, variable attenuator (such as Radio Shack's 42-2459) with your earphone as a volume control. All components are available from JDR Microdevices and from Jameco.[3] Many components are available from other sources.

For best battery life, use CMOS parts. The CMOS 80C31 is somewhat more expensive than the HMOS 8031. The trade-off here is the long-term cost of batteries versus the initial cost of the microcontroller.

Operation

Headphone jack J1 is a stereo jack, but it's intended for use with a *monaural* headset. (Use an adapter if you have stereo headphones.) Plugging in the headphones turns on the unit: The sleeve of the monaural plug makes contact between the battery's ground and circuit ground.

For use during commuting, you can connect the CPC output to a CD player-to-cassette adapter (such as the Radio Shack 12-1951) and listen to Morse code using your vehicle's stereo system.

The CPC's random number generator produces a pseudorandom sequence. This means that the sequence will be the same for a given seed value. The seed is changed every time the push-button switch is held down during mode changes. To generate a different sequence while retaining the default setting, select **X** from the menu.

Operating Modes

The default mode is set to random characters at Farnsworth 5 wpm (this is what our local VEC uses). Push-button switch S1 (**MODE**) is used to change operating modes. Continually pressing the switch cycles through a verbal menu which says *G, R, S, T, 5, 1, 2,* and *X*. Releasing the switch during or immediately after the desired mode is voiced selects that mode. The modes are: G = 5-character groups, R = random characters, S = sequential characters, T = toggle speech on/off, 5 = Farnsworth 5 wpm (character rate of 16 wpm), 1 = Farnsworth 13 wpm (18 wpm character rate), 2 = Farnsworth 20 wpm (23-wpm character rate), X = eXit with no mode change.

Summary

The code-practice computer generates random code and provides feedback help-

[1]Notes appear on page 28.

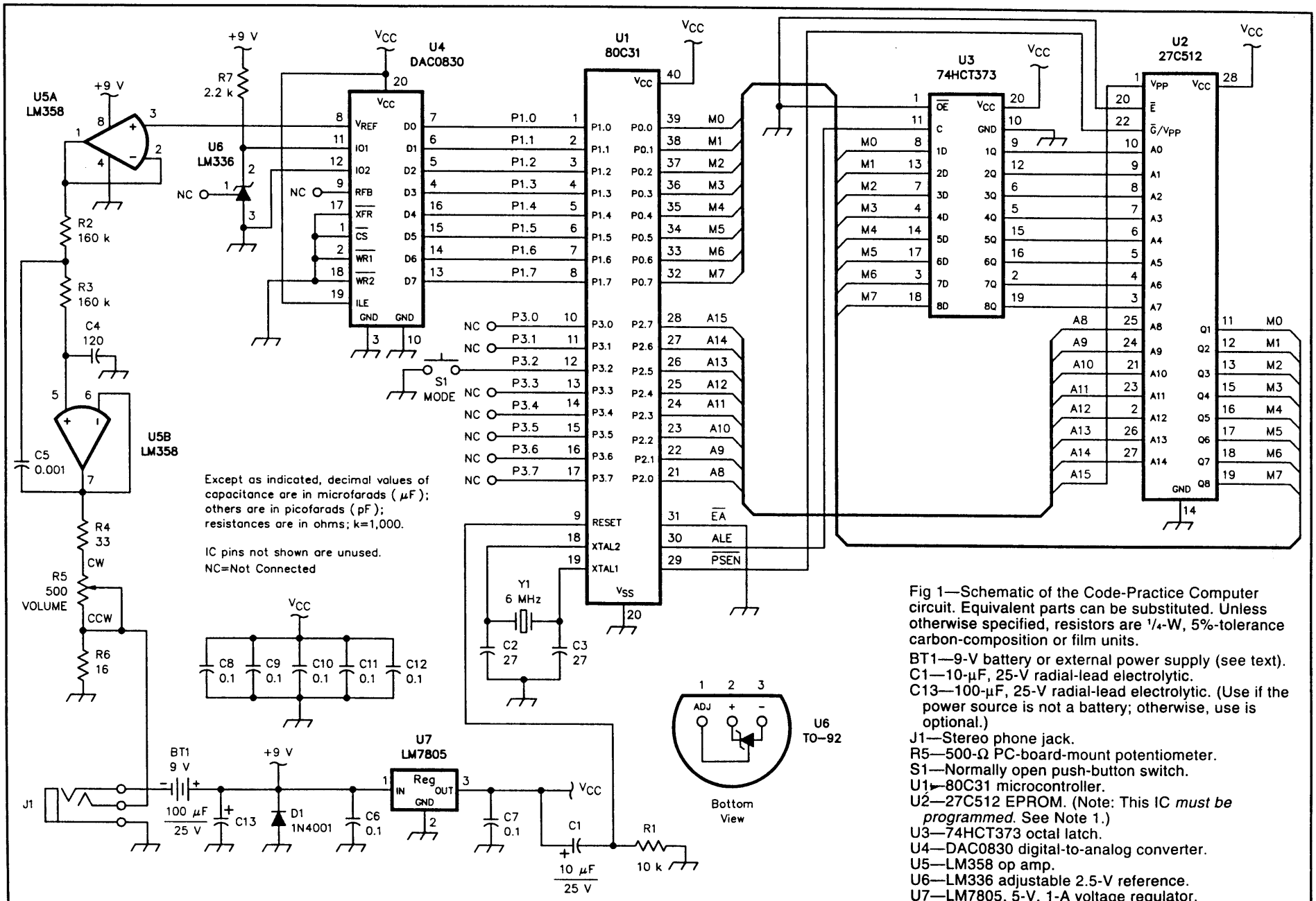


Fig 1—Schematic of the Code-Practice Computer circuit. Equivalent parts can be substituted. Unless otherwise specified, resistors are 1/4-W, 5%-tolerance carbon-composition or film units.

BT1—9-V battery or external power supply (see text).
C1—10-μF, 25-V radial-lead electrolytic.
C13—100-μF, 25-V radial-lead electrolytic. (Use if the power source is not a battery; otherwise, use is optional.)
J1—Stereo phone jack.
R5—500-Ω PC-board-mount potentiometer.
S1—Normally open push-button switch.
U1—80C31 microcontroller.
U2—27C512 EPROM. (Note: This IC *must be programmed*. See Note 1.)
U3—74HCT373 octal latch.
U4—DAC0830 digital-to-analog converter.
U5—LM358 op amp.
U6—LM336 adjustable 2.5-V reference.
U7—LM7805, 5-V, 1-A voltage regulator.

Misc: 28-pin IC socket, enclosure, 9-V battery, battery clip or double-stick tape.

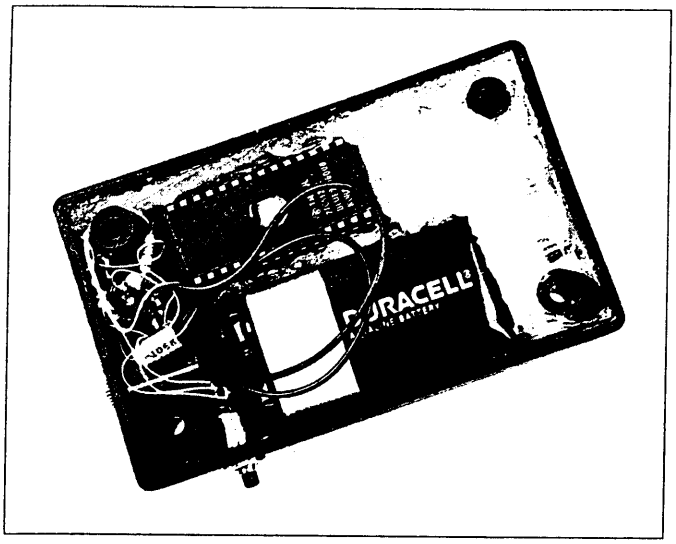


Fig 2—In this small version of the CPC, the micro-controller and attached components have been encapsulated in flexible sealant. The EPROM is left exposed for easy removal and reprogramming.

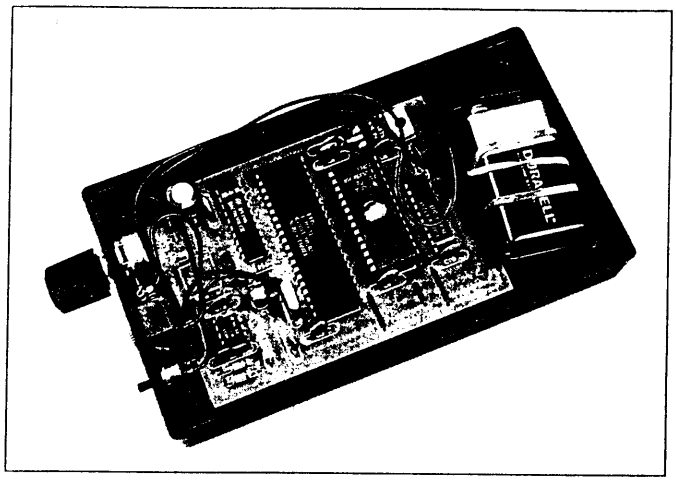


Fig 3—In this PC-board version of the CPC, everything is readily accessible. Double-stick adhesive tape can be used to hold the battery in place. The Serpac 051-I enclosure measures approximately 1×3×5½ inches (HWD) .

ful in learning code quickly. I still use the CPC when I want to practice copying Morse code but can't sit at the radio. My current code speed is 13 wpm; my goal is to reach 20 wpm.

When you no longer use the CPC for Morse code practice, you can turn it into a great single-board computer for another project. I'm sure you'll enjoy building and using this Morse code teaching companion.

Notes

[1]Source code, PC boards, programmed EPROMs and kits are available from The Staton Company, PO Box 2601, El Granada, CA 94018. An IBM-formatted floppy disk with source code is $10; PC board, $15; programmed EPROM, $15; kit (excluding battery and enclosure), $60. Shipping and handling in the US and Canada is $7; all other countries, $15. Payment must be made in US funds drawn on a US bank. California residents must add 8.25% sales tax.

[2]A PC-board template package is available free from the ARRL. Address your request for the STATON CODE-PRACTICE COMPUTER TEMPLATE to: Technical Department Secretary, ARRL, 225 Main St, Newington, CT 06111. Please enclose a business-size SASE. See Note 1.

[3]JDR Microdevices, 2233 Samaritan Dr, San Jose, CA 95124; tel 800-538-5000, 408-494-1400; fax 800-538-5005, BBS 408-494-1430. Jameco, 1355 Shoreway Rd, Belmont, CA 94002-4100, tel 800-831-4242, 415-592-8097, fax (domestic): 800-237-6948 or 415-592-2503, fax (international): 415-592-2503.

The software is available for downloading from CompuServe, the ARRL BBS (203-666-0578) as *STATON.ZIP*, by FTP on the Internet site oak.oakland.edu/pub/hamradio/arrl/infoserve/qst files area, or from the author on floppy disk.

Ken Staton is an electrical engineer at a large US corporation. He has a BSEE from Stanford and an MSECE from UCSB. Although Ken was interested in Amateur Radio since 1976, he didn't get his license until 1992, when he realized there was a no-code option. As this article was being prepared for publication, Ken upgraded to Advanced. QST

The CPC's Software

Most of the software details are included in the source-code files. Since reading program source files is not a fun thing to do, I'll briefly describe the software modules.

I digitize the speech using a personal computer equipped with an 8-bit Sound Blaster sound card, then compress the speech using a first-order adaptive differential pulse-code-modulation algorithm (ADPCM) to achieve 2:1 compression with good sound quality. During operation, the compressed, digitized speech is decompressed and output to the DAC by the CPC software. The CPC is programmed in 8051 assembly. The main program loop is CPO_SPK. CPO_SPK calls subroutines MORSE and SPEAK. MORSE generates code based on a bit pattern encoded in a byte (8 bits). SPEAK speaks digitized speech for the character set (26 + 10 + 3 + 4 = 43 characters).

If you want to make modifications to the CPC program, there are several inexpensive 8051 cross assemblers available. One is *TASM*,* a shareware package from Speech Technology Inc. that supports a variety of processors, including the TI 32010 DSP. A comprehensive 8051 reference is *The 8051 Microcontroller: Hardware, Software and Interfacing,* by James W. Stewart.†

The binary speech data is generated by a series of programs written in Microsoft Visual C that run on a personal computer operating in the Microsoft Windows environment. These programs can be modified easily to compile with other C compilers for execution on different systems. *SPCH_LIB* input is a text file defining the sound-board data files; it outputs a text file for the assembler that defines the speech table and a binary file to be programmed into the EPROM for speech data. *SPCH_LIB* calls *VOC2BIN* and *ADPCM*. *VOC2BIN* converts sound-board data files (for the 8-bit Sound Blaster board) into binary speech files. *ADPCM* converts (compresses) binary files into adaptive, differential, pulse-code-modulated files with an approximate reduction of 2:1 in file size.

*Speech Technology Inc, 837 Front St S, Issaquah, WA 98027, tel 206-392-8150; CompuServe ID 73770,3612. *TASM* is available on CompuServe in the IBM Programming forum as *TASM.ZIP*.

†J. Stewart, *The 8051 Microcontroller: Hardware, Software and Interfacing,* (Simon & Schuster: Englewood Cliffs, NJ 07632), 1993.

New Products

ADD-ONS FOR YAESU FT-1000

◊ Two CW receiving filters for the sub-receiver on the Yaesu FT-1000 HF transceiver let the user select bandwidths of 250 and 400 Hz. Retail price is $130 plus $6 s/h. Tuning Upgraders for the main and sub-tuning controls cut the tuning rate to 2.5 kHz per revolution for finer tuning and bandspread. Retail price $46.95 plus $6 s/h. Robert Pohorence, KB4JMQ, International Radio and Computer Inc, 3804 S US 1, Ft Pierce, FL 34982; tel 407-489-0956, fax 407-464-6386. QST