Pharmacology Biochemistry & Behavior, Vol. 14, pp. 415–417. Printed in the U.S.A.

A Microcomputer System for the Control of Behavioral Experiments

MARILYN E. CARROLL,[1] PETER A. SANTI AND RICHARD L. RUDELL

University of Minnesota, Minneapolis, MN 55455

Received 30 June 1980

CARROLL, M. E., P. A. SANTI AND R. L. RUDELL. *A microcomputer system for the control of behavioral experiments*. PHARMAC. BIOCHEM. BEHAV. 14(3) 415–417, 1981.—A self-contained microcomputer control system (MCS) is described for controlling and recording events from behavioral experiments using a low-cost microcomputer, the Rockwell AIM 65, and the BASIC language. Features of the system include a printer, alphanumeric display, 4K RAM, 8K BASIC on read only memory (ROM), a cassette interface, a power supply, and a control interface, all contained in a compact enclosure. The control interface allows for 16 inputs and 16 outputs, and includes a real-time clock and versatile machine language subroutines for controlling and recording real-time events. These features make BASIC a suitable user language for this application. The use of this microcomputer control system for operant research in behavioral pharmacology is described.

Microcomputer Operant research BASIC

WITH increasing costs of traditional modular electromechanical and solid-state equipment, the microcomputer has become an attractive alternative for programming and recording behavioral events. The use of inexpensive microcomputers, each dedicated to only one or a small number of experimental chambers, also offers an advantage over the large laboratory minicomputer systems in which the functioning of the entire laboratory is dependent upon the well-being of a single computer. In addition to cost, there are other problems which the behavioral scientist must attend to before considering a microcomputer system. First, a reliable, interface must be available to connect the computer to the experimental apparatus. Second, the microcomputers should be durable enough to withstand continuous laboratory use. Finally, it would be an advantage to program a laboratory microcomputer using a general-purpose, high-level language such as BASIC. There are several reports from laboratories where microcomputer systems have been successfully developed (e.g. [1, 4, 5]). However, the wide-spread use of microcomputers in the behavioral labortatory will depend upon the availability of a commercially-available turn-key system which includes the microcomputer, interface, software, output and storage peripherals, and documentation.

The purpose of this paper is to describe the use of such a microcomputer system in a behavioral pharmacology laboratory. The microcomputer control system (MCS) [2] described in this report consists of a Rockwell International AIM 65 microcomputer with 8K BASIC on read only memory (ROM), 4K bytes of static random access memory (RAM), a full-size keyboard, printer, display and a cassette interface. In addition, the MCS contains a DC control interface, two other interface boards and a 2K Interrupt Service Routine (ISR) on ROM which conducts all timing operations with machine language subroutines [2]. All of these components are contained including the power supply in a small plastic enclosure about the size of a portable typewriter [3].

FEATURES OF THE MICROCOMPUTER CONTROL SYSTEM

Hardware

The microcomputer control system (MCS) was specifically designed to control and record behavioral events within operant chambers. This system is based upon the Rockwell AIM 65. However, the ROM BASIC (by Microsoft®) execution speed is too slow to control real-time events, and the input and output lines on the AIM 65 are not designed to control 28 V DC equipment.

In order to make the AIM 65 suitable for programming and recording behavioral events, the MCS contains several interface devices [2]. There is a printed circuit board (DC control board) which provides optical coupling between the low voltage (5 V DC) computer signals and higher voltages (e.g., 28 V DC). This portion of the interface also contains 16 input channels and 16 output channels. A second interface board allows for plug-in connections to and from a standard cassette tape recorder for program storage and loading. There is also a 20 mA or RS232 serial line for communication with peripheral devices such as terminals, modems, larger printers and other computers. The third interface board allows for expansion of the MCS by the addition of peripheral devices such as memory boards or floppy disc controllers.

[1] Send reprint requests to: Dr. Marilyn E. Carroll, Psychiatry Research Unit, Mayo Box 392, University of Minnesota, Minneapolis, MN 55455.

TABLE 1

ISR Commands	ISR Functions
USR (−1)	Initialize the ISR and start the clock.
USR (0)	Turn off all output devices.
USR (1)	Check for an active input (0=none, >0=device number).
USR (2)	Time period (0=none, >0=time period number).
USR (3)	Report elapsed time (seconds and milliseconds).
USR (4)	Load hours (HH), minutes (MM) and seconds (SS) into TI$.
USR (5)	Load the month (MM), day (DD) and year (YY) into DT$.
USR (6). . .USR (25)	Loads up to 25 time periods. (e.g., X=USR (6):X=USR (9.5):X=USR (14):X=USR (25). This statement designates 9:30 a.m. and 2:00 p.m. as time periods 1 and 2. USR (6) and USR (25), respectively, start and stop the loading process.
USR (100+OD)	Turn output device (OD) on where OD=the number of the output device (e.g., X=USR (103) turns output device #3 on).
USR (200+OD)	Set output device (OD) off (default).
USR (300+OD)	Enable pulse stream to output device (OD).
USR (400+OD)	Disable pulse stream to output device (OD).
USR (500+OD.XX)	Set pulse rate on output device (OD) where XX= duty cycle. (e.g., 502.5 assumes a duty cycle of .5 on device 2)
USR (600+ID)	Check if input device (ID) is active where ID= the number of the input device.
USR (700+OD.ID)	Activate output device (OD) if input device (ID) is active. (Used for FR 1 schedules)
USR (800+ID)	Set input device (ID) to be normally open.
USR (900+ID)	Set input device (ID) to be normally closed.
POKE 232, HY	Enables hypertime. The clock speed increases HY times. This allows for testing a long session.
POKE OT+OD,X	Toggles output device OD for X times 10 msec. For example, POKE OT+3, 10 turns output device 3 on for 100 msec.

Software

In addition to the hardware interfaces the MCS provides software on ROM which allows BASIC to efficiently process real-time events. This is accomplished by a collection of machine language subroutines referred to as the Interrupt Service Routine (ISR). The ISR executes all timing operations and processes inputs and outputs. It relieves BASIC of these time consuming operations so that BASIC can be used for logic, counting, printing, and data handling functions. The ISR is activated by a single BASIC command. Once activated, it performs continuously and transparently to the user. The ISR works on an interrupt mode: the microprocessor unit (6502) is interrupted every 10 msec and the functions of the ISR are executed. These functions include an update of the real-time clock, recording elapsed time, recording inputs, turning on or off output devices and providing the user's BASIC program with the time and state of input and output devices. Use of the ISR does not require knowledge of machine language; it is accessed by a few simple BASIC commands which are summarized in Table 1.

Application

Eight Microcomputer Control Systems are used in our laboratory to study oral drug self-administration in rhesus monkeys. Currently one MCS controls a test panel in one chamber, although the MCS could control two or more chambers, depending upon the number of inputs and outputs. Each system costs approximately $1900, which is one third to one half the cost of solid state or electromechanical (relay) equipment depending upon the complexity of the schedule. The input devices on the monkey test panels consist of two lip-contact drinkometers, three levers and a feeder report. The outputs from the MCS consist of a feeder operate, solenoid operates on two drinking devices, several stimulus lights and three pens and the motor start on a standard Gerbrands cumulative recorder.

Several behavioral schedules are used in our experiments. During test sessions there is usually a concurrent FR schedule for liquid (drug and water). For part of the session food pellets are available on a DRL differential reinforcements of low rates) or multiple FI (fixed interval), FR (fixed ratio) schedule. Session length ranges from 3 to 23.5 hr and the intersession food and liquid availability are also controlled by the MCS. Data are automatically recorded continuously and printed at 1, 5, 10 or 15-minute intervals during the session and each hour during the intersession period. The average BASIC program for a relatively complex behavioral schedule ranges from 50–80 lines. The following is an example of the logic from a portion of a program that implements an FR schedule:

```
 90  FR=16
100  ID=USR (1)
110  IF ID<>1 THEN 100
120  R=R+1:R1=R1+1
130  IF R1<>FR THEN 100
140  POKEOT+3,10:D=D+1:R1=0:GOTO 100
```

This program checks for an active input (line 100). If input device 1 has been activated, a response is counted (R1) and cumulative responses (R) are calculated. If the number of responses (R1) equal the FR value, output device 3 is turned on for 100 MSEC (line 140), reinforcements (D) are counted and the response counter (R1) is reset. An FR 1 schedule could be implemented by a single ISR function and one statement:

```
500  X=USR(701.03)
```

This statement activates output 1 for each activation of input 3. A variable ratio (VR) schedule would be similar to the preceding FR schedule, except that VR would be determined by the random number function of BASIC. The following line 80 would be added and 90 replaced by lines 90, 130 and 140 to change the preceding FR 16 schedule to a VR:

```
 80  A=30:B=120
 90  VR=INT(B-A)*RND(1)+A+.05
130  IF RI<>VR THEN 100
140  POKEOT+3,10:D=D+1:R1=0:GOTO 90
```

The variables A and B indicate a minimum and maximum ratio value and a random integer value is calculated within that range in line 90.

The fixed interval (FI) schedule, as used in our experiments also requires only a few BASIC statements:

```
 90  FI=60
100  T=USR (3)
110  ID=USR(1)
120  IF ID<>4 THEN 110
130  R=R+1:ET=USR (3)-T
140  IF ET<FI THEN 110
150  POKEOT+12,50:D=D+1:GOTO 100
```

The FI is defined as 60 sec (line 90). The elapsed time counter is checked (line 100). If an input is received from device 4 (line 120) a response (R) is counted and the elapsed time is checked and compared to the FI. If the elapsed time is greater than the FI value, output device 12 is activated for 0.5 sec; reinforcements are counted and the elapsed time counter is reset (lines 150 and 100). This schedule is changed to a variable interval by adding statements similar to lines 80 and 90 in the VR program. Furthermore, this program is changed to a DRL schedule by changing line 140 to: IF ET<FI THEN 100.

When DRL or interval food schedules are used in our experiments, IRT distributions are obtained to a provide a measure of the effects of drugs in schedule-maintained performance. The following is an example of a portion of a BASIC program that calculates IRT's:

```
20  ID=USR (1)
30  IF ID<>1 THEN 20
40  T=USR(3)
50  ID=USR (1)
60  IF ID<>1 THEN 50
70  IRT=USR(3)-T:PRINTIRT:GOTO 40
```

This program checks for an input on device 7 (line 30) if an input is received (line 40), T is assigned the value of the current elapsed time. If another input is received (line 50), the IRT is calculated by subtracting the value T from the current elapsed time count (USR (3)). The counter is reset (line 70) and the computer checks for another input.

Once written, these program segments can be used repeatedly in different combinations to form a variety of simple or complex schedules. This modular style of programming results in greatly reduced programming time with each successive program. Another efficient aspect of a microcomputer control system is that programs can be changed quickly. If an additional MCS is available for writing and testing programs, programs can be saved on cassette tape and loaded into a dedicated MCS in less than one minute. The use of cassette tapes for program storage also allows several vastly different programs to be executed in a given chamber each day.

REFERENCES

1. Dillon, R. F., B. Millman, J. W. Tombough, W. R. Ferguson and W. R. Bezanson. A microcomputer-controlled laboratory: Hardware. *Behav. Res. Meth. Instrum.* **11:** 293–300, 1979.
2. Micro Interfaces, Inc., P.O. Box 14520, Minneapolis, MN 55414.
3. *The Computerist*, 34 Chelmsford Street, Chelmsford, MA 01824.
4. Thompson, G. C. Behavioral programming with the APPLE II computer. *Behav. Res. Meth. Instrum.* **11:** 585–588, 1979.
5. Tombough, J. W., R. F. Dillon, B. Millman and W. R. Bezanson. A microcomputer-controlled laboratory: Software. *Behav. Res. Meth. Instrum.* **11:** 301–310, 1979.