Behavior Research Methods, Instruments, & Computers
1986, 18 (2), 222-227

A Commodore 64-based experimental psychology laboratory

HOWARD J. KALLMAN
State University of New York, Albany, New York

When used in conjunction with a 6522 interface board, the Commodore 64 (C64) computer can control and precisely time externally generated stimuli. A C64-based experimental psychology laboratory that uses these capabilities is described along with a set of programs that can be used to control experimentation. The pros and cons of using the C64 as a laboratory computer are discussed.

The use of microcomputers in experimental psychology laboratories has increased dramatically in recent years. Numerous articles have described the research capabilities of the Apple II, TRS-80, and PET computers (e.g., Czerny, 1979; Dlhopolsky, 1984; Durrett, 1978; Gordon, Foree, & Eckerman, 1983; Osaka, 1979; Perera, 1981; Perone, 1985; Poltrock & Foltz, 1982; Thompson, 1979). The purpose of this article is to describe some of the capabilities of a relatively inexpensive microcomputer, the Commodore 64 (C64), and outline its use in an experimental psychology research laboratory. In addition to hardware considerations, a series of flexible programs for conducting psychological research using the C64 will be described.

The C64 is based on the 6510 microprocessor, which is an updated version of the 6502. The 6510 is an 8-bit microprocessor which, in the Commodore 64, runs at a clock rate of approximately 1.0225 MHz (Schnedler Systems, 1984). The main difference between the 6510 and the 6502 is that the 6510 has six additional input/output (I/O) lines which are used for memory management and cassette operations; both microprocessors use the same machine code.

The C64 comes equipped with 64K of random-access memory (RAM) and approximately 20K of read only memory (ROM), which can be bank-switched with the RAM. A BASIC 2.0 interpreter and the "Kernal" operating system reside in ROM.

For input and output, there are two 6526 complex interface adapters (CIAs), a 6567 video interface chip (VIC-II), and a 6581 sound interface device (SID), which can be used for sound synthesis. Each 6526 CIA includes 16 I/O lines which may be used for input and/or output. Each CIA also includes two 16-bit interval timers which decrement at the system frequency and may be cascaded; a real-time clock is also accessible, but the resolution is only $\frac{1}{10}$ sec. Timer A in CIA No. 1 generates interrupts every $\frac{1}{60}$ sec to allow for keyboard scans and other housekeeping. Note that the system interrupt rate, unless modified through software, imposes a limit on the accuracy of reaction times that are based on keyboard input.

In the laboratory described here, the Commodore is used to control peripheral laboratory equipment. A separate I/O board rather than the 6526s is used for interfacing because (1) the 6526 appears to have a high failure rate and is difficult to replace, and (2) there are only a few free I/O lines available on the C64's CIAs. Under normal conditions, only Port B of the second CIA can be used for I/O interfacing (De Jong, 1984). As an alternative to using this CIA, a dedicated I/O interface board, such as the Schnedler Systems Model 64IF22 dual VIA board,[1] which includes two 6522 versatile interface adapters (VIAs), may be used for laboratory interfacing. The 6522 is well documented and often used with other 6502-based microcomputers, such as the Apple II. Details on programming and interfacing the 6522 VIA are available in many 6502 assembly language programming books (e.g., De Jong, 1982; Leventhal, 1979; Scanlon, 1980; Zaks, 1979).

Like the 6526 CIA, the 6522 VIA includes 16 I/O lines, two 16-bit timers that decrement at the system frequency, and interrupt capabilities. The Schnedler Systems Model 64IF22 VIA board installs in the expansion port of the C64 and as many as three slave interface boards may be daisy-chained to the master board, although the use of more than one board requires an external 5-V power supply (the laboratory configuration described below is based on a single interface board drawing power from the computer). Input and output to the 6522s is memory mapped. Consequently, assembly language I/O routines for the 6522 that have been developed for use on other 6502-based computers can be used on the C64 by simply changing the I/O addresses.

This research was made possible, in part, by a grant from the General Electric Company. I thank Shirley Brown and Kevin Overbaugh for their helpful comments on an earlier version of this article. Requests for reprints and program listings may be sent to Howard J. Kallman, Psychology Department, State University of New York, 1400 Washington Avenue, Albany, NY 12222. A disk (1541 format) that contains the EXP64 software system may be obtained by sending $5 to cover expenses.

THE HARDWARE

General Configuration

In my laboratory, the C64 controls the parameters of auditory signals which are generated by either a Wavetek 159 programmable oscillator or various Coulbourn Instruments auditory modules. In addition to controlling the parameters of the signals, the computer times events, records and analyzes subject responses, and provides visual feedback to the subject. Although the particular application described here focuses on the control of auditory stimuli, the laboratory configuration and supporting software could be adapted to accommodate other types of stimuli, generated either by the computer itself or by external equipment.

In a typical experiment, the subject listens to stimuli presented through headphones in a soundproof chamber. A Cardco numeric keypad, which serves as an extension of the computer keyboard, is used for entering responses. Visual feedback is presented over a monochrome CRT monitor. The experimenter, computer, and auditory equipment are located outside of the experimental chamber, although there are instances in which it is preferable to have the computer and equipment inside the chamber. The C64 is equipped with a disk drive and a monochrome CRT monitor. The Schnedler Systems interface board is used to interface to the auditory equipment although additional buffering, as described below, is required. The video output from the C64 is split, using a Y-connector, and is directed to both the experimenter's monitor and the subject's monitor. The numeric keypad, which is used for collecting subject responses, is plugged into Control Port 2 on the C64, using a Wico Video System extension cord. A dot-matrix printer with a Centronics parallel port is connected to the C64, using one of the many available C64 printer interfaces; a printer is not necessary to run the research, however. As noted above, in certain applications, the computer is placed in the experimental chamber and the subject responds directly on the computer keyboard; this arrangement eliminates the need for the numeric keypad and the second CRT monitor.

Interfacing

The Schnedler Systems interface board plugs into the expansion port on the C64 (which is where cartridges normally plug in). The board includes four 16-pin DIP sockets, one for each of the two ports on each VIA. Each port has eight I/O lines which are labeled PA0-PA7 (Port A) or PB0-PB7 (Port B). There are two other pins at each port (CA1 and CA2 or CB1 and CB2) which, under certain conditions, may also be used for I/O. Finally, there are five pins which may be used as grounds and a 5-V output pin at each port.

In the current laboratory configuration, Port A of VIA No. 0 is unused, but there are plans to connect response buttons to this port so that reaction times may be recorded with millisecond precision. In this application, a button response would be used to generate an interrupt and one of the interrupt service routines would record the reaction time.

Port B of VIA No. 0 is used for timing stimuli and interstimulus intervals with millisecond precision. This capability requires that the user hardwire PB7, which generates pulses to PB6, which is used as a counter.

Port A of VIA No. 1 is used to gate 5-V signals to the Coulbourn Instruments and Wavetek equipment; in other applications, the 5-V signals could be used for other purposes. An external 5-V power supply is used to provide the voltages which are inverted by a 7404 IC. Port A of VIA No. 1 is mapped to C64 memory location $DE11. Because the voltages are inverted by the 7404, any bit in memory location $DE11 that is set to 0 will result in 5 V on the corresponding output line. Output on lines PA0-PA5 is sent to the Coulbourn Instruments modules. Because most of the Coulbourn Instruments modules require −12-V signals for stimulus parameter control and triggering, the gated 5-V signals are first routed to a Coulbourn Instruments Model S22-06 computer input buffer, which converts each 5-V signal to −12 V. These converted signals are then sent to other Coulbourn Instruments modules to trigger audio gates, adjust the frequencies of tones, and so forth. Line PA7 of Port A, VIA No. 1 is used to send trigger voltages to the Wavetek oscillator; line PA6 is currently unused.

ASCII code is sent to the Wavetek from Port B of VIA No. 1 to control parameters of the Wavetek signal, such as frequency, amplitude, and waveform. Output from pins PB0-PB6 are sent to the Wavetek input data lines XD0-XD6, respectively, which are accessed at the Wavetek using a Molex 03-09-2151 connector. Two 7406 chips with 2.2-kΩ resistors on each output line are used as buffers between the Schnedler board and the Wavetek. Pulses are sent to the Wavetek clock control line from pin CB2 of VIA No. 1; a 7407 chip with a 2.2-kΩ resistor is used to buffer this line, which connects to the Wavetek through the Molex connector. The same external power supply used to power the lines going to the Coulbourn equipment provides 5 V on each of the lines going to the Wavetek. Using the configuration just outlined, it is possible to control the Wavetek and Coulbourn Instruments equipment simultaneously.

The configuration used in other laboratories would be expected to differ from ours due to the use of different peripheral laboratory equipment. The documentation that comes with the Schnedler Systems VIA board is extensive and includes substantial information about interfacing and programming the 6522s. In addition, many of the books on 6502 programming include information on interfacing and programming the 6522.

The Sound Interface Device

The C64 includes a SID, which is capable of generating a wide range of sounds. It is thus possible to conduct auditory experiments without investing in expensive peripheral auditory equipment. The SID can generate square waves, sawtooth waves, triangle waves, and noise.

Up to three voices may be played simultaneously. Most of the signal parameters (e.g., frequency or waveform) of the three voices may be programmed independently, although the amplitudes of the three voices cannot be set independently. The SID includes filtering capabilities, control over the amplitude envelopes of the sounds, and so forth. Despite these capabilities, the SID should not be used in experiments that require precise control over the auditory stimulus unless the relevant signal parameters are measured and found to be satisfactory. The characteristics of the SID's programmable filter vary dramatically from one C64 to another (Lugt, 1986). Furthermore, there is often a high level of noise in the output of the SID, which may result in unacceptable distortion of the signal. And, although the SID is relatively flexible, it is somewhat cumbersome to program. Nevertheless, the SID could prove useful in studies that do not require precise specification of the parameters of the auditory stimulus.

The Video Interface Chip

The C64's video capabilities are outlined in *Commodore 64 Programmer's Reference Guide* (Commodore Business Machines, 1982) and in West (1985), and will not be reviewed here. However, one point may be of particular interest to potential users. The C64 has a raster register at memory location $D012. Reading this location yields the lower 8 bits of the current raster position; the most significant bit is held at location $D011. In addition, an interrupt bit is set whenever the current raster position equals the numbers written to these two locations. These capabilities are potentially useful in experiments that require the measurement of reaction times to visual displays. Because video monitors are refreshed once every 16.7 msec, precise reaction times to the onsets of visual displays can be measured only if the timing routines are synchronized with the refresh cycle (see Poltrock & Foltz, 1982, and Reed, 1979, for discussions of this issue). Because the raster register provides information about the current raster position and may be used to generate interrupts when certain raster positions are reached, it is possible to synchronize visual displays generated by the VIC-II with timing routines.

Additional Hardware Considerations

Initial shipments of the C64 were plagued by reliability problems. These problems have subsided, although the power supply continues to exhibit a high failure rate. To address this problem, replacement power supplies are available from many outlets. Given reliability problems with the Commodore power supply, use of a separate power supply for the Schnedler Systems interface board should be considered.

Another weak link in a Commodore 64 system is the 1541 disk drive. It is very slow and tends to require relatively frequent alignments due, in part, to heat problems. The latter difficulty seems to have lessened somewhat with the introduction of a redesigned version of the disk drive. Nevertheless, when not in use, the disk drive should be turned off to prevent heat buildup.

A number of alernatives to the Commodore 1541 disk drive are available. Of these, the MSD super drive appears to be the most popular. The MSD drives are generally reliable and include an IEEE port along with the standard Commodore serial port. The MSD SD2 is a dual drive and can be used to make quick backups of unprotected disks, something that is burdensome with the Commodore 1541 single disk drive. With an optional IEEE interface adapter (which is manufactured by a number of companies), the MSD disk drives offer speed advantages over the Commodore 1541 disk drives; without an IEEE adapter, the MSD and the 1541 drives operate at approximately the same speed. I have not tested the Schnedler interface board in a C64 equipped with an IEEE interface adapter board, but, at a minimum, an additional power supply would probably be needed, since it is unlikely that enough power would be available from the C64 to drive both. A disadvantage of using disk drives other than the Commodore 1541 (or 1571, discussed below) is that a small number of commercial software packages will not boot with other drives because certain copy protection schemes rely on the presence of the 1541 disk operating system, which is proprietary.

The Commodore 1571 disk drive is a more recent entry than the 1541 and includes all of the capabilities of the 1541 (including the disk operating system) plus additional capabilities that can be accessed from a Commodore 128 computer only. The 1571 disk drive is a good choice for a researcher who uses a C64 but may later acquire a Commodore 128.

THE SOFTWARE

I have written a package of C64 programs to control psychological experimentation, collect responses, and perform preliminary data analyses. Many of the capabilities of the programs require installation of the Schnedler Systems interface board. In addition to obtaining these programs and the interface board, the user would need the Commodore 64 Macro Assembler Development System, or a compatible assembler. Also, a BASIC compiler (such as BLITZ!, distributed by Skyles Electric Works) would be necessary if the researcher wished to modify the program EXP64.

To prepare an experiment, the following sequence is followed:

1. The BASIC program—EXPSET—is run. EXPSET asks the experimenter about the experimental design and subject feedback and stores the responses to disk as a sequential file.

2. A short program specifying the sequence of events on an experimental trial (with the exception of the feedback, which is specified in EXPSET) is written in assem-

bly language using subroutines and macros contained in the library program EXP64/ML. The program is then assembled.

3. The experiment can now be run using the following procedures:

a. If a Cardco numeric keypad is to be used for collecting subject responses, the program KBASIC, which is packaged with the Cardco numeric keypad, should be run. After KBASIC is run, the keypad is considered to be an extension of the computer keyboard. Alternatively, there is a hardwire adapter available for the Cardco keypad which bypasses the need to load KBASIC.

b. The object code generated by the program that specified the events of a trial (Step 2, above) is loaded using the Commodore Assembler package loader.

c. The BASIC program (EXP64), which, among other things, oversees the randomization of trials, the recording of responses, and the presentation of visual feedback, is loaded and run. EXP64 initially asks the experimenter whether a variable file is to be loaded. At this point, the experimenter should answer ''yes'' and specify the name of the relevant sequential file that had been generated by EXPSET.

4. After the experimental session is completed, the experimenter is provided with (1) data about the number of correct responses, and (2) the option to store the data to disk.

5. If the data had been stored to disk, two programs—SUM64 and PCT64—are then available for preliminary data analysis.

Compiled versions of EXPSET, EXP64, SUM64, and PCT64 are available and should be used in place of the original programs unless the user wishes to change the code. It is particularly important to recompile EXP64 after any changes are made; otherwise, the program will run too slowly. More detailed information about each of the programs follows.

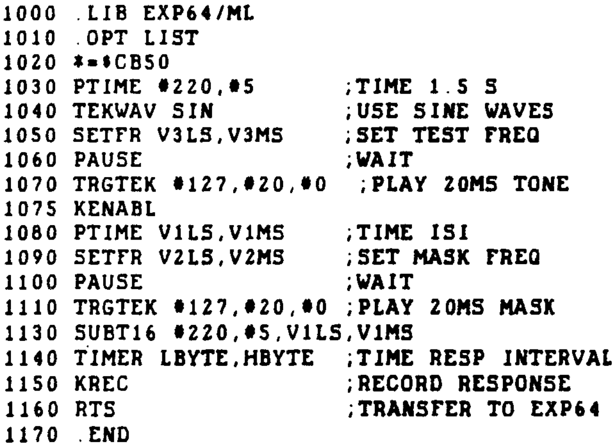
EXPSET

EXPSET creates a data file based on the answers it receives to questions about (1) the number of variables in the design, (2) the number of levels of each variable, and (3) the value of each level. Up to 5 variables may be specified, with a maximum of 10, 4, 4, 3, and 2 levels for Variables 1 through 5, respectively. When EXP64 is run at the time of the experiment, the variable values for each trial are chosen randomly from among those stored by EXPSET. As each trial begins, the variable values for that trial are passed to the machine language program, which controls the stimulus events up until the time of feedback. The variable values input to EXPSET should be values that are interpretable by the machine language program. EXPSET also solicits from the user information about any visual feedback that is to be presented to the subject following his/her response. If feedback is desired, the experimenter must specify (1) which variable feedback is contingent upon, (2) what the feedback should be, (3) the screen location fo the feedback, and (4) the duration of the feedback in milliseconds.

In a representative experiment, a short target tone was presented which had one of two frequencies (770 or 830 Hz). A variable silent interstimulus interval (30, 90, 170, or 300 msec) followed the target, and then a backward mask at one of three frequencies (740, 800, or 860 Hz) was presented. After a response interval of 1.5 sec (timed from the offset of the target), feedback about the correct response (LOW or HIGH) was presented for 500 msec. Based on user input, EXPSET stored the information that the first variable had four levels with values of 30, 90, 170, and 300, the second variable had three levels with values of 740, 800, and 860, and the third variable had two levels with values of 770 and 830. EXPSET also stored the information that feedback was contingent on the value of the third variable and would consist of either the letter string LOW or the letter string HIGH displayed for 500 msec at the center of the screen.

EXP64/ML

This program is actually a library of assembly language subroutines from which the experimenter may draw when the assembly language program specifying the events of an experimental trial is written. The library is too extensive to summarize here, but copies of the listing (available from the author) are extensively documented. A listing of the assembly language program used to conduct the experiment described above appears in Figure 1. The first line specifies that the routines in EXP64/ML should be assembled along with the code that follows. The machine code specifying the sequence of events on a trial always starts at memory location $CB50, hence the instruction *=$CB50. The macro PTIME times an interval with millisecond accuracy but allows other activities to occur during that interval. Following those activities, the PAUSE macro waits until the timed interval is completed before proceeding to the next instruction. The num-



```
1000 .LIB EXP64/ML
1010 .OPT LIST
1020 *=$CB50
1030 PTIME #220,#5          ;TIME 1.5 S
1040 TEKWAV SIN             ;USE SINE WAVES
1050 SETFR V3LS,V3MS        ;SET TEST FREQ
1060 PAUSE                  ;WAIT
1070 TRGTEK #127,#20,#0      ;PLAY 20MS TONE
1075 KENABL
1080 PTIME V1LS,V1MS        ;TIME ISI
1090 SETFR V2LS,V2MS        ;SET MASK FREQ
1100 PAUSE                  ;WAIT
1110 TRGTEK #127,#20,#0 ;PLAY 20MS MASK
1130 SUBT16 #220,#5,V1LS,V1MS
1140 TIMER LBYTE,HBYTE      ;TIME RESP INTERVAL
1150 KREC                   ;RECORD RESPONSE
1160 RTS                    ;TRANSFER TO EXP64
1170 .END
```

Figure 1. Listing of a program that specifies the events of a trial in a representative experiment. The details of the experiment are described in the text.

ber of milliseconds to be timed is held in 2 bytes. The second parameter represents the most significant byte and indicates the number of 256-msec intervals to be timed. The first parameter is the least significant byte and represents the number of additional milliseconds.

The TEKWAV macro sends ASCII code to the Wavetek oscillator to indicate the desired waveform. Once sent, the Wavetek retains this selection until a different TEKWAV command is issued. Similarly, SETFR sends frequency information to the Wavetek; this is specified by two parameter values, the first representing the least significant byte and the second the most significant. Note that the value of Variable 1, which is selected randomly for each trial by the EXP64 program, is written to memory locations V1LS and V1MS, which represent, in the EXP64 system, the least and most significant bytes for this variable. Similarly, SETFR V3LS,V3MS sets the frequency of the Wavetek tone generator to whichever value of Variable 3 had been selected for that trial. TRGTEK plays the tone for a specified number of milliseconds. The second and third parameters specify the tone's duration in milliseconds. The first parameter is a value that activates the Wavetek trigger line.

In order for a subject's keyboard response to register in the keyboard buffer, KENABL must be called. Responses then can be made anytime up to the KREC call, which puts the response into temporary storage. In its present form, EXP64/ML accepts binary responses only and the 1 and 2 keys are used for entering these responses; future revisions will expand these capabilities. After the response is recorded, control is transferred back to the BASIC program, which presents the visual feedback, stores the response (or lack of one if no response is made) according to the experimental condition, and transmits the variable information for the next trial.

Although this program example relies heavily on routines that service the Wavetek tone generator, other equipment that can be controlled by discrete voltages may be used with the EXP64/ML routines, assuming proper buffering of the Schnedler Systems interface board. For example, EXP64/ML has been used to control Coulbourn Instruments laboratory modules by writing appropriate values to memory location $DE11. In one application, PA0 was connected to the trigger line on a Coulbourn Instruments Model S84-04 rise-fall gate and PA1 was connected to a second Model S84-04 rise-fall gate. The gates were used to control which ear received a signal. By writing either 255 or 254 to $DE11 (recall that writing a 0 to a bit activated the output line), the experimenter could control the ear of stimulus presentation. Ear of presentation could be made a variable in an experiment by defining a variable that had two levels with values of 255 and 254. If the variable was defined as Variable 1 in the design, the commands LDA V1LS (load the accumulator with the least significant byte of the current value of Variable 1) and STA $DE11 (write the contents of the accumulator to $DE11) would serve to gate the stimuli to the ear chosen for that trial. In another instance, some of the lines controlled from $DE11 were routed to a Coulbourn Instruments Model S77-08 programmable digital-to-analog (D/A) converter whose analog output was used to control the frequencies generated by a Coulbourn Instruments Model S42-05 voltage-controlled oscillator. By writing the appropriate values to $DE11, the frequencies of the tones could be controlled.

EXP64

This program, written in BASIC, controls the sequence of trials, the recording of responses, the presentation of feedback, and other housekeeping chores necessary during the experiment. Unless the user wishes to modify the capabilities of the EXP64 system, EXP64 is loaded and run at the time of the experiment without any user modification. After loading EXP64, the experimenter is asked to specify the number of replications of the experimental conditions. The program then creates a random ordering of the experimental conditions. At the beginning of each trial, the variable values are passed to locations that are accessed by the machine language program after control is transferred over to it. After the response interval, the subject's response is stored by the machine language program to a specific memory location and control is transferred back to EXP64. The subject's response is then added to a response matrix which is organized by experimental condition, with Variable 1 being the fastest moving. For each condition, the number of "1" responses, "2" responses, and missed responses is recorded in order. EXP64 then controls the presentation of feedback for the trial and proceeds to the next trial. At the end of an experimental session, the response matrix may be stored to disk as a sequential file.

SUM64 and PCT64

SUM64 adds the matrices generated by EXP64 together and stores the resulting matrix to disk. SUM64 can be used to collapse a subject's data across sessions. PCT64 calculates percentages correct for each condition; trials on which subjects did not respond are disregarded. The resulting matrix is stored to disk and may be transmitted to a mainframe for further statistical analyses by such programs as BMDP. A Commodore VICMODEM and a terminal program that includes a send buffer—for example, VIP Terminal—is sufficient for this purpose.

FUTURE DEVELOPMENT AND CONCLUSIONS

One of the major limitations of the current laboratory is that subject responses must be binary. Future implementations of the EXP64 system will include the option of specifying a greater number of button responses. Also, as indicated above, the ability to record reaction times will be incorporated in future implementations. As an alternative to discrete responses, a continuous rating scale option is also in development. This option will allow subjects to rate stimuli on a continuous scale by responding on an appropriately labeled Koala graphics pad (manufactured by Koala Technologies Corporation). The responses would then be normalized and stored according to experimental condition. Other projects include the incorporation of the SID's capabilities into the EXP64 system and the use of computer-generated visual displays as stimuli.

Although the C64 has certain limitations, it can be used productively in the psychology laboratory particularly when equipped with an appropriate interface board. Due to its low price, the C64 should be of particular interest to researchers who must get by on limited budgets.

REFERENCES

Commodore Business Machines (1982). *Commodore 64 programmer's reference guide*. West Chester, PA: Author.

Czerny, P. (1979). Godot: A real-time data acquisition and control program for the PET microcomputer. *Behavior Research Methods & Instrumentation*, **11**, 577-580.

De Jong, M. L. (1982). *Apple II assembly language*. Indianapolis, IN: Sams.

De Jong, M. L. (1984). *Assembly language programming with the Commodore 64*. Bowie, MD: Brady Communications.

Dlhopolsky, J. G. (1984). *Software methods for experimental psychology*. Bayport, NY: Life Science Associates.

Durrett, H. J., Jr. (1978). Inexpensive microcomputer systems for research and instruction: A dream or reality? *Behavior Research Methods & Instrumentation*, **10**, 345-351.

Gordon, W. A., Foree, D., & Eckerman, D. A. (1983). Using an Apple II microcomputer for real-time control in a behavioral laboratory. *Behavior Research Methods & Instrumentation*, **15**, 158-166.

Leventhal, L. A. (1979). *6502 assembly language programming*. Berkeley, CA: Osbourne/McGraw-Hill.

Lugt, K. V. (1986). SID's programmable filter. *The Transactor*, **6**, 42-44.

Osaka, N. (1979). A microprocessor-based real-time BASIC laboratory: A pulse motor controlled visual stimulator. *Behavior Research Methods & Instrumentation*, **11**, 549-552.

Perera, T. B. (1981). A laboratory in experimental psychology using the TRS-80. *Behavior Research Methods & Instrumentation*, **13**, 195-197.

Perone, M. (1985). A software system for real-time laboratory use of TRS-80 microcomputers. *Behavior Research Methods, Instruments, & Computers*, **17**, 119-121.

Poltrock, S. E., & Foltz, G. S. (1982). An experimental psychology laboratory system for the Apple II microcomputer. *Behavior Research Methods & Instrumentation*, **14**, 103-108.

Reed, A. V. (1979). Microcomputer display timing: Problems and solutions. *Behavior Research Methods & Instrumentation*, **11**, 572-576.

Scanlon, L. J. (1980). *6502 software design*. Indianapolis, IN: Sams.

Schnedler Systems (1984). *6522 VIA guidebook for the Commodore-64*. Arlington, VA: Author.

Thompson, G. C. (1979). Behavioral programming with the APPLE II microcomputer. *Behavior Research Methods & Instrumentation*, **11**, 585-588.

West, R. C. (1985). *Programming the Commodore 64: The definitive guide*. Greensboro, NC: Compute! Publications.

Zaks, R. (1979). *6502 applications*. Berkeley, CA: Sybex.

NOTE

[1] The Schnedler Systems Model 64IF22 VIA board for the Commodore 64 may be obtained from Schnedler Systems, 1501 Ivanhoe Street, Arlington, VA 22205.