*Behavior Research Methods, Instruments, & Computers*
*1984, 16 (3), 315-319*

# The KIM-1 microcomputer as an eight-channel data logger

D. JAMES CAMPBELL and JOHN FOREST
*The University of Wollongong, Wollongong, New South Wales, Australia*

**A machine language program and interface hardware to operate the KIM-1 microcomputer as an eight-channel data logger are described. All channels can operate simultaneously, and the data are stored in a specified area of memory. Double precision is used, enabling numbers greater than 255 to be recorded. Data can be collected over sampling intervals of 1 to 255 times 5 sec or 1 to 255 times 10 min. The time base is derived from 50- or 60-Hz mains ac. A method of transferring the data to a PET microcomputer and storing it on eight separate tape or disk files is also given.**

The equipment described here forms part of a system designed to monitor, over specified sampling intervals, the walking activity of field crickets (*Teleogryllus commodus*) in response to computer-generated cricket songs, as described by Campbell and Loher (1983). The crickets are housed in Perspex boxes and are provided with ample food and water. Each movement of a cricket from one end of its box to the other necessitates the crossing of two infrared light beams, which results in the output of a 0.5-sec TTL pulse from a 555 timer driving a 2N2222 transistor. This is sufficient to activate the relays of an Esterline Angus event recorder and can be used also as an input to the KIM-1 eight-channel data logger. The data collected can then be analyzed quantitatively. The equipment can operate at much faster frequencies than occur in this experimental arrangement, as is demonstrated in the Trial Sequence section.

A feature of the system is that the KIM display is always available for use and the contents of any address can be examined or changed if necessary while the program is running.

The program occupies memory space 0200 to 03E3, which leaves very little room for data in the unexpanded KIM, in which RAM finishes at address 03FF. Fortunately, the KIM is already decoded for an additional 4K of RAM memory, so expansion to this capacity is fairly simple, as is described in the user manual provided with the KIM-1. The latest version of the KIM-1[1] has an extra 3K of memory on board between addresses 0400 and 0FFF, which permits the storage of 192 values for each channel of the data logger. The system described assumes this extra 3K memory.

We thank Ben Parker for his help in the early stages of this project, John Murray for the diagram work, and Judy Ward for typing the manuscript. The authors' mailing address is: The University of Wollongong, Wollongong, N.S.W 2500, Australia.

## THE INTERFACE

### Data Input

The interface schematic is given in Figure 1. The eight inputs are via J2. Each input has a LED that is activated when an input pulse occurs. The LEDs facilitate seeing those channels that are recording activity. Each input is via a 4N28 optocoupler. Data then pass to the 74367 tristate input buffers (ICs 1 and 2), which stay high sufficiently long (about 35 microsec) to allow the data to be latched into a temporary data store (0004) for further processing by the computer. The input characteristics of 74367 buffers vary with manufacturer, and some do not work in this application. We used "SIGNETICS" 74LS367 tristate buffers, which gave no trouble because of their low input current bias requirements. The buffers are disabled by program control once the specified memory has been filled with data. When disabled, the buffers allow data transfer through the application port to another computer via a parallel bus. A save to tape can be achieved by disabling the interrupt latch, as described later. One-shots would have the advantage of eliminating the input characteristics problem of the buffers and would allow control of the input pulse length for the specific application, but parallel transfer of data through PORT A to another computer would not be possible because of bus contentions.

Use of the optocouplers is a precaution that is essential for blocking spurious spikes that occur when inductive loads are switched on and off near the equipment. The use of optocouplers for this purpose was discussed by Solomon, Weisz, Clark, Hall, and Babcock (1983). Such spikes can enter via the mains as well as the input leads. It is therefore wise to take the added precaution of isolating equipment such as cassette recorders with mains filters.

### Timing Circuit

The timing circuit consists of a series of counters

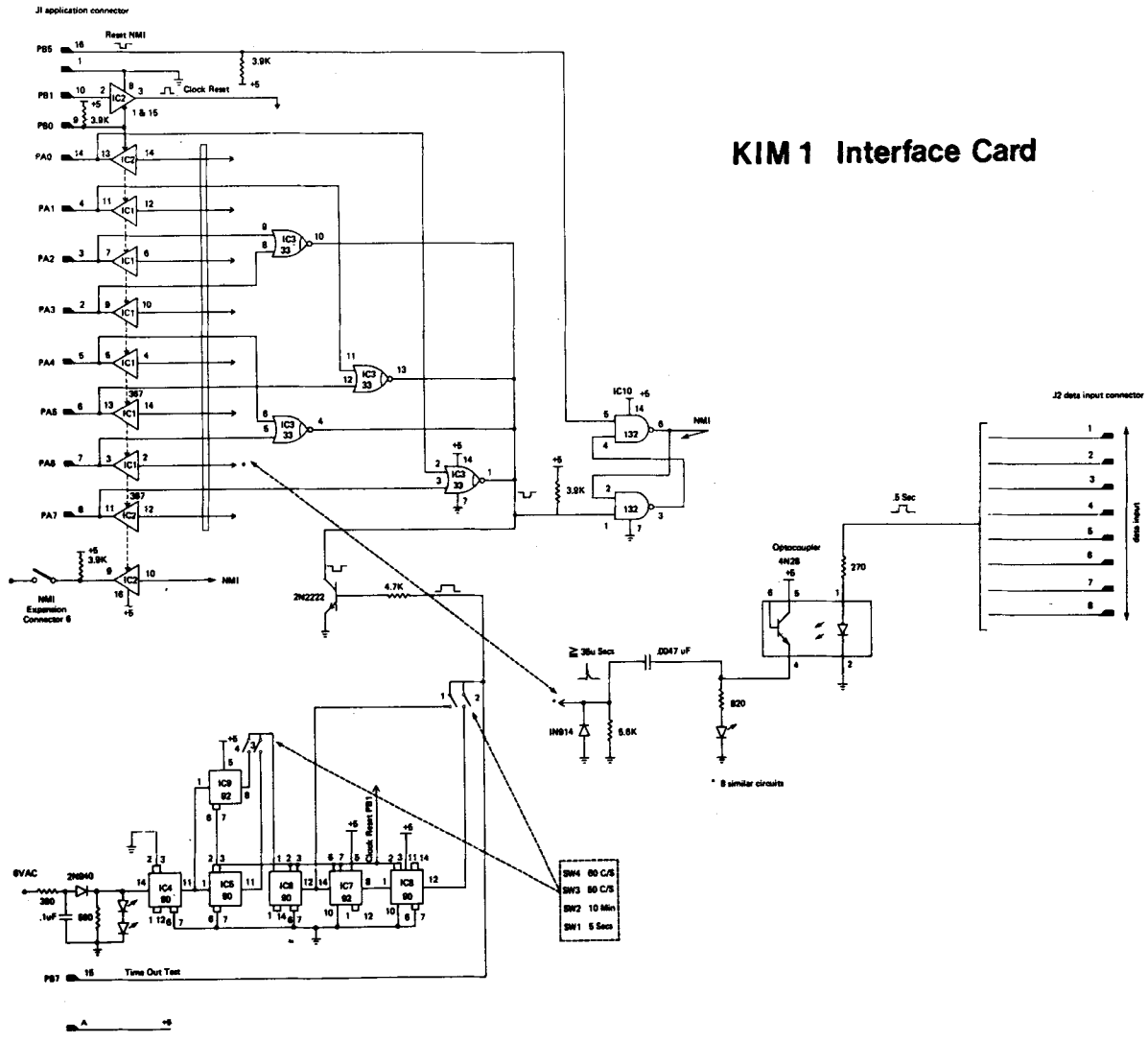


**Figure 1. The interface schematic for the KIM-1 eight-channel data logger.**

(7490s and 7492s). IC4 is wired to divide by 10 and couples with IC5 or IC9. These are wired to divide by 5 and 6, respectively. This produces a low of 0.5-sec duration, provided SW3 or SW4 is closed appropriately for either 50- or 60-Hz mains frequency. ICs 6, 7, and 8 are wired to divide by 10, 12, and 10, respectively. Thus, a low of 5-sec or 10-min duration can be selected, depending on whether SW1 or SW2 is closed. The low is inverted to a high by the 2N2222 transistor. When a time out occurs (when the timing line goes low), a pulse from the computer via IC2 provides sufficient drive to reset the clock, and the timing cycle starts again. Because the timing reset is not applied to IC4, real time is never lost.

## Interrupt Control

Unless a data line goes high or the timing line leaving the 2N2222 transistor goes low, the computer remains in a loop that activates the hexadecimal display. Either of these events produces a falling edge on the NMI line from the 74132 interrupt latch (IC10). The NMI, rather than the IRQ function, is used because the monitor subroutine PCCMD, which is used for displaying the contents of the program counter at the end of each program routine, sets the IRQ interrupt disable. An interrupt results in the KIM-1 momentarily leaving its display loop routine to process the timing and/or data input information. At the same time, the interrupt latch is disabled by the low on pin 1 coming from the timing line or from the 7433 (IC3) open collector NOR gates line, which is pulled low if any of the input buffers are high. Immediately after the data have been processed, pin 5 of the disabled interrupt latch is pulsed low by computer control, and the latch is again enabled. The KIM then returns to its display routine and waits for another interrupt. The interrupt latch minimizes the possibility of the computer stack overflowing in the event of mains glitches. It is also useful for other purposes, which are discussed later.

The microcomputer requires only microseconds to process incoming information. Thus, the chance of losing data when operating multichannels is negligible. A test for data loss is discussed later in the Trial Sequence section. Time outs are never lost because the NMI line from the interrupt latch pulses high when pin 5 is pulsed low by a NMI reset and pin 1 is low. Pin 1 is only low at this time if the clock has not been reset because a time out occurred during the data service routine. The falling edge of the pulse on the NMI line then initiates another interrupt. Data that otherwise would be lost can also be recovered by such multiple interrupts.

## THE PROGRAM

A listing of the machine language program is available on request. It is divided into a number of routines, which are discussed in turn. Unless otherwise stated, numbers are in hexadecimal notation.

### SET STORE CELLS AND POINTERS (0200)

In addition to providing information necessary for the correct running of the program, the purposes of this routine are to control the location of data to be stored and to specify the number of time outs per sampling interval.

The pointers specifying the starting and end addresses for data storage are located at addresses 0000 to 0003, low order first. This routine automatically places the values 00,04 and 00,10 in these locations, so that data are stored between addresses 0400 and 1000. When the specified memory area is filled with data, the input buffer and clock are disabled. The data storage area can be changed by altering the appropriate values between addresses 0209 and 0217 of this routine.

The number of time outs per required sampling interval is specified at address 0006. This routine automatically places 90 here. It can be changed at address 0219 of the routine. With the switches SW1 open and SW2 closed (only one switch should be closed at any one time), time outs occur every 10 min. This value is multiplied by 90 (decimal 144), giving a total sampling interval of 1 day.

### CLEAR DATA STORAGE AREA (023B)

Before data are stored in memory, it is necessary to ensure that the initial values are all zero. Data are then added cumulatively. The purpose of this routine is to place zeros (the value stored in 000A) in all the memory data storage locations as specified by the pointers in 0000 to 0003.

### CHECK DATA STORAGE AREA (026A)

This routine checks all the values in the specified data storage area. If a value other than zero occurs, its address is displayed, low order first. If there are no nonzeros, the next routine starting address (02A6) is displayed.

### INITIALISE (02A6)

Once the store cells and pointers have been set (routine 1) and the data storage area has been filled with zeros and then checked (routines 2 and 3), the program is ready to initialize and run. This sets the eight PORT A terminals (PA0 to PA7) as inputs (for data collection) and the seven PORT B terminals (PB6 is not available to the user) as either inputs or outputs, depending on requirements. PB7 is programmed as an input to test for time outs. PB0, PB1, PB3, and PB5 are programmed as outputs to function, respectively, as the input buffer enable/disable, the clock reset, a timing switch for external devices, and the NMI reset. On initialization, the input buffers and interrupt latch are enabled, the clock is reset, and PB3 for timing the external device is set low. The two PORT B pins not used (PB2 and PB4) are programmed as inputs. After initialization, the KIM sits in its display routine and waits for an interrupt. A reset on the KIM sets all ports as inputs and thus disables the input buffer and NMI latch and stops the clock.

### LATCH DATA (02CA); SERVICE DATA (02D6); SERVICE TIME (030C)

When an interrupt occurs, operation leaves the KIM display routine and enters the latch data routine. Initially, any data in the input buffers are transferred to the temporary data store at address 0004. Then operation moves to the service time routine to check for a time out. Finally, operation moves to the service data routine and then back to the KIM display routine.

### DISABLE/ENABLE INTERRUPT (0357)

It is sometimes necessary to disable the interrupt latch while the program is running. This prevents the NMI from functioning while an operation such as saving data to tape is being performed. This routine does not stop or reset the clock. When the operation is finished, the interrupt latch can be made operative again by the enable routine beginning at address 036A.

### PB3 TIME SWITCH (037E)

This routine was used as part of the cricket experiment mentioned in the introduction. It is included here to illustrate application possibilities. In the experiment (see Campbell & Loher, 1983), crickets are exposed to 2 h of computer-generated song between 10:00 a.m. and 12:00 noon each day. The song is turned on when PB3 goes high and off when it goes low. In this routine, PB3 goes high when the value in the time outs increment cell (0005) equals the value given in 000B (18 for 10:00 a.m.), assuming a starting time of 6:00 a.m., and low again when it equals the value given in 000C (24 for 12:00 noon). Since it is necessary to know the activity of the crickets specifically during this 2-h period of each day, the routine also partitions the data between the lower and upper half of memory at 0A00+ for activity while the song is on and at 0400+ at other times. Similar routines can

make use of PB2 and PB4 (still available) for other timing requirements, for example, for controlling the light/dark cycle.

### MANUAL CONTROL OF PB3 (03BE)

PB3 can be switched high or low using the routines beginning at addresses 03BE and 03DI, respectively. This facility has been included because it is occasionally very useful to be able to control an external device manually, as well as to have it under computer control.

## OPERATION

### Getting Started

If the routine "PB3 TIME SWITCH" is not required, it should be replaced with a 60 (RTS) or the values in 000B and 000C should be made equal. In that way, there is no partitioning of the data memory area. Once the pointers have been arranged for particular requirements and switches SW1, SW2, SW3, and SW4 have been set correctly, the start address (0200) of the program is keyed in. At this point, the reset (RS) button should be pressed and the NMI line switched on. Pressing GO three times should then take the user through the first three routines to the INITIALISE routine starting at address 02A6. Pressing GO the fourth time starts the operational program, and the time outs information is displayed at address 0005. When trying out the data logger, it is a good idea to have SW1 closed and SW2 open for a 5-sec time out. It is then simple to check the time outs incrementing at address 0005.

### Viewing Data

Data are initially recorded in memory at addresses 0400, 0401; 0402, 0403;... 040E, 040F. That is, eight double-precision hexadecimal numbers (one for each channel) occupy 16 memory locations. After 1 day (or timing interval), data are recorded at addresses 0410, 0411; 0412, 0413; etc. Thus, data for channel 1 go to addresses 0400, 0401; 0410, 0411; 0420, 0421; ..., and data for channel 2 go to 0402, 0403; 0412, 0413; 0422, 0423; ..., etc. These double-precision numbers are read high-order first. For example, 000F = 15. Note that the address for data storage is incremented by 10 each day (or sampling interval). Pointers for the current data recording address are given at 0000, 0001, and the current sampling interval number is given at address 0007.

### Transferring Data to a PET

Manually reading data directly from the KIM display is tedious and conducive to error. The suggested procedure for overcoming this difficulty is to transfer the logged data from the KIM to the PET by using the programs KIM-OUT and PET-IN (Campbell, 1983b). The data can then be filed on tape or disk for later analysis with the PET.

If it is necessary to save data on tape while the logger is running, the interrupt latch should be disabled with the subroutine at address 0357. It is best to do this just after a 10-min time out has occurred. When the data have been saved and before the next time out is due, the interrupt latch should be enabled with the subroutine at address 036A. All inputs are ignored, of course, while the interrupt latch is disabled.

Saving data on tape from the KIM is slow, but the whole process can be speeded up with the program HYPERTAPE (Butterfield, 1977). Occasionally, it may be necessary to adjust the KIM phase-locked loop, particularly when using HYPERTAPE. This is easily done with PLL SET (Edwards, 1977).

### Filing Data on PET Tape or Disk

A BASIC program (KIM-PET) for converting the logged hexadecimal double-precision data to decimal and saving it on eight PET tape or disk files, one for each channel, is available on request. A histogram and hard-copy printout of the data saved on these files can be obtained using the editor and high-resolution histogram plotter for the CBM/PET microcomputer (Campbell, 1983a).

## TRIAL SEQUENCE

Two trials were run to test the data logger. In the first test, a pulse generator driven by the 50-Hz mains frequency was applied to each of the channels of the data logger singly for four sampling intervals of 10-sec each. Thus, 500 events were expected for each sampling interval. The logged data are shown in Table 1. In the second trial, eight 555 pulse generators running at different frequencies were attached to the inputs of the data logger simultaneously for four sampling intervals of 1-min each. The experiment was then repeated applying the pulse generators one at a time to each of the channels. The data are shown in Table 2.

An examination of Table 1 shows that the data logger accurately recorded the expected number of events (500), except in the initial sampling interval when the number recorded is always lower than that expected. The reason for this is that the timing reset is not applied to IC4 (see the Timing Circuit section). Thus, IC4 can be anywhere along its count at the time of starting the INITIALISE routine. The counter divides by 10, and therefore the initial sampling interval can be out by 0.2 sec. All subsequent sampling intervals are correct.

**Table 1**
**Data (Number of Events) Logged Over Four Sampling Intervals (SI) of 10 Sec Each Using an Input Signal of 50 Pulses/Sec Applied to Each Channel Sequentially**

| | Channel | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| SI | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 496 | 494 | 493 | 490 | 497 | 490 | 497 | 499 |
| 2 | 500 | 500 | 500 | 500 | 500 | 500 | 500 | 500 |
| 3 | 500 | 500 | 500 | 500 | 500 | 500 | 500 | 500 |
| 4 | 500 | 500 | 500 | 500 | 500 | 500 | 500 | 500 |

Table 2
Data (Number of Events) Logged Over Four Sampling Intervals (SI) of 1-Min Each Using Eight Input Signals of Different Pulse Rate

| | Channel | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| SI | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| | | | | | (A) | | | |
| 1 | 28 | 26 | 66 | 66 | 278 | 287 | 198 | 171 |
| 2 | 27 | 27 | 65 | 66 | 278 | 287 | 198 | 171 |
| 3 | 28 | 26 | 65 | 66 | 278 | 288 | 196 | 171 |
| 4 | 27 | 27 | 65 | 65 | 277 | 290 | 197 | 172 |
| | | | | | (B) | | | |
| 1 | 28 | 27 | 66 | 66 | 282 | 288 | 198 | 172 |
| 2 | 27 | 26 | 65 | 67 | 279 | 290 | 198 | 171 |
| 3 | 28 | 26 | 66 | 66 | 278 | 289 | 198 | 172 |
| 4 | 27 | 27 | 66 | 66 | 279 | 289 | 197 | 172 |

*Note–In A the inputs were applied simultaneously, and in B they were applied sequentially.*

Table 2 provides a comparison of data recorded on the eight channels simultaneously and singly. There is very little difference between the two sets of results, indicating that the channels function effectively independently when operating simultaneously.

## AVAILABILITY

Listings, with comments, of the DATA LOGGER program and the KIM-PET program are available on request.

## REFERENCES


BUTTERFIELD, J. (1977). *HYPERTAPE.* In J. Butterfield, S. Ockers, & E. Rehnke (Eds.), *The first book of KIM.* Rochelle Park, NJ: Hayden.

CAMPBELL, D. J. (1983a). An editor and high-resolution histrogram plotter for the CBM/PET microcomputer. *Behavior Research Methods & Instrumentation*, **15**, 618-619.

CAMPBELL, D. J. (1983b). Machine language programs for high-speed transfer of data between the KIM-1 and the CBM/PET microcomputers. *Behavior Research Methods & Instrumentation*, **15**, 547-548.

CAMPBELL, D. J., & LOHER, W. (1983). A microcomputer-based modulator for simulating insect songs and the response of crickets to an artificial calling song. *Behavior Research Methods & Instrumentation*, **15**, 538-541.

EDWARDS, L. (1977). PLL SET. In J. Butterfield, S. Ockers, & E. Rehnke (Eds.), *The first book of KIM.* Rochelle Park, NJ: Hayden.

SOLOMON, P. R., WEISZ, D. J., CLARK, G. A., HALL, J., & BABCOCK, B. A. (1983). A microprocessor control system and solid state interface for controlling electrophysiological studies on conditioning. *Behavior Research Methods & Instrumentation*, **15**, 56-65.


## NOTES

1. The KIM-1 is currently available from Falk-Baker Associates, 382 Franklin Avenue, Nutley, NJ 07110. This machine is interchangeable with the original KIM, but it has an extra 3K of onboard RAM. The system ROMs and memory map are unaltered.

(Manuscript received October 26, 1983;
revision accepted for publication April 27, 1984.)