Measuring reaction time without measuring movement time: A modification of the Hormann and Allen millisecond timer for the Commodore 64 or 128

CHRISTOPHER J. LEE and LINDA T. MILLER
University of Western Ontario,
London, Ontario, Canada

It is frequently desirable to measure reaction time independently of movement time. The Hormann and Allen (1987) millisecond timer for the Commodore 64 or 128 can be modified to allow the measurement of reaction time with minimal movement time. A BASIC loader is provided for a keyboard version and an external switch version of the modified timer. Machine language code is provided for use with MONITOR, the machine language utility built into the Commodore 128.

In this paper we describe a modification of the Hormann and Allen (1987) millisecond timer for the Commodore 64 or 128. The Hormann and Allen timer measures the interval of time between the presentation of a stimulus on a computer monitor and the press of either of two function keys in response to the stimulus. This measure of reaction time involves a movement-time component, which may be undesirable in some experimental tasks. As described below, a small modification to the Hormann and Allen timer routine allows the measurement of reaction time with minimal movement time.

The magnitude of the movement-time component in the original Hormann and Allen (1987) timer depends on the instructions given to subjects. When subjects are instructed to rest one hand on one key and the other hand on an alternate key, the time required to move the hand and register a response would be minimal. However, such responses are susceptible to confounds related to hand preference. In a similar experimental task, subjects might be instructed to consistently use a particular hand when making responses and to rest that hand on a particular key centered between the two keys used for registering a response. In this case, the time taken to move the hand to a key to register a response is more substantial.

When considering the relationship of reaction times to other variables, the inclusion of movement time may contaminate any estimates of association. For example, if investigating age-related changes in reaction times measured for elementary cognitive tasks, such as the short-term memory scanning task (Sternberg, 1966), the inclusion of movement time in the measurement of reaction time could serve to artificially inflate age differences. It is common practice in the study of individual

differences, particularly in the investigation of the correlates of reaction time, to measure reaction time independently of movement time (e.g., Jensen, 1987; Jensen, Larson, & Paul, 1988; Miller & Vernon, 1992; Smith & Stanley, 1983; Vernon, 1990). However, to the best of our knowledge, timing routines to measure reaction time in this manner are not currently available for the Commodore 64 or 128.

We have modified the Hormann and Allen (1987) timing routine so that it measures reaction time while minimizing movement time. In the keyboard version of this routine, subjects begin each trial by pressing the space bar and are instructed not to release the space bar until they have decided on a response. The timing routine measures the time interval between presentation of the stimulus and release of the space bar. Therefore, the only movement included in the measurement of response time is the movement required to release the space bar. The routine also records which of two designated keys (the "a" or the "=") is pressed in response to the stimulus. On the Commodore 128 keyboard, "a" and "=" are equidistant from the space bar. When programming an experimental task, it is important to verify that the space bar is being pressed before each trial is presented. The keyboard version of the timer routine for use with the Commodore 128 can be loaded by using the BASIC program given in Listing 1. Users who prefer to have subjects register responses by using keys other than "a" and "=" can do so by replacing the values of 10 and 53 on line 113 of Listing 1 with the keyboard buffer code for the alternative keys. It should be noted that, like the original Hormann and Allen (1987) timer, this routine must be run in the 40-column screen mode of the Commodore 128. The following lines should be substituted when using the BASIC loader with the Commodore 64:

```
20  FOR I = 828 TO 939
109  DATA 166,203,134,252,224,60,240,245
112  DATA 133,251,32,159,255,166,203,134
```

A short program illustrating the use of the timer is given in Listing 2.

Hormann and Allen (1987) also provide a version of their timer for use with two external switches connected to the user port. These switches are connected to Pin K (PB6) and Pin L (PB7) of the user port. Our modified version of the user port timer requires a third switch to serve the same function as that of the space bar in the keyboard version. The BASIC loader (Listing 3) assumes that this third switch is connected to Pin J (PB5) of the user port. When the Commodore 64 is used, line 20 of the loader should read, FOR I = 828 TO 935.

Machine language code is provided for use with MONITOR, the machine language utility built into the Commodore 128. The keyboard version is given in Listing 4. When using the user port, the code given in Listing 5

Address correspondence to C. J. Lee, Department of Psychology, University of Western Ontario, London, ON, Canada N6A 5C2.

should be substituted for keyboard timing routine beginning at F0B3A in Listing 4.

To switch from the BASIC language interpreter (which greets the user on start up) to the built-in machine language interface, one simply types MONITOR and presses the return key. When first invoked, MONITOR displays the current status of the program status register, the program counter, the accumulator, the X index register, the Y index register, and the stack pointer. The assemble command (A) is used to translate assembly language instructions into machine language code. This command has three required parameters: start address, assembly language mnemonic, and operand. Optionally, a colon can serve as a logical end-of-line marker. In Listing 4, a colon is used to introduce comments. The first line of the timing routine is entered by typing the following and pressing the return key:

A F0B00 LDA #\$7F

In this instance, F0B00 is the hexadecimal number corresponding to the memory location where MONITOR will store the machine code produced by the assembly language mnemonic LDA #\$7F. One can examine a series of assembly language instructions by using the disassemble command (D). This command has two parameters: start address, and an optional end address. For instance, the machine code stored in memory between F0B00 and F0B08 can be disassembled by typing the following:

D F0B00 F0B08

The save command (S) is used to store the timing routine on disk. The parameters of the save command are filename, drive number, starting address of memory to be saved, and ending address of memory to be saved plus one. The keyboard timing routine is saved on disk as a binary file called RTXMT by typing the following:

S "RTXMT", 8, F0B00, F0B70

Having saved the timing routine to disk, one can easily load the timer binary file using the BLOAD command. The exit command (X) returns the user to the BASIC language interpreter.

REFERENCES

- HORMANN, C. A., & ALLEN, J. D. (1987). An accurate millisecond timer for the Commodore 64 or 128. *Behavior Research Methods, Instruments, & Computers*, **19**, 36-41.
- JENSEN, A. R. (1987). Process differences and individual differences in some cognitive tasks. *Intelligence*, **11**, 107-136.
- JENSEN, A. R., LARSON, G. E., & PAUL, S. M. (1988). Psychometric g and mental processing speed on a semantic verification test. *Personality & Individual Differences*, **9**, 243-255.
- MILLER, L. T., & VERNON, P. A. (1992). The general factor in short-term memory, intelligence, and reaction time. *Intelligence*, **16**, 5-29.
- SMITH, G. A., & STANLEY, G. (1983). Clocking g: Relating intelligence and measures of timed performance. *Intelligence*, **7**, 353-368.
- STERNBERG, S. (1966). High-speed memory scanning in human memory. *Science*, **153**, 652-654.
- VERNON, P. A. (1990). An overview of chronometric measures of intelligence. *School Psychology Review*, **19**, 399-410.

Listing 1 BASIC Loader for the C128 Keyboard Timer

```

10 REM BASIC LOADER FOR C128 KEYBOARD TIMER
20 BANK 15: FOR I = 2816 TO 2927
30 READ X: POKE I,X: NEXT
40 END
100 DATA 169,127,162,254,160,3,141,13
101 DATA 221,142,4,221,140,5,221,169
102 DATA 255,141,6,221,141,7,221,160
103 DATA 81,140,15,221,96,120,169,1
104 DATA 141,25,208,173,17,208,41,127
105 DATA 141,17,208,173,25,208,41,1
106 DATA 240,249,173,17,208,9,16,141
107 DATA 17,208
108 DATA 162,17,142,14,221,32,159,255
109 DATA 166,212,134,252,224,60,240,245
110 DATA 169,0,141,14,221,173,6,221,73
111 DATA 255,133,250,173,7,221,73,255
112 DATA 133,251,32,159,255,166,212,134
113 DATA 252,224,10,240,4,224,53,208,241
114 DATA 88,184,80,159

```

Listing 2 Program for Commodore 128 Demonstrating Use of Timer

```

10 REM DEMO PROGRAM
20 SCNCLR : COLOR 0,1 : COLOR 4,1 : COLOR 5,15
30 SYS 2816 : REM INITIALIZE TIMER
40 REM CHANGE LINE 30 TO SYS 828 WHEN C64 IS USED
50 DIM RT(8), KY(8), DIGIT(8)

```

```

60   POKE 53266, 138 : REM SET RASTLO (SEE HORMANN & ALLEN)
70   FOR T = 1 TO 8 : READ DIGIT(T) : NEXT T
80   DATA 3,8,6,1,4,9,5,2
90   L1$ = "PRESS THE 'A' KEY WHEN AN"
100  L2$ = "EVEN NUMBER APPEARS ON THE SCREEN."
110  L3$ = "PRESS THE '=' KEY WHEN AN"
120  L4$ = "ODD NUMBER APPEARS ON THE SCREEN."
130  L5$ = "DO NOT RELEASE THE SPACEBAR BEFORE"
140  L6$ = "YOU HAVE DECIDED ON YOUR RESPONSE."
150  L7$ = "PRESS THE SPACEBAR TO BEGIN."
160  L1 = INT((40-LEN(L1$))/2):PRINT SPC(L1);L1$
170  L2 = INT((40-LEN(L2$))/2):PRINT SPC(L2);L2$
180  L3 = INT((40-LEN(L3$))/2):PRINT SPC(L3);L3$
190  L4 = INT((40-LEN(L4$))/2):PRINT SPC(L4);L4$
200  L5 = INT((40-LEN(L5$))/2):PRINT SPC(L5);L5$
210  L6 = INT((40-LEN(L6$))/2):PRINT SPC(L6);L6$
220  L7 = INT((40-LEN(L7$))/2):PRINT SPC(L7);L7$
230  REM CHECK IF SPACEBAR IS BEING PRESSED
240  IF PEEK(212) <> 60 THEN GOTO 240
250  REM PEEK(203) WHEN C64 IS USED
260  FOR T = 1 TO 8
270  POKE 53265, PEEK(53265) AND 239 : REM BLANK SCREEN
280  SCNCLR
290  PRINT"[DWN][DWN][DWN][DWN][DWN][DWN][DWN][DWN][DWN][DWN]"
300  PRINT SPC(19);DIGIT(T)
310  TR = INT(RND(1)*60 + 1)
320  TC = 30 + TR: TO = TI: REM VARIABLE INTERTRIAL INTERVAL
330  T1 = TI:IF T1 - T0 < TC THEN GOTO 330
340  REM CHECK IF SPACEBAR IS BEING PRESSED
350  IF PEEK(212) <> 60 THEN GOTO 310
360  REM PEEK(203) INSTEAD WHEN C64 IS USED
370  SYS 2845 : REM UNBLANK SCREEN AND START TIMER
380  REM CHANGE LINE 370 TO SYS 857 WHEN C64 IS USED
390  RT(T) = PEEK(250) + 256*PEEK(251)
400  KY(T) = PEEK(252)
410  NEXT T
420  SCNCLR:PRINT SPC(8);"RT";SPC(7);"KEY"
430  FOR T = 1 TO 8
440  PRINT USING "#####";RT(T),KY(T)
450  NEXT T
460  END

```

Listing 3
BASIC Loader for the C128 User Port Timer

```

10   REM BASIC LOADER FOR C128 USER PORT TIMER
20   BANK 15: FOR I = 2816 TO 2923
30   READ X: POKE I,X: NEXT
40   END
100  DATA 169,127,162,254,160,3,141,13
101  DATA 221,142,4,221,140,5,221,169
102  DATA 255,141,6,221,141,7,221,160
103  DATA 81,140,15,221,96,120,169,1
104  DATA 141,25,208,173,17,208,41,127
105  DATA 141,17,208,173,25,208,41,1
106  DATA 240,249,173,17,208,9,16,141
107  DATA 17,208
108  DATA 162,17,142,14,221,173,1,221
109  DATA 133,252,201,223,240,247,169,0
110  DATA 141,14,221,173,6,221,73,255
111  DATA 133,250,173,7,221,73,255,133
112  DATA 251,173,1,221,133,252,201,191
113  DATA 240,4,201,127,208,243,88,184
114  DATA 80,163

```

Listing 4
Machine Language Code for C128 Keyboard Timer

```

:UNMODIFIED INITIALIZATION ROUTINE
FOB00 LDA    #$7F    :MASK TO DEFEAT CIA IRQ
FOB02 LDX    #$FE    :TIMER LO VALUE
FOB04 LDY    #$03    :TIMER HI VALUE
FOB06 STA    $DD0D    :DISABLE TIMER FROM IRQ
FOB09 STX    $DD04    :SET TIMER A TO 1 MSEC
FOB0C STY    $DD05
FOB0F LDA    #$FF    :SET UP COUNTER
FOB11 STA    $DD06    :RESET COUNTER TO FFFF
FOB14 STA    $DD07
FOB17 LDY    #$51
FOB19 STY    $DD0F    :ENABLE COUNTER
FOB1C RTS

:UNMODIFIED DISPLAY SYNCH ROUTINE
FOB1D SEI                      :TURN OFF SYSTEM IRQ
FOB1E LDA    #$01    :RESET RASTER IRQ FLAG
FOB20 STA    $D019
FOB23 LDA    $D011    :TURN OFF BIT 9 OF SCNLINE
FOB26 AND    #$7F
FOB28 STA    $D011
FOB2B LDA    $D019    :POLL RASTER IRQ FLAG
FOB2E AND    #$01    :IS RASTER AT DISPLAY LINE
FOB30 BEQ    $0B2B    :IF NO, KEEP POLLING
FOB32 LDA    $D011    :UNBLANK SCREEN
FOB35 ORA    #$10
FOB37 STA    $D011

:MODIFIED ROUTINE TIMING ROUTINE FOR KEYBOARD
FOB3A LDX    #$11    :START TIMER
FOB3C STX    $DD0E
FOB3F JSR    $FF9F    :CHECK KEYBOARD
FOB42 LDX    $D4
FOB44 STX    $FC
FOB46 CPX    #$3C    :IS "SPACEBAR" PRESSED?
FOB48 BEQ    $0B3F    :IF YES, CHECK KEYBOARD AGAIN
FOB4A LDA    #$00    :STOP TIMER
FOB4C STA    $DD0E
FOB4F LDA    $DD06    :GET COUNTER LO BYTE
FOB52 EOR    #$FF
FOB54 STA    $FA    :STORE LO BYTE
FOB56 LDA    $DD07    :GET COUNTER HI BYTE
FOB59 EOR    #$FF
FOB5B STA    $FB    :STORE HI BYTE
FOB5D JSR    $FF9F    :CHECK KEYBOARD
FOB60 LDX    $D4
FOB62 STX    $FC
FOB64 CPX    #$0A    :IS "A" PRESSED?
FOB66 BEQ    $0B6C    :IF YES, BRANCH TO FOB6C
FOB68 CPX    #$35    :IS "=" PRESSED?
FOB6A BNE    $0B5D    :IF NO, BRANCH TO FOB5D
FOB6C CLI    :RETURN IRQS TO SYSTEM
FOB6D CLV
FOB6E BVC    $0B0F    :BRANCH TO FOB0F

```

Listing 5
User Port Timing Routine

```

:MODIFIED TIMING ROUTINE FOR USER PORT
FOB3A LDX    #$11    :START TIMER
FOB3C STX    $DD0E
FOB3F LDA    $DD01    :CHECK USER PORT
FOB42 STA    $FC    :STORE VALUE
FOB44 CMP    #$DF    :IS "HOME" SWITCH PRESSED?

```

```

FOB46 BEQ    $0B3F    :IF YES, CHECK USER PORT AGAIN
FOB48 LDA    #$00     :STOP TIMER
FOB4A STA    $DD0E
FOB4D LDA    $DD06    :GET COUNTER LO BYTE
FOB50 EOR    #$FF
FOB52 STA    $FA      :STORE VALUE
FOB54 LDA    $DD07    :GET COUNTER HI BYTE
FOB57 EOR    #$FF
FOB59 STA    $FB      :STORE VALUE
FOB5B LDA    $DD01    :CHECK USER PORT
FOB5E STA    $FC
FOB60 CMP    #$BF     :IS "LEFT" SWITCH PRESSED?
FOB62 BEQ    $0B68    :IF YES, BRANCH TO FOB68
FOB64 CMP    #$7F     :IS "RIGHT" SWITCH PRESSED?
FOB66 BNE    $0B5B    :IF NO, BRANCH TO FOB5B
FOB68 CLI
FOB69 CLV
FOB6A BVC    $0B0F    :BRANCH TO FOB0F

```

(Manuscript received January 27, 1993;
revision accepted for publication April 11, 1994.)