

Apple, Morse, and You

Simple software and cheap hardware make for a great CW keyboard recipe. Just add one operator.

Have you been looking for a CW keyboard program in BASIC for your Apple II computer with Applesoft? The program provided here will enable you to use the Apple as a CW keyboard with sidetone and speeds adjustable from around 5 words per minute to about 70. The program is easily modified to provide for any special characters you might wish to have.

Lines 1000 to 1460 contain the Morse data. The data lines consist of the ASCII value for the character followed by a Morse representation. A 3 signifies a dash, a 1 stands for a dot, and a 0 marks a space. For example, line #1110 says the following:

1110 DATA 65,130

The number 65 is the ASCII code for an upper-

case A. The 130 means dah, followed by a short space to keep it from running into the next letter.

When the program is run, the computer will take the data listed between lines 1000 to 1460 and load it into the string matrix CODE\$. The location within CODE\$ is the ASCII value of the character; in other words, location 65 in CODE\$ contains the data for the Morse letter A. When characters are sent to the computer from the keyboard, the computer changes the letters to their ASCII value (using the BASIC ASC(\$)) command) and looks in CODE\$ (<ASCII value of letter>) for the Morse data. The Morse data is then dissected character for character using the BASIC MID\$(X\$,X,X) command.

Lines 80 to 90 are for the CW sidetone. These lines

POKE a machine language subroutine which will enable the Apple to send the CW through the internal speaker. The value POKED into decimal location 768 on line 70 determines the pitch of the sidetone.

Line 130 gets a single character input from the keyboard. This character is then checked to see if it is a control character used for a special purpose on lines 140 to 160. In this program, the control characters A, B, Y, and Z are used for special functions.

The subroutine at line 200 is what takes the Morse data from the proper location in CODE\$ and generates the Morse signal from it. The Morse character can be heard on the Apple speaker. Also, the annunciator TTL output number 0 will be toggled on and off in Morse code. This signal

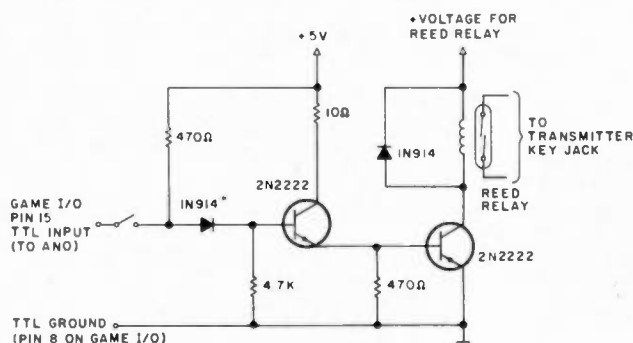


Fig. 1. No parts values are critical. *Any small switching diode can be used.

can be used to drive your transmitter through a suitable driving circuit. The circuit used must take the TTL output and use it to drive a reed relay which, in turn, keys the transmitter. An example of such a circuit is shown in Fig. 1. The lines which turn AN0 on and off are located at line 235 and line 245. Line 240 is a call to the machine language sidetone routine.

Using the program is a simple procedure. When the program is run, the screen will clear and the following will appear: MESSAGE—>

After this prompt, type in a message for the memory. This message can be played back at any time while in the keyboard mode of operation by simply pressing either CTRL A or CTRL B. The message may be up to 254 characters in length. If the message contains commas, enclose it in quotes.

The next thing to appear on the screen is: SPEED

Type in a number between 1 and 85. Note that 85 is the slowest speed. The fastest usable speed is about 6. At speeds faster than this, the transmitter does not key accurately. When set for a speed of 7, the code speed as measured on a Kantronics Field Day II code reader was in excess of 75 words per minute.

After the above two items are entered in, the program will operate as a CW keyboard. When a letter is typed on the keyboard, it is sent through the speaker and AN0 in Morse code and displayed on the monitor. If you want to play back the pre-programmed message, hit control A or B. To reset the speed, use CTRL Z. If you want to change the message, hit the CTRL Y keys.

I gave the following keys special functions: ESC produces the Morse error code (eight dits), and the colon (:)

is the CQ key. To add a special function to any key of your choice, all you must do is determine the ASCII value of the character and put it on a DATA line followed by the Morse representation of what you want sent. For example, if your name were Joe and you wanted the computer to send your name whenever CTRL N was pressed, you would add the following line:

DATA 19, 1333003330010

The line number can be any unused one, as long as it comes before the end of data marker at line 1460.

Because of the structure of this program, it is quite easy to make some useful changes. One thing that would be quite easy to implement would be to have the program output American Morse instead of International Morse. To get the longer dashes used in Continental code, use a 4 or a 5 instead of a 3 for the dash character. This will limit the top speed number that can be entered in to a 51. It also would be necessary to change the end of line 50 to read:

IF S * 5 > 255 THEN 50

```

1 09 = PEEK (49240): REM TURN XMITTER OFF!!
5 REM :WRITTEN BY ROBERT SMIRSKY
10 DIM CODE$(128)
20 HOME
30 INPUT "MESSAGE--->:";M$
40 IF M$ = "" THEN M$ = " "
50 HOME : INPUT "SPEED:";S : IF S * 3 > 255 THEN 50
60 RESTORE
70 POKE 768,50
75 REM :SIDETONE
80 POKE 770,173: POKE 771,48: POKE 772,192: POKE 773,136: POKE 774,208: POKE
775,51: POKE 776,206: POKE 777,11: POKE 778,3: POKE 779,240: POKE 780,9
: POKE 781,202
90 POKE 782,208: POKE 783,245: POKE 784,174: POKE 785,0: POKE 786,3: POKE
787,76: POKE 788,2: POKE 789,3: POKE 790,96: POKE 791,0: POKE 792,0
100 REM
110 GOSUB 310
120 HOME
130 GET R$
140 IF ASC (R$) < 3 THEN GOSUB 370
150 IF ASC (R$) = 26 THEN 50
160 IF ASC (R$) = 25 THEN 20
170 D = ASC (R$)
180 GOSUB 200
190 GOTO 130
200 FOR X = 1 TO LEN (CODE$(D))
210 C$ = MID$(CODE$(D),X,1)
220 IF VAL (C$) = 0 THEN GOTO 350
230 POKE 769,(S * (VAL (C$)))
235 X9 = PEEK (49241)
240 CALL 770
245 X9 = PEEK (49240)
250 FOR X1 = 1 TO 3: NEXT
260 NEXT
270 PRINT R$;
280 FOR X1 = 1 TO S * 1.3: NEXT X1
290 RETURN
300 END
310 READ F,F$
320 IF F = - 1 THEN RETURN
330 LET CODE$(F) = F$
340 GOTO 310
345 REM :SPACE
350 FOR X1 = 1 TO S * 1.7: NEXT
360 GOTO 260
370 FOR H1 = 1 TO LEN (M$)
380 LET R$ = MID$(M$,H1,1)
390 D = ASC (R$)
400 GOSUB 200
410 NEXT H1

```



NEW FOR THE TRS-80* COLOR COMPUTER MORSE CODE



Here's a way to turn your Color Computer into a complete CW Morse Code terminal.

—KA9FSQ CW MODEM Interfaces the computer to your transceiver via the ROM-PAC slot.

Cartridge: \$50.00

—Three CW programs to choose from.

—KA9FSQ TRANSLATOR (Machine language)

TX programmable from 5-60 WPM, RX to 60 WPM. Features split screen operation, a 255 character keyboard buffer, and automatic CQ using your own call letters.

Cassette: \$24.95

—W9AV TRANSLATOR (Extended Color Basic)

TX programmable 5-50 WPM, RX to 30 WPM. Includes 9 message memories, and a CW practice mode which sounds over your TV.

Cassette: \$14.95

—W9AV QSO ROBOT (Extended Color Basic)

TX programmable 5-50 WPM, RX to 30 WPM. Use the keyboard or Auto Mode. Makes complete CW QSO's without human intervention including exchange of calls, RST, QTH, and names. Also keeps a log.

Cassette: \$19.95

For additional information & programming, write to:

MITRONIX

Mike Rice KA9FSQ
5953 N. Teutonia Ave.
Milwaukee WI 53209

*Trademark Tandy Corp.

please include 5% postage #240

This is needed to accommodate the longer dash character.

I hope you will enjoy using this program. If you own a micro other than an Apple II, you can use the

same fundamental program as long as your BASIC contains string manipulation functions. The sidetone and keying of the rig will vary greatly on different types of computers. ■

Program Listing.

```

420 R$ = ""
430 RETURN
1000 DATA 49,133330
1010 DATA 50,113330
1020 DATA 51,111330
1030 DATA 52,111130
1040 DATA 53,11111
1050 DATA 54,31111
1060 DATA 55,33111
1070 DATA 56,33311
1080 DATA 57,33331
1090 DATA 58,33333
1100 DATA 32,00
1110 DATA 65,130
1120 DATA 66,31110
1130 DATA 67,31310
1140 DATA 68,3110
1150 DATA 69,10
1160 DATA 70,11310
1170 DATA 71,33100
1180 DATA 72,11110
1185 DATA 47,311310
1190 DATA 73,110
1200 DATA 74,13330
1210 DATA 75,31300
1220 DATA 76,13110
1230 DATA 77,03300
1240 DATA 78,310
1250 DATA 79,03300
1260 DATA 80,13310
1270 DATA 81,33130
1280 DATA 82,1310
1290 DATA 83,11100
1300 DATA 84,00300
1310 DATA 85,11300
1320 DATA 86,11130
1330 DATA 87,13300
1340 DATA 88,31130
1350 DATA 89,31330
1360 DATA 90,33110
1370 DATA 46,131313
1380 DATA 27,11111111
1390 DATA 13,31113
1400 DATA 44,331133
1410 DATA 38,10111
1420 DATA 45,31113
1430 DATA 58,3131003313
1440 DATA 63,113311
1450 DATA 18,13131
1460 DATA -1,-1

```