An interface for transmitting discrete response information to a computer*

ROLAND F. MILLER† and PETER G. POLSON
Computer Laboratory for Instruction in Psychological Research
University of Colorado, Boulder, Colorado 80302

A circuit is described for interfacing response keys and other discrete inputs from psychological apparatus to a computer. Some of the unique features are: high noise immunity, the ability to transmit data over long distances, and electrical protection of the computer from accidental and potentially catastrophic high voltage inputs.

The use of a real-time computer in the laboratory as a control and data recording device requires that the necessary stimulus displays and response inputs be interfaced to the computer (Uttal, 1968a, b). This paper discusses the general problems involved in recording discrete responses, e.g., buttonpresses, keypecks, the breaking of a photobeam, etc. First, we present the necessary functional properties of a discrete response interface for use on a multiprogramming system or with a program that runs several Ss concurrently. Second, an interface that satisfies these requirements is described in detail.

FUNCTIONAL REQUIREMENTS

A discrete response interface performs some or all of the following functions: (1) It transforms occurrences in the laboratory (a contact closure, a change in a voltage level) into events in the computer that have fixed electrical and temporal parameters. (In other words, an interface defines what a response is to the computer.) (2) An interface should prevent electrical artifacts from being recorded as inputs from a S and protect the computer's circuitry from damage by the accidental input of high voltages. (3) It should be designed to minimize the software necessary for response recording by simplifying the task of response detection and identification. Similar criteria have been discussed by Uttal (1968b) and McLean (1970).

*The design and construction of the interface unit described in this paper and the preparation of this report was supported by National Science Foundation Grant GJ 453 to Dr. Daniel E. Bailey.

†Requests for reprints should be sent to Roland Miller, 1511 University Avenue, Room 4, Computer Laboratory for Instruction in Psychological Research, University of Colorado, Boulder, Colorado 80302. Upon request, printed circuit art work and construction instructions will be included. Material cost is less than $175 per complete unit.

Specific Criteria Underlying Interface Design

The interface transforms all inputs into pulses of a fixed duration. A response (an input) is defined by either the closure of a single-pole double-throw switch or the positive transition of a voltage level. The interface eliminates any information about response duration and only provides the user with information about response onset. Information concerning the occurrence of a given response is all that is of interest in most experimental paradigms. Furthermore, transmitting response duration information can increase the complexity of the response recording software because of problems involved in deciding whether or not seemingly continuous activity of an input line represents one, two, or more responses with short interresponse times. An E can record response duration information by placing two sets of contacts on a manipulandum so that one is made on depression of the bar and the other is made on release of the bar. In summary, defining inputs as only response occurrences simplifies the software required to process input information. Response duration data can be obtained by recording the time between the occurrence of two subresponses.

When the signal source is a switch, relay, or other contact transducer, three problems are involved in transmitting input information to the computer: (1) contact bounce must be eliminated from the input signal; (2) the interface should permit a user whose laboratory is a long distance away from the computer (1,000 ft) to transmit signals to the computer reliably; and (3) the interface should not respond to electrical noise. Only the latter two requirements need be satisfied if the signal source is a change in voltage level. Of course, the interface must also protect the computer from accidental connection of an input line to a high voltage source. It is possible that in the process of checkout a new piece of equipment (or the breakdown of a piece of equipment) may cause a response input line to be connected to a high voltage source, e.g., 110 V or the output of a shocker. The interface must protect the computer from such low-probability but potentially catastrophic accidents.

The number of Ss and/or experiments that can be run concurrently on a real-time computing system is in part determined by the speed with which the computer can detect the occurrence of and identify a response. If a system is unable to react rapidly to real-time inputs, the results can be inaccurate response latencies and/or intermittent failures to record inputs. The interface to be described assumes that the computer has a collection of n-bit external registers that can be read and cleared under program control. Outputs from the interface to the computer set bits in these registers. A unique bit is associated with a given input line; all inputs from one S or an experiment go into one register. When an input occurs on any one of the lines connected to one of the registers, the interface generates a second signal which is sent to the computer's interrupt system. Thus, the computer responds to an input into a given register by trapping to a unique location, if it has a multilevel priority interrupt system. In computers with single-level interrupt systems, e.g., the DEC PDP-8, this second signal must in addition set a flag that can be later interrogated under program control to identify the input register.

The above scheme for inputting discrete responses has several advantages. The computer can detect rapidly the occurrence of responses through the use of its interrupt system, making it possible to accurately measure response latencies. Given that the interrupt system indicates the occurrence of a response, and which S or experiment generated the response, the designated input register contents are read in and decoded to discover which response occurred. Connecting each input to a unique interrupt level is a bad allocation of a computer's limited number of interrupt levels. The input signals are stored in the register until an input subroutine reads the contents of the register and then clears it. Thus, if the computer is performing some higher priority task, its reaction to the occurrence of an input can be delayed but the information will not be lost. In the section that follows, we describe an input circuit that has the functional characteristics outlined in the preceding paragraphs.

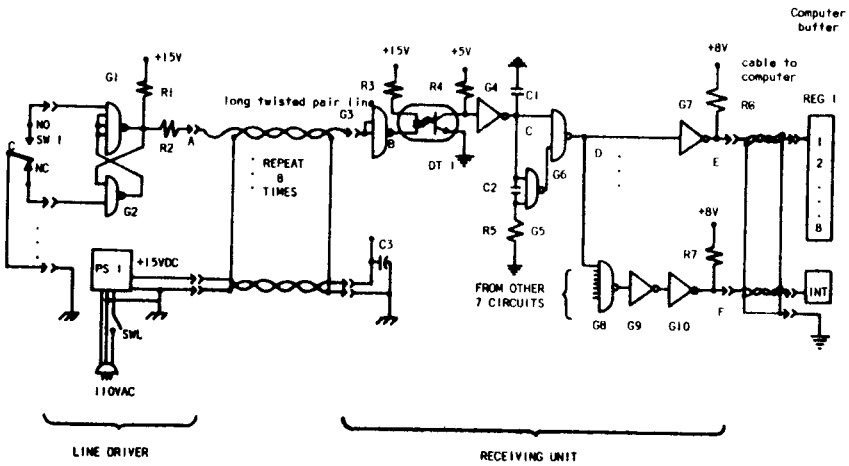

Fig. 1. Schematic diagram of discrete input computer interface.

Table 1
Parts List

Type	Description
SW 1	Single pole double throw switch "dry" circuit quality
R1	620 ½ w 5%
R2	100 ½ w 5%
R3	1K ¼ w 5%
R4	5.1K ¼ w 5%
R5	10K ¼ w 5%
R6	2K ¼ w 5%
R7	2K ¼ w 5%
C1	300 pf disc 5%
C2	2200 pf disc 5%
C3	10 μf 25 wvdc tantalum capacitor
G1	MC 662P Motorola HTL dual line driver
G2	MC 672P Motorola HTL quad NAND
G3	MC 672P Motorola HTL quad NAND
G4	MC 836P Motorola DTL hex inverter
G5	MC 846P Motorola DTL quad NAND
G6	MC 846P Motorola DTL quad NAND
G7	MC 1820P Motorola DTL hex inverter open collector
G8	MC 1802P Motorola DTL 8 input NAND
G9	MC 836P Motorola DTL hex inverter
G10	MC 1820P Motorola DTL hex inverter open collector
DT1	MCT2 Monsanto light-emitting diode—photo transistor isolator

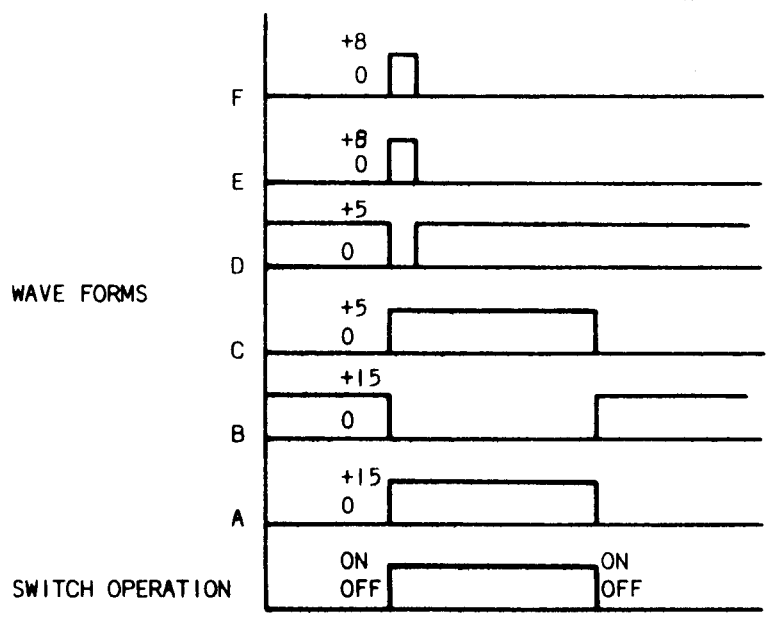

Fig. 2. Signal waveforms for specified points in the circuit shown in Fig. 1.

THE INTERFACE[1]

Physically, the interface comprises two units: a line driver and a receiver. A schematic of the complete input circuit is shown in Fig. 1; Fig. 2 shows circuit waveforms. A parts list is presented in Table 1.

The Line Driver

The sets of single-pole double-throw contacts associated with each button or other manipulandum are connected to one of the input circuits in the line-driving unit, as shown in Fig. 1. These signals set and reset a simple R-S flip-flop formed by Gates G1 and G2. Gate G1 drives the signal on a twisted pair. Resistors R1 and R2 match the driver impedances to the transmission line properties of the twisted-pair cable. High-threshold 15-V logic (Gates G1, G2, and G3) has a very high level of noise rejection. Twisted pairs help reject noise generated by relays, electric motors, power lines, etc. A 15-V power supply for these circuits is part of the line-driving unit. This unit performs the following functions: contact bounce elimination, transmission of signals over long distances (up to 1,000 ft), and noise rejection.

The Receiving Unit

A signal line is terminated by G3, which inverts and squares the signal. The use of high-threshold 15-V logic at both ends of the twisted pair increases the noise immunity of the input circuit. The output of G3 drives DT1, an optical isolator made up of a light-emitting diode-photo transistor pair. DT1 has two functions. The input signal is converted to a light beam by the light-emitting diode and travels through an insulator to a photo transistor. Thus, there is no electrical connection between the signal line from the line-driving unit and the computer. The protective isolation is greater than 10^{10} ohms and 1,000 V before breakdown of the insulator. DT1 also converts the signals to the 5-V logic level used in the remainder of the interface circuit.

The variable-length input signal is converted to a 2.5- to 3.0-microsec pulse by the pulse generator, G5, G6, R5, R6, and C2. Capacitor C1 damps any tendency for the circuit to oscillate. The standard width negative pulses at Point D are OR-ed together by Gate G8. Gates G9 and G10 perform the necessary inversion and level shifting necessary for the computer's interrupt system, INT. The signal at Point D in Fig. 1 goes through Gate G7 and sets a unique bit in REG, an external register that can be read and cleared by the computer. Gates G7 and G10 are open collector inverters which, together with the

associated pullup Resistors R6 and R7, convert the outputs of the interface to the voltage levels required by the computer.

The circuit shown in Fig. 1 was designed to be used with Xerox Data System's Sigma 3. Each input register, REG, is an 8-bit buffer. The logic level necessary to set bits in the input register and trigger the interrupt system is 8 V. A user can modify the interface for use with machines with 5-V logic, e.g., DEC PDP-8/I or -8/e, NOVA, levels by supplying +5 V rather than +8 V to the top of R6 in Fig. 1. By changing this voltage, output pulses from +5 to +30 V may be supplied to any external buffer.

Users who wish to use logic voltages may supply logic signals from either a +12- or a +15-V apparatus. Complementary signals may be applied to the input of the line driver in place of the mechanical switch. A single-level +12- or +15-V logic signal may be applied at the input to G3. If long input lines, i.e., longer than 100 ft are used, twisted pair and the line driver should be used for noise rejection.

AN EVALUATION

Most real-time applications involving discrete input do not require all of the features of the elaborate interface circuit described in the preceding paragraphs. However, having all experiments that are controlled by a real-time computing system interfaced to the computer through such a sophisticated unit has numerous advantages. First, the complex interface hardware simplifies the routines required to process discrete response information. There will need to be only one set of discrete response processing routines in the computer if all such inputs come in through one standard interface. Second, an E need not be concerned with possible electrical artifacts generated by the equipment connected to the computer or with noise sources located between his laboratory and the computer. Finally, the interface provides a very high level of protection against potentially catastrophic electric accidents.

REFERENCES

McLEAN, R. S. Variations of basic black computer interfaces. Behavior Research Methods & Instrumentation, 1970, 2, 31-32.

UTTAL, W. R. *Real time computers, techniques and applications in the psychological sciences.* New York: Harper & Row, 1968a.

UTTAL, W. R. Basic black in computer interfaces for psychological research. Behavior Research Methods & Instrumentation, 1968b, 1, 35-40.

NOTE

1. Readers interested in purchasing printed circuit boards or complete units like the interface described in the following section should contact Jim Geidel, Life Systems Electronics, 7845 N. 41st Street, Longmont, Colorado 80501, for further information.