Behavior Research Methods & Instrumentation
1978, Vol. 10 (4), 468-473

The Commodore PET

Using personal computers for experimental control

ROBERT S. MCLEAN
Ontario Institute for Studies in Education, 252 Bloor Street West, Toronto, Canada M5S 1V6

The popularization of microcomputers makes available computer control of experiments at a price under $1,000. The Commodore PET is used as an illustration of the capabilities and problems an experimenter faces in instrumenting simple experiments with this equipment. Hardware interfacing is simple and the programming is done in BASIC. A sample experiment on instrumental conditioning is used as an example.

Advances in microelectronics have made possible the fabrication of the logic required to implement a computer on a "chip," a small component capable of mass production at low cost (see *Scientific American*, September 1977, for a description of this technology). While the computer "central processor" made this way may cost perhaps $25, additional logic and memory is required to make a computer that would be useful to the consumer. In addition, input and output devices suitable for the intended uses of the computer must be added.

The great bulk of microcomputer chips will be imbedded in other devices (such as automobiles or household appliances) and the user will not specifically be aware of the computer. However, several manufacturers are packaging microcomputers in a form resembling a computer terminal but containing the processor within the device. The microcomputer may be packaged with a television-type display, a keyboard, and a cassette tape recorder for storage of programs and data. Typically, it is programmed with a BASIC interpreter in read-only memory and is ready for use when it is plugged into an electrical outlet and switched on. The intent of this device is to provide an individual with a computer that can be used for relatively simple computations, data processing, and game playing. Because of this nature, these machines have been called "personal computers," and a new field of leisure activity has arisen called "personal computing."

Two early examples of personal computers are those by Radio Shack and Commodore. This report examines a first attempt to utilize the PET ("Personal Electronic Transactor") by Commodore for the purpose of controlling a simple psychological experiment involving variants of instrumental conditioning. I will discuss the hardware that must be added to the PET to allow control of external devices. The program that controls the experiment will be discussed in detail, along with some of the programming conventions that are required but not adequately discussed in the PET user's manual.

CHARACTERISTICS OF THE PET

There are a number of characteristics of the PET that help the system assume its (presumably unintended) role as a laboratory control device.

Programming

The PET is programmed in the BASIC language, with its system programming built-in, in read-only memory. This means that it is ready to function when turned on, without any complicated start-up procedures. The version of BASIC is relatively powerful, including string variables, string functions, and arithmetic functions not always found in microcomputer BASIC implementations. A certain amount of graphic display capability is included, with graphics characters that can be used to create displays. This feature can be used in many types of experiments, such as concept learning and problem solving.

Several BASIC commands are of particular use for manipulations that cannot be done directly in BASIC; these provide access directly to the microcomputer itself. PEEK (A) is a function that allows the BASIC program to get the value stored in memory location A. A similar command, POKE A,B stores value B into memory location A. Since the MOS Technology 6502 microprocessor used in the PET employs memory locations for interfacing to peripheral devices, these commands are essential to communicate with external devices. In addition, it is theoretically possible to implement machine-language subroutines and execute them from BASIC. This approach requires a relatively sophisticated knowledge of the 6502 and is not needed for the work reported here.

Clock

The PET has a real-time clock, accessible in two forms from BASIC. It increments in 60ths of a second, presumably as a by-product of the television display. According to the user's manual, it is crystal controlled

```
 5    P=59459
10    POKE P,1
20    POKE P,0
30    GO TO 10
```

Figure 1. Clock timing task.

and, hence, not derived from the power line. The BASIC variable TI contains the value of the clock in 60ths ("jiffies"). TI$ is a string variable that contains six digits that represent the current hours, minutes, and seconds. Both of these increment automatically through the use of interrupts built into the monitor system. This is an aid to the user, since he/she need not be concerned with this technicality. It is a minor problem in that interrupts imply that the timing of short pieces of code will not be constant. Thus, a program loop to output a square wave at one of the output ports (shown in Figure 1) took 5.5 msec to execute the main loop of Statements 10-30 most of the time, but appeared to take .5 msec longer some of the time, presumably once each jiffy, so that the interrupt code could be processed.

Additional timing information is given in the first users' group newsletter (Commodore, Note 1).

Input/Output Connections

The PET provides four connectors for attachment of additional devices.

(1) External devices employing the IEEE-488 standard protocol can be connected directly. This is the protocol employed by Hewlett-Packard and others. It is a moderately complex interface with extensive handshaking over the data bus. There exist many laboratory devices that conform to these specifications, but they are mostly expensive instruments that overshadow the economics of a personal computer.

(2) An 8-bit parallel port, directly connected to half of a 6522 Parallel Interface Adaptor (PIA), is probably more useful and can be used for either input or output on a bit-by-bit basis. If the control requirements of an experiment can be accommodated in some combination of input and output not exceeding 8 bits, this is the simplest connection. This is the connector used in the example to be reported below. The connector also gives a few signals that allow one to connect another television monitor for group display of the PET's screen, for use in classroom situations, for example. (Caution: It appears that the display signals are not the normal composite video normally provided to a TV monitor, but separate sync and video that may have to be mixed for use with a monitor.)

(3) A connection for a second tape cassette is provided. At this writing it appears that no cassette on the market can be used with this connection due to the particular recording scheme used by the PET. Commodore will provide an optional tape cassette for this connector in the near future, and perhaps independent companies will also provide such a device.

(4) A final connector brings out the memory data and address buses, so that additional memory or other peripheral devices can be attached. Use of these signals requires knowledge of the microcomputer hardware and is best left to the experienced user. This connection is not needed for simple control problems such as the one reported here.

A SAMPLE EXPERIMENT

The experiment to be discussed is one investigating learned helplessness (Baumal, Note 2). The design calls for yoked quartets of subjects. The first subject of the quartet receives a number of trials in which an aversive stimulus (noise in this case) can be escaped by pressing a button twice. The other three subjects of the quartet receive the same stimuli as the first one obtained through his/her escape behavior. A pair of lights provide feedback to the subject as to whether he/she succeeded in terminating the noise or whether it was terminated by the machine.

A test phase follows the training trials, in which all subjects can learn to avoid or escape the aversive stimulus (noise again) through appropriate manipulation of a human shuttlebox. Escape or avoidance involves moving the handle of the shuttlebox to alternate ends of the box on alternate trials.

Data collected include the number of buttonpresses during the presentation of noise in the training trials, latency until escape during the training trials, buttonpresses per second during 5 sec before noise and 2 sec after noise during training trials, and latency of escape/avoidance during the test trials. The design is summarized in Figure 2.

HARDWARE

The experimental hardware is shown schematically in Figure 3. Since this experiment was a partial replication of previous research, existing equipment was used. This accounted for the fact that the two lights used in the training trials were 24-V dc lamps, while the warning light for the test trials was a 120-V 3-W lamp. The noise was controlled by a relay which interrupted the earphone circuit.

Connection to the PET is accomplished through the PIA connector that yields 8 bits of information exchange, the direction of which can be set under program control. The PIA signals are logic signals that are TTL compatible and referenced to the signal ground, which is also available on the connector. No power is available from the PET and any external logic must supply its own. A simple dc power supply (even the type used to substitute for batteries in a tape recorder or radio) and a 5-V regulator of the 7805 type supply enough power for a typical interface unless lamps or relays are to be driven by the logic. In this case, a larger 24-V power supplies the lamps and relay, with a 7805 providing logic power.

Training trials (40)

for first subject:	for others:
Monitor buttonpressing behavior for 5 sec.	Same.
Turn on noise.	Turn on noise.
After two buttonpresses or 5 sec turn off noise.	Turn off noise after same time as first subject had.
If terminated by button, show blue lamp, or else show red lamp, for 2 sec.	Show same lamp as first subject saw.
Wait random ITI.	Same as first subject had.

Break time during which noncomputer procedure is followed for all subjects.

Shuttlebox trials (20) for all subjects:

Shuttlebox light on for up to 5 sec. (Terminate when handle moved to correct end of shuttlebox.)
Present noise for 10 sec or until subject responds correctly (omit if correct response during light).
Wait remainder of the 10 sec allocated to noise.

Figure 2. Experimental design.

While a simple switch and resistor combination can be used to provide response input, a stage of logic isolation should be provided between the subject and the PET's PIA. Among other reasons, the PIA can be damaged by static discharges; the interposed TTL device provides protection against that happening. In this case, one pushbutton and two microswitches were to be connected. Each switch was pulled up to 5-V by a 1K resistor and input to a section of a 7414 hex Schmitt trigger. The outputs of the 7414 were connected to the low-order bits of the PIA. This arrangement gives an inversion to the switch signal, such that a closed switch (to ground) is seen as a "1" by the PET program.

Logic isolation will always be required on the output signals. The experiment required three types of control. An audio signal from a hearing test apparatus is controlled by a relay, in order to overcome any considerations about the type of signal being switched. The relay is driven by a 75451 integrated circuit driver that is also useful for driving lamps rated at up to 30-V and 250 mA. If the resulting sense of the lamp or relay is not convenient (i.e., if a "0" from the computer turns on the lamp or relay), the output signal can be inverted with a 7404 inverter first, as was done in the case of the relay. Alternatively, a 75452, which functions as an "and" rather than a "nand" device, can be used.

For controlling voltages, a relay can be used, but there are now logic-controlled optically coupled Triacs available. In this application, Motorola MOC-3010 or 3011 was used. It has a Triac usable up to 250-V and 5 W; it can be used also to switch another Triac with higher ratings. The light-emitting diode of the 3010/3011 can be driven by the output of a TTL gate, as is shown.

PROGRAMMING

The program for the experiment, shown in Figure 4, consists of the following main components:

(1) A main loop (lines 40-204) indexed by Q, which cycles through the two parts of the experiment for each of the four subject members of the quartet.

(2) The training trials (lines 50-119) during which the first subject (Q = 1) establishes the conditions for all subjects by learning to escape the noise. He/she also determines the feedback received by all four subjects, that is, whether the red or blue light comes on in line 88.

(3) There's "break time" (lines 130-134) between the training and test trials, during which there are some

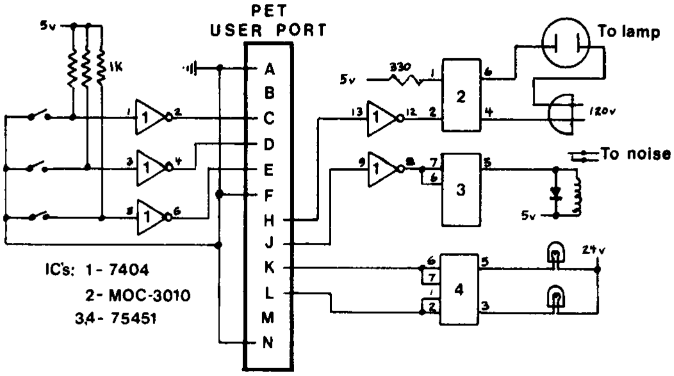

Figure 3. Experimental hardware.

```
Program runs on 8K PET; listing was transcribed by hand....
Strange characters in print statements are cursor controls.
First, make sure program tape is removed; then get time
and date and open a file for the data.
  1      PRINT "*QQ REMOVE PROGRAM TAPE
  2      PRINT "QQ THEN TYPE 1 AND PRESS RETURN
  3      INPUT A
  5      PRINT "*Q ENTER TIME (HHMMSS), DATE
  7      INPUT T$,D$
  8      TI$ = T$
 10      DIM W(45), L (45), Z (45), Y(30)
 11      A = 59471
 20      OPEN 1,1,2, "DATA" + D$
 21      POKE A-12, 240
 22      POKE A,0
 24      PRINT #1, ">>>> DATA" TI$; D$
This is the start of a loop that runs a quartet. The first
subject's performance (Q=1) is stored in the arrays and
determines the stimuli for the other 3 subjects.
 40      FOR Q = 1 TO 4
 42      PRINT "*QQQ START OF EXPERIMENT QQQ CONDITION = "Q
 43      PRINT "QQ INPUT SUBJECT IDENTIFICATION Q
 44      INPUT S$
 46      PRINT #1, "*****" TI$; s$
The following loop is executed for the 40 training
trials. (Loop ends in 119.)
 50      FOR T = 1 TO 40: PRINT "TRIAL" T " 0
 52      IF Q > 1 GO TO 60
For the first subject of the quartet only, the stimulus
conditions are set up here. L(T) contains the maximum length
of the tone; it will be set to the actual length in 86 if
S is able to turn tone off. W(T) is set to a random number of
jiffies to cause an intertrial interval up to 10 seconds.
Z(T) contains the light to turn on (64 if tone not turned off).
 54      L (T) = 300: P = 0: W(T) = 600*RND (1)
 56      Z (T) = 64
 60      J = 1: S = TI + 300: F = S + L (T): X = TI + 60
 62      B = 0: N = 0: M = 0
Subroutine at 980 looks for a response (which it counts
in N) and also records number of responses each second in Y(J).
 70      GOSUB 980
 72      IF TI < S GO TO 70
 76      POKE A,32
 78      N = 0
 80      GOSUB 980
 82      IF TI > F GO TO 88
 84      IF Q > 1 OR N < 2 GO TO 80
 86      L (T) = TI - S: Z (T) = 128
 88      POKE A, Z(Y)
 90      V = TI + 120
 92      GOSUB 980
 94      IF V > TI GO TO 92
 96      POKE A,0
100      M = TI + W(T)
101      PRINT "]]]]]]]" Z(T); L(T); N
102      PRINT #1, "T" T; Z(T); L(T); N
104      FOR I = 1 TO J: PRINT #1, Y(I); : NEXT I
108      PRINT #1
110      IF TI > M THEN PRINT "ITI ERROR", TI-M
112      IF TI < M GO TO 112
119      NEXT T
```

```
This is the end of the 40 training trials; the experiment
now has a break during which some activities occur
which are not controlled by the computer. Computer control
resumes wnen a button press is detected in line 134.
130      PRINT "*QQQQ BREAK TIME
131      PRINT "QQ Push button to continue QQQQ
132      PRINT TI$,  "0
134      IF (PEEK (A) AND 4) = 0 GO TO 132
140      PRINT #1 "++++" TI$;" START SHUTTLE BOX TRIALS
Loop for shuttle box trials (loop ends in 185).
141      PRINT "*QQQQ SHUTTLE BOX TRIALS QQ
150      FOR T =1 TO 20
151      PRINT "TRIAL"T"0
152      E = (1 AND T) +1
154      N = 32: M = TI+300: F = TI
Turn on light on shuttle box.
156      POKE A,16
170      X = 3 AND PEEK (A)
E contains correct switch value (which alternates on successive
trials). If the handle has been moved to the correct end of
shuttle box, noise is avoided by going to 178.
171      IF X = E GO TO 178
172      IF TI < M GO TO 170
Subject has not avoided, so turn on noise.
173      POKE A, 32: M = TI+600
174      X = 3 AND PEEK (A)
If subject responds correctly, he escapes from noise at 178.
176      IF X = E GO TO 178
177      IF TI < M GO TO 174
At 178, get time since trial began, turn off light
or noise, and set M for intertrial interval of 2 to 7 seconds.
178      F = TI-F : POKE A,0 : M = TI+120+300*RND(1)
179      PRINT "]]]]]" F; E; X
180      PRINT #1, T, F
181      IF TI < M GO TO 181
185      NEXT T
This is the end of the loop. Now there is a short
pause and then the program goes on to the next subject in quartet.
200      PRINT "*QQQ END OF SUBJECT" S$ : M = TI
202      IF TI < M +  600 GO TO 202
204      NEXT Q
210      PRINT "*QQQ END OF QUARTET
211      PRINT "  ...Do you want another run (Y or N)
212      INPUT A$* IF A$ & "Y" GO TO 48
215      IF A$ <> "N" GO TO 210
219      PRINT #1, "END
220      CLOSE 1
221      STOP

Subroutine 980 is used to look for responses and to record
the number of responses per second in Y(J).
980      IF (4 AND PEEK (A)) = 4 GO TO 984
982      M = 0: GO TO 990
984      M = M +1: IF M <> 2 GO TO 990
986      B = B +1: N = N +1
990      IF TI < X THEN RETURN
992      X = X + 60: Y (J) = B: B = 0: J = J + 2
994      RETURN
```

Figure 4. Program to run a quartet of subjects.

experimental manipulations that are not under computer control. The time is displayed during this interval.

(4) The test trials (lines 140-185) with the shuttlebox and escape/avoidance of the noise follow. The remainder of the program deals with such matters as data recording on the cassette, informing the experimenter of the progress of the experiment, etc.

The yoking of subjects in Conditions 2-4 is accomplished through the retention of the first subject's performance in the data arrays given in the DIM statements (line 10). W saves the intertrial interval (computed randomly during the first S) L saves the duration of the noise in jiffies (5 sec if no escape), and Z saves the light that was turned on (64 = red light, 128 = blue light). Y is used within each trial to record the number of buttonpresses during each second (line 992) and is output to tape at the end of the trial (line 104).

During the training trials a subroutine (lines 980-994) is called repeatedly to determine if a button has been pressed and whether a second has elapsed. In line 980 the PIA is consulted with a PEEK statement. If the bit to which the pushbutton is connected is a "1" (giving value 4 because it is Bit 2, giving a binary value of 2^2 or 4), the value M is incremented; otherwise, M is reset to zero. If M = 2, the button is considered pushed; through this technique the button is "debounced" by requiring that it be on during two consecutive readings. The number of buttonpresses during a second is recorded in B and N, in line 986. Approximately each second (lines 990-992), the value in B is transferred to the Y array for recording at the end of the trial. N is used in the main loop for determining the number of buttonpresses while the noise is presented (and, in Condition 1, for terminating the noise when two presses have been made).

There are a few programming techniques and requirements that are not directly evident to the person familiar with traditional BASIC. Some of the following are not even evident in the "user manual" that accompanies the PET. These are discussed in the order that they occur in the program.

In line 1 the inclusion of control characters in the literal to be printed on the screen provides for the clearing of the screen and positioning of information. They function like the form feed and line feed on normal terminals.

The OPEN statement (line 20) specifies (in the order of the parameters) that the file to be referred to as 1 in print statements (e.g., in lines 46, 108, etc.) is to be opened on Device 1 (the cassette recorder–the only device that can be opened on the basic PET), for Command Type 2 which is writing (PRINT). The OPEN statement further specifies that the file is to have the file name "DATA" plus whatever string was typed in Statement 7. This procedure allows the data to be PRINTed onto the cassette with the PRINT 1 commands. The file remains "open" until the CLOSE 1 statement is executed in line 220.

The PIA interface that connects the experimental equipment is accessed by PEEKing and POKEing its registers. Two registers are of interest for simple experiments. The data is transferred in the register at an address that translates into 59471 decimal; for convenience, the variable A has been set to this value in line 11. All data bits are initially assumed to be input by the automatic power-on procedure.) Those that are to be output must have a "1" bit in the "data direction register," located 12 (decimal) locations below the data register. Thus, line 21 sets the four high-order bits for output, since 240 = 128 + 64 + 32 + 16, the binary weights of these bits converted to decimal.

Timing of the experiment is accomplished in "jiffies," that is, 60ths of a second. To obtain a time delay–for example, the 5 sec during which buttonpressing is monitored but no stimulus is presented–the starting value of TI is added to the length of the period in jiffies and saved in a variable. The variable is then compared with the value of TI. If TI is less, the program kills time in a loop, returning to this test. If TI is equal or greater, the interval is finished and the program can go on. Examples are found in lines 70-72, 80-84, 92-94, 112, and 170-174. In some of these, a call to the subroutine at line 980 repetitively checks for responses. This procedure seems to allow about 30 checks for stimuli per second. This is certainly not millisecond accuracy, but probably more accurate than an experimenter with a stopwatch. Much more sophisticated programming is needed to enable responses to interrupt the program and always be processed during the jiffy in which they occur.

Recording of data from the experiment onto the cassette tape has been programmed to occur during the intertrial intervals (lines 102-110). Information to be recorded is buffered in memory until a block of information is ready to be written. At that time the cassette motor is started and the block is written. Information is not double buffered, so processing ceases while the block is written. Since it is difficult to predict exactly when this mechanical action will occur, there is an uncontrollable variation in the progress of the program when writing to tape. For this reason, line 110 was included to give the experimenter some indication of the frequency of occurrence of delays which lengthen the calculated intertrial interval. The intertrial interval is not particularly critical in the experiment and the infrequent occurrence of this "error" was not considered important.

In certain cases, one can perform logical manipulations with information; in other, semmingly similar cases, one cannot. Obviously, the examples given (e.g., in lines 152, 170, 980) work. Certain other orderings of the variables in these expressions yield syntax errors. If the coding in lines 170-174 seems more involved than necessary, it is because it is the result of such a try-and-revise process of permuting the statement components. Another factor is that combining of statements on one line separated by colons does not work after IF statements.

ASSOCIATED PROGRAMS

Two other programs are of utility with this experiment. One (see Figure 5) provides a test of the equipment and gives the experimenter confidence that all is ready for a day's experimentation. The other (Figure 6) provides a means for reviewing the information recorded on the cassette tape during the experiment. Other programs would have to be written to analyze the experimental data, if that is to be' done on the PET. These are beyond the scope of the present discussion.

```
 1 PRINT "*TESTS OUTPUT -- TYPE IN BITS TO
 2 PRINT "BE OUTPUT QQQQ
 3 PRINT " 16 - SHUTTLE BOX LIGHT
 4 PRINT " 32 - AUDIO (NOISE) CONNECTION
 5 PRINT " 64 - RED LIGHT
 6 PRINT "128 - BLUE LIGHT
 7 PRINT "  TYPE 1 TO GO TO INPUT TEST
 8 POKE 59459,240: POKE 59471,0
 9 PRINT "EVERYTHING SHOULD BE OFF NOW.
10 INPUT A
11 IF A = 1 GO TO 40
20 POKE 59471,A
30 GO TO 10
40 PRINT "INPUT TEST
45 S = 7 AND PEEK (59471)
50 PRINT "SWITCHS ON: " S "O
55 GO TO 45
```

Figure 5. Equipment test program.

```
10 OPEN 1, 1, 0
20 INPUT #1, A$
30 PRINT A$
40 GO TO 20
```

Figure 6. Tape listing program.

EXPERIENCE TO DATE

This paper reports my initial experience with the instrumentation of an experiment using a "personal computer." The PET is only one of several microcomputers that might be suitable for research. The experience of actually using the PET is based on only one machine, and generalizing beyond the one machine may not be warranted.

At this writing, the program described has been used by two experimenters doing research in an elementary school setting. These experimenters have no previous computer experience and are naive with respect to computer programming. Both report no problems with setting up or running the experiment, with the exception of the following noted computer problems.

For experimental control, a microcomputer system should be reliable. It should continue to carry out the experiment without hardware or software interruptions at nonanticipated times. The experiment described here requires error-free performance for continuous periods of about 3 h per quartet. This is not a particularly long period of operation for an experiment, considering that some studies may continue for weeks.

We experienced several instances in which the computer program stopped working, without any error indication. After the first few instances all components in the PET were reseated in their sockets and a possibly defective memory chip was removed, as prescribed in the PET user's manual. Since then, this problem reoccurred once in 2 weeks of experimentation.

It is difficult to trace the source of this problem since no error indication is given and the only restart procedure provided is to power-down the machine, a procedure that destroys any information in the machine's memory. The only recourse in this case is to salvage the data already collected on tape, which may allow additional subjects to be tested in the yoked condition.

There have been considerable problems with the tape cassette unit. The program loading procedure has not always been successful: There is some indication that under some circumstances a program can appear to load correctly but, in fact, be incorrect in memory. This can be disastrous, particularly if some parts of a program are used only at the end of the experiment. This problem seems to have diminished after reseating the chips.

Tape problems continue into the retrieval of data from the experiment. To date, the data-reading procedure has been marred by the frequent skipping of blocks of data on tape. This is overcome in this instance by the experimenter recording on paper the important data as insurance that there will be data to analyze. Of course, this is not satisfactory as a general solution.

It must be emphasized that these problems are based on one early experience with a single machine. Generalization to other personal computers is not warranted, except to observe that these problems certainly may occur. The typical researcher has only one machine, particularly when starting out. It is unlikely that the experimenter experiencing such problems will have much support in trying to trace and/or correct the problems should they occur.

REFERENCE NOTES

1. Commodore. *The transactor* (Bulletin 1). Palo Alto: Commodore Systems, 901 California Avenue, April 30, 1978.

2. Baumal, R. *The development and reversibility of learned helplessness.* PhD dissertation in progress, University of Toronto, 1978.

(Received for publication May 23, 1978;
accepted June 27, 1978.)